

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

**Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής**



Πτυχιακή Εργασία

«ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΘΗΜΕΡΙΝΩΝ ΑΝΑΓΚΩΝ»

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ : ΚΑΡΡΑΣ ΑΠΟΣΤΟΛΗΣ
ΑΜ:3856**

ΕΙΣΗΓΗΤΗΣ: ΠΑΠΑΔΑΚΗΣ ΝΙΚΟΛΑΟΣ

Ευχαριστίες

Με την ολοκλήρωση αυτής της πτυχιακής θα ήθελα να ευχαριστήσω όλους του παράγοντες που με βοήθησαν στην ολοκλήρωση των σπουδών μου και της πτυχιακής μου εργασίας επίσης θέλω να ευχαριστήσω ιδιαιτέρως τους συναδέλφους που με βοήθησαν να βελτιωθώ σαν φοιτητής και τέλος θα ήθελα να ευχαριστήσω τους ανθρώπους που συμβάλλαν με τις πληροφορίες τους καθότι χωρίς την πολύτιμη και καθοριστική τους βοήθεια δεν θα ήταν δυνατή η ολοκλήρωση αυτής της πτυχιακής και η ολοκλήρωση των σπουδών.

Πάνω από όλα θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή της πτυχιακής μου Δρ. Παπαδάκη Νικόλαο τον οποίο θαυμάζω και εκτιμώ καθότι η βοήθεια το ήταν πολύ σημαντική στην πορεία της πτυχιακής μου εργασίας.

Abstract

This thesis with the subject “Management System Of Daily Necessities” develops the design for an application which provides the users with the ability to find any store which they are interested in. This application requires first to sign up using a Username and a Password and second to sign in. Then it will give the opportunity via a Real Time Map (Google Maps) achieve discover places they interest him, while and many several information’s as the estimated time of arrival from our location to selected point, the route, the name, the vicinity and the schedule of the point. At the continuity through an Algorithm and the help of Google Api it will rank with an order the places of interest so that user can choose the nearest shop from him and afterward route will become more distant. Of course the user can have a choice to select from Algorithm or from his map and continue with his meeting. If his current liabilities allow user can close the meeting, this happen if he have not a meeting at same time or the shop is open. At the final activity user have possibility to set route by the layout of meetings or set an alarm. To accomplish this I used tools like: Android Studio, JAVA, Google Api android, PHP, sql (online database),sqlite, joda timer,Adode Photoshop, Adode After effects.

Σύνοψη

Στην παρούσα πτυχιακή εργασία με θέμα «Σύστημα διαχείρισης καθημερινών αναγκών» αναπτύσσεται ο σχεδιασμός μιας εφαρμογής η οποία παρέχει την δυνατότητα σε χρήστες να κάνουν εύρεση οποιουδήποτε καταστήματος που τους ενδιαφέρει. Η εφαρμογή αυτή απαιτεί την εγγραφή και την είσοδο χρήστη με την μορφή username και password. Έπειτα, θα δίνει την δυνατότητα στον χρήστη μέσω ενός πραγματικού χάρτη (Real-time map) να πραγματοποιήσει εύρεση σημείων που τον ενδιαφέρουν καθώς κι άλλες διαφορές πληροφορίες όπως την ώρα διαδρομής, την απόσταση, το όνομα, την περιοχή και το ωράριο του σημείου αυτού. Στην συνέχεια, μέσω ενός αλγορίθμου και με την βοήθεια του Google Api θα κατατάσσει τα σημεία σε μια σειρά που θα επιλέξει ο εκάστοτε χρήστης της εφαρμογής αυτής έτσι ώστε να μπορεί να διαλέξει το πλησιέστερο κατάστημα προς τον ίδιο και στην συνέχεια θα απομακρύνεται. Όταν ο χρήστης διαλέξει το σημείο που επιθυμεί μπορεί να μεταβεί μέσω του αλγορίθμου ή μέσω της δικιάς του επιλογής για να διαλέξει την ημερομηνία συνάντησης. Αν οι υποχρεώσεις του το επιτρέπουν δηλαδή αν δεν έχει ραντεβού στην ίδια ώρα ή το κατάστημα είναι ανοικτό θα κατατάσσεται στις συναντήσεις του. Τέλος ο χρήστης έχει την δυνατότητα να θέσει την δρομολόγηση αυτή μέσω της καρτέλας των συναντήσεων του ή να θέσει μια υπενθύμιση. Για να επιτευχθούν όλα αυτά θα χρησιμοποιηθούν τα παρακάτω εργαλεία: Android Studio, JAVA, Google Api android, PHP, sql (online database),sqlite, joda timer, Adobe Photoshop, Adobe After effects.

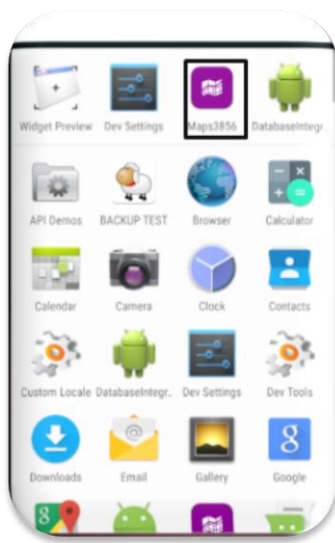
Αναλυτικά τι θα παρουσιαστεί:

1. Εισαγωγή στην εφαρμογή μέσω ενός Intro video.
2. Αρχικό layer όπου ο χρήστης θα κάνει είσοδο η εγγραφή.
3. Είσοδος και επιλογές επανασύνδεσης, μεταφορά στον χάρτη ή είσοδος σε συναντήσεις.
4. Χάρτης 2 μέθοδοι αναζητήσεις σημείων αυτόματα ή χειροκίνητα .
5. Navigation drawers (swipe left or right) δεξιά και αριστερά όπου ο δεξιάς θα εκτελεί τον αλγόριθμο και θα επιλέγει ο χρήστης το σημείο μετάβασης Grocery, Cinema ή Gas Shop και θα τον μεταβιβάσει από το ποιο κοντινό στο ποιο μακρινό για κάθε κατηγορία, ο αριστερός θα δίνει πληροφορίες σχετικά με το πρόγραμμα του καταστήματος και μια σχετική φωτογραφία αν υπάρχει στους server της google.
6. Επιλογή ενός σημείου που θα δείχνει στο layer το όνομα του και την περιοχή του.
7. Στο layer όπου εμφανίζεται ο χάρτης θα παραθέτονται δυο κουμπιά ένα όπου θα κάνει set Route διαδρομής και ένα Set Meeting.
8. Το κουμπί Set Route θα θέτει την διαδρομή και στο layer θα εμφανίζεται η

απόσταση και η ώρα διαδρομής.

9. Το κουμπί Set Meeting θα τον οδηγεί στο layer όπου θα του παρουσιάζεται πάλι το πρόγραμμα του καταστήματος και θα θέτει ημερομηνία και χρόνο παραμονής .Αφού επιτρέπεται και δεν έχει άλλη υποχρέωση η είναι ανοικτό το κατάστημα θα θέτει την συνάντηση.
10. Στην συνέχεια θα μεταβαίνει στην καρτέλα με τις συναντήσεις όπου θα επιλεγεί μέσω της συνάντησης ή να την επαναθέτει στον χάρτη ή να βάλει σημείωση .
11. Στο layer που έχουμε τον χάρη θα έχει ένα spinner όπου θα επιλεγεί αν θέλει έξοδο ή να δει τα σημεία συνάντησης.
12. Εμπλουτισμός Εφαρμογής με εικόνες, custom buttons κτλπ.

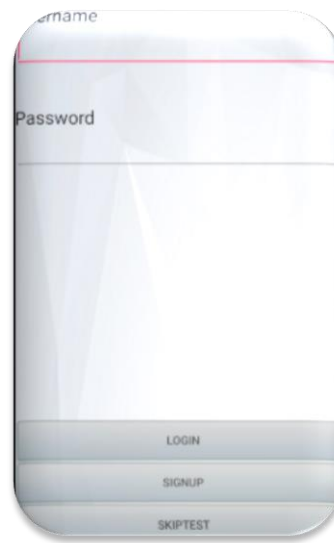
Συγκεκριμένα μερικές φωτογραφίες μέσα από την εφαρμογή (οι αριθμοί αναθέτουν πάνω):



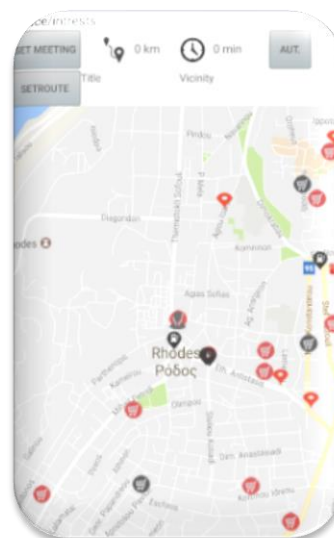
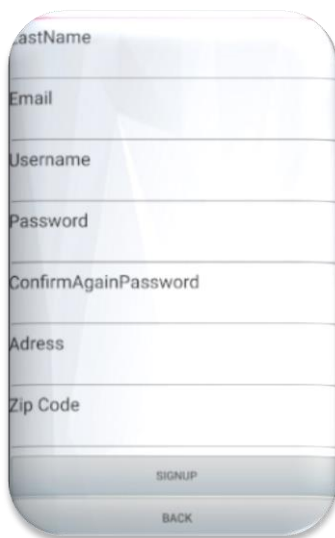
Logo icon



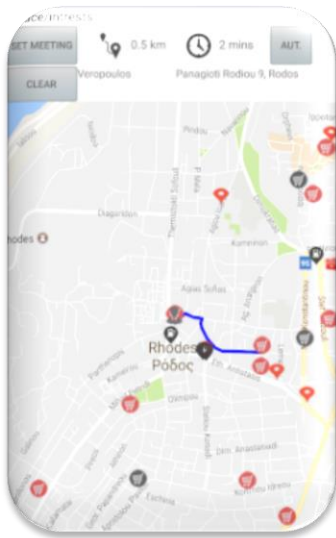
1



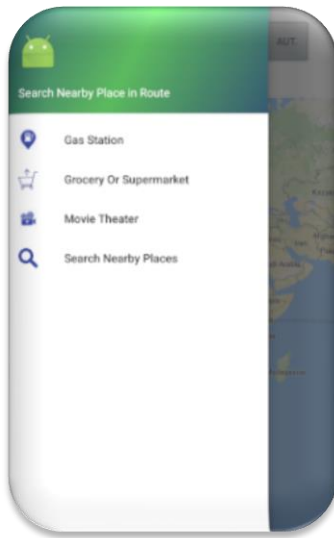
2



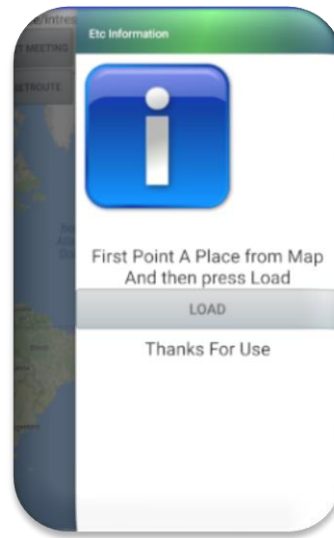
2



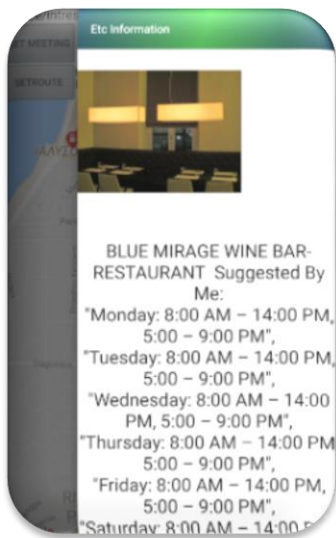
3



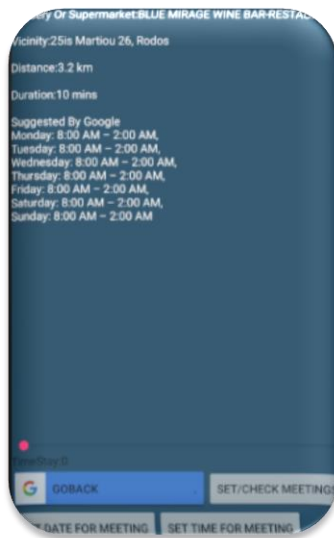
4,6,7,11



4,6,7,8,11



5



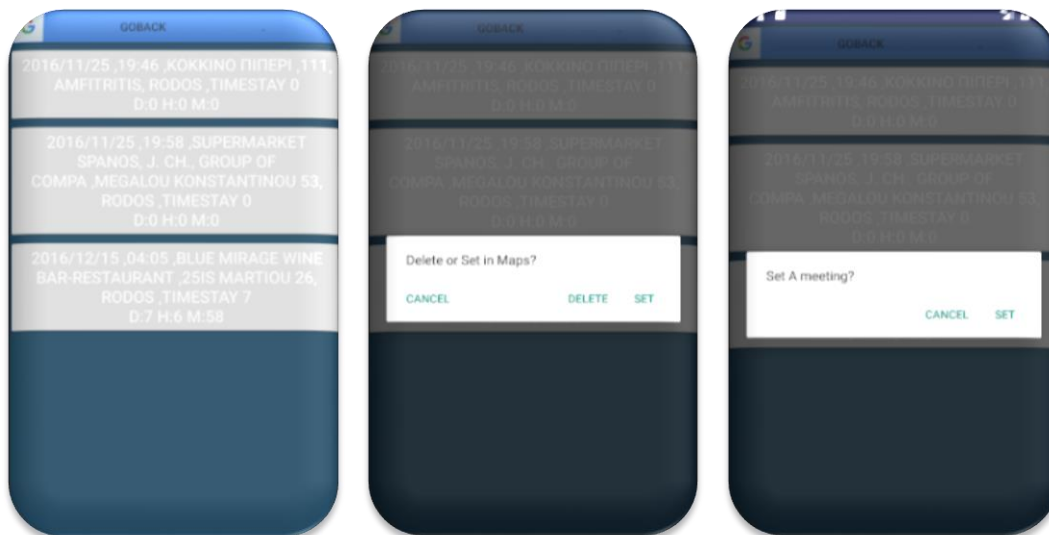
5



5

9

9



10

10

10

Ο σκοπός:

Ο βασικός σκοπός της πτυχιακής αυτής είναι η καλύτερη δυνατή παρουσίαση των πραγματικών συνθηκών και προκλήσεων που καλείται να αντιμετωπίσει ένας μηχανικός λογισμικού τόσο στον σχεδιασμό, στις απαιτήσεις του σήμερα όσο και στην εφαρμογή αυτού. Αυτό περιλαμβάνει την πραγματοποίηση μια σύγχρονης και λειτουργικά σωστής εφαρμογής όπου θα προσφέρει ανετότητα στην χρήση και εύκολη προσβασιμότητα με άμεσο σκοπό την εκπλήρωση των αναγκών των χρηστών στα πλαίσια της καταναλωτικής αγοράς.

Πίνακας Περιεχομένων.

- ❖ Δημιουργία αρχικού intro.....8
 - After Effects.....9
 - Photoshop.....11
 - After Effects ΣΥΝΕΧΕΙΑ.....15
- ❖ Δημιουργία online database.....28
- ❖ Δημιουργία φηρ αρχείων για την επικοινωνία με την online database.....32
 - ΚΩΔΙΚΑΣ.....32
 - ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ (login)32
 - ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ (register).....34
- ❖ Δημιουργία Google Api Key.....36
- ❖ Σχήμα εφαρμογής τι κάνει η κάθε κλάση και οι κυρίως μεταβλητές.....38
 - Επεξήγηση.....38
 - Static Variables.....42
 - Τι είναι το activity.....43
 - Flow Diagram.....44

❖ Δημιουργία εφαρμογής.....	47
❖ Δημιουργία του project.....	47
Αρχείο Manifest.....	50
build.gradle.....	53
Intro.....	55
Layout intro.....	55
Class Intro.....	57
Layout signin.....	59
Class Signin.....	63
Class LoginRequest.....	66
Layout signup.....	67
Class Signup.....	70
Class RegisterUserClass.....	75
Layout welcome.....	78
Class Welcome.....	80
Layout navheader_main1.....	85
Layout activity_main_drawer.....	86
Class NavigationLeft.....	87
Layout nav_header_main2.....	103
Layout navi_right.....	104
Class NaviRightLoader.....	105
Class DownloadImageTask.....	111
Layout placeandintrests.....	113
Class PlaceAndIntrests.....	119
Class FindPlaces.....	142
Class VariablesFindPlaces.....	157
Class GetYourPlace.....	158
Class MarkerListener.....	159
Class SetRoute.....	162
Class SetRouteVariables.....	173
Layout activity_maps.....	173
Class RouteOnly.....	175
Class DirectionFinder.....	180
Interface DirectionFinderListener.....	183
Class Distance.....	184
Class Duration.....	184
Class Route.....	184
Class GetPlaceSchedule.....	185
Class GetStringShopSuggested.....	190
Class SetMeetingsVariables.....	192
Layout setmeeting.....	193
Class SetMeeting.....	195
Class Database.....	213
Class DatabaseInterface.....	222
Layout checkseemeetings.....	235
Class CheckSeeMeetings.....	236
Class MyAlarmService.....	251
ΕΙΚΟΝΗΔΙΟ ΕΦΑΡΜΟΓΗΣ.....	253

Δημιουργία αρχικού intro.

Με την χρήση δυο προγραμμάτων το adobe after effects και το photoshop.

Το Adobe After Effects είναι μια εφαρμογή που δημιουργεί ψηφιακά οπτικά εφέ και κινούμενα γραφικά όπου αναπτύχθηκε από την Adobe Systems και χρησιμοποιείται στο post-production διαδικασία λήψης ταινιών και τηλεοπτικών παραγωγών. Μεταξύ άλλων, το After adobe Effects μπορεί να χρησιμοποιηθεί για την παρακολούθηση, compositing και animation. Επίσης, λειτουργεί ως ένας πολύ βασικός μη γραμμικός συντάκτης, επεξεργαστής ήχου και επανακωδικοποιητής πολυμέσων.

Το Photoshop, είναι ένα πρόγραμμα επεξεργασίας γραφικών που αναπτύχθηκε και κυκλοφόρησε από την Adobe Systems. Αυτή τη στιγμή αποτελεί σημαντικό πρόγραμμα επεξεργασίας εικόνων, και είναι το προϊόν - σήμα κατατεθέν της Adobe Systems.

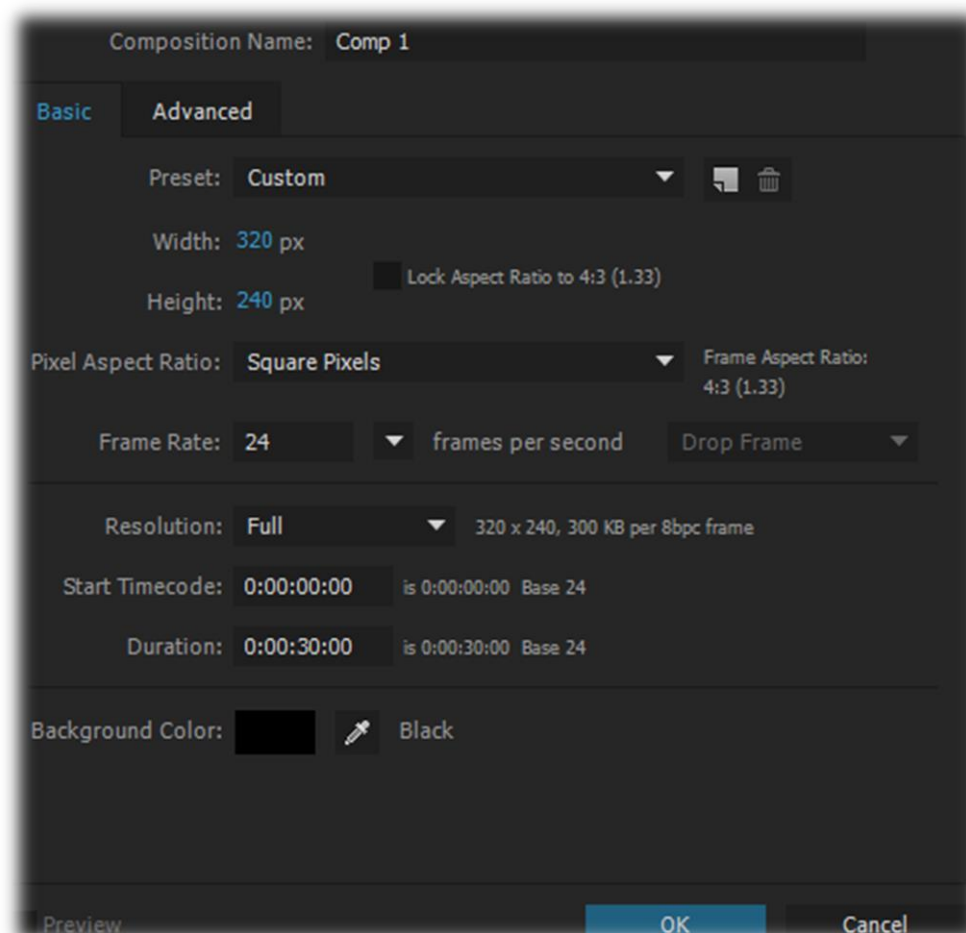
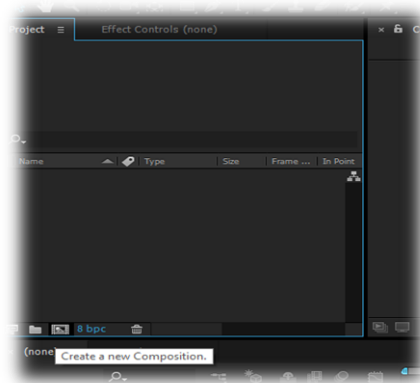
Τα προγράμματα μπορεί κανείς να τα βρει στην επίσημη σελίδα <http://www.adobe.com/> σαν δωρεάν δόκιμες για κάποιο διάστημα.

Πριν ξεκινήσω να περιγραφώ την διαδικασία παραθέτω κάποια στιγμιότυπα από κάτω για να προαναφέρω τι θα ειπωθεί.



After Effects

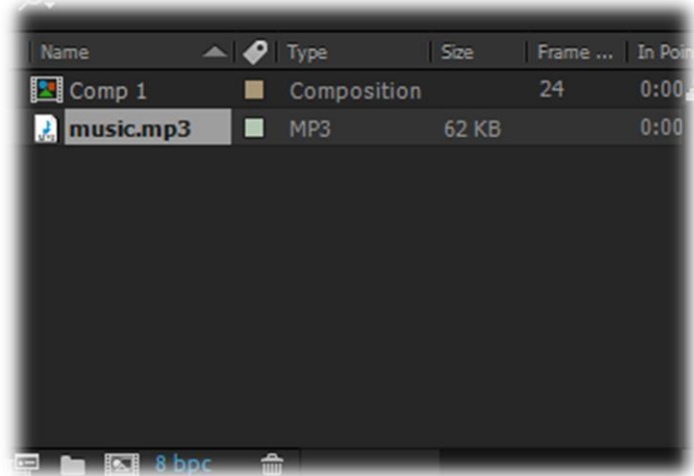
- Πρώτο Βήμα είναι να φτιάξουμε ένα νέο composition.



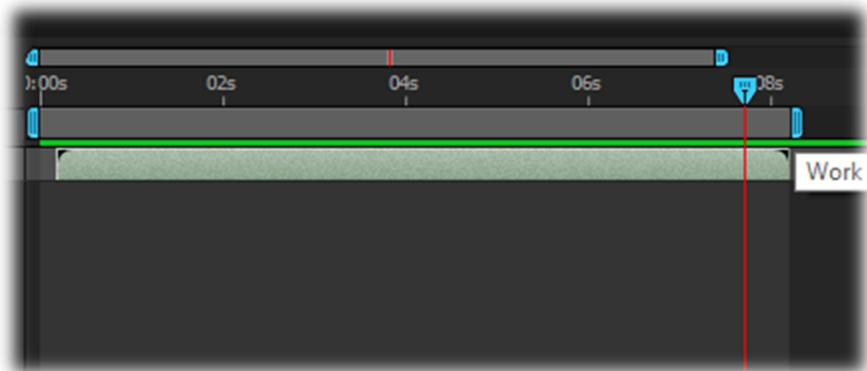
Στην παραπάνω εικόνα παρουσιάζονται μερικές επιλογές όπως fps background color

width height κτλπ.

Για να προσθέσουμε νέα αντικείμενα κάνουμε drag and drop.



Στην κάτω μπάρα μπορούμε να καθορίσουμε το πόσο θα διαρκέσει το animation στην δίκια μας περίπτωση θέλουμε να διαρκέσει περίπου όσο ο ήχος 8 περίπου δευτερόλεπτα.



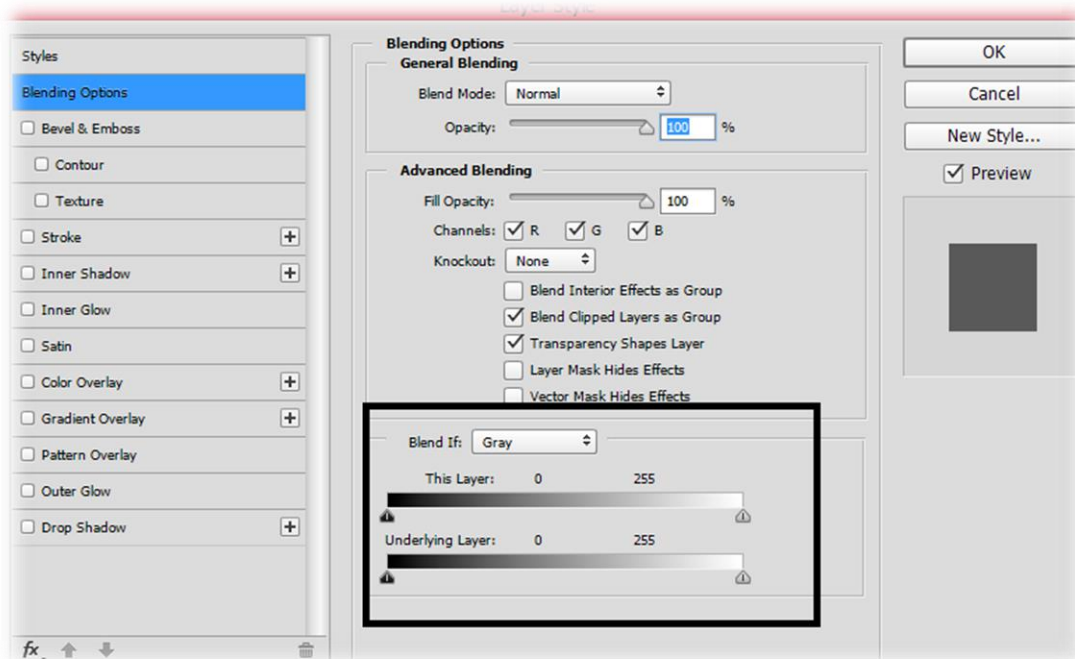
Μέχρι στιγμής έχει φτιαχτεί ένα μπλε layer με έναν ήχο στην συνέχεια ακολουθεί η διαδικασία του animation όμως πρώτα θέλουμε και εικόνα, τώρα θα μεταβούμε στο Photoshop οπού θα διαμορφώσουμε μια κατάλληλη εικόνα για ένα μπλε layer και η εικόνα που θα χρησιμοποιήσω θα είναι του TEI Κρήτης.

Photoshop

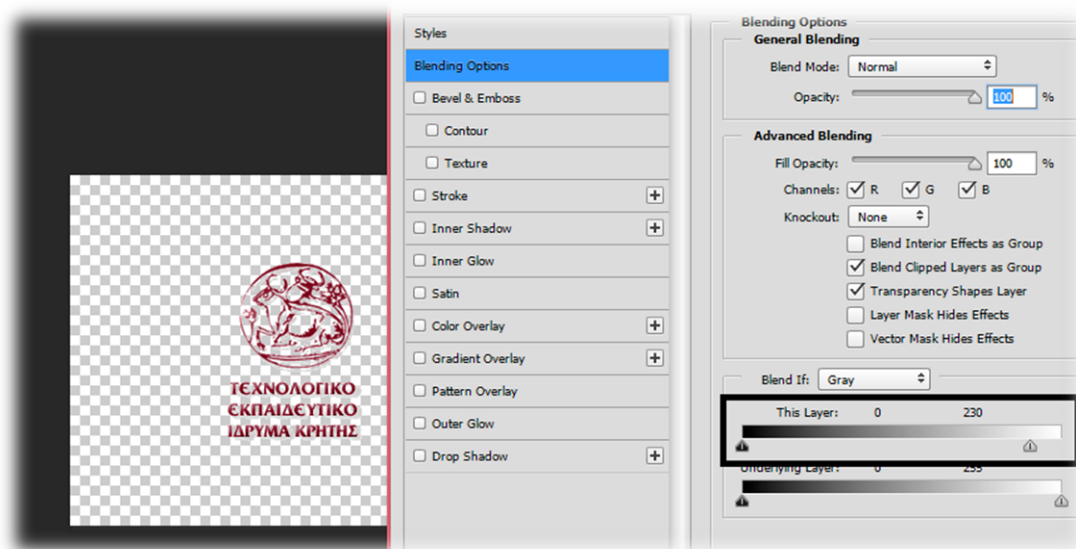
Πάλι όπως και στο Adobe σέρνουμε την εικόνα στο Photoshop και προσαρμόζουμε το μέγεθος του καμβά.



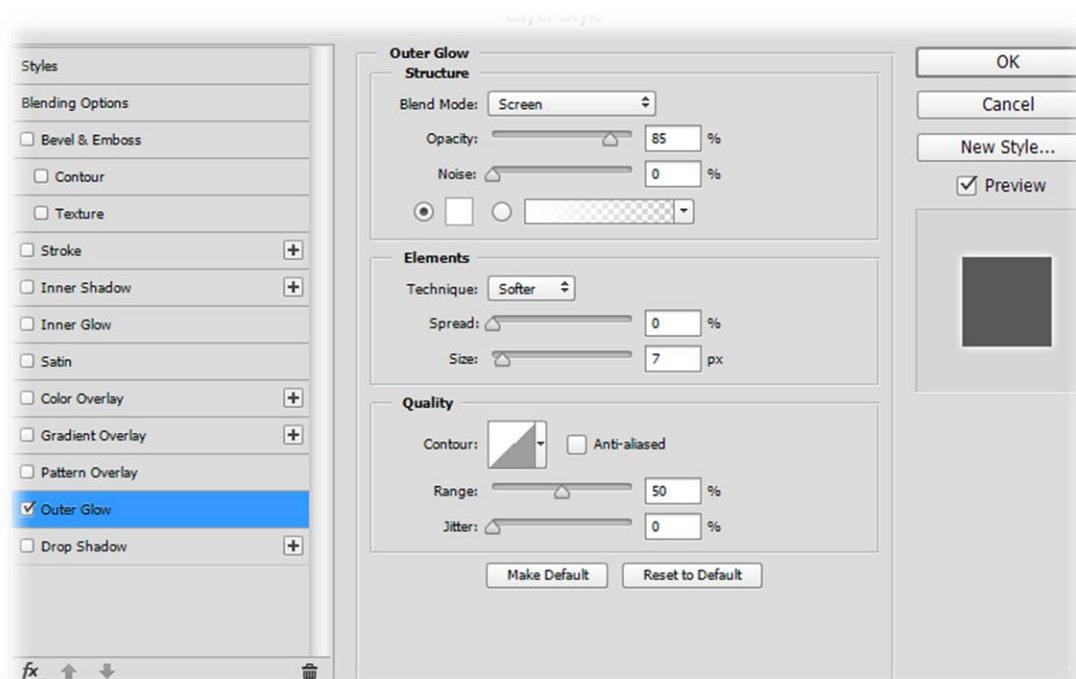
Αφού έγινε αυτό στην κάτω γωνιά θα πρέπει να αφαιρεθεί το background layer και να μείνει μονό το Layer της εικόνας το επόμενο βήμα θα είναι να πατήσουμε blending options στην εικόνα και να το αφαιρέσουμε πειράζοντας τις μπάρες στην έξυς εικόνα.



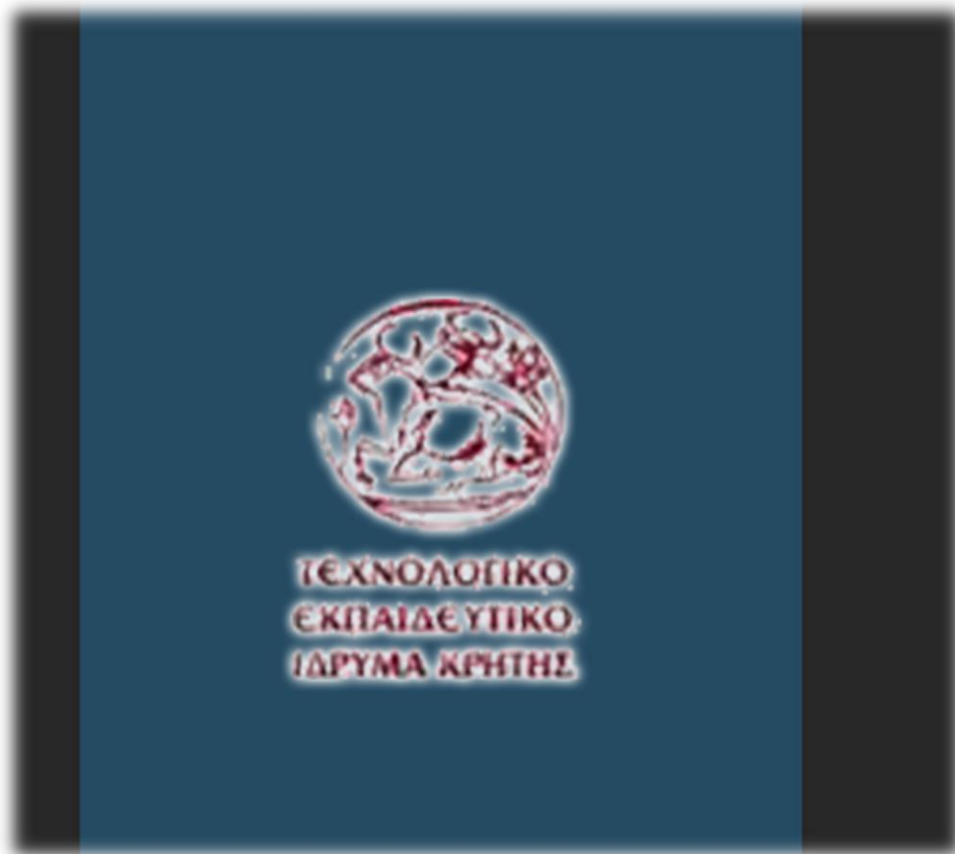
Θα μπορούσε να γίνει με το magic eraser tool αλλά το αποτέλεσμα δεν θα ήταν ίδια ποιοτικό, με την κατάλληλη επιλογή της μπάρας θα μείνει η εικόνα όπως θέλουμε.



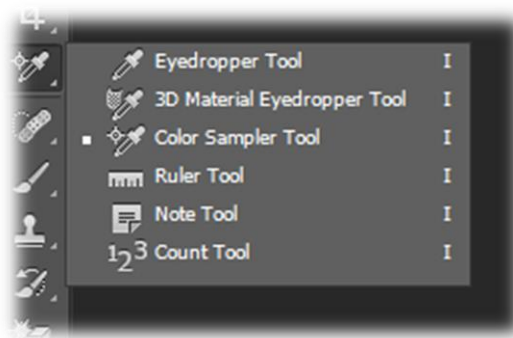
Η δουλειά όμως δεν τέλειωσε εδώ καθώς αυτό θα αφήσει μερικά άσπρα pixels στην εικόνα οπότε συνεχίζουμε προσθέτοντας ένα μπλε Layer η ένα layer με τετράγωνο το οποίο το κάνουμε μπλε αυτό για να εμφανίσει τα αποτελέσματα και να έχουμε καλύτερη διαχείριση. Με τις συνεχόμενες αλλαγές κατέληξα στις έξι παρεμβάσεις.

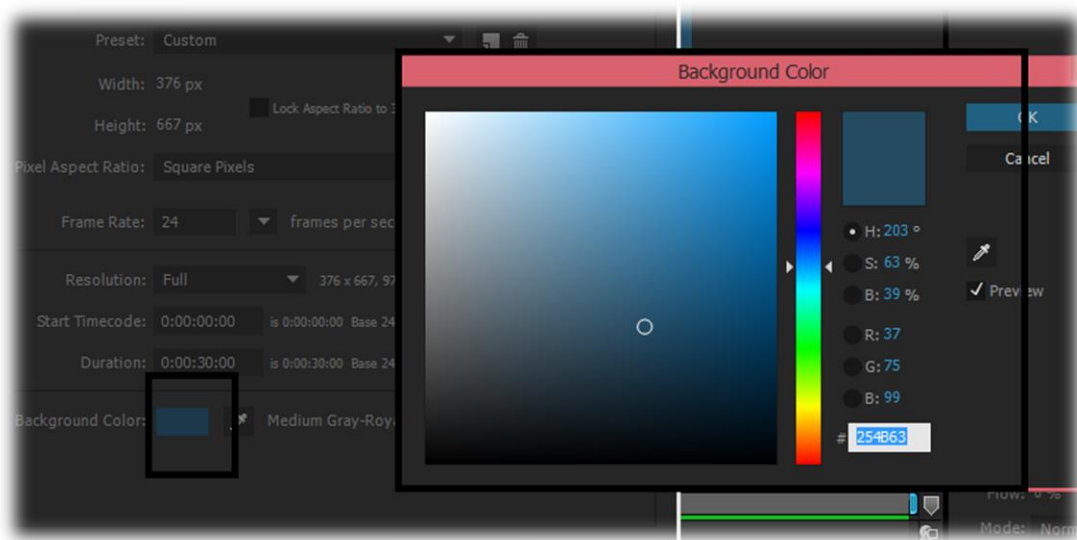


Με τελικό Αποτέλεσμα.



Αν θέλουμε μπορούμε να χρησιμοποιήσουμε το colour sampling ένα εργαλείο στο photoshop και να το εφαρμόσουμε την ώρα που βάζουμε στο composition του after effects.



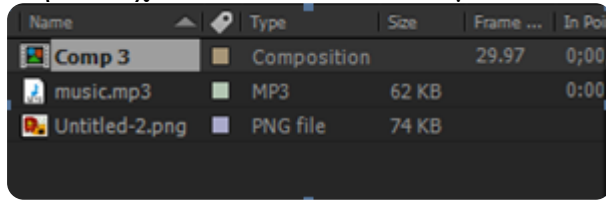


Στο τέλος διαγράφουμε το βοηθητικό Layer του Photoshop και αποθηκεύουμε σε εικόνα μορφή είδους png διότι με jpg θα εμφανιστεί άσπρο background αφού τελειώσουμε επιστρέφουμε στο adobe after effects και έχουμε τους αριθμούς του RGB στην άκρη σε περίπτωση που θέλουμε να αλλάξουμε το βασικό background στο After Effects.

After Effects ΣΥΝΕΧΕΙΑ:

Λόγο ανάγκη για καλύτερη διαχείριση και περισσότερη πληροφορία έβαλα το composition settings 29 fps.

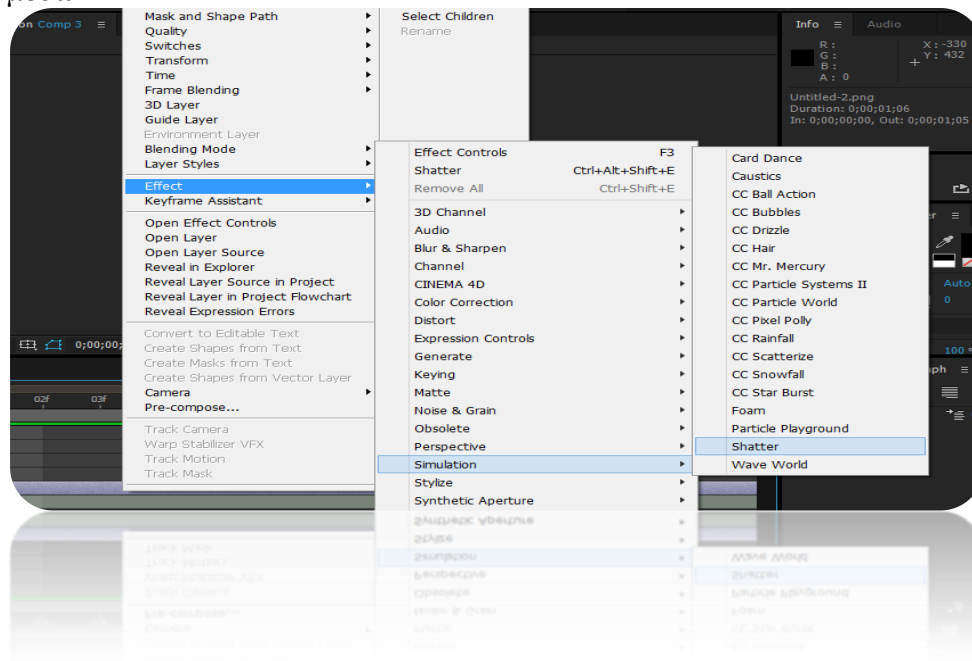
Στην συνέχεια τοποθετώ το αντικείμενο που είναι η εικόνα μου.



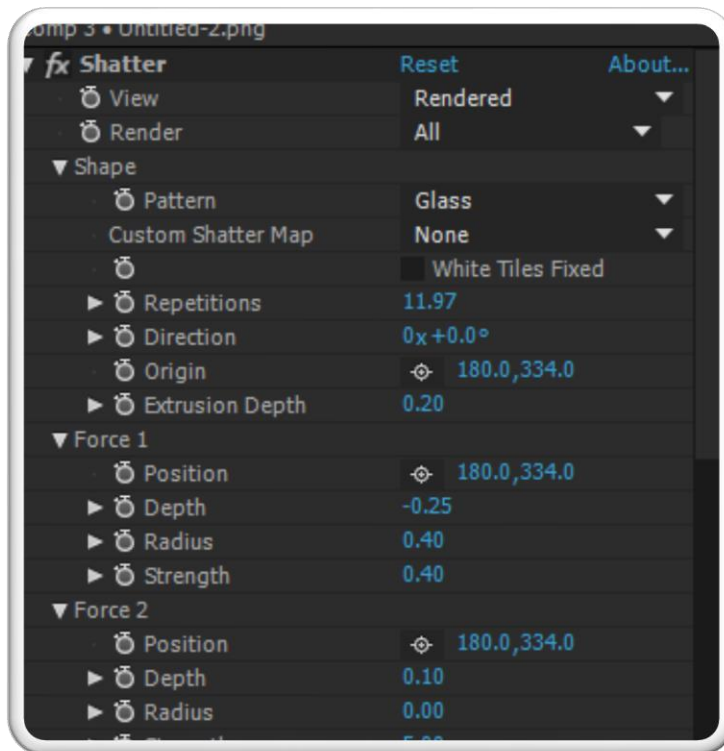
Και μετέπειτα στην γραμμή μετά αντικείμενα.



Το animation που θέλω να φτιάξω έχει σχέση με shatter και για να το επιλέξω πήγα μέσω



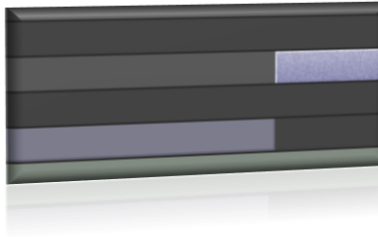
Τώρα για λογούς μουσικού συγχρονισμού με την κίνηση δημιούργησα δυο layer ώστε στο επόμενο να προσθέσω την κίνηση .Οι παρακάτω τιμές υποδηλώνουν δίκες μου προτιμήσεις. Όπως βλέπουμε έχουμε αρκετά στοιχεία ,με το view μπορούμε να δούμε την κίνηση πάνω στην εικόνα η σε σχεδία (πχ. Σχέδια κτηρίου) το Pattern δείχνει σε τι σχήμα θα σπάσει η εικόνα το Repetitions δείχνει τα κομμάτια στα όποια θέλουμε να σπάσει πολλά η λίγα το Origin είναι ένα σημαντικό στοιχείο που δείχνει το σημείο που θέλουμε να γίνει το κέντρο της “έκρηξης” με τις forces βλέπουμε βάθος, ακτίνα και strength επίσης το strength μεταβάλλει πόσο απότομα θα θρυμματιστεί η εικόνα. .



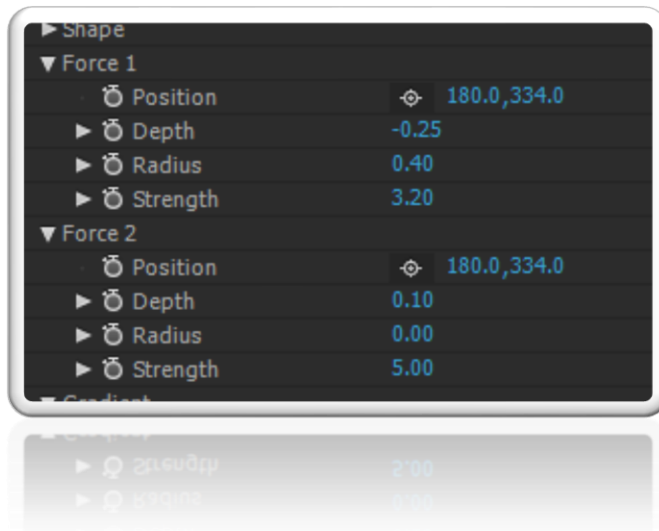
Επιπρόσθετα έχω βάλει και physics gravity 0 διότι δεν θέλω να πέφτει το σχήμα μου.



Αφού τα εφαρμόσω θα έχουμε την εξής μορφή κάτω.

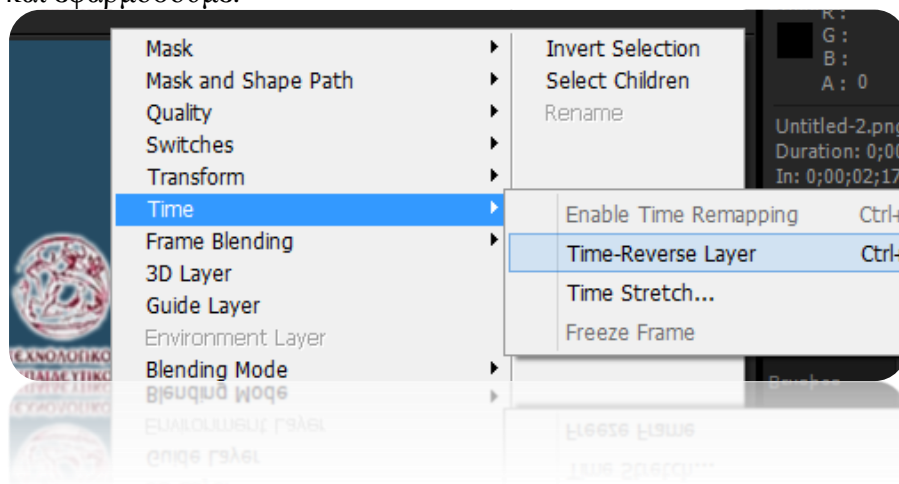


Δυο Layer ένα με τις βασικές ρυθμίσεις και ένα με δύναμη. Το επόμενο μου βήμα ήταν να αντιγράψω πάλι το 2 layer που βλέπουμε με την πάνω μπλε μπάρα και να δημιουργήσω ένα τρίτο αυτό το κάνω πάλι λόγω μουσικής υπόκρουσης διότι έχει δυο κύματα και θέλω να δώσω ένα απότομο σπάσιμο και οι ρυθμίσεις που έκανα είναι οι εξής:

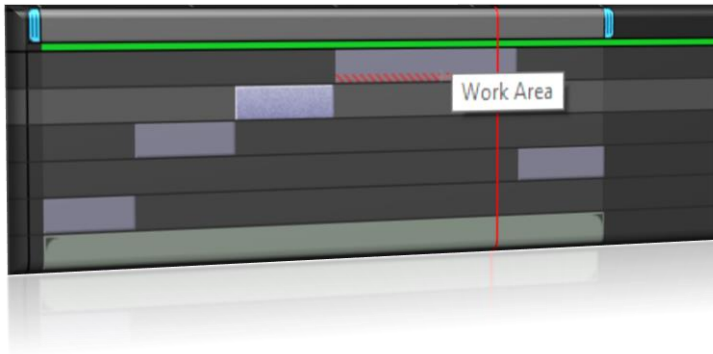


Παραμένουν ίδιες αλλάξα μόνο την δύναμη σε 3.2 .

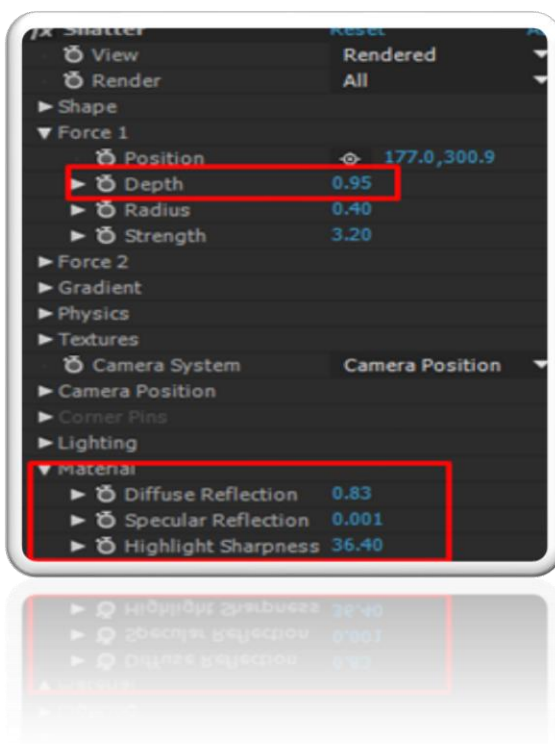
Αφού έκανα την κίνηση προς τα μπρος θα εφαρμόσω και την ανάποδη κίνηση ώστε μαζέψει τα κομμάτια πίσω αυτό γίνεται πολύ εύκολα αν αντιγράψουμε το τρίτο layer και εφαρμόσουμε.

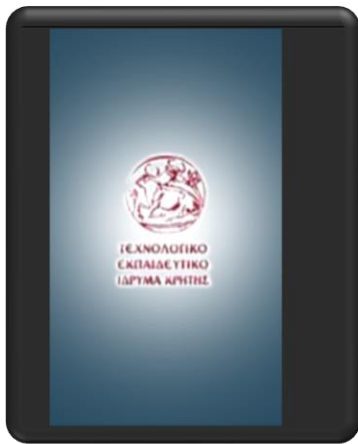


Θα το πάρουμε ανάποδα Η περιοχή εργασίας θα μοιάζει έτσι.

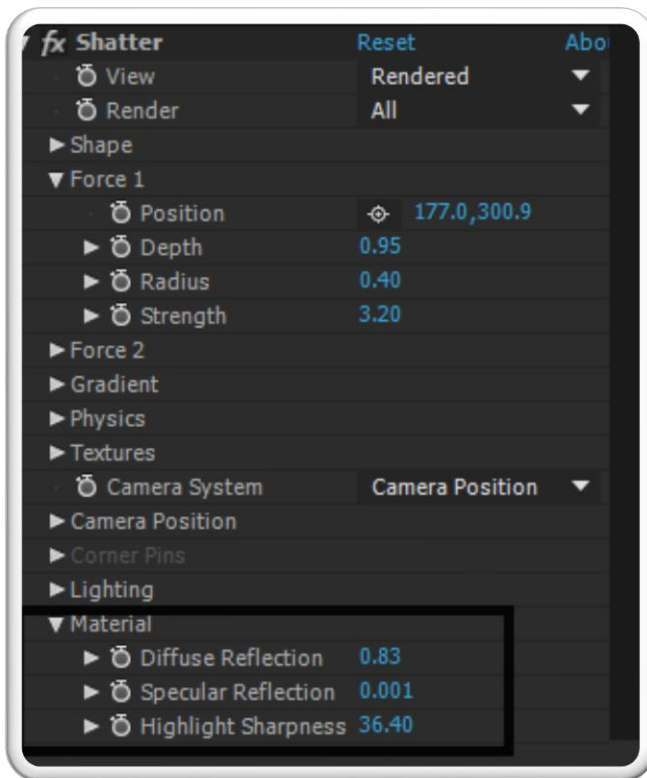


Αν εφαρμόσουμε ένα shattering effect στο layer το πρώτο και με αυτά τα στοιχεία θα έχουμε μια ωραία φωτεινή εικόνα .

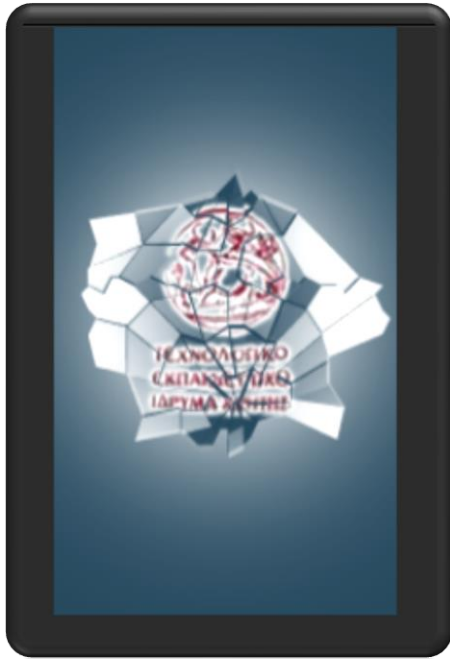




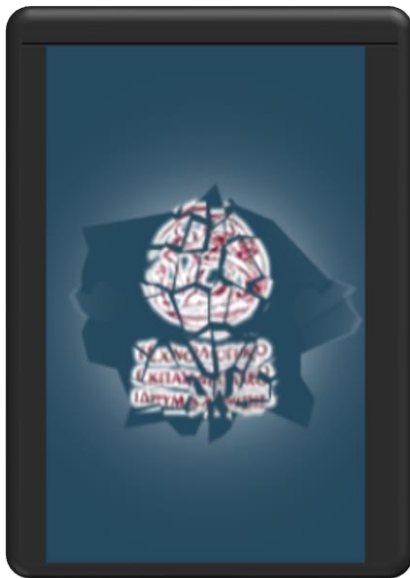
Και στην συνέχεια για τα επόμενα layer εφαρμόσα τις έξι ρυθμίσεις



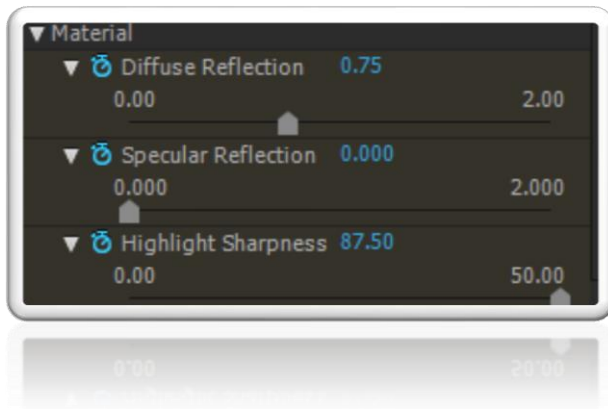
Και θα πάρουμε το έξις αποτέλεσμα οπού θα δώσει την παρακάτω εικόνα



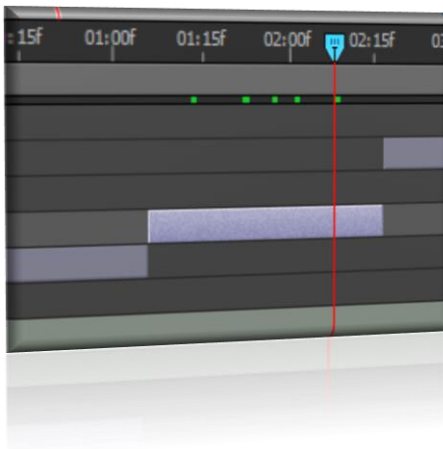
Το ζήτημα είναι ότι αν αφήσουμε έτσι τις ρυθμίσεις θα παραμείνει κάτι τέτοιο σε όλο το layer το δεύτερο συγκεκριμένα.



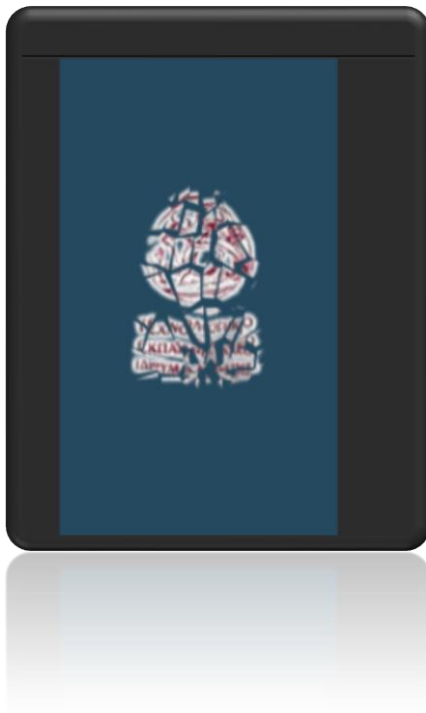
Κάτι που είναι ενοχλητικό στο μάτι αφού έσπασε το γυαλί πρέπει να φεύγει κάπως , έτσι λοιπόν αν πειράξουμε τις ρυθμίσεις ως εξής:



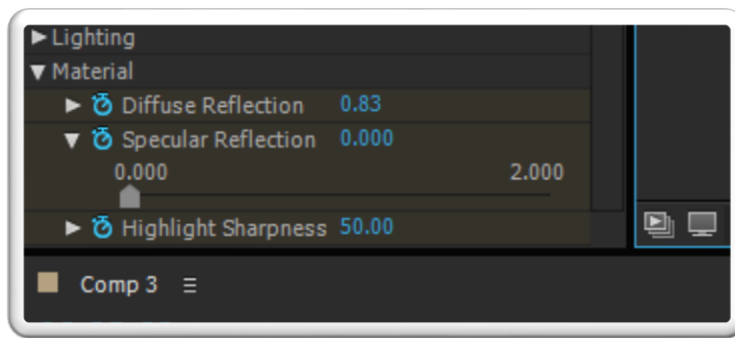
Εξαιρετο θα ήταν να λάβουμε υπόψη τα ρολόγια που είναι τονισμένα με μπλε τότε το layer αποκτά υποστάσεις και η συνέχεια σε κάθε υπόσταση θα δώσει διαφορετική εικόνα έτσι λοιπόν σε αυτό το σημείο:



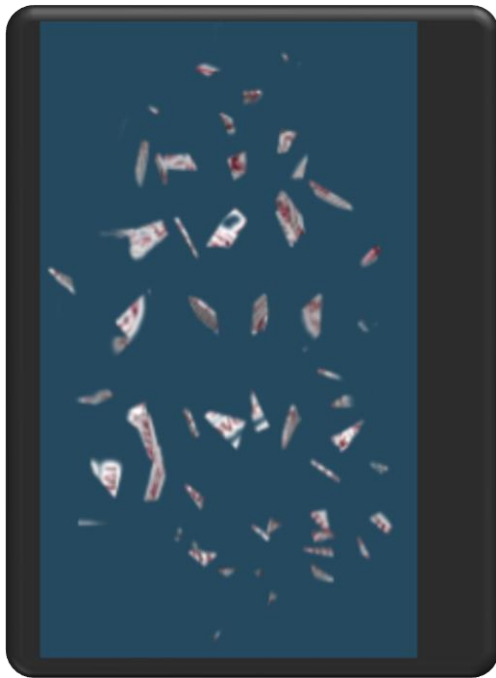
Θα πάρουμε κάτι τέτοιο.

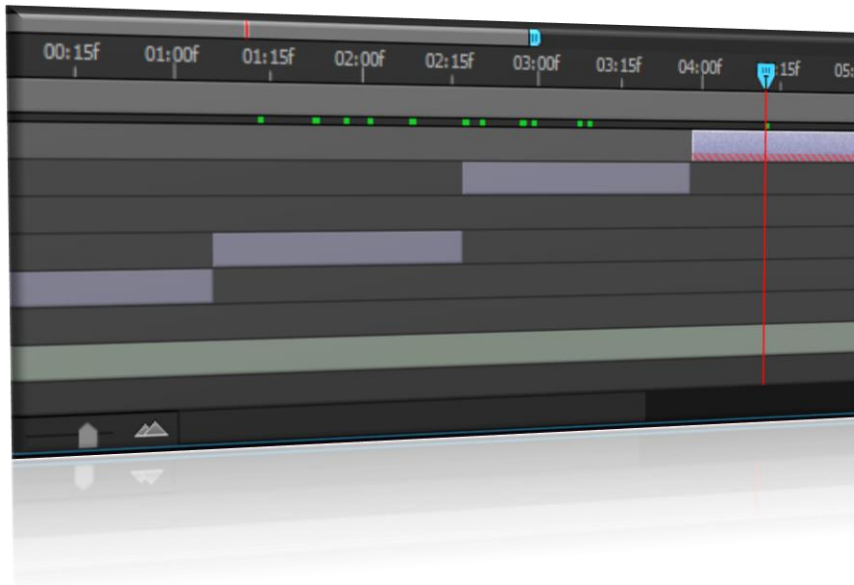


Στην συνέχεια στο τρίτο layer θα κάνουμε κάποιες ΕΠΙΠΡΟΣΘΕΤΕΣ ρυθμίσεις

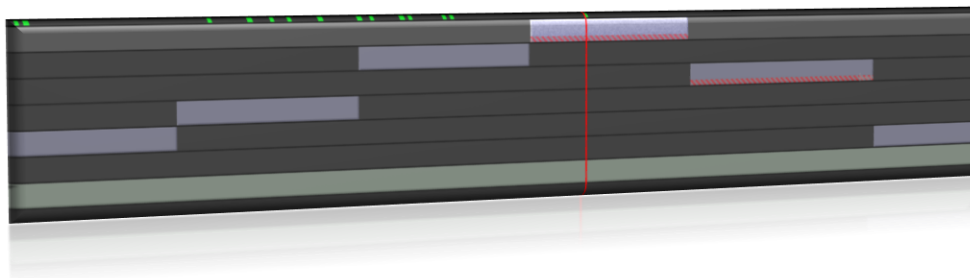


Σε αυτό το σημείο παρατηρούμε το εξής το specular είναι 0 κάτι το οποίο είναι το σημείο όπου δείχνει τις γραμμές της περιμέτρου του σπασίματος το highlight ο τονισμός της εικόνας είναι σε μια υψηλή τιμή άρα η εικόνα θα είναι έντονη και το diffuse τονίζει χρωματικά.





Στον 4 layer επέλεξα την αντίστροφη αλλά πάλι έκανα τις κατάλληλες ρυθμίσεις με τις οποίες ήθελα να εμφανίσω το animation το οποίο δεν θα περιγράψω περαιτέρω καθώς δεν έχει ιδιαίτερη σημασία γιατί ακλούθησα περίπου τα ίδια με πάνω άπλα διαφορετικές τιμές.



Στο 5 layer έχουμε πάλι έχουμε κάνει layer reverse από το τρίτο και το τελευταίο Layer είναι μια αντιγραφή του πρώτου, έτσι έχουμε τελειώσει το animation το οποίο σε μερικές εικόνες έχει ως εξής:

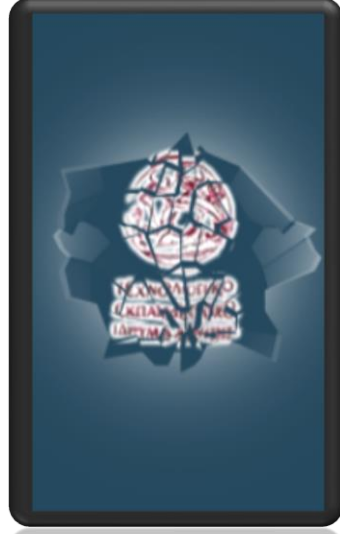
Layer 1



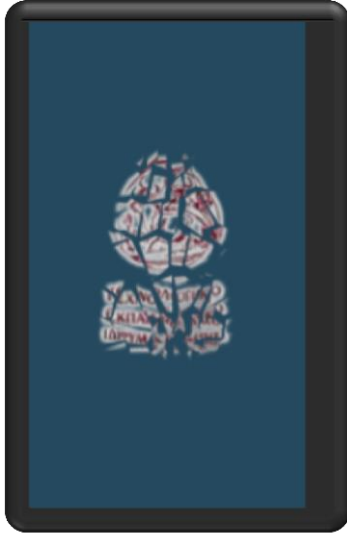
Layer 2



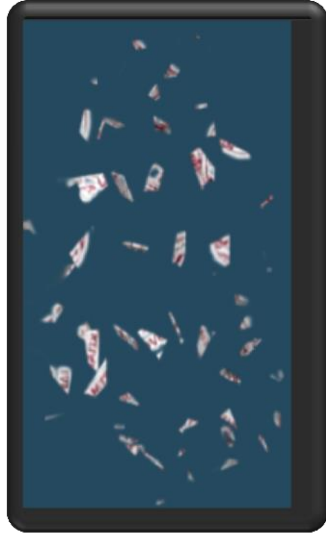
Layer 2



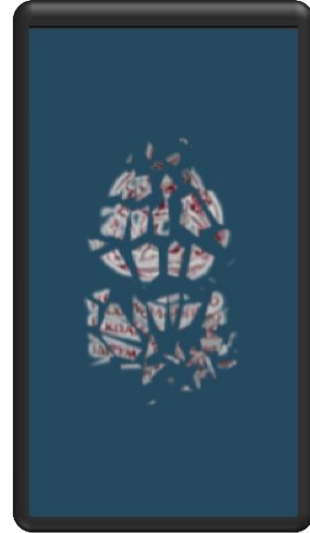
Layer 2



Layer 3



Layer 4



Layer 5



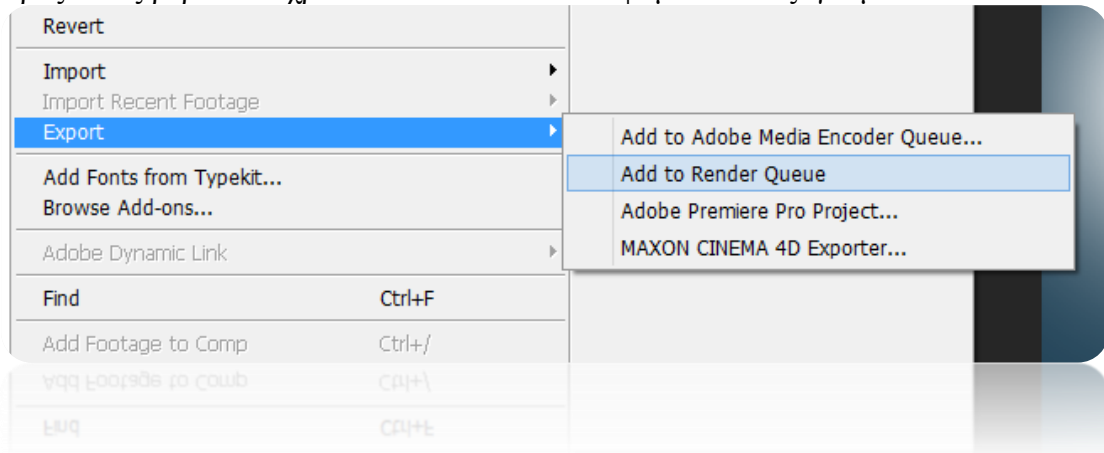
Layer 5



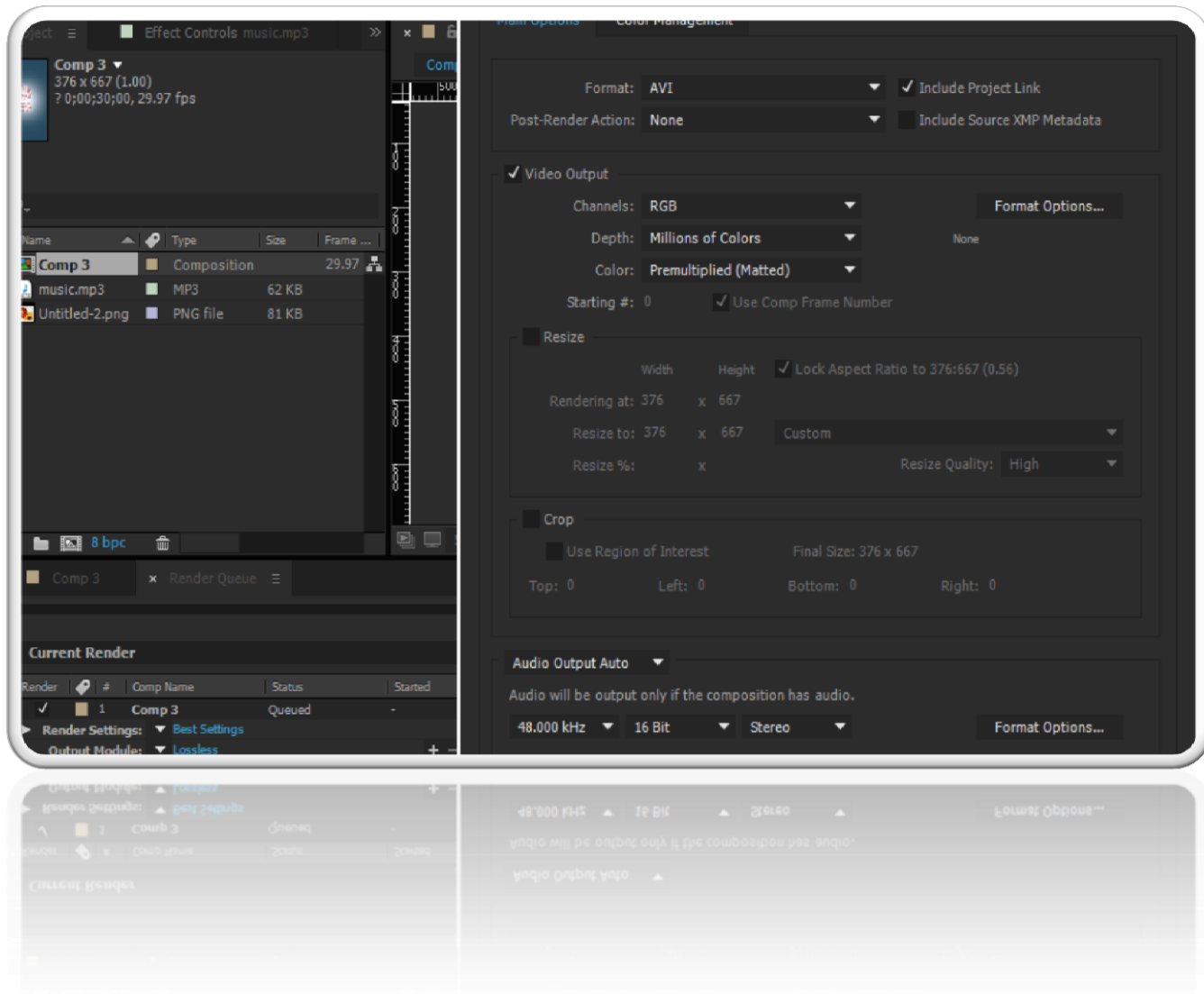
Layer 6



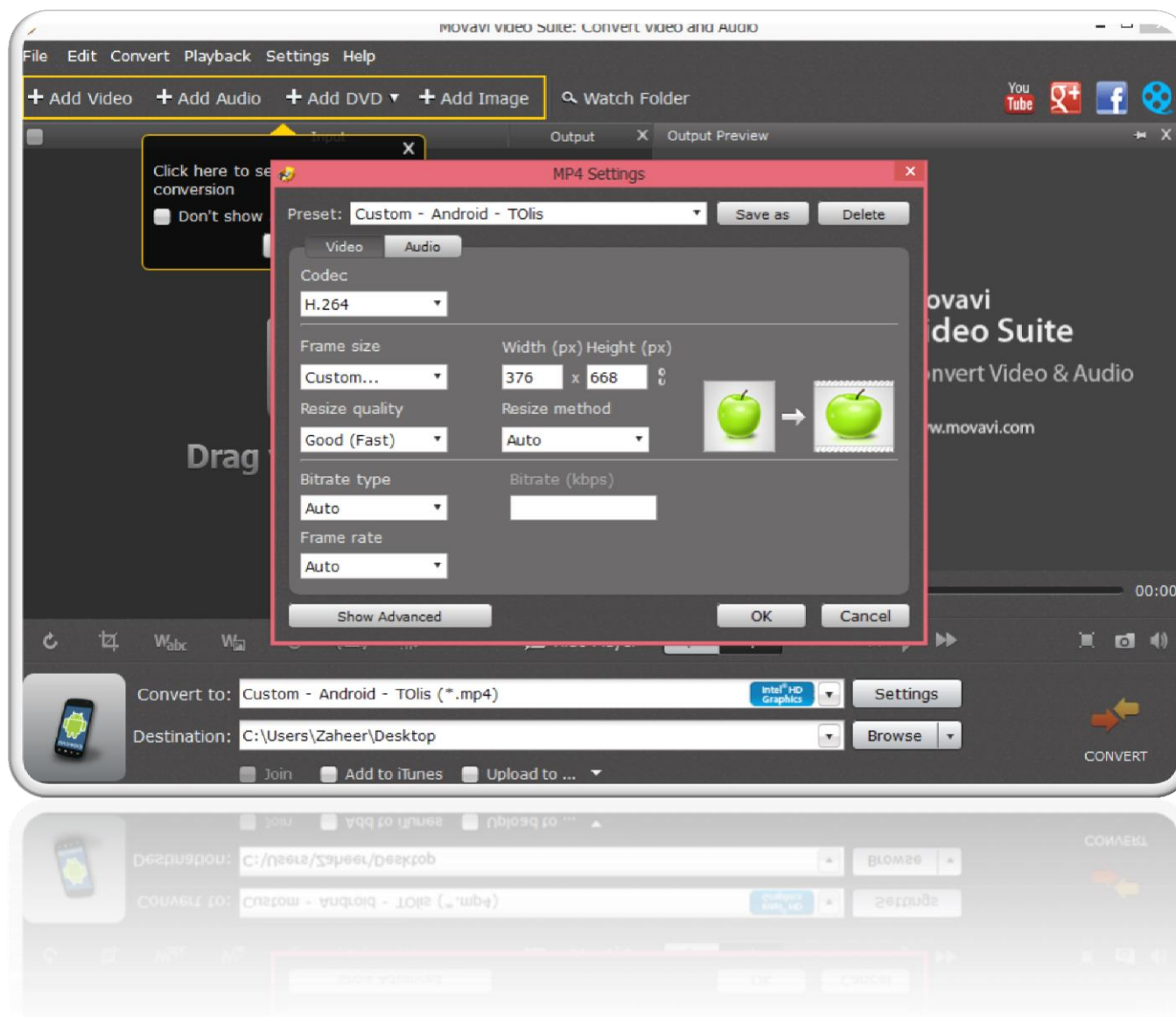
Προς τέλος βεβαία θα χρειαστεί αυτό που δουλέψαμε να το εξάγουμε έτσι λοιπόν:



Και στην συνέχεια να βγάλουμε μια εξαγόμενη μορφή εγώ επέλεξα την .mov με πολύ χαμηλά στατιστικά όπως 22khz audio ώστε να έχω ένα μικρό αρχείο



Αφού διαλέξω ανάλογα με την επιλογή μου θα πάω στο κουμπί κάτω γωνιά δεξιά και θα πατήσω render όταν τελειώσω την διαδικασία είμαι σχεδόν έτοιμος αλλά για να υπάρξει σωστό αποτέλεσμα και να παίζει στο android studio θα χρειαστεί να κάνουμε ένα convert διότι 1 ακόμα έχω μεγάλο αρχείο (60mb στην περίπτωση μου) και 2 δεν θα παίζει άρα λοιπόν για να το κάνω αυτό θα χρησιμοποιήσω ένα πρόγραμμα που λέγεται Μοναβί.



Το resize method θα μου απομακρύνει τοίχων μαύρα πλαίσια δίπλα στο video που δημιουργούνται από το convert άρα ή θα βάλω τα ακριβώς στοιχεία όπως φαίνονται (376 x 668) ή θα επιλέξω αντί auto το stretch, πατώ convert και έχω έτοιμο το video.

Δημιουργία online database.

Για να δημιουργήσουμε online database θα χρειαστούμε δυο παράγοντες

- 1) Domain name
- 2) Server Host όπου θα φιλοξενεί την βάση μας και την σελίδα μας και θα έχει τα εργαλεία για να έχουμε πρόσβαση σε αυτά

Την Βάση θα την αξιοποιήσουμε στην είσοδο έξοδο χρήστη από την εφαρμογή μέσα.

Τι είναι ένα domain name;

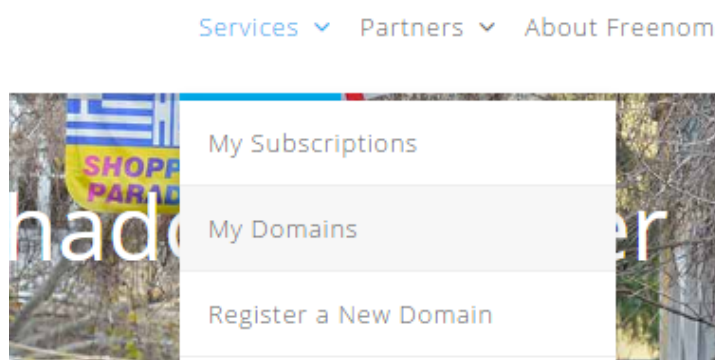
Επειδή οι IP διευθύνσεις είναι δύσκολο να αποτυπωθούν στο μυαλό μας υπήρχε η ανάγκη επίλυσης αυτού του προβλήματος. Φανταστείτε να θέλαμε να μπούμε σε μια συγκεκριμένη ιστοσελίδα και να πρέπει να θυμόμασταν μια IP διεύθυνση (π.χ.140.15.89.8) θα ήταν κάτι πολύ Κουραστικό καθότι υπάρχουν πολλές ιστοσελίδες στο διαδίκτυο όπου κάθε μια έχει την δική της IP διεύθυνση. Γι αυτούς τους λόγους αναπτύχθηκε το σύστημα ονοματοδοσίας περιοχών (Domain Name System - DNS) το οποίο αναλαμβάνει την διαδικασία αντιστοίχισης μιας IP διεύθυνσης σε ένα όνομα (domain) που είναι μοναδικό. (Παράθεση από πτυχιακή Αλέξανδρου Μπριτζολάκη link στις πληροφορίες)

Πώς όμως θα αποκτήσουμε ένα από πού θα το πάρουμε;;;

Αυτό μπορούμε να το δανειστούμε από πολλά site βέβαια δεν πρέπει να υπάρχει το όνομα. Για την υλοποίηση αυτής της πτυχιακής χρησιμοποίησα την υπηρεσία <https://my.freenom.com/clientarea.php>.

Και το ολοκλήρωσα ως εξής:

Όπως βλέπουμε πάμε στην παρακάτω εικόνα και επιλεγούμε το Register a New Domain.



Αφού πληκτρολογήσουμε ένα όνομα της δικής μας επιλογής ελέγχουμε άμα είναι διαθέσιμο και οπύς δείχνει η παρακάτω εικόνα ανάλογος με την επιλογή έχεις και τις υποχρεώσεις

TestTeiCrete Check Availability

Get one of these domains. They are free!

testteicrete .tk	FREE	EUR 0.00	Get it now!
testteicrete .ml	FREE	EUR 0.00	Get it now!
testteicrete .ga	FREE	EUR 0.00	Get it now!
testteicrete .cf	FREE	EUR 0.00	Get it now!
testteicrete .gq	FREE	EUR 0.00	Get it now!

Cost Price Domains cannot be found cheaper elsewhere!

testteicrete .com	COST PRICE	EUR 8.21	Select
testteicrete .net	COST PRICE	EUR 6.57	Select
testteicrete .org	COST PRICE	EUR 9.38	Select






Αφού το επιλέξουμε συνεχίζουμε στην αγορά του όπου θα δούμε για πόσο καιρό το θέλουμε και πάμε στο My Domains στην παρακάτω εικόνα βλέπουμε σχετικές πληροφορίες σχετικά με αυτό που επιλέξαμε.

Domain	Registration Date	Expiry date	Status	Type		
██████████	27/07/2016	27/07/2017	ACTIVE	Free	Manage Domain	Manage GoSite

Για να συνδέσουμε αυτό το όνομα σε ένα server θα πάμε στο Manage Domain , Management tools NameServers και θα κάνουμε add τους domain servers όπου θα μας δώσει ο Server Host που θα έχουμε κάνει και εκεί έναν λογαριασμό.

Εγώ επέλεξα τον <https://members.000webhost.com> για την πτυχιακή , μετά την εγγραφή θα αποσταλεί ένα μήνυμα με διαφορές πληροφορίες όπως Name Servers Username Pass για Ftp Server. Επίσης ένα σημείωμα θα ήταν όταν συνδέσουμε της υπηρεσίες μεταξύ τους παίρνει περίπου μια μέρα η λειτουργιά τους.

Έχουμε σχεδόν τελειώσει αυτό το κομμάτι το μονό που μένει πλέον είναι να δημιουργήσουμε την Βάση μας.

 Subdomains	 Parked Domains	 Add-on Domains	 Domain Tools	 Website Promotion Guide
---	---	---	--	---

» Advertisement

» Get a Premium Domain Name!







Get YourDomain.com for free!

Enter domain name, for example *yourdomain.com*




www. .com ▼

* Order our paid hosting service and we will register a personal domain name for you and give unlimited disk space / bandwidth for your website! Get learn more check our [upgrade page](#) or go to our premium hosting website www.Hosting24.com directly.

» Email

 Change Mailbox Password	 Manage Email Accounts	 WebMail	 Email Forwarders	 Modify MX Record	 More Email Services
--	--	---	---	--	--

» Database

 Change Database Password	 MySQL	 phpMyAdmin
--	--	---

Στην παραπάνω εικόνα αναθέτω την εικόνα από το menu της υπηρεσίας. Εδώ αν επιλέξουμε MySQL και φτιάξουμε την βάση θα βάλουμε ένα όνομα και pass που είναι σημαντικά όταν θέλουμε να κάνουμε εγγραφή η είσοδο αρά τα γραφούμε κάπου ασφαλές (για ιστορικούς λόγους επειδή έχω ασχοληθεί με το site και ευχαριστώ ιδιαιτέρως τους ανθρώπους πίσω από αυτό όποιος θέλει να φτιάξει βάση καλό θα ήταν να κάνει 3 φορές επανεγγραφή).

Αφού έχουμε ακολουθήσει σωστά τα παραπάνω βήματα θα δημιουργήσουμε τώρα ένα table. Στην παρακάτω εικόνα βλέπουμε πως δημιουργείται ένα table.

Server: localhost Database: a8920330_3856

Structure SQL Search Query Export Import Operations

Table	Action	Records	Type	Collation
users		7	MyISAM	latin1_swedish_ci
1 table(s)	Sum	7	MyISAM	latin1_general_ci

Check All / Uncheck All With selected: ▾

Print view Data Dictionary

Create new table on database a8920330_3856

Name: Number of fields:

Συμπληρώνοντας το όνομα και πόσα πεδία θέλουμε ερχόμαστε εδώ

Field	Type	Length/Values	Collation	Attributes	Null	Default	Extra
int	INT	2	greek_bin	UNSIGNED	not null		auto_increment
	TEXT				not null		
	VARCHAR				not null		

Table comments: Storage Engine: MyISAM Collation:

Save Or Add 1 field(s) Go

όπου επιλέγουμε τι μεταβλητές πρέπει να είναι κάθε πεδίο να είναι Null ή auto_increment επιπλέον μπορεί κάποιος να βάλει και κωδικοποίηση κειμένου κτλπ.

Για αυτή την πτυχιακή δημιουργήσα αυτά τα πεδία με αυτές τις ιδιότητες.

Field	Type	Collation	Attributes	Null	Default	Extra	Action
username	varchar(20)	latin1_swedish_ci		No			
password	varchar(50)	latin1_swedish_ci		No			
fname	varchar(20)	latin1_swedish_ci		No			
lname	varchar(30)	latin1_swedish_ci		No			
email	varchar(30)	latin1_swedish_ci		No			
address	varchar(30)	latin1_swedish_ci		No			
zipcode	varchar(30)	latin1_swedish_ci		No			
id	int(100)			No		auto_increment	

Check All / Uncheck All With selected:

Print view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After username Go

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	7		id
user	UNIQUE	7		username
email	UNIQUE	7		email

Create an index on 1 columns Go

Δημιουργία php αρχείων για την επικοινωνία με την online database.

Για να συνδεθούμε με την βάση και να έχουμε την ανάλογη ανταπόκριση θα χρειαστούμε ένα διαμεσολαβητή και στην περίπτωση μας θα είναι το php αρχείο.

Τι είναι το php;

PHP είναι ένα ευρέως χρησιμοποιούμενο ανοικτού κώδικα γενικού σκοπού scripting γλώσσα που είναι ιδιαίτερα κατάλληλη για την ανάπτυξη Web και μπορεί να ενσωματωθεί στην HTML. (Παράθεση από <http://php.net/manual/en/intro-what-is.php>)

ΚΩΔΙΚΑΣ:

ΕΙΣΟΔΟΣ ΧΡΗΣΤΗ (login):

Αρχείο dbConnect.php

```
<?php
define('HOSTA','mysql6.000webhost.com');
define('USERA','a8920330_3856');
define('PASSA', );
define('DBA','a8920330_3856');
$con = mysqli_connect(HOSTA,USERA,PASSA,DBA) or die('Unable to Connect');
```

Ο παραπάνω κώδικας περιγράφει την δήλωση μεταβλητών σύνδεσης και την εντολή με την οποία ο χρήστης συνδέεται στην βάση η αποτυγχάνει.

Αρχείο Login.php

```
<?php
$username = $_POST["username"];
$password = $_POST["password"];
require_once('dbConnect.php');
$stmt = mysqli_prepare($con, "SELECT * FROM users WHERE username = ?
AND password = ? ");
mysqli_stmt_bind_param($stmt, "ss", $username, $password);
mysqli_stmt_execute($stmt);
mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $username, $password, $fname, $lname,
$email,$address,$zipcode,$id);
$response = array();
$response["success"] = false;

while(mysqli_stmt_fetch($stmt)){
    $response["success"] = true;
    $response["username"] = $username;
    $response["password"] = $password;
    $response["fname"] = $fname;
    $response["lname"] = $lname;
    $response["email"] = $email;
    $response["address"] = $address;
    $response["zipcode"] = $zipcode;
    $response["id"] = $id;
}
echo json_encode($response);
?>
```

ΠΡΟΣΟΧΗ ΔΕΝ ΒΑΖΟΥΜΕ ΚΑΝΕΝΑ ECHO ΑΝΑΜΕΣΑ ΔΙΟΤΙ ΘΑ ΕΧΟΥΜΕ ERROR.

Όταν ο χρήστης καλέσει από το url του το αρχείο php με την μέθοδο Post Θα περάσει τις μεταβλητές Username Pass Που θα δώσει στις τοπικές μεταβλητές του αρχείου αυτού στην συνέχεια συνδεόμαστε με την βάση με την προσθήκη του αρχείου **dbConnect.php** μετέπειτα προετοιμάζουμε την δήλωση και δεσμεύουμε τις μεταβλητές τις οποίες θέλουμε να αναζητήσουμε στην βάση. Όταν εκτελέσουμε την αναζήτηση αποθηκεύουμε τα αποτελέσματα και τα δεσμεύουμε στις κατάλληλες μεταβλητές όνομα κτλπ. Μετά δημιουργούμε ένα πίνακα ο οποίος έχει ένα πεδίο την επιτυχία αυτό είναι χρήσιμο για να ελέγξουμε αν έχουμε επιτυχία η αποτυχία και επακολουθεί η while η οποία αποδίδει τις τιμές στα πεδία .Στο τέλος βλέπουμε ένα json encode αυτό θα χρειαστεί ιδιαίτερα για να πάρουμε τις τιμές κωδικοποιημένες ώστε από την μεριά του πελάτη να αποκωδικοποιηθούν ευκολότερα με την χρήση της Java όλη την πληροφορία.

ΕΓΓΡΑΦΗ ΧΡΗΣΤΗ (register):

Αρχείο register.php

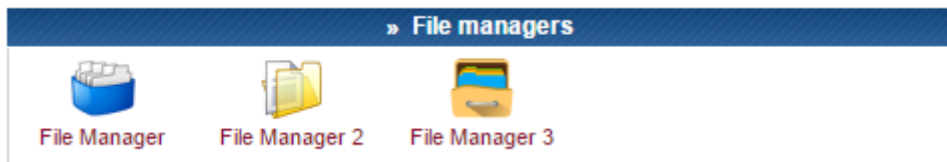
```
<?php
if($_SERVER['REQUEST_METHOD']=='POST'){
$usernameww = $_POST['username'];
$passwordww = $_POST['password'];
$fname = $_POST['fname'];
$lname = $_POST['lname'];
$email = $_POST['email'];
$address = $_POST['add'];
$zipcode = $_POST['zip'];
if($usernameww == "" || $passwordww == "" || $fname == "" || $lname == "" || $email == "" ||
$address == "" || $zipcode == ""){
echo 'please fill all values';
}else{
require_once('dbConnect.php');
$sql = "SELECT * FROM users WHERE username='$username' OR email='$email'";
$check = mysqli_fetch_array(mysqli_query($con,$sql));
if(isset($check)){
echo 'username or email already exist';
}else{
$sql = "INSERT INTO users
(username,password,fname,lname,email,address,zipcode) VALUES
('$usernameww','$passwordww','$fname','$lname','$email','$address','$zipcode)";
if(mysqli_query($con,$sql)){
echo 'successfully registered';
}else{
echo 'Email or username already exist! Please try again!';
}}
mysqli_close($con);
}}
else{
echo 'error';
}?>
```

ΠΡΟΣΟΧΗ ΔΕΝ ΒΑΖΟΥΜΕ ΚΑΝΕΝΑ ECHO ΑΝΑΜΕΣΑ ΔΙΟΤΙ ΘΑ ΕΧΟΥΜΕ ERROR.

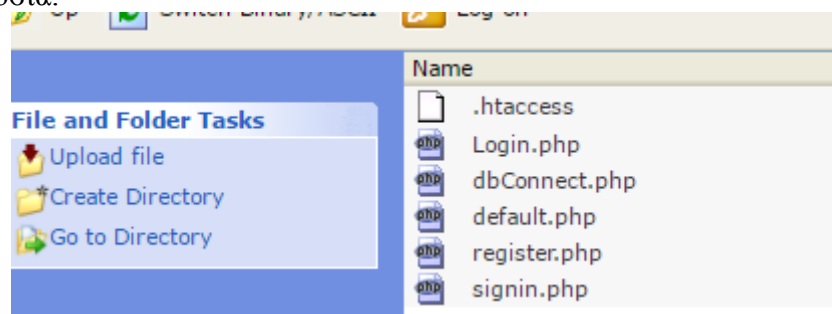
Και εδώ πέρα ακλουθούμε παρόμοια λογική παίρνουμε τις τιμές με την μέθοδο Post , συνδεόμαστε και βλέπουμε αν είδη υπάρχει το username η το email που είναι τα κύρια κλειδιά (όχι διπλότυπα) το id όπως είδαμε παραπάνω στην εικόνα είναι και αυτό primary αλλά auto incremental οπότε δεν μας ενδιαφέρει ο έλεγχος του. Οπότε αν δεν ισχύει το από πάνω συνεχίζουμε στην εισαγωγή τιμών στην βάση και στο κλείσιμο της σύνδεσης.

Το Ανέβασμα αρχείων θα γίνει ως εξής:

Όταν είμαστε στο menu της υπηρεσίας βλέπουμε κάποιον file manager όπου κάνουμε είσοδο με τα απαραίτητα στοιχεία.



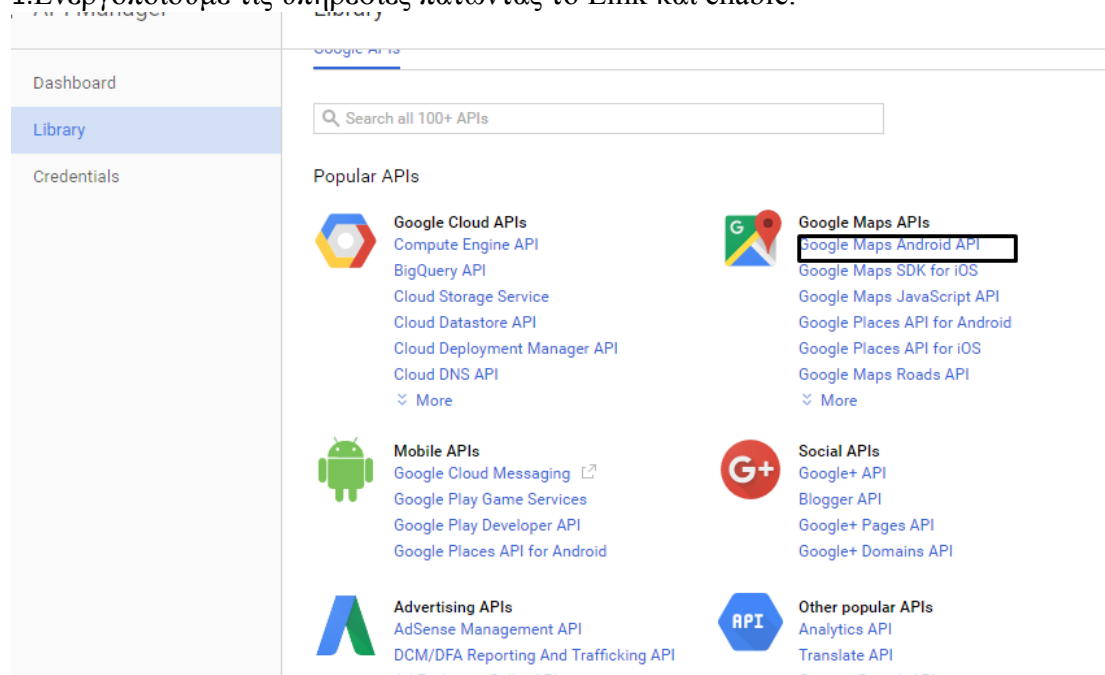
Και θα ανεβάσουμε όλα τα αρχεία μας στο public folder διότι μόνο εκεί μπορούν να είναι δημόσια.



Δημιουργία Google Api Key

Πολύ σημαντικό πράγμα πριν μπούμε στην κυρίως εφαρμογή είναι να ξέρουμε πως μπορούμε να στείλουμε τα αιτήματα στην google. Για να μπορέσουμε λοιπόν μετέπειτα να στέλνουμε αιτήματα και να παίρνουμε δεδομένα χρειάζεται να δημιουργήσουμε ένα Api Key το οποίο είναι καλό να διαφέρει ανάλογος τη συσκευή και τι θέλουμε να αιτηθούμε (πχ. Map Api η Direction Api). Αυτό το κάνει η google για λόγους ασφαλείας όπως συνεχόμενες επιθέσεις δηλαδή να ελέγχει το όγκο ζητούμενης πληροφορίας. Το Κλειδί λοιπόν το βρίσκουμε από αυτή την σελίδα <https://console.developers.google.com/apis/library> και για να το δημιουργήσουμε ακολουθούμε τα εξής βήματα:

1. Ενεργοποιούμε τις υπηρεσίες πατώντας το Link και enable.



API	Requests	Errors	Error ratio
Google Maps Directions API	67	0	0%
Google Places API Web Service	38	25	65.79%
Google Maps Android API	6	0	0%
Google Maps Geocoding API	–	–	–
Google Play Game Services	–	–	–

Η εικόνα αυτή μας δείχνει ότι χρησιμοποιούμε αυτές τις τρεις υπηρεσίες πράγμα που θα προμηθύνει και 3 Api Key.

2. Στα credentials παίρνουμε το κλειδί το οποίο μετά μπορούμε να του βάλουμε Restriction ώστε να είμαστε σίγουρη ότι εμείς χρησιμοποιούμε μόνο αυτό κλειδί διότι Επέχει περιορισμούς στα καθημερινά αιτήματα (παρακάτω αφού δημιουργήσουμε την εφαρμογή θα ειπωθεί).

Σχήμα εφαρμογής τι κάνει η κάθε κλάση και οι κυρίως μεταβλητές.

Η τακτική με την οποία θέλω να παρουσιάσω όπως καταλάβατε μέχρι αυτό σημείο είναι να δώσω ένα γενικό πλάνο και να προετοιμάσω τον μελλοντικό αναγνώστη τι θα δει, δεν θα τον βέβαια πιστεύω κατά την άποψη μου προχωρώντας να δείχνω πράγματα διότι περισσότερο σύγχυση παρά βοήθεια θα μπορούσα να προκαλέσω ελπίζω να φαίνεται σωστά η παρουσίαση μου και όσο δυνατό λιγότερο κουραστική, ευχαριστώ.

Παρακάτω όπως αναφέρετε ο τίτλος θα επακολουθεί μια σύντομη περιγραφή του κώδικα και των κύριων μεταβλητών που λειτουργεί σαν βοήθημα στην ανάγνωση και στην εμφάνιση ενός γενικού πλάνου.

Επεξήγηση:

CheckSeeMeetings

Σε αυτή την κλάση θα αναφερθώ στην παρουσίαση των συναντήσεων σε μορφή button οι οποίες πληροφορίες εισάγονται μέσα από την τοπική βάση που θα είναι σε χρονολογική σειρά. Επιπλέον θα έχω δύο listeners οι οποίοι θα είναι short and long press αναλόγως το πάτημα γίνεται επαναφορά δρομολόγησης στο χάρτη παίρνοντας πληροφορίες μέσα από την βάση και επιστρέφοντας στην κλάση PlaceAndIntrests δηλώνοντας την οποία έρχομαι ή θέτω ειδοποίηση η οποία ενημερώνει 20 λεπτά πριν μέχρι 24 μέρες η ενεργοποίηση του.

Intro

Σε αυτή την κλάση θα αναφερθώ στην παρουσίαση του βίντεο που έχω δημιουργήσει αυτό φορτώνεται μέσα από τις πηγές αρχείων σε περίπτωση που κάποιος βιάζεται υπάρχει ένα transparent κουμπί ορισμένο σε όλη την οθόνη που πατάς και συνεχίζει γρήγορα.

MyAlarmService

Σε αυτή την κλάση παρουσιάζεται ο τρόπος με τον οποίο θέτετε ένας alarm reminder και με τις ιδιότητες του πχ αν θα βγάλει εικονίδιο μουσική text στις υπενθυμίσεις ακόμα την συμπεριφορά του πχ το πάτημα δημιουργεί μια συντόμευση απευθείας στην εφαρμογή σε κάποια δραστηριότητα

PlaceAndIntrests

Αυτή η κλάση που περιγραφώ είναι και η κυρίως κλάση όλα στηρίζονται εδώ όποτε θέλει ιδιαίτερη προσοχή. Το πρώτο πράγμα που κάνει λοιπόν είναι να κάνει εισαγωγή listeners χάρτη, σύνδεσης με το google και τοποθεσίας είναι όλοι εξίσου σημαντικοί γιατί με τον listener τοποθεσίας τρέχει μέθοδος που ενημερώνει την νέα θέση όταν κάνω refresh ή γίνει αυτόματα μας παρουσιάζεται με την σύνδεση αν κάνω pause resume τι θα τρέξει τι πρέπει να κάνω και με τον χάρτη αν κάνω κλικ σε ένα αντικείμενο. Στην συνέχεια δημιουργώ αντικείμενα για τους navigation drawers δεξιά και αριστερά που έχουν και αυτοί μια λειτουργικότητα και ένα κώδικα που πρέπει να τρέξει. Από μεθόδους υπάρχουν αρκετές όπως το να ελέγγω κάθε κουμπί αν πρέπει να πατηθεί τότε μπορείς που θες να πας και μια μεγάλη λειτουργικότητα από πίσω

με try catch καθώς είναι ιδιαίτερα σημαντικά. Θα δούμε την μέθοδο που ελέγχει πάνω τον spinner .Επίσης έχει την SetMeeting σε αυτή ελέγγω αν έχω ειδή έρθει από την κλάση CheckSeeMeetings αν μπορώ να στήσω πάλι την διαδρομή. Επιπρόσθετα έχω μια μέθοδο που ελέγχει το Automatic Update το Manual όπου το καθένα έχει την συνεχεία του (πχ στο Manual αν ο χρήστης κουνήσει με ιδιαίτερο τρόπο το κινητό θα κάνει update).Και τέλος έχουμε συναρτήσεις σύνδεσης όπως ανάφερα παραπάνω και μια κυρία την handleNewLocation στην αλλαγή θέσης και όταν απαιτητέ τρέχει αυτή.

```
{
```

```
RouteOnly
```

Είναι η κλάση και η πρώτη πληροφορία που βρήκα για να ξεκινήσω το project και την έχω βάλει σαν activity για λόγους ιστορικούς επίσης ένα ακόμα ευχαριστώ στον [Hier Mai Thanh](https://www.youtube.com/watch?v=CCZPUeY94MU) και η εργασία του βρίσκεται εδώ

<https://www.youtube.com/watch?v=CCZPUeY94MU>

```
DirectionFinder
```

Εδώ πέρα δημιουργείτε μια σύνδεση όπου αποστέλλουμε το αίτημα μας αν επιστρέφει το αποκωδικοποιούμε και το περνάμε για να το τοποθετήσουμε στις διαδρομές

```
Polylines
```

```
DirectionFinderListener
```

```
Distance
```

```
Duration
```

```
Route
```

Ο κύριος αυτός έκανε με την εργασία του αναφορά στοDirection Api και την Δρομολόγηση

```
}
```

```
SetMeeting
```

Αυτή η κλάση είναι η συνέχεια της PlaceAndIntrests δηλαδή όταν επιλέξω σημείο ενδιαφέροντος έρχομαι εδώ που στην συνέχεια αντλώ πληροφορίες μέσα από τους πίνακες με το id που δημιουργώ το google query και παίρνω τα δεδομένα μέσω της GetYourPlace τα παρουσιάζω ελέγγω αν δεν υπάρχουν να δώσω κάποια δικιά μου διότι σε κάποια καταστήματα δεν υπάρχει καμιά φορά το ωράριο. Μετά έχουμε την επιλογή ημερομηνίας ώρας τον χρόνο παραμονής και αφού το επιτρέπει η βάση και πάρω τα ok από τα validations τότε προσχωρώ στην CheckSeeMeetings όπου καταχωρώ την συνάντηση στην βάση. Επιπρόσθετα και τελευταίο εδώ πέρα γίνεται και εισαγωγή τιμών ωραρίου στους πίνακες.

```
Signin
```

Η κλάση Signin Παρουσιάζει την φόρμα υποδοχής και εισόδου για τον χρήστη , έχει τρία κουμπιά ένα γρήγορη σύνδεση που το έχω για δοκιμές ένα εγγραφή και ένα είσοδο. Σε περίπτωση που ο χρήστης πατήσει την δοκιμαστική είσοδο τότε θα έχει και ψεύτικες πληροφορίες προφίλ στην άλλη περίπτωση δημιουργεί ένα αίτημα ως προ τον server όπου λαμβάνω πίσω τα δεδομένα και τα παίρνω στην συνέχεια.

```
Signup
```

Όπως και η κλάση Signin και η Signup έχει παρόμοια λογική όπου εδώ βέβαια

γίνεται η εγγραφή.

WelcomeUser

Η κλάση WelcomeUser είναι μια εισαγωγική κλάση όπου ο χρήστης βλέπει το προφίλ του και του δίνονται επιλογές που θα κινηθεί μέσα στην εφαρμογή.

Database

Στην Database κλάση παρουσιάζεται η τοπική βάση με το table columns και τα πεδία επίσης θα δούμε μεθόδους καταχώρησης συν μεθόδους αναζητήσεων σύμφωνα με τις ανάγκες επίσης επέλεξα να τοποθετώ τις πληροφορίες που επιστρέφονται σε Json Object-Array για λόγους ευκολίας.

DatabaseInterface

Είναι μια βοηθητική κλάση στην Database όπου κάνει επιπλέον τον έλεγχο επαλήθευσης των ραντεβού αναλόγως του προγράμματος καταστήματος και υποχρεώσεων.

FindPlaces

Σε αυτήν την κλάση θα δούμε την δημιουργία του query και με ποιους μεθόδους το υλοποίησα για να πάρω όλη την πληροφορία για να επιστρέψω στον χάρτη τα δεδομένα (σημεία ενδιαφέροντος).

VariablesFindPlaces

Κλάση που περιέχει μεταβλητές.

GetYourPlace

Εδώ δημιουργώ ένα query και παίρνω την ονομασία της τοποθεσίας που βρίσκομαι όταν έχω ανοικτή την εφαρμογή.

MarkerListener

Είναι η κλάση listener όσο αφορά την επιλογή σημείου από τον χάρτη, άμα ο χρήστης πατήσει το σινεμά έχει μια διαφορετική αντίδραση που θα τον στείλει για αναζήτηση στον google.

NavigationLeft

Είναι η κλάση που αφορά το Left Navigation Drawer συγκεκριμένα θα δούμε το αλγόριθμο τον οποίο εκτελείται για να μπουν σε ταξινόμηση τα στοιχεία και τον Listener Που βλέπει αν πατηθεί οπουδήποτε κουμπί μέσα από το μενού για να έχει την κατάλληλη αντίδραση.

LoginRequest

Η κλάση παρουσιάζει τον τρόπο που γίνεται το LoginRequest και τι response παίρνω πίσω.

RegisterUserClass

Η κλάση παρουσιάζει τον τρόπο που γίνεται το RegisterRequest και τι response παίρνω πίσω.

DownloadImageTask

Είναι μια μικρή κλάση η οποία τρέχει ασύγχρονα για να κατεβάσει το image του

καταστήματος όταν ζητηθεί από το Right Navigation Drawer.

NaviRightLoader

Είναι το Right Navigation Drawer στο οποίο όταν επιλέξω ένα σημείο και πατήσω Load θα δώσει πληροφορίες ωραρίου καταστήματος και την φωτογραφία του σύμφωνα με την google.

GetStringShopSuggested

Στην κλάση αυτή παραθέτω δικό μου πρόγραμμα καταστήματος άμα η google δεν έχει καμιά πληροφορία.

GetPlaceSchedule

Σε αυτήν την κλάση θα δούμε την δημιουργία του query και με ποιους μεθόδους το υλοποίησα για να πάρω όλη την πληροφορία για να επιστρέψω την πληροφορία του προγράμματος σύμφωνα με το Google Id του.

SetMeetingsVariables

Κλάση που περιέχει μεταβλητές.

SetRoute

Σε αυτή την κλάση δημιουργούνται οι βάσεις για να δημιουργηθεί το Direction query να σχεδιαστεί η δρομολόγηση πάνω στον χάρτη και να επιστρέψει τις σημαντικές τιμές χρόνου και απόστασης αναλόγως από πού καλώ την κλάση και φτιάχνω το αντικείμενο έχω και διαφορετικό αποτέλεσμα δηλαδή για να τρέξω τον αλγόριθμο κοντινών σημείων έπρεπε να χρησιμοποιήσω τα βασικά στοιχεία της Set Route κλάσης και να μην ξεχάσω να δώσω το ok στις πληροφορίες που χρειάζεται για να τρέξει ο αλγόριθμος.

SetRouteVariables

Κλάση που περιέχει μεταβλητές.

Static Variables

public static ArrayList<Integer> Duration

Λίστα με τις διάρκειες αποστάσεων των σημείων.

public static ArrayList<String> Places_Vicinity = new ArrayList<String>();

Λίστα με την ονομασία περιοχής του κάθε σημείου.

public static ArrayList<String> Places_photo_reference = new ArrayList<String>();

Λίστα με το κωδικό όνομα της φωτογραφίας του κάθε σημείου που έχει δοθεί από την google.

public static ArrayList<String> Places_LongLatitubee

Λίστα με τις θέσεις των σημείων.

public static double currentLatitude;

Τωρινή θέση στην οποία βρίσκομαι Latitude.

public static double currentLongitude;

Τωρινή θέση στην οποία βρίσκομαι Longitude.

public static int existisnextpage;

Όσο αφορά αυτή εδώ την μεταβλητή η google όταν στέλνει δεδομένα τα στέλνει σε σελίδα κωδικοποιημένα σε μορφή json όπου δεν είναι πλήρες δηλαδή αντί να στέλνει με τη μια 60 σημεία στέλνει 20/20/20 για να ελέγξω λοιπόν εγώ τοίχων προβλήτα όπως αν πείρα όλα τα δεδομένα έπρεπε να δημιουργήσω μια τέτοια μεταβλητή.

public static int if_fine_then_search

Χρήσιμη μεταβλητή του αλγορίθμου που εκτελείτε στο Navigation Drawer και επιτρέπει αν μπορείς να ψάξεις αφού βέβαια έχεις έτοιμα τα δεδομένα.

public static GoogleMap mMap;

Ο χάρτης μας.

public static ArrayList<String> PlacesId = new ArrayList<String>();

Λίστα με τα id των σημείων εξίσου σημαντικός διότι θα θέλουμε να κάνουμε queries με πληροφορίες καταστημάτων τότε πρέπει να έχουμε το Id για να επικοινωνήσουμε με την google και να στείλει τα δεδομένα.

public static ArrayList<String> PlacesName = new ArrayList<String>();

Λίστα με τα ονόματα των σημείων.

public static ArrayList<Integer> typeofmark = new ArrayList<Integer>();

Πίνακας που δηλώνει των τύπων των σημείων τα έχω κωδικοποίηση με 1 2 3 για λόγους ευκολίας.

public static int Navigation_ok = 0;

Είναι μια μεταβλητή που δεν επιτρέπει να πατήσει κάτι από το menu στο Left Navigation Drawer ο χρήστης για μετάβαση αν δεν έχει εκτελεστεί πλήρως ο αλγόριθμος.

Πίνακες με τις ώρες λειτουργίας

```
{  
public static String Monday[], Tuesday[], Wednesday[], Thursday[], Friday[],  
Saturday[], Sunday[];  
}
```

public static int checkernull;

Μεταβλητή που ελέγχει αν υπάρχουν δεδομένα

public static JSONObject types;

Json Object που έχει πληροφορία από το ωράριο του ζητούμενου καταστήματος.

public static int flagondemand;

Μεταβλητή η οποία δείχνει την κατάσταση που είμαι σε αυτόματη ενημέρωση η όχι.

public static int setRoute_clear_button_state;

Δείχνει την κατάσταση που βρίσκεται το κουμπί SetRoute η Clear διότι πρέπει να ξέρουμε αν έχω δρομολόγηση σχεδιασμένη η όχι.

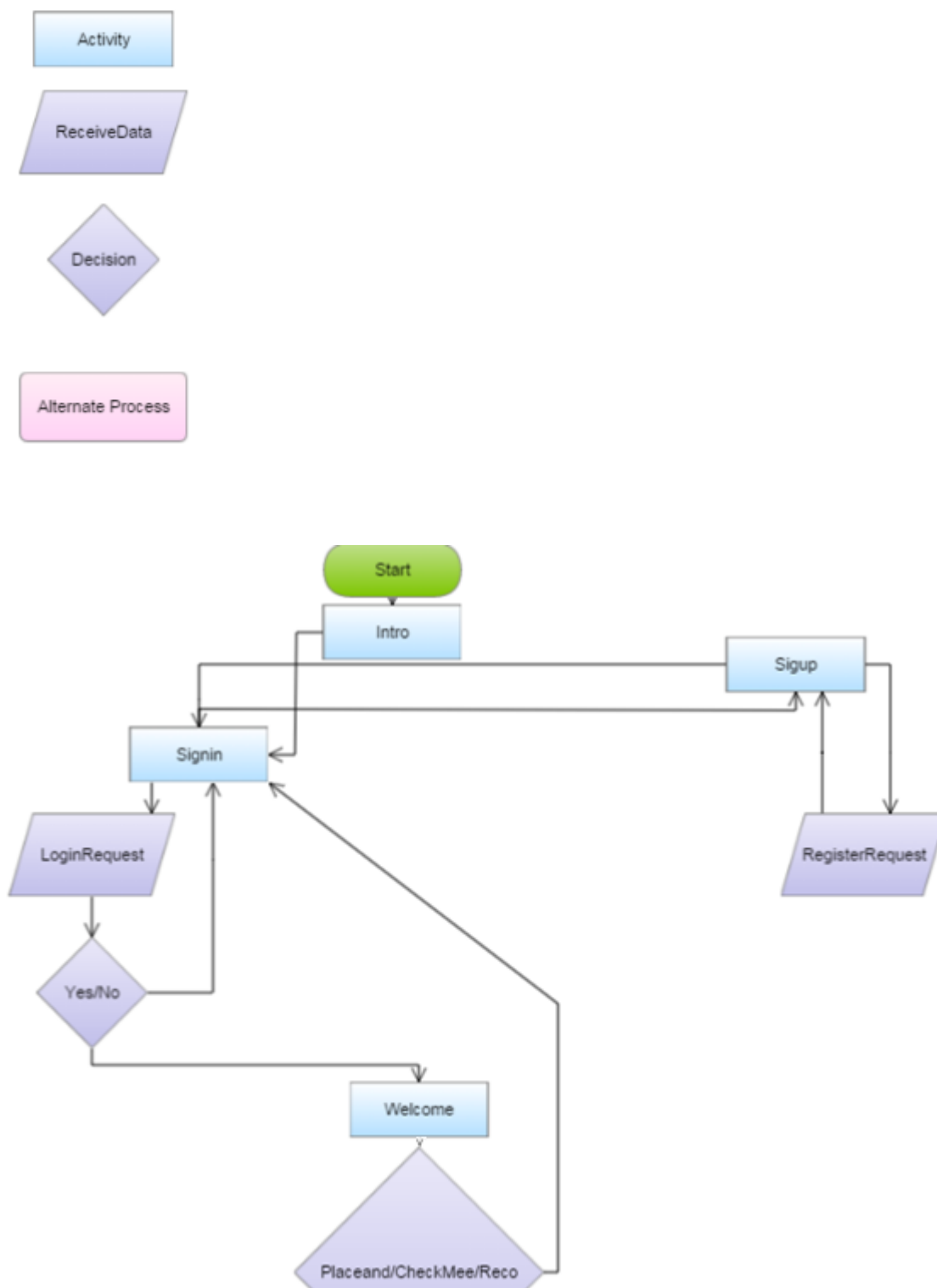
```
{  
Μεταβλητές για την σχεδίαση της δρομολόγησης στο χάρτη  
public static List<Marker> originMarkers = new ArrayList<>();  
public static List<Marker> destinationMarkers = new ArrayList<>();  
public static List<Polyline> polylinePaths = new ArrayList<>();  
}
```

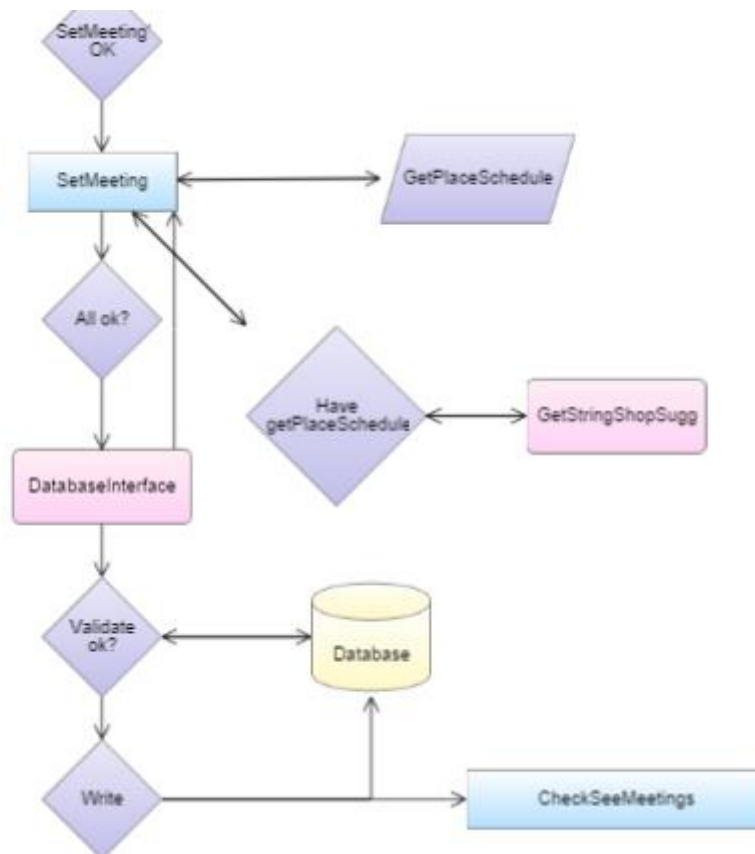
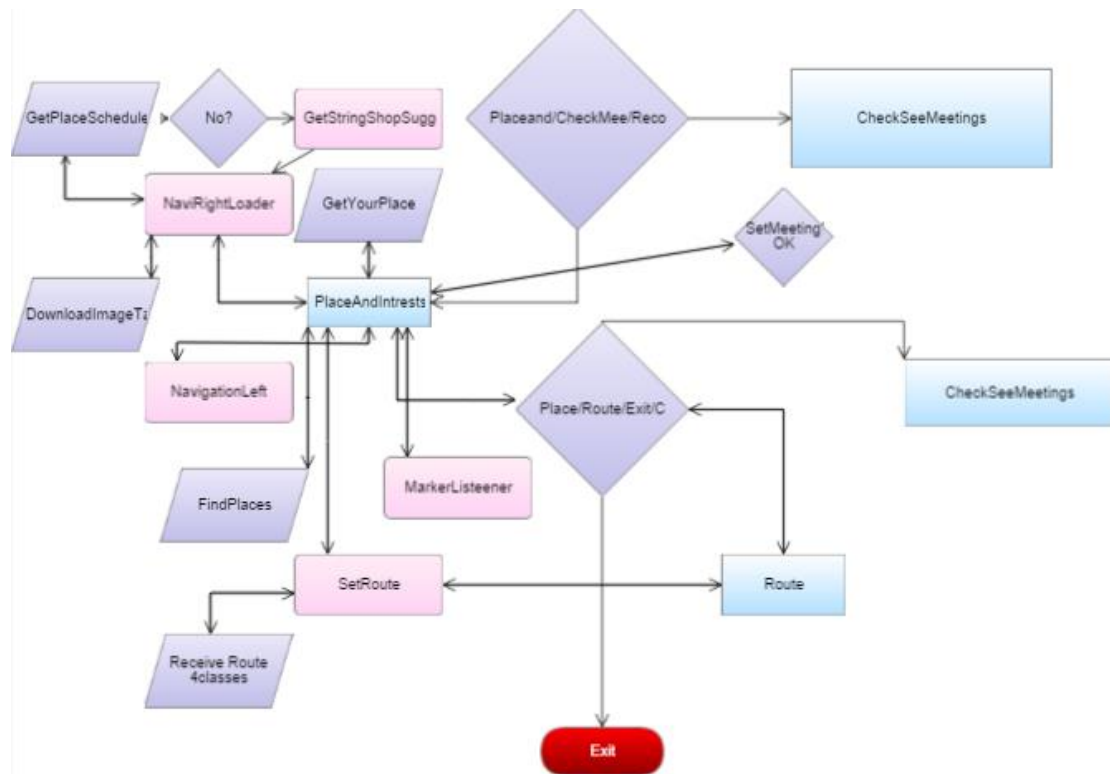
Τι είναι το activity

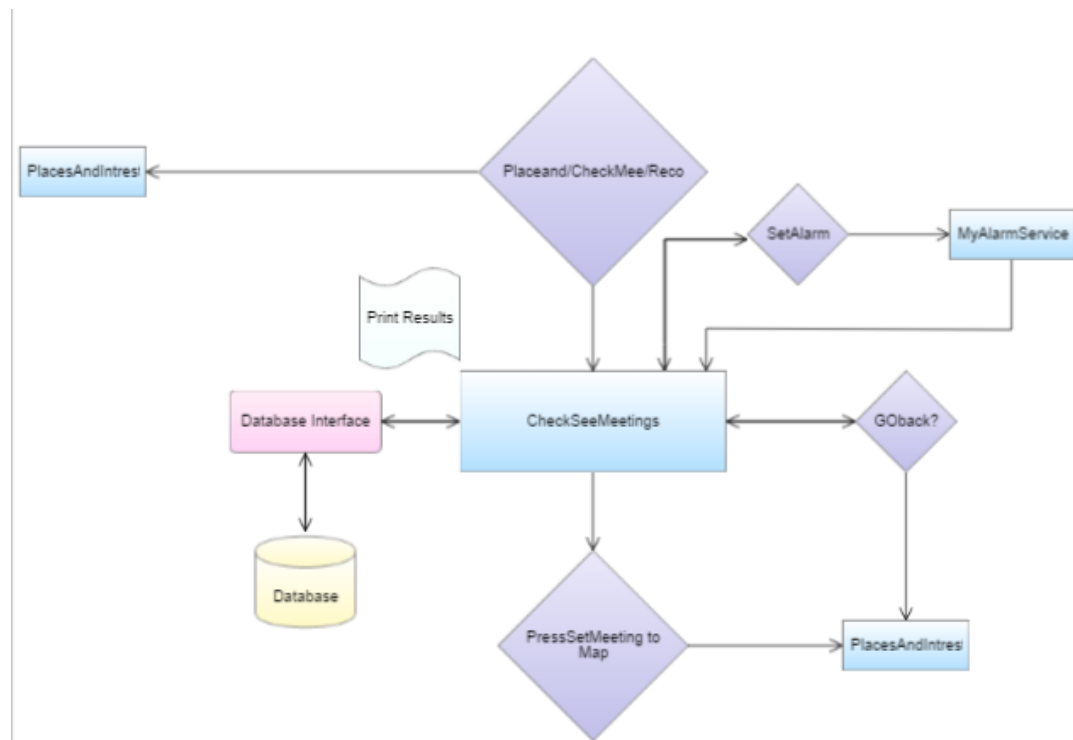
Μια δραστηριότητα είναι ένα στοιχείο στην εφαρμογή που παρέχει μια οθόνη με την οποία οι χρήστες μπορούν να αλληλεπιδρούν με σκοπό να κάνουν κάτι, όπως είναι να τραβήξετε μια φωτογραφία, να στείλετε ένα e-mail, ή να δείτε ένα χάρτη. Κάθε δραστηριότητα δίνεται σε ένα παράθυρο στο οποίο μπορείς να σχεδιάσεις την διεπαφή του χρήστη. Το παράθυρο γεμίζει συνήθως την οθόνη, αλλά μπορεί να είναι μικρότερο από την οθόνη και επιπλέον να είναι πάνω από άλλα παράθυρα. (Παράθεση από <https://developer.android.com/guide/components/activities.html>)

Flow Diagram (<https://dashboard.genmymodel.com>)

Ακολουθεί μια παρουσίαση φωτογραφιών πως λειτουργεί η ροή της εφαρμογής







Δημιουργία εφαρμογής

Προτού ξεκινήσω την εισαγωγή για την εφαρμογή στα παρακάτω link θα βρείτε που θα κατεβάσετε το android studio και οδηγίες για την εγκατάσταση του

1. Κατέβασμα

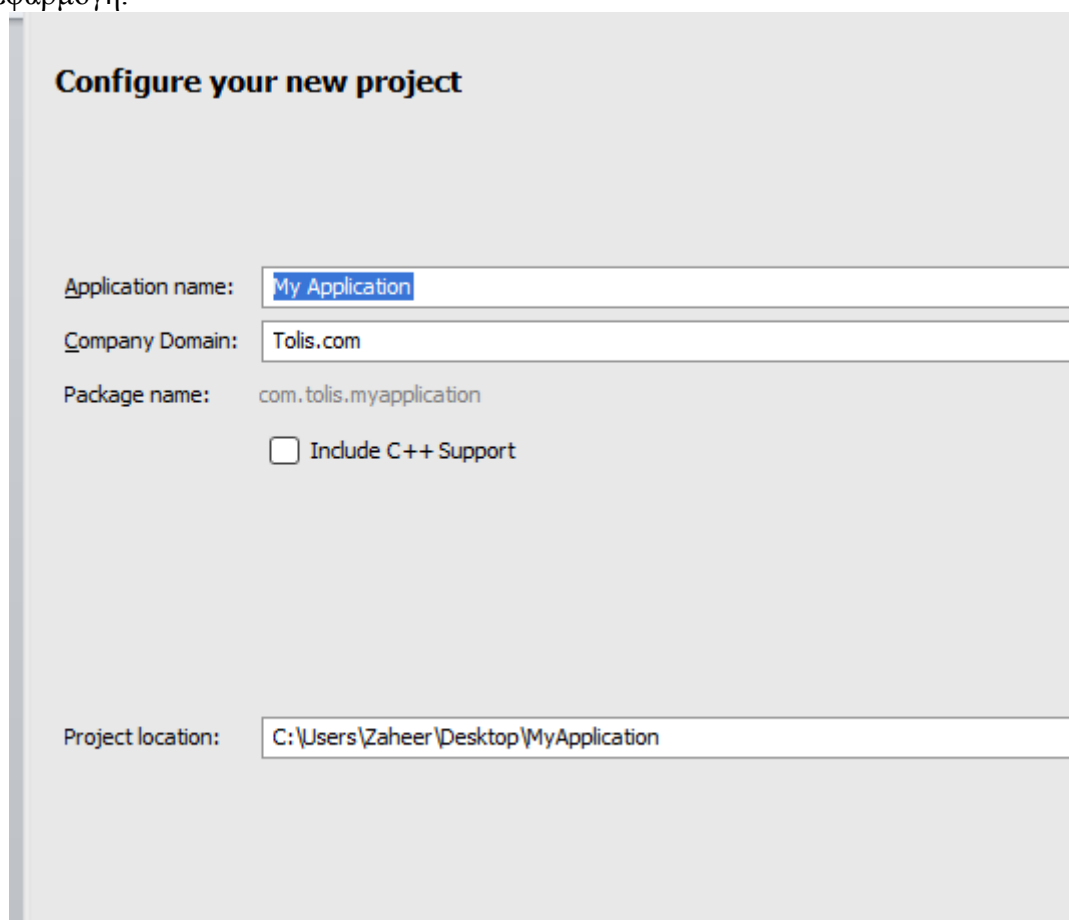
<https://developer.android.com/studio/index.html>

2. Οδηγίες

<https://developer.android.com/studio/install.html>

Δημιουργία του project

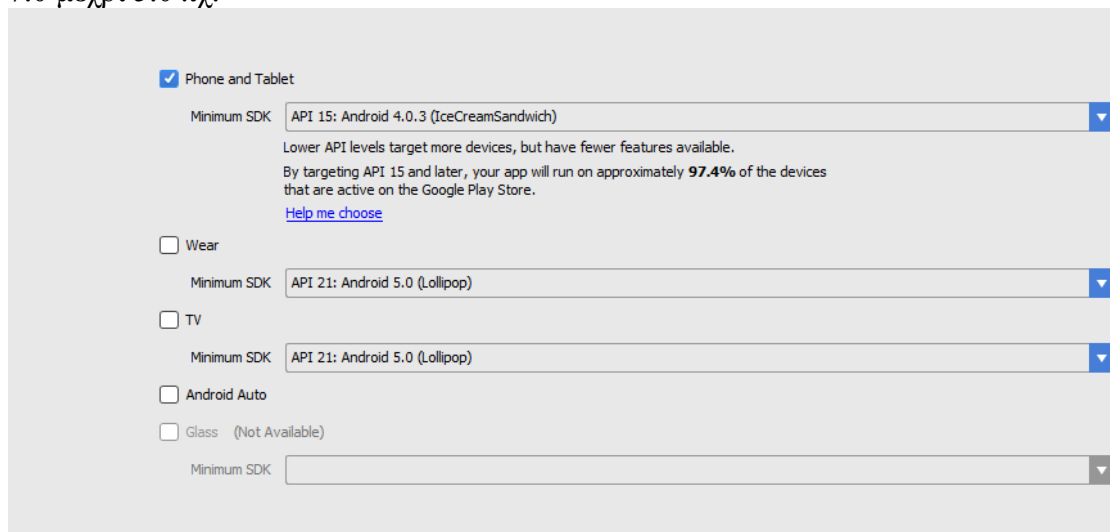
Μια μικρή εισαγωγή του πως θα κάνουμε το Project , όπως βλέπουμε και παρακάτω συμπληρώνουμε αναλόγως ...Όνομα εφαρμογής Όνομα ονομασίας πακέτου εκεί θα βρίσκονται τα αρχεία μας και είναι σημαντικό καθώς θα το ενώσουμε μετά με το google api , επίσης στο τέλος βλέπουμε και την τοποθεσία όπου θα αποθηκευτεί η εφαρμογή.



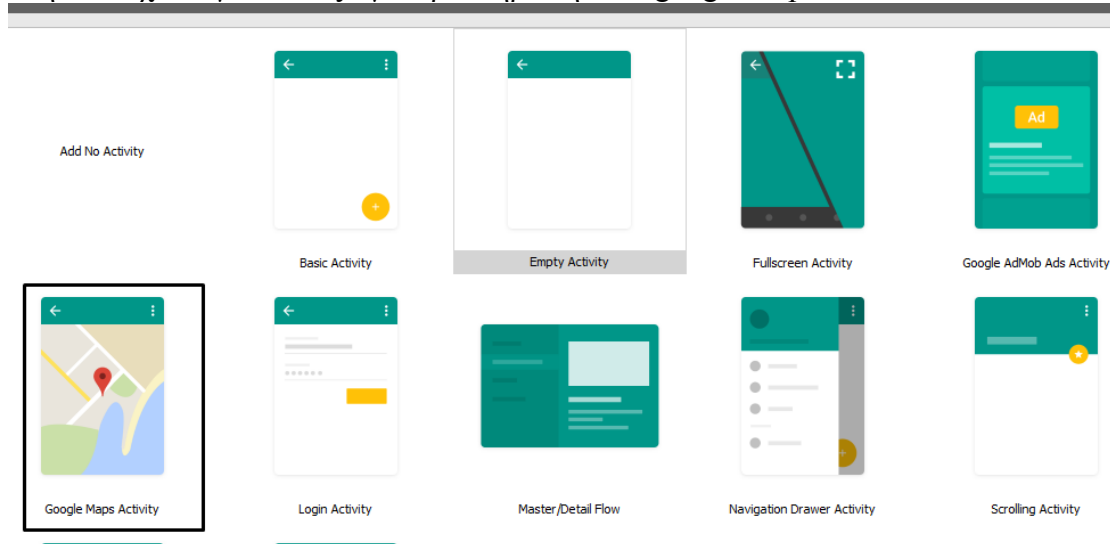
The screenshot shows the 'Configure your new project' dialog box in Android Studio. The dialog has a title bar and a main content area with the following fields and options:

- Application name:** My Application
- Company Domain:** Tolis.com
- Package name:** com.tolis.myapplication
- Include C++ Support
- Project location:** C:\Users\Zaheer\Desktop\MyApplication

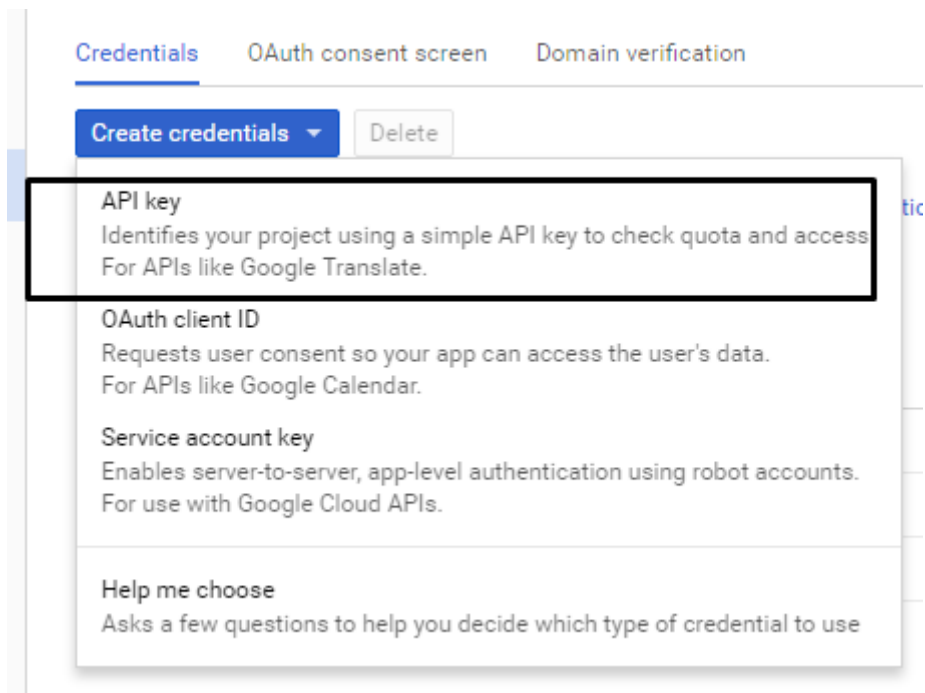
Και ακολουθεί από πού έως που θέλουμε η εφαρμογή να υποστηρίζεται από android 7.0 μέχρι 5.0 πχ.



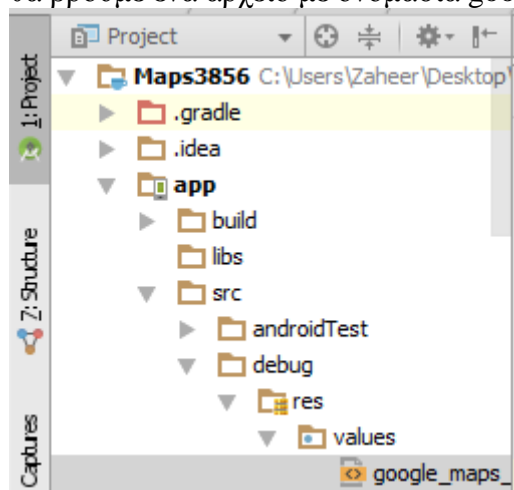
Στην συνέχεια εγώ επέλεξα για δραστηριότητα το google maps.



Προτού ξεκινήσει και λειτουργεί σωστά η εφαρμογή πρέπει να της δώσουμε ένα κλειδί επικοινωνίας με τους google servers αυτό θα το κάνουμε από εδώ <https://console.developers.google.com/apis/credentials>



Στο project μας κάπου θα βρούμε ένα αρχείο με ονομασία google maps api



Το αρχείο μοιάζει κάπως έτσι

```
<resources>
|
|  <!--
|  <!-- TODO: Before you run your application, you need a Google Maps API key.
|
|  To get one, follow this link, follow the directions and press "Create" at the end:
|
|  https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&keyType=CLIENT_SIDE_ANDROID&ar=E8:66:28:0C:5E:8D:FC:71:6
|
|  You can also add your credentials to an existing key, using this line:
|  [REDACTED].com.tolis.maps3856
|
|  Alternatively, follow the directions here:
|  https://developers.google.com/maps/documentation/android/start#get-key
|
|  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
|  string in this file.
|  -->
|  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">[REDACTED]</string>
|
| </resources>
```

Στην πρώτη μαυρισμένη γραμμή θα βρείτε SHA-1 certificate fingerprint της μορφής

E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8: E8

Και στην δεύτερη θα τοποθετήσετε το Api Key που σας δόθηκε για να το διασφαλίσουμε να μην το χρησιμοποιούν τρίτοι στην σελίδα της google εκεί που δείχνει τα κλειδιά και το θαυμαστικό αν πατήσουμε πάνω θα οδηγηθούμε στην παρακάτω εικόνα όπου για Key restriction θα βάλουμε Android apps package name όπως είναι στην εφαρμογή και SHA-1 από το αρχείο που είδαμε παραπάνω

Created by

API key

Name

API key 3

Key restriction

Key restriction lets you specify which web sites, IP addresses, or apps can use this key. [Learn more](#)

None

HTTP referrers (web sites)

IP addresses (web servers, cron jobs, etc.)

Android apps

iOS apps

Restrict usage to your Android apps (Optional)

Add your package name and SHA-1 signing-certificate fingerprint to restrict usage to your Android apps

Get the package name from your AndroidManifest.xml file. Then use the following command to get the fingerprint:

```
$ keytool -list -v -keystore mystore.keystore
```

Package name	SHA-1 certificate fingerprint
com.tolis.maps3856	

Έτσι λοιπόν αν πατήσουμε start θα μπορούμε να δούμε τον χάρτη της google χωρίς κάτι ιδιαίτερο πάνω όπως σημεία ενδιαφέροντος θέση και τα λοιπά.

Αρχείο Manifest

Κάθε εφαρμογή έχει ένα AndroidManifest αρχείο μέσα στο root directory. Το AndroidManifest παρέχει κύριες πληροφορίες σχετικά με την εφαρμογή που εκτελείται στο σύστημα του Android, το οποίο εννοείται πρέπει να το έχει εκτελέσει πριν εκτελέσει τον κώδικα της εφαρμογής. Ο κώδικας του αρχείου:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tolis.maps3856">

    <uses-permission
        android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">

        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

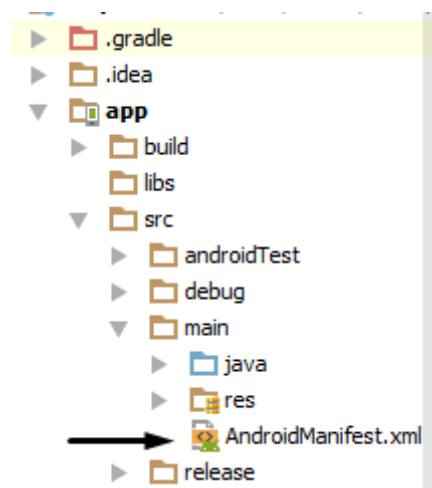
        <activity
            android:name="com.tolis.maps3856.Intro"
            android:configChanges="orientation|screenSize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="com.tolis.maps3856.RouteOnly"
            android:configChanges="orientation|screenSize"
            android:label="@string/title_activity_maps"></activity>
        <activity
            android:name="com.tolis.maps3856.SetMeeting"
            android:configChanges="orientation|screenSize"></activity>
        <activity
            android:name="com.tolis.maps3856.CheckSeeMeetings"
            android:configChanges="orientation|screenSize"></activity>
        <activity
            android:name="com.tolis.maps3856.PlaceAndIntrests"
            android:configChanges="orientation|screenSize"></activity>
        <activity
            android:name="com.tolis.maps3856.Signin"></activity>
        <activity
            android:name="com.tolis.maps3856.Signup">
        </activity>
        <activity
            android:name="com.tolis.maps3856.WelcomeUser"></activity>
        <receiver android:name=".MyAlarmService" />
    </application>

</manifest>

```

Στο κώδικα του AndroidManifest όπως δίνεται βλέπουμε τα εξής

1. Το όνομα του πακέτου `package="com.tolis.maps3856">` , αρχείων root της εφαρμογής
2. Τα δικαιώματα για την πρόσβαση της ακριβής θέσης του χρήστη
3. Το tag **meta-data** που θα πάρει το κλειδί του google api
4. Τις δραστηριότητες που θα τρέξουν
5. Τον receiver που είναι για το Alarm
6. Και φυσικά την πρώτη δραστηριότητα που θα τρέξει με τον άνοιγμα της εφαρμογής που επιτυγχάνεται με: `<intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter>`



build.gradle

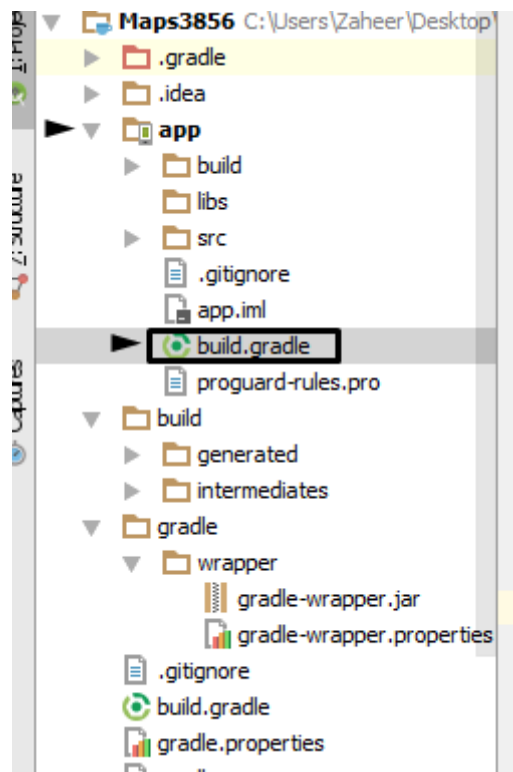
Κάθε εφαρμογή έχει ένα build.gradle .Τι είναι το build.gradle ; Είναι ένα σύστημα που συντάσσει τους πόρους, τον κώδικα της εφαρμογής, τα packages μέσα σε ένα executable αρχείο το λεγόμενο APKs.Το Android Studio χρησιμοποιεί το Gradle (<https://gradle.org/>), μια ειδικευμένη εργαλειοθήκη ώστε αυτόματα να χειρίζεται όλη την διαδικασία χτίσιματος καθώς σου επιτρέπει να επεξεργάζεσαι τι θα βάλεις μέσα πράγμα που το καθίστα ευέλικτο. Εννοείτε κάθε ρύθμιση στο χτίσιμο μπορεί να ορίσει δίκες της πόρους και κώδικες όσο επαναχρησιμοποιεί κομμάτια κοινά σε όλες τις έκδοσης της εφαρμογής.

```
apply plugin: 'com.android.application'
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"
    useLibrary 'org.apache.http.legacy'

    defaultConfig {
        applicationId "com.tolis.maps3856"
        minSdkVersion 16
        targetSdkVersion 23
        versionCode 1
        multiDexEnabled true
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:design:23.1.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.2.1'
    compile 'com.google.android.gms:play-services:8.4.0'
    compile 'joda-time:joda-time:2.9.4'
    compile 'com.android.volley:volley:1.0.0'
}
```

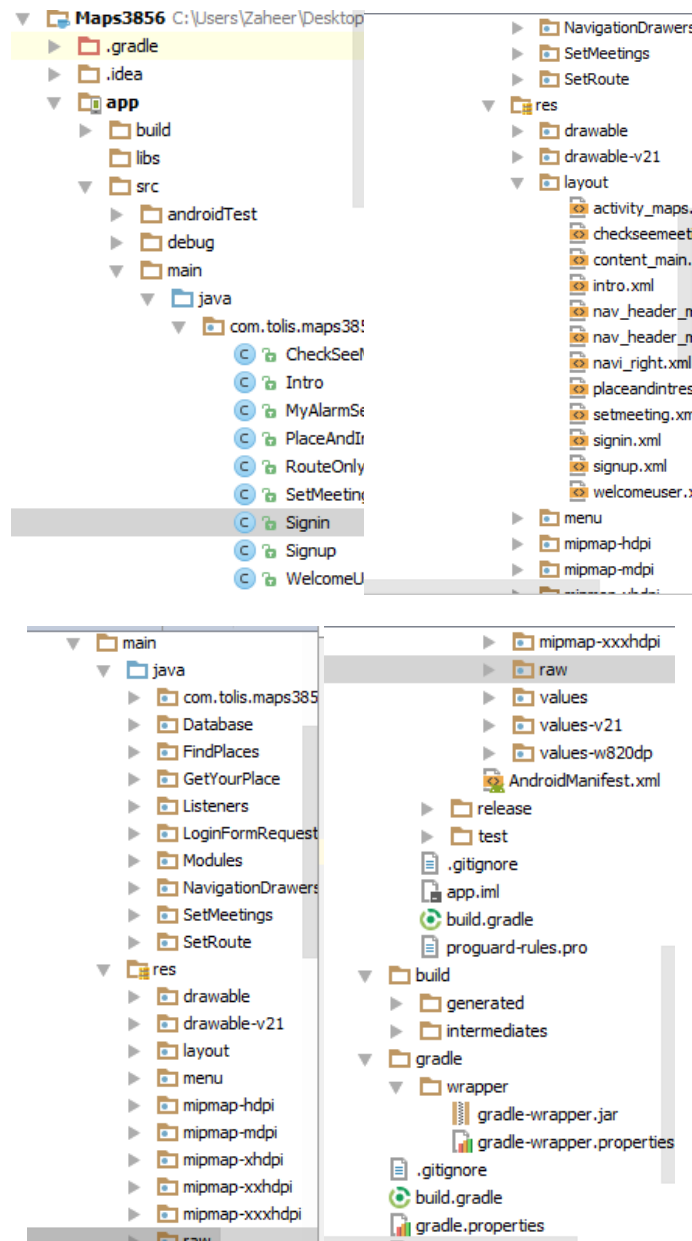
Όπως βλέπουμε στο αρχείο αυτό έχουμε compileSdkVersion 23 δηλαδή το android 6.0 όπου κτίζουμε στις βιβλιοθήκες πάνω αυτού και λίγο παρακάτω defaultConfig 16 για ιστορικούς λόγους εγώ πρόλαβα το 24 αλλά επειδή ήταν σε πειραματικό στάδιο έβαλα το 23. **Επίσης κύριο multiDexEnabled είναι αυτό το κομμάτι να το κάνουμε enable** διότι αν χτίσουμε πολλές μεθόδους θα μας πετάει error στην περίπτωση της εργασίας αυτής ενεργοποιήθηκε διότι χρειαζόταν. Και συνεχίζουμε με τα compiles όπου: **compile 'com.android.support:design:23.1.1'**
compile 'com.android.support:appcompat-v7:23.2.1'
compile 'com.android.volley:volley:1.0.0'
χρήσιμες βασικές βιβλιοθήκες που αξίζει να σημειωθεί ότι πρέπει να τις προσέχουμε ώστε λειτουργεί πάνω η εφαρμογή,
compile 'com.google.android.gms:play-services:8.4.0' τα λεγόμενα google play services **compile 'joda-time:joda-time:2.9.4'** timer που θα χρησιμεύσουν στην υπολειπομένη ώρα των συναντήσεων.

Όσον αφορά για τις βασικές βιβλιοθήκες όπως του **design** αυτή μπορεί να παρέχει διάφορα πράγματα σχεδίασης όπως navigation drawers επιφανειακά παράθυρα που είναι πάνω από άλλα scroll events κτλπ στην **appcompat** έχουμε προσθήκης βοήθειας για υλοποιήσεις κυρίων κλάσεων όπως Fragment and Loader κτλπ. Και η **volley** που είναι μια HTTP library και χρειάζεται τα διαδικτυακά κομμάτια.



Αυτά ήταν τα κύρια σημεία που πρέπει να ξέρει κάποιος πριν κτίσει μια εφαρμογή.

Οι παρακάτω εικόνες δείχνουν ένα μέρος της αρχειοθέτησης των αρχείων με τους φακέλους τους.



Intro

Σε αυτό το σημείο πρώτα θα φτιαχτεί το layout και το activity και μετά θα συμπληρώνει στο manifest διότι θα βγάλει error ότι δεν υπάρχει κλάση είναι μικρό το λάθος αλλά πρέπει να διευκρινιστεί, ας πάμε τώρα στο layout θέλει κάποιος να φτιάξει ένα εισαγωγικό βίντεο για την εφαρμογή είναι λογικό γιατί είναι ένα λογότυπο που διαφημίζει την εταιρία το άτομο κτλπ. Σε αυτή τη εργασία για λόγους ταχύτητας και ευκολίας έβαλα και ένα αόρατο κουμπί σε όλη την οθόνη για να κάνω skip.

Layout intro

```

<?xml version="1.0" encoding="utf-8"?>
<!--TA tools είναι οι "βιβλιοθήκες" για το xml-->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"

    tools:context="com.tolis.maps3856.Intro">
    <VideoView
        android:id="@+id/VideoView1"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:background="@android:color/transparent" />
</RelativeLayout>

```

RelativeLayout LinearLayout έχουν να κάνουν με το πως εμφανίζονται τα στοιχεία που έχω βάλει στην εφαρμογή πχ στο LinearLayout θα μπουν το ένα κάτω από το άλλο σαν μια στοίβα ,στην συνέχεια μέσα στα tags βλέπουμε layout width και height έχουμε να διαλέξουμε σε τρεις επιλογές, μια να βάλουμε δικές μας τιμές που δεν το θεωρώ καλό γιατί από οθόνη σε οθόνη θα έχει διαφορές παραδείγματος χάριν μικρή εικόνα και σε fill parent= match parent (νέα έκδοση) με wrap content όπου το fill θα γεμίσει πλήρως ενώ το wrap όσο χρειάζεται το preset μέγεθος ενός κουμπιού πχ. Στο video view δημιουργούμε ένα παράθυρο εμφάνισης του βίντεο όπου καλό θα είναι γενικά σε όλα να δίνουμε πάντα ένα id ποτέ δεν ξέρουμε αν χρειαστεί μετέπειτα η όχι και παρακάτω το κουμπί που αναφέρθηκε παραπάνω.

Class Intro

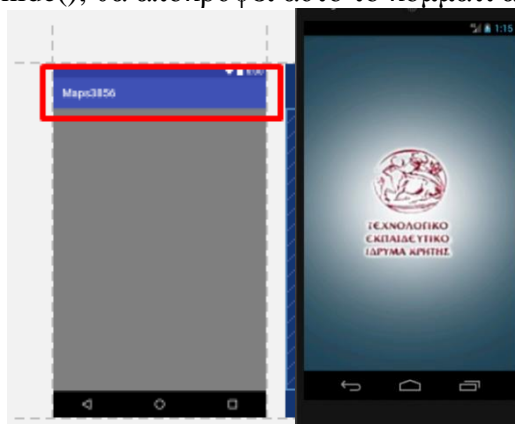
```
package com.tolis.maps3856;
import android.content.Intent;
import android.graphics.PixelFormat;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.VideoView;

public class Intro extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.intro);
        getSupportActionBar().hide();
        Button buttonPlayVideo2 = (Button) findViewById(R.id.button1);
        getWindow().setFormat(PixelFormat.UNKNOWN);
        VideoView mVideoView2 = (VideoView) findViewById(R.id.VideoView1);
        String uriPath2 = "android.resource://com.tolis.maps3856/" + R.raw.fum;
        Uri uri2 = Uri.parse(uriPath2);
        mVideoView2.setVideoURI(uri2);
        mVideoView2.requestFocus();
        mVideoView2.start();
        mVideoView2.setOnCompletionListener(new
MediaPlayer.OnCompletionListener() {

            @Override
            public void onCompletion(MediaPlayer mp) {
                Intent intent = new Intent(Intro.this, Signin.class);
                Intro.this.startActivity(intent);
                Intro.this.finish();
            }
        });
        // videoView.sto
        buttonPlayVideo2.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(Intro.this, Signin.class);
                Intro.this.startActivity(intent);
                Intro.this.finish();
            }
        });
    }
}
```

Το package δηλώνει την ονομασία που είχαμε δώσει στην αρχή , τα imports είναι προσθήκες βιβλιοθηκών που πρέπει να γίνουν compile αν κάποιος δεν χρειάζεται το android φροντίζει να σας τα υποδείξει , η ερώτηση που θα μπορούσε να κάνει κάποιος μα καλά έχει όλες τις βιβλιοθήκες έτοιμο το android; Φυσικά όχι για αυτό έχει το gradle που κατεβάζει την κυρίως βιβλιοθήκη πχ volley για http και μετά προσθέτεις ποιο κομμάτι η τομέας της να μπει στο project σου διότι άμα είχε πλήρως την βιβλιοθήκη και αχρείαστη θα ήταν και μη χρησιμοποιημένη . **Η onCreate που θα δούμε σε όλες τις κλάσεις activity είναι σαν την main της c είναι αυτή που θα τρέχει πάντα πρώτη με το που δημιουργηθεί η κλάση.** Λίγο παρακάτω βλέπουμε την setContentView που με αυτή υποδεικνύουμε ότι η εικόνα το layout που θέλουμε να δείξουμε και να “ενώσουμε” με την κλάση είναι το layout.intro. Το getSupportActionBar.hide(); θα αποκρύψει αυτό το κομμάτι από την εφαρμογή.



Στην συνέχεια βλέπουμε την δημιουργία ενός κουμπιού και τον setOnClickListener που άμα γίνει trigger θα τρέξει αυτό τον κώδικα `Intent = new Intent(Intro.this, Signin.class); this.startActivity(intent); Intro.this.finish();` Ο όποιος θα δημιουργήσει ένα νέο activity και θα το τρέξει σβήνοντας το προηγούμενο με την εντολή finish. Στο video σύμφωνα με τον κώδικα θα δημιουργήσουμε ένα video view όπου θα του δώσουμε id αυτό με το Layout ώστε να το δείξει εκεί που το έχουμε θέσει , μετά θα συμπληρώσουμε το string location του video `String uriPath2 = "android.resource://com.tolis.maps3856/" + R.raw.fum;`



θα το ξεκινήσουμε και θα του βάλουμε έναν listener με την μέθοδο του `public void onCompletion(Media Player MP)` και όταν τελειώσει θα γίνει ότι με την μέθοδο του button.

Layout signin

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/login"
    android:orientation="vertical"
    android:weightSum="1">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Username"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/TFu"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView3"
        android:inputType="none"
        android:textAppearance="?android:attr/textAppearanceLarge"
    />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/TFu"
        android:layout_marginTop="45dp"
        android:text="Password"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/TFp"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView3"
        android:ems="10"
        android:inputType="textPassword"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:textColorLink="#000000" />

    <TextView
        android:id="@+id/textView15"
```

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1.00" />
```

<Button

```
android:layout_marginTop="3dp"  
android:layout_marginLeft="3dp"  
android:layout_marginRight="3dp"  
android:id="@+id/bl"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_above="@+id/bSu"  
android:alpha="0.7"  
android:background="@drawable/button_rounded"  
android:onClick="OnButtonClick"  
android:text="Login" />
```

<Button

```
android:layout_marginTop="3dp"  
android:layout_marginLeft="3dp"  
android:layout_marginRight="3dp"  
android:id="@+id/bSu"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:alpha="0.7"  
android:background="@drawable/button_rounded"  
android:onClick="OnButtonClick"  
android:text="SignUp" />
```

<Button

```
android:layout_marginTop="3dp"  
android:layout_marginLeft="3dp"  
android:layout_marginRight="3dp"  
android:id="@+id/skip"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:alpha="0.7"  
android:onClick="OnButtonClick"  
android:background="@drawable/button_rounded"  
android:text="SkipTest" />
```

</LinearLayout>

Τα νέα στοιχεία που παρατηρούνται είναι
android:background="@drawable/login"

Είναι η εικόνα που μπαίνει στο background αυτή είναι τοποθετημένη στον φάκελο drawable λίγο παραπάνω στις αρχιεπιθέσεις μπορεί κανείς να δει που είναι
android:orientation="vertical"

Ένα LinearLayout είναι γραμμικό αλλά σαν default είναι οριζόντια γραμμικό για να φανούν λοιπόν όλα τα στοιχεία επιλεγούμε να είναι κατακόρυφο

android:inputType="textPassword "

Απαγορεύετε η εμφάνιση κειμένου στο χρήστη μόνο σε μορφή κωδικοποιημένη είναι βέβαια πολύ χρήσιμο για την απόκρυψη κωδικού.

android:text="Username"

Θέτουμε ένα κείμενο μέσα στο αντικείμενο

android:textAppearance="?android:attr/textAppearanceLarge" />

Ο τρόπος που θα παρουσιαστεί η μορφή κειμένου εδώ συγκεκριμένα μεγάλο φόντο
<TextView>

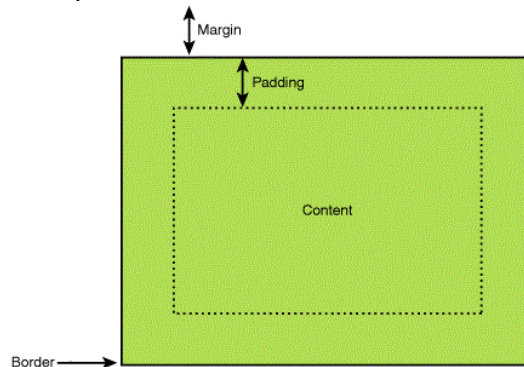
Είναι το αντικείμενο που μπορεί κανείς να θέση σαν επικεφαλίδα ώστε να δηλώσει κάτι όπως ότι το παρακάτω κείμενο αφορά το όνομα χρήστη

<EditText>

Είναι το αντικείμενο που μπορεί κανείς να εισάγει κείμενο χρήσιμο για φόρμες αποστολή μηνυμάτων κτλπ.

android:layout_marginLeft="3dp"

Το dp είναι μια μονάδα μέτρησης του android που διαφέρει από συσκευή σε συσκευή επειδή θα δούμε πολλές ρυθμίσεις σχετικά με content και τον τρόπο παρουσίασης του καλό στο νου να έχουμε αυτή την εικόνα όπου υποδεικνύει πως οι αλλαγές επηρεάζουν την εμφάνιση του content



android:alpha="0.7"

Σε περίπτωση που θέλουμε ένα button να είναι ποιο διαφανές

android:onClick="OnButtonClick"

Όταν θέλουμε να προκαλέσουμε μια μορφή listener μέσα από το layout και να εκτελεστεί κάποια μέθοδος.

android:background="@drawable/button_rounded"

Είναι ένα αρχείο με έτοιμες ρυθμίσεις για αλλαγή εμφάνισης στα κουμπιά που βρίσκεται μέσα στον φάκελο drawable

<?xml version="1.0" encoding="utf-8" ?>

```

<selector xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:state_pressed="true">
    <shape android:shape="rectangle">
      <corners android:radius="3dip" />
      <stroke android:width="1dip" android:color="#5e7974" />
      <gradient android:angle="-90" android:endColor="#689a92"
android:startColor="#345953" />
    </shape>
  </item>
  <item android:state_focused="true">
    <shape android:shape="rectangle">
      <corners android:radius="3dip" />
      <stroke android:width="1dip" android:color="#5e7974" />
      <solid android:color="#87ABA6" />
    </shape>
  </item>
  <item>
    <shape android:shape="rectangle">
      <corners android:radius="3dip" />
      <stroke android:width="1dip" android:color="#5e7974" />
      <gradient android:angle="-90" android:endColor="#9aaab3"
android:startColor="#D2D2D2" />
    </shape>
  </item>
</selector>

```

Ένα κουμπί δεν έχει πάντα μια κατάσταση αναλόγως αν πατιέται η έχουμε από πάνω το ποντίκι θα πρέπει να έχει μια διάδραση ώστε να δείξουμε ότι είναι ενεργό όπως βλέπουμε ο παραπάνω κώδικα αποτελεί αυτό το πράγμα κάτω θα δουμε το σχημα `<shape android:shape="rectangle">` τι γωνίες θελουμε να εχει χρωμα και τροπος εμφανισης μετα εχουμε το `<item android:state_pressed="true">` όπου δηλώνουμε τι θα γίνει αν πατηθεί και `<item android:state_focused="true">` αν χρησιμοποιεί κάποιος dpad η trackball όχι για touch .

Εικόνα από το Signin Layout



Class Signin

```
package com.tolis.maps3856;
import android.app.AlertDialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;
import org.json.JSONException;
import org.json.JSONObject;
import LoginFormRequests.LoginRequest;

public class Signin extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.signin);
        getSupportActionBar().hide();
    }

    public void OnButtonClick(View v) {
        if (v.getId() == R.id.bSu) {

            Intent intent = new Intent(Signin.this, Signup.class);
            Signin.this.startActivity(intent);
            Signin.this.finish();
        }

        if (v.getId() == R.id.skip) {
            Intent intent = new Intent(Signin.this, WelcomeUser.class);
            intent.putExtra("username", "TestUsername");
            intent.putExtra("id", "9999");
            intent.putExtra("email", "TestEmail");
            intent.putExtra("fname", "TestfName");
            Signin.this.startActivity(intent);
            Signin.this.finish();
        }

        if (v.getId() == R.id.bl) {

            EditText a = (EditText) findViewById(R.id.TFu);
            String str = a.getText().toString();
```

```

EditText b = (EditText) findViewById(R.id.TFp);
String str2 = b.getText().toString();

final String username = str;
final String password = str2;

Response.Listener<String> responseListener = new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject jsonResponse = new JSONObject(response);
            boolean success = jsonResponse.getBoolean("success");

            if (success) {
                String name = jsonResponse.getString("email");
                String fname = jsonResponse.getString("fname");
                int id = jsonResponse.getInt("id");

                Intent intent = new Intent(Signin.this,
com.tolis.maps3856.WelcomeUser.class);
                intent.putExtra("username", username);
                intent.putExtra("id", String.valueOf(id));
                intent.putExtra("email", name);
                intent.putExtra("fname", fname);
                Signin.this.startActivity(intent);
                Signin.this.finish();
            } else {
                AlertDialog.Builder builder = new AlertDialog.Builder(Signin.this);
                builder.setMessage("Login Failed")
                    .setNegativeButton("Retry", null)
                    .create()
                    .show();
            }

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
};

LoginRequest loginRequest = new LoginRequest(username, password,
responseListener);
RequestQueue queue = Volley.newRequestQueue(Signin.this);
queue.add(loginRequest);
}

}
}

```


Σε αυτήν την κλάση έχουμε τρεις επιλογές 3 κουμπιά μια να κάνουμε σύνδεση μια εγγραφή και μια δοκιμαστικό login test. Αν πατηθεί το Skip τότε θα δώσουμε μερικές ψεύτικες δοκιμαστικές μεταβλητές στο Intent και θα συνεχίσουμε στην νέα κλάση ,αν πατήσουμε την Sign Up τότε θα πάμε στο layout εγγραφής όπου και πάλι σβήνουμε το προηγούμενο activity .Σε περίπτωση όμως που πατηθεί το sign in γίνεται μια διαδικασία πρώτον δημιουργούμε έναν response listener ο οποίος δέχεται δεδομένα string που είναι ουσιαστικά τύπου json και αναλόγως επιτυχίας η αποτυχίας έχει και το κατάλληλο μήνυμα και στην συνέχεια είμαστε έτοιμοι να δημιουργήσουμε την εντολή σύνδεσης η οποία θα περιγραφεί παρακάτω και μια ουρά αιτημάτων κάτω από αυτό για την εκτέλεση του. Περισσότερη πληροφορία πως λειτουργεί το Request θα την βρείτε εδώ <https://developer.android.com/training/volley/simple.html>. Προτού συνεχίσω περιγράφοντας την κλάση LoginRequest θα ήθελα να αναφέρω ότι το json object το προτιμώ για μορφή λήψης δεδομένων λόγω ασχολίας με το google Api κατάλαβα αν θες να στείλεις δεδομένα με πολλές μεταβλητές που μπορεί να είναι κλεισμένες σε πίνακα σαν δομή είναι πολύ εύκολο να το αποκωδικοποίησης αντί να χρησιμοποιείς διάφορες μεθόδους διαχωρισμού τιμών πχ.

```

"results" : [
  {
    "geometry" : {
      "location" : {
        "lat" : 36.4353311,
        "lng" : 28.2243919
      },
      "viewport" : {
        "northeast" : {
          "lat" : 36.43548615,
          "lng" : 28.22444385
        },
        "southwest" : {
          "lat" : 36.43486595,
          "lng" : 28.22428725
        }
      }
    },
    "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/shopping-71.png",
    "id" : "378bad7ec987f1177cee21d534b04e68621ee988",
    "name" : "SUPERMARKET SPANOS, J. CH., GROUP OF COMPA",
    "opening_hours" : {
      "open_now" : true,
      "weekday_text" : []
    },
    "photos" : [
      {
        "height" : 2340,
        "html_attributions" : [
          "\u003ca href=\\"https://maps.google.com/maps/contrib/115853780114203717194/photos/\\" \u003epauline wong\u003c/a\u003e"
        ],
        "photo_reference" : "CoQBdwAAABhKqJBPx0X3MAM0uIPvyMaFPXyh9mBt7x_K_tpHC9c7EChaeobggwrCEy8BLK6v_Cif5S2Qk0wPatjWYJfgxGAP1E9hj1tD0CFk",
        "width" : 4160
      }
    ],
    "place_id" : "ChIJj11cR3Rh1RQRN6CLc_OCQI",
    "rating" : 4,
    "reference" : "CmRRAAAA8sH85xyba08ZfQtY1i1pmmeTFwAehtGtBL4LAn6cml.d1LBs0NTANLJdm1a0xQEK2fa8BR9XDnqBpjh_fC0unhBXhwQz5TCyfeZKfL4vWuuEr5PXL",
    "scope" : "GOOGLE",
    "types" : [
      "atm",
      "grocery_or_supermarket",
      "Food",
      "finance",
      "store",
      "point_of_interest",
      "establishment"
    ],
    "vicinity" : "Megalou Konstantinou 53, Rodos"
  },
  {
    "geometry" : {
      "location" : {
        "lat" : 36.4508372,
        "lng" : 28.2227142
      }
    }
  }
]

```

Εδώ πέρα καταλαβαίνει κάποιος τι όγκο και τι πολυπλοκότητα μπορεί να έχει ένα αίτημα για πληροφορία οπότε ο διαχωρισμός με αγκύλες είναι ο κατάλληλος όπου το μεγάλο είναι το Json Array όπου διαχωρίζεται μέχρι το επόμενο geometry σαν πίνακας και το geometry οι αγκύλες του η το location μπορεί να θεωρηθούν για Json Object.

Class LoginRequest

```
package LoginFormRequests;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;
import java.util.HashMap;
import java.util.Map;

public class LoginRequest extends StringRequest {
    private static final String LOGIN_REQUEST_URL = "http://.....tk/Login.php";
    private Map<String, String> params;

    public LoginRequest(String username, String password,
Response.Listener<String> listener) {
        super(Method.POST, LOGIN_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("username", username);
        params.put("password", password);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}
```

Στο κώδικα login request βλέπουμε το πέρασμα username password και του Listener , στο Link τις τοποθεσίας που είναι το αρχείο Php μέσα στις παραμέτρους του αιτήματος βάζουμε μόνο το όνομα και τον κωδικό καθώς μόνο αυτά χρειάζονται και έτσι επιστρέφουμε στην αρχική κλάση Signin.

Layout signup

```
<?xml version="1.0" encoding="utf-8" ?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@drawable/login"
        android:orientation="vertical"
        android:weightSum="1">

        <TextView
            android:id="@+id/textView4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:scrollbars="vertical"
            android:text="FirstName"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <EditText
            android:id="@+id/TFfn"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/textView5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="LastName"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <EditText
            android:id="@+id/TFln"
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

        <TextView
            android:id="@+id/textView6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Email"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <EditText
            android:id="@+id/TFe"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

<TextView
    android:id="@+id/textView7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Username"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
    android:id="@+id/TFuser"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<TextView
    android:id="@+id/textView8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Password"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
    android:id="@+id/TFpass"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPassword" />

<TextView
    android:id="@+id/textView9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="ConfirmAgainPassword"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
    android:id="@+id/TFcap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPassword" />

<TextView
    android:id="@+id/textView10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Adress"
    android:textAppearance="?android:attr/textAppearanceLarge" />

```

```

<EditText
    android:id="@+id/TFadd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<TextView
    android:id="@+id/textView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Zip Code"
    android:textAppearance="?android:attr/textAppearanceLarge" />

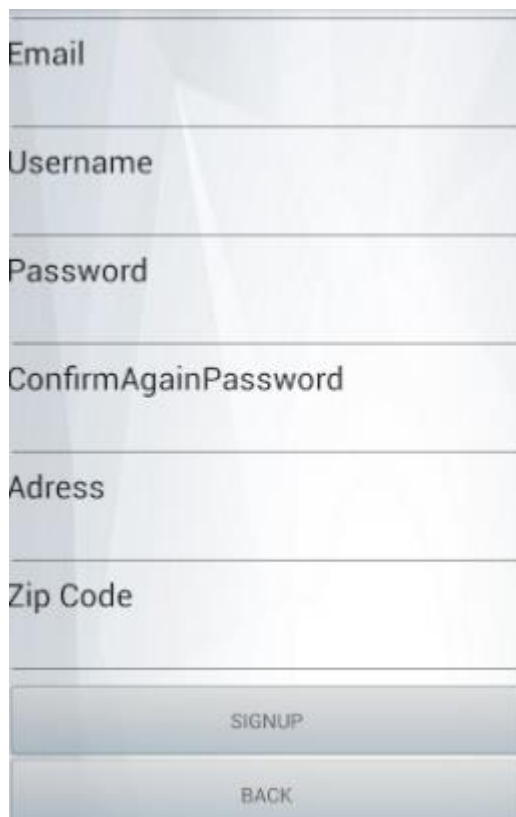
<EditText
    android:id="@+id/TFzc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<Button
    android:layout_marginTop="3dp"
    android:layout_marginLeft="3dp"
    android:layout_marginRight="3dp"
    android:id="@+id/bSu2"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:alpha="0.7"
    android:background="@drawable/button_rounded"
    android:onClick="OnSignUpClick"
    android:text="SignUp" />

<Button
    android:layout_marginTop="3dp"
    android:layout_marginLeft="3dp"
    android:layout_marginRight="3dp"
    android:id="@+id/bb"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:alpha="0.7"
    android:background="@drawable/button_rounded"
    android:onClick="OnBackClick"
    android:text="Back" />
</LinearLayout>
</ScrollView>

```

Όσον αφορά αυτό το layout δεν έχουμε κάτι καινούργιο εκτός από ότι το κλείνουμε σε scroll view διότι είναι πολύ μεγάλο να φανούν όλα τα πεδία. Και Εδώ ακλούθησα όπως το sign in παρόμοια λογική.



The image shows a vertical list of input fields for a sign-up form. The fields are labeled: Email, Username, Password, ConfirmAgainPassword, Adress, and Zip Code. Below the fields are two buttons: SIGNUP and BACK. The form is displayed on a light blue background.

Class Signup

```
package com.tolis.maps3856;  
import android.app.ProgressDialog;  
import android.content.Intent;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.support.v7.app.AppCompatActivity;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.Toast;  
import java.util.HashMap;  
import LoginFormRequests.RegisterUserClass;
```

```
public class Signup extends AppCompatActivity {
```

```
    private static final String REGISTER_URL = "http://.....tk/register.php";
```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.signup);
    getSupportActionBar().hide();
}

public void OnSignUpClick(View v) {
    if (v.getId() == R.id.bSu2) {

        EditText a = (EditText) findViewById(R.id.TFfn);
        EditText b = (EditText) findViewById(R.id.TFln);
        EditText c = (EditText) findViewById(R.id.TFe);
        EditText d = (EditText) findViewById(R.id.TFuser);
        EditText e = (EditText) findViewById(R.id.TFpass);
        EditText f = (EditText) findViewById(R.id.TFadd);
        EditText g = (EditText) findViewById(R.id.TFzc);
        EditText h = (EditText) findViewById(R.id.TFcap);

        String fname = a.getText().toString();
        String lname = b.getText().toString();
        String email = c.getText().toString();
        String username = d.getText().toString();
        String password = e.getText().toString();
        String pass2 = h.getText().toString();
        String add = f.getText().toString();
        String zip = g.getText().toString();

        if (!password.equals(pass2)) {
            Toast passTest = Toast.makeText(Signup.this, "Passwords does not
much", Toast.LENGTH_SHORT);
            passTest.show();
        } else if (validation(fname, lname, email, username, password, add, zip)) {
            //do nothing
        }

        else {
            username = username.trim().toLowerCase();
            password = password.trim().toLowerCase();
            fname = fname.trim().toLowerCase();
            lname = lname.trim().toLowerCase();
            email = email.trim().toLowerCase();
            add = add.trim().toLowerCase();
            zip = zip.trim().toLowerCase();
            register(username, password, fname, lname, email, add, zip);
        }
    }
}
}

```

```

private boolean validation(String fname, String lname, String email, String
username, String password, String add, String zip) {

    if (!fname.matches("[a-zA-Z]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Fname",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!lname.matches("[a-zA-Z]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Lname",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!username.matches("[a-zA-Z0-9_\\.\\-]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Username",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!password.matches("[a-zA-Z0-9_\\.\\-]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Password",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!add.matches("[a-zA-Z0-9_\\.\\-]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Add",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!zip.matches("[0-9_\\.\\-]+")) {
        Toast.makeText(getApplicationContext(), "Check Field Zip",
Toast.LENGTH_SHORT).show();
        return true;
    }

    if (!email.matches("[a-zA-Z0-9_\\.\\-]+@[a-zA-Z0-9_\\.\\-]+\\.[a-zA-Z0-
9]{2,4}")) {
        Toast.makeText(getApplicationContext(), "Check Field Email",
Toast.LENGTH_SHORT).show();
        return true;
    }
    return false;
}

public void OnBackClick(View v) {

```



```

if (v.getId() == R.id.bb) {
    Intent ir = new Intent(Signup.this, Signin.class);
    startActivity(ir);
    this.finish();
}
}

private void register(String username, String password, String fname, String
lname, String email, String add, String zip) {
    class RegisterUser extends AsyncTask<String, Void, String> {
        ProgressDialog loading;
        RegisterUserClass ruc = new RegisterUserClass();

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            loading = ProgressDialog.show(Signup.this, "Please Wait", null, true,
true);
        }

        @Override
        protected void onPostExecute(String s) {
            super.onPostExecute(s);
            loading.dismiss();
            Toast.makeText(getApplicationContext(), s,
Toast.LENGTH_LONG).show();
        }

        @Override
        protected String doInBackground(String... params) {

            HashMap<String, String> data = new HashMap<String, String>();
            data.put("username", params[0]);
            data.put("password", params[1]);
            data.put("fname", params[2]);
            data.put("lname", params[3]);
            data.put("email", params[4]);
            data.put("add", params[5]);
            data.put("zip", params[6]);
            String result = ruc.sendPostRequest(REGISTER_URL, data);

            return result;
        }
    }

    RegisterUser ru = new RegisterUser();
    ru.execute(username, password, fname, lname, email, add, zip);
}

```

Σε αυτή την κλάση θα δούμε δύο κουμπιά το Back και το Sign Up. Όπου το Back θα εκτελέσει αυτόν τον κώδικα `Intent ir = new Intent(Signup.this, Signin.class); startActivity(ir); this.finish();` και θα πάει πίσω στην Signin κλείνοντας το προηγούμενο activity. Σε περίπτωση που κάποιος πατήσει το Sign up γίνονται τα εξής:

- Εισαγωγή τιμών από τα πεδία

```
EditText a = (EditText) findViewById(R.id.TFfn);  
Βρίσκουμε το id του πεδίου και παίρνουμε την τιμή  
String fname = a.getText().toString();  
Μετά την μετατρέπουμε σε string
```

- Έλεγχος αν ο κώδικας είναι διπλότυπος

```
!password.equals(pass2)
```

- Έλεγχο πεδίων αν είναι λογικά σωστά δηλαδή το email να είναι της μορφής xxx@xxx.xx

```
if (!email.matches("[a-zA-Z0-9_\\-]+@[a-zA-Z0-9\\-]+\\.+[a-zA-Z0-9]{2,4}"))  
τι εννοούμε με αυτό το [a-zA-Z0-9_\\-] δηλώνει ότι θέλουμε τιμές από την αλφάβητο  
κεφαλαία η μικρά ,δεκτοί οι αριθμοί η _ και η - μετά λέμε + δηλαδή θέλουμε και κάτι  
άλλο @[a-zA-Z0-9\\-] της μορφής @hotmail09- και το τελευταίο κομμάτι δηλώνει  
.+[a-zA-Z0-9]{2,4}" ότι μετά την τελεία θέλουμε να είναι από 2 μέχρι 4 χαρακτήρες  
από αλφάβητο αριθμούς . Παρόμοια είναι και τα αλλά και ποιο απλοποιημένα.
```

- Αφού όλα είναι εντάξει θα πάει στην else όπου θα κάνει τα γράμματα μικρά trim δηλαδή θα τα κουρέψει

Αυτά ήταν τα βήματα για ετοιμαστούμε ,άρα τώρα μπορεί να δημιουργηθεί το request όσο αφορά αυτόν τον τρόπο δεν είναι ο ίδιος με πάνω και είναι λίγο πιο σύνθετος. Για αυτό θα δημιουργήσουμε μια AsyncTask μια διαδικασία που δεν τρέχει μαζί με την κυρίως. Στην `onPreExecute` και `onPostExecute` είναι οι μέθοδοι που χρησιμεύουν για την εμφάνιση και σβήσιμο του loading animation όταν δηλαδή ξεκινάμε θα εκτελέσουμε την `OnPre` και όταν τελειώσουμε την `OnPost`. Στην `do in Background` περνάμε τα αρχεία στην `RegisterUserClass` και αφού τελειώσουμε. Τα κάνουμε `execute`. Για να περιγραφτεί όλο το κομμάτι από κάτω θα σχολιάσω την κλάση όπου αναφέρθηκα λίγο παραπάνω.

Class RegisterUserClass

```
package LoginFormRequests;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.InputStreamReader;
import java.io.OutputStream;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.net.URLEncoder;
import java.util.HashMap;
import java.util.Map;
import javax.net.ssl.HttpURLConnection;

public class RegisterUserClass {

    public String sendPostRequest(String requestURL,
                                  HashMap<String, String> postDataParams) {

        URL url;
        String response = "";
        try {
            url = new URL(requestURL);

            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(15000);
            conn.setConnectTimeout(15000);
            conn.setRequestMethod("POST");
            conn.setDoInput(true);
            conn.setDoOutput(true);

            OutputStream os = conn.getOutputStream();
            BufferedWriter writer = new BufferedWriter(
                new OutputStreamWriter(os, "UTF-8"));
            writer.write(getPostDataString(postDataParams));

            writer.flush();
            writer.close();
            os.close();
            int responseCode=conn.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP_OK) {
                BufferedReader br=new BufferedReader(new
InputStreamReader(conn.getInputStream()));
                response = br.readLine();
            }
        }
    }
}
```

```

    }
    else {
        response="Error Registering";
    }
} catch (Exception e) {
    e.printStackTrace();
}

return response;
}

private String getPostDataString(HashMap<String, String> params) throws
UnsupportedEncodingException {
    StringBuilder result = new StringBuilder();
    boolean first = true;
    for(Map.Entry<String, String> entry : params.entrySet()){
        if (first)
            first = false;
        else
            result.append("&");

        result.append(URLEncoder.encode(entry.getKey(), "UTF-8"));
        result.append("=");
        result.append(URLEncoder.encode(entry.getValue(), "UTF-8"));
    }

    return result.toString();
}
}
}

```

Επίσης για πληροφορία του αναγνώστη όταν ζητάμε να κατεβάσουμε μια φωτογραφία τότε θα μας πετάξει NetwotkNullPointer αν δεν τα έχουμε σε AsyncTask όσο αφορά αυτήν είναι είδη σε αυτή σύμφωνα με τον παραπάνω κώδικα. Εδώ τα βήματα είναι τα εξής παίρνουμε τα δεδομένα String Url πληροφορίες εγγραφής δημιουργούμε την σύνδεση

```

HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setReadTimeout(15000);
conn.setConnectTimeout(15000);
conn.setRequestMethod("POST");
conn.setDoInput(true);
conn.setDoOutput(true);

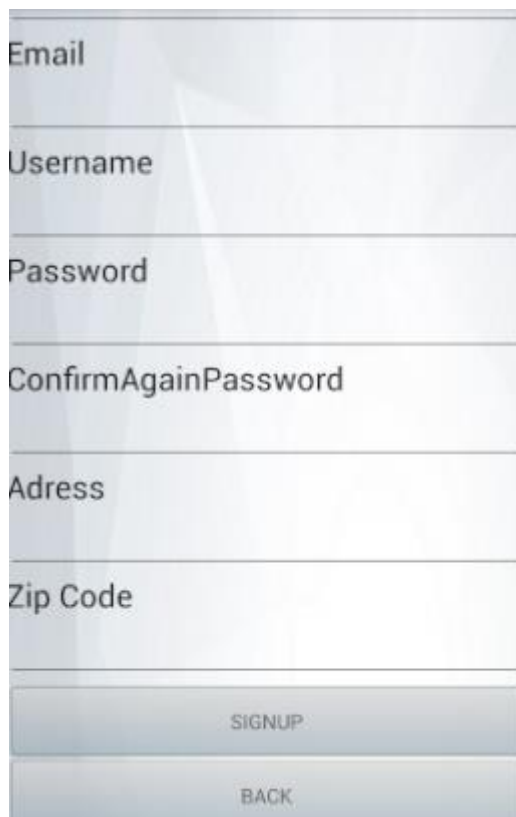
```

με διάφορα άλλα πράγματα όπως η λήξη της σύνδεσης να είναι 15 λεπτά η μέθοδος θα είναι post είσοδο και έξοδο στην συνέχεια κωδικοποιούμε τα δεδομένα μας σε UTF-8, χρειάζεται? Μπορεί να αν στην βάση έχουμε κάνει καμιά τέτοια επιλογή και αφού τελειώσουμε με την εγγραφή κλείνουμε τα στοιχεία που ανοίξαμε.

Στην συνέχεια μας έρχεται μια απάντηση ένα response αυτό γίνεται με το echo στο αρχείο php εδώ όμως θα γίνει

```
BufferedReader br=new BufferedReader(new  
InputStreamReader(conn.getInputStream()));  
response = br.readLine();
```

Έτσι θα πάρουμε την απάντηση αλλά ακόμα θέλει ένα βήμα. Στην μέθοδο `getPostDataString` θα δούμε ότι με την `for` βάζουμε τα δεδομένα σε `Key` και `value` ώστε να ξεχωρίζουμε. **Σημείωση το πρώτο `if (first)` το απορρίπτουμε διότι είναι το πεδίο `success`.**



The image shows a vertical registration form with the following fields and buttons:

- Email
- Username
- Password
- ConfirmAgainPassword
- Adress
- Zip Code
- SIGNUP
- BACK

Εικόνα μέσα από την εφαρμογή.

Layout welcome

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/images"
    android:orientation="vertical"
    android:weightSum="1">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome:"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/Tvu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=" "
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView12"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="WaitLoadData" />

    <TextView
        android:id="@+id/textView13"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="WaitLoadData" />

    <TextView
        android:id="@+id/textView14"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="WaitLoadData" />

    <TextView
        android:id="@+id/textView15"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="1.00" />
```

```
<Button
    android:layout_marginTop="3dp"
    android:layout_marginLeft="3dp"
    android:layout_marginRight="3dp"
    android:alpha="0.7"
    android:background="@drawable/button_rounded"
    android:id="@+id/goMap"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="OnButtonClick"
    android:text="GoMap" />
```

```
<Button
    android:layout_marginTop="3dp"
    android:layout_marginLeft="3dp"
    android:layout_marginRight="3dp"
    android:alpha="0.65"
    android:background="@drawable/button_rounded"
    android:id="@+id/goMeetings"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="OnButtonClick"
    android:text="GoMeetings" />
```

```
<Button
    android:layout_marginTop="3dp"
    android:layout_marginLeft="3dp"
    android:layout_marginRight="3dp"
    android:alpha="0.65"
    android:background="@drawable/button_rounded"
    android:id="@+id/reco"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="OnButtonClick"
    android:text="Reconnect" />
```

```
</LinearLayout>
```

Και αυτό το layout είναι παρόμοιο με τα άλλα δεν έχει καμιά ιδιαιτερότητα.

Class Welcome

```
package com.tolis.maps3856;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

public class WelcomeUser extends AppCompatActivity {

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        getSupportActionBar().hide();
        setContentView(R.layout.welcomeuser);
        String username = getIntent().getStringExtra("username");
        String email = getIntent().getStringExtra("email");
        String fname = getIntent().getStringExtra("fname");
        int id = Integer.parseInt(getIntent().getStringExtra("id"));
        TextView tv = (TextView) findViewById(R.id.Tvu);
        tv.setText("Mr." + fname);

        TextView etUsername = (TextView) findViewById(R.id.textView12);
        TextView etEmail = (TextView) findViewById(R.id.textView13);
        TextView etId = (TextView) findViewById(R.id.textView14);

        // Display user details
        String message = "Username:" + username;
        etUsername.setText(message);
        String message2 = "Email:" + email;
        etEmail.setText(message2);
        String message3 = "Id:" + id;
        etId.setText(message3);
    }

    public void OnButtonClick(View v) {
        if (v.getId() == R.id.goMeetings) {
            Intent intent = new Intent(WelcomeUser.this, CheckSeeMeetings.class);
            WelcomeUser.this.startActivity(intent);
            WelcomeUser.this.finish();
        }
        if (v.getId() == R.id.goMap) {
            Intent intent = new Intent(WelcomeUser.this, PlaceAndIntrests.class);
            WelcomeUser.this.startActivity(intent);
            WelcomeUser.this.finish();
        }
        if (v.getId() == R.id.reco) {
```



```

Intent intent = new Intent(WelcomeUser.this, Signin.class);
WelcomeUser.this.startActivity(intent);
WelcomeUser.this.finish();
    }
}
}

```

Στην παρούσα κλάση παίρνουμε τα στοιχεία από την Signin String username = getIntent().getStringExtra("username"); με αυτόν τον τρόπο και στην συνέχεια τα εμφανίζουμε στο χρήστη με την χρήση του Id ενός πεδίου με το set: TextView tv = (TextView) findViewById(R.id.Tvu); tv.setText("Mr." + fname);. Όπως βλέπετε και παραπάνω στην τελευταία μέθοδο για κάθε κουμπί το στέλνω και σε διαφορετικό activity αν έχετε παρατήρηση παραπάνω το flow diagram είναι το σημείο που καθορίζει αν θέλει να πάει πίσω για επανασύνδεση να πάει στο χάρτη η στα meeting.

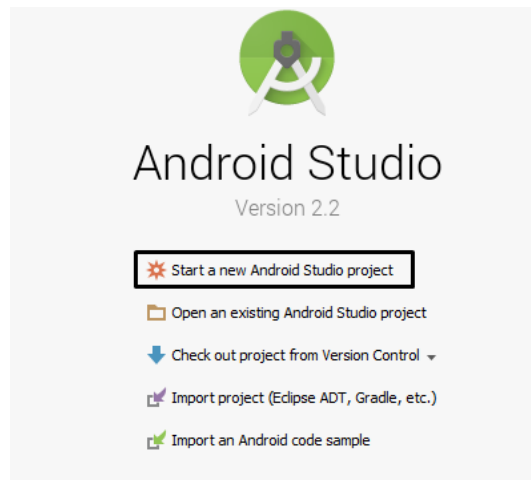


Επίσης το σημείο με το Maps3856 δεν εμφανίζετε καθόλου είναι η αρχική ρύθμιση που έκανα στο Intro με το hide και στα επόμενα activity το ίδιο. Πριν ξεκινήσω να περιγράψω την PlaceandIntrests θα αναφερθώ στο Navigation drawer και στο τρόπο με τον οποίο τον δημιούργησα, εφάρμοσα μέσα στην εφαρμογή.

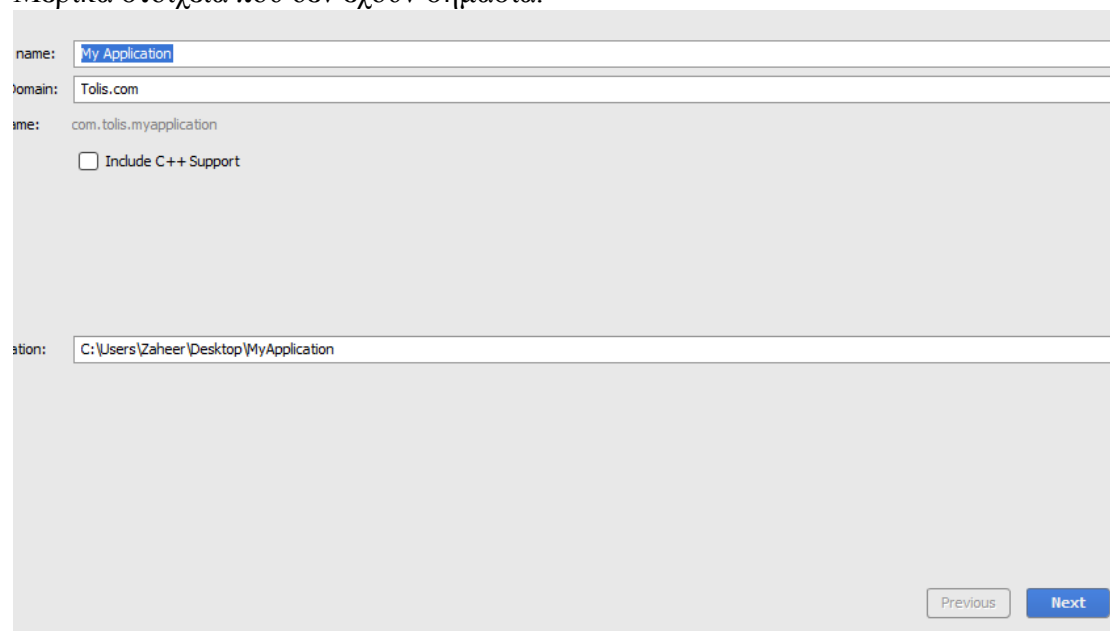
Η αναζήτηση μου για να δημιουργήσω το navigation drawer δεν ήταν και τόσο επιτυχής αλλά σε κάποιο σημείο με την ενασχόληση μου παρατήρησα ότι το android studio παρέχει είδη , το δύσκολο κομμάτι όμως ήταν πως θα το εφαρμόσω από άλλο project και θα μπορέσω να το ελέγγω μέσα από άλλες κλάσεις παρακάτω θα αναλύσω συγκεκριμένα τα βήματα πως δημιούργησα και έκανα εισαγωγή το έτοιμο navigation drawer που παρέχεται.

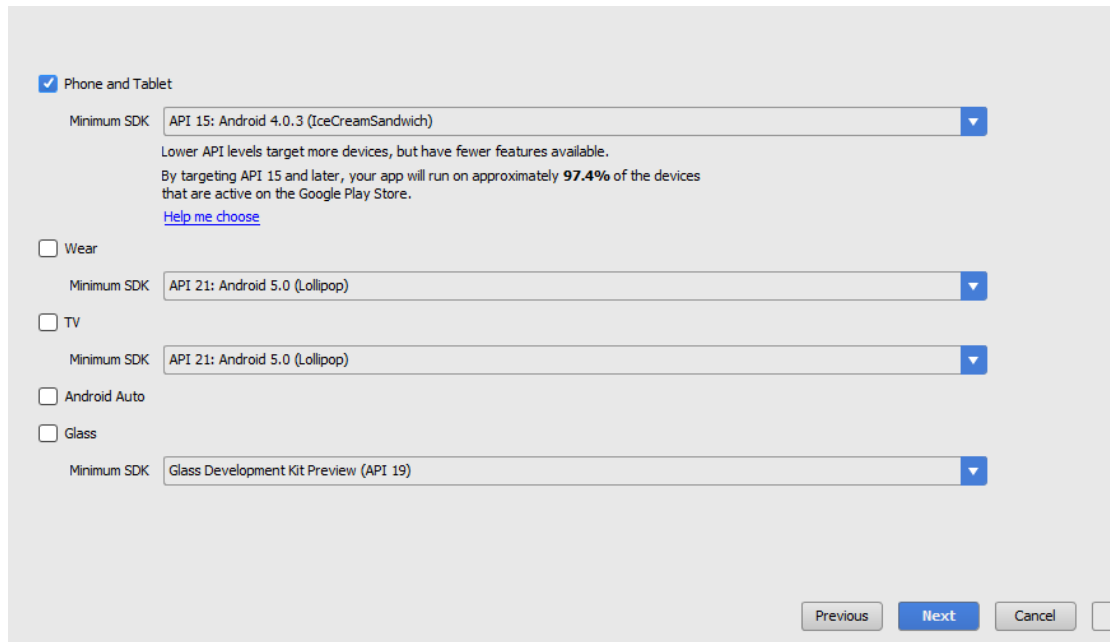
Τα βήματα είναι τα εξής:

Δημιούργησα ένα νέο project

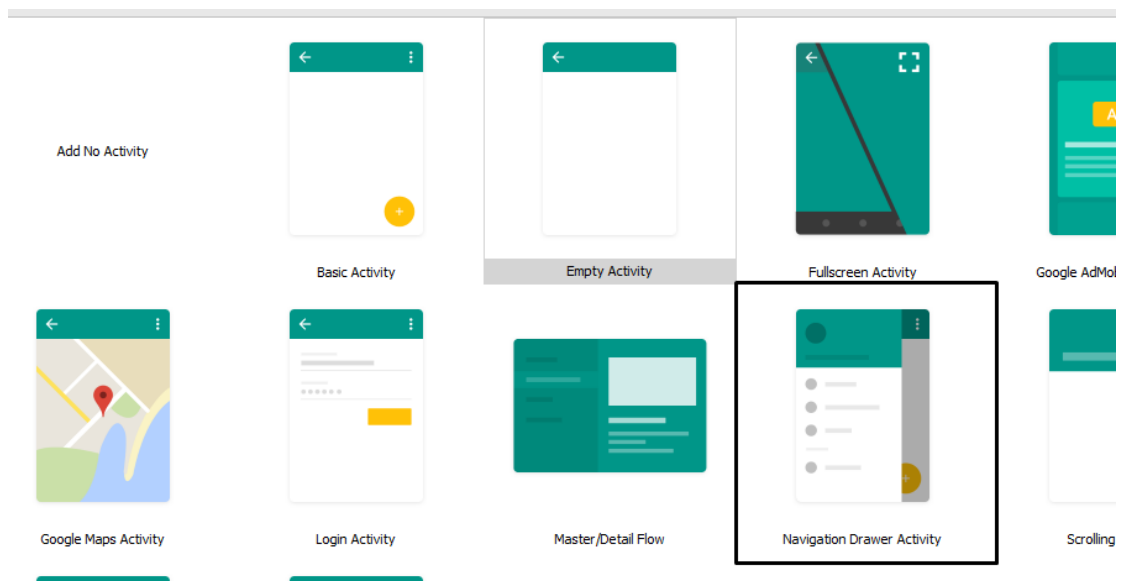


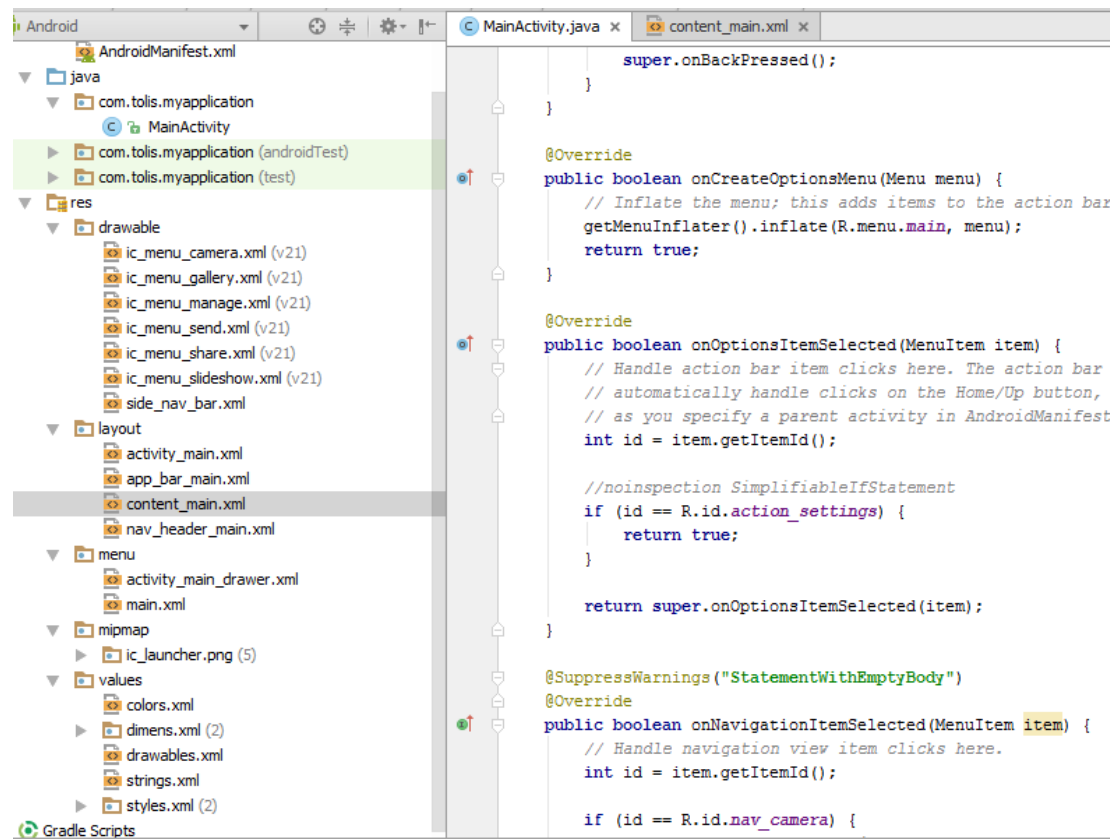
Μερικά στοιχεία που δεν έχουν σημασία.



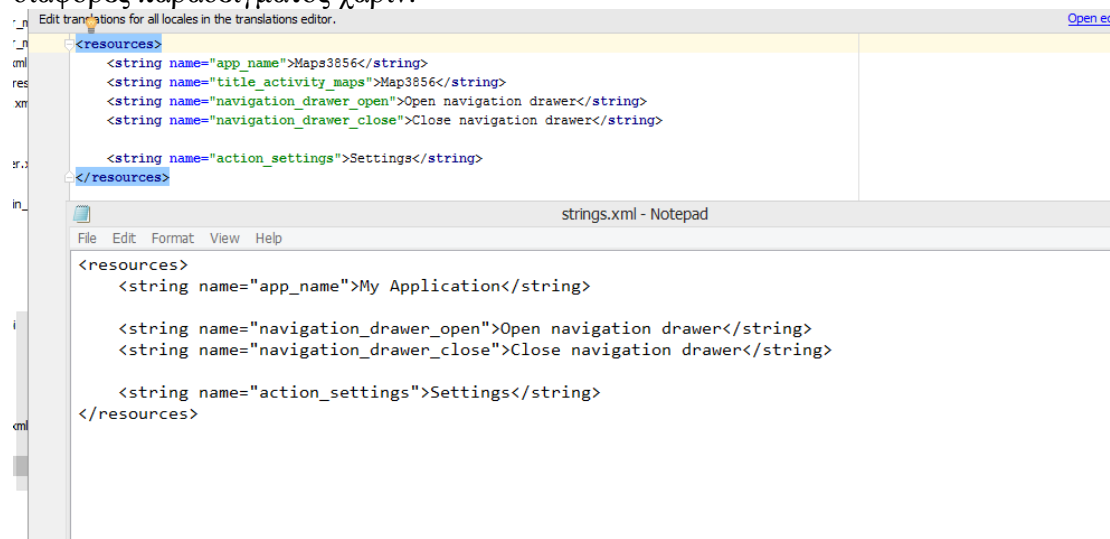


Και τα σημαντικό επέλεξα το Navigation drawer.





Η επόμενη κίνηση μου ήταν να πάρω ότι αρχείο χρειαζόμουν από εδώ και να το εφαρμόσω κατάλληλα στο δικό μου project. Οποτε πείρα ότι αρχείο δεν υπήρχε στο δικό μου και το έκανα copy paste με την δημιουργία των κατάλληλων φακέλων, βεβαία όμως κάποια αρχεία ειδή υπήρχαν τότε σημαίνει ότι είμαστε έτοιμοι; Όχι φυσικά τα αρχεία που υπάρχουν στις δύο μεριές είναι διαφορετικά διότι αποσκοπούν σε διαφορετικές ρυθμίσεις αυτό που έκανα πήγα στο project και συμπλήρωσα τις διαφορές παραδείγματος χάριν.

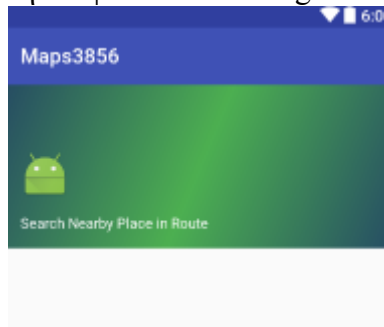


Στο πάνω μέρος είναι ο κώδικας από το από την πτυχιακή και στο κάτω παρουσιάζεται ο κώδικας από το νέο Project που μόλις έδειξα. Οπως βλέπουμε θέλει μια προσαρμογή/προσθήκη των διαφορετικών tags. Πριν όμως τελειώσουμε με όλα αυτά περισεύει και ένα άχρηστο αρχείο στην παρούσα πτυχιακή που είναι αχρειαστο

και είναι το `app_bar_main.xml` αυτό θα παραληφτεί και ότι αναφέρεται σε αυτό καθώς είναι μια μπάρα στο κάτω μέρος της οθόνης ενεργοποιήσαμε με ένα κουμπί. Για να ξεκαθαριστεί πως θα το εφαρμόσουμε και πως θα εμφανίζεται θα δείξω πρώτα το layout των drawers με τους κώδικες τους και στην συνέχεια θα παρουσιάσω πως αυτό μπορεί να καλέσει να γίνει include σε ένα ξένο Layout που αυτό θα το χειρίζεται στην προκείμενη περίπτωση θα είναι το `placeandintrests` layout.

Layout navheader_main1

Είναι το Layout που αφορά την κεφαλίδα του navigation drawer



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="@dimen/nav_header_height"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/aaa"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@android:drawable/sym_def_app_icon" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="Search Nearby Place in Route "
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

</LinearLayout>
```

Σε αυτό το layout έχουμε αρκετή πληροφορία πρώτον

```
android:layout_height="@dimen/nav_header_height"
```

Αυτό σημαίνει ότι θα πάει στο αρχείο dimen και θα δει το πεδίο nav_header ώστε να πάρει τις διαστάσεις

```
android:gravity="bottom"
```

Αυτό σημαίνει ότι θα σύρει γενικά το content τα στοιχεία λίγο πιο κάτω πχ. Σαν να κατεβάζουμε ένα κάδρο στο σπίτι μας 10 εκατοστά κάτω

```
android:paddingBottom="@dimen/activity_vertical_margin"
```

```
android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
android:paddingRight="@dimen/activity_horizontal_margin"
```

```
android:paddingTop="@dimen/activity_vertical_margin"
```

Αυτό σημαίνει ότι θα πάει στο αρχείο dimen και θα δει το πεδίο activity_header να πάρει τις διαστάσεις πχ 3dp

```
android:theme="@style/ThemeOverlay.AppCompat.Dark">
```

Αυτό σημαίνει ότι θα πάει στο αρχείο style στο ThemeOverlay για να πάρει την πληροφορία

```
<ImageView
```

```
android:id="@+id/aaa"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:src="@android:drawable/sym_def_app_icon" />
```

Αφορά την εικόνα τις διαστάσεις τις και το src η πηγή της το source.

```
<TextView>
```

Και το text view το έχουμε ξαναδεί.

Layout activity_main_drawer

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<group android:checkableBehavior="single">
```

```
<item
```

```
android:id="@+id/nav_gas_station"
```

```
android:icon="@drawable/autogasl"
```

```
android:title="Gas Station" />
```

```
<item
```

```
android:id="@+id/nav_shop_grocery"
```

```
android:icon="@drawable/shop"
```

```
android:title="Grocery Or Supermarket" />
```

```
<item
```

```
android:id="@+id/nav_movie_theater"
```

```
android:icon="@drawable/cinemas"
```

```
android:title="Movie Theater" />
```

```
<item
```

```
android:id="@+id/nav_search"
```

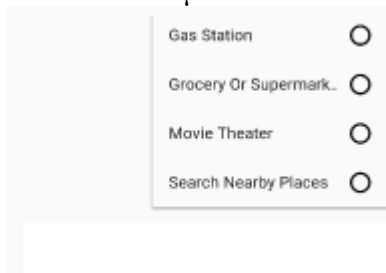
```
android:icon="@drawable/search"
```

```
android:title="Search Nearby Places" />
```

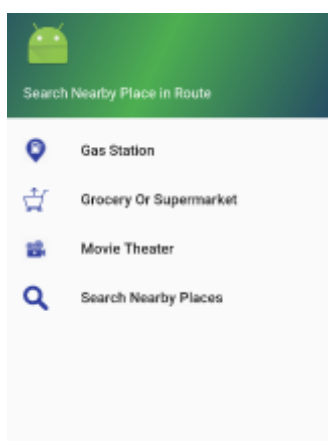
```
</group>
```

```
</menu>
```

Αυτό το layout είναι το main body του navigation drawer τα items είναι τα αντικείμενα όπου έχουν εικόνα id και κείμενο.



Στο σύνολο όταν το καλέσουμε θα μοιάζει έτσι.



Class NavigationLeft

```
package NavigationDrawers;
import android.app.Activity;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.util.Log;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.Toast;
import com.tolis.maps3856.R;
import java.util.ArrayList;
import SetRoute.SetRoute;
import static FindPlaces.VariablesFindPlaces.Duration;
import static FindPlaces.VariablesFindPlaces.Places_LongLatitubee;
import static FindPlaces.VariablesFindPlaces.Places_Vicinity;
import static FindPlaces.VariablesFindPlaces.if_fine_then_search;
import static SetMeetings.SetMeetingsVariables.PlacesName;
import static SetMeetings.SetMeetingsVariables.typeofmark;
import static SetRoute.SetRouteVariables.Navigation_ok;
import static com.tolis.maps3856.PlaceAndIntrests.flagondemand;
import static com.tolis.maps3856.PlaceAndIntrests.setRoute_clear_button_state;
```

```
public class NavigationLeft implements
```

```

NavigationView.OnNavigationItemSelectedListener{
    private int first_time = 0;
    private Activity activity;
    private ArrayList<String> placename;
    private ArrayList<String> vicinity;
    private ArrayList<String> pl;
    private ArrayList<Integer> duration;
    private ArrayList<Integer> typeofmark_navliste;
    private int start_point_oftype_1, start_point_oftype_2, start_point_oftype_3,
end_point_oftype_1, end_point_oftype_2, end_point_oftype_3;
    private int be_now_oftype_1, be_now_oftype_2, be_now_oftype_3;
    private int created_matrix = 0;

    public void setActivityUi(Activity activity) {
        this.activity = activity;
    }

    public void NaviListe() {
        NavigationView navigationView = (NavigationView)
activity.findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);

    }
}

```

```

@SuppressWarnings("StatementWithEmptyBody")

```

```

public boolean onNavigationItemSelectedListener(MenuItem item) {
    Log.d("matrixSize", String.valueOf(PlacesName.size()));
    Log.d("matrixSize", String.valueOf(Places_Vicinity.size()));
    Log.d("matrixSize", String.valueOf(Places_LongLatitubee.size()));
    Log.d("matrixSize", String.valueOf(Duration.size()));
    Log.d("matrixSize", String.valueOf(typeofmark.size()));

    if (flagondemand == 1) {
        Toast.makeText(activity, "Available Only on Man Mode",
Toast.LENGTH_SHORT).show();
        return true;
    }
    if (if_fine_then_search == 1) {

        // Handle navigation view item clicks here.
        int id = item.getItemId();
        if (Navigation_ok == 1) {
            if (created_matrix == 0)

```



```

        try {
            create_matrix();
        } catch (IndexOutOfBoundsException e) {
            Toast.makeText(activity, "Refresh and Search Again Pls",
Toast.LENGTH_LONG).show();
            Navigation_ok=0;
            if_fine_then_search=0;
            return true;
        }
        created_matrix = 1;
    }

    if (id == R.id.nav_gas_station) { //gas station
        if (Navigation_ok == 0) {
            Toast.makeText(activity, "First Hit button bellow Search",
Toast.LENGTH_LONG).show();
            return true;
        }
        if (start_point_oftype_1 == -1) {
            Toast.makeText(activity, "Sorry Not Exist In map",
Toast.LENGTH_LONG).show();
            return true;
        }
        SetRoute setroute = new SetRoute();
        setroute.setActivityUi(activity);
        setroute.setRoutefromNaviPanel(pl.get(be_now_oftype_1),
placename.get(be_now_oftype_1), vicinity.get(be_now_oftype_1));
        be_now_oftype_1 += 1;
        if (be_now_oftype_1 > end_point_oftype_1)
            be_now_oftype_1 = start_point_oftype_1;
    } else if (id == R.id.nav_shop_grocery) { //shop grocery
        if (Navigation_ok == 0) {
            Toast.makeText(activity, "First Hit button bellow Search",
Toast.LENGTH_LONG).show();
            return true;
        }
        if (start_point_oftype_3 == -1) {
            Toast.makeText(activity, "Sorry Not Exist In map",
Toast.LENGTH_LONG).show();
            return true;
        }
        SetRoute setroute = new SetRoute();
        setroute.setActivityUi(activity);
        setroute.setRoutefromNaviPanel(pl.get(be_now_oftype_3),
placename.get(be_now_oftype_3), vicinity.get(be_now_oftype_3));
        be_now_oftype_3 += 1;
        if (be_now_oftype_3 > end_point_oftype_3)
            be_now_oftype_3 = start_point_oftype_3;
    } else if (id == R.id.nav_movie_theater) { //movie theater

```

```

        if (Navigation_ok == 0) {
            Toast.makeText(activity, "First Hit button bellow Search",
Toast.LENGTH_LONG).show();
            return true;
        }
        if (start_point_oftype_2 == -1) {
            Toast.makeText(activity, "Sorry Not Exist In map",
Toast.LENGTH_LONG).show();
            return true;
        }
        SetRoute setroute = new SetRoute();
        setroute.setActivityUi(activity);
        setroute.setRoutefromNaviPanel(pl.get(be_now_oftype_2),
placename.get(be_now_oftype_2), vicinity.get(be_now_oftype_2));
        be_now_oftype_2 += 1;
        if (be_now_oftype_2 > end_point_oftype_2)
            be_now_oftype_2 = start_point_oftype_2;
    } else if (id == R.id.nav_search) {

        final Button button = (Button) activity.findViewById(R.id.setroute);
        activity.runOnUiThread(new Runnable() {
            public void run() {
                button.setText("SetRoute");
                setRoute_clear_button_state = 0;

            }

        });

        if (first_time != 0) {
            first_time = 0;
            Duration.removeAll(Duration);
            Navigation_ok = 0;
            created_matrix = 0;
        } else
            first_time = 1;
        if (PlacesName.size() != Duration.size())
            Toast.makeText(activity, "Wait Load Data",
Toast.LENGTH_LONG).show();

        final SetRoute sr = new SetRoute();
        sr.setActivityUi(activity);
        sr.findroute(0);
        first_time = 1;
    }

    DrawerLayout drawer = (DrawerLayout)
activity.findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);

```

```

        return true;
    } else {
        Toast.makeText(activity, "Sorry You Have To Wait All Load ALL Data",
Toast.LENGTH_LONG).show();
        return true;
    }
}

```

```

private void create_matrix() throws IndexOutOfBoundsException {
    //no clone no add all
    placename = new ArrayList<String>(PlacesName.size());
    vicinity = new ArrayList<String>(Places_Vicinity.size());
    pl = new ArrayList<String>(Places_LongLatitubee.size());
    duration = new ArrayList<Integer>(Duration.size());
    typeofmark_navliste = new ArrayList<Integer>(typeofmark.size());

    for(String temp : PlacesName) {
        placename.add(new String(temp));
    }
    for(String temp : Places_Vicinity) {
        vicinity.add(new String(temp));
    }
    for(String temp : Places_LongLatitubee) {
        pl.add(new String(temp));
    }
    for(int temp : Duration) {
        duration.add(new Integer(temp));
    }
    for(int temp : typeofmark) {
        typeofmark_navliste.add(new Integer(temp));
    }

    sort_by_type();
    find_typeofplaces_start_end_in_matrix();
    sort_by_duration();

    be_now_oftype_1 = start_point_oftype_1;
    be_now_oftype_2 = start_point_oftype_2;
    be_now_oftype_3 = start_point_oftype_3;
}

```

```

private void sort_by_duration() {
    int need_sort = 1;
    Log.d("matrix", String.valueOf(duration));
    if (start_point_oftype_1 == end_point_oftype_1) need_sort = 0;

    while (need_sort == 1) {
        need_sort = 0;
        for (int i = start_point_oftype_1; end_point_oftype_1 > i; i++)

```

```

if (duration.get(i) > duration.get(i + 1)) {

    String temp2 = placename.get(i + 1);
    placename.set(i + 1, placename.get(i));
    placename.set(i, temp2);

    String temp3 = vicinity.get(i + 1);
    vicinity.set(i + 1, vicinity.get(i));
    vicinity.set(i, temp3);

    String temp4 = pl.get(i + 1);
    pl.set(i + 1, pl.get(i));
    pl.set(i, temp4);

    int temp5 = duration.get(i + 1);
    duration.set(i + 1, duration.get(i));
    duration.set(i, temp5);

    need_sort = 1;
}

}

need_sort = 1;
if (start_point_oftype_2 == end_point_oftype_2) need_sort = 0;

while (need_sort == 1) {
    need_sort = 0;
    for (int i = start_point_oftype_2; end_point_oftype_2 > i; i++)
        if (duration.get(i) > duration.get(i + 1)) {

            String temp2 = placename.get(i + 1);
            placename.set(i + 1, placename.get(i));
            placename.set(i, temp2);

            String temp3 = vicinity.get(i + 1);
            vicinity.set(i + 1, vicinity.get(i));
            vicinity.set(i, temp3);

            String temp4 = pl.get(i + 1);
            pl.set(i + 1, pl.get(i));
            pl.set(i, temp4);

            int temp5 = duration.get(i + 1);
            duration.set(i + 1, duration.get(i));
            duration.set(i, temp5);
        }
    }
}

```

```

        need_sort = 1;
    }

}
need_sort = 1;
if (start_point_oftype_3 == end_point_oftype_3) need_sort = 0;

while (need_sort == 1) {
    need_sort = 0;
    for (int i = start_point_oftype_3; end_point_oftype_3 > i; i++)
        if (duration.get(i) > duration.get(i + 1)) {

            String temp2 = placename.get(i + 1);
            placename.set(i + 1, placename.get(i));
            placename.set(i, temp2);

            String temp3 = vicinity.get(i + 1);
            vicinity.set(i + 1, vicinity.get(i));
            vicinity.set(i, temp3);

            String temp4 = pl.get(i + 1);
            pl.set(i + 1, pl.get(i));
            pl.set(i, temp4);

            int temp5 = duration.get(i + 1);
            duration.set(i + 1, duration.get(i));
            duration.set(i, temp5);

            need_sort = 1;
        }

    }
    Log.d("matrix", String.valueOf(duration));
}

private void find_typeofplaces_start_end_in_matrix() {
    start_point_oftype_1 = -1;
    start_point_oftype_2 = -1;
    start_point_oftype_3 = -1;
    end_point_oftype_1 = -1;
    end_point_oftype_2 = -1;
    end_point_oftype_3 = -1;

    int first_time_met_type1 = 0;
    int first_time_met_type2 = 0;
    int first_time_met_type3 = 0;
}

```

```

for (int i = 0; typeofmark_navliste.size() > i; i++) {
    if (typeofmark_navliste.get(i) == 1) {
        if (first_time_met_type1 == 0)
            start_point_oftype_1 = i;
        end_point_oftype_1 = i;
        first_time_met_type1 = 1;
    }
    if (typeofmark_navliste.get(i) == 2) {
        if (first_time_met_type2 == 0)
            start_point_oftype_2 = i;
        end_point_oftype_2 = i;
        first_time_met_type2 = 1;
    }
    if (typeofmark_navliste.get(i) == 3) {
        if (first_time_met_type3 == 0)
            start_point_oftype_3 = i;
        end_point_oftype_3 = i;
        first_time_met_type3 = 1;
    }
}

}

private void sort_by_type() throws IndexOutOfBoundsException {

    int need_sort = 1;
    Log.d("matrix", String.valueOf(typeofmark_navliste));

    while (need_sort == 1) {
        need_sort = 0;
        for (int i = 0; typeofmark_navliste.size() - 1 > i; i++)
            if (typeofmark_navliste.get(i) > typeofmark_navliste.get(i + 1)) {

                int temp = typeofmark_navliste.get(i + 1);
                typeofmark_navliste.set(i + 1, typeofmark_navliste.get(i));
                typeofmark_navliste.set(i, temp);
                need_sort = 1;

                String temp2 = placename.get(i + 1);
                placename.set(i + 1, placename.get(i));
                placename.set(i, temp2);

                String temp3 = vicinity.get(i + 1);
                vicinity.set(i + 1, vicinity.get(i));
                vicinity.set(i, temp3);
            }
    }
}

```

```

String temp4 = pl.get(i + 1);
pl.set(i + 1, pl.get(i));
pl.set(i, temp4);

int temp5 = duration.get(i + 1);
duration.set(i + 1, duration.get(i));
duration.set(i, temp5);
}
}
Log.d("matrix", String.valueOf(typeofmark_navliste));
}
}
}

```

Αυτήν την κλάση την δημιουργούμε στην PlaceandIntrests που είναι σαν την main του προγράμματος η οποία δημιουργείται μια φορά και ελέγχει το left navigation drawer. Αφού δημιουργηθεί καλώ την μέθοδο

```

public void setActivityUi(Activity activity) {
this.activity = activity;}

```

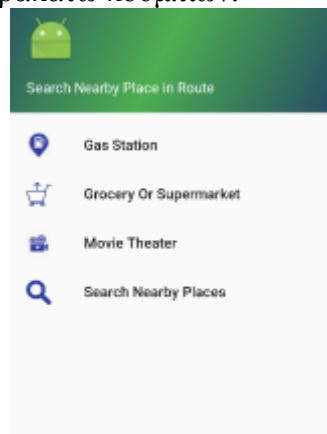
Στην οποία θα τις περάσω το activity από την PlaceandIntrests. Αυτό το κάνω για να μπορώ να ελέγγω το ξένο layout της PlaceandIntrests button ή ακόμα να μπορώ να εμφανίζω μηνύματα του τύπου Toast (“alert messages”). Αν προσέξατε καλά στην κλάση αυτή έχει ένα implement το

```

NavigationView.OnNavigationItemSelectedListener με το οποίο θα προσθέσω τις
μεθόδους που χρειάζονται για τον Navigation drawer καθώς και τον Listener
public void NaviListe() {
NavigationView navigationView = (NavigationView)
activity.findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);}

```

Όστε να μπορεί να υπάρχει ένας ελεγκτής αν υπάρχει πάτημα κουμπιού από τον χρήστη στο πάτημα έκτων παρακάτω κουμπιών.



Τα Log.d είναι βοηθητικά μηνύματα που παρουσιάζονται στο android studio και σε βοηθάμε στο να δεις κάποιες τιμές, είναι παρά πολύ σημαντικά καθώς χωρίς αυτά δεν θα είχε επιτευχτεί αυτή η πτυχιακή διότι αποτελούν μια μορφή debugging. Στην πτυχιακή αυτή η ανανέωση σημείων γίνεται με δυο τρόπους αυτόματα ή χειροκίνητα για αυτό σκοπό δημιούργησα την μεταβλητή `flagondemand` που όταν έχει την τιμή 1 είναι αυτόματα και 2 χειροκίνητα.

```

if (flagondemand == 1) {
    Toast.makeText(activity, "Available Only on Man Mode",
    Toast.LENGTH_SHORT).show();
    return true;
}

```

Η λειτουργία παρακάτω είναι διαθέσιμη μόνο όταν η εφαρμογή είναι σε χειροκίνητη μορφή ,γιατί; Διότι όταν είναι αυτόματα έχουμε πολλές αλλαγές και δημιουργούνται αρκετά προβλήματα. Πριν συνεχίσω να περιγράψω πως λειτουργούν τα 3 πρώτα κουμπιά θα πρέπει να ξεκαθαρίσω ότι για να παίξουν πρέπει να πατηθεί το τελευταίο το search ώστε ο αλγόριθμος να κατατάξει τα σημεία βάση αποστάσεων. Τι γίνετε λοιπόν:

```

else if (id == R.id.nav_search) {

    final Button button = (Button) activity.findViewById(R.id.setroute);
    activity.runOnUiThread(new Runnable() {
        public void run() {
            button.setText("SetRoute");
            setRoute_clear_button_state = 0;

        }

    });

    if (first_time != 0) {
        first_time = 0;
        Duration.removeAll(Duration);
        Navigation_ok = 0;
        created_matrix = 0;
    } else
        first_time = 1;
    if (PlacesName.size() != Duration.size())
        Toast.makeText(activity, "Wait Load Data", Toast.LENGTH_LONG).show();

    final SetRoute sr = new SetRoute();
    sr.setActivityUi(activity);
    sr.findroute(0);
    first_time = 1;
}

```

Βρίσκουμε καταρχάς το κουμπί Set Route και το επαναφέρουμε στην αρχική του μορφή διαγράφοντας αν υπάρχει κάποια δρομολόγηση σχεδιασμένη στο χάρτη και αντί Clear σε Set Route (το κείμενο του κουμπιού) επιπλέον και σε αυτό το κουμπί έχω θέσει μια μεταβλητή για να ελέγγω την κατάσταση του την οποία αλλάζω . Μετά ακόλουθη κάτι παράξενο

```

activity.runOnUiThread(new Runnable() { public void run() {} });

```

αυτό σημαίνει ότι για να αλλάξω κάτι σε ένα Layout από μια δεύτερη κλάση θα πρέπει να τρέχω και στο main Thread είναι παρά πολύ σημαντικό και θα το συναντήσουμε συχνά. Άρα για να τρέξω `button.setText("SetRoute");` αυτό πρέπει

να είναι μέσα σε `activity.runOnUiThread`.

```
if (first_time != 0) {
    first_time = 0;
    Duration.removeAll(Duration);
    Navigation_ok = 0;
    created_matrix = 0;
} else
    first_time = 1;
if (PlacesName.size() != Duration.size())
    Toast.makeText(activity, "Wait Load Data", Toast.LENGTH_LONG).show();
final SetRoute sr = new SetRoute();
sr.setActivityUi(activity);
sr.findroute(0);
first_time = 1;
```

Με αυτό το κομμάτι κώδικα εννοώ τα έξης αν μπεις την πρώτη φορά θα πας στο else όπου θα κανείς την μεταβλητή `first_time = 1`; . Μετά θα εμφανίσει το μήνυμα ότι φορτώνεις δεδομένα , αυτό το κάνει πάντα όσες φορές και αν το πατήσουμε. Αυτό που ελέγχεται με την μεταβλητή ουσιαστικά είναι το καθάρισμα του πίνακα `Duration` των αποστάσεων του κάθε σημείου , της μεταβλητής έλεγχου ότι πατήθηκε το κουμπί `created_matrix` και ότι φορτωθήκαν τα δεδομένα από την ξένη κλάση `Navigation_ok = 0`; . Δηλαδή με λίγα λόγια πατάς το κουμπί Search και ελέγχεις από μεταβλητή αν το έχουμε ξαναπατήσει, αν ναι τότε σημαίνει ότι έχει ειδή πατηθεί μια φορά αρά υπάρχουν δεδομένα στο πίνακα Duration που πρέπει να διαγραφτούν διότι κάθε φορά μπορεί να έχεις αλλά σημεία περισσότερα λιγότερα με πόλους διαφορετικούς χρόνους .

```
Επίσης αφού θέλουμε να κάνουμε έξοδο
DrawerLayout drawer = (DrawerLayout)
activity.findViewById(R.id.drawer_layout);
drawer.closeDrawer(GravityCompat.START);
return true;
Προσοχή όχι ποτέ false
```

```
if (flagondemand == 1) {
    Toast.makeText(activity, "Available Only on Man Mode",
    Toast.LENGTH_SHORT).show();
    return true;
}
Είναι διαθέσιμο μόνο όταν είναι χειροκίνητο.
```

```
if (if_fine_then_search == 1) {

    // Handle navigation view item clicks here.
    int id = item.getItemId();
    if (Navigation_ok == 1) {
        if (created_matrix == 0)
            try {
                create_matrix();
            }
    }
}
```

```

    } catch (IndexOutOfBoundsException e) {
        Toast.makeText(activity, "Refresh and Search Again Pls",
Toast.LENGTH_LONG).show();
        Navigation_ok=0;
        if_fine_then_search=0;
        return true;
    }
    created_matrix = 1;
}

```

if_fine_then_search == 1 ελέγχουμε αν έχουν κατεβεί όλα τα δεδομένα από το google api όσον αφορά τα σημεία ενδιαφέροντος αλλιώς πρέπει να περιμένουμε να φορτωθούν όλα

Navigation_ok ελέγχουμε αν έχουμε φορτώσει όλες τις αποστάσεις σημείων ...είναι διαφορετικό google query σε σχέση με το πάνω

created_matrix ελέγχει αν έχουμε τρέξει τον αλγόριθμο διότι τρέχει αφού πατηθεί μια φορά το search και μετά κάποιο κουμπί .Ακόλουθη ο αλγόριθμος και στην συνέχεια η διενέργεια που γίνεται όταν πατηθεί κάθε κουμπί.

Ο αλγόριθμος λοιπόν ξεκινά καλώντας την μέθοδο **private void** create_matrix() η οποία μπορεί να κάνει **throw** IndexOutOfBoundsException σε περίπτωση λάθους σε πίνακα. **Προς το παρόν θα αγνοήσω την προέλευση των έξυπνων πινάκων αλλά θα εξηγηθεί μετά (PlacesName...)**. Έχουμε λοιπόν περάσει μέσα στην κλάση 5 πίνακες 1 το όνομα του σημείου ενδιαφέροντος , την περιοχή , τις συντεταγμένες , την διάρκεια που κάνουμε για να πάμε στο σημείο και τι είδος είναι. Σε αυτήν την πτυχιακή θεώρησα για λόγους ευκολίας να αναφερθώ σε αριθμούς αντί το όνομα του είδους το οποίο είναι 0 κάτι διαφορετικό 1 Gas station , 2 Movie Theater 3 Grocery Or Supermarket. Πως όμως θα περάσω στην νέα λίστα πινάκων τα στοιχεία χωρίς να επηρεάσω τις προηγούμενες; Όπως φαίνεται στον παρακάτω κώδικα ο αποτελεσματικός τρόπος για να δημιουργήσω μια λίστα παράλληλα με μια άλλη. Πρώτα δημιουργώ σύμφωνα με το μέγεθος τις άλλης *PlacesName.size()*; μετέπειτα με μια for όπου θα περάσω τις τιμές στην νέα λίστα πίνακα , **Προσοχή η χρήση ετοιμών μεθόδων clone etc. θα δημιουργήσει πρόβλημα καθώς οι λίστες μεταξύ τους αναφέρονται στην ίδια περιοχή της μνήμης.** Αυτό δεν σημαίνει ότι είναι σε αύξουσα τα σημεία, αυτό που θα γίνει είναι να μπουν σε μια σειρά.

```

    placename = new ArrayList<String>(PlacesName.size());
    .....

    for(String temp : PlacesName) {
        placename.add(new String(temp));
    }
    .....

```

Στην συνέχεια ταξινομήσα τους πέντε πίνακες σύμφωνα με το type τους πχ.

type	duration	Name	Vicinity	LongLat
1	12	Example1	Ag. Ioannou	34.22,23.89
3	11	Example2	Alikarnassos	34.21,23.86
2	8	Example3	Estauromenos	34.24,23.85
2	22	Example4	Limani	34.27,23.83

Σε:

type	duration	Name	Vicinity	LongLat
1	12	Example1	Ag. Ioannou	34.22,23.89
2	8	Example3	Estauromenos	34.24,23.85
2	22	Example4	Limani	34.27,23.83
3	11	Example2	Alikarnassos	34.21,23.86

Αυτό το κάνω με σκοπό να μην δημιουργήσω πίνακες για κάθε είδος παρά έναν ταξινομημένο κατάλληλα με μεταβλητές όπου θα διαχωρίζεται , δηλαδή το καθένα θα έχει μεταβλητή αρχή και τέλους μέσα στο πίνακα σαν είδος (pointers start/end) και το σημείο στο οποίο βρίσκεται (pointer of location).

```

int need_sort = 1;
while (need_sort == 1) {
    need_sort = 0;
    for (int i = 0; typeofmark_navliste.size() - 1 > i; i++)
        if (typeofmark_navliste.get(i) > typeofmark_navliste.get(i + 1)) {

            int temp = typeofmark_navliste.get(i + 1);
            typeofmark_navliste.set(i + 1, typeofmark_navliste.get(i));
            typeofmark_navliste.set(i, temp);
            need_sort = 1;

            String temp2 = placename.get(i + 1);
            placename.set(i + 1, placename.get(i));
            placename.set(i, temp2);
        }
}

```

Αυτός είναι ο κλασικός αλγόριθμος ταξινόμησης όπως ξέρουμε στον οποίο προτίμησα **while** και χρήση μεταβλητής ώστε να ξέρω την ακριβή ταξινόμηση και να μην εκτελώ περιττά εντολές , αν παρατηρήσουμε μέσα στην **for** , μέσα στην **if** αν εκτελεστεί κάποια αλλαγή σημαίνει ότι χρειάζεται ταξινόμηση . Παρομοίως λοιπόν θα συμπληρώσουμε κατάλληλα και τους νέους πίνακες λίστες ώστε να τους ταξινομήσουμε όπως παραδόθηκε στον παραπάνω πλήρες κώδικα.

Παρακάτω με την μέθοδο `find_typeofplaces_start_end_in_matrix()`

```
start_point_oftype_1 = -1;
start_point_oftype_2 = -1;
start_point_oftype_3 = -1;
end_point_oftype_1 = -1;
end_point_oftype_2 = -1;
end_point_oftype_3 = -1;

int first_time_met_type1 = 0;
int first_time_met_type2 = 0;
int first_time_met_type3 = 0;

for (int i = 0; typeofmark_navliste.size() > i; i++) {
    if (typeofmark_navliste.get(i) == 1) {
        if (first_time_met_type1 == 0)
            start_point_oftype_1 = i;
        end_point_oftype_1 = i;
        first_time_met_type1 = 1;
    }
    if (typeofmark_navliste.get(i) == 2) {
        ...
    }
    if (typeofmark_navliste.get(i) == 3) {
        ...
    }
}
```

Έκανα τα έξης : όσο ήταν το μέγεθος του πίνακα/τα σημεία ενδιαφέροντος τοποθέτησα στο καθένα που αρχίζει το κάθε είδος (θέση πίνακα) με μεταβλητές και έθεσα στην αρχή αρνητικές μεταβλητές σε περίπτωση που δεν υπάρχει κάποιο είδος .Παραδείγματος χάριν βλέπε την παρακάτω εικόνα.

	type	duration	Name	Vicinity	LongLat
start_point_oftype_1/end_point_oftype_1	1	12	Example1	Ag. Ioannou	34.22,23.89
start_point_oftype_2	2	8	Example3	Estauomenos	34.24,23.85
end_point_oftype_2	2	22	Example4	Limani	34.27,23.83
start_point_oftype_3/end_point_oftype_3	3	11	Example2	Alikarnassos	34.21,23.86

Και τελικά στην `sort_by_duration` κάνω ταξινόμηση των πινάκων λιστών με την προϋπόθεση της διάρκειας που απέχει κάποιος από το σημείο.

```

private void sort_by_duration() {
    int need_sort = 1;

    if (start_point_oftype_1 == end_point_oftype_1) need_sort = 0;

    while (need_sort == 1) {
        need_sort = 0;
        for (int i = start_point_oftype_1; end_point_oftype_1 > i; i++)
            if (duration.get(i) > duration.get(i + 1)) {

                String temp2 = placename.get(i + 1);
                placename.set(i + 1, placename.get(i));
                placename.set(i, temp2);

                String temp3 = vicinity.get(i + 1);
                vicinity.set(i + 1, vicinity.get(i));
                vicinity.set(i, temp3);

                String temp4 = pl.get(i + 1);
                pl.set(i + 1, pl.get(i));
                pl.set(i, temp4);

                int temp5 = duration.get(i + 1);
                duration.set(i + 1, duration.get(i));
                duration.set(i, temp5);

                need_sort = 1;
            }
        }
    }
}

```

Και αυτός ο κώδικας είναι ίδιος με παραπάνω αλλά αξίζει να προσέξουμε ότι δεν ταξινομούμε όλα τα σημεία αλλά από εκεί που αρχίζει μέχρι εκεί που τελειώνει κάθε είδος δηλαδή τα duration 2 και 5 θα ταξινομηθούν σύμφωνα με το type 1 καθώς ανήκουν εκεί, όπου το 5 και το 2 θα αλλάξουν θέση.

	type	duration
for (int i = <u>start_point_oftype_1</u> ; <u>end_point_oftype_1</u> > i; i++)	1	5
	1	2
	2	12
	2	9
	2	3

Το όνομα , η περιοχή και οι συντεταγμένες ταξινομούνται παράλληλα. Επίσης πριν

γυρίσουμε στην κλάση περνάμε σαν τωρινή κατάσταση την αρχική.

```
be_now_ofotype_1 = start_point_ofotype_1;  
be_now_ofotype_2 = start_point_ofotype_2;  
be_now_ofotype_3 = start_point_ofotype_3;
```

Τι συμβαίνει όμως που τα βρήκαμε αυτά πως θα τα εισάγουμε στο πρόγραμμα μας ώστε να τα χειριζόμαστε;

```
if (id == R.id.nav_gas_station) { //gas station  
    if (Navigation_ok == 0) {  
        Toast.makeText(activity, "First Hit button bellow Search",  
Toast.LENGTH_LONG).show();  
        return true;  
    }  
    if (start_point_ofotype_1 == -1) {  
        Toast.makeText(activity, "Sorry Not Exist In map",  
Toast.LENGTH_LONG).show();  
        return true;  
    }  
    SetRoute setroute = new SetRoute();  
    setroute.setActivityUi(activity);  
    setroute.setRouteFromNaviPanel(pl.get(be_now_ofotype_1),  
placename.get(be_now_ofotype_1), vicinity.get(be_now_ofotype_1));  
    be_now_ofotype_1 += 1;  
    if (be_now_ofotype_1 > end_point_ofotype_1)  
        be_now_ofotype_1 = start_point_ofotype_1;  
} else if (id == R.id.nav_shop_grocery) { ...
```

Έχουμε φτάσει στο σημείο όπου μπορούμε να έχουμε όλες τις δυνατές πληροφορίες ώστε να κάνουμε αναζήτηση κατά χρόνο απόστασης σύμφωνα με το είδος του αντικείμενου που αναζητάμε. Έστω λοιπόν πατάμε το κουμπί για να οδηγήσει στο ποιο κοντινό σημείο για βενζινάδικο αν δεν έχουμε κάνει αναζήτηση μας εμφανίζει το κατάλληλο μήνυμα ,εάν πάλι δεν έχουμε κανένα σημείο πάλι θα δούμε ένα κατάλληλο μήνυμα αν όντως όλα είναι καλά τότε δημιουργούμε ένα αντικείμενο της SetRoute, μετά περνάμε το activity πολύ κύριο δεν το ξεχνάμε αν θέλουμε πχ να εμφανίσουμε μήνυμα στην οθόνη από άλλη κλάση .Όπου στο αντικείμενο θα του δώσουμε σύμφωνα με την θέση όπου βρισκόμαστε στον πίνακα **be_now_ofotype_1** όνομα , περιοχή και συντεταγμένες. Ακόλουθη η αύξηση του **be_now_ofotype_1** της τωρινής θέσης ,ο έλεγχος αν έχει ξεπεράσει τον αριθμό της τελευταίας θέσης στο είδος (πχ. Gas Station) εάν ναι τότε την θέτουμε ξανά στην αρχή κάτι σαν μια κυκλική στοίβα. Ύστερα από αυτά μένει η εξήγηση του Right Drawer και η εισαγωγή στην κυρίως PlaceAnd.. κλάση.

Layout nav_header_main2

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="80dp"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:icon="@drawable/cinemas"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        android:text="Etc Information "
        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />

</LinearLayout>
```



Αυτός ο κώδικας αφορά το navigation right drawer και συγκεκριμένα την κεφαλίδα του, όπως βλέπουμε δεν υπάρχει ιδιαίτερη διάφορα με τον παραπάνω του το navigation left drawer head από εδώ αφαιρέθηκε μονό η εικόνα.

Layout navi_right

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:clickable="true"
android:orientation="vertical">

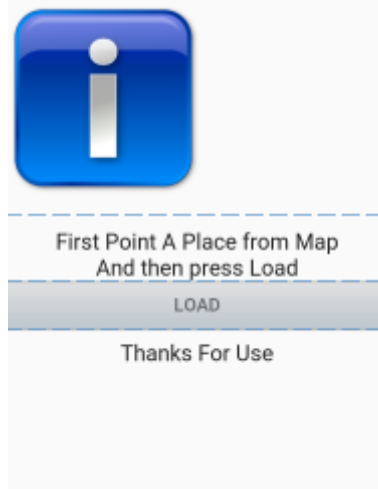
<ImageView
    android:id="@+id/imagenavi"
    android:layout_width="200dp"
    android:layout_centerHorizontal="true"
    android:layout_height="200dp"
    android:src="@drawable/info" />

<TextView
    android:layout_marginTop="30dp"
    android:id="@+id/footer_item_3"
    android:layout_width="match_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:maxLines = "1000"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="First Point A Place from Map\nAnd then press Load" />

<Button
    android:id="@+id/loadRightNavi"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:alpha="0.65"
    android:background="@drawable/button_rounded"
    android:text="Load"
    android:textSize="18sp" />
<TextView
    android:id="@+id/footer_item_2"
    android:layout_width="match_parent"
    android:layout_height="48dp"
    android:gravity="center"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Thanks For Use" />
</LinearLayout>
```

Ένα κύριο σημείο είναι το `android:src="@drawable/info"` θα παρακαλούσα την προσοχή εδώ καθώς αν απευθυνθούμε σε νέους μεθόδους φόρτωσης εικόνας η

διάφορα με τα sdk προκαλεί θέμα όχι μόνο για εδώ και σε αρκετά πράγματα. Αρά καταλαβαίνουμε η αντιγραφή δεν οδηγεί πάντα σε καλό και εφιστά την προσοχή μας.



Επίσης σε αυτό τον κώδικα κυρίως είναι τα id καθώς μετέπειτα στην φόρτωση εικόνας στο κουμπί και στο text view θα χρειαστούμε να επικοινωνήσουμε. Στην παραπάνω εικόνα παρουσιάζεται το Navigation right drawer main body το οποίο εμφανίζεται με το τράβηγμα της οθόνης από τα δεξιά στα αριστερά.

Class NaviRightLoader

```
package NavigationDrawers;
import android.app.Activity;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.tolis.maps3856.R;
import org.json.JSONException;
import SetMeetings.GetPlaceSchedule;
import SetMeetings.GetStringShopSuggested;
import static FindPlaces.VariablesFindPlaces.Places_photo_reference;
import static SetMeetings.SetMeetingsVariables.PlacesId;
import static SetMeetings.SetMeetingsVariables.PlacesName;
import static SetMeetings.SetMeetingsVariables.typeofmark;
import static SetMeetings.SetMeetingsVariables.types;
import static com.tolis.maps3856.PlaceAndIntrests.flagondemand;

public class NaviRightLoader {
    Activity activity;

    public NaviRightLoader(Activity activity) {
        this.activity = activity;
        goNext();
    }
}
```

```

private void goNext() {
    LinearLayout outer = (LinearLayout) activity.findViewById(R.id.outer2);
    final Button button = (Button) outer.findViewById(R.id.loadRightNavi);
    final TextView tectview = (TextView)
outer.findViewById(R.id.footer_item_3);
    final TextView tectview2 = (TextView) activity.findViewById(R.id.Vicinity);

    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(flagondemand==1)
                return;
            //run code
            int position = 0;
            //this is right oint for if ,if you want to have listener
            if (tectview2.getText().equals("Vicinity") ||
tectview2.getText().equals(""))
                return;
            TextView textView = (TextView) activity.findViewById(R.id.Title);
            if (textView.getText().equals("Title"))
                return;
            String title = (String) textView.getText();
            for (int i = 0; i < PlacesName.size(); i++) {
                if (PlacesName.get(i).equals(title))
                    break;
                position += 1;
            }
            if (position == Places_photo_reference.size()) {
                Toast.makeText(activity, "No NaviRight found",
Toast.LENGTH_SHORT).show();
                return;
            }
            Log.d("testa", String.valueOf(position));
            GetPlaceSchedule gps = new GetPlaceSchedule();
            gps.updatePlaces(PlacesId.get(position));
            final int position_handler = position;
            final String Name=PlacesName.get(position);
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    try {
                        final String temp=
String.valueOf(types.getJSONArray("weekday_text"));
                        activity.runOnUiThread(new Runnable() {
                            public void run() {
                                String temp_to_temp=temp;
                                temp_to_temp = temp.replace("[", "");
                                temp_to_temp = temp_to_temp.replace("]", "");

```

```

        temp_to_temp = temp_to_temp.replace("\\'", "'");
        temp_to_temp = temp_to_temp.replace("T", "\nT");
        temp_to_temp = temp_to_temp.replace("W", "\nW");
        temp_to_temp = temp_to_temp.replace("F", "\nF");
        temp_to_temp = temp_to_temp.replace("S", "\nS");
        tectview.setText(Name+" "+temp_to_temp);
    } });
} catch (NullPointerException e) {
    e.printStackTrace();
    String temp = "";
    if (typeofmark.get(position_handler) == 1) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Gas_Station();
    }
    if (typeofmark.get(position_handler) == 2) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Movie_Theater();
    }
    if (typeofmark.get(position_handler) == 3) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Grocery_Or_Supermarket();
    }
    if (typeofmark.get(position_handler) == 0) {

        temp = "no Data for this Place";
    }

    final String temp2= temp;
    activity.runOnUiThread(new Runnable() {
        public void run() {
            tectview.setText(Name+" "+temp2);
        } });
}
} catch (JSONException e){
}
}
}, 10);
String photo_refe = Places_photo_reference.get(position);
String url_string =
"https://maps.googleapis.com/maps/api/place/photo?maxwidth=500&photoreference=" + photo_refe + "&key=...";
DownloadImageTask DIT = new DownloadImageTask(activity);
if (photo_refe != null) {
    DIT.execute(url_string);
}
}
});
}
}

```

Πριν σχολιάσω τον κώδικα αξίζει να δούμε όταν θέλω να δημιουργήσω ένα layout το οποίο θα αποτελείται από πολλά layouts και αυτά με την σειρά τους θα έχουν διάφορα αντικείμενα παραδείγματος χάριν ένα κουμπί τότε θα πρέπει όταν εισάγω αυτά τα layouts να του δίνω ένα id είναι πολύ κύριο καθώς με σκέτα τα id των αντικειμένων δεν θα μπορώ να τα ελέγξω. Συγκεκριμένα η παρακάτω εικόνα, στην όποια γίνεται η εισαγωγή των παραπάνω layouts.

```

} <android.support.design.widget.NavigationView
    android:id="@+id/nav_view2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="end"
    android:scrollbars="vertical"
    android:fitsSystemWindows="true">
} <ScrollView
    android:id="@+id/SCROLLER_ID"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbars="vertical"
    android:fillViewport="true">
} <LinearLayout
    android:id="@+id/lightness"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <include android:id="@+id/outer" layout="@layout/nav_header_main2" />
    <include android:id="@+id/outer2" layout="@layout/navi_right" />
} </LinearLayout>
} </ScrollView>
} </android.support.design.widget.NavigationView>

```

Ξεκινώντας πάμε στην μέθοδο

```

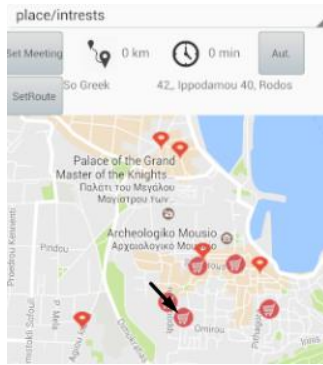
public NaviRightLoader(Activity activity) {
    this.activity = activity;
    goNext();
}

```

όπου περνά την δραστηριότητα στις μεταβλητές του αντικείμενου
`LinearLayout outer = (LinearLayout) activity.findViewById(R.id.outer2);`
final `Button button = (Button) outer.findViewById(R.id.loadRightNavi);`
 συνεχίζω στο ορισμό μια μεταβλητής τύπου `LinearLayout` όπου θα βρει το layout σύμφωνα με το συγκεκριμένο id και έτσι θα πάρουμε το κουμπί το οποίο βρίσκεται στο Layout αλλιώς θα παρουσιαζόταν πρόβλημα, για αυτό στην αρχή `outer.findViewById` βάζουμε το `outer` μπροστά από το `findViewById`.

`button.setOnClickListener`
 Listener για το κουμπί

Ακόλουθοι ο έλεγχος ο οποίος ελέγχει πρώτον αν είμαστε σε manual και αν έχουμε φορτώσει σημείο βλέποντας τα
`if(flagondemand==1)`
`if (tectiview2.getText().equals("Vicinity") || tectiview2.getText().equals(""))`



if (position == *Places_photo_reference*.size()) { ... **return;** }
 Δεν υπάρχει το ζητούμενο αντικείμενο.

for (int i = 0; i < *PlacesName*.size(); i++) {
 if (*PlacesName*.get(i).equals(title)) **break;** position += 1;}

Βρίσκουμε την θέση την οποία έχουμε το όνομα του σημείου και εν συνεχεία το *Places_photo_reference* .

type	duration	Name	Vicinity	LongLat	Places_photo_reference
1	12	Example1	Ag. Ioannou	34.22,23.89	HGYTUYJ67-XJ
3	11	Example2	Alikarnassos	34.21,23.86	HGYTUYJ67-XA
2	8	Example3	Estauromenos	34.24,23.85	HGYTUYJ67-XB
2	22	Example4	Limani	34.27,23.83	HGYTUYJ67-XC
1	2	Example5	42,Ipppodamou	34.27,23.84	HGYTUYJ67-XP

Το *Places_photo_reference* είναι ένας πίνακας λίστα ο οποίος δίνει ένα string κειμένου το οποίο έχει βρεθεί από την κλάση FindPlaces και έχει παρθεί όταν ζητάμε για σημεία ενδιαφέροντος από την google το string αποσκοπεί σε έναν κώδικα φωτογραφίας. Το κάθε αντικείμενο μπορεί να έχει μπορεί όμως να μην έχει αυτό εξαρτάται αν οι servers της google έχουν την φωτογραφία για το ανάλογο κατάσταση.

```

    "geometry" : {
      "location" : {
        "lat" : 36.4352612,
        "lng" : 28.224676
      },
      "viewport" : {
        "northeast" : {
          "lat" : 36.4353813,
          "lng" : 28.22485080000001
        },
        "southwest" : {
          "lat" : 36.4349009,
          "lng" : 28.2241516
        }
      }
    },
    "icon" : "https://maps.gstatic.com/mapfiles/plac
    "id" : "378bad7ec987f1177cee21d534b04e68621ee988
    "name" : "SUPERMARKET SPANOS, J. CH., GROUP OF C
    "opening_hours" : {
      "open_now" : false,
      "weekday_text" : []
    },
    "photos" : [
      {
        "height" : 2340,
        "html_attributions" : [
          "\u003ca href=\\"https://maps.google.com
          ],
        "photo_reference" : "CoQ8dwAAAPR_N1qC1IKjm
        "width" : 4160
      }
    ]
  }

```

Μετά θα δούμε μια κλάση `GetPlaceSchedule` η οποία θα μας δώσει το `Schedule` του σημείου ενδιαφέροντος (εξηγείτε παρακάτω).

```
handler.postDelayed(new Runnable() {  
    @Override  
    public void run() {  
  
    }, 10);
```

Ο handler είναι ένας χειρίστης ο οποίος του λεμέ να εκτελέσει κώδικα ύστερα από κάποια `MS` αυτό είναι αναγκαίο όταν εκτελούμε `web queries` και θέλουμε ο κώδικας παρακάτω να εκτελεστεί λίγο αργότερα. Μέσα σε αυτόν εκτελούμε κώδικα με τον οποίο αν υπάρχει απάντηση σχετικά με τις πληροφορίες προγράμματος καταστήματος αν όχι θα εκτελεστεί το `catch` με το οποίο δίνω δίκες μου τιμές συγκεκριμένα

```
if (typeofmark.get(position_handler) == 2) {  
    GetStringShopSuggested gsss = new GetStringShopSuggested();  
    temp = gsss.Movie_Theater();  
}
```

Εδώ θέτουμε το `text view` του `Layout` προσοχή σε `runOnUiThread`

```
activity.runOnUiThread(new Runnable() {  
    public void run() { tectview.setText(Name+" "+temp2); } });
```

Ύστερα περνούμε το `string` από την θέση του σημείου ενδιαφέροντος
`String photo_refe = Places_photo_reference.get(position);`

```
String url_string =
```

```
"https://maps.googleapis.com/maps/api/place/photo?maxwidth=500&photoreference=" + photo_refe  
    + "&key=... ";
```

```
Google query
```

Και εδώ εκτελούμε το κατέβασμα της φωτογραφίας εξηγείται ακριβώς από κάτω.

```
DownloadImageTask DIT = new DownloadImageTask(activity);
```

```
if (photo_refe != null) {  
    DIT.execute(url_string);  
}
```

Class DownloadImageTask

```
package NavigationDrawers;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Handler;
import android.widget.ImageView;
import android.widget.LinearLayout;
import com.tolis.maps3856.R;
import java.io.InputStream;

public class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
    Activity activity;
    public DownloadImageTask(Activity activity) {
        this.activity = activity;
    }

    protected Bitmap doInBackground(String... urls) {
        String urldisplay = urls[0];
        Bitmap mIcon11 = null;
        try {
            InputStream in = new java.net.URL(urldisplay).openStream();
            mIcon11 = BitmapFactory.decodeStream(in);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return mIcon11;
    }

    @Override
    protected void onPostExecute(final Bitmap result) {
        super.onPostExecute(result);
        LinearLayout outer = (LinearLayout)activity.findViewById(R.id.outer2);
        final ImageView it= (ImageView) outer.findViewById(R.id.imagenavi);
        if(result!=null)
            activity.runOnUiThread(new Runnable() {
                public void run() {
                    Handler handler = new Handler();
                    handler.postDelayed(new Runnable() {
                        @Override
                        public void run() {
                            it.setImageBitmap(result);
                        }
                    }, 10);
                }
            });
        if(result==null)
```

```

activity.runOnUiThread(new Runnable() {
    public void run() {
        String uriPath2 = "android.resource://com.tolis.maps3856/" +
R.drawable.no_image;
        Uri uri2 = Uri.parse(uriPath2);
        it.setImageURI(uri2);
    } });
}
}

```

Στην μέθοδο **protected** Bitmap doInBackground περνάμε το Url /google query και κατεβάζουμε την εικόνα την οποία , την περνάμε στην **protected void** onPostExecute όπου:

Βρίσκουμε το layout να πάρουμε id

```

LinearLayout outer = (LinearLayout)activity.findViewById(R.id.outer2);
final ImageView it= (ImageView) outer.findViewById(R.id.imagenavi);

```

Και αν βρεθεί η εικόνα την τοποθετούμε στο κατάλληλο σημείο βλέπε εικόνα παρακάτω

```

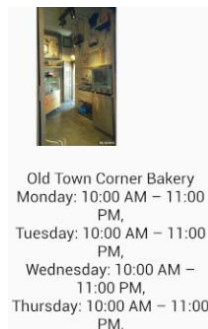
if(result!=null)

```

```

it.setImageBitmap(result);

```



Αλλιώς

```

if(result==null)

```

```

activity.runOnUiThread(new Runnable() {

```

```

    public void run() {

```

```

        String uriPath2 = "android.resource://com.tolis.maps3856/" +

```

```

R.drawable.no_image;

```

```

        Uri uri2 = Uri.parse(uriPath2);

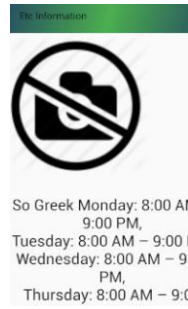
```

```

        it.setImageURI(uri2);
    }
}

```

Βάζουμε μια δικιά μας η οποία είναι αποθηκευμένη στο συγκεκριμένο Path. Στην περίπτωση αυτής της πτυχιακής έβαλα κάποια συγκεκριμένη ώστε να τονίσουμε την απουσία εικόνας.



Layout placeandintrests

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/drawer_layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context="com.tolis.maps3856.PlaceAndIntrests"
tools:openDrawer="start">
```

```
<LinearLayout
android:id="@+id/three"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical">
```

```
<Spinner
android:id="@+id/SpinToWin"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
/>
```

```
<LinearLayout
android:id="@+id/twolayout"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="vertical">
```

```
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:weightSum="1">
```

```
<Button
android:id="@+id/btnSetMeeting"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:alpha="0.65"
android:background="@drawable/button_rounded"
android:onClick="OnButtonClick2"
android:text="Set Meeting"
android:textSize="12dp" />
```

<ImageView

```
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginLeft="20dp"
android:layout_marginTop="5dp"
android:src="@drawable/ic_distance" />
```

<TextView

```
android:id="@+id/tvDistance"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="5dp"
android:text="0 km" />
```

<ImageView

```
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_marginLeft="20dp"
android:layout_marginTop="5dp"
android:padding="5dp"
android:src="@drawable/ic_clock" />
```

<TextView

```
android:id="@+id/tvDuration"
android:layout_width="29dp"
android:layout_height="wrap_content"
android:layout_marginLeft="5dp"
android:layout_weight="0.48"
android:text="0 min" />
```

<Button

```
android:id="@+id/btndemand"
android:layout_width="51dp"
android:layout_height="wrap_content"
android:alpha="0.65"
android:background="@drawable/button_rounded"
android:onClick="checkdemand"
android:text="Aut."
android:textSize="12dp" />
```

</LinearLayout>

</LinearLayout>

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:orientation="horizontal"
    android:paddingTop="2dp"
    android:weightSum="1">

    <Button
        android:id="@+id/setroute"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:alpha="0.65"
        android:background="@drawable/button_rounded"
        android:onClick="OnButtonClick"
        android:text="SetRoute"
        android:textSize="12dp" />

    <TextView
        android:id="@+id/Title"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="0.60"
        android:text="Title"
        android:textSize="13dp" />

    <TextView
        android:id="@+id/Vicinity"
        android:layout_width="123dp"
        android:layout_height="match_parent"
        android:layout_weight="0.41"
        android:text="Vicinity"
        android:textSize="13dp" />
</LinearLayout>

<EditText
    android:id="@+id/placedime"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name"
    android:visibility="gone" />

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
</LinearLayout>

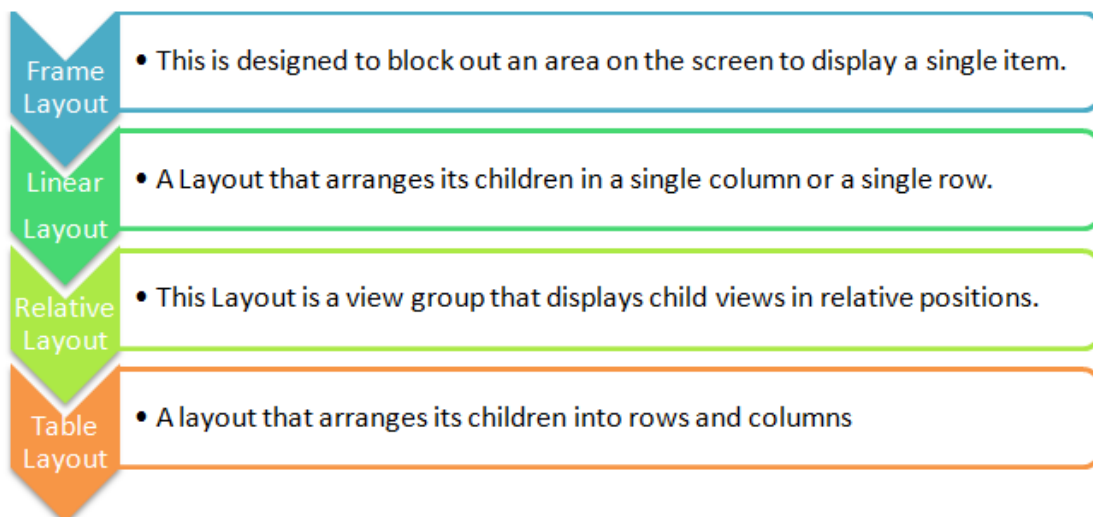
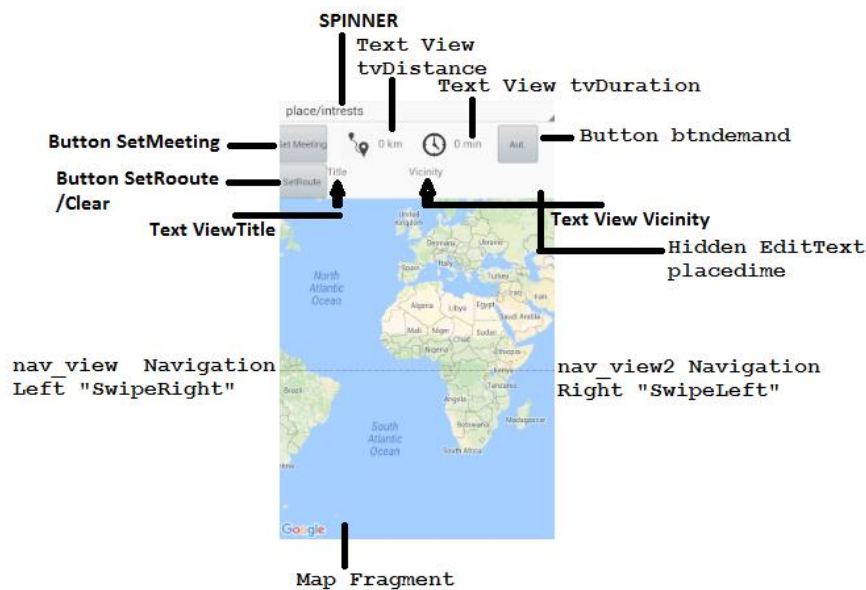
```

```

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="left"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_main"
    app:itemIconTint="@color/colorPrimaryDark"
    app:menu="@menu/activity_main_drawer" />

<android.support.design.widget.NavigationView
    android:id="@+id/nav_view2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="end"
    android:scrollbars="vertical"
    android:fitsSystemWindows="true">
    <ScrollView
        android:id="@+id/SCROLLER_ID"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical"
        android:fillViewport="true">
        <LinearLayout
            android:id="@+id/lightness"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <include android:id="@+id/outer"
layout="@layout/nav_header_main2" />
            <include android:id="@+id/outer2" layout="@layout/navi_right" />
        </LinearLayout>
    </ScrollView>
</android.support.design.widget.NavigationView>
</android.support.v4.widget.DrawerLayout>

```



Το Drawer Layout αποτελείται ως συνήθως από Frame Layout και ένα List view το λεγόμενο menu.

Σε αυτό νέο layout παρατηρούμε της εξής διαφορές σε σχέση με τα προηγούμενα:

- 1) **<android.support.v4.widget.DrawerLayout>**
Το πρώτο που συναντάμε και κλείνει σε tag όλο το layout , χρησιμεύει στην εμφάνιση των navigation drawers.
- 2) **<Spinner>**
Είναι η μπάρα με τις επιλογές για να αλλάζουμε activities η exit.
- 3) **<fragment**

```

android:id="@+id/map"
android:name="com.google.android.gms.maps.SupportMapFragment"
android:layout_width="match_parent"

```

```
android:layout_height="match_parent" />
```

Είναι το layout που αναφέρεται στον χάρτη ,γιατί match ; Για να γεμίσει πλήρως το layout.

```
4) <android.support.design.widget.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="left"
    android:fitsSystemWindows="true"
    app:headerLayout="@layout/nav_header_main"
    app:itemIconTint="@color/colorPrimaryDark"
    app:menu="@menu/activity_main_drawer" />
```

Όπως βλέπουμε αυτό είναι το πρώτο navigation drawer left και η εισαγωγή του γίνεται έτσι , τα 3 πρώτα χαρακτηριστικά είναι γνωστά το 4 αφορά την μεριά που θα μπει το layout `layout_gravity="left"` και συγκεκριμένα την δεξιά στην συνέχεια βλέπουμε την εισαγωγή μιας κεφαλίδας layout `header Layout` και το `menu` layout .

```
5) <android.support.design.widget.NavigationView
    android:id="@+id/nav_view2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="end"
    android:scrollbars="vertical"
    android:fitsSystemWindows="true">
    <ScrollView
        android:id="@+id/SCROLLER_ID"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:scrollbars="vertical"
        android:fillViewport="true">
        <LinearLayout
            android:id="@+id/lightness"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">
            <include android:id="@+id/outer"
                layout="@layout/nav_header_main2" />
            <include android:id="@+id/outer2" layout="@layout/navi_right" />
        </LinearLayout>
    </ScrollView>
</android.support.design.widget.NavigationView>
```

Τέλος αυτό είναι το αριστερό Navigation Drawer όπου τοποθετούμε έτσι `android:layout_gravity="end"` στην αριστερή μεριά, `<ScrollView>` με το οποίο θα είναι η μπάρα που θα ανεβοκατεβαίνει το layout με τις πληροφορίες καταστήματος.



```
<include android:id="@+id/outer" layout="@layout/nav_header_main2" />
<include android:id="@+id/outer2" layout="@layout/navi_right" />
```

Και το πιο σημαντικό πράγμα είναι αυτό όπου τοποθετούμε τα ξένα layout στα οποία βάζουμε id για τον λόγο της πρόσβασης μέσα στα στοιχεία των layout.

Class PlaceAndIntrests

```
package com.tolis.maps3856;
import android.Manifest;
import android.content.Intent;
import android.content.IntentSender;
import android.content.pm.PackageManager;
import android.content.res.Configuration;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.NonNull;
import android.support.annotation.Nullable;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.Spinner;
import android.widget.TextView;
```

```

import android.widget.Toast;
import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;

```

```

import NavigationDrawers.NaviRightLoader;
import FindPlaces.FindPlaces;
import GetYourPlace.GetYourPlace;
import Listeners.MarkerListener;
import NavigationDrawers.NavigationLeft;
import SetRoute.SetRoute;

```

```

import static FindPlaces.VariablesFindPlaces.Duration;
import static FindPlaces.VariablesFindPlaces.Places_LongLatitude;
import static FindPlaces.VariablesFindPlaces.currentLatitude;
import static FindPlaces.VariablesFindPlaces.currentLongitude;
import static FindPlaces.VariablesFindPlaces.existisnextpage;
import static FindPlaces.VariablesFindPlaces.if_fine_then_search;
import static FindPlaces.VariablesFindPlaces.mMap;
import static SetMeetings.SetMeetingsVariables.PlacesId;
import static SetMeetings.SetMeetingsVariables.PlacesName;
import static SetMeetings.SetMeetingsVariables.typeofmark;
import static SetRoute.SetRouteVariables.Navigation_ok;

```

```

public class PlaceAndIntrests extends FragmentActivity implements
OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks,
GoogleApiClient.OnConnectionFailedListener, LocationListener {
    private Spinner s;
    private GoogleApiClient mGoogleApiClient;
    public static final String TAG = PlaceAndIntrests.class.getSimpleName();
    private String[] arraySpinner;
    private LinearLayout mainlayout;
    private final static int CONNECTION_FAILURE_RESOLUTION_REQUEST
= 9000;
    private LocationRequest mLocationRequest;
    public static int setRoute_clear_button_state = 0;
    private int doublerotation = 0;
    public static int flagondemand = 2;

```



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    Toast.makeText(getApplicationContext(), "Rotate Screen 2 times for refresh
in Man Mode", Toast.LENGTH_LONG).show();
    setRoute_clear_button_state = 0; //obvious reasons if a back is made its mean that
you have not any set
    super.onCreate(savedInstanceState);
    Intent intent = getIntent();
    setContentView(R.layout.placeandintrests);
    // Obtain the SupportMapFragment and get notified when the map is ready to be
used.
    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
    .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);

    Duration.removeAll(Duration);
    Navigation_ok = 0;
    if_fine_then_search=0; //for navigation panel if you refresh then search again
    NavigationLeft nl = new NavigationLeft();
    nl.setActivityUi(this);
    nl.NaviListe();

    NaviRightLoader ll = new NaviRightLoader(this);

    SpinToWin();
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
    mLocationRequest = LocationRequest.create()
        .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
        .setInterval(10 * 1000) // 10 seconds, in milliseconds
        .setFastestInterval(5 * 1000); // 5 second, in milliseconds

    if (mMap != null) {
        //ok - proceed
        mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        //update location
    }
    try {
        String ActivityBefore = getIntent().getStringExtra("ActivityBefore");
        if (ActivityBefore.equals("CheckSeeMeetings"))
            setMeeting();
    } catch (Exception e) {
    }
}

```

```

}

private void setMeeting() {
    String Title = getIntent().getStringExtra("Title");
    String Vicinity = getIntent().getStringExtra("Vicinity");
    String LongLatofplace = getIntent().getStringExtra("LongLatofplace");
    Toast.makeText(PlaceAndIntrests.this, "Wait Load Route From Meeting",
Toast.LENGTH_LONG).show();
    TextView myAwesomeTextView = (TextView) findViewById(R.id.Title);
    TextView myAwesomeTextView2 = (TextView) findViewById(R.id.Vicinity);
    final TextView placedime1 = (TextView) findViewById(R.id.placedime);
    myAwesomeTextView.setText(Title);
    myAwesomeTextView2.setText(Vicinity);
    placedime1.setText(LongLatofplace);
    setRoute_clear_button_state = 1;
    final int camefrommeeting = 1;
    SetRoute setroute = new SetRoute();
    setroute.setRoute(this);
    Button p1_button = (Button) findViewById(R.id.setroute);
    p1_button.setText("Clear");
    Handler handler2 = new Handler();
    handler2.postDelayed(new Runnable() {
        @Override
        public void run() {
            updatemanual(camefrommeeting);
        }
    }, 3000);
}
}

```

```

public void OnButtonClick(View v) {
    TextView Vicinity = (TextView) findViewById(R.id.Vicinity);
    String Vicinity2 = Vicinity.getText().toString();
    if (v.getId() == R.id.setroute) {
        Button b = (Button) v;
        if (b.getText().toString().equals("Clear")) {
            setRoute_clear_button_state = 0;
            Button p2_button = (Button) findViewById(R.id.setroute);
            p2_button.setText("SetRoute");
            SetRoute setroute = new SetRoute();
            setroute.clearRoute(this);
            return;
        }
    }

    if (!Vicinity2.equals("Vicinity") && !Vicinity2.equals("")) {
        if (v.getId() == R.id.setroute) {

```

```

Button b = (Button) v;
if (b.getText().toString().equals("SetRoute")) {
    setRoute_clear_button_state = 1;
    SetRoute setroute = new SetRoute();
    setroute.setRoute(this);
    Button p1_button = (Button) findViewById(R.id.setroute);
    p1_button.setText("Clear");
}
}
}
}

public void OnButtonClick2(View v) {
if (v.getId() == R.id.btnSetMeeting) {
try {
    if(flagondemand==1)
        return;
    if (setRoute_clear_button_state == 1) {
        TextView placedime1 = (TextView) findViewById(R.id.placedime);
        String LongLatofplace = placedime1.getText().toString();
        LongLatofplace = LongLatofplace.replace("lat", "");
        LongLatofplace = LongLatofplace.replace("/", "");
        LongLatofplace = LongLatofplace.replace("lng", "");
        LongLatofplace = LongLatofplace.replace(":", "");
        LongLatofplace = LongLatofplace.replace("(", "");
        LongLatofplace = LongLatofplace.replace(")", "");
        TextView myAwesomeTextView = (TextView)
findViewById(R.id.Title);
        TextView myAwesomeTextView2 = (TextView)
findViewById(R.id.Vicinity);
        TextView myAwesomeTextView3 = (TextView)
findViewById(R.id.tvDistance);
        TextView myAwesomeTextView4 = (TextView)
findViewById(R.id.tvDuration);
        int position = 0;
        for (String str : PlacesName) {

            if (str.contains(myAwesomeTextView.getText().toString()))
                break;
            position += 1;
        }
        if (typeofmark.get(position) != 1 && typeofmark.get(position) != 2 &&
typeofmark.get(position) != 3)
            return;

        Intent intent = new Intent(PlaceAndIntrests.this, SetMeeting.class);
        intent.putExtra("Title", myAwesomeTextView.getText().toString());
        intent.putExtra("Vicinity",
myAwesomeTextView2.getText().toString());
        if (myAwesomeTextView3.getText().toString().equals("0 km"))//quick

```

fingers.....

```
        return;
        intent.putExtra("Distance",
myAwesomeTextView3.getText().toString());
        intent.putExtra("Duration",
myAwesomeTextView4.getText().toString());
        intent.putExtra("LongLatofplace", LongLatofplace);

        PlaceAndIntrests.this.startActivity(intent);
        PlaceAndIntrests.this.finish();
    }
} catch (Exception e) {
    Toast.makeText(getApplicationContext(), "Something Went Wrong Wait
Load Data", Toast.LENGTH_LONG).show();
    return;
}
}
}
```

```
public void checkdemand(View v) {

    if (v.getId() == R.id.btndemand) {
        Button b = (Button) v;
        if (b.getText().toString().equals("Aut.")) {
            flagondemand = 1;
            if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    requestPermissions(new String[]
                        {Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.INTERNET}
                        , 30);
                }
            }
            mLocationRequest = LocationRequest.create()
                .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
                .setInterval(10 * 1000) // 10 seconds, in milliseconds
                .setFastestInterval(5 * 1000); // 5 second, in milliseconds

            LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
            Button p1_button = (Button) findViewById(R.id.btndemand);
            p1_button.setText("Man.");
            return;
        }
    }
}
```

```

    }

    if (b.getText().toString().equals("Man.")) {
        mLocationRequest = null;

LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
this);
        flagondemand = 2;
        Button p1_button = (Button) findViewById(R.id.btndemand);
        p1_button.setText("Aut.");
        return;
    }
}

}

//on rotate print a message
public void onConfigurationChanged(Configuration newConfig) {
    super.onConfigurationChanged(newConfig);

    if (flagondemand == 2) {
        doublerotation = doublerotation + 1;
        if (doublerotation == 2) {
            final int camefrommeeting = 0; //not come from meeting
            updatemanual(camefrommeeting);
            doublerotation = 0;
        }
    }
}

private void updatemanual(final int camefrommeeting) {
    try {
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                requestPermissions(new String[]
                    {Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION,
                    Manifest.permission.INTERNET}
                    , 30);
            }
        }
        mLocationRequest = LocationRequest.create()
            .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
            .setInterval(10 * 1000) // 10 seconds, in milliseconds
            .setFastestInterval(2 * 1000); // 2 second, in milliseconds
    }
}

```

```

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
    Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            if (ActivityCompat.checkSelfPermission(PlaceAndIntrests.this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(PlaceAndIntrests.this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                    requestPermissions(new String[]
                        {Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.INTERNET}
                        , 30);
                }
                // finish();
                // return;
            }
            remove();
            Navigation_ok=0;
            if_fine_then_search=0;//for navigation panel if you refresh then search
again
            Location location =
LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
            handleNewLocation(location);
            final SetRoute setroute = new SetRoute();
            if (camefrommeeting == 0)
                if (setRoute_clear_button_state == 1)
                    setroute.setRoute(PlaceAndIntrests.this);
            if (camefrommeeting == 1) {
                Handler handler = new Handler();
                handler.postDelayed(new Runnable() {
                    @Override
                    public void run() {
                        setroute.setRoutefromMeeting(PlaceAndIntrests.this);
                    }
                }, 2 * 2000);
            }
            if (!mGoogleApiClient.isConnected()) {
                return;
            }

```

```

LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
PlaceAndIntrests.this);
        mLocationRequest = null;

```

```

    }
    }, 4000);
} catch (IllegalStateException e) {
    Toast.makeText(getApplicationContext(), "Something Went Wrong Try
again after some seconds", Toast.LENGTH_LONG).show();

} catch (Exception e) {
    Toast.makeText(getApplicationContext(), "Something Went Wrong Try
again after some seconds", Toast.LENGTH_LONG).show();
}
}

```

```

private void remove()//clear present values for matrix
{
    typeofmark.removeAll(typeofmark);
    PlacesName.removeAll(PlacesName);
    PlacesId.removeAll(PlacesId);
    mMap.clear();
}

```

```

private void SpinToWin() {
    this.arraySpinner = new String[]{
        "place/intrests", "Route Only", "See Meetings", "Exit"
    };
    s = (Spinner) findViewById(R.id.SpinToWin);
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, arraySpinner);

    s.setAdapter(adapter);
    mainlayout = (LinearLayout) this.findViewById(R.id.twolayout);
    s.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            String selectedItem = parent.getItemAtPosition(position).toString();
            if (selectedItem.equals("place/intrests")) {
                mainlayout.setVisibility(LinearLayout.VISIBLE); // do your stuff
            } else {
                mainlayout.setVisibility(LinearLayout.GONE);
                if (selectedItem.equals("Route Only")) {
                    Intent intent = new Intent(PlaceAndIntrests.this, RouteOnly.class);

                    PlaceAndIntrests.this.startActivity(intent);
                    PlaceAndIntrests.this.finish();
                }
                if (selectedItem.equals("See Meetings")) {
                    Intent intent = new Intent(PlaceAndIntrests.this,

```

CheckSeeMeetings.**class**);

```
        PlaceAndIntrests.this.startActivity(intent);
        PlaceAndIntrests.this.finish();
    }
    if (selectedItem.equals("Exit")) {
        PlaceAndIntrests.this.finish();
        System.exit(0);
    }
}
} // to close the onItemSelected

public void onNothingSelected(AdapterView<?> parent) {

}
});
}
```

@Override

```
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    MarkerListener ml = new MarkerListener();
    ml.setActivityUi(this);
    ml.Markerlistener();
}

private void handleNewLocation(Location location) {
    try {
        Log.d(TAG, location.toString());
        currentLatitude = location.getLatitude();
        currentLongitude = location.getLongitude();
        if (setRoute_clear_button_state == 1) {
            SetRoute setroute = new SetRoute();
            setroute.setRoute(this);
        }
        //its not work on pc so put rhodes home location

        if (currentLongitude == -122.084000) {
            currentLatitude = 36.436285;
            currentLongitude = 28.217539;
            // currentLatitude = 35.319647;
            // currentLongitude =25.102336;
        }
        LatLng latLng = new LatLng(currentLatitude, currentLongitude);
        GetYourPlace gyp = new GetYourPlace();
        gyp.setActivityUi(this);
        String s = gyp.getCompleteAddressString(currentLatitude, currentLongitude);
```



```

MarkerOptions options = new MarkerOptions()
    .title(s)
    .icon(BitmapDescriptorFactory.fromResource(R.drawable.user))
    .position(latLng);

mMap.addMarker(options);
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 15));
Places_LongLatitubee.removeAll(Places_LongLatitubee);
Duration.removeAll(Duration);
final FindPlaces fp = new FindPlaces();
existisnextpage = 1;
fp.updatePlaces();
} catch (IllegalStateException e) {
} catch (NullPointerException e) {

}

}

}

@Override
public void onConnected(@Nullable Bundle bundle) {

    Log.i(TAG, "Location services connected.");
    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new String[]
                {Manifest.permission.ACCESS_COARSE_LOCATION,
Manifest.permission.ACCESS_FINE_LOCATION,
                Manifest.permission.INTERNET}
                , 30);
        }
    }

    Button p1_button = (Button) findViewById(R.id.btndemand);
    p1_button.setText("Aut.");
    flagondemand = 2;

}

@Override
public void onConnectionSuspended(int i) {
    Log.i(TAG, "Location services suspended. Please reconnect.");
}

```

```

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    if (connectionResult.hasResolution()) {
        try {
            // Start an Activity that tries to resolve the error
            connectionResult.startResolutionForResult(this,
CONNECTION_FAILURE_RESOLUTION_REQUEST);
        } catch (IntentSender.SendIntentException e) {
            e.printStackTrace();
        }
    } else {
        Log.i(TAG, "Location services connection failed with code " +
connectionResult.getErrorCode());
    }
}

```

```

@Override
protected void onResume() {

    super.onResume();
    // setUpMapIfNeeded();
    mGoogleApiClient.connect();
}

```

```

@Override
protected void onPause() {
    super.onPause();
    if (mGoogleApiClient.isConnected()) {

```

```

LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
this);
        mGoogleApiClient.disconnect();
    }
}

```

```

@Override
public void onLocationChanged(Location location) {
    if (flagondemand == 1) {
        remove();
        handleNewLocation(location);
        if (setRoute_clear_button_state == 1) {
            SetRoute setroute = new SetRoute();
            setroute.setRoute(this);
        }
    }
}
}
}
}

```

```

public class PlaceAndIntrests implements OnMapReadyCallback, GoogleApiClient.ConnectionCa
    private Spinner s;
    private GoogleApiClient mGoogleApiClient;
    public static final String TAG = PlaceAndIntrests.class.getSimpleName();
    private String[] arraySpinner;
    private LinearLayout mainlayout;
    private final static int CONNECTION_FAILURE_RESOLUTION_REQUEST = 9000;
    private LocationRequest mLocationRequest;
    public static int setRoute_clear_button_state = 0;
    private int doublerotation = 0;
    public static int flagondemand = 2;

    // LinearLayout one = (LinearLayout) findViewById(R.id.one);
    //one.setVisibility(View.GONE);
    @Override
    protected void onCreate(Bundle savedInstanceState) {

        setRoute_clear_button_state = 0; //obvious reasons if a back is made its mean that
        super.onCreate(savedInstanceState);
        Intent intent = getIntent();
        setContentView(R.layout.placeandintrests);
        // Obtain the SupportMapFragment and get notified when the map is ready to be use
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.map);

```

Αυτή είναι μια φωτογραφία όπου έχω αφαίρεση επίτηδες το extend Frame Layout για να δείξω την σημαντικότητα της κληρονομιάς από αυτό.

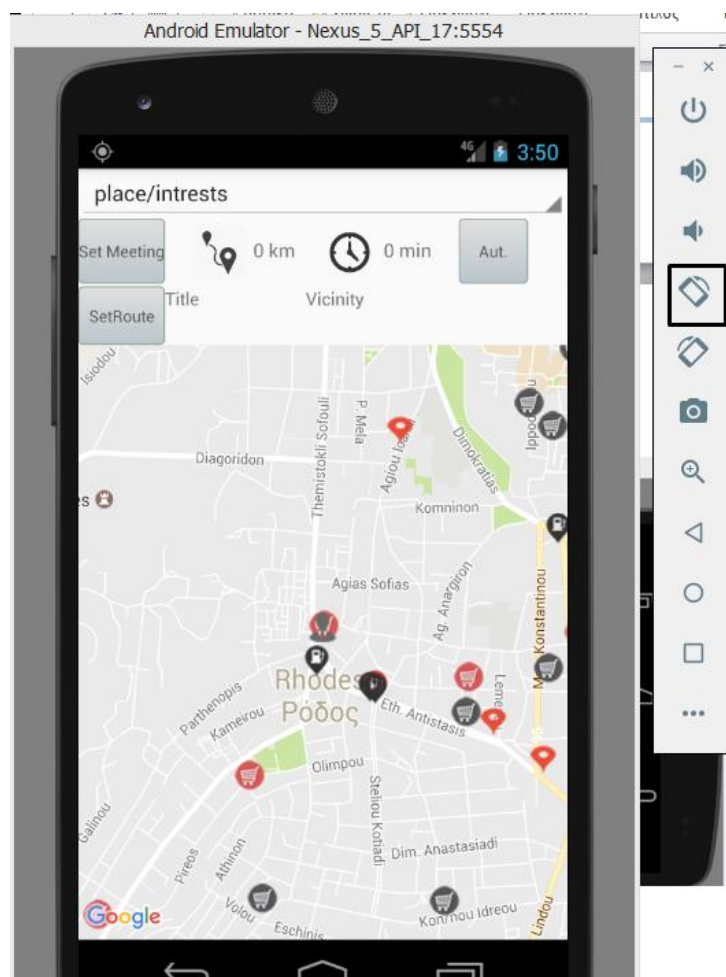
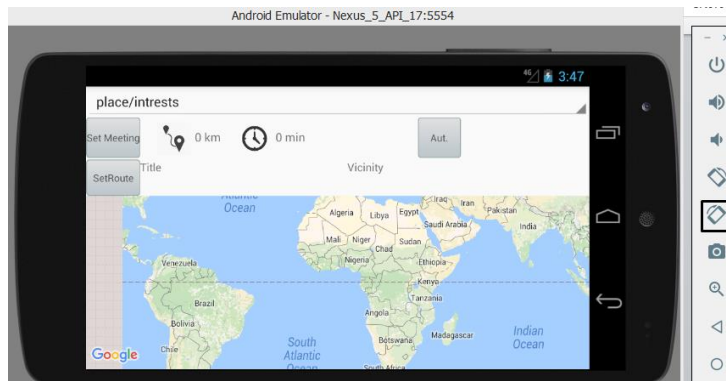
Όσον αφορά αυτή η κλάση είναι η κεντρική και ποιο σημαντική για αυτό θέλει και ιδιαίτερη προσοχή. Λοιπόν στην αρχή πριν μπω στις κλάσεις και πριν το flow diagram περιέγραψα τις static μεταβλητές ώστε να μην υπάρχει θέμα πολυπλοκότητας από κάτω θα εξηγηθεί γενικά τι γίνεται χωρίς να τους δώσω ιδιαίτερη σημασία.

```

private Spinner s;
private GoogleApiClient mGoogleApiClient;
private String[] arraySpinner;
private LinearLayout mainlayout;
private final static int CONNECTION_FAILURE_RESOLUTION_REQUEST =
9000;
private LocationRequest mLocationRequest;
private int doublerotation = 0;

```

Αυτές οι τοπικές μεταβλητές αφορούν το Spinner, τον GoogleClient, **arraySpinner** ένα πίνακα όπου θα κρατώ τα string του spinner , mainlayout αφορά τον spinner ,mLocationRequest ένα από τα ποιο σημαντικά για την εύρεση θέσης και τον τρόπο ζήτησης της και doublerotation για την περιστροφή της οθόνης και ζήτηση ανανέωσης.



Κάπως έτσι με το γύρισμα της οθόνης θα πετύχουμε την ανανέωση με Manual Mode.

protected void onCreate(Bundle savedInstanceState)

Τι βλέπουμε σε αυτήν την μέθοδο:

1. Πρώτον θέτουμε την μεταβλητή `setRoute_clear_button_state = 0`; που είναι σαν θέτουμε ότι το SetRoute Button δεν έχει καμιά διαδρομή και είναι ελεύθερο να εντοπίσει μια.
2. `SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.map);`
`mapFragment.getMapAsync(this);`
Αυτή είναι αυτόματη εισαγωγή όταν δημιουργήσαμε το activity με τον χάρτη.
3. Όταν ξεκινήσαμε ένα activity γυρνάμε πίσω από drawers κλπ πρέπει να διαγράψουμε και τις παλαιές μεταβλητές για να μην έχουμε τοίχων θέματα έτσι λοιπόν θα αναθέσουμε τον πίνακα λίστα με τις διάρκειες των σημείων ενδιαφέροντος σε null , το Navigation σε 0 που βλέπουμε άμα έχει γίνει πλήρως ενημέρωση σε όλα τα σημεία ο χρόνος διάρκειας απόστασης και το if_fine που ελέγχει αν έχουμε κατεβάσει της βασικές πληροφορίες για όλα τα σημεία όπως θέση vicinity κλπ.
`Duration.removeAll(Duration);`
`Navigation_ok = 0;`
`if_fine_then_search=0;`
4. `NavigationLeft nl = new NavigationLeft();`
`nl.setActivityUi(this);`
`nl.NaviListe();`
`NaviRightLoader ll = new NaviRightLoader(this);`
Εισαγωγή των Navigation Drawers.
5. `SpinToWin();`
Μέθοδος του spinner
6. `mGoogleApiClient = new GoogleApiClient.Builder(this)`
`addConnectionCallbacks(this)`
`.addOnConnectionFailedListener(this)`
`.addApi(LocationServices.API)`
`.build();`

Με αυτό λεμέ στις δυο πρώτες γραμμές ότι αυτά που θα γίνουν για σύνδεση και τα λοιπά θα τα διαχειρίζεται αυτή η κλάση με το προτελευταίο βάζουμε το Api για τα location και με το built το κτίζουμε.

7. `mLocationRequest = LocationRequest.create()`
`.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)`
`.setInterval(10 * 1000) // 10 seconds, in milliseconds`
`.setFastestInterval(5 * 1000); // 4 second, in milliseconds`

Το location request το οποίο είναι χρήσιμο στην ανανέωση του χάρτη δηλαδή αν έχουμε επιλέξει την Aut. Mode τότε η ανανέωση θα γίνεται κάθε 10 δευτερόλεπτα εκτός και αν εμφανιστεί κάτι πιο νωρίς στα 5 με το Fast(πχ αν κάποια άλλη εφαρμογή χρησιμοποιεί location services).

```

8. if (mMap != null) {
    //ok - proceed
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    //update location
}

```

mMap Είναι η μεταβλητή του χάρτη και λεμέ αν είναι διαφορετική του null τότε θέτουμε τον χάρτη σε normal type ...Υπάρχουν και αλλά είδη χαρτών για αυτή την πτυχιακή προτίμησα το κανονικό.

```

9. try {
    String ActivityBefore = getIntent().getStringExtra("ActivityBefore");
    if (ActivityBefore.equals("CheckSeeMeetings"))
        setMeeting();
} catch (Exception e) {
}

```

Με την βοήθεια των try και catch έλεγχο αν σε περίπτωση έχω έρθει από την δραστηριότητα **CheckSeeMeetings** και τότε τρέχω την παρακάτω μέθοδο.

private void setMeeting()

Τι βλέπουμε σε αυτήν την μέθοδο:

- String Title = getIntent().getStringExtra(**"Title"**);
Είσοδος στοιχείων από την κλάση CheckSeeMeetings ώστε να μπορούμε να θέσουμε την διαδρομή
- TextView myAwesomeTextView2 = (TextView) findViewById(R.id.**Vicinity**);
myAwesomeTextView2.setText(Vicinity);
Θέτουμε τα στοιχεία στα αντίστοιχα TextView.
- setRoute_clear_button_state = 1;
Button p1_button = (Button) findViewById(R.id.**setroute**);
p1_button.setText(**"Clear"**);
Θέτουμε την κατάσταση κουμπιού ότι υπάρχει διαδρομή για να κάνουμε clear.
- SetRoute setroute = **new** SetRoute();
setroute.setRoute(**this**);
Εξηγείται παρακάτω.
- Handler handler2 = **new** Handler();
handler2.postDelayed(**new** Runnable() {
 @Override
 public void run() {
 updatemanual(**camefrommeeting**);
 }
}, 3000);

Μετά από 3 δευτερόλεπτα τρέχουμε την συγκεκριμένη μέθοδο updatemanual.

public void OnButtonClick(View v)

Τι βλέπουμε σε αυτήν την μέθοδο:

1. Όπως και στο Intro έτσι και εδώ έχω τοποθετήσει μια button μέθοδο αυτή ενώνεται με το Layout που κάπου στις ρυθμίσεις λέει όταν κανείς on click έλα και τρέξε αυτή την μέθοδο.
2. Η διάφορα εδώ είναι ότι το χρησιμοποιώ σαν state button on/off που αν πατήσω `p2_button.setText("SetRoute");`
Αλλάζω και το text.
3. `SetRoute setroute = new SetRoute();`
`setroute.clearRoute(this);`
return;
Διαγράφω την διαδρομή από τον χάρτη μέσω της κλάσης.
4. `setRoute_clear_button_state = 1;`
`SetRoute setroute = new SetRoute();`
`setroute.setRoute(this);`
`Button p1_button = (Button) findViewById(R.id.setroute);`
`p1_button.setText("Clear");`
Και τέλος εδώ θέτω την διαδρομή στον χάρτη όπου θα γίνει η αλλαγή στο text view στο button.
5. `if (!Vicinity2.equals("Vicinity") && !Vicinity2.equals(""))`
Δεν ξεχνάμε τις προφανείς παραμέτρους που χρειάζεται για να θέσω την διαδρομή αν στο text view παραμένει το Vicinity τότε σημαίνει πως δεν έχω επιλέξει σημείο ενδιαφέροντος για να κάνω SetRoute.

public void OnButtonClick2(View v)

Τι βλέπουμε σε αυτήν την μέθοδο:

1. `if(flagondemand==1)`
return;
Εάν έχω το Aut. Mode να μην πάει να συνεχίσει για να μην υπάρχουν επιπλοκές με την συνέχεια ενημέρωση.
2. `if (setRoute_clear_button_state == 1)`
Εάν υπάρχει διαδρομή δηλαδή έχουμε βρει διαδρομή.
3. `TextView placedime1 = (TextView) findViewById(R.id.placedime);`
`String LongLatofplace = placedime1.getText().toString();`
`LongLatofplace = LongLatofplace.replace("lat", "");`
Πήγαινε στο κρυφό place dime1 και πάρε τα στοιχεία του, αυτό ίσως και είναι μια κακή κίνηση προτιμότερο θα είναι να γίνει σε κάποιο πινάκα με static. Μετά ακολουθούνε κάποια replace γιατί όμως; Γιατί όταν περνά τα στοιχεία από σημείο είναι σε μορφή lat/long:(...) εγώ όμως θέλω το μέσα περιεχόμενο, και γιατί το θέλω; Γιατί όταν δημιουργώ query το θέλει σε συγκεκριμένη μορφή του τύπου 234324,234234
4. `int position = 0;`
`for (String str : PlacesName) {`

`if (str.contains(myAwesomeTextView.getText().toString()))`
`break;`
`position += 1;`
`}`
`if (typeofmark.get(position) != 1 && typeofmark.get(position) != 2 &&`

```
typeofmark.get(position) != 3)
```

```
return;
```

Όπως βλέπει κανείς εδώ βρίσκουμε την θέση του σημείου μα γιατί;;; Για να ελέγξουμε αν είναι ένα από τα τρία σημεία ενδιαφέροντος που ζητηθήκαν στην πτυχιακή.

5. Ύστερα θα δούμε να περνά τιμές από τα text view όνομα κλπ όπου αυτό στην επομένη κλάση το χρησιμεύω σαν στοιχείά του σημείου ενδιαφέροντος
Intent intent = new Intent(PlaceAndIntrests.this, SetMeeting.class);
TextView myAwesomeTextView4 = (TextView)
findViewById(R.id.tvDuration);
Όπου τα κάνουμε pass με αυτή την μορφή
intent.putExtra("Distance", myAwesomeTextView3.getText().toString());
σε νέο activity.
6. PlaceAndIntrests.this.startActivity(intent);
PlaceAndIntrests.this.finish();
Και έτσι θα περάσουμε στο νέο activity

```
if(ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_FINE_L  
OCATION) != PackageManager.PERMISSION_GRANTED  
&&ActivityCompat.checkSelfPermission(this,Manifest.permission.ACCESS_COAR  
SE_LOCATION) != PackageManager.PERMISSION_GRANTED) {  
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
requestPermissions(new String[]  
{Manifest.permission.ACCESS_COARSE_LOCATION,  
Manifest.permission.ACCESS_FINE_LOCATION,  
Manifest.permission.INTERNET}  
, 30);  
}  
}
```

Αυτό το σημείο κώδικα θα το δούμε συνεχώς αφορά τα δικαιώματα που θέλει η εφαρμογή , αυτόματα το πέτα ο compiler εγώ έκανα μια μικρή προσαρμογή ώστε να ρώτα.

```
public void checkdemand(View v)
```

Αυτή η μέθοδος αφορά το demand το Aut. Man.

```
if (v.getId() == R.id.btndemand) {  
Button b = (Button) v;  
if (b.getText().toString().equals("Aut. ")) {  
flagondemand = 1;  
if ..... 30);  
}  
}  
mLocationRequest = LocationRequest.create()  
.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)  
.setInterval(10 * 1000) // 10 seconds, in milliseconds  
.setFastestInterval(5 * 1000); // 2 second, in milliseconds
```



```
LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,  
mLocationRequest, this);
```

```
    Button p1_button = (Button) findViewById(R.id.btndemand);  
    p1_button.setText("Man.");  
    return;
```

Όπως βλέπουμε και στο παραπάνω κώδικα αν πατήσουμε το κουμπί και είναι Auto. Τότε κάνει το flag demand 1 πράγμα που υποδεικνύει ότι είμαστε σε Auto. Mode μετά κάνει το έλεγχο δικαιωμάτων , στην συνέχεια δημιουργώ το Location Request όπου θα είναι αυτό που θα καθορίσω το χρόνο ανανέωσης και τέλος βλέπουμε τον λόγο όπου ήθελε δικαιώματα το LocationServices.. όπου αυτό είναι το τελικό βήμα κάτι σαν build request αυτό θα μας βοηθήσει στο να καθορίσουμε βασικά πράγματα που θα τρέξουμε τις απαντήσεις και χρόνο ανανέωσης μέσω των arguments.

```
mLocationRequest = null;
```

```
LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,  
this);
```

```
flagondemand = 2;
```

```
Button p1_button = (Button) findViewById(R.id.btndemand);  
p1_button.setText("Aut.");
```

```
return;
```

Εξίσου σημαντικό και αυτό αφού πατήσουμε Man. Σημαίνει ότι τελειώσαμε; Όχι φυσικά πρέπει να σταματήσουμε τα Location updates και σε αυτή την πτυχιακή λόγο δημιουργίας να ενημερώσουμε την μεταβλητή flag on demand.

```
public void checkdemand(View v)
```

Αυτή η μέθοδος αφορά το Man. mode.

```
super.onConfigurationChanged(newConfig);
```

```
if (flagondemand == 2) {
```

```
    doublerotation = doublerotation + 1;
```

```
    if (doublerotation == 2) {
```

```
        final int camefrommeeting = 0; //not come from meeting activity
```

```
        updatemanual(camefrommeeting);
```

```
        doublerotation = 0;
```

```
    }
```

```
}
```

Πρώτον από όλα ελέγχουμε αν είμαστε Man Mode. Μέσω της μεταβλητής *flagondemand* που συνδέσαμε στο κουμπί μετέπειτα για κάθε γύρισμα της οθόνης αυξάνουμε μια άλλη μεταβλητή όπου στα 2 γυρίσματα κάνει update μπαίνει στην μέθοδο *updatemanual*.

private void updatemanual

Τι βλέπουμε σε αυτήν την μέθοδο:

1. Πρώτα βλέπουμε τον έλεγχο δικαιωμάτων όπου αναγκαστικά μπαίνει πριν κάθε request
2. Δημιουργία του location Request.
3. remove();

```
Navigation_ok=0;
```

```
if_fine_then_search=0;//for navigation panel if you refresh then search again
```

```
Location location =
```

```
LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
```

```
handleNewLocation(location);
```

Στην remove που υπάρχει παρακάτω κάνουμε διαγραφή πινάκων και χάρτη ,εν συνεχεία ζητάμε την νέα θέση όπου καλούμε την handleNewLocation για να διαχειριστή τα νέα δεδομένα και σημαντικότερα την νέα θέση.

4. Μετά έχω τοποθετήσει δυο if όπου σε αυτές καθορίζω αν έχω έρθει από το ξένο activity ώστε να σχεδιάσω την διαδρομή , πολύ σημαντικό είναι να αναφερθεί για να σχεδιάσουμε την διαδρομή χρειαζόμαστε και την παρούσα θέση μας

```
if (camefrommeeting == 0)
```

```
if (setRoute_clear_button_state == 1)
```

```
setroute.setRoute(PlaceAndIntrests.this);
```

```
if (camefrommeeting == 1) {
```

```
Handler handler = new Handler();
```

```
handler.postDelayed(new Runnable() {
```

```
@Override
```

```
public void run() {
```

```
setroute.setRoutefromMeeting(PlaceAndIntrests.this);
```

```
}
```

```
}, 2 * 2000);
```

```
}
```

Ο handler έχει μπει για λόγους ασφάλειας να σχεδιάσει ύστερα από κάποια λεπτά διότι το γρήγορο δεν προλαβαίνει να ενημερώσει την θέση και κόλα.

5. if (!mGoogleApiClient.isConnected()) {

```
return;
```

```
}
```

```
LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, PlaceAndIntrests.this);
```

```
mLocationRequest = null;
```

εάν δεν έχει συνδεθεί ο client τότε δεν έχει νόημα να συνεχίσουμε και απλά γυρνάμε επίσης δεν ξεχνάμε αφού είμαστε σε Map. Mode πρέπει να διαγράψουμε το αίτημα ώστε να μην συνεχίσει να επικοινωνεί αθέμιτα.

private void remove()

Τι βλέπουμε σε αυτήν την μέθοδο:

```
typeofmark.removeAll(typeofmark);
```

```
PlacesName.removeAll(PlacesName);
```

```
PlacesId.removeAll(PlacesId);
```

```
mMap.clear();
```

Την διαγραφή προηγούμενων τιμών ώστε να μπου οι νέες και την διαγραφή παλαιών σημείων του χάρτη.

```
private void SpinToWin()
```

Αυτή η μέθοδος αφορά το Spinner.

```
this.arraySpinner = new String[]{  
    "place/intrests", "Route Only", "See Meetings", "Exit"  
};
```

```
s = (Spinner) findViewById(R.id.SpinToWin);
```

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
    android.R.layout.simple_spinner_item, arraySpinner);
```

```
s.setAdapter(adapter);
```

```
mainlayout = (LinearLayout) this.findViewById(R.id.twolayout);
```

Όπου αυτά είναι τα βασικά στοιχεία για να έχουμε ένα spinner με ένα menu

```
s.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> parent, View view, int position, long  
    id) {
```

```
        String selectedItem = parent.getItemAtPosition(position).toString();
```

```
if (selectedItem.equals("place/intrests")) {
```

```
    mainlayout.setVisibility(LinearLayout.VISIBLE); // do your stuff
```

```
    } else {
```

```
        mainlayout.setVisibility(LinearLayout.GONE);
```

```
        if (selectedItem.equals("Route Only")) {
```

```
            Intent intent = new Intent(PlaceAndIntrests.this, RouteOnly.class);
```

```
.....
```

```
    }
```

```
    } // to close the onItemClick
```

```
    public void onNothingSelected(AdapterView<?> parent) {
```

```
    }
```

```
});
```

Και με αυτόν τρόπο δημιουργούμε έναν Listener όπου αναλόγως τι έχουμε επιλέξει η θα αλλάξει activity η θα κάνει exit.

```
public void onMapReady(GoogleMap googleMap) {
```

```
    mMap = googleMap;
```

```
    MarkerListener ml = new MarkerListener();
```

```
    ml.setActivityUi(this);
```

```
    ml.Markerlistener();
```

```
}
```

Αυτή η μέθοδος δημιουργείται αυτόματα όταν ο χάρτης είναι έτοιμος και είναι το κατάλληλο σημείο για να δημιουργήσω την κλάση Marker Listener.

private void handleNewLocation

Αυτή η μέθοδος αφορά το τι θα γίνει άμα αλλάξει η θέση όπως βλέπουμε και στο κώδικα περνάμε την μεταβλητή της θέσης όπου περιέχει συντεταγμένες .Τι θα γίνει όταν υπάρχει ανανέωση θέσης αλλά έχουμε ειδή μια διαδρομή;;;

```
if (setRoute_clear_button_state == 1) {  
    SetRoute setroute = new SetRoute();  
    setroute.setRoute(this);  
}
```

Με αυτό τον τρόπο αποφεύγουμε και την διαγραφή της διαδρομής απλώς δημιουργώντας μια νέα με τα νέα στοιχεία της δικιάς μας θέσης.

```
GetYourPlace gyp = new GetYourPlace();
```

```
gyp.setActivityUi(this);
```

```
String s = gyp.getCompleteAddressString(currentLatitude, currentLongitude);
```

Αυτό αφορά το Vicinity της δικιάς μας θέσης έχω δημιουργήσει μια κλάση όπου θα ενημερώνει αυτό, επεξήγηση παρακάτω.

```
MarkerOptions options = new MarkerOptions()
```

```
.title(s)
```

```
.icon(BitmapDescriptorFactory.fromResource(R.drawable.user))
```

```
.position(latLng);
```

```
mMap.addMarker(options);
```

```
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(latLng, 15));
```

Όταν δημιουργείτε νέα θέση πρέπει να την εμφανίσω με τον παραπάνω κώδικα παρουσιάζετε ο τρόπος δημιουργίας θέσης και εστίασης κάμερας.

Τελευταίο και σημαντικό κομμάτι είναι οι ενημέρωση σημείων ενδιαφέροντος που γίνεται με αυτήν την κλάση.

```
final FindPlaces fp = new FindPlaces();
```

```
existisnextpage = 1;
```

```
fp.updatePlaces();
```

```
public void onConnected(@Nullable Bundle bundle) {
```

```
    Button p1_button = (Button) findViewById(R.id.btndemand);
```

```
    p1_button.setText("Aut.");
```

```
    flagondemand = 2;
```

```
}
```

Αυτή η μέθοδος αφορά τι γίνεται όταν συνδεόμαστε και εδώ πέρα έχω ορίσει ότι απλά τοποθετώ το button σε Man Mode.

```

public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
    if (connectionResult.hasResolution()) {
    ....
    }
}

```

```

protected void onResume() {

    super.onResume();
    // setUpMapIfNeeded();
    mGoogleApiClient.connect();
}

```

```

protected void onPause() {
    super.onPause();
    if (mGoogleApiClient.isConnected()) {

```

```

LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
this);
    mGoogleApiClient.disconnect();
    }
}

```

Επίσης και αυτές είναι χρήσιμες μέθοδοι όπου καθορίζουν τι γίνεται σε περίπτωση pause resume και connection failed.

Τελειώνοντας αυτή την κλάση παρουσιάζεται και μια σημαντική μέθοδος,

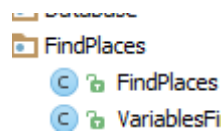
```

public void onLocationChanged(Location location) {
    if (flagondemand == 1) {
        remove();
        handleNewLocation(location);
        if (setRoute_clear_button_state == 1) {
            SetRoute setroute = new SetRoute();
            setroute.setRoute(this);
        }
    }
}

```

αυτή η μέθοδος τρέχει αυτόματα σε περίπτωση αλλαγής θέσης και την έχω βάλει για το Aut. Mode μονό όπου θα πάει στην μέθοδο διαχείρισης νέας θέσης και θα ελέγξει αν έχω μια διαδρομή ώστε να την ξανασχεδιάσει.

Class FindPlaces



```
package FindPlaces;
import android.os.AsyncTask;
import android.os.Handler;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.tolis.maps3856.R;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import static FindPlaces.VariablesFindPlaces.Places_LongLatitubee;
import static FindPlaces.VariablesFindPlaces.Places_Vicinity;
import static FindPlaces.VariablesFindPlaces.Places_photo_reference;
import static FindPlaces.VariablesFindPlaces.currentLatitude;
import static FindPlaces.VariablesFindPlaces.currentLongitude;
import static FindPlaces.VariablesFindPlaces.existsnextpage;
import static FindPlaces.VariablesFindPlaces.if_fine_then_search;
import static FindPlaces.VariablesFindPlaces.mMap;
import static SetMeetings.SetMeetingsVariables.PlacesId;
import static SetMeetings.SetMeetingsVariables.PlacesName;
import static SetMeetings.SetMeetingsVariables.typeofmark;

public class FindPlaces extends FragmentActivity {

    public int UserIcon = R.drawable.user, CinemaIcon = R.drawable.cinemas,
        GasStationIcon = R.drawable.autogas, ShopIcon = R.drawable.spanosl,
    Ather_Icon = R.drawable.othericon,
        CinemaIconl = R.drawable.cinema, GasStationIconl = R.drawable.autogas,
    ShopIconl = R.drawable.spanos;
    private String placesSearchStrnew;
    public MarkerOptions[] places;
    private final int MAX_PLACES = 60;
    public Marker[] placeMarkers;
```

```

public void updatePlaces() {
    placeMarkers = new Marker[MAX_PLACES];
    // Places_LongLatitude.removeAll(Places_LongLatitude);
    if_fine_then_search = 0;
    //build places query string
    placesSearchStrnew =
"https://maps.googleapis.com/maps/api/place/nearbysearch/" +
    "json?location=" + currentLatitude + "," + currentLongitude +
    "&radius=4000&sensor=true" +
    "&types=movie_theater|gas_station|grocery_or_supermarket|bank" +
    "&key=....."; //ADD KEY

    for (int i = 1; i <= 3; i++) {
        if (existisnextpage == 1) {
            Handler handler = new Handler();
            handler.postDelayed(new Runnable() {
                @Override
                public void run() {
                    new GetPlaces().execute(placesSearchStrnew);
                }
            }, i * 2000);
            //why i* bccs its for and you do not want to start 3 handler at same time
        }
    }
    //if in the future google add more than 60 points or 4 pages then do what have be
    done if exist 3 page
    Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            if (existisnextpage == 1) {
                if_fine_then_search = 1; //for navigation panel
                existisnextpage = 2;
            }
        }
    }, 10000);
}

```

```

public class GetPlaces extends AsyncTask<String, Void, String> {

    @Override
    protected String doInBackground(String... placesURL) {
        //fetch places
        // setContentView(R.layout.placeandintrests);
        StringBuilder placesBuilder = new StringBuilder();
        for (String placeSearchURL : placesURL) {
            try {

```

```

URL requestUrl = new URL(placeSearchURL);
URLConnection connection = (URLConnection)
requestUrl.openConnection();
connection.setRequestMethod("GET");
connection.connect();
int responseCode = connection.getResponseCode();

if (responseCode == HttpURLConnection.HTTP_OK) {

    BufferedReader reader = null;

    InputStream inputStream = connection.getInputStream();
    if (inputStream == null) {
        return "";
    }
    reader = new BufferedReader(new InputStreamReader(inputStream));

    String line;
    while ((line = reader.readLine()) != null) {

        placesBuilder.append(line + "\n");
    }

    if (placesBuilder.length() == 0) {
        return "";
    }
    connection.disconnect();
    Log.d("test", placesBuilder.toString());
} else {
    Log.i("test", "Unsuccessful HTTP Response Code: " +
responseCode);
}
} catch (MalformedURLException e) {
    Log.e("test", "Error processing Places API URL", e);
} catch (IOException e) {
    Log.e("test", "Error connecting to Places API", e);
}
}

try {
    if (existsnextpage == 1) {
        JSONObject responseObject = null;
        responseObject = new JSONObject(placesBuilder.toString());
        String next_page_token = responseObject.getString("next_page_token");
        if (!next_page_token.equals("")) {

            placesSearchStrnew =
"https://maps.googleapis.com/maps/api/place/nearbysearch/" +
                "json?location=" + currentLatitude + "," + currentLongitude +

```



```

"&types=movie_theater|gas_station|grocery_or_supermarket|bank" +
    "&key=..." + "&pagetoken=" + next_page_token;
    Log.d("testa", placesSearchStrnew);
} else {

    existisnextpage = 2;
}
}
} catch (JSONException e) {

    existisnextpage = 2;
    if_fine_then_search = 1;//for navigation panel
    e.printStackTrace();

}
return placesBuilder.toString();
}

//process data retrieved from doInBackground
protected void onPostExecute(String result) {
    //parse place data returned from Google Places
    //remove existing markers
    if (result.equals(""))
        return;

    try {
        //parse JSON

        //create JSONObject, pass string returned from doInBackground
        JSONObject resultObject = new JSONObject(result);
        int ifcinematype = 0;
        //get "results" array
        JSONArray placesArray = resultObject.getJSONArray("results");
        //marker options for each place returned
        places = new MarkerOptions[placesArray.length()];

        //loop through places
        for (int p = 0; p < placesArray.length(); p++) {
            //parse each place
            //if any values are missing we won't show the marker
            boolean missingValue = false;

            LatLng placeLL = null;
            String placeName = "";
            String vicinity = "";
            String Open = "";

            typeofmark.add(0);

```

```

int currIcon = Ather_Icon;
try {

    //attempt to retrieve place data values
    missingValue = false;
    //get place at this index
    JSONObject placeObject = placesArray.getJSONObject(p);

    //get location section
    JSONObject loc = placeObject.getJSONObject("geometry")
        .getJSONObject("location");
    //read lat lng
    placeLL = new LatLng(Double.valueOf(loc.getString("lat")),
        Double.valueOf(loc.getString("lng")));

    //get types
    JSONArray types = placeObject.getJSONArray("types");
    //loop through types
    for (int t = 0; t < types.length(); t++) {
        //what type is it
        try {
            JSONObject Open2 =
placeObject.getJSONObject("opening_hours");
            // placeLL = new LatLng(Double.valueOf(loc.getString("lat")),
            Boolean Open33 = new
Boolean(Open2.getString("open_now"));

            if (Open33)
                Open = "O";
        } catch (JSONException e) {
            Open = "C";
        }

        String thisType = types.get(t).toString();

        //check for particular types - set icons
        if (thisType.contains("gas_station")) {
            typeofmark.remove(typeofmark.size() - 1); //if it getinside here it
will change last value of type
            typeofmark.add(1);
            currIcon = GasStationIcon;
            if (Open == "O")
                currIcon = GasStationIcon;

            break;
        } else if (thisType.contains("movie_theater")) {
            typeofmark.remove(typeofmark.size() - 1);
            typeofmark.add(2);
            currIcon = CinemaIcon;

```

```

        if (Open == "O")
            currIcon = CinemaIcon;
            ifcinematype = 1;
            break;

    } else if (thisType.contains("grocery_or_supermarket")) {
        typeofmark.remove(typeofmark.size() - 1);
        typeofmark.add(3);
        currIcon = ShopIcon;
        if (Open == "O")
            currIcon = ShopIcon;

        break;
    }
}
//vicinity
vicinity = placeObject.getString("vicinity");

//get id and name of the place to find later...

PlacesId.add(placeObject.getString("place_id"));
//name
placeName = placeObject.getString("name");
PlacesName.add(placeName);
String lname = String.valueOf(placeLL);
lname = lname.replace("lat", "");
lname = lname.replace("/", "");
lname = lname.replace("lng", "");
lname = lname.replace(":", "");
lname = lname.replace("(", "");
lname = lname.replace(")", "");
Log.v("PLACES", lname);

Places_LongLatitubee.add(lname);
Places_Vicinity.add(vicinity);
try {
    JSONArray photoArray = placeObject.getJSONArray("photos");
    JSONObject photoObject = photoArray.getJSONObject(0);
    Places_photo_reference.add(
photoObject.getString("photo_reference"));
    Log.v("photoreference", String.valueOf(Places_photo_reference));
    Log.v("photoreference2",
String.valueOf(Places_photo_reference.size()));
}
catch (JSONException e)
{
    Places_photo_reference.add("null");
}

```

```

        Log.v("photoreference", String.valueOf(Places_photo_reference));
        Log.v("photoreference2",
String.valueOf(Places_photo_reference.size()));
    }
    // Log.v("PLACES", String.valueOf(placeLL));
    // placeName=placeName+Open;
} catch (JSONException jse) {
    Log.v("PLACES", "missing value");
    missingValue = true;
    jse.printStackTrace();
}
//if values missing we don't display
if (missingValue) places[p] = null;
else if (ifcinematype == 0) {
    places[p] = new MarkerOptions()
        .position(placeLL)
        .title(placeName)
        .icon(BitmapDescriptorFactory.fromResource(currIcon))
        .snippet(vicinity);
}
if (ifcinematype == 1) {
    ifcinematype = 0;
    places[p] = new MarkerOptions()
        .alpha(2)
        .position(placeLL)
        .title(placeName)
        .icon(BitmapDescriptorFactory.fromResource(currIcon))
        .snippet(vicinity);
}
}

}
} catch (Exception e) {
    e.printStackTrace();
}
}
if (places != null && placeMarkers != null) {
    for (int p = 0; p < places.length && p < placeMarkers.length; p++) {
        //will be null if a value was missing
        if (places[p] != null)
            placeMarkers[p] = mMap.addMarker(places[p]);
    }
}

}
}
}
}
}

```

Σε αυτήν την κλάση θα δούμε πως δημιουργούμε google query, πώς να βρούμε τα σημεία ενδιαφέροντος , πως να τα αποκωδικοποιήσουμε, την εμφάνιση τους και αλλά

διάφορα.

```
public int UserIcon = R.drawable.user, CinemaIcon = R.drawable.cinemat,  
    GasStationIcon = R.drawable.autogasl, ShopIcon = R.drawable.spanosl,  
Ather_Icon = R.drawable.othericon,  
    CinemaIconl = R.drawable.cinema, GasStationIconl = R.drawable.autogas,  
ShopIconl = R.drawable.spanos;  
private String placesSearchStrnew;  
public MarkerOptions[] places;  
private final int MAX_PLACES = 60;  
public Marker[] placeMarkers;
```

Οι μεταβλητές αυτής της κλάσης συγκεκριμένα οι πρώτες αναφέρονται σε εικονίδια, η δεύτερη **placesSearchStrnew** αναφέρεται στο string του επόμενου query , στην συνέχεια βλέπουμε μεταβλητές σχετικά με τα σημεία.

```
if_fine_then_search = 0;
```

Η μεταβλητή για να ελέγξουμε αν κατεβήκαν όλα τα σημεία.

```
placesSearchStrnew =  
"https://maps.googleapis.com/maps/api/place/nearbysearch/" +  
    "json?location=" + currentLatitude + "," + currentLongitude +  
    "&radius=4000&sensor=true" +  
    "&types=movie_theater|gas_station|grocery_or_supermarket|bank" +  
    "&key=...."; //ADD KEY
```

Το string του google query για αναζήτηση σημείων σε απόσταση 4 χιλιομέτρων για βενζινάδικα μαγαζιά σινεμά και τράπεζες.

```
for (int i = 1; i <= 3; i++) {  
    if (existisnextpage == 1) {  
        Handler handler = new Handler();  
        handler.postDelayed(new Runnable() {  
            @Override  
            public void run() {  
                new GetPlaces().execute(placesSearchStrnew);  
            }  
        }, i * 2000);  
    } //why i* bccs its for and you do not want to start 3 handler at same time  
}
```

Ιδιαίτερη σημασία πρέπει να δώσουμε στην επανάληψη με τον handler συγκεκριμένα στο σημείο $I * 2000$, όταν κάνουμε μια επανάληψη οι handlers μπαίνουν ο ένας μετά τον άλλο αλλά ο χρόνος που θα τοποθετηθούν είναι ο χρόνος της επανάληψης κάτι πολύ μικρό με την υπολογιστική ισχύ για αυτό έβαλα I, μπορεί να μην είναι ακριβώς ο χρόνος του πρώτου με διάφορα 2 δευτερόλεπτα από τον δεύτερο αλλά είναι υπερβολικά κοντά. Και γιατί το κάνουμε αυτό ο λόγος είναι ο έξης : η google δεν στέλνει όλες της πληροφορίες με την μια ανά κάθε σελίδα στέλνει 20 σημεία με μέγιστο 3 σελίδες για αυτό τον λόγο δεν είναι δυνατό κάνει να ξέρεις το επόμενο

query αν δεν έρθει το πρώτο και ο λόγος των 2 δευτερόλεπτων είναι ένας πολύ μεγάλος εφικτός χρόνος για πάρουμε την πρώτη σελίδα με της πληροφορίες και το google next query.

```

    "html_attributions" : [],
    "next_page_token": "CqQCFAEAAHnG-
    CZu57RI59Bx9MUpInUpwC1rmIIIfi1ZFrc3LfRRb0jBkscMHIK1gOkK56Eyw1NCffQXfd8XfEuv1BYHADzgpugj6g_
    Wb1PLj1c5b1BrI0De5VIE89Lmx9sQcFpkimUzLAoavJjsRFQ1_-rTqxMRFsEs8aroYDURQ2fxBfr7Ks8Ku6TQsoe02QC
    Tehf73sgrdsBoUmebEqBU_TA02aN-Oqo77bbSdlwU",
    "results" : [
      {
        "geometry" : {
          "location" : {
            "lat" : 36.4352612,
            "lng" : 28.224676
          },
          "viewport" : {
            "northeast" : {
              "lat" : 36.4353813,
              "lng" : 28.2248508000001
            },
            "southwest" : {
              "lat" : 36.4349009,
              "lng" : 28.2241516
            }
          }
        },
        "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/shopping-71.png",
        "id" : "378bad7ec987f1177cee21d534b04e68621ee988",
        "name" : "SUPERMARKET SPANOS, J. CH., GROUP OF COMPA",
        "opening_hours" : {
          "open_now" : false,
          "weekday_text" : []
        },
        "photos" : [
          {
            "height" : 2340,

```

//if in the future google add more than 60 points or 4 pages then do what have be done if exist 3 page

```

Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {
        if (existisnextpage == 1) {
            if_fine_then_search = 1; //for navigation panel
            existisnextpage = 2;
        }
    }
}, 10000);

```

Αν στο μέλλον μπουν πάνω από 3 σελίδες για να λειτουργεί η εφαρμογή έβαλα ένα stop.

```

protected String doInBackground(String... placesURL) {

    StringBuilder placesBuilder = new StringBuilder();
    for (String placeSearchURL : placesURL) {
        try {
            .....
        }
    }
}

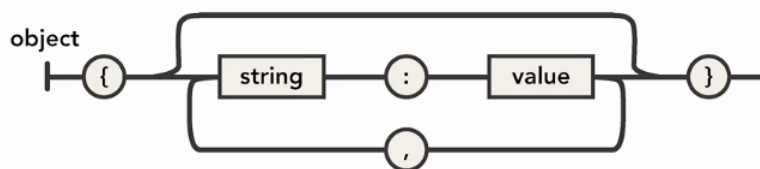
```

Αυτό το κομμάτι αφορά τον τρόπο σύνδεσης και το τι μέτρα πρέπει να μειωθούν αν δεν υπάρχει μια.

```
try {
    if (existisnextpage == 1) {
        JSONObject resultObject = null;
        resultObject = new JSONObject(placesBuilder.toString());
        String next_page_token = resultObject.getString("next_page_token");
        if (!next_page_token.equals("")) {

            placesSearchStrnew =
                "https://maps.googleapis.com/maps/api/place/nearbysearch/" +
                "json?location=" + currentLatitude + "," + currentLongitude +
                "&types=movie_theater|gas_station|grocery_or_supermarket|bank" +
                "&key=..." + "&pagetoken=" + next_page_token;
            Log.d("testa", placesSearchStrnew);
        } else {
            existisnextpage = 2;
        }
    }
} catch (JSONException e) {
    existisnextpage = 2;
    if_fine_then_search = 1; //for navigation panel
    e.printStackTrace();
}
return placesBuilder.toString();
}
```

JSON objects



Όπως βλέπουμε αυτήν και την παραπάνω εικόνα το **next_page_token** αποτελεί ένα μέλος του json object και το περνούμε με την μέθοδο

```
JSONObject resultObject = null;
resultObject = new JSONObject(placesBuilder.toString());
Μετατρέπουμε το string δεδομένων σε json αφού αυτό είναι και στην συνέχεια
String next_page_token = resultObject.getString("next_page_token");
```

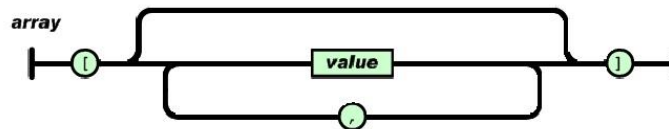
Σε περίπτωση που δεν υπάρχει σημαίνει ότι είμαστε στην τελευταία σελίδα και θα γυρίσει ένα error όπου με την catch θα αλλάξουμε τις μεταβλητές

```
existisnextpage = 2;
if_fine_then_search = 1;
για να μάθουμε ότι η αναζήτηση ολοκληρώθηκε.
```

Syntax of JSON

- **Array**

- An *array* is an ordered collection of values.
- An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).



Και επειδή οι τιμές του κάθε σημείου πρέπει να είναι ξεχωριστές η google τα τοποθέτησε σε Json Array.

```
protected void onPostExecute(String result) {
```

Σε αυτήν μεθοδο βλέπουμε:

```
JSONObject resultObject = new JSONObject(result);  
JSONArray placesArray = resultObject.getJSONArray("results");
```

Πρώτα παίρνω το String σε Json object .

```
for (int p = 0; p < placesArray.length(); p++) {
```

Όπως αναφέρθηκα ας πούμε ότι έχουμε 20 σημεία τότε θα πρέπει να πάρουμε το μέγεθος του Json Array και να κάνουμε επανάληψη για να βρούμε τις πληροφορίες και των 20 σημείων .

```
LatLng placeLL = null;  
String placeName = "";  
String vicinity = "";  
String Open = "";
```

Σε κάθε επανάληψη πρέπει να έχω και διαφορετικό string για αυτό αναθέτω τα string ξανά.

```
typeofmark.add(0);
```

```
int currIcon = Ather_Icon;  
try {  
    missingValue = false;  
    JSONObject placeObject = placesArray.getJSONObject(p);  
    JSONObject loc = placeObject.getJSONObject("geometry")
```



```

        .getJSONObject("location");
placeLL = new LatLng(Double.valueOf(loc.getString("lat")),
    Double.valueOf(loc.getString("lng")));

```

Από εδώ παίρνω τις πληροφορίες long και lat.

```

    {
        "geometry" : {
            "location" : {
                "lat" : 36.4352612,
                "lng" : 28.224676
            },

```

```

JSONArray types = placeObject.getJSONArray("types");
for (int t = 0; t < types.length(); t++) {
    try {
        JSONObject Open2 =
placeObject.getJSONObject("opening_hours");
        Boolean Open33 = new Boolean(Open2.getString("open_now"));
        if (Open33)
            Open = "O";
    } catch (JSONException e) {
        Open = "C";
    }
    String thisType = types.get(t).toString();
    if (thisType.contains("gas_station")) {
        typeofmark.remove(typeofmark.size() - 1);
        typeofmark.add(1);
        currIcon = GasStationIcon;
        if (Open == "O")
            currIcon = GasStationIcon;
        break;
    } else if (thisType.contains("movie_theater")) {
        typeofmark.remove(typeofmark.size() - 1);
        typeofmark.add(2);
        currIcon = CinemaIcon;
        if (Open == "O")
            currIcon = CinemaIcon;
        ifcinematype = 1;
        break;
    } else if (thisType.contains("grocery_or_supermarket")) {
        typeofmark.remove(typeofmark.size() - 1);
        typeofmark.add(3);
        currIcon = ShopIcon;
        if (Open == "O")
            currIcon = ShopIcon;

        break;
    }
}

```

```
}
```

Από εδώ θέτω τις εικόνες του κάθε σημείου σε περίπτωση που δεν υπάρχει έχω θέση ένα τριγωνάκι κόκκινο σε περίπτωση όμως που υπάρχει τότε ανάλογος αν η τιμή είναι open true η false τότε μπαίνει κόκκινο καλάθι δηλαδή κλειστό μαύρο ανοικτό.

```
"geometry" : {  
  "location" : {  
    "lat" : 36.4352612,  
    "lng" : 28.224676  
  },  
  "viewport" : {  
    "northeast" : {  
      "lat" : 36.4353813,  
      "lng" : 28.22485080000001  
    },  
    "southwest" : {  
      "lat" : 36.4349009,  
      "lng" : 28.2241516  
    }  
  }  
},  
"icon" : "https://maps.gstatic.com/ma  
"id" : "378bad7ec987f1177cee21d534b04  
"name" : "SUPERMARKET SPANOS, J. CH.,  
"opening hours" : {  
  "open_now" : false,  
  "weekday_text" : []  
},
```

```
//vicinity
```

```
vicinity = placeObject.getString("vicinity");
```

```
//get id and name of the place to find later...
```

```
PlacesId.add(placeObject.getString("place_id"));
```

```
//name
```

```
placeName = placeObject.getString("name");
```

```
PlacesName.add(placeName);
```

```
String lname = String.valueOf(placeLL);
```

```
lname = lname.replace("lat", "");
```

```
lname = lname.replace("/", "");
```

```
lname = lname.replace("lng", "");
```

```
lname = lname.replace(":", "");
```

```
lname = lname.replace("(", "");
```

```
lname = lname.replace(")", "");
```

```
Log.v("PLACES", lname);
```

```
"location" : {  
  "lat" : 36.4352612,  
  "lng" : 28.224676  
},
```

Καθαρίζω τα υπόλοιπα και κρατώ τους αριθμούς και εύκολη αναζήτηση.

```
Places_LongLatitude.add(lname);
Places_Vicinity.add(vicinity);
```

Ένα σημαντικό κομμάτι εδώ παίρνω το όνομα του σημείου και το τοποθετώ στον πίνακα λίστα παρομοίως και παρακάτω.

```
try {
    JSONArray photoArray = placeObject.getJSONArray("photos");
    JSONObject photoObject = photoArray.getJSONObject(0);
    Places_photo_reference.add(
photoObject.getString("photo_reference"));
String.valueOf(Places_photo_reference.size());
}
catch (JSONException e)
{
    Places_photo_reference.add("null");
String.valueOf(Places_photo_reference.size());
}
```

Με αυτόν τον τρόπο αποθηκεύω το κάθε “link” της εικόνας του σημείου σε πίνακα αν δεν υπάρχει θέτω null

```
    "geometry" : {
      "location" : {
        "lat" : 36.4352612,
        "lng" : 28.224676
      },
      "viewport" : {
        "northeast" : {
          "lat" : 36.4353813,
          "lng" : 28.22485080000001
        },
        "southwest" : {
          "lat" : 36.4349009,
          "lng" : 28.2241516
        }
      }
    },
    "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/shopping-71.png",
    "id" : "378bad7ec987f1177cee21d534b04e68621ee988",
    "name" : "SUPERMARKET SPANOS, J. CH., GROUP OF COMPA",
    "opening_hours" : {
      "open_now" : false,
      "weekday_text" : []
    },
    "photos" : [
      {
        "height" : 2340,
        "html_attributions" : [
          "\u003ca href=\u0022https://maps.google.com/maps/contrib/115853780114203717194/photos\u0022\u003e\u003c\/a\u003e"
        ],
        "photo_reference" : "CoQ8dwAAAAHfoLypb1lhDJ-5yZ7KhCvPJQr-jUVBuhC0eb7knaUV3YvUWwMI5V-
jj0sjuUaU7H1djkovCG81sRbnPg68UWje8ZLZCqRTQnd4uk8p0G4C1VTNZrmdsX2BwXa8jLeeFjYHK6tDhJk1zUCurD3a19Y3r7rhyu4TW
"width" : 4160
      }
    ],
    "place_id" : "ChIJj11cRJRh1RQRNr6CLc_OCQI",
  } catch (JSONException jse) {
    missingValue = true;
    jse.printStackTrace();
  }
}
```

Δεν υπάρχει πλήρες σημείο.

```
if (missingValue) places[p] = null;
else if (ifcinematype == 0) {
```

```

places[p] = new MarkerOptions()
    .position(placeLL)
    .title(placeName)
    .icon(BitmapDescriptorFactory.fromResource(currIcon))
    .snippet(vicinity);
}
if (ifcinematype == 1) {
    ifcinematype = 0;
    places[p] = new MarkerOptions()
        .alpha(2)
        .position(placeLL)
        .title(placeName)
        .icon(BitmapDescriptorFactory.fromResource(currIcon))
        .snippet(vicinity);
}

```

Εδώ πέρα για λογούς ευκολίας στην αρχή είχα βάλει αν το είδος είναι σινεμά να βγάζει διαφορετικό `alpha(2)` αλλά αυτό μπορεί να γίνει και με πίνακα. Όπως βλέπουμε σε αυτό το σημείο δημιουργούμε κάνουμε `build` το σημείο με το όνομα θέση περιοχή.

```

if (places != null && placeMarkers != null) {
    for (int p = 0; p < places.length && p < placeMarkers.length; p++) {
        //will be null if a value was missing
        if (places[p] != null)
            placeMarkers[p] = mMap.addMarker(places[p]);
    }
}

```

Εδώ πέρα θέτουμε στο χάρτη τα σημεία αλλά και σε έναν πίνακα όπου δεν το χρειάστηκα ιδιαίτερα σε αυτήν την πτυχιακή.

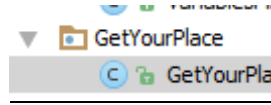
Class VariablesFindPlaces

```
package FindPlaces;
import com.google.android.gms.maps.GoogleMap;
import java.util.ArrayList;

public class VariablesFindPlaces {
    public static int existisnextpage = 1;
    public static double currentLatitude;
    public static double currentLongitude;
    public static GoogleMap mMap;
    //for navi panel
    public static ArrayList<String> Places_LongLatitubee = new
    ArrayList<String>();//use for navigation panel in places and intrests activity
    public static ArrayList<String> Places_Vicinity = new ArrayList<String>();//use
for navigation panel in places and intrests activity
    public static int if_fine_then_search = 0;
    public static ArrayList<Integer> Duration = new ArrayList<Integer>();//use for
navigation panel in places and intrests activity
    public static ArrayList<String> Places_photo_reference = new
    ArrayList<String>();//use for place foto in the right navi panel
}
```

Οι Static μεταβλητές που χρησιμεύουν σε πολλές κλάσεις προσπάθησα να τις απομονώσω σε μια ξεχωριστή.

Class GetYourPlace



```
package GetYourPlace;
import android.app.Activity;
import android.location.Address;
import android.location.Geocoder;
import java.util.List;
import java.util.Locale;

public class GetYourPlace {
    private Activity activity;

    public void setActivityUi(Activity activity) {
        this.activity = activity;
    }

    public String getCompleteAddressString(double LATITUDE, double
LONGITUDE) {
        String strAdd = "";
        Geocoder geocoder = new Geocoder(activity, Locale.getDefault());
        try {
            List<Address> addresses = geocoder.getFromLocation(LATITUDE,
LONGITUDE, 1);
            if (addresses != null) {
                String address = addresses.get(0).getAddressLine(0);
                String city = addresses.get(0).getLocality();

                strAdd = address + "," + city;

            } else {
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return strAdd;
    }
}
```

Με αυτήν την κλάση παίρνω την διεύθυνση και την πόλη στην οποία είμαι με την βοήθεια του Geocoder.

Class MarkerListener

```
package Listeners;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AlertDialog;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.Marker;
import com.tolis.maps3856.R;
import static FindPlaces.VariablesFindPlaces.mMap;
import static com.tolis.maps3856.PlaceAndIntrests.setRoute_clear_button_state;
```

```
public class MarkerListener {

    private Activity activity;

    public void setActivityUi(Activity activity) {
        this.activity = activity;
    }

    public void Markerlistener() {
        mMap.setOnMarkerClickListener(new GoogleMap.OnMarkerClickListener() {

            @Override
            public boolean onMarkerClick(final Marker arg0) {
                if (setRoute_clear_button_state == 1) {
                    Toast.makeText(activity, "Please First Clear To Select new Place",
                    Toast.LENGTH_LONG).show();
                    return true;
                }
                float a = mMap.getCameraPosition().zoom;
                CameraUpdate location =
                CameraUpdateFactory.newLatLngZoom(arg0.getPosition(), a);
                mMap.animateCamera(location, 500, null);
                final TextView myAwesomeTextView = (TextView)
                activity.findViewById(R.id.Title);
                final TextView placedime1 = (TextView)
                activity.findViewById(R.id.placedime);
                final TextView myAwesomeTextView2 = (TextView)
                activity.findViewById(R.id.Vicinity);
```

```

activity.runOnUiThread(new Runnable() {
    public void run() {
        myAwesomeTextView.setText(arg0.getTitle());
        placedime1.setText(arg0.getPosition().toString());
        myAwesomeTextView2.setText(arg0.getSnippet());
    }
});
if (arg0.getAlpha() == 2) {

    new AlertDialog.Builder(activity)
        .setMessage("Search Cinema In Google?")
        .setCancelable(false)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://google.com/search?q=" + arg0.getTitle() + " " +
arg0.getSnippet()));
                activity.startActivity(browserIntent);

                dialog.cancel();
            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        }).show();
    }
return true;
}
});
}
}
}
}

```

Αυτήν την κλάση την καλούσαμε όταν δημιουργούσαμε τα σημεία στην handle new location και με αυτήν θέτουμε listener για τα σημεία ενδιαφέροντος τι γίνεται λοιπόν:

Πρώτον εάν έχω ειδή ένα Route μια διαδρομή δεν πρέπει να επιλέγω άλλο σημείο αλλά πρώτα να κάνω clear.

```

if (setRoute_clear_button_state == 1) {
    Toast.makeText(activity, "Please First Clear To Select new Place",
Toast.LENGTH_LONG).show();
    return true;
}

```



```

float a = mMap.getCameraPosition().zoom;
CameraUpdate location = CameraUpdateFactory.newLatLngZoom(arg0.getPosition(),
a);
mMap.animateCamera(location, 500, null);
final TextView myAwesomeTextView = (TextView)
activity.findViewById(R.id.Title);
final TextView placedime1 = (TextView) activity.findViewById(R.id.placedime);
final TextView myAwesomeTextView2 = (TextView)
activity.findViewById(R.id.Vicinity);
activity.runOnUiThread(new Runnable() {
    public void run() {
        myAwesomeTextView.setText(arg0.getTitle());
        placedime1.setText(arg0.getPosition().toString());
        myAwesomeTextView2.setText(arg0.getSnippet());
    }
});

```

Εδώ πέρα κάνουμε zoom την camera στο σημείο ενδιαφέροντος σύμφωνα με το υπάρχον zoom και όχι με preset zoom και στην συνέχεια τοποθετούμε τα στοιχεία στα κατάλληλα text view επίσης δεν ξεχνάμε αφού είμαστε από άλλη κλάση τότε θέλουμε runOnUiThread και το **activity** της κλάσης αυτής.

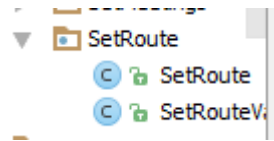
```

if (arg0.getAlpha() == 2) {
    new AlertDialog.Builder(activity)
        .setMessage("Search Cinema In Google?")
        .setCancelable(false)
        .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                Intent browserIntent = new Intent(Intent.ACTION_VIEW,
Uri.parse("http://google.com/search?q=" + arg0.getTitle() + " " +
arg0.getSnippet()));
                activity.startActivity(browserIntent);
                dialog.cancel();
            }
        })
        .setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        })
        .show();
}

```

Και σε περίπτωση που το Alpha είναι 2 τότε σημαίνει ότι πατήσαμε cinema και μας κατευθύνει με ερώτηση αν θέλουμε να ψάξουμε το προγράμματα του στο google.

Class SetRoute



```
package SetRoute;
import android.app.Activity;
import android.app.AlertDialog;
import android.graphics.Color;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import com.tolis.maps3856.R;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;
import Modules.DirectionFinder;
import Modules.DirectionFinderListener;
import Modules.Route;
import static FindPlaces.VariablesFindPlaces.Duration;
import static FindPlaces.VariablesFindPlaces.Places_LongLatitude;
import static FindPlaces.VariablesFindPlaces.currentLatitude;
import static FindPlaces.VariablesFindPlaces.currentLongitude;
import static FindPlaces.VariablesFindPlaces.mMap;
import static SetRoute.SetRouteVariables.Navigation_ok;
import static SetRoute.SetRouteVariables.destinationMarkers;
import static SetRoute.SetRouteVariables.originMarkers;
import static SetRoute.SetRouteVariables.polylinePaths;
import static com.tolis.maps3856.PlaceAndIntrests.setRoute_clear_button_state;

public class SetRoute extends FragmentActivity implements
DirectionFinderListener {

    private Activity activity;
    private int check = 0;
    private AlertDialog progress;

    public void setActivityUi(Activity activity) {
        this.activity = activity;
    }
}
```

```

}

public void clearRoute(Activity act) {

    if (polylinePaths != null) {
        for (Polyline polyline : polylinePaths) {
            polyline.remove();
        }
    }

    final TextView myAwesomeTextView = (TextView)
act.findViewById(R.id.Title);

    final TextView myAwesomeTextView2 = (TextView)
act.findViewById(R.id.Vicinity);

    final TextView myAwesomeTextView3 = (TextView)
act.findViewById(R.id.tvDistance);

    final TextView myAwesomeTextView4 = (TextView)
act.findViewById(R.id.tvDuration);

    //you can't update things on the ui thread from anywhere
    // but the ui thread, you need to use this .Where mAct
    // is the main activity but you have to pass it in a parameter
    // to the other class so it has access to it because it can't
    // be a static reference
    act.runOnUiThread(new Runnable() {
        public void run() {
            myAwesomeTextView.setText("Title");
            myAwesomeTextView2.setText("Vicinity");
            myAwesomeTextView3.setText("0 km");
            myAwesomeTextView4.setText("0 min");
        }
    });
    return;
}

// this.findViewById(android.R.id.content).context
// MyApplication.context
public void setRoute(Activity act) {
    setActivityUi(act);

    final TextView placedime1 = (TextView) act.findViewById(R.id.placedime);
    act.runOnUiThread(new Runnable() {
        public void run() {

            String lname = placedime1.getText().toString();
            // lname=lname.substring(10);
            // lname= lname.substring(0,lname.length()-1);

```

```

lname = lname.replace("lat", "");
lname = lname.replace("/", "");
lname = lname.replace("lng", "");
lname = lname.replace(":", "");
lname = lname.replace("(", "");
lname = lname.replace(")", "");
// placename1.setText(lname.toString());

String s = currentLatitude + "," + currentLongitude;
try {
    new DirectionFinder(SetRoute.this, s, lname).execute();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
}
});
}

```

```

public void setRoutefromNaviPanel(String PlaceLongLat, final String
Placenamegiven, final String Vicinitygiven) {

```

```

    setRoute_clear_button_state = 1;

```

```

    final TextView myAwesomeTextView3 = (TextView)
activity.findViewById(R.id.Vicinity);
    final TextView myAwesomeTextView4 = (TextView)
activity.findViewById(R.id.Title);
    final Button button = (Button) activity.findViewById(R.id.setroute);

```

```

//you can't update things on the ui thread from anywhere
// but the ui thread, you need to use this .Where mAct
// is the main activity but you have to pass it in a parameter
// to the other class so it has access to it because it can't
// be a static reference

```

```

activity.runOnUiThread(new Runnable() {
    public void run() {
        button.setText("Clear");
        myAwesomeTextView3.setText(Vicinitygiven);
        myAwesomeTextView4.setText(Placenamegiven);
    }
}

```

```

});

```

```

String s = currentLatitude + "," + currentLongitude;
try {
    new DirectionFinder(SetRoute.this, s, PlaceLongLat).execute();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}

```

```

    }
}

public void setRoutefromMeeting(Activity act) {
    setActivityUi(act);

    final TextView placedime1 = (TextView) act.findViewById(R.id.placedime);
    act.runOnUiThread(new Runnable() {
        public void run() {

            String lname = placedime1.getText().toString();
            if (currentLongitude == -122.084000) {
                currentLatitude = 36.436285;
                currentLongitude = 28.217539;
                // currentLatitude = 35.319647;
                // currentLongitude = 25.102336;
            }

            String s = currentLatitude + "," + currentLongitude;
            try {
                new DirectionFinder(SetRoute.this, s, lname).execute();
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            }
        }
    });
}

public void findroute(int i_counter) {
    progress = new ProgressDialog(activity);
    progress.setMessage("Searching Nearby Places");
    progress.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    String s = currentLatitude + "," + currentLongitude;
    String name_place = null;
    try {
        check = 1;
        for (int k = i_counter; k < Places_LongLatitubee.size(); k++) {
            name_place = Places_LongLatitubee.get(k);
            name_place = name_place.replace(" ", "");
            new DirectionFinder(SetRoute.this, s, name_place).execute();
            Log.v("PLACES", String.valueOf(Places_LongLatitubee));
        }
        // if(Duration.size()==Places_LongLatitubee.size())

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}
}

```

@Override

```
public void onDirectionFinderStart() {  
  
    if (originMarkers != null) {  
        for (Marker marker : originMarkers) {  
            marker.remove();  
        }  
    }  
  
    if (destinationMarkers != null) {  
        for (Marker marker : destinationMarkers) {  
            marker.remove();  
        }  
    }  
  
    if (polylinePaths != null) {  
        for (Polyline polyline : polylinePaths) {  
            polyline.remove();  
        }  
    }  
}
```

@Override

```
public void onDirectionFinderSuccess(List<Route> routes) {  
  
    polylinePaths = new ArrayList<>();  
    originMarkers = new ArrayList<>();  
    destinationMarkers = new ArrayList<>();  
  
    for (final Route route : routes) {  
        float a = mMap.getCameraPosition().zoom;  
        CameraUpdate location =  
CameraUpdateFactory.newLatLngZoom(route.startLocation, a);  
        mMap.animateCamera(location, 500, null);  
  
        final TextView myAwesomeTextView3 = (TextView)  
activity.findViewById(R.id.tvDuration);  
        final TextView myAwesomeTextView4 = (TextView)  
activity.findViewById(R.id.tvDistance);  
  
        //you can't update things on the ui thread from anywhere  
        // but the ui thread, you need to use this .Where mAct  
        // is the main activity but you have to pass it in a parameter  
        // to the other class so it has access to it because it can't  
        // be a static reference  
        activity.runOnUiThread(new Runnable() {  
            public void run() {
```



```

public void clearRoute(Activity act) {
    if (polylinePaths != null) {
        for (Polyline polyline : polylinePaths) {
            polyline.remove();
        }
    }
    final TextView myAwesomeTextView = (TextView)
act.findViewById(R.id.Title);

    ....
    act.runOnUiThread(new Runnable() {
        public void run() {
            myAwesomeTextView.setText("Title");
        }
    });
    return;
}

```

Όπως βλέπουμε εδώ γίνεται η διαγραφή της διαδρομής (*polylinePaths*) και η διαγραφή παλαιού ονόματος σημείου περιοχής χρόνου απόστασης και απόστασης.

```

public void setRoute(Activity act) {
    setActivityUi(act);
    final TextView placedime1 = (TextView) act.findViewById(R.id.placedime);
    act.runOnUiThread(new Runnable() {
        public void run() {
            ....
        }
    });
}

```

Εδώ πέρα είναι το απλό SetRoute που χρειάζεται συντεταγμένες σημείων και έναν Listener. Δεν ξεχνάμε ότι το σημείο έχει διαφόρους χαρακτήρες και απομονώνω τους αριθμούς με το κόμμα.

```

public void setRoutefromNaviPanel(String PlaceLongLat, final String
Placenamegiven, final String Vicinitygiven) {

    setRoute_clear_button_state = 1;

    final TextView myAwesomeTextView3 = (TextView)
activity.findViewById(R.id.Vicinity);
    final TextView myAwesomeTextView4 = (TextView)
activity.findViewById(R.id.Title);
    final Button button = (Button) activity.findViewById(R.id.setroute);

    activity.runOnUiThread(new Runnable() {
        public void run() {
            button.setText("Clear");
            myAwesomeTextView3.setText(Vicinitygiven);
        }
    });
}

```



```

        myAwesomeTextView4.setText(Placenamegiven);
    }

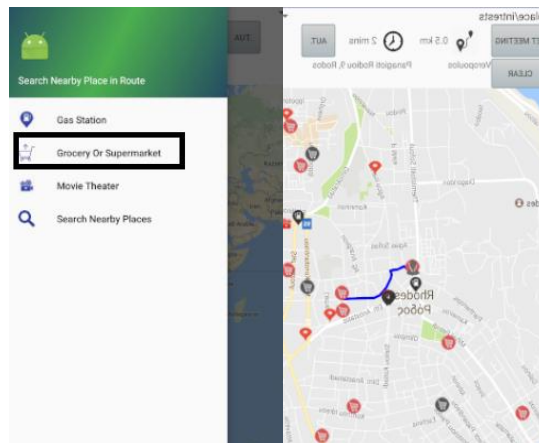
});

String s = currentLatitude + "," + currentLongitude;
try {
    new DirectionFinder(SetRoute.this, s, PlaceLongLat).execute();
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
}
}

```

Όπως λέει και η ονομασία της μεθόδου εδώ θέτουμε την διαδρομή από το Navigation Drawer το οποίο χρειαζόμαστε όνομα και Vicinity για να βάλουμε στα Text view, και το PlaceLongLat και έτσι το εκτελούμε

```
new DirectionFinder(SetRoute.this, s, PlaceLongLat).execute();
```



```

public void setRoutefromMeeting(Activity act) {
    setActivityUi(act);

    final TextView placedime1 = (TextView) act.findViewById(R.id.placedime);
    act.runOnUiThread(new Runnable() {
        public void run() {

            String lname = placedime1.getText().toString();
            if (currentLongitude == -122.084000) {
                currentLatitude = 36.436285;
                currentLongitude = 28.217539;
                // currentLatitude = 35.319647;
                // currentLongitude = 25.102336;
            }

            String s = currentLatitude + "," + currentLongitude;
            try {
                new DirectionFinder(SetRoute.this, s, lname).execute();
            }
        }
    });
}

```

```

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

});

}

```

Αυτή η μέθοδος χρησιμεύει όταν θέλω μέσω της PlaceandIntrests κλάσης/δραστηριότητας να θέσω μια διαδρομή στην οποία έχω θέσει meeting. Εδώ επιπλέον βλέπουμε μια if με τα long lat αυτά έχουν γίνει λογού του emulator διότι κάποιες φορές θέτει την αρχική θέση στην Αμερική από εκεί που έχει Preset value στο user Long Lat.

```

public void findroute(int i_counter) {
    progress = new ProgressDialog(activity);
    progress.setMessage("Searching Nearby Places");
    progress.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    String s = currentLatitude + "," + currentLongitude;
    String name_place = null;
    try {
        check = 1;
        for (int k = i_counter; k < Places_LongLatitubee.size(); k++) {
            name_place = Places_LongLatitubee.get(k);
            name_place = name_place.replace(" ", "");
            new DirectionFinder(SetRoute.this, s, name_place).execute();
            Log.v("PLACES", String.valueOf(Places_LongLatitubee));
        }
        // if(Duration.size()==Places_LongLatitubee.size())

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

```

Αυτή η μέθοδος είναι η μέθοδος με την οποία βρίσκω αποστάσεις για να εκτελέσω τον αλγόριθμο, επίσης βλέπουμε μια μεταβλητή την check η οποία ελέγχει αν έχω έρθει από εδώ ώστε παρακάτω να εκτελέσω κάποια αλλά πράγματα και τέλος στην for κάνω μια επανάληψη με τις συντεταγμένες του κάθε σημείου μέσα από πινάκα ο οποίος έχει δημιουργηθεί μέσα στην FindPlaces.

```

public void onDirectionFinderStart() {

    if (originMarkers != null) {
        for (Marker marker : originMarkers) {
            marker.remove();
        }
    }

    if (destinationMarkers != null) {
        for (Marker marker : destinationMarkers) {
            marker.remove();
        }
    }

    if (polylinePaths != null) {
        for (Polyline polyline : polylinePaths) {
            polyline.remove();
        }
    }
}

```

Μεταβλητές που χρειάζονται για την διαδρομή με την όποια μέθοδο κάνω remove προηγούμενες τιμές.

```

public void onDirectionFinderSuccess(List<Route> routes) {

```

Τελευταία μέθοδος και εξίσου σημαντική είναι αυτή που δηλώνει τι γίνεται όταν κριθεί επιτυχία.

```

    for (final Route route : routes)
        float a = mMap.getCameraPosition().zoom;
        CameraUpdate location =
        CameraUpdateFactory.newLatLngZoom(route.startLocation, a);
        mMap.animateCamera(location, 500, null);

```

Κάνουμε zoom την κάμερα πάνω στην διαδρομή.

```

    final TextView myAwesomeTextView3 = (TextView)
    activity.findViewById(R.id.tvDuration);
    ...

    activity.runOnUiThread(new Runnable() {
        public void run() {
            myAwesomeTextView3.setText(route.duration.text);
        }
    });
    ..

```

Ενημερώνουμε τα Text View με τα νέα δεδομένα της διαδρομής χρόνος κλπ.

```
PolylineOptions polylineOptions = new PolylineOptions().
    geodesic(true).
    color(Color.BLUE).
    width(10);

for (int i = 0; i < route.points.size(); i++)
    polylineOptions.add(route.points.get(i));

polylinePaths.add(mMap.addPolyline(polylineOptions));
```

Εδώ ζωγραφίζουμε στο χάρτη την διαδρομή.

```
if (check == 1) {
    clearRoute(activity);
    String s = route.duration.text;
    s = s.replace("mins", "");
    s = s.replace("min", "");
    s = s.replace(" ", "");
    Duration.add(Integer.valueOf(s));
    Log.v("AAA", String.valueOf(Duration));
    count = Duration.size();
    activity.runOnUiThread(new Runnable() {
        public void run() {
            progress.show();
            if (Duration.size() == Places_LongLatitubee.size()) {
                progress.hide();
                Navigation_ok = 1;
            }
            if (Duration.size() > Places_LongLatitubee.size()) {
                Toast.makeText(activity, "Something went wrong run again or
refresh", Toast.LENGTH_LONG).show();
            }
            return;
        }
    });
}
```

Όπως βλέπουμε στον παραπάνω κώδικα, περνάμε στον πίνακα τον χρόνο αποστάσεων και συγκεκριμένα την τιμή του χρόνου χωρίς το min για αυτό κάνουμε και replace. Παρακάτω τελειώνουμε με τον έλεγχο `Duration.size() == Places_LongLatitubee.size()` αν έχουμε φορτώσει όλες τις τιμές στον πίνακα τότε σημαίνει ότι μπορούμε να εκτελέσουμε τον αλγόριθμο `Navigation_ok = 1;` και να κρύψουμε `progress.hide()`; αν όχι τότε εμφανίζετε ένα κατάλληλο μήνυμα. Αυτό στην περίπτωση όπου κάνουμε search τους χρόνους των σημείων από την αρχική μας θέση.

Class SetRouteVariables

```
package SetRoute;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.Polyline;
import java.util.ArrayList;
import java.util.List;

public class SetRouteVariables {
    public static int Navigation_ok = 0;

    //static to be deleted or they will not be cleared from map
    public static List<Marker> originMarkers = new ArrayList<>();
    public static List<Marker> destinationMarkers = new ArrayList<>();
    public static List<Polyline> polylinePaths = new ArrayList<>();
}
```

Χρήσιμες μεταβλητές για την κλάση SetRoute.

Εν συντομία τώρα θα δείξω τις κλάσεις για το Route που με βοήθησαν ιδιαίτερος και πάλι ευχαριστώ στον [Hier Mai Thanh](https://www.youtube.com/watch?v=CCZPUeY94MU) και η εργασία του βρίσκεται εδώ <https://www.youtube.com/watch?v=CCZPUeY94MU>. Οι οποίες δεν είναι πλήρως παρόμοιες.

Layout activity_maps

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/two"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.tolis.maps3856.RouteOnly">

    <Spinner
        android:id="@+id/SpinToWin"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />

    <LinearLayout
        android:id="@+id/one"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
```

```

<EditText
    android:id="@+id/etOrigin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter origin address" />

<EditText
    android:id="@+id/etDestination"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter destination address" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btnFindPath"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Find path" />

    <ImageView
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:src="@drawable/ic_distance" />

    <TextView
        android:id="@+id/tvDistance"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"
        android:text="0 km" />

    <ImageView
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="5dp"
        android:padding="5dp"
        android:src="@drawable/ic_clock" />

    <TextView
        android:id="@+id/tvDuration"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5dp"

```

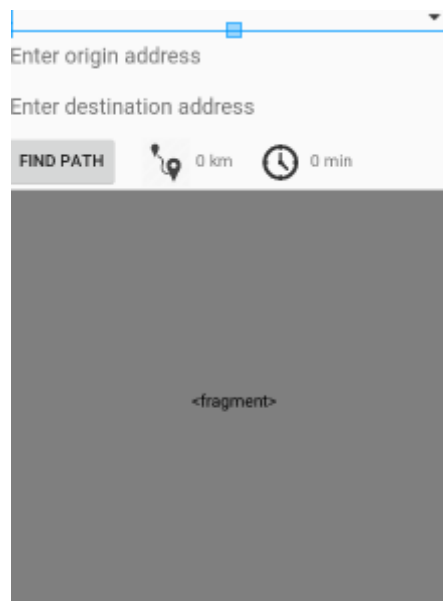
```

        android:text="0 min" />
    </LinearLayout>
</LinearLayout>

<fragment
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

</LinearLayout>

```



Class RouteOnly

```

package com.tolis.maps3856;
import android.Manifest;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.LinearLayout;
import android.widget.Spinner;

```

```

import android.widget.TextView;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;
import Modules.DirectionFinder;
import Modules.DirectionFinderListener;
import Modules.Route;

```

```

public class RouteOnly extends FragmentActivity implements
OnMapReadyCallback, DirectionFinderListener {

```

```

    private Spinner s;
    private GoogleMap mMap;
    private Button btnFindPath;
    private EditText etOrigin;
    private EditText etDestination;
    private List<Marker> originMarkers = new ArrayList<>();
    private List<Marker> destinationMarkers = new ArrayList<>();
    private List<Polyline> polylinePaths = new ArrayList<>();
    private ProgressDialog progressDialog;
    private String[] arraySpinner;
    private LinearLayout mainlayout;

```

```

    // LinearLayout one = (LinearLayout) findViewById(R.id.one);
    //one.setVisibility(View.GONE);

```

```

    @Override

```

```

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be
        used.

```

```

        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
        .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);

```

```

        btnFindPath = (Button) findViewById(R.id.btnFindPath);
        etOrigin = (EditText) findViewById(R.id.etOrigin);
        etDestination = (EditText) findViewById(R.id.etDestination);

```



```

btnFindPath.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        sendRequest();
    }
});
SpinToWin();
}

private void SpinToWin() {
    this.arraySpinner = new String[]{
        "Route Only", "place/intrests", "See Meetings", "Exit"
    };
    s = (Spinner) findViewById(R.id.SpinToWin);
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, arraySpinner);
    //
    s.setDropDownViewResource(android.R.layout.simple_list_item_single_choice);
    s.setAdapter(adapter);
    mainlayout = (LinearLayout) this.findViewById(R.id.one);
    s.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {
            String selectedItem = parent.getItemAtPosition(position).toString();
            if (selectedItem.equals("Route Only"))
                {//Toast.makeText(getApplicationContext(), "Give Starting Place and End",
                Toast.LENGTH_LONG).show();
                mainlayout.setVisibility(LinearLayout.VISIBLE); // do your stuff
            } else {
                if (selectedItem.equals("place/intrests")) {
                    Intent intent = new Intent(RouteOnly.this, PlaceAndIntrests.class);
                    RouteOnly.this.startActivity(intent);
                    RouteOnly.this.finish();
                }
                if (selectedItem.equals("See Meetings")) {
                    Intent intent = new Intent(RouteOnly.this, CheckSeeMeetings.class);

                    RouteOnly.this.startActivity(intent);
                    RouteOnly.this.finish();
                }
            }
            if (selectedItem.equals("Exit")) {
                RouteOnly.this.finish();
                System.exit(0);
            }
        }
    }) // to close the onItemClick

    public void onNothingSelected(AdapterView<?> parent) {

```

```

    }
  });
}

private void sendRequest() {
    String origin = etOrigin.getText().toString();
    String destination = etDestination.getText().toString();
    if (origin.isEmpty()) {
        Toast.makeText(this, "Please enter origin address!",
Toast.LENGTH_SHORT).show();
        return;
    }
    if (destination.isEmpty()) {
        Toast.makeText(this, "Please enter destination address!",
Toast.LENGTH_SHORT).show();
        return;
    }

    try {
        new DirectionFinder(this, origin, destination).execute();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}

@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    LatLng hcmus = new LatLng(36.435475, 28.215388);

    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(hcmus, 18));
    originMarkers.add(mMap.addMarker(new MarkerOptions()
        .title("MaMa Rusia")
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.user))
        .position(hcmus)));

    if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            requestPermissions(new String[]
                {Manifest.permission.ACCESS_COARSE_LOCATION,

```

```

Manifest.permission.ACCESS_FINE_LOCATION,
                Manifest.permission.INTERNET
                , 30);
        return;
    }
}
mMap.setMyLocationEnabled(true);
}

```

@Override

```

public void onDirectionFinderStart() {
    progressDialog = ProgressDialog.show(this, "Please wait.",
        "Finding direction..!", true);

    if (originMarkers != null) {
        for (Marker marker : originMarkers) {
            marker.remove();
        }
    }

    if (destinationMarkers != null) {
        for (Marker marker : destinationMarkers) {
            marker.remove();
        }
    }

    if (polylinePaths != null) {
        for (Polyline polyline : polylinePaths) {
            polyline.remove();
        }
    }
}

```

@Override

```

public void onDirectionFinderSuccess(List<Route> routes) {
    progressDialog.dismiss();
    polylinePaths = new ArrayList<>();
    originMarkers = new ArrayList<>();
    destinationMarkers = new ArrayList<>();

    for (Route route : routes) {

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(route.startLocation,
16));
        ((TextView) findViewById(R.id.tvDuration)).setText(route.duration.text);
        ((TextView) findViewById(R.id.tvDistance)).setText(route.distance.text);

        originMarkers.add(mMap.addMarker(new MarkerOptions()
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.start_blue))

```

```

        .title(route.startAddress)
        .position(route.startLocation));
    destinationMarkers.add(mMap.addMarker(new MarkerOptions()
        .icon(BitmapDescriptorFactory.fromResource(R.drawable.end_green))
        .title(route.endAddress)
        .position(route.endLocation)));

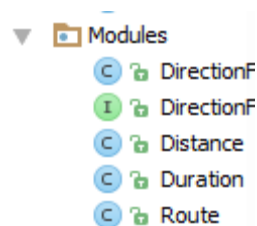
    PolylineOptions polylineOptions = new PolylineOptions()
        .geodesic(true)
        .color(Color.BLUE)
        .width(10);

    for (int i = 0; i < route.points.size(); i++)
        polylineOptions.add(route.points.get(i));

    polylinePaths.add(mMap.addPolyline(polylineOptions));
    }
}
}

```

Class DirectionFinder



```

package Modules;
import android.os.AsyncTask;
import com.google.android.gms.maps.model.LatLng;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.List;

public class DirectionFinder {
    private static final String DIRECTION_URL_API =

```

```

"https://maps.googleapis.com/maps/api/directions/json?";
private static final String GOOGLE_API_KEY = "....";
private DirectionFinderListener listener;
private String origin;
private String destination;

public DirectionFinder(DirectionFinderListener listener, String origin, String
destination) {
    this.listener = listener;
    this.origin = origin;
    this.destination = destination;
}

public void execute() throws UnsupportedOperationException {
    listener.onDirectionFinderStart();
    new DownloadRawData().execute(createUrl());
}

private String createUrl() throws UnsupportedOperationException {
    String urlOrigin = URLEncoder.encode(origin, "utf-8");
    String urlDestination = URLEncoder.encode(destination, "utf-8");

    return DIRECTION_URL_API + "origin=" + urlOrigin + "&destination=" +
urlDestination + "&key=" + GOOGLE_API_KEY;
}

private class DownloadRawData extends AsyncTask<String, Void, String> {

    @Override
protected String doInBackground(String... params) {
        String link = params[0];
        try {
            URL url = new URL(link);
            InputStream is = url.openConnection().getInputStream();
            StringBuffer buffer = new StringBuffer();
            BufferedReader reader = new BufferedReader(new InputStreamReader(is));

            String line;
            while ((line = reader.readLine()) != null) {
                buffer.append(line + "\n");
            }

            return buffer.toString();

        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
}

```

```

}

@Override
protected void onPostExecute(String res) {
    try {
        parseJSoN(res);
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
}

private void parseJSoN(String data) throws JSONException {
    if (data == null)
        return;

    List<Route> routes = new ArrayList<Route>();
    JSONObject jsonData = new JSONObject(data);
    JSONArray jsonRoutes = jsonData.getJSONArray("routes");
    for (int i = 0; i < jsonRoutes.length(); i++) {
        JSONObject jsonRoute = jsonRoutes.getJSONObject(i);
        Route route = new Route();

        JSONObject overview_polylineJson =
jsonRoute.getJSONObject("overview_polyline");
        JSONArray jsonLegs = jsonRoute.getJSONArray("legs");
        JSONObject jsonLeg = jsonLegs.getJSONObject(0);
        JSONObject jsonDistance = jsonLeg.getJSONObject("distance");
        JSONObject jsonDuration = jsonLeg.getJSONObject("duration");
        JSONObject jsonEndLocation = jsonLeg.getJSONObject("end_location");
        JSONObject jsonStartLocation = jsonLeg.getJSONObject("start_location");

        route.distance = new Distance(jsonDistance.getString("text"),
jsonDistance.getInt("value"));
        route.duration = new Duration(jsonDuration.getString("text"),
jsonDuration.getInt("value"));
        route.endAddress = jsonLeg.getString("end_address");
        route.startAddress = jsonLeg.getString("start_address");
        route.startLocation = new LatLng(jsonStartLocation.getDouble("lat"),
jsonStartLocation.getDouble("lng"));
        route.endLocation = new LatLng(jsonEndLocation.getDouble("lat"),
jsonEndLocation.getDouble("lng"));
        route.points = decodePolyLine(overview_polylineJson.getString("points"));

        routes.add(route);
    }

    listener.onDirectionFinderSuccess(routes);
}

```

```

private List<LatLng> decodePolyLine(final String poly) {
    int len = poly.length();
    int index = 0;
    List<LatLng> decoded = new ArrayList<LatLng>();
    int lat = 0;
    int lng = 0;

    while (index < len) {
        int b;
        int shift = 0;
        int result = 0;
        do {
            b = poly.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlat = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lat += dlat;

        shift = 0;
        result = 0;
        do {
            b = poly.charAt(index++) - 63;
            result |= (b & 0x1f) << shift;
            shift += 5;
        } while (b >= 0x20);
        int dlng = ((result & 1) != 0 ? ~(result >> 1) : (result >> 1));
        lng += dlng;

        decoded.add(new LatLng(
            lat / 100000d, lng / 100000d
        ));
    }

    return decoded;
}

```

Interface DirectionFinderListener

```

package Modules;
import java.util.List;

public interface DirectionFinderListener {
    void onDirectionFinderStart();

    void onDirectionFinderSuccess(List<Route> route);
}

```

Class Distance

```
package Modules;

public class Distance {
    public String text;
    public int value;

    public Distance(String text, int value) {
        this.text = text;
        this.value = value;
    }
}
```

Class Duration

```
package Modules;

public class Duration {
    public String text;
    public int value;

    public Duration(String text, int value) {
        this.text = text;
        this.value = value;
    }
}
```

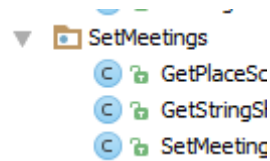
Class Route

```
package Modules;
import com.google.android.gms.maps.model.LatLng;
import java.util.List;

public class Route {
    public Distance distance;
    public Duration duration;
    public String endAddress;
    public LatLng endLocation;
    public String startAddress;
    public LatLng startLocation;

    public List<LatLng> points;
}
```


Class GetPlaceSchedule



```
package SetMeetings;
import android.os.AsyncTask;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import static SetMeetings.SetMeetingsVariables.checkernull;
import static SetMeetings.SetMeetingsVariables.types;

public class GetPlaceSchedule extends FragmentActivity {

    public void updatePlaces(String Placeid) {
        //build place query string
        String PlaceScheduleLink =
"https://maps.googleapis.com/maps/api/place/details/json?placeid=" +
        Placeid + "&key=...."//ADD KEY
        new Getplaceschedule().execute(PlaceScheduleLink);
    }

    public class Getplaceschedule extends AsyncTask<String, Void, String> {

        @Override
        protected String doInBackground(String... placesURL) {
            //fetch places
            // setContentView(R.layout.placeandintrests);
            StringBuilder placeBuilder = new StringBuilder();
            for (String placeSearchURL : placesURL) {
                try {

                    URL requestUrl = new URL(placeSearchURL);
                    HttpURLConnection connection = (HttpURLConnection)
requestUrl.openConnection();
                    connection.setRequestMethod("GET");
                    connection.connect();
                    int responseCode = connection.getResponseCode();
```

```

if (responseCode == HttpURLConnection.HTTP_OK) {

    BufferedReader reader = null;

    InputStream inputStream = connection.getInputStream();
    if (inputStream == null) {
        return "";
    }
    reader = new BufferedReader(new InputStreamReader(inputStream));

    String line;
    while ((line = reader.readLine()) != null) {
        placeBuilder.append(line + "\n");
    }

    if (placeBuilder.length() == 0) {
        return "";
    }
    connection.disconnect();
    Log.d("test", placeBuilder.toString());
} else {
    Log.i("test", "Unsuccessful HTTP Response Code: " +
responseCode);
}
} catch (MalformedURLException e) {
    Log.e("test", "Error processing Places API URL", e);
} catch (IOException e) {
    Log.e("test", "Error connecting to Places API", e);
}
}
return placeBuilder.toString();
}
//process data retrieved from doInBackground
protected void onPostExecute(String result) {
    //parse place data returned from Google Place
    if (result.equals(""))
        return;
    try {
        //parse JSON
        //create JSONObject, pass sting returned from doInBackground
        JSONObject resultObject = new JSONObject(result);
        types =
resultObject.getJSONObject("result").getJSONObject("opening_hours");
        checkernull = 0;
        // Log.d("testa",types.toString());
    } catch (JSONException e) {
        checkernull = 1;
        e.printStackTrace();
    }
}

```

```
}  
  
}  
}
```

Σε αυτήν την κλάση θα δούμε πως γίνεται η ενημέρωση του Place Schedule.

```
,  
{  
  "close" : {  
    "day" : 3,  
    "time" : "2200"  
  },  
  "open" : {  
    "day" : 3,  
    "time" : "0700"  
  }  
},  
{  
  "close" : {  
    "day" : 4,  
    "time" : "2200"  
  },  
  "open" : {  
    "day" : 4,  
    "time" : "0700"  
  }  
},  
{  
  "close" : {  
    "day" : 5,  
    "time" : "2200"  
  },  
  "open" : {  
    "day" : 5,  
    "time" : "0700"  
  }  
},  
{  
  "close" : {  
    "day" : 6,  
    "time" : "1000"  
  },  
  "open" : {  
    "day" : 6,  
    "time" : "0700"  
  }  
}  
},  
"weekday_text" : [  
  "Monday: 7:00 AM - 10:00 PM",  
  "Tuesday: 7:00 AM - 10:00 PM",  
  "Wednesday: 7:00 AM - 10:00 PM",  
  "Thursday: 7:00 AM - 10:00 PM",  
  "Friday: 7:00 AM - 10:00 PM",  
  "Saturday: 7:00 - 10:00 AM",  
  "Sunday: 9:00 AM - 10:00 PM"  
]
```

Query:

<https://maps.googleapis.com/maps/api/place/details/json?placeid=ChIJ891nP-1hRQRejvwq6MzL3o&key=.....>

```

"result" : {
  "address_components" : [
    {
      "long_name" : "8",
      "short_name" : "8",
      "types" : [ "street_number" ]
    },
    {
      "long_name" : "Diagoridon",
      "short_name" : "Diagoridon",
      "types" : [ "route" ]
    },
    {
      "long_name" : "Rodos",
      "short_name" : "Rodos",
      "types" : [ "locality", "political" ]
    },
    {
      "long_name" : "Rodos",
      "short_name" : "Rodos",
      "types" : [ "administrative_area_level_5", "political" ]
    },
    {
      "long_name" : "Rhodes",
      "short_name" : "Rhodes",
      "types" : [ "administrative_area_level_4", "political" ]
    },
    {
      "long_name" : "Rodos",
      "short_name" : "Rodos",
      "types" : [ "administrative_area_level_3", "political" ]
    },
    {
      "long_name" : "Greece",
      "short_name" : "GR",
      "types" : [ "country", "political" ]
    },
    {
      "long_name" : "851 00",
      "short_name" : "851 00",
      "types" : [ "postal_code" ]
    }
  ],
  "adr_address" : "\u003cspan class=\"street-address\"\u003eDiagoridoi
00\u003c/span\u003e, \u003cspan class=\"country-name\"\u003eGreece\u003c/
"formatted_address" : "Diagoridon 8, Rodos 851 00, Greece",
"formatted_phone_number" : "2241 029620",
"geometry" : {
  "location" : {
    "lat" : 36.44094990000001,

```

Αυτή η κλάση είναι αρκετά απλή ξεκινάμε έχοντας το place id το οποίο το περνούμε από την FindPlaces ύστερα το τοποθετούμε στο query και εκτελούμε.

```
String PlaceScheduleLink =  
"https://maps.googleapis.com/maps/api/place/details/json?placeid=" +  
Placeid + "&key=AIzaSyCa9TerPn2Hmy9wbAQMX-  
CEd_fOOJ4r0bw";//ADD KEY  
new Getplaceschedule().execute(PlaceScheduleLink);
```

Με παρόμοιο τρόπο θα έχουμε και την **protected** String doInBackground όπου κάνει ακριβώς τα ίδια όπως και στην FindPlaces ,σύνδεση κλπ.

Και τέλος έχουμε την onPostExecute
Όπου μέσα από το Json object θα πάρουμε το Opening hours.

```
},  
"icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png"  
"id" : "0f661c1bf298f586f1c247267b37c184c2af4074",  
"international_phone_number" : "+30 2241 029620",  
"name" : "Moka Bakery",  
"opening_hours" : {  
  "open_now" : true,  
  "periods" : [  
    {  
      "close" : {  
        "day" : 0,  
        "time" : "2200"  
      },  
      "open" : {  
        "day" : 0,  
        "time" : "0900"  
      }  
    }  
  ]  
}
```

```
types = responseObject.getJSONObject("result").getJSONObject("opening_hours");  
Επίσης αναλόγως επιτυχίας η αποτυχίας αλλάζουμε την checkernull = 1;  
πχ αν έχουμε αποτυχία.
```

Τι θα γίνει αν έχουμε αποτυχία θα δώσουμε τιμές Null ;

Class GetStringShopSuggested

```
package SetMeetings;
import static SetMeetings.SetMeetingsVariables.Friday;
import static SetMeetings.SetMeetingsVariables.Monday;
import static SetMeetings.SetMeetingsVariables.Saturday;
import static SetMeetings.SetMeetingsVariables.Sunday;
import static SetMeetings.SetMeetingsVariables.Thursday;
import static SetMeetings.SetMeetingsVariables.Tuesday;
import static SetMeetings.SetMeetingsVariables.Wednesday;

public class GetStringShopSuggested {

    public String Gas_Station() {
        Monday = new String[]{"0600", "2230", "2300", "0100"};
        Tuesday = new String[]{"2300", "0200", "9999", "9999"};
        Wednesday = new String[]{"0600", "2400", "9999", "9999"};
        Thursday = new String[]{"0600", "2400", "9999", "9999"};
        Friday = new String[]{"0600", "2400", "9999", "9999"};
        Saturday = new String[]{"0600", "2230", "9999", "9999"};
        Sunday = new String[]{"0600", "2230", "9999", "9999"};
        return " Suggested By Me:\n\
Monday: 6:00 AM – 10:30 PM, 23:00 – 01:00
PM\,\n" +
        "\Tuesday: 23:00 PM – 02:00 PM\,\n" +
        "\Wednesday: 6:00 AM – 12:00 PM\,\n" +
        "\Thursday: 6:00 AM – 12:00 PM\,\n" +
        "\Friday: 6:00 AM – 12:00 PM\,\n" +
        "\Saturday: 6:00 AM – 10:30 PM\,\n" +
        "\Sunday:6:00 AM – 10:30 PM\''";
    }

    public String Movie_Theater() {

        Monday = new String[]{"1720", "2300", "9999", "9999"};
        Tuesday = new String[]{"1720", "2300", "9999", "9999"};
        Wednesday = new String[]{"1720", "2300", "9999", "9999"};
        Thursday = new String[]{"1720", "2300", "9999", "9999"};
        Friday = new String[]{"1720", "2300", "9999", "9999"};
        Saturday = new String[]{"1720", "2300", "9999", "9999"};
        Sunday = new String[]{"1720", "2300", "9999", "9999"};
        return " Suggested By Me:\n \
Monday: 5:20 PM – 11:00 PM\,\n" +
        "\Tuesday: 5:20 PM – 11:00 PM\,\n" +
        "\Wednesday: 5:20 PM – 11:00 PM\,\n" +
        "\Thursday: 5:20 PM – 11:00 PM\,\n" +
        "\Friday: 5:20 PM – 11:00 PM\,\n" +
        "\Saturday: 5:20 PM – 11:00 PM\,\n" +
        "\Sunday: 5:20 PM – 11:00 PM\''";
    }
}
```

```

}

public String Grocery_Or_Supermarket() {
    Monday = new String[]{"0800", "1400", "1700", "2100"};
    Tuesday = new String[]{"0800", "1400", "1700", "2100"};
    Wednesday = new String[]{"0800", "1400", "1700", "2100"};
    Thursday = new String[]{"0800", "1400", "1700", "2100"};
    Friday = new String[]{"0800", "1400", "1700", "2100"};
    Saturday = new String[]{"0800", "1400", "9999", "9999"};
    Sunday = new String[]{"9999", "9999", "9999", "9999"};//closed

    return " Suggested By Me:\n\"Monday: 8:00 AM – 14:00 PM, 5:00 – 9:00
PM\", \n" +
        "\"Tuesday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM\", \n" +
        "\"Wednesday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM\", \n" +
        "\"Thursday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM\", \n" +
        "\"Friday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM\", \n" +
        "\"Saturday: 8:00 AM – 14:00 PM\", \n" +
        "\"Sunday: Closed\"";
}

public void Close() {
    Monday = new String[]{"7777", "9999", "9999", "9999"};
    Tuesday = new String[]{"7777", "9999", "9999", "9999"};
    Wednesday = new String[]{"7777", "9999", "9999", "9999"};
    Thursday = new String[]{"7777", "9999", "9999", "9999"};
    Friday = new String[]{"7777", "9999", "9999", "9999"};
    Saturday = new String[]{"7777", "9999", "9999", "9999"};
    Sunday = new String[]{"7777", "9999", "9999", "9999"};
}
}

```

Σε αυτήν την κλάση έβαλα ξεχωριστά τα ποιο συχνά schedule κατά την άποψη μου αν δεν υπάρχει κάποιο γνωστό στο google. Αυτό όμως που αξίζει να πω είναι ότι δημιούργησα πίνακα για κάθε ημέρα με 4 θέσεις οι οποίες αντιπροσωπεύουν ώρα.

Πχ

```

Monday = new String[]{"0800", "1400", "1700", "2100"};
Monday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM\

```

Τι είναι όμως το 7777,9999;

```

Monday = new String[]{"7777", "9999", "9999", "9999"};

```

Αν το μαγαζί είναι ανοικτό όλη την ημέρα βάζω στην πρώτη τιμή 7777 αν είναι κλειστό,

```

Sunday = new String[]{"9999", "9999", "9999", "9999"};//closed.

```

Class SetMeetingsVariables

```
package SetMeetings;
import org.json.JSONObject;
import java.util.ArrayList;

public class SetMeetingsVariables {
    public static ArrayList<String> PlacesId = new ArrayList<String>();//use to hold
placeid for google query
    public static ArrayList<String> PlacesName = new ArrayList<String>();//use to
hold places name after google query
    public static JSONObject types;
    public static ArrayList<Integer> typeofmark = new ArrayList<Integer>();//type of
place of interest
    public static int checkernull;
    public static String Monday[], Tuesday[], Wednesday[], Thursday[], Friday[],
Saturday[], Sunday[];
}
```

Οι μεταβλητές αυτές χρησιμεύουν για την κλάση SetMeeting.

Layout setmeeting

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#254B63"
    android:orientation="vertical"
    android:weightSum="1"
    tools:context="com.tolis.maps3856.SetMeeting">

    <TextView
        android:id="@+id/WeekendSchedule"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:text="Loading...."
        android:textColor="@color/common_google_signin_btn_text_dark" />

    <SeekBar
        android:id="@+id/seekBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/delay"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delay:" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:weightSum="1">

        <Button
            android:alpha="0.65"
            android:id="@+id/goBack"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"

            android:background="@drawable/common_google_signin_btn_text_dark"
            android:onClick="onButtonClick"
            android:text="GoBack"                ." />

        <Button
```

```

        android:alpha="0.65"
        android:id="@+id/setcheckmeetings"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:onClick="onButtonClick"
        android:text="Set/Check meetings" />
</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"

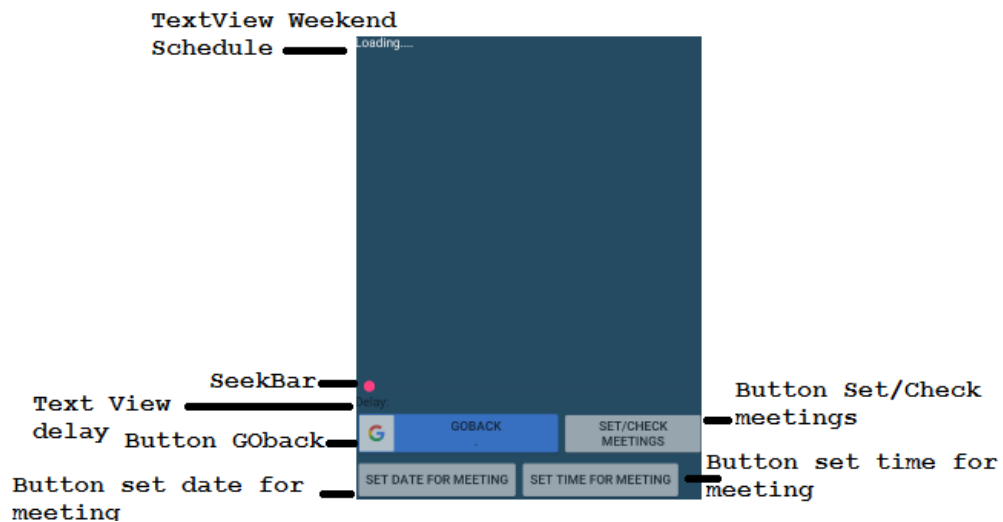
    android:weightSum="1">

    <Button
        android:alpha="0.65"
        android:id="@+id/sdfm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:onClick="onButtonClick2"
        android:text="set date for meeting"
        />

    <Button
        android:alpha="0.65"
        android:id="@+id/stfm"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="0.5"
        android:onClick="onButtonClick2"
        android:text="set time for meeting" />
</LinearLayout>
</LinearLayout>

```

Η διάφορα με τα προηγούμενα layout είναι ότι εδώ έχουμε μια SeekBar επιπλέον.



Class SetMeeting

```

package com.tolis.maps3856;
import android.app.DatePickerDialog;
import android.app.Dialog;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.support.v4.app.FragmentActivity;
import android.util.Log;
import android.view.View;
import android.widget.DatePicker;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.util.Arrays;
import java.util.Calendar;
import Database.DatabaseInterface;
import SetMeetings.GetPlaceSchedule;
import SetMeetings.GetStringShopSuggested;
import static SetMeetings.SetMeetingsVariables.Friday;
import static SetMeetings.SetMeetingsVariables.Monday;
import static SetMeetings.SetMeetingsVariables.PlacesId;
import static SetMeetings.SetMeetingsVariables.PlacesName;

```

```

import static SetMeetings.SetMeetingsVariables.Saturday;
import static SetMeetings.SetMeetingsVariables.Sunday;
import static SetMeetings.SetMeetingsVariables.Thursday;
import static SetMeetings.SetMeetingsVariables.Tuesday;
import static SetMeetings.SetMeetingsVariables.Wednesday;
import static SetMeetings.SetMeetingsVariables.checkernull;
import static SetMeetings.SetMeetingsVariables.typeofmark;
import static SetMeetings.SetMeetingsVariables.types;

public class SetMeeting extends FragmentActivity {
    private int position;
    private JSONArray weekSche = null;
    private String Title;
    private String Vicinity;
    private String Distance;
    private String Duration;
    private String LongLatofplace;
    private int delay = 0;
    private int Mondaycounter, Tuesdaycounter, Wednesdaycounter,
Thursdaycounter, Fridaycounter, Saturdaycounter, Sundaycounter;
    private int year_picked, month_picked, day_picked, hour_picked,
minute_picked, typeofplace;
    private String yearr, monthh, dayy, hourr, minutee;
    static final int DILOG_ID = 0, DILOG2_ID = 1;
    private int all_ok = 0, all_ok2 = 0;

    protected void onCreate(Bundle savedInstanceState) {
        position = 0;
        super.onCreate(savedInstanceState);
        setContentView(R.layout.setmeeting);
        Title = getIntent().getStringExtra("Title");
        Vicinity = getIntent().getStringExtra("Vicinity");
        Distance = getIntent().getStringExtra("Distance");
        Duration = getIntent().getStringExtra("Duration");

        delay = 0;
        LongLatofplace = getIntent().getStringExtra("LongLatofplace");
        TextView delay_text = (TextView) findViewById(R.id.delay);
        delay_text.setText("TimeStay:" + delay);
        for (String str : PlacesName) {
            if (str.contains(Title))
                break;
            position += 1;
        }
        Calendar c = Calendar.getInstance();
        hour_picked = c.get(Calendar.HOURLY);
        minute_picked = c.get(Calendar.MINUTE);
        hourr = String.valueOf(hour_picked);
        minutee = String.valueOf(minute_picked);
    }
}

```

```

getplaceinfo();

seekBar();

}

private void seekBar() {
    int temp = 50 ;
    SeekBar seekBar = (SeekBar) findViewById(R.id.seekBar);
    seekBar.setMax(temp); //8 hours - duration to catch the bus...
    // seekBar.setProgress(Integer.parseInt(Duration));
    seekBar.setOnSeekBarChangeListener(
        new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
            }

            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
                boolean fromUser) {
                delay = progress;
                TextView delay_text = (TextView) findViewById(R.id.delay);
                delay_text.setText("TimeStay:" + delay);
            }
        }
    );
}

```

```

public void onClick(View v) {
    if (v.getId() == R.id.goBack) {
        Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
        SetMeeting.this.startActivity(intent);
        SetMeeting.this.finish();
    }
    if (v.getId() == R.id.setcheckmeetings) {
        if (all_ok == 1 && all_ok2 == 1) { //check if day and time set
            DatabaseInterface di = new DatabaseInterface();
            di.setActivityUi(this);
            di.createdb();
            boolean continuee = di.insert(yearr, monthh, dayy, hourr, minutee,
                LongLatofplace, Title, Vicinity, Distance, Duration, typeofplace, delay);
            if (continuee == false)
                return;
            Intent intent = new Intent(SetMeeting.this, CheckSeeMeetings.class);

```

```

        SetMeeting.this.startActivity(intent);
        SetMeeting.this.finish();
    } else
        Toast.makeText(getApplicationContext(), "Please Select Day Or Hour",
Toast.LENGTH_SHORT).show();
    }
}

```

```

public void onClick2(View v) {
    if (v.getId() == R.id.sdfm) {
        getDate();
        all_ok = 1;
    }
    if (v.getId() == R.id.stfm) {
        getTime();
        all_ok2 = 1;
    }
}

```

```

public void getDate() {
    this.showDialog(DILOG_ID);
    new DatePickerDialog(this, DpickerListener, year_picked, month_picked,
day_picked);
}

```

```

public void getTime() {
    this.showDialog(DILOG2_ID);
    new TimePickerDialog(this, TimepickerListener, hour_picked,
minute_picked, false);
}

```

@Override

```

public Dialog onCreateDialog(int id) {

    if (id == DILOG_ID) {
        Calendar c = Calendar.getInstance();
        year_picked = c.get(Calendar.YEAR);
        month_picked = c.get(Calendar.MONTH);
        day_picked = c.get(Calendar.DAY_OF_MONTH);
        return new DatePickerDialog(this, DpickerListener, year_picked,
month_picked, day_picked);
    }
    if (id == DILOG2_ID) {
        Calendar c = Calendar.getInstance();
        hour_picked = c.get(Calendar.HOURLY_OF_DAY);
        minute_picked = c.get(Calendar.MINUTE);
        return new TimePickerDialog(this, TimepickerListener, hour_picked,
minute_picked, false);
    }
}

```

```

    }
    return null;
}

```

```

private DatePickerDialog.OnDateSetListener DpickerListener = new
DatePickerDialog.OnDateSetListener() {

```

```

    @Override
    public void onDateSet(DatePicker view, int year, int monthOfYear, int
dayOfMonth) {
        year = String.valueOf(year);
        monthh = String.valueOf(monthOfYear + 1); //cause android trolling month
start from 0
        dayy = String.valueOf(dayOfMonth);
        if (monthh.length() == 1) {
            monthh = "0" + monthh;
        }

        if (dayy.length() == 1) {
            dayy = "0" + dayy;
        }
        Log.d("year", year);
        Log.d("month", monthh);
        Log.d("day", dayy);
    }
};

```

```

private TimePickerDialog.OnTimeSetListener TimepickerListener = new
TimePickerDialog.OnTimeSetListener() {

```

```

    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
        hourr = String.valueOf(hourOfDay);
        minutee = String.valueOf(minute);
        if (hourr.length() == 1) hourr = "0" + hourr;
        if (minutee.length() == 1) minutee = "0" + minutee;
        Log.d("hour", hourr);
        Log.d("mminute", minutee);
    }
};

```

```

private void getplaceinfo() {
    GetPlaceSchedule gps = new GetPlaceSchedule();

    gps.updatePlaces(PlacesId.get(position));
    Handler handler = new Handler();
}

```

```

handler.postDelayed(new Runnable() {
    @Override
    public void run() {

        try {

            if (checkernull == 0)
                weekSche = types.getJSONArray("weekday_text");
            setText();
            //second way
            SetMeetings.SetMeetingsVariables.typeofmark.removeAll(SetMeetings.SetMeetingsVa
            riables.typeofmark);
            typeofmark.removeAll(typeofmark);
            PlacesName.removeAll(PlacesName);
            PlacesId.removeAll(PlacesId);
            if (checkernull == 0)
                getDetails();
        } catch (JSONException e) {
            e.printStackTrace();
            Toast.makeText(getApplicationContext(), "Something Went Wrong
Returned back", Toast.LENGTH_LONG).show();
            Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
            SetMeeting.this.startActivity(intent);
            SetMeeting.this.finish();

        } catch (NullPointerException e) {
            e.printStackTrace();
            Toast.makeText(getApplicationContext(), "Something Went Wrong
Returned back", Toast.LENGTH_LONG).show();
            Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
            SetMeeting.this.startActivity(intent);
            SetMeeting.this.finish();

        }

    }

}, 2000);

}

private void setText() {
    String Type = "";
    String streamschedule = "";
    try {
        typeofplace = typeofmark.get(position);
        if (typeofmark.get(position) == 1) Type = "Gas station";
        if (typeofmark.get(position) == 2) Type = "Movie Theater";
        if (typeofmark.get(position) == 3) Type = "Grocery Or Supermarket";
    } catch (IndexOutOfBoundsException e) {

```



```

        Toast.makeText(getApplicationContext(), "Something Went Wrong
Returned back", Toast.LENGTH_LONG).show();
        Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
        SetMeeting.this.startActivity(intent);
        SetMeeting.this.finish();
    }
    TextView WS = (TextView) findViewById(R.id.WeekendSchedule);
    if (checkernull == 0) {
        String temp = weekSche.toString().replace("[", "");
        temp = temp.replace("]", "");
        temp = temp.replace("\\", "");
        temp = temp.replace("T", "\nT");
        temp = temp.replace("W", "\nW");
        temp = temp.replace("F", "\nF");
        temp = temp.replace("S", "\nS");
        WS.setText(Type + ":" + Title + "\n\n" + "Vicinity:" + Vicinity + "\n\n" +
"Distance:" + Distance + "\n\n" + "Duration:" + Duration + "\n\n" +
"Suggested By Google\n" + temp);
        weekSche = new JSONArray();
    } else if (checkernull == 1) {
        String temp = "";
        if (typeofmark.get(position) == 1) {
            GetStringShopSuggested gsss = new GetStringShopSuggested();
            temp = gsss.Gas_Station();
        }
        if (typeofmark.get(position) == 2) {
            GetStringShopSuggested gsss = new GetStringShopSuggested();
            temp = gsss.Movie_Theater();
        }
        if (typeofmark.get(position) == 3) {
            GetStringShopSuggested gsss = new GetStringShopSuggested();
            temp = gsss.Grocery_Or_Supermarket();
        }
        WS.setText(Type + ":" + Title + "\n\n" + "Vicinity:" + Vicinity + "\n\n" +
"Distance:" + Distance + "\n\n" + "Duration:" + Duration + "\n\n" + temp);
        Log.d("testa", Arrays.toString(Sunday));
        Log.d("testa", Arrays.toString(Saturday));
        Log.d("testa", Arrays.toString(Friday));
        Log.d("testa", Arrays.toString(Tuesday));
        Log.d("testa", Arrays.toString(Wednesday));
        Log.d("testa", Arrays.toString(Thursday));
        Log.d("testa", Arrays.toString(Monday));
    }
}

public void resetdayhours() {
    Monday = new String[]{"9999", "9999", "9999", "9999"};
    Tuesday = new String[]{"9999", "9999", "9999", "9999"};
    Wednesday = new String[]{"9999", "9999", "9999", "9999"};
    Thursday = new String[]{"9999", "9999", "9999", "9999"};
}

```

```

Friday = new String[]{"9999", "9999", "9999", "9999"};
Saturday = new String[]{"9999", "9999", "9999", "9999"};
Sunday = new String[]{"9999", "9999", "9999", "9999"};
Mondaycounter = 0;
Tuesdaycounter = 0;
Wednesdaycounter = 0;
Thursdaycounter = 0;
Fridaycounter = 0;
Saturdaycounter = 0;
Sundaycounter = 0;
}

```

```

private void getDetails() {
    resetdayhours();
    try {
        JSONArray periods = types.getJSONArray("periods");

        for (int i = 0; i < periods.length(); i++) {
            JSONObject OpenClose = periods.getJSONObject(i);
            JSONObject Open = OpenClose.getJSONObject("open");
            JSONObject Close = OpenClose.getJSONObject("close");
            String Day = Open.getString("day");
            String Hour = Open.getString("time");
            for (int j = 0; j <= 1; j++) {
                if (Day.equals("1") || Close.equals("1"))//Monday
                {
                    Monday[Mondaycounter] = Hour;
                    Mondaycounter += 1;
                }
                if (Day.equals("2") || Close.equals("2"))//Tuesday
                {
                    Tuesday[Tuesdaycounter] = Hour;
                    Tuesdaycounter += 1;
                }
                if (Day.equals("3") || Close.equals("3"))//Wednesday
                {
                    Wednesday[Wednesdaycounter] = Hour;
                    Wednesdaycounter += 1;
                }
                if (Day.equals("4") || Close.equals("4"))//Thursday
                {
                    Thursday[Thursdaycounter] = Hour;
                    Thursdaycounter += 1;
                }
                if (Day.equals("5") || Close.equals("5"))//Friday
                {
                    Friday[Fridaycounter] = Hour;
                    Fridaycounter += 1;
                }
            }
        }
    }
}

```

```

    if (Day.equals("6") || Close.equals("6"))//Saturday
    {
        Saturday[Saturdaycounter] = Hour;
        Saturdaycounter += 1;
    }
    if (Day.equals("0") || Close.equals("0"))//Sunday
    {
        Sunday[Sundaycounter] = Hour;
        Sundaycounter += 1;
    }
    Day = Close.getString("day");
    Hour = Close.getString("time");
}

}
} catch (JSONException e) {
    //24 hour
    GetStringShopSuggested gsss = new GetStringShopSuggested();
    gsss.Close();
    e.printStackTrace();
}
Log.d("testa", Arrays.toString(Sunday));
Log.d("testa", Arrays.toString(Saturday));
Log.d("testa", Arrays.toString(Friday));
Log.d("testa", Arrays.toString(Tuesday));
Log.d("testa", Arrays.toString(Wednesday));
Log.d("testa", Arrays.toString(Thursday));
Log.d("testa", Arrays.toString(Monday));
}
}
}

```

```

private int position;
private JSONArray weekSche = null;
private String Title;
private String Vicinity;
private String Distance;
private String Duration;
private String LongLatofplace;
private int delay = 0;
private int Mondaycounter, Tuesdaycounter, Wednesdaycounter,
Thursdaycounter, Fridaycounter, Saturdaycounter, Sundaycounter;
private int year_picked, month_picked, day_picked, hour_picked,
minute_picked, typeofplace;
private String yearr, monthh, dayy, hourr, minutee;
static final int DILOG_ID = 0, DILOG2_ID = 1;
private int all_ok = 0, all_ok2 = 0;

```

Οι μεταβλητές αφορούν τα έξης: θέση του σημείου ενδιαφέροντος στον πίνακα ,weekShe τα αποτελέσματα από το Json Array που πήραμε από την google, όνομα περιοχή κλπ., καθυστέρηση χρόνου που θα περάσει κανείς στο κατάστημα, εβδομαδιαίο πρόγραμμα σε πίνακες, ημερομηνία και ώρα που διάλεξε ο χρήστης από τα Dialogs ,Id για τα Dialogs και μεταβλητή που τσεκάρουμε αν όντως έχει επιδεχθεί ημερομηνία.

Στην onCreate ξεκινάμε περνώντας τις τιμές του σημείου ενδιαφέροντος όνομα περιοχή απόσταση χρόνος μέσω της δραστηριότητας του PlaceandIntrests, στην συνέχεια διαμορφώνουμε το text στο text view του delay delay_text.setText("TimeStay:" + delay);, μετά βρίσκουμε την θέση του σημείου ενδιαφέροντος

```

for (String str : PlacesName) {
    if (str.contains(Title))
        break;
    position += 1;
}

```

,δημιουργούμε ένα στιγμιότυπο του ημερολογίου

```
Calendar c = Calendar.getInstance();
```

όπου περνούμε την ώρα, καλούμε την μέθοδο getplaceinfo(); και την seekBar();

```
private void seekBar
```

```
int temp = 50 ;
```

```
SeekBar seekBar = (SeekBar) findViewById(R.id.seekBar);
```

```
seekBar.setMax(temp);
```

```
seekBar.setOnSeekBarChangeListener
```

Σε αυτήν την μέθοδο θέτουμε την μέγιστη τιμή του SeekBar και βάζουμε έναν listener

Αυτό με την σειρά του θα δώσει μεθόδους ανάλογος τι γίνεται. Αυτό τώρα που χρειαζόμαστε είναι να εντοπίζουμε τις αλλαγές και εμφανίζουμε την τιμή στο *delay* TextView.

```
public void onProgressChanged(SeekBar seekBar, int progress,  
                             boolean fromUser) {  
    delay = progress;  
    TextView delay_text = (TextView) findViewById(R.id.delay);  
    delay_text.setText("TimeStay:" + delay);  
}
```

```
public void onClick(View v) {  
    if (v.getId() == R.id.goBack) {  
        Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);  
        SetMeeting.this.startActivity(intent);  
        SetMeeting.this.finish();  
    }  
    if (v.getId() == R.id.setcheckmeetings) {  
        if (all_ok == 1 && all_ok2 == 1) {//check if day and time set  
            DatabaseInterface di = new DatabaseInterface();  
            di.setActivityUi(this);  
            di.createdb();  
            boolean continuee = di.insert(yearr, monthh, dayy, hourr, minuttee,  
LongLatofplace, Title, Vicinity, Distance, Duration, typeofplace, delay);  
            if (continuee == false)  
                return;  
            Intent intent = new Intent(SetMeeting.this, CheckSeeMeetings.class);  
            SetMeeting.this.startActivity(intent);  
            SetMeeting.this.finish();  
        } else  
            Toast.makeText(getApplicationContext(), "Please Select Day Or Hour",  
Toast.LENGTH_SHORT).show();  
    }  
}
```

Σε αυτήν την μέθοδο βλέπουμε όταν ο χρήστης επιχειρεί να πατήσει το κουμπί πίσω τότε τον προωθεί στο activity PlaceAndIntrests και διακόπτει το τωρινό . Αν όμως προχωρήσει και θέλει να καταχωρήσει το meeting θα πρέπει πρώτα να ελέγξει αν όντως έχει επιλεγθεί ημερομηνία και ώρα .Στην συνέχεια θα εκτελεστεί το πέρασμα στην βάση όπου παρακάτω στην κλάση DatabaseInterface θα αναφερθώ πλήρως τι γίνεται ,αν όντως ισχύει θα περαστούν και οι κατάλληλες τιμές, αν όχι θα εμφανιστεί το κατάλληλο μήνυμα.

```

public void onClick2(View v) {
    if (v.getId() == R.id.sdfm) {
        getDate();
        all_ok = 1;
    }
    if (v.getId() == R.id.stfm) {
        getTime();
        all_ok2 = 1;
    }
}

```

Αναλόγως τι θα πατήσουμε θα προχωρήσουμε παρακάτω.

```

public void getDate() {
    this.showDialog(DIALOG_ID);
    new DatePickerDialog(this, DpickerListener, year_picked, month_picked,
    day_picked);
}

```

```

public void getTime() {
    ....
}

```

@Override

```

public Dialog onCreateDialog(int id) {

    if (id == DIALOG_ID) {
        Calendar c = Calendar.getInstance();
        year_picked = c.get(Calendar.YEAR);
        month_picked = c.get(Calendar.MONTH);
        day_picked = c.get(Calendar.DAY_OF_MONTH);
        return new DatePickerDialog(this, DpickerListener, year_picked,
    month_picked, day_picked);
    }
    if (id == DIALOG2_ID) {
    ....
}

```

Όπως αναφέρθηκα με τον κώδικα παραπάνω όταν στην αρχή καλούμε για να πάρουμε ένα Instance του ημερολογίου περνούμε και τις τιμές του, αυτό το κάνουμε γιατί όταν θέλουμε να δώσουμε ημερομηνία ώρα πρέπει να εμφανιστεί μια τωρινή ημερομηνία και όχι κάποιο 2006.

```
private DatePickerDialog.OnDateSetListener DpickerListener = new  
DatePickerDialog.OnDateSetListener() {
```

```
    @Override
```

```
    public void onDateSet(DatePicker view, int year, int monthOfYear, int  
    dayOfMonth) {
```

```
        yearr = String.valueOf(year);  
        monthh = String.valueOf(monthOfYear + 1);  
        dayy = String.valueOf(dayOfMonth);  
        if (monthh.length() == 1) {  
            monthh = "0" + monthh;  
        }  
    }
```

```
        if (dayy.length() == 1) {  
            dayy = "0" + dayy;  
        }  
        Log.d("year", yearr);  
        Log.d("month", monthh);  
        Log.d("day", dayy);  
    }  
};
```

```
private TimePickerDialog.OnTimeSetListener TimepickerListener = new  
TimePickerDialog.OnTimeSetListener() {
```

```
    @Override
```

```
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
```

```
        hourr = String.valueOf(hourOfDay);  
        minuttee = String.valueOf(minute);  
        if (hourr.length() == 1) hourr = "0" + hourr;  
        if (minuttee.length() == 1) minuttee = "0" + minuttee;  
        Log.d("hour", hourr);  
        Log.d("mminute", minuttee);  
    }
```

```
};
```

Εδώ όπως βλέπουμε είναι ο Listener για το κάθε κουτί ημερολόγιο η ρολόι και περνούμε τις τιμές , οι οποίες αναθέτονται με την σειρά τους. Επίσης ιδιαίτερη σημασία στο μήνα διότι η τιμή που δίνει το android είναι μια μικρότερη.

```

public void resetdayhours() {
    Monday = new String[]{"9999", "9999", "9999", "9999"};
    Tuesday = new String[]{"9999", "9999", "9999", "9999"};
    ....
    Mondaycounter = 0;
    Tuesdaycounter = 0;
    ....
}

```

Επαναφέρουμε τους πίνακες στην αρχική κατάσταση.

```

private void getDetails() {
    resetdayhours();
}

```

```

{
  "periods" : [
    {
      "close" : {
        "day" : 1,
        "time" : "2130"
      },
      "open" : {
        "day" : 1,
        "time" : "0700"
      }
    },
    {
      "close" : {
        "day" : 2,
        "time" : "2130"
      },
      "open" : {
        "day" : 2,
        "time" : "0700"
      }
    }
  ]
}

```

Και εδώ έχουμε ένα Json object το οποίο το περνούμε από
 JSONArray periods = *types*.getJSONArray("periods");
 συγκεκριμένα από την κλάση GetPlaceShedule


```

for (int i = 0; i < periods.length(); i++) {
    JSONObject OpenClose = periods.getJSONObject(i);
    JSONObject Open = OpenClose.getJSONObject("open");
    JSONObject Close = OpenClose.getJSONObject("close");
    String Day = Open.getString("day");
    String Hour = Open.getString("time");
}

```

Στην συνέχεια με την for θα κάνουμε επανάληψη όσο μεγάλο είναι το Period Length όσες είναι οι μέρες δηλαδή με το πρόγραμμα και μετά περνούμε αυτές τις τιμές με το

```

JSONObject Open
JSONObject Close
String Day = Open.getString("day");
String Hour = Open.getString("time");

```

Παρακάτω βάζουμε αυτές τις τιμές μέσα σε πίνακα με μια διπλή επανάληψη, διπλή επανάληψη; Αυτή η διπλή επανάληψη την δημιουργήσα ώστε να περάσω και τις τιμές του close με τον ίδιο τρόπο διότι όπως είδαμε παραπάνω

```

String Day = Open.getString("day");
String Hour = Open.getString("time");

```

Δεν έχουμε το close.

```

for (int j = 0; j <= 1; j++) {
    if (Day.equals("1") || Close.equals("1"))//Monday
    {
        Monday[Mondaycounter] = Hour;
        Mondaycounter += 1;
    }
    ....
    Day = Close.getString("day");
    Hour = Close.getString("time");
}

```

Όπου στο τέλος της επανάληψης τοποθετείτε για να παρθούν και αυτές οι τιμές.

Μα καλά αν μια μέρα λείπει;

Όπως αναφέρθηκα παραπάνω ο πίνακας με τις μέρες έχει παντού 9999 αν κάποια μέρα λείπει τότε από μόνη της έχει το αριθμό και δεν χρειάζεται καμιά επιπρόσθετη ενεργεία.

Και αν δεν έχουμε αποτέλεσμα;

Τότε θα πάμε στο catch όπου αυτό από μόνο του έχω τοποθετήσει κώδικα για να δώσεις τις τιμές στους πίνακες ημερών εβδομάδας.

```

catch (JSONException e) {
    //24 hour
    GetStringShopSuggested gsss = new GetStringShopSuggested();
    gsss.Close();
    e.printStackTrace();
}

```

```

private void setText() {
    String Type = "";
    String streamschedule = "";
    try {
        typeofplace = typeofmark.get(position);
        if (typeofmark.get(position) == 1) Type = "Gas station";
        if (typeofmark.get(position) == 2) Type = "Movie Theater";
        if (typeofmark.get(position) == 3) Type = "Grocery Or Supermarket";
    } catch (IndexOutOfBoundsException e) {
        Toast.makeText(getApplicationContext(), "Something Went Wrong Returned
back", Toast.LENGTH_LONG).show();
        Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
        SetMeeting.this.startActivity(intent);
        SetMeeting.this.finish();
    }
    TextView WS = (TextView) findViewById(R.id.WeekendSchedule);

```

Σε αυτήν την μέθοδο όπως βλέπουμε ετοιμαζόμαστε για να δείξουμε στο *WeekendSchedule* TextView τα δεδομένα, για αυτό θα χρειαστεί να εμφανίσουμε ένα πλήρως μήνυμα με το είδος του κλπ, όσον αφορά το είδος θα το μάθουμε από την θέση του σημείου συνάντησης που επιλέξαμε, αυτή η θέση μπορεί να βρεθεί εύκολα ξέροντας το όνομα.

```

if (checkernull == 0) {
    String temp = weekSche.toString().replace("[", "");
    temp = temp.replace("]", "");
    temp = temp.replace("\\\"", "");
    temp = temp.replace("T", "\\nT");
    temp = temp.replace("W", "\\nW");
    temp = temp.replace("F", "\\nF");
    temp = temp.replace("S", "\\nS");
    WS.setText(Type + ":" + Title + "\\n\\n" + "Vicinity:" + Vicinity + "\\n\\n" +
"Distance:" + Distance + "\\n\\n" + "Duration:" + Duration + "\\n\\n" +
"Suggested By Google\\n" + temp);
    weekSche = new JSONArray();
}

```

```

},
"weekday_text" : [
    "Monday: 7:00 AM - 9:30 PM",
    "Tuesday: 7:00 AM - 9:30 PM",
    "Wednesday: 7:00 AM - 9:30 PM",
    "Thursday: 7:00 AM - 9:30 PM",
    "Friday: 7:00 AM - 9:30 PM",
    "Saturday: 7:00 AM - 9:30 PM",
    "Sunday: Closed"

```

Για να αποφύγουμε αυτήν την ασχημία σε μήνυμα κάνουμε τα κατάλληλα replace και εμφανίζουμε.

```

else if (checkernull == 1) {
    String temp = "";
    if (typeofmark.get(position) == 1) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Gas_Station();
    }
    if (typeofmark.get(position) == 2) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Movie_Theater();
    }
    if (typeofmark.get(position) == 3) {
        GetStringShopSuggested gsss = new GetStringShopSuggested();
        temp = gsss.Grocery_Or_Supermarket();
    }
    WS.setText(Type + ":" + Title + "\n\n" + "Vicinity:" + Vicinity + "\n\n" +
    "Distance:" + Distance + "\n\n" + "Duration:" + Duration + "\n\n" + temp);
}
}
}

```

Αν από την άλλη δεν έχουμε πληροφορίες από το google περνούμε GetStringShopSuggested από την κλάση τις δίκες μου πληροφορίες σχετικά με αυτό και μετά εμφανίζουμε.

Grocery Or Supermarket:ΚΟΚΚΙΝΟ ΠΙΠΕΡΙ	Grocery Or Supermarket:SUPERMARKET SPANOS, J. CH. GROUP OF COMPA
Vicinity:111, Amfitritis, Rodos	Vicinity:Megalou Konstantinou 53, Rodos
Distance:7 m	Distance:0.8 km
Duration:1 min	Duration:3 mins
Suggested By Me: "Monday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM", "Tuesday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM", "Wednesday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM", "Thursday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM", "Friday: 8:00 AM – 14:00 PM, 5:00 – 9:00 PM", "Saturday: 8:00 AM – 14:00 PM", "Sunday: Closed"	Suggested By Google Monday: 7:00 AM – 9:30 PM, Tuesday: 7:00 AM – 9:30 PM, Wednesday: 7:00 AM – 9:30 PM, Thursday: 7:00 AM – 9:30 PM, Friday: 7:00 AM – 9:30 PM, Saturday: 7:00 AM – 9:30 PM, Sunday: Closed

```
private void getplaceinfo() {
    GetPlaceSchedule gps = new GetPlaceSchedule();
    gps.updatePlaces(PlacesId.get(position));
```

Σε αυτό σημείο βλέπουμε να παίρνουμε το PlaceSchedule και παρακάτω με τον handler περιμένουμε κάποια δευτερόλεπτα ώστε να ολοκληρωθεί το query για να είναι εφικτή η συνέχεια.

```
Handler handler = new Handler();
handler.postDelayed(new Runnable() {
    @Override
    public void run() {

        try {

            if (checkernull == 0)
                weekSche = types.getJSONArray("weekday_text");
            setText();
            typeofmark.removeAll(typeofmark);
            PlacesName.removeAll(PlacesName);
            PlacesId.removeAll(PlacesId);
            getDetails();
        }

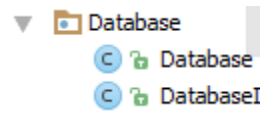
        },
        "weekday_text" : [
            "Monday: 7:00 AM - 9:30 PM",
            "Tuesday: 7:00 AM - 9:30 PM",
            "Wednesday: 7:00 AM - 9:30 PM",
            "Thursday: 7:00 AM - 9:30 PM",
            "Friday: 7:00 AM - 9:30 PM",
            "Saturday: 7:00 AM - 9:30 PM",
            "Sunday: Closed"
```

Με αυτό αποκτούμε το text και το εμφανίζουμε με την μέθοδο setText δεν ξεχνάμε να διαγράψουμε από τους πίνακες λίστες τις τιμές και με το getDetails διαβιβάζουμε τις τιμές στους πίνακες των ημερών της εβδομάδας.

```
catch (JSONException e) {
    .....
} catch (NullPointerException e) {
    e.printStackTrace();
    Toast.makeText(getApplicationContext(), "Something Went Wrong
Returned back", Toast.LENGTH_LONG).show();
    Intent intent = new Intent(SetMeeting.this, PlaceAndIntrests.class);
    SetMeeting.this.startActivity(intent);
    SetMeeting.this.finish();
}
```

Σε περίπτωση λάθους πηγαίνω πάλι πίσω στην Placeandinstrests.

Class Database



```
package Database;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class Database extends SQLiteOpenHelper {

    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "MeetinsDatabase.db";
    private static final String TABLE_NAME = "Meetins";
    private static final String COLUMN_Year = "Year";
    private static final String COLUMN_Month = "Month";
    private static final String COLUMN_Day = "Day";
    private static final String COLUMN_Hour = "Hour";
    private static final String COLUMN_Minute = "Minute";
    private static final String COLUMN_LongLatofplace = "LongLatofplace";
    private static final String COLUMN_Title = "Title";
    private static final String COLUMN_Vicinity = "Vicinity";
    private static final String COLUMN_Distance = "Distance";
    private static final String COLUMN_Duration = "Duration";
    private static final String COLUMN_Delay = "Delay";
    SQLiteDatabase db;
    private static final String TABLE_CREATE = "create table " + TABLE_NAME
+ " (id integer primary key not null , " +
    "Year text not null , Month text not null , Day text not null , Hour text not
null" + " , Minute text not null , LongLatofplace text not null , Title text
not null , " + "Vicinity text not null , Distance text not null , Duration text
not null , Delay text not null)";

    public Database(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(TABLE_CREATE);
        this.db = db;
    }
}
```

```

public void insert(String Year, String Month, String Day, String Hour, String
Minute, String LongLatofplace,
                String Title, String Vicinity, String Distance, String Duration, String
delay) {

```

```

    db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    String query = "select * from " + TABLE_NAME;
    db.rawQuery(query, null);
    values.put(COLUMN_Year, Year);
    values.put(COLUMN_Month, Month);
    values.put(COLUMN_Day, Day);
    values.put(COLUMN_Hour, Hour);
    values.put(COLUMN_Minute, Minute);
    values.put(COLUMN_LongLatofplace, LongLatofplace);
    values.put(COLUMN_Title, Title);
    values.put(COLUMN_Vicinity, Vicinity);
    values.put(COLUMN_Distance, Distance);
    values.put(COLUMN_Duration, Duration);
    values.put(COLUMN_Delay, delay);
    db.insert(TABLE_NAME, null, values);
    db.close();
}

```

@Override

```

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    String query = "DROP TABLE IF EXISTS " + TABLE_NAME;
    db.execSQL(query);

    this.onCreate(db);
}

```

//json array include jason object cause its many to return

```

public JSONArray Search() {
    db = this.getReadableDatabase();
    String query = "SELECT * FROM " + TABLE_NAME + " ORDER BY
Year, Month, Day, Hour, Minute ASC;";
    Cursor cursor = db.rawQuery(query, null);
    String Year;
    String Month;
    String Day;
    String Hour;
    String Minute;
    String id;
    String LongLatofplace;
    String Title;
    String Vicinity;
    String Distance;
}

```

```

String Duration;
String Delay;
JSONArray results = new JSONArray();
if (cursor.moveToFirst()) {
    do {
        id = cursor.getString(0);
        Year = cursor.getString(1);
        Month = cursor.getString(2);
        Day = cursor.getString(3);
        Hour = cursor.getString(4);
        Minute = cursor.getString(5);
        LongLatofplace = cursor.getString(6);
        Title = cursor.getString(7);
        Vicinity = cursor.getString(8);
        Distance = cursor.getString(9);
        Duration = cursor.getString(10);
        Delay = cursor.getString(11);
        Log.d("delay", Delay);
        JSONObject UserInfo = new JSONObject();
        try {
            UserInfo.put("Year", Year);
            UserInfo.put("Month", Month);
            UserInfo.put("Day", Day);
            UserInfo.put("Hour", Hour);
            UserInfo.put("Minute", Minute);
            UserInfo.put("Title", Title);
            UserInfo.put("Vicinity", Vicinity);
            UserInfo.put("Delay", Delay);
            UserInfo.put("Id", id);

            results.put(UserInfo);

        } catch (JSONException e) {
            e.printStackTrace();
        }
    } while (cursor.moveToNext());

}
db.close();
return results;
}

//google map set meeting back json object cause its one
public JSONObject getinfomation(int id) {
    db = this.getReadableDatabase();
    String query = "select * from " + TABLE_NAME + " where id = " +
Integer.toString(id);

    Cursor cursor = db.rawQuery(query, null);
    String id_db;

```

```

String LongLatofplace;
String Title;
String Vicinity;
String Distance;
String Duration;
if (cursor.moveToFirst()) {
    id_db = cursor.getString(0);
    LongLatofplace = cursor.getString(6);
    Title = cursor.getString(7);
    Vicinity = cursor.getString(8);

    JSONObject UserInfo = new JSONObject();
    try {

        UserInfo.put("Title", Title);
        UserInfo.put("Vicinity", Vicinity);
        UserInfo.put("LongLatofplace", LongLatofplace);
        db.close();
        return UserInfo;

    } catch (JSONException e) {
        e.printStackTrace();
    }
}
db.close();
return null; //no way hapen just in case
}

//for notification ,clock
public JSONObject getinfomationtime(int id) {
    db = this.getReadableDatabase();
    String query = "select * from " + TABLE_NAME + " where id = " +
Integer.toString(id);

    Cursor cursor = db.rawQuery(query, null);
    String Year;
    String Month;
    String Day;
    String Hour;
    String Minute;

    if (cursor.moveToFirst()) {

        Year = cursor.getString(1);
        Month = cursor.getString(2);
        Day = cursor.getString(3);
        Hour = cursor.getString(4);
        Minute = cursor.getString(5);
    }
}

```



```

JSONObject UserInfo = new JSONObject();
try {

    UserInfo.put("Year", Year);
    UserInfo.put("Month", Month);
    UserInfo.put("Day", Day);
    UserInfo.put("Hour", Hour);
    UserInfo.put("Minute", Minute);
    return UserInfo;

} catch (JSONException e) {
    e.printStackTrace();
}
}
db.close();
return null;//no way hapen just in case
}

public void delete(int Id) {
    db = this.getWritableDatabase();
    String query = "delete from " + TABLE_NAME + " where id = " +
Integer.toString(Id);
    db.execSQL(query);
    db.close();
}
}

```

Αυτή η κλάση αφορά την βάση και μεθόδους αποθήκευσης, αναζήτησης προϊόντων.

```

private static final int DATABASE_VERSION = 1;
private static final String DATABASE_NAME = "MeetinsDatabase.db";
private static final String TABLE_NAME = "Meetins";
private static final String COLUMN_Year = "Year";

```

Database Version η έκδοση της βάσης ,Database Name το όνομα,Table Name πολύ σημαντικό είναι το πινακάκι/τραπέζι όπου θα αποθηκεύονται οι τιμές και παρακάτω έχουμε τις στήλες δηλαδή τις πληροφορίες που θα αποθηκεύσουμε.

```

private static final String TABLE_CREATE = "create table " + TABLE_NAME +
" (id integer primary key not null , " +
"Year text not null , Month text not null , Day text not null , Hour text not
null" +
" , Minute text not null , LongLatofplace text not null , Title text not null , "
+
"Vicinity text not null , Distance text not null , Duration text not null ,
Delay text not null)";

```

Η εντολή σε string μορφή για την δημιουργία του table.

Id	Year	Month	Day	Hour	Minute	LongLat	Title	Vicinity	Distance	Duration	Delay
0	2009	1	3	5	02	22.1,22.3	Exa1	Exa3	14	2	40
1	2016	2	3	15	24	22.7,22.6	Exa2	Exa4	15	3	20

```
public Database(Context context) {  
    super(context, DATABASE_NAME, null, DATABASE_VERSION);  
}
```

@Override

```
public void onCreate(SQLiteDatabase db) {  
    db.execSQL(TABLE_CREATE);  
    this.db = db;  
}
```

Οι μέθοδοι που παρουσιάζονται βοηθούν στην δημιουργία βάσης και table.

```
public void insert(String Year, String Month, String Day, String Hour, String Minute,  
String LongLatofplace,  
String Title, String Vicinity, String Distance, String Duration, String  
delay) {
```

```
    db = this.getWritableDatabase();  
    ContentValues values = new ContentValues();  
    String query = "select * from " + TABLE_NAME;  
    db.rawQuery(query, null);  
    values.put(COLUMN_Year, Year);  
    .....  
    values.put(COLUMN_Delay, delay);  
    db.insert(TABLE_NAME, null, values);  
    db.close();  
}
```

Σε αυτό το σημείο/μέθοδο βλέπουμε την εισαγωγή στην βάση, όπου στην αρχή **db = this.getWritableDatabase();** την κάνουμε εγγράψιμη **ContentValues values = new ContentValues();** δημιουργούμε την μεταβλητή όπου θα μπουν οι τιμές **db.insert(TABLE_NAME, null, values); db.close();** Κάνουμε Insert και κλείνουμε την βάση.

Προσοχή όπως τα βάζουμε θα μπουν με μια σειρά και πρέπει να την λάβουμε υπόψη στην αναζήτηση.

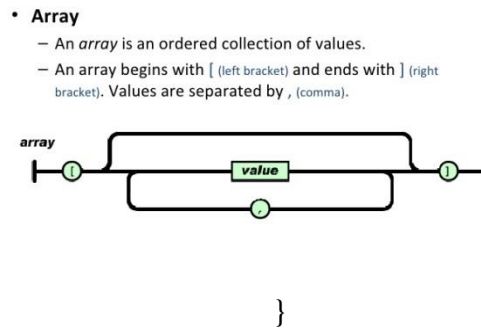
Και στο id επειδή είναι primary key not null αυτόματα γίνεται auto incremental Και δεν χρειάζεται να μπει κάποια τιμή.

```

public JSONArray Search() {
    db = this.getReadableDatabase();
    String query = "SELECT * FROM " + TABLE_NAME + " ORDER BY Year,
Month, Day, Hour, Minute ASC;";
    Cursor cursor = db.rawQuery(query, null);
}

```

Syntax of JSON



Εδώ γίνεται η αναζήτηση και η επιστροφή τιμών σε Json Array γιατί σε αυτό; Διότι μου ήταν ευκολότερο στην αποκωδικοποίηση της κάθε τιμής παρά ένα συνολικό string με / και άλλους διαφόρους χαρακτήρες ή λίστες.

```

String Year;
...
String Delay;
JSONArray results = new JSONArray();
if (cursor.moveToFirst()) {
    do {
        id = cursor.getString(0);
        Year = cursor.getString(1);
        Month = cursor.getString(2);
        Day = cursor.getString(3);
        ....
        Log.d("delay", Delay);
        JSONObject UserInfo = new JSONObject();
        try {
            UserInfo.put("Year", Year);
            UserInfo.put("Month", Month);
            ....
            results.put(UserInfo);
        } catch (JSONException e) {
            e.printStackTrace();
        }
    } while (cursor.moveToNext());
}

```

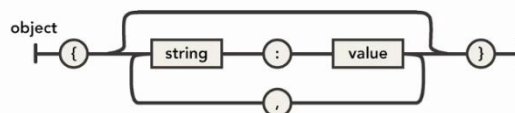
```

    }
    db.close();
    return results;
}

```

Cursor είναι ο δείκτης μπροστά από κάθε γραμμή του table πως μπαίνει;; όταν εκτελώ το query `Cursor cursor = db.rawQuery(query, null);` η επιστροφή είναι ένας Cursor , `moveToFirst` πάει στην πρώτη γραμμή όπου διαβάζω τις τιμές με `cursor.getString(3);` και αφού διαβάσω τις τιμές δημιουργώ ένα JSONObject `UserInfo = new JSONObject();` Json Object

JSON objects



Όπου θα του περάσω τις τιμές τις κάθε γραμμής `UserInfo.put("Year", Year);` Στο τέλος δεν ξεχνάμε να κλείνουμε την βάση `db.close();` για leaks.

```

public JSONObject getinfomation(int id) {
    db = this.getReadableDatabase();
    String query = "select * from " + TABLE_NAME + " where id = " +
Integer.toString(id);

```

```

    Cursor cursor = db.rawQuery(query, null);
    String id_db;
    String LongLatofplace;
    String Title;
    String Vicinity;
    if (cursor.moveToFirst()) {
        LongLatofplace = cursor.getString(6);
        ...
        JSONObject UserInfo = new JSONObject();
        try {
            UserInfo.put("Title", Title);
            ...
            db.close();
            return UserInfo;

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    db.close();
    return null;//no way hapen just in case
}

```

Παρόμοια μέθοδος με την παραπάνω δεν έχει κάτι διαφορετικό είναι ποιο απλοποιημένη μονό σε Json Object διότι χρειάζεται οδηγίες για μια μονό γραμμή μέσα στο table.

```

public JSONObject getInformationtime(int id) {
    db = this.getReadableDatabase();
    String query = "select * from " + TABLE_NAME + " where id = " +
Integer.toString(id);

    Cursor cursor = db.rawQuery(query, null);
    String Year;
    ...
    if (cursor.moveToFirst()) {
        Year = cursor.getString(1);
    ...
    JSONObject UserInfo = new JSONObject();
    try {

        UserInfo.put("Year", Year);
    ...
        return UserInfo;
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
db.close();
return null; //no way hapen just in case
}

```

Παρόμοια μέθοδος με την `getInformation` δεν έχει κάτι διαφορετικό. Χρησιμοποιεί για την εύρεση ώρας μιας γραμμής μέσα στο πινάκα δηλαδή κάποιου σημείου ενδιαφέροντος.

```

public void delete(int Id) {
    db = this.getWritableDatabase();
    String query = "delete from " + TABLE_NAME + " where id = " +
Integer.toString(Id);
    db.execSQL(query);
    db.close();
}

```

Εδώ γίνεται η διαγραφή το query είναι γνωστό, εκτελούμε και κλείνουμε την βάση.

Class DatabaseInterface

```
package Database;
import android.app.Activity;
import android.util.Log;
import android.widget.Toast;
import org.joda.time.DateTime;
import org.joda.time.Days;
import org.joda.time.Minutes;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import static SetMeetings.SetMeetingsVariables.Friday;
import static SetMeetings.SetMeetingsVariables.Monday;
import static SetMeetings.SetMeetingsVariables.Saturday;
import static SetMeetings.SetMeetingsVariables.Sunday;
import static SetMeetings.SetMeetingsVariables.Thursday;
import static SetMeetings.SetMeetingsVariables.Tuesday;
import static SetMeetings.SetMeetingsVariables.Wednesday;
```

```
public class DatabaseInterface {
    private String Year;
    private String Month;
    private String Day;
    private String Hour;
    private String Minute;
    private String LongLatofplace;
    Database database;
    private Activity activity;

    public void setActivityUi(Activity activity) {
        this.activity = activity;
    }

    public void createdb() {
        database = new Database(activity);
    }

    public boolean insert(String Year, String Month, String Day, String Hour, String
    Minute, String LongLatofplace,
        String Title, String Vicinity, String Distance, String Duration, int
    type, int delay) {
```

```

    if (validate_from_schedule(Year, Month, Day, Hour, Minute, type, delay) ==
false) {
        Toast.makeText(activity, "Check Schedule again",
Toast.LENGTH_SHORT).show();
        return false;
    }

    if (validate_from_database(Year, Month, Day, Hour, Minute, delay) == false) {

        return false;
    }

    database.insert(Year, Month, Day, Hour, Minute, LongLatofplace, Title,
Vicinity, Distance, Duration, String.valueOf(delay));

    return true;
}

private boolean validate_from_database(String Year, String Month, String Day,
String Hour, String Minute, int delay_given) {
    JSONArray object = database.Search();
    for (int i = 0; i < object.length(); i++) {

        try {
            JSONObject placeObject = object.getJSONObject(i);

            //time difference between days
            SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy
HH:mm:ss");
            String inputString1 = placeObject.getString("Month") + "/" +
placeObject.getString("Day") + "/" + placeObject.getString("Year")
                + " " + placeObject.getString("Hour") + ":" +
placeObject.getString("Minute") + ":" + 0;
            String inputString2 = Month + "/" + Day + "/" + Year
                + " " + Hour + ":" + Minute + ":" + 0;
            Date d2 = null;
            Date d1 = null;
            try {
                d2 = format.parse(inputString2);
                d1 = format.parse(inputString1);
            } catch (ParseException e) {
                e.printStackTrace();
            }
            DateTime dt1 = new DateTime(d1);
            DateTime dt2 = new DateTime(d2);

            int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes();
            if (minutes >= 0) {
                int delay = Integer.parseInt(placeObject.getString("Delay"));

```

```

        if (minutes <= delay)
            if (Days.daysBetween(dt1, dt2).getDays() == 0) {
                String message_colission = "Meeting Collision " +
                placeObject.getString("Hour") + ":" + placeObject.getString("Minute") + "/" +
                delay;
                Toast.makeText(activity, message_colission,
                Toast.LENGTH_SHORT).show();
                return false;
            }
        }

        minutes = Minutes.minutesBetween(dt2, dt1).getMinutes();
        if (minutes >= 0) {
            if (minutes <= delay_given)
                if (Days.daysBetween(dt2, dt1).getDays() == 0) {
                    String message_colission = "Meeting Collision " +
                    placeObject.getString("Hour") + ":" + placeObject.getString("Minute");
                    Toast.makeText(activity, message_colission,
                    Toast.LENGTH_SHORT).show();
                    return false;
                }
            }
        }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

return true;
}

```

```

private boolean validate_from_schedule(String Year, String Month, String Day,
String Hour, String Minute, int type, int delay) {

```

```

    //get day of week
    Date date = null;
    SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy");
    String inputString2 = Month + "/" + Day + "/" + Year;
    try {
        date = format.parse(inputString2);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    Calendar c = Calendar.getInstance();
    c.setTime(date);
    int dayOfWeek = c.get(Calendar.DAY_OF_WEEK);

```



```

List list = new ArrayList(Arrays.asList(Sunday));
list.addAll(Arrays.asList(Monday));
list.addAll(Arrays.asList(Tuesday));
list.addAll(Arrays.asList(Wednesday));
list.addAll(Arrays.asList(Thursday));
list.addAll(Arrays.asList(Friday));
list.addAll(Arrays.asList(Saturday));
Object[] WeekSche = list.toArray();
// WeekSche [9999, 9999, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100,
9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100, 9999,
9999, 0800, 2000, 9999, 9999]
if (dayOfWeek == 1)
    dayOfWeek = 0;
else
    dayOfWeek -= 1;
int TimeSche1 = Integer.parseInt((String) WeekSche[(dayOfWeek * 4)]);
int TimeSche2 = Integer.parseInt((String) WeekSche[1 + (dayOfWeek * 4)]);
int TimeSche3 = Integer.parseInt((String) WeekSche[2 + (dayOfWeek * 4)]);
int TimeSche4 = Integer.parseInt((String) WeekSche[3 + (dayOfWeek * 4)]);
Log.d("testa", String.valueOf(TimeSche1));
Log.d("testa", String.valueOf(TimeSche2));
Log.d("testa", String.valueOf(TimeSche3));
Log.d("testa", String.valueOf(TimeSche4));

if (TimeSche1 == 7777) {
    //open 24 hour
    return true;
}

if (TimeSche1 == 9999 && TimeSche2 == 9999 && TimeSche3 == 9999 &&
TimeSche4 == 9999) {
    Toast.makeText(activity, "Closed At This Date",
Toast.LENGTH_LONG).show();
    return false;
}

int hour_int = Integer.valueOf(Hour);
// if(hour_int==0) hour_int=24;
int minute_int = Integer.valueOf(Minute);
int time = hour_int * 100 + minute_int;
// day alteration // day alteration // day alteration // day alteration //
day alteration
if (TimeSche1 > TimeSche2) {
    //copie them for test for new day alteration
    int hour_int_new = hour_int;
    int minute_int_new = minute_int;

    minute_int_new = minute_int_new + delay;
    //if overflow this hour then you are on new hour cause 23:99 NOT EXIST
    if (minute_int_new >= 60) {

```

```

    hour_int_new = hour_int_new + 1;
    minute_int_new = minute_int_new - 60;
}
if (hour_int_new >= 24)
    hour_int_new = hour_int_new - 24;
int time_new = hour_int_new * 100 + minute_int_new;

if (TimeSche1 < time_new && time_new < 2400)
    return true;
if (0 < time_new && time_new < TimeSche2)
    return true;

}
// day alteration // day alteration // day alteration // day alteration //
day alteration
if (TimeSche3 > TimeSche4) {
    //copie them for test for new day alteration
    int hour_int_new = hour_int;
    int minute_int_new = minute_int;

    minute_int_new = minute_int_new + delay;
    //if overflow this hour then you are on new hour cause 23:99 NOT EXIST
    if (minute_int_new >= 60) {
        hour_int_new = hour_int_new + 1;
        minute_int_new = minute_int_new - 60;
    }
    if (hour_int_new >= 24)
        hour_int_new = hour_int_new - 24;
    int time_new = hour_int_new * 100 + minute_int_new;

    if (TimeSche3 < time_new && time_new < 2400)
        return true;
    if (0 < time_new && time_new < TimeSche4)
        return true;

}

//build for no day alteration //build for no day alteration //build for no day alteration
//build for no day alteration
if (time >= TimeSche1 && time <= TimeSche2 || time >= TimeSche3 && time
<= TimeSche4) {
    minute_int = minute_int + delay;
    if (minute_int >= 60) {
        hour_int += 1;
        minute_int = minute_int - 60;
    }
    time = hour_int * 100 + minute_int;
    if (time >= TimeSche1 && time <= TimeSche2 || time >= TimeSche3 &&
time <= TimeSche4) {

```

```

        return true;
    } else {
        Toast.makeText(activity, "Delay Too much",
Toast.LENGTH_LONG).show();
        return false;
    }

    } else {
        Toast.makeText(activity, "Closed At This Hour",
Toast.LENGTH_LONG).show();
        return false;
    }

}

public JSONArray Search() {
    return database.Search();
}

public JSONObject getinfomation(int id) {
    return database.getinfomation(id);
}

public JSONObject getinfomationtime(int id) {
    return database.getinfomationtime(id);
}

public void delete(int Id) {
    database.delete(Id);
}
}

```

Αυτή η κλάση είναι ένα βήμα πριν την κανονική βάση αυτήν περισσότερο την έκανα ώστε να έχω εδώ τα validations και να μην φορτώνω την κλάση με την βάση.

```

private String Year;
private String Month;
private String Day;
private String Hour;
private String Minute;
private String LongLatofplace;
Database database;
private Activity activity;

```

Οι πρώτες μεταβλητές όπως βλέπουμε περιγράφουν την ημερομηνία ώρα long lat, μετά έχουμε την μεταβλητή της βάσης **database** και τέλος το **activity**.

```

public void setActivityUi(Activity activity) {
    this.activity = activity;
}

```

```

public void createdb() {
    database = new Database(activity);
}

```

Όταν προσπαθώ να φτιάξω την βάση περνώ και το activity.

```

public boolean insert(String Year, String Month, String Day, String Hour, String
Minute, String LongLatofplace,
    String Title, String Vicinity, String Distance, String Duration, int type,
int delay) {

    if (validate_from_schedule(Year, Month, Day, Hour, Minute, type, delay) == false)
    {
        Toast.makeText(activity, "Check Schedule again",
Toast.LENGTH_SHORT).show();
        return false;
    }

    if (validate_from_database(Year, Month, Day, Hour, Minute, delay) == false) {

        return false;
    }

    database.insert(Year, Month, Day, Hour, Minute, LongLatofplace, Title, Vicinity,
Distance, Duration, String.valueOf(delay));

    return true;
}

```

Σε αυτήν μέθοδο γίνεται το Insert όπου εδώ περνάμε τα στοιχεία μας και ελέγχουμε μέσω των μεθόδων validate αν μπορούμε να κάνουμε την εισαγωγή. Επίσης ένα καλό παράδειγμα που χρειαζόμαστε το activity.

```

Toast.makeText(activity, "Check Schedule again",
Toast.LENGTH_SHORT).show();

```

```
private boolean validate_from_schedule(String Year, String Month, String Day,
String Hour, String Minute, int type, int delay) {
```

Πριν κάνουμε την εισαγωγή στην βάση πρέπει να ελέγξουμε φυσικά αν όντως ισχύει με το πρόγραμμα του καταστήματος σε αυτήν την μέθοδο θα δούμε αυτό το πράμα.

```
//get day of week
Date date = null;
SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy");
String inputString2 = Month + "/" + Day + "/" + Year;
try {
    date = format.parse(inputString2);
} catch (ParseException e) {
    e.printStackTrace();
}
```

Αυτό είναι από την βιβλιοθήκη της joda όπου εδώ δημιουργώ σε ένα string μια διαμόρφωση για την ημερομηνία όπου έχει δώσει ο χρήστης.

```
Calendar c = Calendar.getInstance();
c.setTime(date);
int dayOfWeek = c.get(Calendar.DAY_OF_WEEK);
```

Παίρνω την ημερομηνία date που φτιάξαμε από τον χρήστη την βάζω στο ημερολόγιο και έτσι παίρνω την ημέρα της εβδομάδος.

```
List list = new ArrayList(Arrays.asList(Sunday));
list.addAll(Arrays.asList(Monday));
list.addAll(Arrays.asList(Tuesday));
list.addAll(Arrays.asList(Wednesday));
list.addAll(Arrays.asList(Thursday));
list.addAll(Arrays.asList(Friday));
list.addAll(Arrays.asList(Saturday));
Object[] WeekSche = list.toArray();
```

Βάζω τις τιμές όλης της εβδομάδας σε μια τέτοια μορφή , WeekSche [9999, 9999, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2100, 9999, 9999, 0800, 2000, 9999, 9999].

```

if (dayOfWeek == 1)
    dayOfWeek = 0;
else
    dayOfWeek -= 1;
int TimeSche1 = Integer.parseInt((String) WeekSche[(dayOfWeek * 4)]);
int TimeSche2 = Integer.parseInt((String) WeekSche[1 + (dayOfWeek * 4)]);
int TimeSche3 = Integer.parseInt((String) WeekSche[2 + (dayOfWeek * 4)]);
int TimeSche4 = Integer.parseInt((String) WeekSche[3 + (dayOfWeek * 4)]);

```

Εδώ είναι λίγο περίεργα τα πράγματα όμως είναι πολύ απλά , όταν στην google περνάμε το πρόγραμμα με τις ημέρες/ώρες του καταστήματος η google για κάθε ημέρα η google έδινε ένα αριθμό

0 Κυριακή

1 δευτέρα

2 τρίτη κλπ.

Εγώ όμως θεώρησα λίγο το αντίθετο

0 δευτέρα

1 τρίτη

Έτσι η Κυριακή βρίσκεται στο τέλος

Επίσης η αρίθμηση του πίνακα ξεκινά από το 0 για αυτό

```

if (dayOfWeek == 1)
    dayOfWeek = 0;

```

Τα TimeSche βρίσκουν μέσα στον πίνακα της ώρες της ημέρας που ψάχνουμε πχ θέλουμε τρίτη

Τότε dayOfWeek=1

Και τα στοιχειά βρίσκονται

Στις έξις θέσης 4,5,6,7 αν όμως θέλαμε δευτέρα 0,1,2,3

Μπορούσα και εγώ να ξεκινούσα όπως η google με πρώτη ημέρα Κυριακή αλλά δεν το προτίμησα διότι στα μαγαζιά η πρώτη μέρα ως συνήθως είναι η δευτέρα.

```

if (TimeSche1 == 7777) {

```

```

    return true;

```

```

}

```

```

if (TimeSche1 == 9999 && TimeSche2 == 9999 && TimeSche3 == 9999 &&
TimeSche4 == 9999) {

```

```

    Toast.makeText(activity, "Closed At This Date",

```

```

    Toast.LENGTH_LONG).show();

```

```

    return false;

```

```

}

```

Όπως αναφέρθηκα αρκετά παραπάνω αν έχουμε 7777 τότε είναι ανοικτό όλη μέρα, δεν χρειάζεται να ελέγξουμε πτπ και έτσι επιστρέφουμε αν όντως πχ. ψάχνουμε για τρίτη που είναι όλη μέρα ανοικτά. Αν από την άλλη συναντήσουμε 9999 τότε είναι όλη μέρα κλειστά.

Τώρα αυτό που μένει είναι να συγκρίνουμε την ώρα με τα TimeSche και αν είναι ανάμεσα τότε είμαστε οκ δεν ξεχνάμε έχουμε είδη μετατρέψει τα string σε integer λίγο ποιο πάνω.

```
//build for no day alteration
```

```
    if (time >= TimeSche1 && time <= TimeSche2 || time >= TimeSche3 && time
<= TimeSche4) {
        minute_int = minute_int + delay;
        if (minute_int >= 60) {
            hour_int += 1;
            minute_int = minute_int - 60;
        }
        time = hour_int * 100 + minute_int;
        if (time >= TimeSche1 && time <= TimeSche2 || time >= TimeSche3 &&
time <= TimeSche4) {
            return true;
        } else {
            Toast.makeText(activity, "Delay Too much",
Toast.LENGTH_LONG).show();
            return false;
        }
    } else {
        Toast.makeText(activity, "Closed At This Hour",
Toast.LENGTH_LONG).show();
        return false;
    }
}
```

Αυτή η If δεν αφορά την αλλαγή της ημέρας δηλαδή είναι για ωράρια του τύπου 7-8 το πρωί 2-9 το απόγευμα.

```
time >= TimeSche1 && time <= TimeSche2 || time >= TimeSche3 && time <=
TimeSche4
```

Πχ Έχουμε την ώρα 8 50 απόγευμα η πρωί και έχουμε πάρει το ωράριο τι θα γίνει:

- 1) Ωράριο Έστω 8πμ-2μμ και 5μμ-9μμ
 $850 >= 800$ ναι είναι και $850 <= 1400$ ναι
Αρά περνά
Αν είχαμε απόγευμα
- 2) Ωράριο Έστε 8πμ-2μμ και 5μμ-9μμ
 $2050 >= 800$ όχι αρά αποκλείουμε την πρώτη και πάμε στην δεύτερη συνθήκη
 $2050 >= 1700$ ναι είναι και $2050 <= 2100$ ναι είναι αρά οκ

Με αυτό τον τρόπο λοιπόν έλεγχο την συνθήκη

Στην συνέχεια μεταβιβάζουμε το delay μέσα στην ώρα μας που είναι αριθμός με το κατάλληλο τρόπο

Πχ

Έχουμε 1220 και delay 50 το αποτέλεσμα θα γίνει

$50 + 20 = 70$ $70 = 1$ ώρα και 10 λεπτά αρά θα αυξήσω την ώρα +1 και τα λεπτά 70-60

Τελικό 1310 ($12 + 1 = 13$ $70 - 60 = 10$) όπου θα γίνει ο έλεγχος άλλη μια φορά.

```

int hour_int = Integer.valueOf(Hour);
// if(hour_int==0) hour_int=24;
int minute_int = Integer.valueOf(Minute);
int time = hour_int * 100 + minute_int;
// day alteration
if (TimeSche1 > TimeSche2) {
    //copie them for test for new day alteration
    int hour_int_new = hour_int;
    int minute_int_new = minute_int;

    minute_int_new = minute_int_new + delay;
    //if overflow this hour then you are on new hour cause 23:99 NOT EXIST
    if (minute_int_new >= 60) {
        hour_int_new = hour_int_new + 1;
        minute_int_new = minute_int_new - 60;
    }
    if (hour_int_new >= 24)
        hour_int_new = hour_int_new - 24;
    int time_new = hour_int_new * 100 + minute_int_new;

    if (TimeSche1 < time_new && time_new < 2400)
        return true;
    if (0 < time_new && time_new < TimeSche2)
        return true;
}

```

Έστω λοιπόν ότι στο πρόγραμμα έτυχε και έχουμε ώρα 0200 και 1200 δηλαδή 2 πρωί 12 μεσημέρι τότε εκτελούμε αυτήν την if .Για αρχή περνούμε την ώρα στο time (2:00->2*100+0=200) και υπολογίζουμε πάλι τα νέα λεπτά με το delay που έβαλε ο χρήστης ,αν τα λεπτά κάνουν overflow το 60 κλασικά ώρα +1 και λεπτά λεπτα_νεα-60 αφού κάνουμε αυτά οδηγούμαστε στην if ,Πχ ψάχνουμε την ώρα 3 το πρωί

- 1) Ωράριο 12 μεσημέρι 2 πρωί
200<300 ναι είναι το 300<2400 ναι αρά είμαστε εντάξει
- 2) Ωραίο 11 το βραδύ 4 πρωί
Η πρώτη if είναι λάθος **if** (TimeSche1 < time_new && time_new < 2400)
2300>300
Πάμε στην δεύτερη 300>0 οκ και 300<400 οκ

Έτσι λοιπόν κοιτώ και ελέγχω την ώρα στις ώρες καταστήματος.

```

if (TimeSche3 > TimeSche4) {
    ....
}

```

Παρόμοιο με το παραπάνω με την έννοια ότι οι ώρα που άλλαζε βρίσκεται αντί στις θέσεις του πίνακα 4,5 στις 6,7.

```

if (hour_int_new >= 24)
    hour_int_new = hour_int_new - 24;

```

Επίσης η ώρα αν περνά το 24ωρο και πάει 25 σημαίνει ότι είναι 1 αρά 25-24.


```
private boolean validate_from_database(String Year, String Month, String Day,
String Hour, String Minute, int delay_given) {
    JSONArray object = database.Search();
```

Αυτή είναι η κλάση όπου ελέγχουμε από τα meeting μας, για αρχή περνούμε τις πληροφορίες από την βάση.

```
    for (int i = 0; i < object.length(); i++) {
```

Και για κάθε meeting όσο το Json Array κάνουμε έλεγχο

```
        try {
            JSONObject placeObject = object.getJSONObject(i);
            και περνούμε την κάθε πληροφορία σε Json Object.
            //time difference between days
            SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy
HH:mm:ss");
            String inputString1 = placeObject.getString("Month") + "/" +
            placeObject.getString("Day") + "/" + placeObject.getString("Year")
                + " " + placeObject.getString("Hour") + ":" +
            placeObject.getString("Minute") + ":" + 0;
            String inputString2 = Month + "/" + Day + "/" + Year
                + " " + Hour + ":" + Minute + ":" + 0;
            Date d2 = null;
            Date d1 = null;
            try {
                d2 = format.parse(inputString2);
                d1 = format.parse(inputString1);
            } catch (ParseException e) {
                e.printStackTrace();
            }
            DateTime dt1 = new DateTime(d1);
            DateTime dt2 = new DateTime(d2);
```

Αυτός ο κώδικας ο οποίος βλέπουμε τώρα είναι χάρις τα joda libraries
Πρώτα καθορίζουμε τι μορφή έχει η ημερομηνία ώρα μας μετά περνάμε τα στοιχεία
από την βάση και μετά τα στοιχεία που δώσω ο χρήστης και αφού τα μετατρέψουμε
σε αντικείμενο ημερομηνίας δημιουργούμε το DateTime.

```

int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes();
if (minutes >= 0) {
    int delay = Integer.parseInt(placeObject.getString("Delay"));
    if (minutes <= delay)
        if (Days.daysBetween(dt1, dt2).getDays() == 0) {
            String message_colission = "Meeting Collision " +
placeObject.getString("Hour") + ":" + placeObject.getString("Minute") + "/" +
delay;
            Toast.makeText(activity, message_colission,
Toast.LENGTH_SHORT).show();
            return false;
        }
    }
}

```

Πριν σχολιάσω καλό θα ήταν να πω ότι η joda επιστρέφει διάφορα ημερών ώρας και λεπτών ανάμεσα σε 2 ημέρες και θα βασιστώ σε αυτό.

Τι βλέπουμε εδώ για αρχή παίρνουμε τα λεπτά από την διάφορα ώρας και ελέγχουμε αν είναι μεγαλύτερη του 0 ή 0.τι εννοούμε με αυτό

Αημ-Βημ=ημερεςΑβ,ωρεςΑβ,μινΑβ

```
if (minutes <= delay)
```

Από εδώ μετά ελέγχουμε την καθυστέρηση αν είναι μεγαλύτερη των λεπτών τότε σημαίνει ότι έχουμε collision meeting η όποια αφορά την δεύτερη ημερομηνία ,τι σημαίνει όμως αυτό; Ότι αν είχαμε βάλει την καθυστέρηση στα λεπτά της δεύτερης ημερομηνίας τότε θα είχαμε κάτι αρνητικό αρά μια νωρίτερα ημερομηνία και για να το επιβεβαιώσω ελέγγω αν είχαμε διάφορα ημερών 0 .

```

minutes = Minutes.minutesBetween(dt2, dt1).getMinutes();
if (minutes >= 0) {
    if (minutes <= delay_given)
        if (Days.daysBetween(dt2, dt1).getDays() == 0) {
            String message_colission = "Meeting Collision " +
placeObject.getString("Hour") + ":" + placeObject.getString("Minute") ;
            Toast.makeText(activity, message_colission,
Toast.LENGTH_SHORT).show();
            return false;
        }
    }
}

```

Έστω ότι έχουμε επιλέξει ένα ραντεβού και το καταχωρήσαμε ο έλεγχος που κάναμε παραπάνω έγινε μονό από την μια μεριά ότι η A>B αν όμως B>A ,σε αυτό λοιπόν κοιτάμε και το ανάστροφο για να έχουμε πλήρες έλεγχο ημερομηνίας και ώρας σε ένα ραντεβού και νωρίτερα και μετά. Αφού όλα αυτά βγουν λάθος τότε προχωράμε στην for ψάχνοντας το επόμενο μέχρι να τελειώσει και να επιστρέψει true.

Layout checkseemeetings

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#254B63"
    android:orientation="vertical">

    <LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/l1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:weightSum="1"
    tools:context="com.tolis.maps3856.CheckSeeMeetings">

        <LinearLayout
            android:id="@+id/l1"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="1">

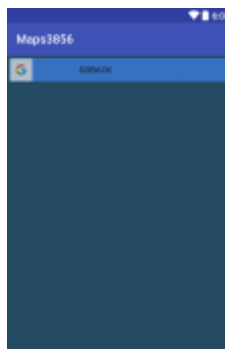
            <Button
                android:alpha="0.65"
                android:id="@+id/goBack"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_weight="0.5"

                android:background="@drawable/common_google_signin_btn_text_dark"
                android:onClick="onButtonClick"
                android:text="GoBack"                ."/>

        </LinearLayout>

    </LinearLayout>

</ScrollView>
```



Σε αυτό το Layout έβαλα ένα Scroll View εξαρχής διότι μετά θα το γεμίσω δυναμικά κουμπιά όπου θα εμφανίζονται τα meetings.

Class CheckSeeMeetings

```
package com.tolis.maps3856;  
import android.app.AlarmManager;  
import android.app.PendingIntent;  
import android.content.DialogInterface;  
import android.content.Intent;  
import android.graphics.Color;  
import android.os.Bundle;  
import android.os.Handler;  
import android.support.v4.app.FragmentActivity;  
import android.support.v7.app.AlertDialog;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.LinearLayout;  
import android.widget.Toast;  
import org.joda.time.DateTime;  
import org.joda.time.Days;  
import org.joda.time.Hours;  
import org.joda.time.Minutes;  
import org.json.JSONArray;  
import org.json.JSONException;  
import org.json.JSONObject;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Calendar;  
import java.util.Date;  
import Database.DatabaseInterface;
```

```

public class CheckSeeMeetings extends FragmentActivity {

    private final int refreshTime = 23000; //23 seconds
    private static int forhandler_old_hour, forhandler_old_days,
    forhandler_old_minute;

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.checkseemeetings);

        DatabaseInterface di = new DatabaseInterface();
        di.setActivityUi(this);
        di.createdb();
        JSONArray object = di.Search();
        for (int i = 0; i < object.length(); i++) {
            LinearLayout ll = (LinearLayout) findViewById(R.id.II);
            // TextView dynamicTextView = new TextView(this);
            final Button button = new Button(this);
            // dynamicTextView.setLayoutParams(new
            LayoutParams(LayoutParams.MATCH_PARENT,
            LayoutParams.WRAP_CONTENT));
            // R.layout.checkseemeetings.
            // TextView WS = (TextView)findViewById(R.id.WeekendSchedule);
            // WS.setText(object.toString());
            try {
                final JSONObject placeObject = object.getJSONObject(i);

                button.setId(Integer.valueOf(placeObject.getString("Id")));
                Log.d("testa", placeObject.getString("Id"));

                button.setTextColor(Color.rgb(220, 250, 250, 250));
                button.setTextSize(20);

                final String Info = placeObject.getString("Year") + "/" +
                placeObject.getString("Month") + "/" + placeObject.getString("Day") + "," +
                placeObject.getString("Hour") + ":" +
                placeObject.getString("Minute") + "," + placeObject.getString("Title") + "," +
                placeObject.getString("Vicinity") + " ,TimeStay " +
                placeObject.getString("Delay");
                Calendar c = Calendar.getInstance();

                //date difference
                SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy
                HH:mm:ss");
                String inputString2 = placeObject.getString("Month") + "/" +
                placeObject.getString("Day") + "/" + placeObject.getString("Year")
                + " " + placeObject.getString("Hour") + ":" +
                placeObject.getString("Minute") + ":" + c.get(Calendar.SECOND);

```

```

        String inputString1 = (c.get(Calendar.MONTH) + 1) + "/" +
c.get(Calendar.DAY_OF_MONTH) + "/" + c.get(Calendar.YEAR)
        + " " + c.get(Calendar.HOUR_OF_DAY) + ":" +
c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND);
        Date d2 = null;
        Date d1 = null;
        try {
            d2 = format.parse(inputString2);
            d1 = format.parse(inputString1);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        DateTime dt1 = new DateTime(d1);
        DateTime dt2 = new DateTime(d2);
        int days = Days.daysBetween(dt1, dt2).getDays();
        int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes() % 60;
        int hours = Hours.hoursBetween(dt1, dt2).getHours() % 24;
        if (Days.daysBetween(dt1, dt2).getDays() < 1) days = 0;
        if (Minutes.minutesBetween(dt1, dt2).getMinutes() % 60 < 1) minutes = 0;
        if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;
        button.setText(Info + "\nD:" + days + " H:" + hours + " M:" + minutes);
        forhandler_old_days = days;
        forhandler_old_minute = minutes;
        forhandler_old_hour = hours;

//update time with handler
        if (forhandler_old_days != 0 || forhandler_old_hour != 0 ||
forhandler_old_minute != 0) {
            final Handler handler = new Handler();

            handler.postDelayed(new Runnable() {
                public void run() {
                    Calendar c = Calendar.getInstance();
                    String inputString2 = "";
                    String inputString1 = "";
                    Date d2 = null;
                    Date d1 = null;
                    try {
                        inputString2 = placeObject.getString("Month") + "/" +
placeObject.getString("Day") + "/" + placeObject.getString("Year")
                        + " " + placeObject.getString("Hour") + ":" +
placeObject.getString("Minute") + ":" + c.get(Calendar.SECOND);
                        inputString1 = (c.get(Calendar.MONTH) + 1) + "/" +
c.get(Calendar.DAY_OF_MONTH) + "/" + c.get(Calendar.YEAR)
                        + " " + c.get(Calendar.HOUR_OF_DAY) + ":" +
c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND);
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            }, 1000);
        }
    }
}

```

```

        SimpleDateFormat format = new
SimpleDateFormat("MM/dd/yyyy HH:mm:ss");
        d2 = format.parse(inputString2);
        d1 = format.parse(inputString1);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    DateTime dt1 = new DateTime(d1);
    DateTime dt2 = new DateTime(d2);
    int days = Days.daysBetween(dt1, dt2).getDays();
    int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes() % 60;
    int hours = Hours.hoursBetween(dt1, dt2).getHours() % 24;
    if (Days.daysBetween(dt1, dt2).getDays() < 1) days = 0;
    if (Minutes.minutesBetween(dt1, dt2).getMinutes() % 60 < 1) minutes
= 0;
    if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;

    button.setText(Info + "\nD:" + days + " H:" + hours + " M:" +
minutes);
    //no reason re run handler if all zero
    if (forhandler_old_days != 0 || forhandler_old_hour != 0 ||
forhandler_old_minute != 0) {
        forhandler_old_days = days;
        forhandler_old_minute = minutes;
        forhandler_old_hour = hours;
        handler.postDelayed(this, refreshTime);
    }
    }, refreshTime);
}
} catch (JSONException e) {
    e.printStackTrace();
}

ll.addView(button);
ClickListener(button);

}

}

public void onButtonClick(View v) {
    if (v.getId() == R.id.goBack) {
        Intent intent = new Intent(CheckSeeMeetings.this, PlaceAndIntrests.class);
        CheckSeeMeetings.this.startActivity(intent);
        CheckSeeMeetings.this.finish();
    }
}
}

```

```

public void ClickListener(final Button buttonOne) {
    buttonOne.setOnClickListener(new Button.OnClickListener() {
        public void onClick(final View v) {
            new AlertDialog.Builder(CheckSeeMeetings.this)
                .setMessage("Delete or Set in Maps?")
                .setCancelable(false)
                .setPositiveButton("Set", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
                        DatabaseInterface di = new DatabaseInterface();
                        di.setActivityUi(CheckSeeMeetings.this);
                        di.createdb();
                        Log.d("testa", Integer.toString(v.getId()));

                        try {
                            JSONObject object = di.getinfromation(v.getId());

                            Intent intent = new Intent(CheckSeeMeetings.this,
PlaceAndIntrests.class);
                            intent.putExtra("Title", object.getString("Title"));
                            intent.putExtra("Vicinity", object.getString("Vicinity"));
                            intent.putExtra("LongLatofplace",
object.getString("LongLatofplace"));
                            intent.putExtra("ActivityBefore", "CheckSeeMeetings");
                            CheckSeeMeetings.this.startActivity(intent);
                            CheckSeeMeetings.this.finish();
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                        dialog.cancel();
                    }
                })

                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.cancel();
                    }
                })

                .setNegativeButton("Delete", new DialogInterface.OnClickListener()
{
                    public void onClick(DialogInterface dialog, int which) {
                        DatabaseInterface di = new DatabaseInterface();
                        di.setActivityUi(CheckSeeMeetings.this);
                        di.createdb();
                        di.delete(buttonOne.getId());
                        dialog.cancel();
                        finish();
                    }
                })
        }
    });
}

```



```

        startActivity(getIntent());
    }
    }).show();
}

});
buttonOne.setOnLongClickListener(new Button.OnLongClickListener() {
    @Override
    public boolean onLongClick(final View v) {
        new AlertDialog.Builder(CheckSeeMeetings.this)
            .setMessage("Set A meeting?")
            .setCancelable(false)
            .setPositiveButton("Set", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    DatabaseInterface di = new DatabaseInterface();
                    di.setActivityUi(CheckSeeMeetings.this);
                    di.createdb();
                    Calendar c = Calendar.getInstance();
                    String inputString2 = "";
                    String inputString1 = "";
                    Date d2 = null;
                    Date d1 = null;

                    try {
                        JSONObject object =
di.getInformationTime(buttonOne.getId());

                        inputString2 = object.getString("Month") + "/" +
object.getString("Day") + "/" + object.getString("Year")
                        + " " + object.getString("Hour") + ":" +
object.getString("Minute") + ":" + c.get(Calendar.SECOND);
                        inputString1 = (c.get(Calendar.MONTH) + 1) + "/" +
c.get(Calendar.DAY_OF_MONTH) + "/" + c.get(Calendar.YEAR)
                        + " " + c.get(Calendar.HOUR_OF_DAY) + ":" +
c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND);

                    } catch (JSONException e) {
                        e.printStackTrace();
                    }

                    try {
                        SimpleDateFormat format = new
SimpleDateFormat("MM/dd/yyyy HH:mm:ss");
                        d2 = format.parse(inputString2);
                        d1 = format.parse(inputString1);
                    } catch (ParseException e) {
                        e.printStackTrace();
                    }
                }
            })
    }
});

```

```

DateTime dt1 = new DateTime(d1);
DateTime dt2 = new DateTime(d2);
int days = Days.daysBetween(dt1, dt2).getDays();
int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes() %
60;

int hours = Hours.hoursBetween(dt1, dt2).getHours() % 24;
if (Days.daysBetween(dt1, dt2).getDays() < 1) days = 0;
if (Minutes.minutesBetween(dt1, dt2).getMinutes() % 60 < 1)
    minutes = 0;
if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;

if (days == 0 && minutes == 0 && hours == 0) {
    Toast.makeText(getApplicationContext(), "Day Expired",
Toast.LENGTH_LONG).show();
    return;
}
if (days == 0 && minutes <= 20 && hours == 0) {
    Toast.makeText(getApplicationContext(), "Less Than 20 min.
remaining notification not set", Toast.LENGTH_LONG).show();
    return;
}
if (days >= 24) {
    Toast.makeText(getApplicationContext(), "More than 24 days
not set", Toast.LENGTH_LONG).show();
    return;
}
if (days <= 23 && minutes >= 20) {
    Log.d("days", String.valueOf(days));
    Log.d("hours", String.valueOf(hours));
    Log.d("minutes", String.valueOf(minutes));

    int millisec = (days * 24 * 60 + hours * 60 + minutes - 20) * 60
* 1000;//20 minutes later notify
    Toast.makeText(CheckSeeMeetings.this, "Notification Set
Successfully", Toast.LENGTH_LONG).show();
    Intent intent = new Intent(CheckSeeMeetings.this,
MyAlarmService.class);
    PendingIntent pendingIntent =
PendingIntent.getBroadcast(CheckSeeMeetings.this, 0, intent,
PendingIntent.FLAG_UPDATE_CURRENT);
    AlarmManager alarmManager = (AlarmManager)
getSystemService(ALARM_SERVICE);
    Log.d("millisec", String.valueOf(millisec));
    alarmManager.set(AlarmManager.RTC_WAKEUP,
System.currentTimeMillis() + millisec, pendingIntent);
    //System current time cause the time reseted from calerand
above
}
dialog.cancel();

```

```

    }
    })
    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
    }).show();
    return true;
}
});
}

}
}
}

```

```

private final int refreshTime = 23000; //23 seconds
private static int forhandler_old_hour, forhandler_old_days,
forhandler_old_minute;

```

Πριν σχολιάσω αυτές τις μεταβλητές πρέπει να αναφερθώ

GoBack	
2016/11/26 ,12:38 ,Veropoulos ,Panagioti Rodiou 9, Rodos ,TimeStay 0	D:0 H:0 M:0
2016/11/26 ,13:32 ,BP ,Empona Rodoy, Rhodes ,TimeStay 38	D:0 H:0 M:0
2016/11/27 ,13:47 ,BP ,Empona Rodoy, Rhodes ,TimeStay 0	D:0 H:0 M:0
2016/12/15 ,18:08 ,So Greek ,42,, Ippodamou 40, Rodos ,TimeStay 0	D:0 H:0 M:0

ότι τα meeting παρουσιάζονται με αυτήν την μορφή και έχουν κάποια ημέρα ώρα λεπτά, αυτό όμως σημαίνει ένα πράγμα ότι πρέπει να τα ανανεώνουμε. Όσον αφορά τις παραπάνω μεταβλητές κάνουν αυτή την δουλειά.

```

super.onCreate(savedInstanceState);
setContentView(R.layout.checkseemeetings);
DatabaseInterface di = new DatabaseInterface();
di.setActivityUi(this);
di.createdb();

```

Δημιουργία Content view και της βάσης όπου είναι χρήσιμη για ευνόητους λογούς.

```

JSONArray object = di.Search();
for (int i = 0; i < object.length(); i++)

```

Όπως αναφέρεται παραπάνω κάνουμε εύρεση των γραμμών του table από το Search, παραδείγματος χάριν σε αυτήν την περίπτωση θα είχαμε 4.

Date	Time	Location	TimeStay
2016/11/26	12:38	Veropoulos, Panagioti Rodiou 9, Rodos	0
2016/11/26	13:32	BP, Empona Rodoy, Rhodes	38
2016/11/27	13:47	BP, Empona Rodoy, Rhodes	0
2016/12/15	18:08	So Greek, 42, Ippodamou 40, Rodos	0

```

LinearLayout ll = (LinearLayout) findViewById(R.id.ll);

```

Περνούμε το id του Layout ώστε να βάλουμε το button αργότερα

```

final Button button = new Button(this);
final JSONObject placeObject = object.getJSONObject(i);
button.setId(Integer.valueOf(placeObject.getString("Id")));
button.setTextColor(Color.argb(220, 250, 250, 250));
button.setTextSize(20);

```

Δημιουργούμε ένα κουμπί το οποίο παίρνει το id από το table row, αυτό γίνεται ώστε όταν διαγράψω το κουμπί του επιλεγμένου meeting να διαγράφεται και το table row ουσιαστικά λειτουργεί σαν καθρέπτης. Στην συνέχεια αναθέτω χρώμα και μέγεθος κειμένου.

```
final String Info = placeObject.getString("Year") + "/" +  
placeObject.getString("Month") + ...
```

```
SimpleDateFormat format = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");  
String inputString2 = placeObject.getString("Month") ..+ c.get(Calendar.SECOND);  
String inputString1 = (c.get(Calendar.MONTH) + 1) + "/" +  
c.get(Calendar.DAY_OF_MONTH) + ..;  
Date d2 = null;  
Date d1 = null;  
try {  
    d2 = format.parse(inputString2);  
    d1 = format.parse(inputString1);  
} catch (ParseException e) {  
    e.printStackTrace();  
}  
DateTime dt1 = new DateTime(d1);  
DateTime dt2 = new DateTime(d2);
```

Όπως είδαμε και παραπάνω στην DatabaseInterface και παρατηρούμε εδώ. Δημιουργούμε ημερομηνίες με τις τιμές του table row/της σημερινής ημέρας για τον λόγο ώστε να βρούμε την διάφορα και να ξέρουμε ποτέ έχουμε συνάντηση.

```
int days = Days.daysBetween(dt1, dt2).getDays();  
int minutes = Minutes.minutesBetween(dt1, dt2).getMinutes() % 60;  
int hours = Hours.hoursBetween(dt1, dt2).getHours() % 24;  
if (Days.daysBetween(dt1, dt2).getDays() < 1) days = 0;  
if (Minutes.minutesBetween(dt1, dt2).getMinutes() % 60 < 1) minutes = 0;  
if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;
```

Βρίσκουμε την διάφορα με την βοήθεια αυτού του κώδικα και στην περίπτωση που συναντήσουμε κάτι αρνητικό όπως η συνάντηση γινόταν πριν 5 μέρες ή πριν 5 λεπτά μηδενίζουμε την τιμή.

```
button.setText(Info + "\nD:" + days + " H:" + hours + " M:" + minutes);  
forhandler_old_days = days;  
forhandler_old_minute = minutes;  
forhandler_old_hour = hours;
```

Εμφανίζουμε τις τιμές και τις κρατάμε σαν παλαιές για τον λόγο των ανανεώσεων.

```
//update time with handler
```

```
if (forhandler_old_days != 0 || forhandler_old_hour != 0 || forhandler_old_minute != 0)
```

Εάν οι τιμές είναι διαφορές του 0 μπορώ να συνεχίσω ώστε να αποφύγουμε περιττές εντολές.

```
handler.postDelayed(new Runnable() {  
    public void run() {  
    }  
}, refreshTime );
```

Ο κώδικας που θα δούμε παρακάτω γίνεται στον handler δεν έχω βάλει for απλά ξανακαλώ τον handler από μέσα του.

```
Calendar c = Calendar.getInstance();  
...  
Date d1 = null;  
try {  
    inputString2 = placeObject.getString("Month") + "/" + ...  
c.get(Calendar.MINUTE) + ":" + c.get(Calendar.SECOND);  
} catch (JSONException e) {  
...  
} catch (ParseException e) {  
    e.printStackTrace();  
}  
DateTime dt1 = new DateTime(d1);  
...  
if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;
```

```
button.setText(Info + "\nD:" + days + " H:" + hours + " M:" + minutes);
```

Πλήρες αντιγραφή όπως και ο πάνω κώδικας ξαναβρίσκω την διάφορα στην ημερομηνία ώρα.

```
if (forhandler_old_days != 0 || forhandler_old_hour != 0 || forhandler_old_minute != 0) {  
    forhandler_old_days = days;  
    forhandler_old_minute = minutes;  
    forhandler_old_hour = hours;  
    handler.postDelayed(this, refreshTime );  
}
```

Και αν όντως είναι οι τιμές διάφορες του 0 τότε ξανακαλώ όπως είπα τον handler.

```
ll.addView(button);  
ClickListener(button);
```

Τοποθετώ το κουμπί στο layout και φορτώνω έναν Listener.

```

public void onButtonClick(View v) {
    if (v.getId() == R.id.goBack) {
        Intent intent = new Intent(CheckSeeMeetings.this, PlaceAndIntrests.class);
        CheckSeeMeetings.this.startActivity(intent);
        CheckSeeMeetings.this.finish();
    }
}

```

Αν πατηθεί το κουμπί επιστρέφω στον χάρτη και σβήνω το προηγούμενο Activity.

```

public void ClickListener(final Button buttonOne)

```

Η μέθοδος για τα κουμπιά.

```

buttonOne.setOnClickListener(new Button.OnClickListener() {
    public void onClick(final View v) {
        new AlertDialog.Builder(CheckSeeMeetings.this)
            .setMessage("Delete or Set in Maps?")
            .setCancelable(false)

```

Εδώ προσθέτω την ιδιαιτερότητα του κουμπιού αναλόγως τι θα πατήσω.

```

        .setPositiveButton("Set", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                DatabaseInterface di = new DatabaseInterface();
                di.setActivityUi(CheckSeeMeetings.this);
                di.createdb();
                Log.d("testa", Integer.toString(v.getId()));
                try {
                    JSONObject object = di.getinformation(v.getId());

                    Intent intent = new Intent(CheckSeeMeetings.this,
PlaceAndIntrests.class);
                    intent.putExtra("Title", object.getString("Title"));
                    intent.putExtra("Vicinity", object.getString("Vicinity"));
                    intent.putExtra("LongLatofplace",
object.getString("LongLatofplace"));
                    intent.putExtra("ActivityBefore", "CheckSeeMeetings");
                    CheckSeeMeetings.this.startActivity(intent);
                    CheckSeeMeetings.this.finish();
                } catch (JSONException e) {
                    e.printStackTrace();
                }
                dialog.cancel();
            }
        }
    }
}

```

Αν πατηθεί το Set παίρνω της πληροφορίες από την βάση σύμφωνα με το Id του κουμπιού που το σύνδεσα προηγούμενος με το table row έτσι παίρνω στο activity τις τιμές και όταν δημιουργηθεί η PlaceandIntrests κλάση στην onCreate με το try catch θα εξακολουθήσει πορεία για το σετάρισμα της διαδρομής.

```

        .setNeutralButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        })
    })

```

Επιστροφή πίσω.

```

        .setNegativeButton("Delete", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                DatabaseInterface di = new DatabaseInterface();
                di.setActivityUi(CheckSeeMeetings.this);
                di.createdb();
                di.delete(buttonOne.getId());
                dialog.cancel();
                finish();
                startActivity(getIntent());
            }
        }).show();

```

Η Delete λειτουργεί όπως η Search όπου με το id του κουμπιού θα διαγράψω το table row που σχετίζεται με το button και θα κάνω refresh το activity.

Αυτά όλα με το απλό πάτημα του κουμπιού.

```

buttonOne.setOnLongClickListener(new Button.OnLongClickListener() {
    @Override
    public boolean onLongClick(final View v) {
        new AlertDialog.Builder(CheckSeeMeetings.this)
            .setMessage("Set A meeting?")
            .setCancelable(false)

```

Και εδώ θα περιγράψω τι γίνεται όταν έχεις κρατημένο το κουμπί. Ειδικότερα θα παρουσιάσω από κάτω για το alarm.

```

        .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
                dialog.cancel();
            }
        }).show();

```

Όπως και πάνω μια επιλογή για ακύρωση.


```
.setPositiveButton("Set", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int which) {
```

Σε αυτήν την μέθοδο θα προσπαθήσω να κάνω Set Alarm το οποίο θα συνδεθεί με την κλάση Alarm Service που θα περιγραφτεί από κάτω.

```
DatabaseInterface di = new DatabaseInterface();
di.setActivityUi(CheckSeeMeetings.this);
di.createdb();
Calendar c = Calendar.getInstance();
....
DateTime dt1 = new DateTime(d1);
DateTime dt2 = new DateTime(d2);
int days = Days.daysBetween(dt1, dt2).getDays();
....
if (Hours.hoursBetween(dt1, dt2).getHours() % 24 < 1) hours = 0;
```

Και εδώ όπως πάνω βρίσκω την διάφορα ημερομηνίας σε σχέση με την αποθηκευμένη συνάντηση και την τωρινή ημερομηνία ώρα. Στο οποίο πάλι μηδενίζω τις αρνητικές τιμές.

```
if (days == 0 && minutes == 0 && hours == 0) {
    Toast.makeText(getApplicationContext(), "Day Expired",
    Toast.LENGTH_LONG).show();
    return;
}
if (days == 0 && minutes <= 20 && hours == 0) {
    Toast.makeText(getApplicationContext(), "Less Than 20 min. remaining
notification not set", Toast.LENGTH_LONG).show();
    return;
}
```

Στην συνέχεια αποφάσισα όταν δεν έχω διάφορα μέρας ώρας κλπ να μην κάνω Set Alarm επίσης αν έχω λιγότερο από 20 λεπτά.

```
if (days >= 24) {
    Toast.makeText(getApplicationContext(), "More than 24 days not set",
    Toast.LENGTH_LONG).show();
    return;
}
```

Επειδή το Alarm που θα κάνω Set έχει αριθμητικό περιορισμό με τις πράξεις που έκανα στις 24 μέρες σαν εκατοστά του δευτερόλεπτου αρκούν για να γίνει αριθμητική υπερχειλίση.

```

if (days <= 23 && minutes >= 20)
    int millisec = (days * 24 * 60 + hours * 60 + minutes - 20) * 60 * 1000; //20
minutes later notify
    Toast.makeText(CheckSeeMeetings.this, "Notification Set Successfully",
    Toast.LENGTH_LONG).show();

```

Αν η συνθήκη με τις μέρες και λεπτά ικανοποιείται τότε

```

    Intent intent = new Intent(CheckSeeMeetings.this, MyAlarmService.class);
    PendingIntent pendingIntent =
    PendingIntent.getBroadcast(CheckSeeMeetings.this, 0, intent,
    PendingIntent.FLAG_UPDATE_CURRENT);
    AlarmManager alarmManager = (AlarmManager)
    getSystemService(ALARM_SERVICE);
    alarmManager.set(AlarmManager.RTC_WAKEUP,

```

Τα βήματα είναι τα εξής:

1. Δημιουργούμε ένα Intent που θα μας οδηγήσει στο Broadcast Receiver .
2. Δημιουργούμε το PendingIntent που χρειάζεται για το Alarm Parameter μετά συμπληρώνουμε το Content το unique code το intent και τέλος την σημαία που σημαίνει ότι το Intent μπορεί να γίνει update στο μέλλον.
3. Δημιουργούμε τον alarm manager με περιεχόμενο την alarm_service.
4. Τέλος τον κάνουμε set.

```

    System.currentTimeMillis() + millisec, pendingIntent);

```

Επίσης για όσους δεν ξέρουν, όταν δημιουργώ ένα Instance του ρολογιού και μετά θέλω να επιστρέψω την παρούσα ώρα σε δευτερόλεπτα μέσω του Calendar θα έχουμε πρόβλημα για αυτό χρησιμοποίησα αυτήν την εντολή.

Class MyAlarmService

```
package com.tolis.maps3856;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.support.v4.app.NotificationCompat;
import static android.content.Context.NOTIFICATION_SERVICE;
```

```
public class MyAlarmService extends BroadcastReceiver {
```

```
    private NotificationManager nm;
```

```
    @Override
```

```
    public void onReceive(Context k1, Intent k2) {
```

```
        // TODO Auto-generated method stub
```

```
        nm = (NotificationManager)
```

```
        k1.getSystemService(NOTIFICATION_SERVICE);
```

```
        // nm.cancel(0);
```

```
        Intent intent = new Intent(k1, CheckSeeMeetings.class);
```

```
        intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
```

```
        PendingIntent pendingIntent = PendingIntent.getActivity(
```

```
            k1, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);
```

```
        NotificationCompat.Builder builder = new NotificationCompat.Builder(k1)
```

```
            .setContentIntent(pendingIntent)
```

```
            .setContentTitle("You Have a Meeting in 20 min.")
```

```
            .setContentText("click to go to the app")
```

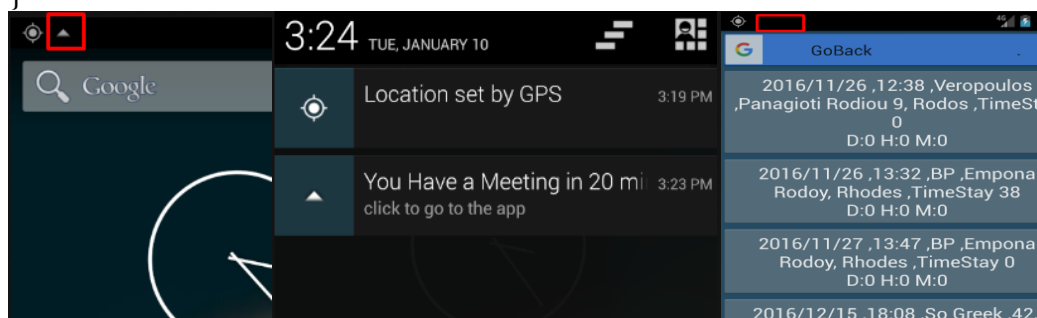
```
            .setSmallIcon(android.R.drawable.arrow_up_float)
```

```
            .setAutoCancel(true);
```

```
        nm.notify(0, builder.build());
```

```
    }
```

```
}
```



```

    nm = (NotificationManager)
k1.getSystemService(NOTIFICATION_SERVICE);
    // nm.cancel(0);
    Intent intent = new Intent(k1, CheckSeeMeetings.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent pendingIntent = PendingIntent.getActivity(
        k1, 0, intent, PendingIntent.FLAG_UPDATE_CURRENT);

    NotificationCompat.Builder builder = new NotificationCompat.Builder(k1)
        .setContentIntent(pendingIntent)
        .setTitle("You Have a Meeting in 20 min.")
        .setText("click to go to the app")
        .setSmallIcon(android.R.drawable.arrow_up_float)
        .setAutoCancel(true);

    nm.notify(0, builder.build());

```

Όπως περιγράφουν χαρακτηριστικά οι εικόνες με αυτόν τον κώδικα θέτουμε το notification message που εμφανίζεται το οποίο με την σειρά μας στέλνει μέσα στην εφαρμογή και διαγράφεται.

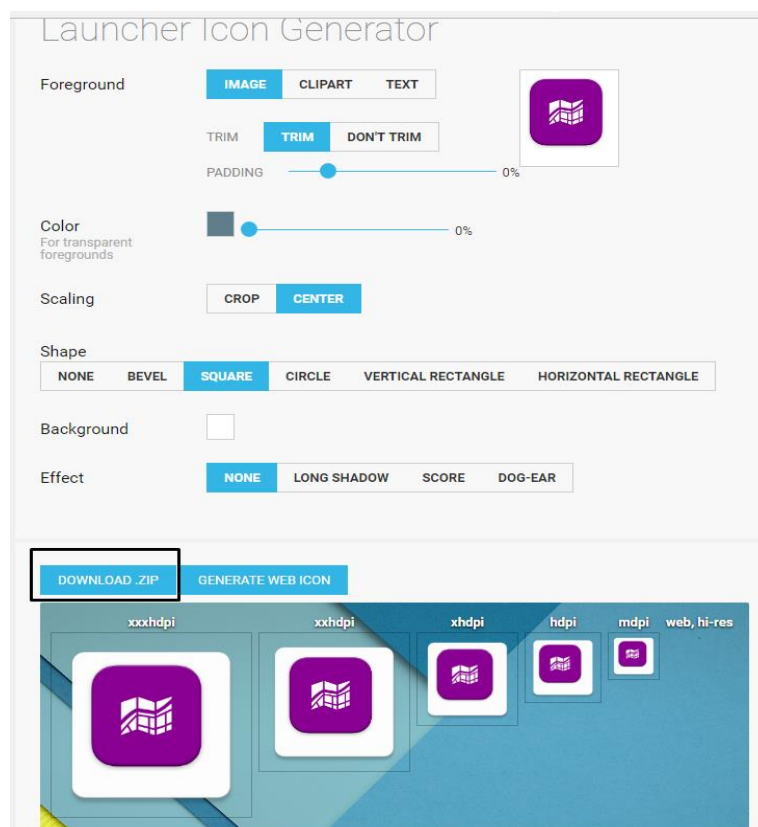
Το pending intent που βάζουμε με setContentIntent δημιουργεί την έναρξη της εφαρμογής στο CheckSeeMeetings activity και την ανανέωση του notification bar μετά θέτουμε τίτλο βασικό κείμενο και εικόνα.

ΕΙΚΟΝΗΔΙΟ ΕΦΑΡΜΟΓΗΣ.

Εργαλεία:

1. <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html#foreground.type=image&foreground.space.trim=1&foreground.space.padding=0.05&foreColor=ff5722%2C0&crop=0&backgroundShape=bevel&backColor=c50949%2C100&effects=none>

2. Μια εικόνα



Αφού το κατεβάσουμε το τοποθετούμε στην έξης περιοχή
Maps3856\app\src\main\res

Η μέσω τις εφαρμογής στο φάκελο res.

ΕΠΙΛΟΓΟΣ ΣΥΜΠΕΡΑΣΜΑ

Με το κλείσιμο αυτής της πτυχιακής ο αναγνώστης θα είναι σε θέση να κατανοήσει τις ενέργειες που επιτελεί και λαμβάνει ένας μηχανικός λογισμικού. Αναφέραμε με ποιον τρόπο πρέπει να σχεδιάζει μια εφαρμογή σε περιβάλλον android τι πρέπει να προσέχει, πώς να κάνει εισαγωγή νέων βιβλιοθηκών και να τις χειρίζεται, επιπρόσθετα να αντιλαμβάνεται τις ανάγκες ενός μελλοντικού χρηστή, πως στο μέλλον θα μπορεί να αντιμετωπίσει τυχόν πρόβλημα δηλαδή να υποστηρίξει το κομμάτι της συντήρησης διότι το να κτίζεις η να προωθείς ένα προϊόν όπως είναι μια εφαρμογή πρέπει να έχεις στο νου ότι μπορεί να προκύψει να είσαι σε αυτή την θέση διότι πλέον αυτό το κομμάτι θα επηρεάσει κατά κυριών λέγον έναν πελάτη. Επίσης δείξαμε πώς να χειρίζεται κανείς τα εργαλεία της google και μου προκάλεσε και εμένα σαν σχεδιαστής ιδιαίτερο ενδιαφέρον το πώς έχουν στήσει το σύστημα πληροφορίας τους. Ευτυχώς θέλω να πιστεύω ότι επηρέασα θετικά τον αναγνώστη στο τρόπο σκέψης και αντιμετώπισης. Κλείνοντας θα ήθελα να πω ότι αυτή η εφαρμογή αν όχι ότι τέλεια έκανε καλή χρήση υπηρεσιών που χρησιμοποιούνται σήμερα ημέρα και θα ζητούσα από οποιοδήποτε να μην βιάζεται να κάνει μια πρόχειρη δουλειά απλά για να πει ότι την έκανε κάτι αλλά να επιμένει για το καλύτερο καθώς ο κόσμος έχει ανάγκη τέτοιο κόσμο, επίσης και τέλος μην φοβάστε την αποτυχία εγώ σαν σχεδιαστής αυτής της εφαρμογής δεν ήξερα καν το μενού του android το οποίο μέσα σε 3 μήνες έφερα εις πέρας όλον αυτήν την εργασία όπου βεβαία υπάρχουν αρκετά περιθώρια βελτίωσης , ευχαριστώ καλή συνέχεια.

ΠΑΛΗΡΟΦΟΡΙΕΣ-ΗΛΕΚΤΡΟΝΙΚΕΣ ΠΗΓΕΣ

1. <http://nefeli.lib.teicrete.gr/browse/stef/epp/2015/BritzolakisAlexandros/document-1434443335-642563-10369.tkl> (Η πτυχιακή του Συναδέλφου Μπριτζολάκη Αλέξανδρου 2015)
2. <https://www.youtube.com/watch?v=zWHrCT2LCIg>
3. <https://www.youtube.com/watch?v=rWkWm7tmsOU>
4. <http://android-coffee.com/tutorial-how-to-play-youtube-videos-in-android-with-videoview/>
5. <http://www.vogella.com/tutorials/AndroidAnimation/article.html>
6. <https://www.youtube.com/watch?v=voYDInfcchs>
7. <https://www.youtube.com/watch?v=rWkWm7tmsOU>
8. <https://developer.android.com/training/maps/index.html>
9. <https://developers.google.com/maps/documentation/android-api/intents>
10. <https://developers.google.com/places/android-api/place-details>
11. <http://stackoverflow.com/questions/29717177/how-to-get-gas-station-along-my-route-in-google-maps-android-api-v2>
12. <https://www.youtube.com/watch?v=5sEepRYnHa8>
13. <https://developer.android.com/guide/components/fragments.html>
14. <https://www.youtube.com/watch?v=qhs9SZ265dU>
15. <http://android-coffee.com/tutorial-how-to-play-youtube-videos-in-android-with-videoview/>
16. <https://www.youtube.com/watch?v=ViwazAAR-vE>
17. https://www.youtube.com/watch?v=-xljI2_TRZg
18. https://www.youtube.com/watch?annotation_id=6bc9d639-bd1b-4f98-9716-2daae31adaef&feature=cards&src_vid=x6jQAALz1O8&v=o6gQ-oYn1aY
19. <https://www.youtube.com/watch?v=NT1qxmQH1eM>
20. <https://www.youtube.com/watch?v=KxILsk5j3rY>
21. <https://www.youtube.com/watch?v=PFvnSZHO3cI>
22. <https://www.simplifiedcoding.net/android-user-registration-tutorial/>
23. <https://github.com/tonikami/NEWLoginRegister>
24. <https://www.simplifiedcoding.net/android-login-and-registration-with-php-mysql/>
25. <https://www.youtube.com/watch?v=JQXfIidfFMo>
26. <https://www.youtube.com/watch?v=x6jQAALz1O8>
27. <https://developer.android.com/guide/topics/location/strategies.html>
28. <http://stackoverflow.com/questions/32290045/error-invoke-virtual-method-double-android-location-location-getlatitude-on>
29. <http://stackoverflow.com/questions/37583440/calculate-nearest-location-google-map-android-studio>
30. <https://www.youtube.com/watch?v=CCZPUeY94MU>
31. <https://www.youtube.com/watch?v=Rh9x90lqPHc>
32. <http://blog.teamtreehouse.com/beginners-guide-location-android>
33. <http://stackoverflow.com/questions/32153318/httpclient-wont-import-in-android->

- [studio](#)
34. <http://stackoverflow.com/questions/30336876/how-to-display-the-results-of-nearbysearch-in-android-using-google-places-api>
 35. <http://stackoverflow.com/questions/21933247/this-ip-site-or-mobile-application-is-not-authorized-to-use-this-api-key>
 36. <https://code.tutsplus.com/tutorials/android-sdk-working-with-google-maps-displaying-places-of-interest--mobile-16145>
 37. https://developers.google.com/places/supported_types
 38. <https://developers.google.com/places/web-service/search>
 39. <http://stackoverflow.com/questions/16635530/android-google-map-clicked-marker-opens-new-activity-or-bigger-window>
 40. <https://developers.google.com/maps/documentation/android-api/marker>
 41. <https://github.com/googlemaps/android-samples/blob/master/ApiDemos/app/src/main/java/com/example/mapdemo/MarkerDemoActivity.java>
 42. <http://stackoverflow.com/questions/19452269/android-set-text-to-textview>
 43. http://www.tutorialspoint.com/android/android_custom_fonts.htm
 44. <http://stackoverflow.com/questions/32161757/how-to-animate-the-camera-to-a-specified-location-in-google-maps-v2-for-android>
 45. <http://stackoverflow.com/questions/14884398/how-to-get-current-googlemap-zoom-level>
 46. <http://stackoverflow.com/questions/9409195/how-to-get-complete-address-from-latitude-and-longitude>
 47. https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=36.435475,28.215388&key=AIzaSyCa9TerPn2Hmy9wbAQMx-CEd_fOOJ4r0bw
 48. <http://stackoverflow.com/questions/14213231/how-can-i-get-button-name-inside-onclick-method>
 49. <http://stackoverflow.com/questions/5620772/get-text-from-pressed-button>
 50. <http://stackoverflow.com/questions/3851507/android-widget-how-to-change-the-text-of-a-button>
 51. <http://stackoverflow.com/questions/11715371/yes-no-alert-dialog-box-in-android>
 52. <http://stackoverflow.com/questions/12832379/couldnt-open-file-on-client-side-trying-server-side-error-in-android>
 53. <https://developer.android.com/training/location/retrieve-current.html>
 54. <http://stackoverflow.com/questions/6293563/android-listen-for-orientation-change>
 55. <https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest>
 56. <https://www.youtube.com/watch?v=czKLAx750N0>
 57. <https://www.mkyong.com/java/how-to-calculate-date-time-difference-in-java/>
 58. <http://stackoverflow.com/questions/5238491/check-if-string-contains-only-letters>
 59. <http://stackoverflow.com/questions/32285889/iterators-vs-arraylist-addall-with-clone-method-in-java>
 60. <https://www.youtube.com/watch?v=1fV9NmvxXJo>