ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΛΟΓΙΣΤΙΚΗΣ ΚΑΙ ΧΡΗΜΑΤΟΟΙΚΟΝΟΜΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ANDROID ΕΦΑΡΜΟΓΗΣ ΠΡΟΗΓΜΕΝΟΥ ΗΜΕΡΟΛΟΓΙΟΥ & ΣΗΜΕΙΩΜΑΤΑΡΙΟΥ ΜΕ ΧΡΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΤΟΥ MIT APP INVENTOR

ΤΖΑΝΝΙΔΑΚΗΣ ΕΜΜΑΝΟΥΗΛ Α.Μ.:9980

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ
ΔΗΜΟΤΙΚΑΛΗΣ ΙΩΑΝΝΗΣ

ΗΡΑΚΛΕΙΟ

ΜΑΡΤΙΟΣ 2017

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η κατανόηση και η εξοικείωση με ένα απλό περιβάλλον δημιουργίας βασικών εφαρμογών σε περιβάλλον android, με απώτερο σκοπό οι γνώσεις που θα αποκομίσει ο φοιτητής να αποτελέσουν μελλοντικά εφόδια για τη δημιουργία αντίστοιχων εφαρμογών. Η εφαρμογή μπορεί να χρησιμοποιηθεί ως ημερολόγιο-σημειωματάριο από επιχειρήσεις βελτιώνοντας έτσι την οργάνωση και την διαχείριση των εργασιών και των υποχρεώσεων αλλά και από απλούς χρήστες που επιθυμούν την υπενθύμιση ή καταγραφή απλών γεγονότων. Η εφαρμογή υλοποιήθηκε με το MIT app inventor και χρησιμοποιεί πολύ απλή βάση δεδομένων. Σχεδιάστηκε για να είναι απλή στη χρήση και με ελάχιστες κινήσεις και γνώσεις να μπορείς να περιηγηθείς σε όλο το εύρος λειτουργιών της εφαρμογής. Κατά την ανάπτυξη της εφαρμογής βρήκα στο περιβάλλον του app inventor ότι μπορούν να δημιουργηθούν απλές λειτουργικά εφαρμογές εύκολα και γρήγορα ενισχύοντας την μηχανογράφηση και την οργάνωση των επιχειρήσεων βελτιώνοντας την παραγωγικότητα και αποδοτικότητα. Όμως η απλότητα της εφαρμογής έχει και τα μειονεκτήματά της καθώς δεν μπορεί να καλύψει σε βάθος τις απαιτήσεις που μπορεί να έχει μια επιχείρηση. Η μηχανογράφηση είναι απαραίτητη για όλες τις επιχειρήσεις καθώς επαναλαμβανόμενες διαδικασίες μπορούν να αυτοματοποιηθούν και μεγάλες ποσότητες πληροφοριών μπορούν να αποθηκευτούν και αναζητηθούν εύκολα και γρήγορα. Για λόγους ασφαλείας η σχεδίαση της εφαρμογής πρέπει να γίνει πολύ προσεκτικά και να εξεταστεί από διάφορες πλευρές καθώς επεξεργάζεται και διαχειρίζεται σημαντικά και ίσως προσωπικά δεδομένα οπότε πρέπει να διασφαλιστεί η ασφάλεια και η εμπιστευτικότητα των δεδομένων που χρησιμοποιούν οι χρήστες. Για την δημιουργία αυτής της εφαρμογής λαμβάνονται υπόψιν οι απαιτήσεις που έχουν οι επιχειρήσεις καθώς και ανάγκες των στελεχών τους.

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ANDROID, MIT APP INVENTOR, ΕΦΑΡΜΟΓΗ, ΗΜΕΡΟΛΟΓΙΟ & ΣΗΜΕΙΩΜΑΤΑΡΙΟ

ANDROID APPLICATION DEVELOPMENT ADVANCED CALENDAR & NOTEBOOK USING THE IMPLEMENTATION OF MIT APP INVENTOR

ABSTRACT

The purpose of this paper is to understand and familiarize with a simple android application development environment, having as an ulterior motive to acquire the skills to develop software applications. The application we are developing can be used from enterprises as a tool to help organize and manage tedious repetitive tasks or simple users (as a calendar/notepad) as a reminder or a journal. Our application was developed with MIT App Inventor suite and is also supported by a simple database. It was designed with simplicity in mind, so much that every function is just two touches away. During the development process i found that MIT App Inventor can be used to easily develop functionally simple applications that can be used to amplify the productivity of enterprises or simple users. However this advantage also becomes its weakness as complex applications are out of App Inventor's capabilities. Computerization (the process to automate tasks) is essential to all enterprises because of the competitive advantage it gives, from boosting productivity to giving deeper insights. Finally the security of the application must take the highest priority as important and critical data are processed, managed and exchanged among all the corresponding peers, thus the protection of privacy is imperative. Also for the development of this app the individual and unique needs/demands of enterprises are considered.

KEYWORDS: **ANDROID, MIT APP INVENTOR, APPLICATION, CALENDAR & NOTEBOOK**

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Στη σημερινή εποχή όλοι έχουμε αντιληφθεί την ραγδαία ανάπτυξη της τεχνολογίας και πως αναμφισβήτητα τα έξυπνα τηλέφωνα (smartphones) έχουν γίνει απαραίτητα εργαλεία στην καθημερινότητα μας. Αυτό συμβαίνει κυρίως για το λόγο ότι οι δυνατότητες που ενσωματώνουν τα smartphones είναι πολύ περισσότερες και πιο χρήσιμες από αυτές που είχαν τα συμβατικά τηλέφωνα τα προηγούμενα χρόνια.

Σκοπός της πτυχιακής εργασίας είναι η ανάπτυξη ενός προηγμένου ημερολογίου και σημειωματάριου που προορίζεται για τα παραπάνω κινητά τηλέφωνα με χρήση της εφαρμογής MIT App Inventor (θα αναφερθούμε παρακάτω). Η εφαρμογή αναπτύχθηκε με γνώμονα τη οργάνωση των επιχειρήσεων αλλά και τη διευκόλυνση των εργαζομένων, του διοικητικού προσωπικού και των στελεχών που την απαρτίζουν. Με αυτή τη συγκεκριμένη εφαρμογή θα μπορούν να αποθηκεύουν διάφορες σημειώσεις και εργασίες που πρέπει να γίνουν διασφαλίζοντας έτσι τη σωστή λειτουργία όλων των μερών και των καθορισμό των προτεραιοτήτων ανά χρήστη.

Η εφαρμογή του προηγμένου ημερολογίου σημειωματάριου είναι πολύ απλή και εύκολη στη χρήση και δεν χρειάζονται συγκεκριμένες γνώσεις για να την χρησιμοποιήσει κάποιος. Αυτό ήταν απαραίτητη προϋπόθεση για την δημιουργία της εν λόγω εφαρμογής, να είναι δηλαδή φιλική προς το χρήστη.

1.1 ΠΕΡΙΦΡΑΦΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Με τη συγκεκριμένη εφαρμογή ο χρήστης μπορεί να αποτυπώσει γρήγορα τις σκέψεις του και να λάβει υπενθύμιση αργότερα, στο σωστό μέρος ή στη σωστή στιγμή. Ο χρήστης μπορεί να βγάλει φωτογραφία μια αφίσα, απόδειξη ή έγγραφο και να τα εντοπίσει εύκολα αργότερα μέσω της αναζήτησης για να τα οργανώσει. Η εφαρμογή διευκολύνει την καταγραφή των σκέψεων ή μιας λίστας και επιτρέπει την κοινή χρήση.

1.2 ΠΡΟΣΩΠΙΚΗ ΕΡΕΥΝΑ

Για την υλοποίηση της εφαρμογής ημερολόγιο-σημειωματάριο για επιχειρήσεις έγινε μια μεγάλη έρευνα σχετικά με το περιβάλλον του MIT App Inventor και των modules που μπορούσαν να χρησιμοποιηθούν. Έτσι ξεκίνησε η αναζήτηση των μέσων για τη δημιουργία της εφαρμογής.

Η πρώτη πηγή που επισκέφτηκα ήταν προφανώς το επίσημο site του App Inventor http://appinventor.mit.edu/explore/ai2/tutorials.html. Εκεί υπάρχει μία αρκετά μεγάλη συλλογή παραδειγμάτων και αρκετό υλικό για να μάθει κάποιος τα βασικά για τα modules και τα blocks. Όμως οι απαιτήσεις τις εφαρμογής μας δεν καλύπτονται από αυτά και αναγκαστικά η έρευνα πρέπει να συνεχίσει.

Η αναζήτηση με οδήγησε στο https://puravidaapps.com/tutorials.php, το οποίο περιέχει αρκετά παραδείγματα για τη διαχείριση των βάσεων δεδομένων.

Τέλος το https://developers.google.com/fusiontables/docs/v2/getting_started περιέχει αναλυτικά το api του fusion tables και κάλυψε ότι κενά και απορίες υπήρχαν από τα προηγούμενα.

ΚΕΦΑΛΑΙΟ 2: ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΣΤΟ APP INVENTOR

2.1 ΤΙ ΕΙΝΑΙ ΤΟ APP INVENTOR ΓΙΑ ANDROID

Το MIT app inventor για λειτουργικό σύστημα Android είναι μια web εφαρμογή ανοιχτού κώδικα που αρχικά δημιουργήθηκε από την διεθνούς φήμης εταιρεία της Google και αυτή τη στιγμή συντηρείται από το Τεχνολογικό Ινστιτούτο της Μασαχουσέτης (MIT).

Είναι ουσιαστικά ένα δωρεάν εργαλείο για τη δημιουργία εφαρμογών για κινητά Android. Επιτρέπει ακόμα και σε αρχάριους τον προγραμματισμό ηλεκτρονικών υπολογιστών να δημιουργούν πολύ εύκολα εφαρμογές λογισμικού για το λειτουργικό σύστημα Android. Ένα από τα ιδιαίτερα χαρακτηριστικά του είναι πως το γραφικό περιβάλλον που χρησιμοποιεί είναι πολύ παρόμοιο με το Scratch και το User Interface StarLogo TNG (γλώσσες προγραμματισμού). Επιτρέπει δηλαδή στους χρήστες με την λειτουργία 'Drag and Drop' να μετακινούν οπτικά αντικείμενα για να δημιουργήσουν μια εφαρμογή που να τρέχει σε συσκευές Android (Android phones). Κατά τη δημιουργία του App Inventor, η Google βασίστηκε σε πολύ προηγμένη έρευνα σχετικά με την εκπαίδευση μέσω ηλεκτρονικών υπολογιστών και όλες οι εργασίες γίνονται μέσω της Google σε online περιβάλλον ανάπτυξης.

Το App Inventor και τα έργα που μπορούν να δημιουργηθούν μέσω αυτού στηρίζονται σε συγκεκριμένες κατασκευαστικές θεωρίες μάθησης που θέλουν να προωθήσουν τον προγραμματισμό ως ένα πολύ ισχυρό μέσω για την σύλληψη μεγάλων ιδεών μέσω της ενεργούς μάθησης. Οι χρήστες μπορούν πρακτικά να δημιουργήσουν ότι εφαρμογή θέλουν (με βάση τις δυνατότητες της εφαρμογής). Ως εκ τούτου είναι αναμφισβήτητα μέρος της συνεχιζόμενης εξέλιξης στην εκπαίδευση μέσω των ηλεκτρονικών υπολογιστών που ξεκίνησε με το έργο του Seymour Papert και του MIT Logo Group το 1960 καθώς και μέσω της δουλειάς του Mitchel Resnick στο Lego Mindstorms και στο StarLogo. Το πρόγραμμα MIT App Inventor επιδιώκει να εκδημοκρατίσει την ανάπτυξη λογισμικού για την δημιουργία εφαρμογών ενδυναμώνοντας όλους τους ανθρώπους και ιδιαίτερα τους νέους ώστε από καταναλωτές της τεχνολογίας να γίνουν δημιουργοί της.

Για να μπορεί να χρήστης να εισέλθει και να εργαστεί στο App Inventor πρέπει να κατέχει ή να δημιουργήσει ένα λογαριασμό gmail καθώς όπως είπαμε παραπάνω το MIT συνεργάζεται με την Google.

2.2 Η ΙΣΤΟΡΙΑ ΤΟΥ APP INVENTOR

Η εφαρμογή του App Inventor ήταν έτοιμη στις 12 Ιουλίου 2010 όμως κυκλοφόρισε στο ευρύ κοινό στις 15 Δεκεμβρίου 2010. Η ομάδα του App Inventor είχε επικεφαλή τον Hal Abelson και τον Mark Friedman. Το δεύτερο εξάμηνο του 2011 η Google είχε κοινοποιήσει τον βασικό κώδικα, σταμάτησε τους δικούς τις servers και επιχορήγησε τη δημιουργία του κέντρου MIT για την εκμάθηση της κινητής πλατφόρμας την οποία διεύθυνε ο δημιουργός και εφευρέτης της εφαρμογής Hal Abelson και οι ακόλουθοι και καθηγητές του MIT Eric klopfer και Mitchel Resnick. Η έκδοση του MIT ξεκίνησε το Μάρτιο του 2012.

Στις 6 Δεκεμβρίου 2013 το MIT κυκλοφόρησε το App Inventor 2 μετονομάζοντας την αρχική έκδοση App Inventor Classic. Η νέα βελτιωμένη έκδοση έδωσε τη δυνατότητα σε πραγματικό χρόνο εντοπισμού σφαλμάτων για τις συνδεδεμένες συσκευές μέσω Wi-Fi ενώ μέχρι πρότινος γινόταν μόνο μέσω σύνδεσης USB. Από το Μάϊο του 2014 υπήρχαν 87.000 ενεργοί χρήστες της υπηρεσίας εβδομαδιαίος και 1,9 εκατομμύρια εγγεγραμμένοι χρήστες σε 195 χώρες για συνολικά 4,7 εκατομμύρια εφαρμογές που είχαν δημιουργηθεί. Ενώ το Δεκέμβριο του 2015 είχε 140.000 ενεργούς χρήστες εβδομαδιαίος και 4 εκατομμύρια εγγεγραμένους χρήστες σε 195 χώρες και συνολικά είχαν δημιουργηθεί 12 εκατομμύρια εφαρμογές.

2.3 ΣΤΟΧΟΙ ΚΑΙ ΠΛΕΟΝΕΚΤΗΜΑΤΑ

- Ανάπτυξη ικανοτήτων και εκπαίδευση μέσω ηλεκτρονικών υπολογιστών για ενήλικους και νέους ανθρώπους σε ολόκληρο το κόσμο. Έχει δημιουργηθεί εκπαιδευτικό υλικό που βοηθάει και υποστηρίζει τη δημιουργία εφαρμογών μέσω του συγκεκριμένου προγράμματος.
- Προώθηση της επιστήμης των υπολογιστών μέσω μιας πιο απλής γλώσσας προγραμματισμού προσιτή σε όλους.
- Διατήρηση και ενίσχυση της λειτουργίας της εφαρμογής με κύριο χαρακτηριστικό την δωρεάν προσφορά του στο κοινό. Για το σκοπό αυτό γίνεται συνέχεια βελτίωση της λειτουργίας, εντοπισμός σφαλμάτων και αύξηση της απόδοσης με προσθήκη νέων χαρακτηριστικών.
- Βελτίωση επιχειρήσεων σε συνεργασία με δημόσιους φορείς και ιδιωτικές επιχειρήσεις για την υποστήριξη μοναδικών εφαρμογών μέσω του App Inventor που εξυπηρετούν τις ανάγκες κάθε οργανισμού.
- Διεξαγωγή και υποστήριξη κοινωνικής έρευνας μέσω δημοσιεύσεων σχετικά με τις έρευνες που γίνονται παράλληλα με την ανάπτυξη, τον

έλεγχο και την αξιολόγηση της χρήσης του MIT App Inventor σε ολόκληρο το κόσμο.

Υπάρχουν αρκετά πλεονεκτήματα που κάνουν το app inventor μοναδικό:

- Διαθέτει ένα εύκολο στη χρήση περιβάλλον με πάρα πολλές δυνατότητες.
- Δίνει κίνητρα ακόμα και σε μαθητές να γνωρίσουν μία απλή γλώσσα προγραμματισμού και να διασκεδάσουν χρησιμοποιώντας τις δυνατότητες της εφαρμογής.
- Εκμάθηση μέσω της λύσης προβλημάτων.
- Δυνατότητα προσομοίωσης εφαρμογής για τον έλεγχο των modules που έχουν χρησιμοποιηθεί.
- Υποστήριξη και προώθηση από την εταιρία Google.

2.4 ΑΝΑΦΟΡΕΣ ΣΧΕΤΙΚΑ ΜΕ ΤΟ APP INVENTOR

2.4.1 ΤΟ MIT APP INVENTOR ΕΠΙΤΡΕΠΕΙ ΣΤΟΝ ΚΑΘΕΝΑ ΤΗΝ ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ANDROID

Το MIT App Inventor αποτελεί ένα προσιτό εργαλείο προγραμματισμού για το σχεδιασμό και τη κατασκευή πλήρως λειτουργικών εφαρμογών για κινητά τηλέφωνα με λογισμικό Android. Το App Inventor προωθεί μια νέα προοπτική στις δυνατότητες προγραμματισμού ακόμα και μέσω των κινητών τηλεφώνων καθώς ο χρήστης μπορεί να χρησιμοποιήσει αυτή τη τεχνολογία στην καθημερινότητα του λύνοντας σημαντικά προβλήματα με το κινητό του τηλέφωνο ακόμα και σε δύσκολες καταστάσεις. Το περιβάλλον του App Inventor και οι ατελείωτες δυνατότητες που δίνουν στο χρήστη τη δυνατότητα να προγραμματίσει με βάση τη λογική χωρίς να γνωρίζει κάτι συγκεκριμένο για τις γλώσσες προγραμματισμού καθώς δεν απαιτείται η δημιουργία κώδικα έναντι άλλων γλωσσών προγραμματισμού. Υπάρχει συνεχόμενη εξέλιξη και εμπλουτισμός των δυνατοτήτων με την υποστήριξη της Google και του MIT.

2.4.2 ΣΧΕΔΙΑΣΗ ΕΦΑΡΜΟΓΩΝ ΓΙΑ ΤΗΝ ΔΙΔΑΣΚΑΛΙΑ ΚΑΙ ΕΚΜΑΘΗΣΗ: ΟΙ ΕΜΠΕΙΡΙΕΣ ΤΩΝ ΕΚΠΑΙΔΕΥΤΙΚΩΝ ΣΕ ΔΙΑΔΙΚΤΥΑΚΟ ΜΑΘΗΜΑ.

Αυτή η εργασία διερευνά πως εκπαιδευτικοί με περιορισμένες ή καθόλου εμπειρίες στον προγραμματισμό έμαθαν τον σχεδιασμό και ανάπτυξη εφαρμογών για smartphones με καθοδήγηση και υποστήριξη από κατάλληλο εκπαιδευτή. Οι εκπαιδευτικοί ήταν θετικοί για την αίσθηση της κοινότητας σε αυτό το διαδικτυακό μάθημα. Επίσης θεώρησαν το App Inventor ένα εξαιρετικό εργαλείο

προγραμματισμού για την ανάπτυξη χρήσιμων και πλήρως λειτουργικών εφαρμογών. Ένιωθαν την ενδυνάμωση από την δημιουργία μοναδικών εφαρμογών με App Inventor και πως η δική τους σχεδιαστική δουλειά και η δημιουργικότητα των λύσεων είναι εμπνευσμένη από άλλες εφαρμογές που έχουν αναπτυχθεί και μοιραστεί από άλλους χρήστες Οι μαθησιακές δραστηριότητες όπως το να μοιράζεσαι τις εφαρμογές σου, η παροχή απαντήσεων σε ερωτήσεις, η σύνθεση σχεδιαστικών ιδεών, η διατήρηση ημερολογίου σχεδίων αλληλοσυμπληρώνονται μεταξύ τους δημιουργώντας μία αίσθηση κοινότητας αρκετά ικανή να υποστηρίξει την εκμάθηση ανάπτυξης και σχεδίασης εφαρμογών. Αυτή η μελέτη βοήθησε να αποκαλυφθεί η μορφωτική αξία της δημιουργίας εφαρμογών που μπορεί να είναι χρήσιμες για τις ποικίλες ανάγκες της διδασκαλίας και της εκμάθησης.

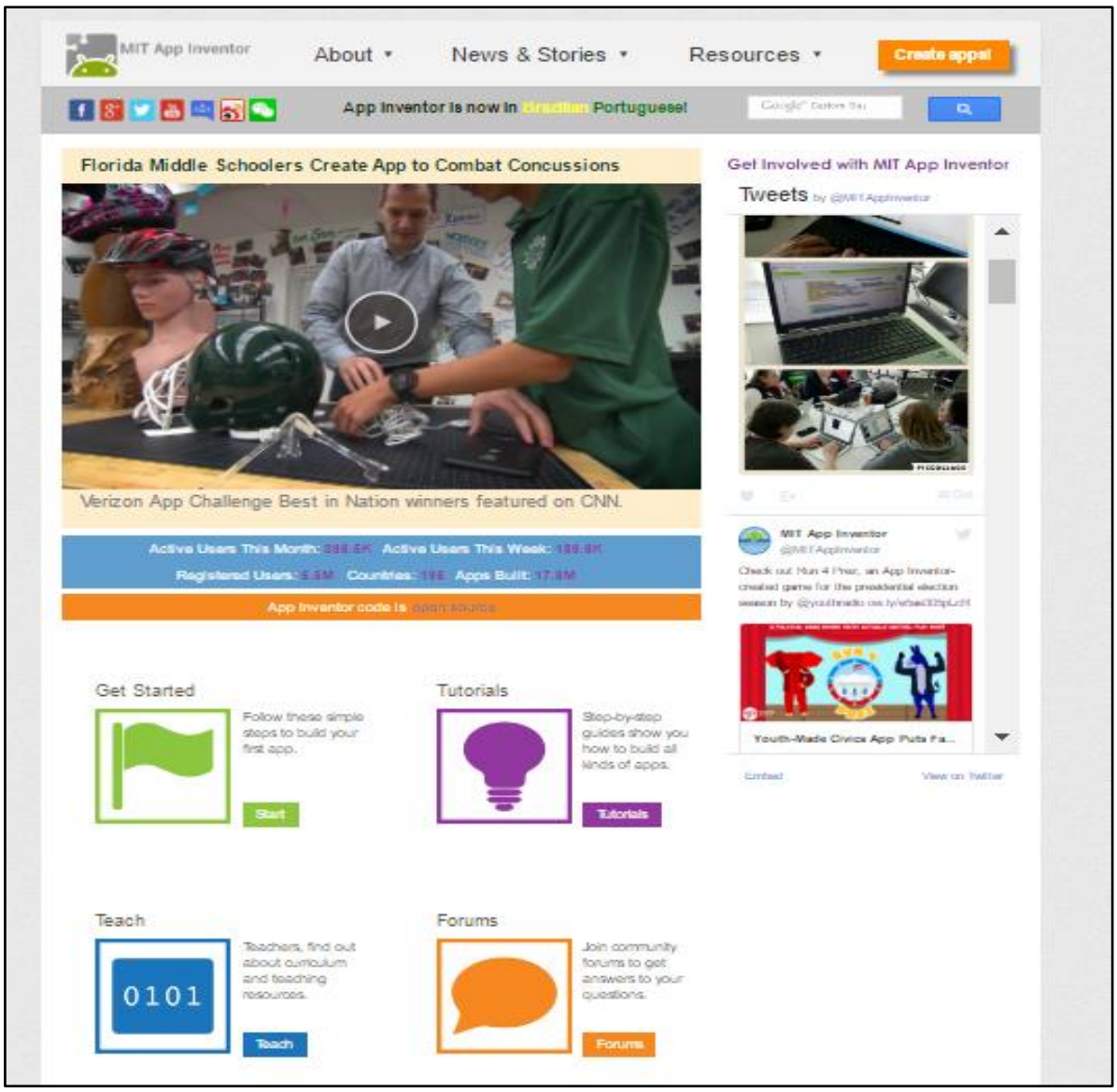
ΚΕΦΑΛΑΙΟ 3: ΣΧΕΔΙΑΣΜΟΣ ΣΤΟ APP INVENTOR

Το App Inventor χρησιμοποιείται για την δημιουργία android εφαρμογών σε ένα ευέλικτο και ευχάριστο περιβάλλον. Ο χρήστης δεν χρειάζεται συγκεκριμένες γνώσεις προγραμματισμού καθώς με το συγκεκριμένο πρόγραμμα η δημιουργία μιας εφαρμογής είναι πιο εύκολη υπόθεση.

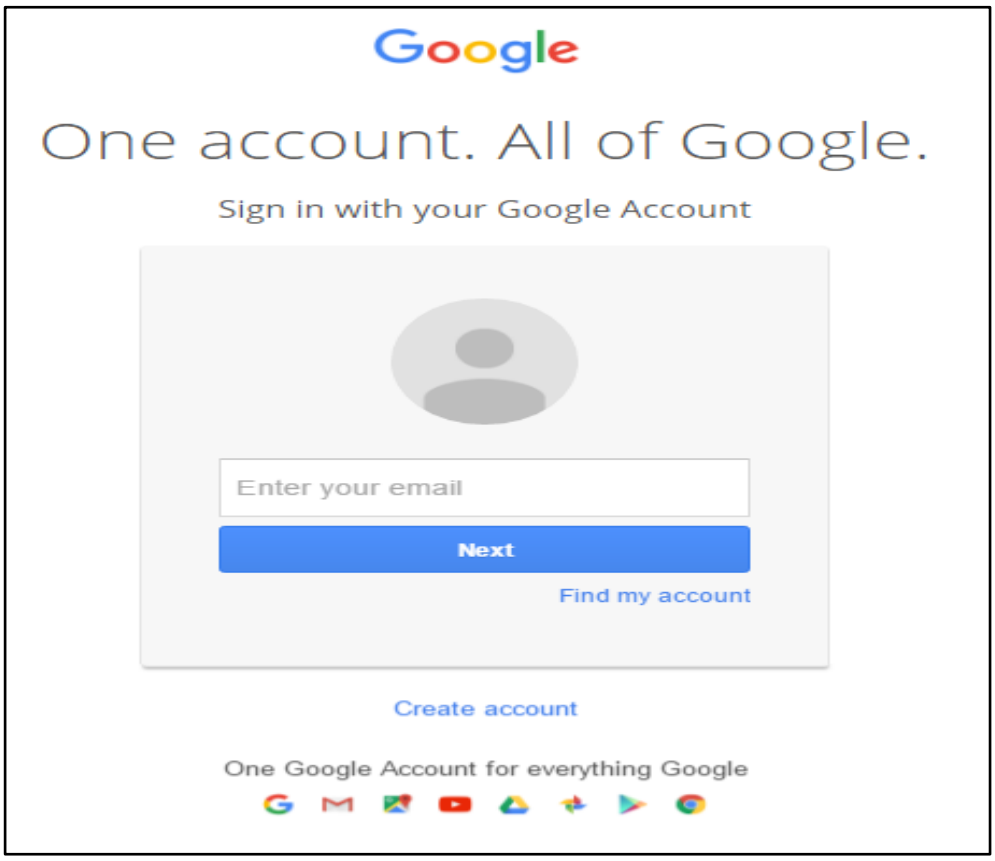
Όποιος αναζητήσει το περιβάλλον του App Inventor θα ξεκινήσει από την αρχική σελίδα που υπάρχει παρακάτω. Στην αρχική αυτη σελίδα μπορεί κάποιος να διαβάσει για τους δημιουργούς του App Inventor (ποιοί είναι και πως έφτασαν στην δημιουργία της εφαρμογής) καθώς και να ενημερωθεί για τους χορηγούς του και την έρευνα που έχουν κάνει. Επίσης ο επισκέπτης της ιστοσελίδας μπορεί να ενημερωθεί για τις εξελίξεις και τα νέα του App Inventor και το πιο σημαντικό μπορεί να μάθει πολύ εύκολα πως να το χρησιμοποιεί για την κατασκευή εφαρμογών μέσα από video’s. Πρόκειται για μία πολύ κατατοπιστική αρχική σελίδα που προκαλεί τον επισκέπτη να δοκιμάσει το App Inventor.

3.1 ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΩΝ

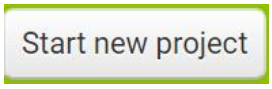
Για να ξεκινήσει κάποιος να δημιουργεί εφάρμογες το μόνο που έχει να κάνει είναι να πατήσει το 

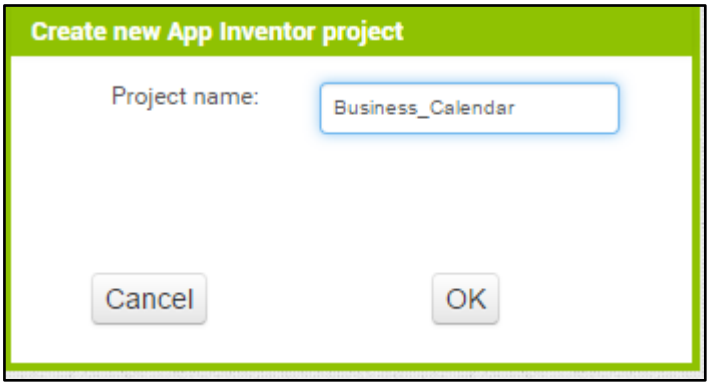


Στη συνέχεια πρέπει να γίνει Log In από το χρήστη στο λογαριασμό Google που διαθέτει. Αν δεν υπάρχει ήδη λογαριασμός ο χρήστης μπορεί να δημιουργήσει έναν πολύ εύκολα.

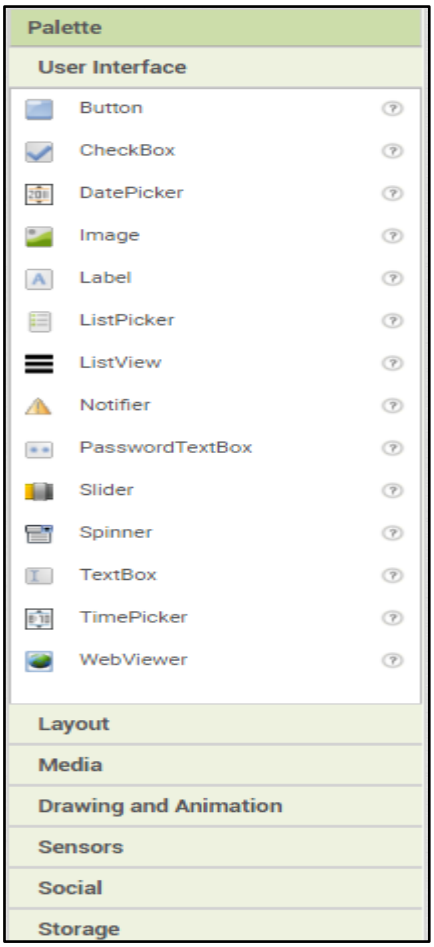


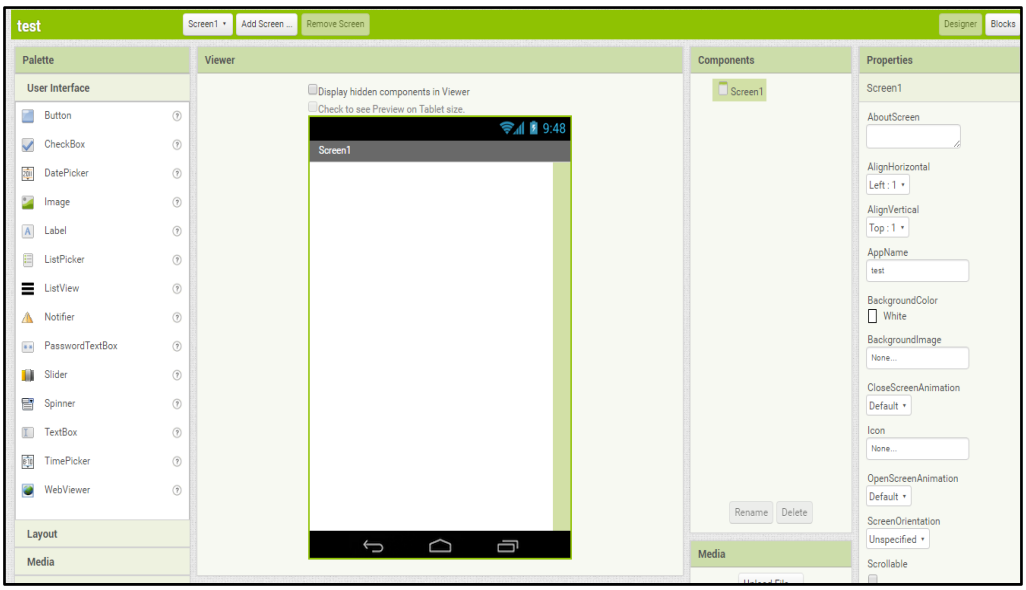
Όταν ο χρήστης κάνει Log In στο λογαριασμό του θα βρεθεί στην κεντρική σελίδα του App Inventor.

Επιλέγωντας  ο χρήστης μπορεί να ξεκινήσει τη δημιουργία της εφαρμογής.

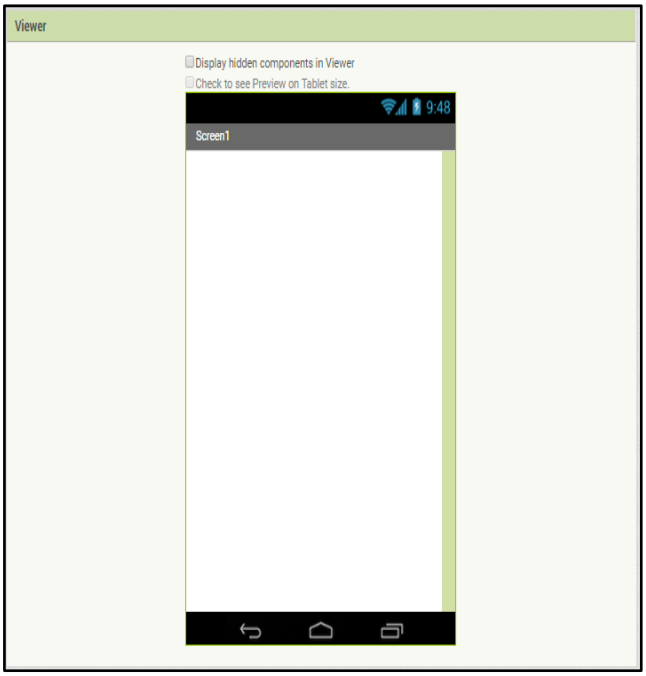


Παρακάτω φαίνεται το περιβάλλον δημιουργίας της σουίτας App Inventor. Στο designer παρακάτω κατασκευάζουμε τη δομή του user Interface.

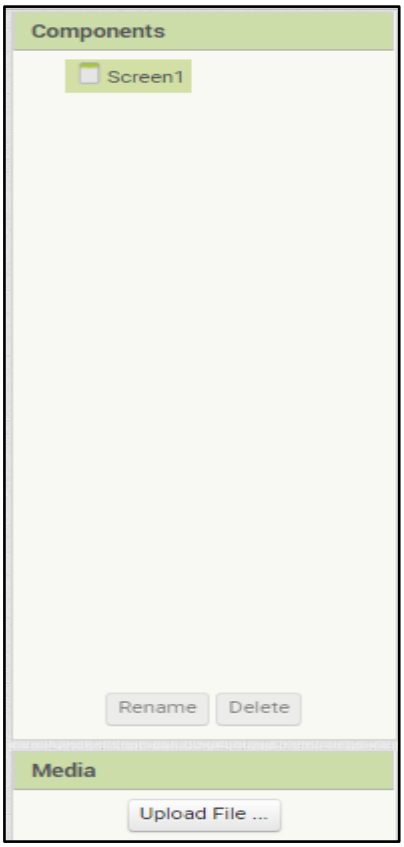




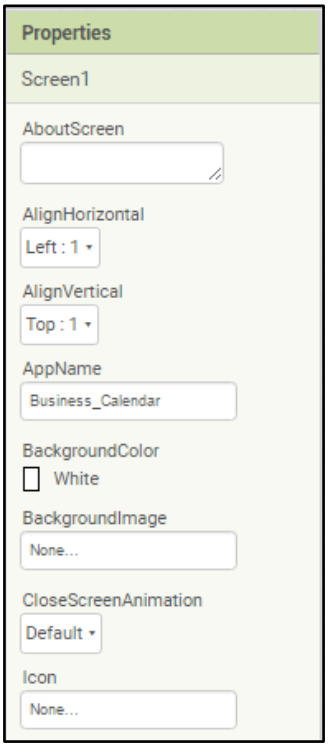
Αριστερά, είναι το Pallete box που περιέχει όλα τα components που θα χρειαστούμε για την εφαρμογή μας. Με drag and drop σέρνεις το component που θες στο Viewer box κατασκευάζοντας έτσι κομμάτι κομμάτι το interface.



Εδώ φαίνεται το Viewer Box, στο οποίο στην ουσία φαίνεται σε πραγματικό χρόνο η προεπισκόπηση της εφαρμογής. Ότι νέο component προσθέτουμε ή αφαιρούμε φαίνεται εδώ, ώστε να ξέρουμε πάντα πως θα μοιάζει η εφαρμογή μας.

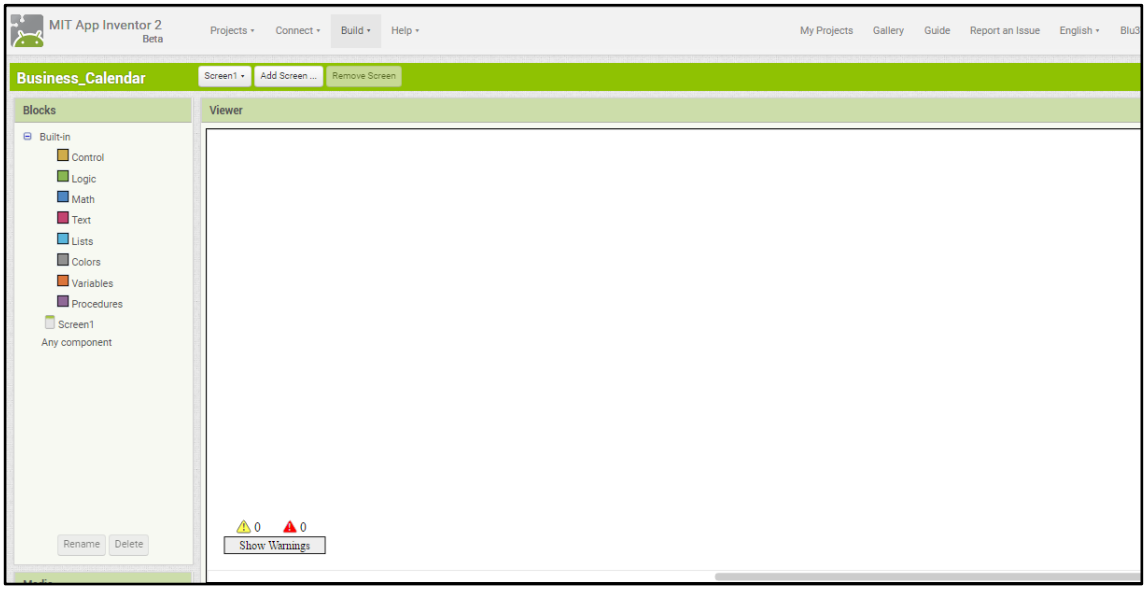


Εδώ φαίνεται ποια components έχουμε προσθέσει καθώς και η δενδρική τους διάταξη (parents - childs), τα ονόματα τους κτλπ. Επίσης από αυτό το box μπορούμε να διαγράψουμε ή να μετονομάσουμε τα components μας.



Τέλος έχουμε το Properties Box, από το οποίο έχουμε πρόσβαση στις παραμέτρους κάθε component προσαρμόζοντας το στα μέτρα μας.

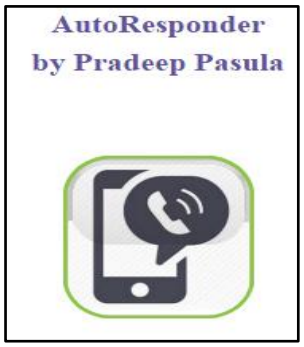
Έπειτα θα αναλύσουμε το blocks editor. Εδώ στην ουσία μέσω των blocks ορίζουμε την συμπεριφορά μεμονωμένων block και εν τέλει όλο το λειτουργικό κομμάτι της εφαρμογής μας.



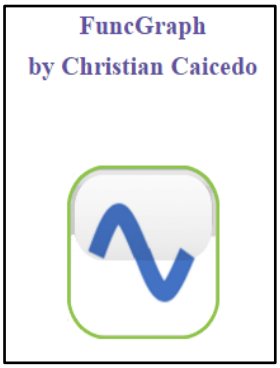
Στο αριστερό Panel έχουμε τις γενικές κατηγορίες blocks συν τα blocks των components που έχουμε προσθέσει ήδη. Στο δεξί panel το viewer, απεικονίζει γραφικά τις σχέσεις και συσχετίσεις όλων των blocks της σελίδας μας δίνοντας μας έτσι την δυνατότητα να έχουμε μια ολοκληρωμένη εικόνα.

3.2 ΜΕΡΙΚΕΣ ΑΠΟ ΤΙΣ ΚΑΛΥΤΕΡΕΣ ΕΦΑΡΜΟΓΕΣ

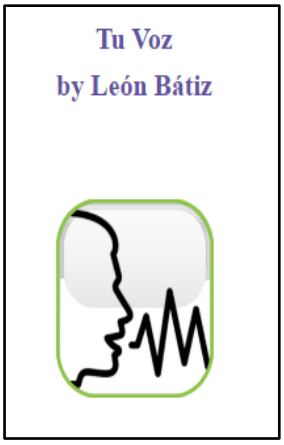
Το App Inventor διαθέτει μια μεγάλη γκάμα εργαλείων που αν χρησιμοποιηθούν σωστά και με φαντασία μπορούν δημιουργήσουν ένα πολύ μεγάλο εύρος εφαρμογών με δυνατότητες εφάμιλλες των επαγγελματικών εφαρμογών. Παρακάτω θα αναφέρουμε μερικές από τις καλύτερες εφαρμογές που έχουν δημιουργηθεί με το App Invnentor με βάση την ιστοσελίδα:
http://appinventor.mit.edu/explore/app-month-gallery.html



Αυτή η εφαρμογή που κατασκευάστηκε από ένα φοιτητή ηλεκτρολόγο μηχανικό στην Ινδία, επιτρέπει την αυτόματη απάντηση σε μηνύματα με πολλαπλές διαφορετικές και προσαρμόσιμες απαντήσεις. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=6389621200257024



Η συγκεκριμένη εφαρμογή επιτρέπει την γραφική απεικόνιση διαφόρων μαθηματικών συναρτήσεων για ανάλυση και συμπεράσματα. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=4829843227410432



Αυτή η εφαρμογή επιτρέπει σε άτομα με ακουστικά προβλήματα ή τελείως κωφάλαλους να επικοινωνούν με ένα απλό και λειτουργικό περιβάλλον. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=5742795216388096

Spark Interview
by Sabrina Kundu

Αυτή η εφαρμογή δημιουργήθηκε από μαθήτρια λυκείου σε σχολείο της Αμερικής και βοηθά εφήβους και γενικά ανθρώπους νέους στον κόσμο της εργασίας να αριστεύουν στις συνεντεύξεις ή να φτιάχνουν εξαιρετικά βιογραφικά. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=4657739771150336

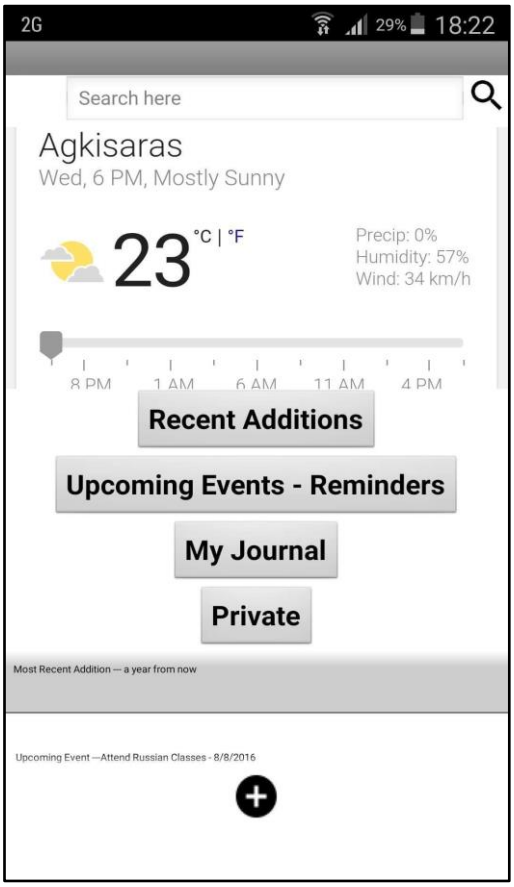
Arduino Multiple
Controller
by Sivarajan

Η εφαρμογή αυτή επιτρέπει τον έλεγχο ενός arduino (μητρική πλακέτα ανοιχτού κώδικα με ενσωματωμένο μικροελεγκτή)
μέσω bluetooth. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=6504828576727040

PingU Texting App
by Mihir

Αυτή η εφαρμογή επιτρέπει την αποστολή μηνυμάτων χωρίς ο χρήστης να χρησιμοποιεί χέρια. Περισσότερα γι' αυτή την εφαρμογή μπορείτε να δείτε στην ιστοσελίδα:
http://ai2.appinventor.mit.edu/?galleryId=6012194615721984

ΚΕΦΑΛΑΙΟ 4: ΕΦΑΡΜΟΓΗ ΠΡΟΗΓΜΕΝΟ ΗΜΕΡΟΛΟΓΙΟ & ΣΗΜΕΙΩΜΑΤΑΡΙΟ ΓΙΑ ANDROID.



4.1 ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ ΚΑΙ ΡΥΘΜΙΣΗ ΛΟΓΑΡΙΑΣΜΟΥ GOOGLE

Η συγκεκριμένη εφαρμογή δίνει την ικανότητα στο χρήστη να καταχωρεί κάποια event/υπενθυμίσεις λειτουργώντας σαν ημερολόγιο με σκοπό να ενημερώνει τους χρήστες περί αυτών. Επίσης λειτουργεί σαν κλασικό σημειωματάριο αποθηκεύοντας σε μορφή σημείωσης/”αρχείου”. Επίσης η εφαρμογή προσφέρει την δυνατότητα ενός private μέρους για τις ευαίσθητες πληροφορίες. Να σημειωθεί επίσης πως εκτός απο την επεξεργασία και διαγραφή η εφαρμογή προσφέρει και την συσχέτιση εικόνων με την κάθε σημείωση.

Πριν μπορέσει το App Inventor να χρησιμοποιήσει το fusion tables, πρέπει να υπάρχει ένας προσωπικός λογαριασμός διεύθυνσης ηλεκτρονικού ταχυδρομείου. Αυτός ο προσωπικός λογαριασμός επιτρέπει στους χρήστες να χρησιμοποιούν το fusion tables.

4.2 ΔΗΜΙΟΥΡΓΙΑ FUSION TABLE ΚΑΙ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.

Η δημιουργία ενός fusion table είναι τόσο απλή και εύκολη όσο η δημιουργία ενός εγγράφου Google. Ο χρήστης αρχικά συνδέεται στον λογαριασμό του στο Gmail. Έπειτα ο χρήστης εισέρχετε στο Google Drive και κάνει κλικ στο κόκκινο κουμπί ΄΄New΄΄ και επιλέγει το Google Fusion Tables από τη λίστα των επιλογών που θα εμφανιστούν. (Εαν δεν μπορείτε να δείτε το Google Fusion Tables στο μενού, τότε κάντε κλίκ στο ΄΄+΄΄ και συνδεθείτε σε περισσότερες εφαρμογές. Μετακινηθείτε προς τα κάτω για να βρείτε το Fusion Tables. Κάντε κλικ στο κουμπί ΄΄+΄΄ και στη συνέχεια κάντε κλικ στο κουμπί OK. Τώρα όταν κάνετε ξανά κλικ στο κουμπί ΄΄New΄΄ το Fusion Tables θα εμφανιστεί.) Δίδονται μερικές διαφορετικές επιλογές για τη δημιουργία του Fusion Tables. Επιλέξτε Create empty table. Έτσι ο νέος πίνακας εισέρχεται αυτόματα. Σε αυτό το σημείο ο χρήστης μπορεί να αλλάξει τα όνοματα των στηλών, να καθορίσει τον τύπο που θα έχουν τα δεδομένα (πχ κείμενο ή αριθμός) και να κάνει όποιες αλλαγές και προσθήκες επιθυμεί.

Ο προσωπικός λογαριασμός του χρήστη ΄΄user interface΄΄ ενεργεί ως εικονικός χρήστης που μπορεί να διαβάσει και να επεξεργαστεί τον πίνακα. Στο Fusion Table κάντε κλικ στο κουμπί ΄΄Share΄΄, βρείτε την επιλογή ΄΄Invite People΄΄ και συμπληρώστε τη διεύθυνση του mail σας. Βεβαιωθείται ότι το κουμπί ΄΄Can Edit΄΄ είναι επιλεγμένο. Απεπιλέξτε την επιλογή ΄΄Notify people΄΄ και κάντε κλικ στο ΄΄OK΄΄.

Η βάση δεδομένων μας είναι το σημαντικότερο κομμάτι της εφαρμογής μας καθότι φιλοξενεί τις καταχωρήσεις μας. Όπως βλέπουμε κάθε καταχώρηση θα έχει Ημ/νία, Τίτλο, το κύριο μέρος της, κατηγορίες στις οποίες ανήκει, ώρα εκχώρησης, και συσχετισμένες εικόνες.

CalendarApp_Data

Database for my appinventor app

Attribution unknown - Edited on 2016 June 4

File Edit Tools Help | Rows 1 | Cards 1 | Map of Location

Filter | No filters applied. Sorted by Date

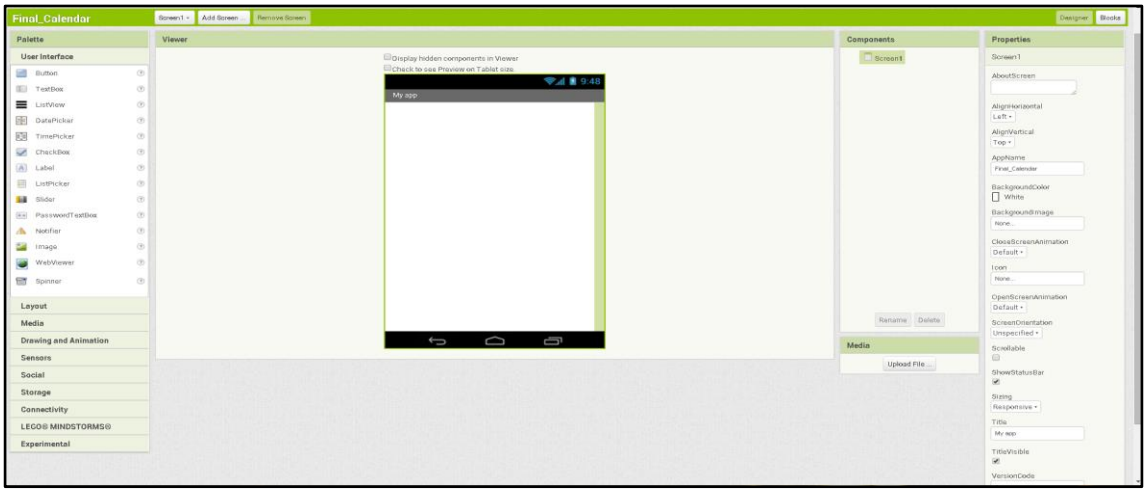
1-7 of 7

Date	Title	Note	Tags	EntryTimestamp	Images
03/26/2015	My days of glory		(Log)	2016-06-04 05:29	
03/25/2016	Trip to Rome	the first day we arrived was pretty epic	(Log)	2016-06-04 04:51	
06/01/2016	have a good month		()	2016-06-04 04:55	
06/06/2016	exams	hm	(Note)	2016-06-04 04:53	
08/08/2016	Attend Russian Classes	Start of semister -books - pens - notepad - Brain	(Reminder)	2016-04-15 10:58	
07/23/2017	a year from now		(Event)	2016-05-23 04:34	
06/04/2020	Olympic to Tokyo	make sure to attend	(Event)	2016-06-04 04:52	

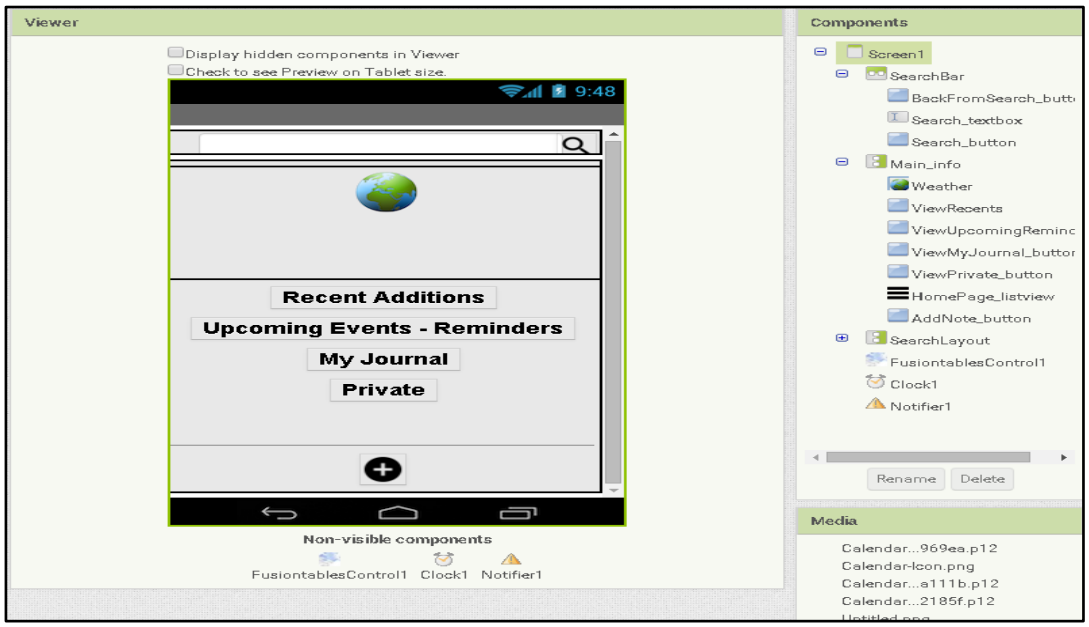
4.3 ΑΝΑΛΥΣΗ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

4.3.1 ΔΗΜΙΟΥΡΓΙΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Κάνουμε σύνδεση με την ιστοσελίδα Inventor Web App και ξεκινάμε ένα νέο έργο ΄΄new project΄΄. Ονομάζουμε το νέο μας έργο και ξεκινάμε τη δημιουργία.



4.3.2 ΑΡΧΙΚΗ ΣΕΛΙΔΑ



Όνομα Component	**Χρησιμότητα του Component**
SearchBar	Το layout που περιλαμβάνει τα components που θα χρησιμοποιηθουν στο search: -Seach textbox (περιέχει τα κριτήρια αναζήτησης) - Search Button (Ξεκινάει τη διαδικασία αναζήτησης) - Back from Search (Επιστρέφει στην Αρχική από την αναζήτηση)
Main_info	Το κύριο layout που περιέχει τις βασικές πληροφορίες της αρχικής οθόνης: - Weather (web component που έχει αρχικοποιηθεί στο weather της google) - Buttons (Το κάθε button ανοίγει το αντίστοιχο screen) - Το Homepage_listView (Λίστα στην οποία έχει καταχωρηθεί η πιο πρόσφατη καταχώρηση και το προσεχές event/reminder)
SearchLayout	Το search layout που περιέχει τη λίστα με τα αποτελέσματα αναζήτησης

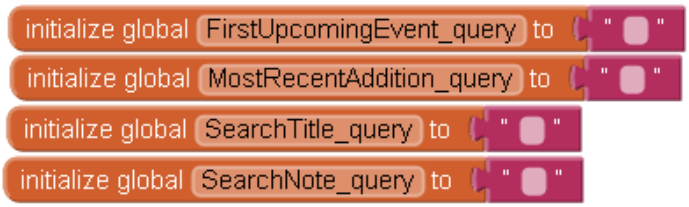
Blocks Editor

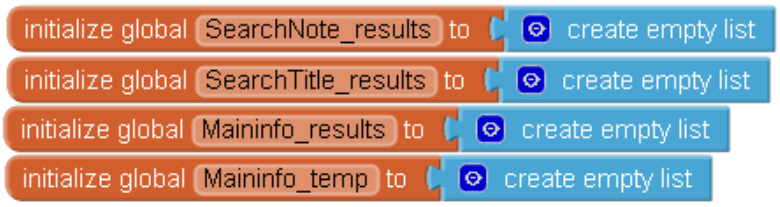
Ανοίγουμε το Blocks Editor ώστε να μπορούμε να προγραμματίσουμε τη συμπεριφορά της εφαρμογής. Αρχικά παρακάτω περιγράφονται οι μεταβλητές της εφαρμογής οι οποίες είναι βοηθητικές και απαραίτητες για τη σωστή εκτέλεση της εφαρμογής

Block Type	**Χρησιμότητα**
	Αρχικοποιείται στην σημερινή ημ/νία με τη βοήθεια του clock.
	Αρχικοποιείται στο table id του fusion table που χρησιμοποιούμε.
	Αρχικοποιείται στο query του προσεχούς event/ reminder.
	Αρχικοποιείται στο query της πιο πρόσφατης καταχώρησης.
	Αρχικοποιεί το query της αναζήτησης στον τίτλο.
	Αρχικοποιεί το query της αναζήτησης στο κυρίως κείμενο της σημείωσης.
	Λίστα που αποθηκεύει τα αποτελέσματα της αναζήτησης στην κύρια σημείωση.
	Λίστα που αποθηκεύει τα αποτελέσματα της αναζήτησης στον Τίτλο.
	Λίστα που αποθηκεύει το προσεχές event/reminder και την πιο πρόσφατη καταχώρηση.
	Προσωρινή λίστα που αποθηκεύει τα επιμέρους αποτελέσματα μέχρι να καταχωρηθούν στην Maininfo_results.

Αρχικοποίηση

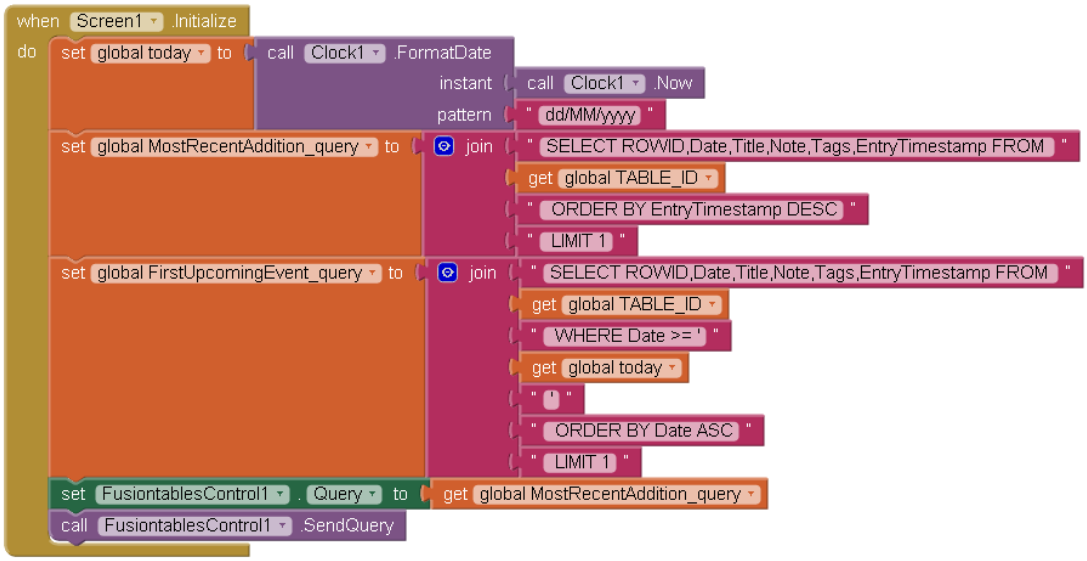
Είναι σημαντικό να κάνουμε κάποια βήματα αρχικοποίησης όταν ξεκινάει η εφαρμογή. Το table ID επειδή είναι στατικό και δεν πρόκειται να αλλάξει το ορίζουμε εξ'αρχής. Όσες μεταβλητές αρχικοποιούνται με το block  είναι βοηθητικές και θα πάρουν τιμή κατά τη διάρκεια εκτέλεσης του προγράμματος.





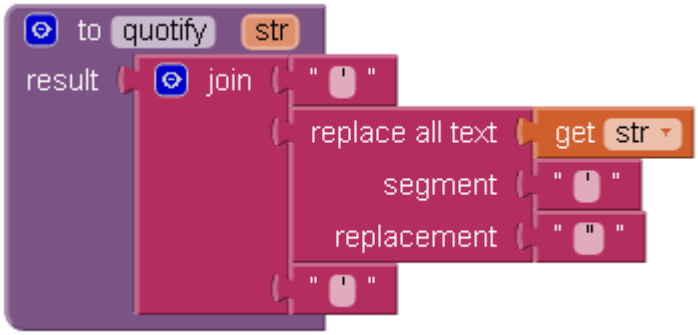
Αντίστοιχα οι υπόλοιπες αρχικοποιούνται ως κενές λίστες με παρόμοια χρησιμότητα.

Στη συνέχεια αναλαμβάνει η συνάρτηση  , που αρχικοποιεί τις μεταβλητές που πρέπει να περάσουν από μία διαδικασία πριν πάρουν τιμή. Έπειτα καλεί το πρώτο query το οποίο επιστρέφει την πιο πρόσφατη καταχώρηση.

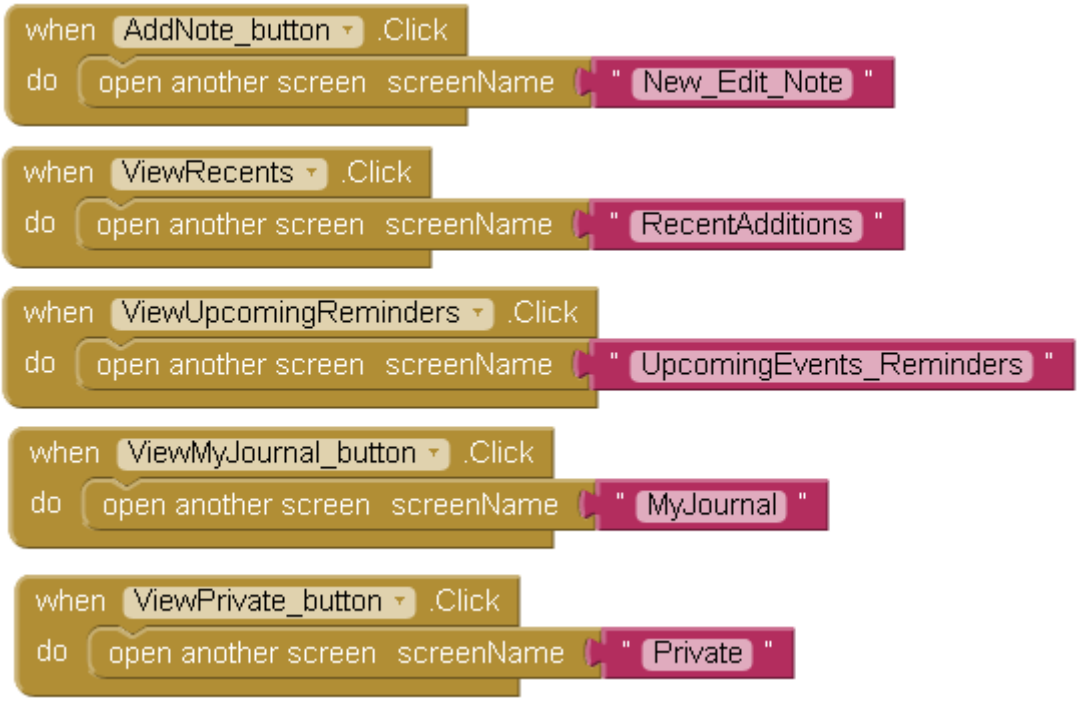


Στη συνέχεια ορίζουμε τη συνάρτηση 

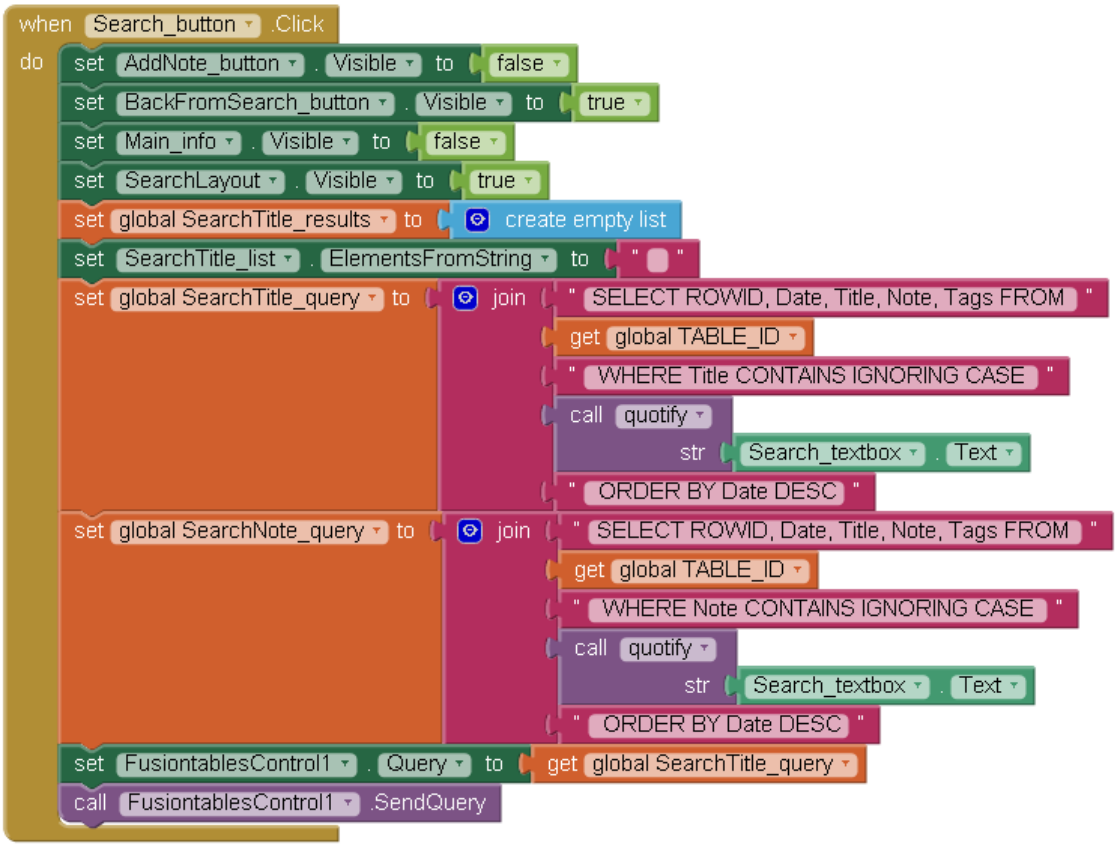
η οποία “τυλίγει” το string που της δίνεται σαν input σε “” για να μπορέσουμε να το χρησιμοποιήσουμε σε query.



Παρακάτω ρυθμίζουμε τη πλοήγηση του προγράμματος, δηλαδή τη περιήγηση μεταξύ οθονών.



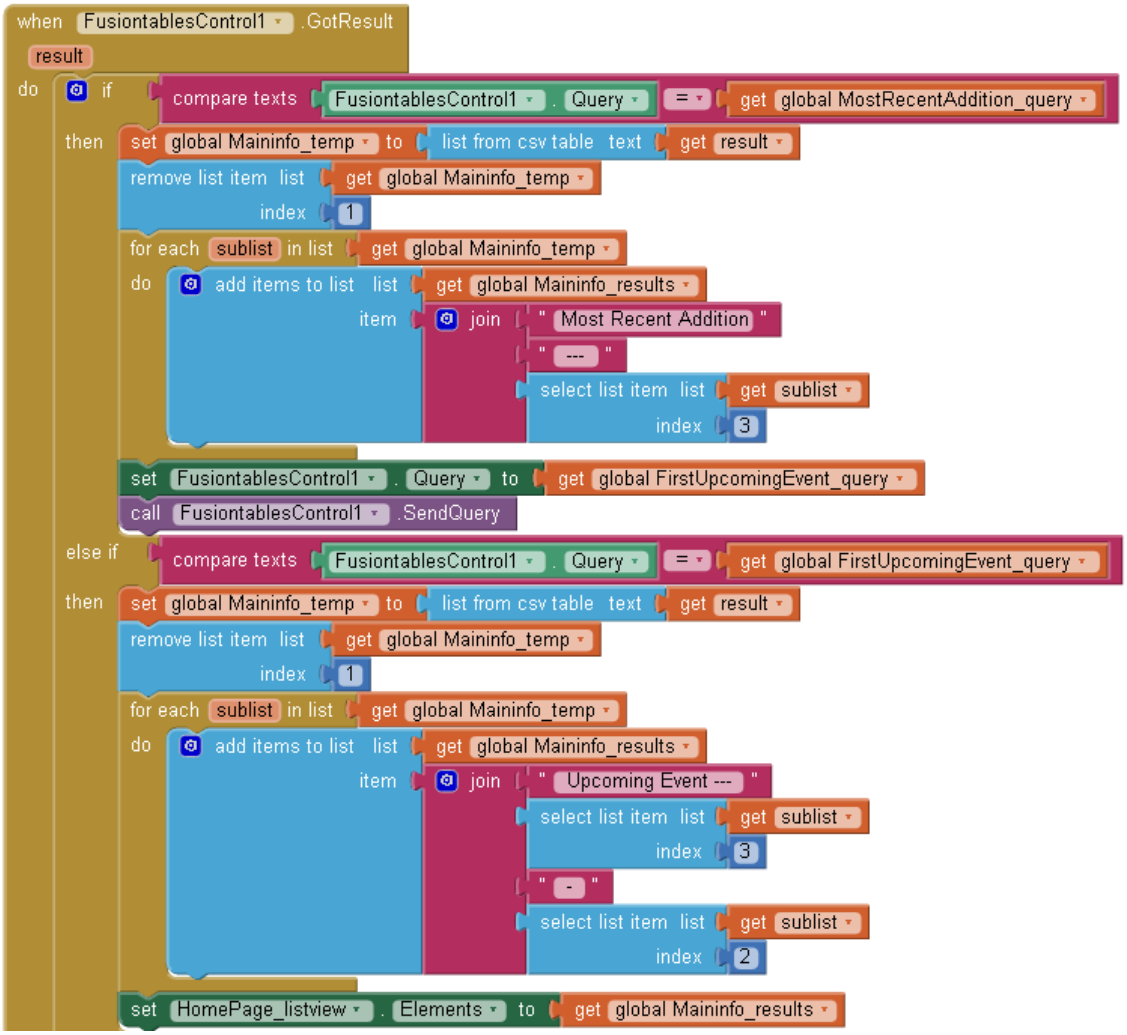
Η διαδικασία αναζήτησης παρουσιάζεται παρακάτω. Όταν πατήσουμε το search button κάνουμε hide το main_info layout, εμφανίζουμε το search layout και το back from search button. Έπειτα παίρνουμε το string που βρίσκεται στο search textbox και το υποβάλλουμε ως κριτήριο αναζήτησης και το υποβάλλουμε ως database query. Ενα query για τον τίτλο και ένα για το κύριο μέρος της αναζήτησης αφού το fusiont tables δεν επιτρέπει ως τελεστή το OR. Τέλος όταν πατηθεί το back from search γίνεται η αντίστροφη διαδικασία στην εμφάνιση/απόκρυψη layout.



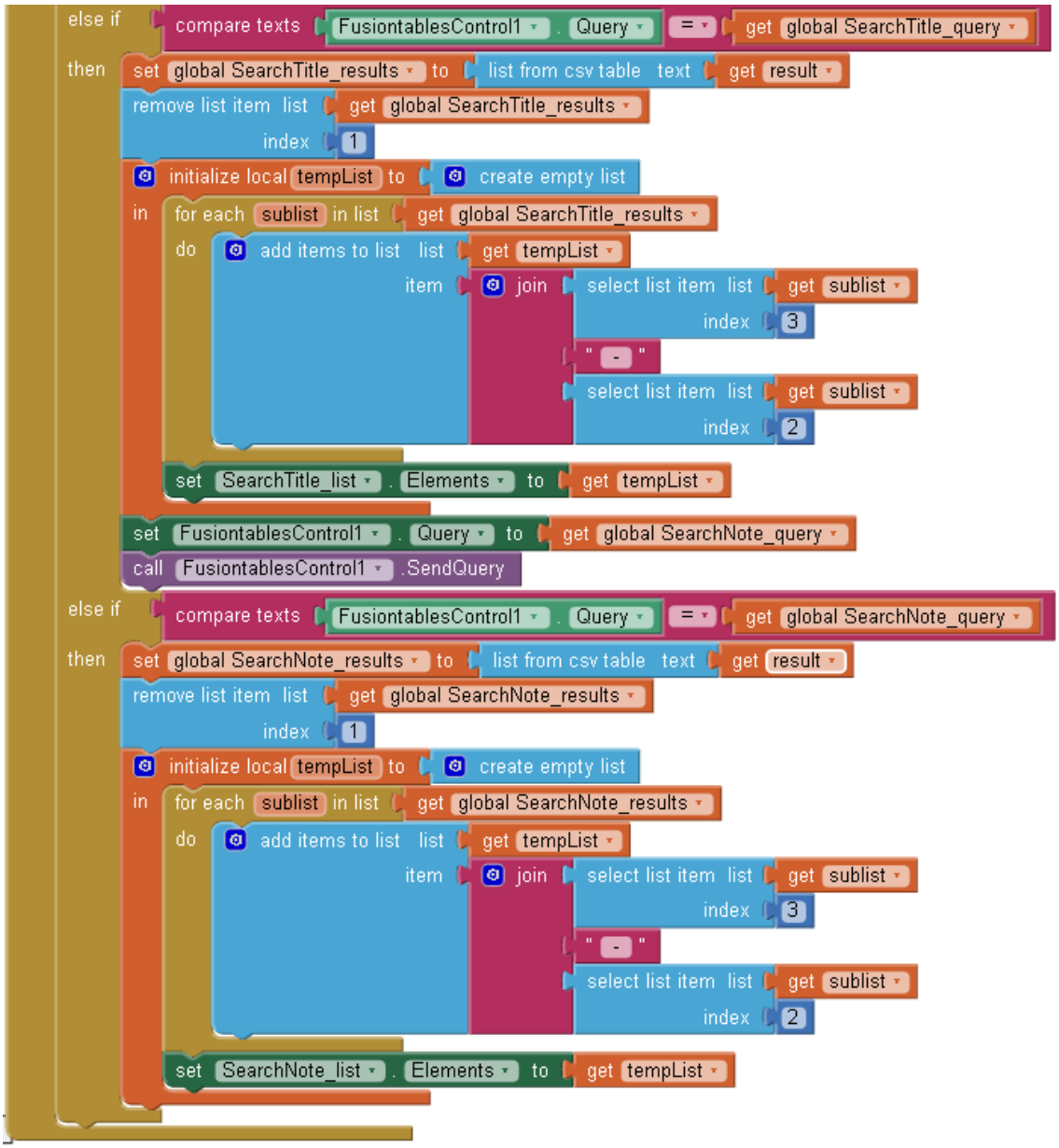
Τέλος θα περιγράψω το τελευταίο κομμάτι του παζλ που είναι υπεύθυνο για τη λειτουργικότητα της αρχικής οθόνης. Αυτό δεν είναι άλλο από τη συνάρτηση , η οποία είναι υπεύθυνη/εκτελείται όταν η βάση δεδομένων επιστρέφει απάντηση. Επειδή η συνάρτηση είναι μεγάλη και δεν χωράει σε ένα print screen θα τη σπάσω σε 2. Στο πρώτο μισό θα δούμε τον "κώδικα" που είναι υπεύθυνος για την προβολή της πιο πρόσφατης καταχώρησης και του πιο επερχόμενου event/reminder.

Αν θυμάστε στη συνάρτηση αρχικοποίησης είχαμε εκτελέσει το query για την πιο πρόσφατη καταχώρηση. Όταν λοιπόν η database στέλνει την απάντηση εμείς την κάνουμε catch με την if και εκτελούμε το κομμάτι του κώδικα που θέλουμε. Έτσι λοιπόν την απάντηση που παίρνουμε την καταχωρούμε στο Main_info list view και στη συνέχεια εκτελούμε αλυσιδωτά το δεύτερο query (επερχόμενο event/remider).

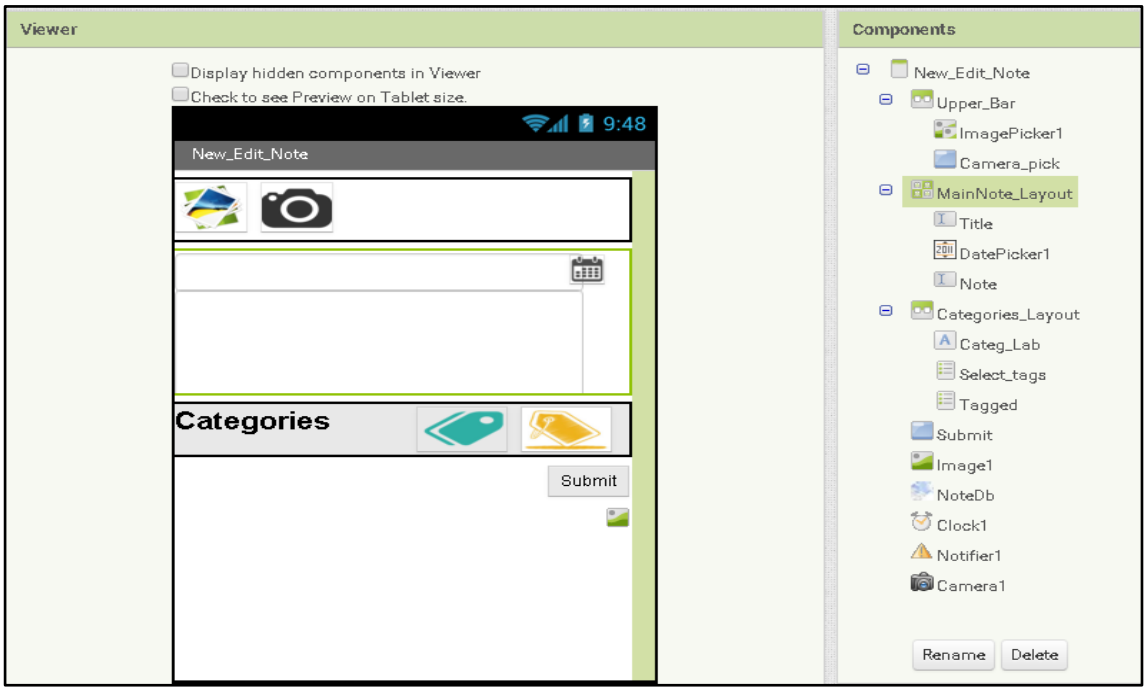
Μόλις λάβει απάντηση αυτή τη φορά εκτελείται η δεύτερη else if και το αντίστοιχο κομμάτι κώδικα, το οποίο βάζει επίσης την απάντηση στο Main_info listview για να προβληθούν. Αυτό που καταφέρνει αυτή η υλοποίηση είναι με το που εκτελείται το Most Recent query, αυτό να εκτελεί το First Upcoming Event query, επιτυγχάνοντας έτσι μια αλυσιδωτή αντίδραση για αυτά τα δύο αλληλένδετα queries.



Το δέυτερο κομμάτι της συνάρτησης αναλαμβάνει το response από τα search queries, και αναπαράγει άλλη μία αλυσιδωτή αντίδραση, αυτή τη φορά για τα searchTitle και searchNote queries


else if
compare texts
FusiontablesControl1
Query
=
get global SearchTitle_query
then
set global SearchTitle_results to
list from csv table text
get result
remove list item list
get global SearchTitle_results
index
1
initialize local tempList to
create empty list
in
for each sublist in list
get global SearchTitle_results
do
add items to list list
get tempList
item
join
select list item list
get sublist
index
3
" - "
select list item list
get sublist
index
2
set SearchTitle_list
Elements
to
get tempList
set FusiontablesControl1
Query
to
get global SearchNote_query
call FusiontablesControl1
.SendQuery
else if
compare texts
FusiontablesControl1
Query
=
get global SearchNote_query
then
set global SearchNote_results to
list from csv table text
get result
remove list item list
get global SearchNote_results
index
1
initialize local tempList to
create empty list
in
for each sublist in list
get global SearchNote_results
do
add items to list list
get tempList
item
join
select list item list
get sublist
index
3
" - "
select list item list
get sublist
index
2
set SearchNote_list
Elements
to
get tempList

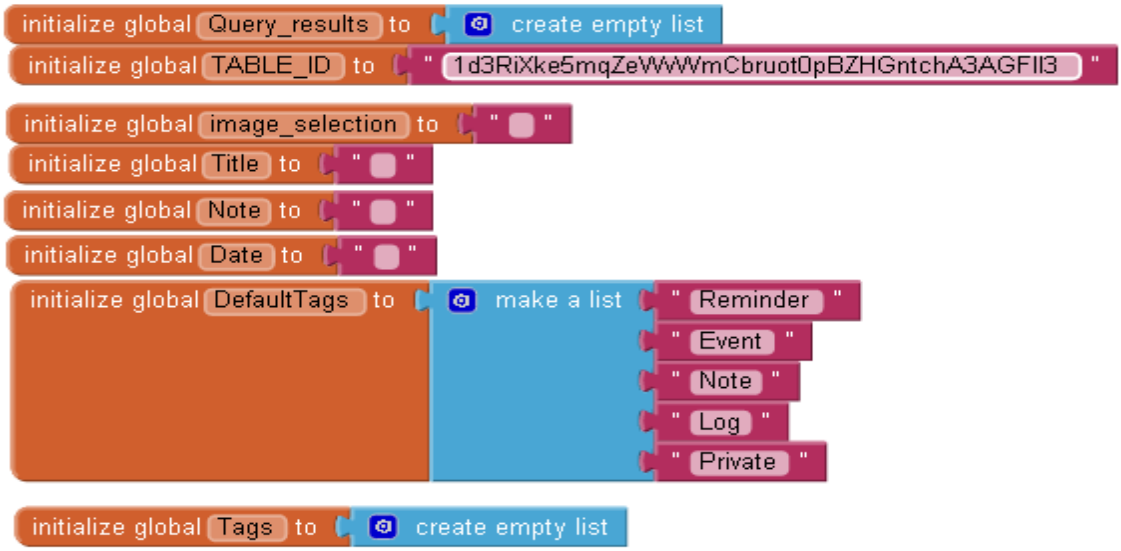
4.3.3 ΚΑΤΑΧΩΡΗΣΗ ΝΕΑΣ ΣΗΜΕΙΩΣΗΣ Η ΕΠΕΞΕΡΓΑΣΙΑ ΗΔΗ ΥΠΑΡΧΟΥΣΑΣ



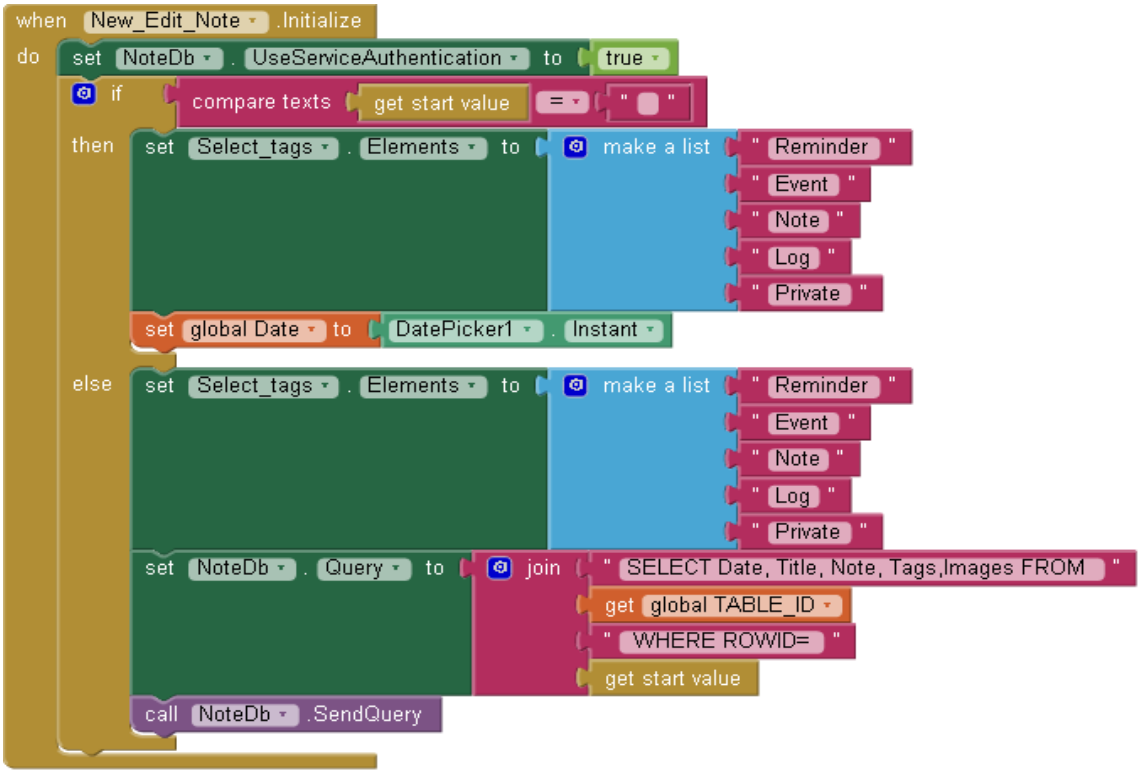
Όνομα Component	**Χρησιμότητα του Component**
Upper_Bar	Layout Component που περιλαμβάνει το image picker και το camera picker component, με τα οποία επιλέγουμε φωτογραφίες για τη σημείωση.
MainNote_layout	Layout που περιλαμβάνει το Title, Note textboxes στα οποία καταχωρούμε τα κύρια κομμάτια της σημείωσης καθώς και το Date picker
Categories_Layout	Layout που περιλαμβάνει 2 list pickers, το πρώτο περιέχει όλες τις διαθέσιμες κατηγορίες ενώ το δεύτερο αυτές που έχουμε επιλέξει από το πρώτο
Submit_button	Button που ξεκινάει τη διαδικασία αποθήκευσης
Noteb	Fusion Table component χρησιμεύει στην αποθήκευση των σημειώσεων
Clock	Clock Component που χρησιμοποιείται για να λάβουμε την ακριβή ώρα της καταχώρησης της σημείωσης
Notifier	Notifier Component που χρησιμοποιείται για να δείχνει alert στον χρήστη καθώς και για να επιλογή ενεργειών

Αρχικοποίηση

Κάνουμε και εδώ κάποια βήματα αρχικοποίησης τα οποία φαίνονται παρακάτω. Όπως και πριν το tableId στατικό, το defaultTags τα πιθανά tags η λίστα Tags αυτά που έχουν επιλεχθεί.

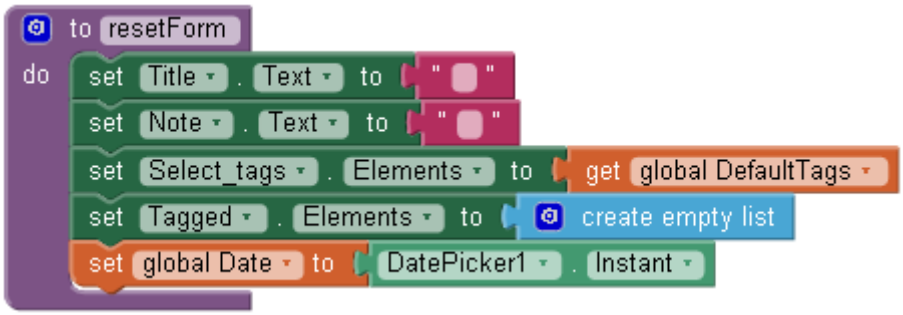


Η συνάρτηση initialize εδώ έχει ένα ιδιαίτερο ρόλο αφού αν το start value (είναι οι τιμές που περνιούνται από μία οθόνη σε μία άλλη - και εδώ της δίνουμε την τιμη rowid ή αλλιώς το id της σημείωσης που θέλουμε να κάνουμε edit) είναι κενό σημαίνει πως θέλουμε να καταχωρήσουμε νέα σημείωση, σε διαφορετική περίπτωση (η start value εχει τιμή) θέλουμε να κάνουμε edit και υποβάλλουμε query με id σημείωσης το start value. Έτσι θα πάρουμε πίσω τις τιμές της σημείωσης που μας ενδιαφέρει για να τις μεταβάλλουμε πιθανώς και να τις αποθηκεύσουμε ξανά.



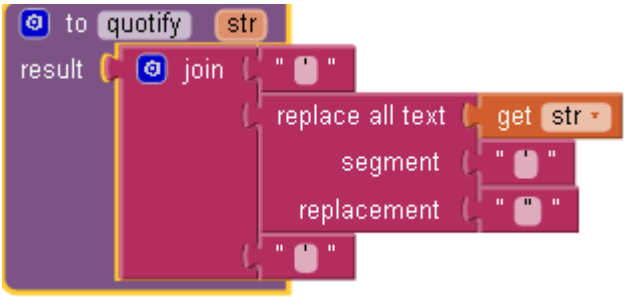
Συναρτήσεις που χρησιμοποιούμε

Η συνάρτηση αυτή επαναφέρει τις τιμές στις αρχικές τους τιμές

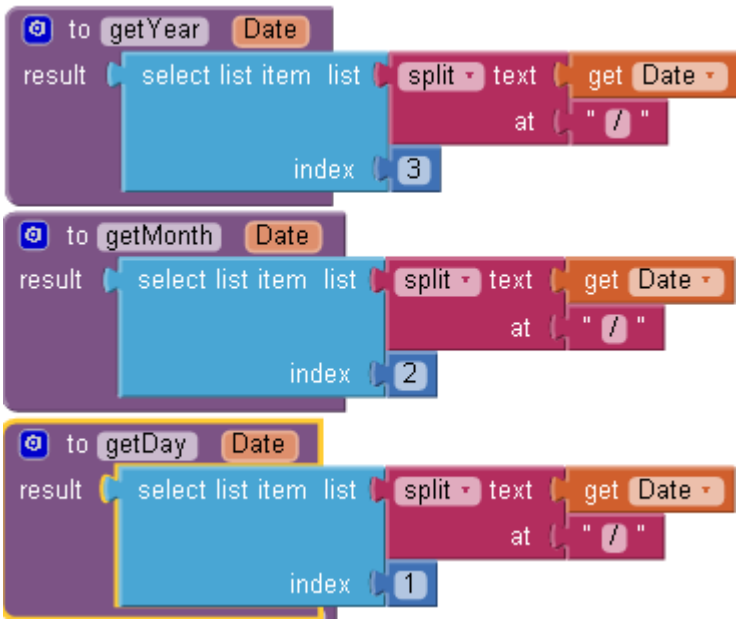


Η συνάρτηση αυτή ελέγχει αν έχουν καταχωρηθεί τιμές στα δύο βασικά πεδία Title και Note, αν δεν έχει εισαχθεί σε κανένα δεν τίθεται θέμα καταχώρησης και η εφαρμογή προτρέπει τον χρήστη να τα εισάγει.

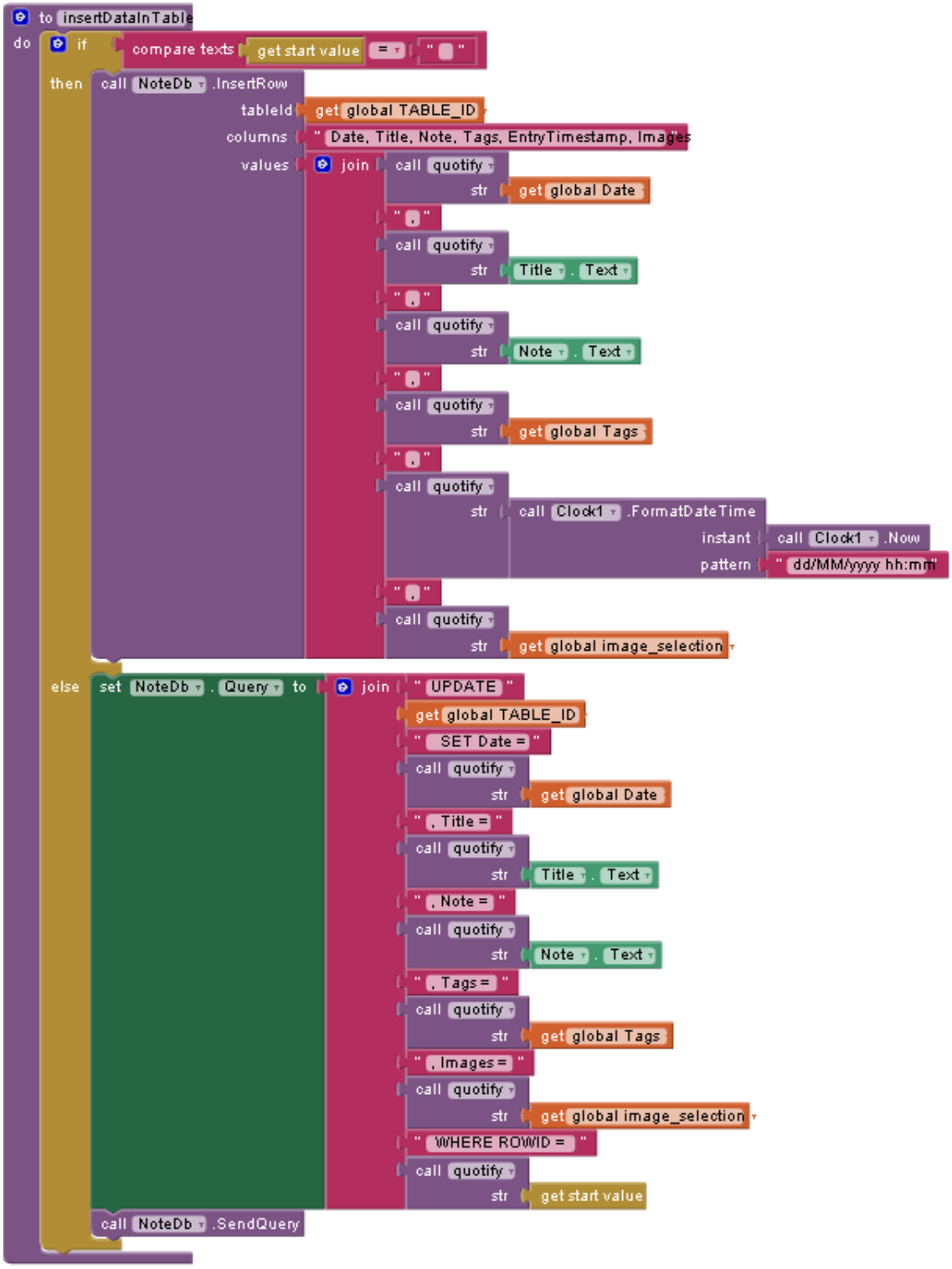
Για την quotify έχουμε αναφερθεί προηγουμένως.



Οι συναρτήσεις αυτές εξάγουν ξεχωριστά τα επιμέρους κομμάτια μιας ημ/νιας.

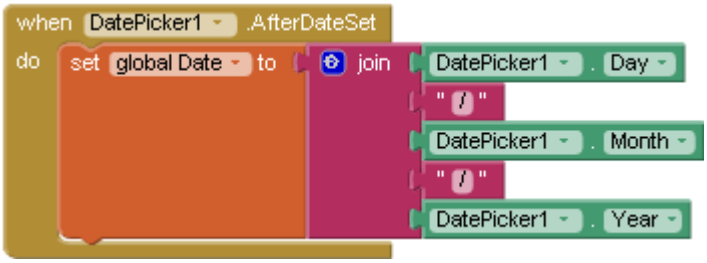


Τέλος η συνάρτηση InsertDatainTable αν είναι νέα η σημείωση που θέλουμε να καταχωρήσουμε απλώς την εισάγει στη βάση, αλλιώς (σημείωση προς επεξεργασία) τότε ενημερώνει τις τιμές της σημείωσης προς επεξεργασία στη βάση δεδομένων.



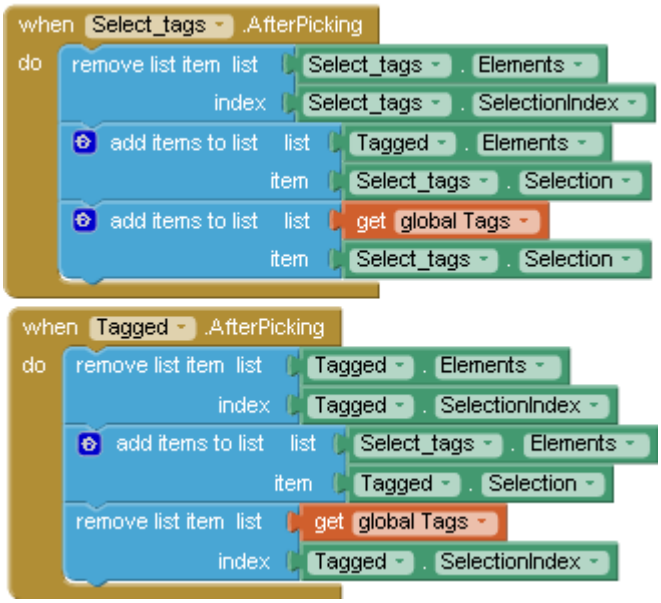
Blocks συμπεριφορών

Το DatePicker όταν εκτελεστεί ένα event select (όταν δηλαδή ο χρήστης επιλέξει ημ/νια) τότε το block αυτό καταχωρεί στη μεταβλητή Date την ημ/νία στο format dd/MM/yyyy.



Το tag picker λειτουργεί ως εξής, υπάρχουν δύο λίστες, μία που περιέχει όλα τα πιθανά tags και μία κενή που θα περιέχει αυτά που έχουν επιλεχθεί από την πρώτη λίστα. Έτσι λοιπόν όταν επιλεχθεί ένα tag από την λίστα προς επιλογή, το στοιχείο αυτό αφαιρείται απο αυτή τη λίστα και προστίθεται στην άλλη. Σκοπός μας είναι και κανένα στοιχείο να μην περιέχεται και στις δύο λίστες ταυτόχρονα, άρα όσο πιο πολλά στοιχεία επιλεχθούν τόσο λιγότερα θα μένουν προς επιλογή.

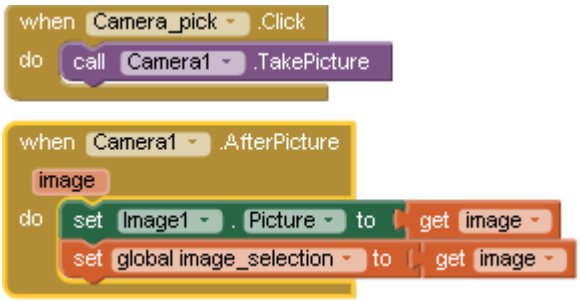
Αντίστροφα αν ένα στοιχείο επιλεχθεί από την λίστα Tagged () τότε στην ουσία αφαιρείται και προστίθεται πάλι στη λίστα Select_tags. Έτσι ο χρήστης μπορεί ελεύθερα να προσθέτει και να αφαιρεί tags.



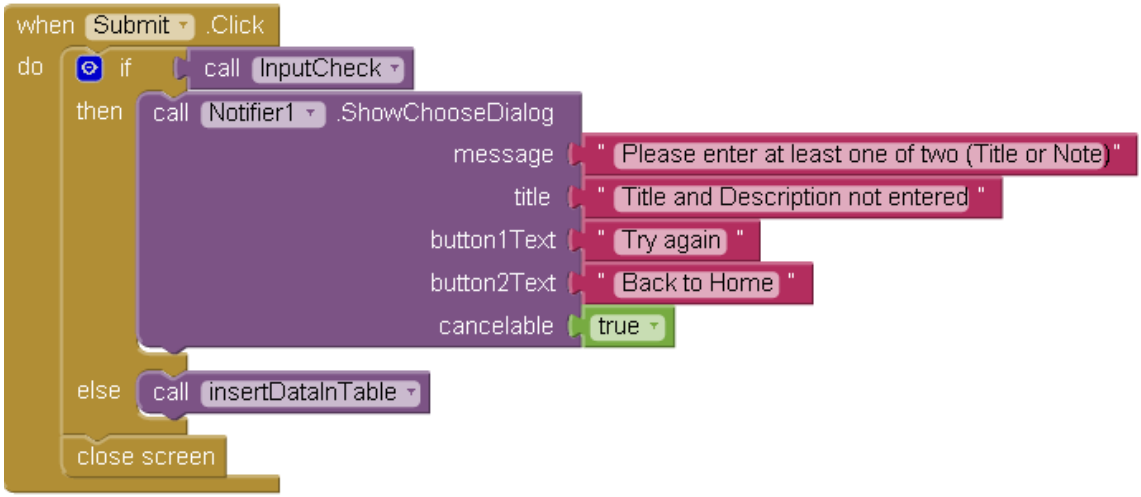
Όταν πατηθεί το Back (το φυσικό back button στη συσκευή) τότε επιστρέφει πίσω στην αρχική.

Όταν εκτελεστεί το event pick (δηλαδή ο χρήστης επιλέξει photo από το gallery) τότε η φωτογραφία αυτή προβάλλεται σε ένα image component και καταχωρείται στη μεταβλητή image_selection για αποθήκευση στη DB (στην πραγματικότητα δεν αποθηκεύεται η εικόνα αλλά η διεύθυνση οποία βρίσκεται αυτή στην συσκευή του χρήστη).

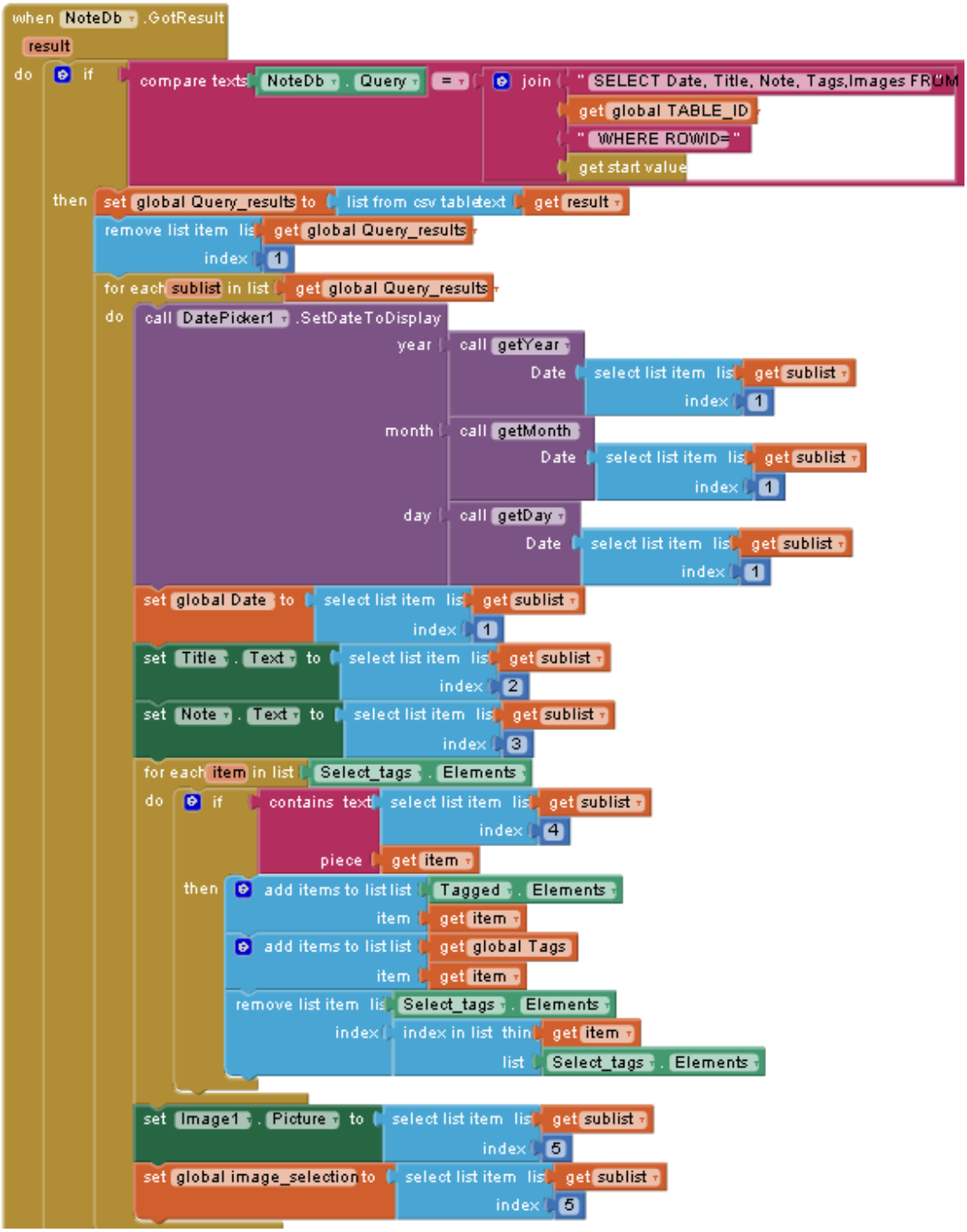
Όταν πατηθεί το camera button τότε εκκινείται το take picture method και ανοίγει η κάμερα, στην συνέχεια όταν ο χρήστης πάρει τη φωτογραφία που επιθυμεί η διαδικασία είναι ακριβώς η ίδια με το image picker.



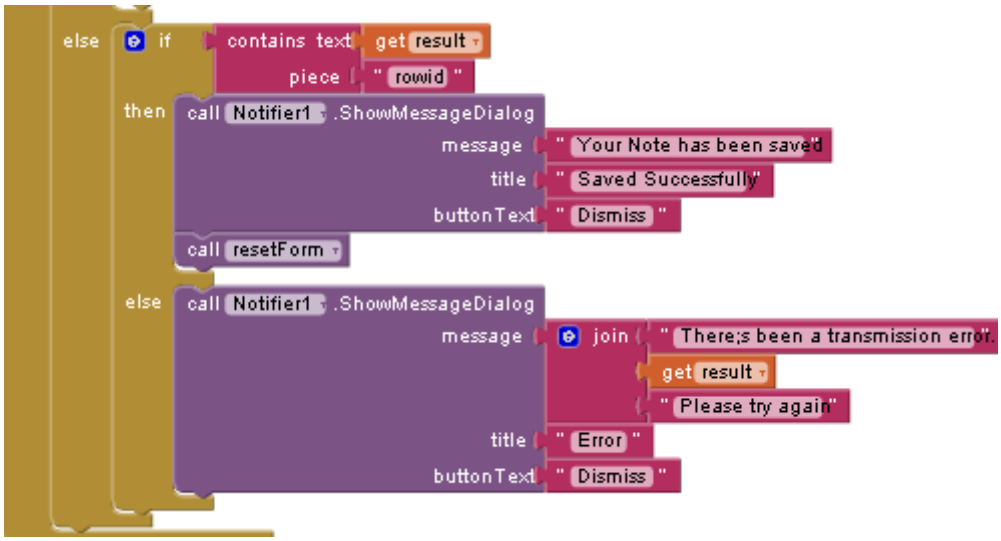
Το submit button μόλις πατηθεί καλεί την συνάρτηση inputCheck η οποία ελέγχει τα inputs. Αν υπάρχει πρόβλημα καλεί το notifier και εμφανίζει το κατάλληλο μήνυμα στο χρήστη. Αν όλα είναι σωστά τότε καλεί τη συνάρτηση insertDataInTable και καταχωρεί ή αναβαθμίζει την σημείωση.



Τέλος έχουμε τη gotresult που όπως έχουμε πει προηγουμένως αναλαμβάνει να διαχειριστεί το response απο την ΒΔ. Εδώ λοιπόν (στο πρώτο print screen) όταν γίνεται query απο την initialize το αποτέλεσμα που επιστρέφεται καταχωρείται και προβάλλεται στο αντίστοιχο πεδίο του (δηλαδή το Title στο Title, το Date στο date picker το image στο image κτλπ).

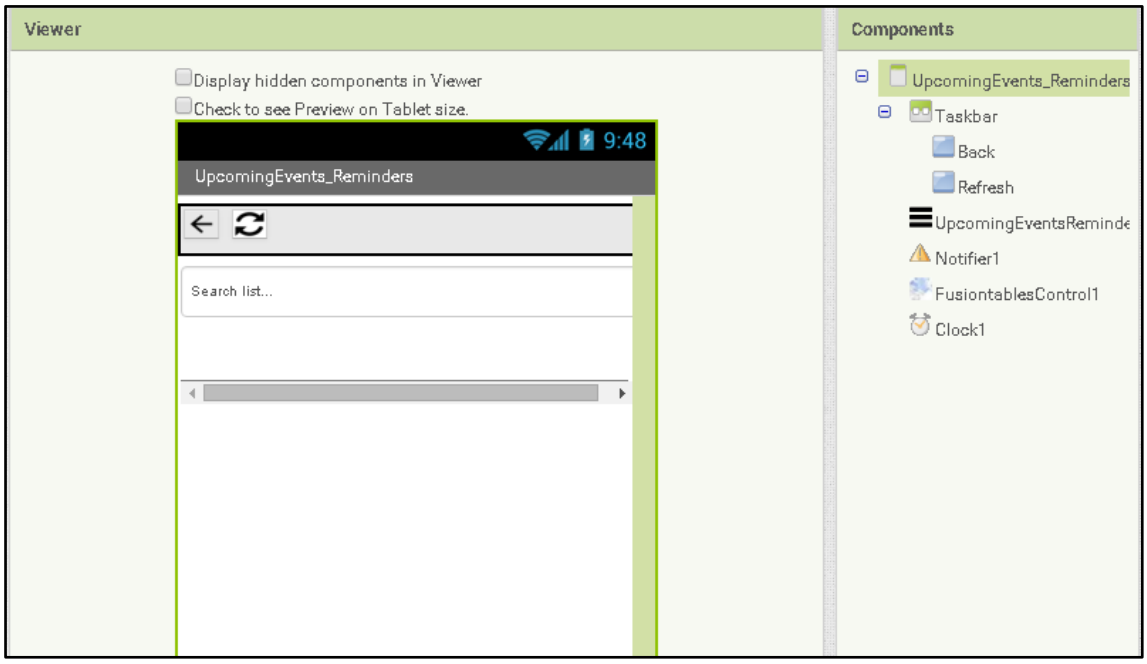


Στο else (δεύτερο μισό), απλώς εμφανίζει αν η σημείωση έχει καταχωρηθεί/ενημερωθεί με επιτυχία η αν έχει αποτύχει για οποιοδήποτε λόγο (σφάλμα δικτύου κτλπ). Αυτό επιτυγχάνεται ανάλογα με την απάντηση της DB.



4.3.4 UPCOMING EVENTS, MY JOURNAL, RECENT ADDITIONS, PRIVATE SCREENS

Όλες αυτές οι οθόνες χρησιμοποιούν ακριβώς τα ίδια components με μόνη διαφορά ότι προβάλουν διαφορετικές καταχωρήσεις(υποβάλλεται διαφορετικό query στο fusion tables)

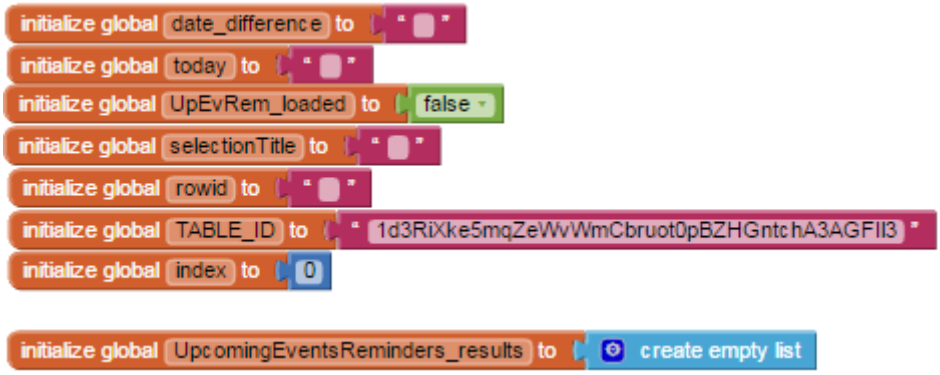


Όνομα Component	**Χρησιμότητα του Component**
Taskbar	Layout component που περιέχει το back button που επιστρέφει στην αρχική και το refresh button που υποβάλλει ξανά αίτημα στη βάση δεδομένων
UpcomingEventsReminder_listview	Το listview component που χρησιμοποιείται για να προβάλει τα αποτελέσματα που επιστρέφει η db
Notifier	Notifier Component που εμφανίζεται κάθε φορά που επιλέγεται μία σημείωση από τη παραπάνω λίστα και χρησιμοποιείται για την επιβεβαίωση της διαγραφής αλλά και επιλογή της λειτουργίας επεξεργασίας.
Clock	Clock component που χρησιμοποιείται για την έρευση της σημερινής ημερομηνίας ώστε κάθε φορά να στέλνονται οι

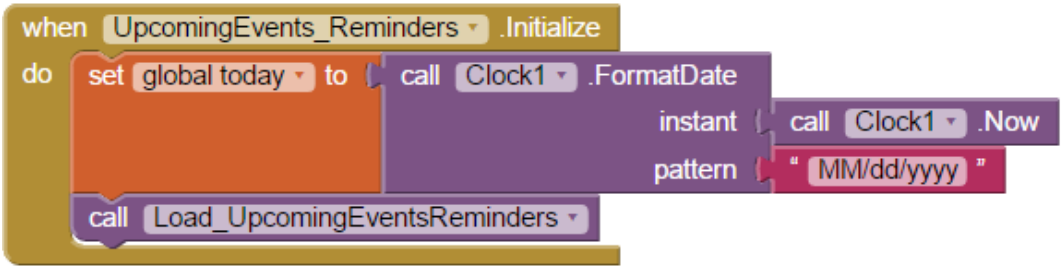
	επιθυμητές σημειώσεις, π.χ. για το upcoming events απο σήμερα και μετά, για το MyJournal από χθες και παλαιότερα. για το RecentAdditions ελέγχεται το στιγμιότυπο καταχώρησης

Αρχικοποίηση

Οι μεταβλητές που θα χρειαστούμε κατά τη διάρκεια της εκτέλεσης. Θα χρειαστεί να επεξηγήσουμε μόνο το rowid, SelectionTitle, index. Κάθε φορά που θα επιλέγεται από το εκάστοτε Listview μία καταχώρηση, θα κρατάμε αυτήν την επιλογή σε μία μεταβλητή για μετέπειτα χρήση. Παρομοίως για το selectionTitle το οποίο θα εμφανίζεται όταν γίνεται ενέργεια διαγραφής(πχ. θα εμφανίζεται στο Notifirier θέλετε να διαγράψετε το …..)

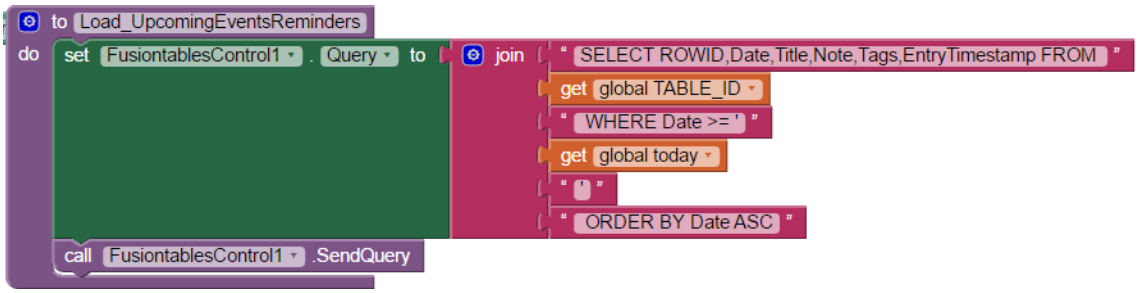


Η συνάρτηση αρχικοποίησης όπως βλέπουμε αρχικοποιεί το today και εκτελεί τη συνάρτηση load_upcomingEventsReminders.

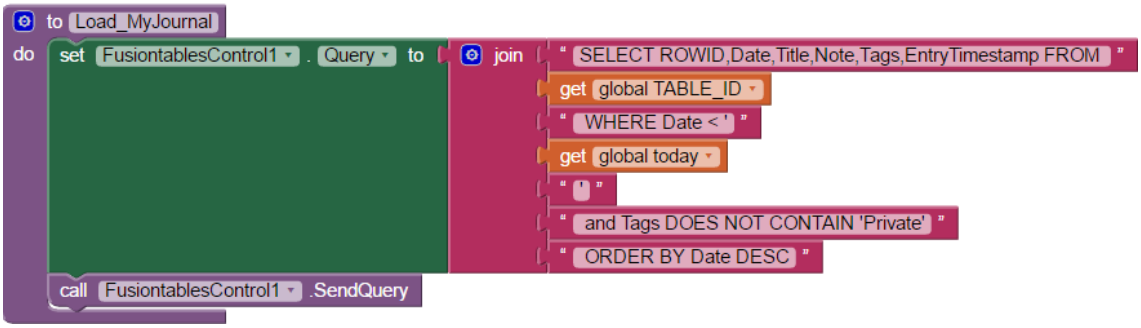


Συναρτήσεις που χρησιμοποιούμε

Συνάρτηση που υποβάλλει το αίτημα προς τη βάση δεδομένων. Για τη σελίδα UpcomingEventsReminders θέλουμε τις καταχωρήσεις που έχουν ημ/νία από σήμερα και έπειτα.



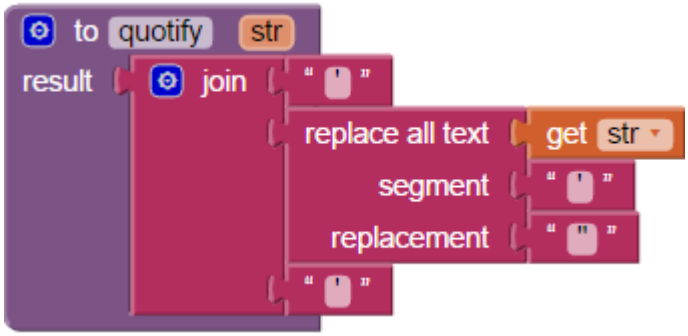
Για τη σελίδα MyJournal θα χρειαστούμε τις παρελθοντικές καταχωρήσεις, δηλαδή αυτές με ημ/νία μικρότερη της σημερινής.



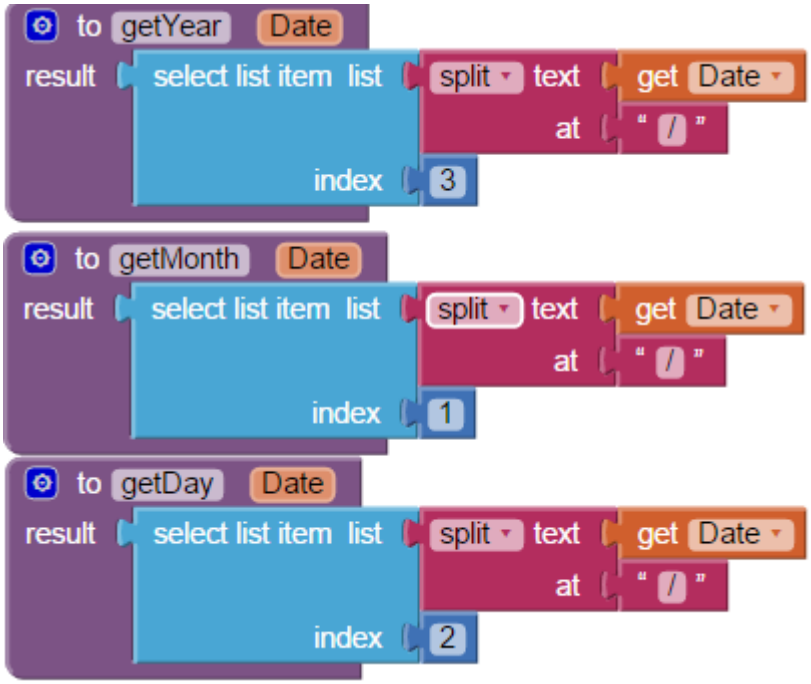
Για τη σελίδα RecentAdditions όλες τις καταααχωρήσεις, ταξινομημένες ανάλογα με την ημερομηνία καταχώρησης.

Για την σελίδα Private όλες αυτές που έχουν το tag Private

Έχουμε αναφερθεί και παραπάνω στη συνάρτηση quotify, εν συντομία τυλίγει το κείμενο που της δίνεται σε ¨¨.

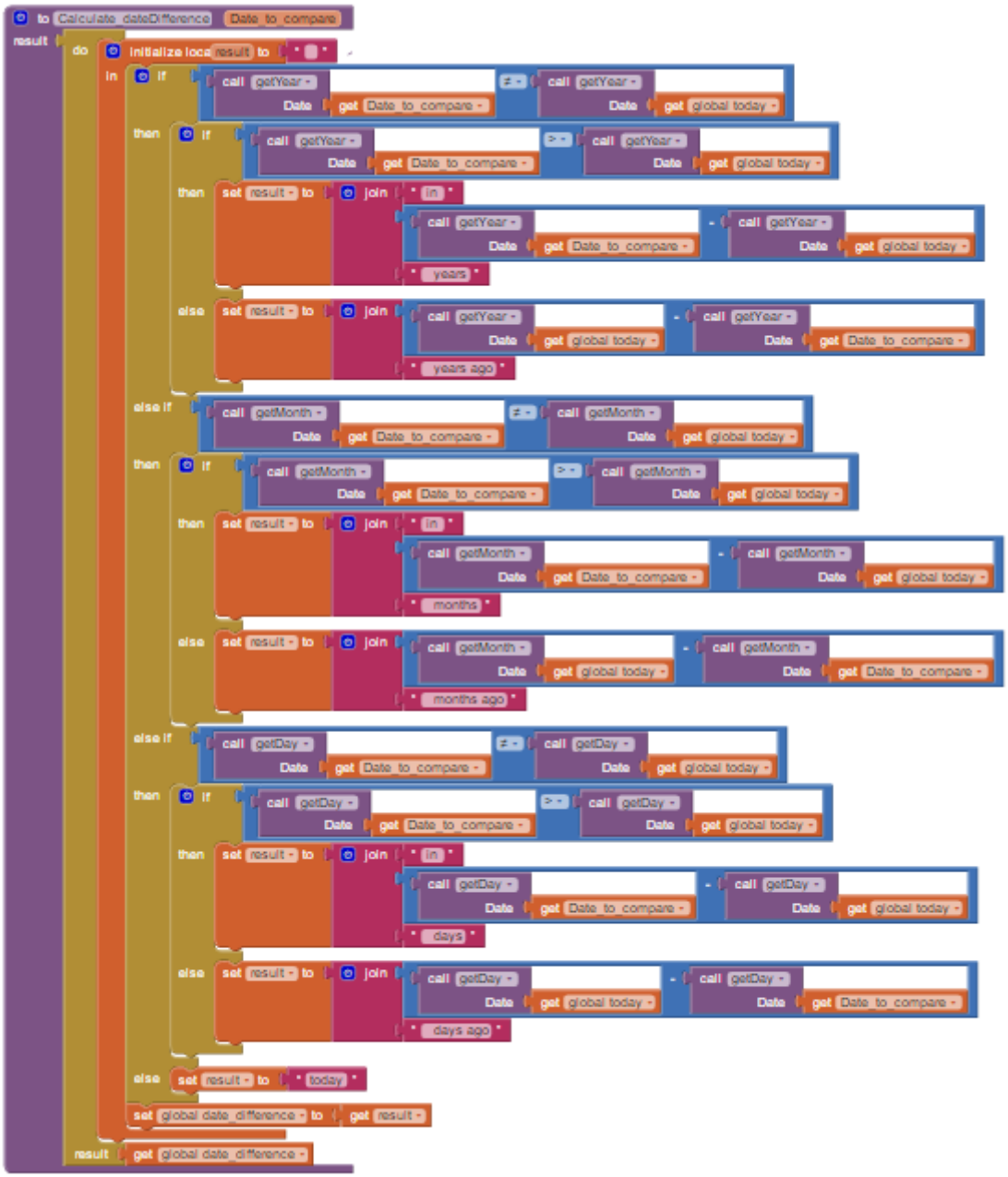


Συναρτήσεις που εξάγουν τα επιμέρους στοιχεία μιας Ημ/νίας της μορφής 18/04/2016.



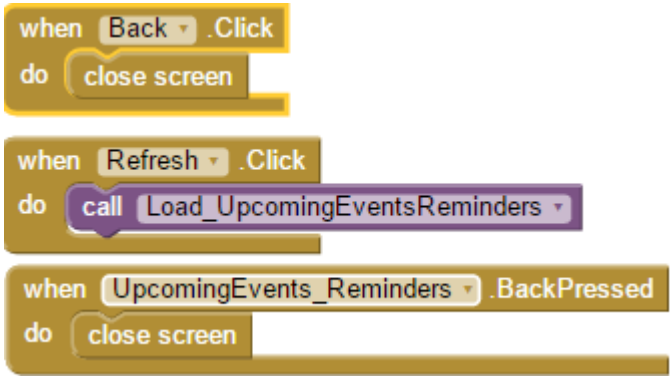
Παρακάτω είναι η συνάρτηση που υπολογίζει τη διαφορά μεταξύ 2 ημ/νιών (της σημερινής με μίας άλλης) ώστε π.χ. στα αποτελέσματα να φαίνεται το χρονικό διάστημα που μεσολαβεί και όχι μία αδιάφορη προς το χρήστη ημερομηνία.

1- Attend Russian Classes - in 2 months
2- a year from now - in 1 years
3- Olympic to Tokyo - in 4 years

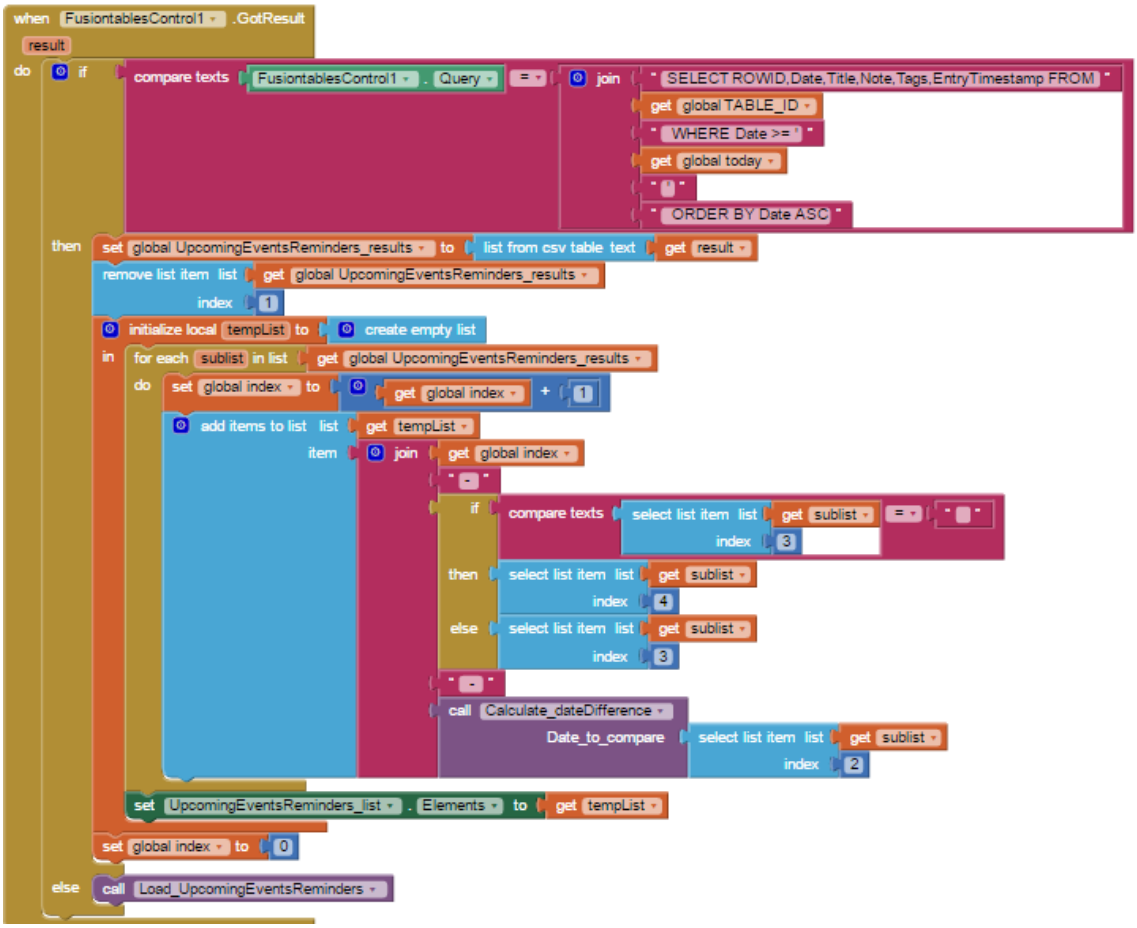


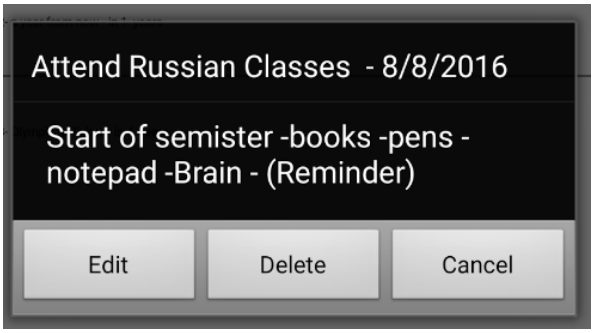
Blocks συμπεριφορών

Παρακάτω φαίνεται τι γίνεται όταν πατηθούν τα βασικά κουμπιά, πχ όταν πατηθεί είτε το back button είτε το physical back key η οθόνη θα κλείσει και το πρόγραμμα θα ανοίξει την προηγούμενη του. Στην περίπτωση του refresh απλώς θα ξαναφορτώσει την εκάστοτε σελίδα.

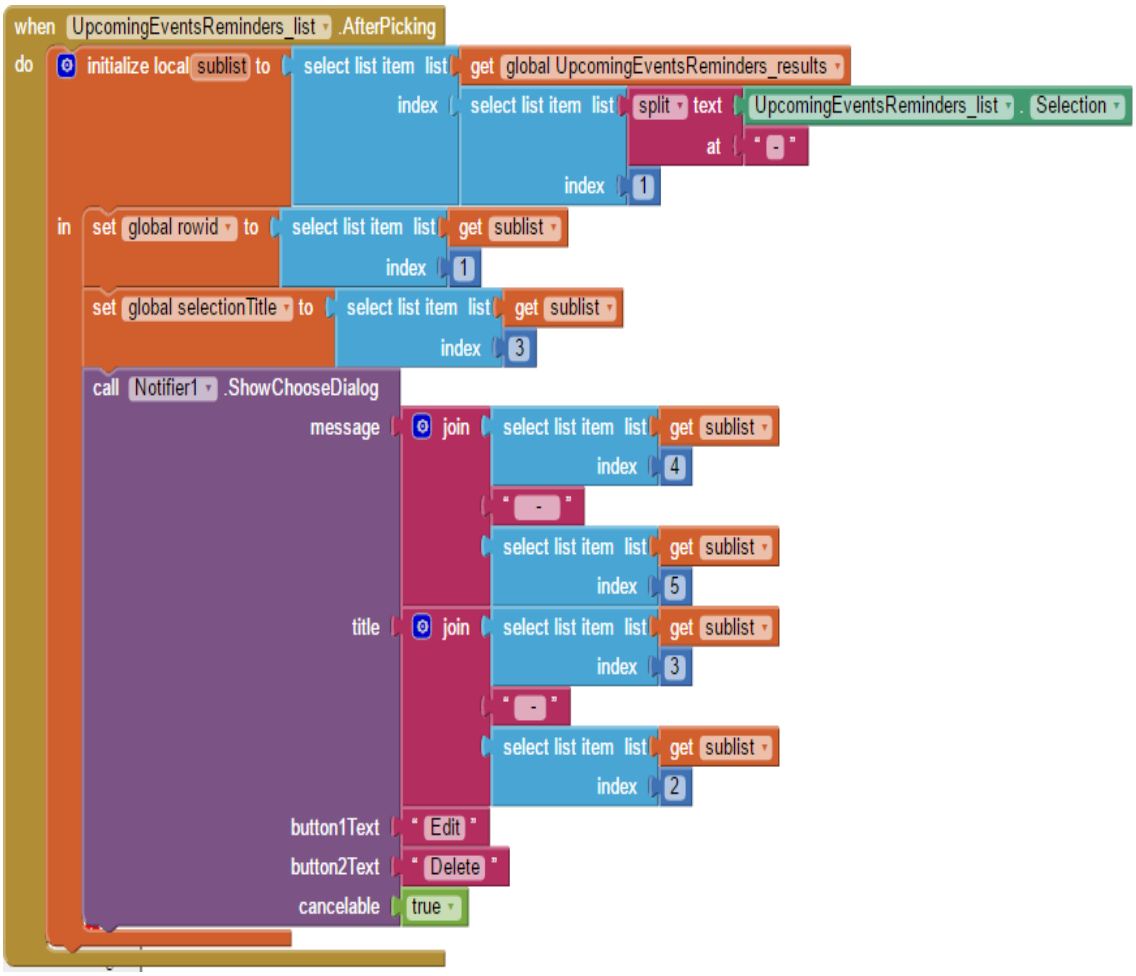


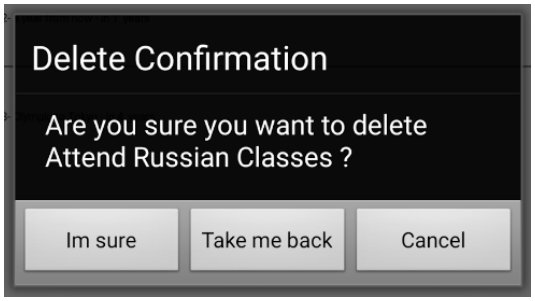
Πατρακάτω φαίνεται η συνάρτηση got.result που διαχειρίζεται το αποτέλεσμα που επιστρέφει η ΒΔ. Στη συνέχεια κάθε καταχώρηση τη βάζει στο listview μαζί με τον χρόνο που απομένει ως της πλήρωση της.



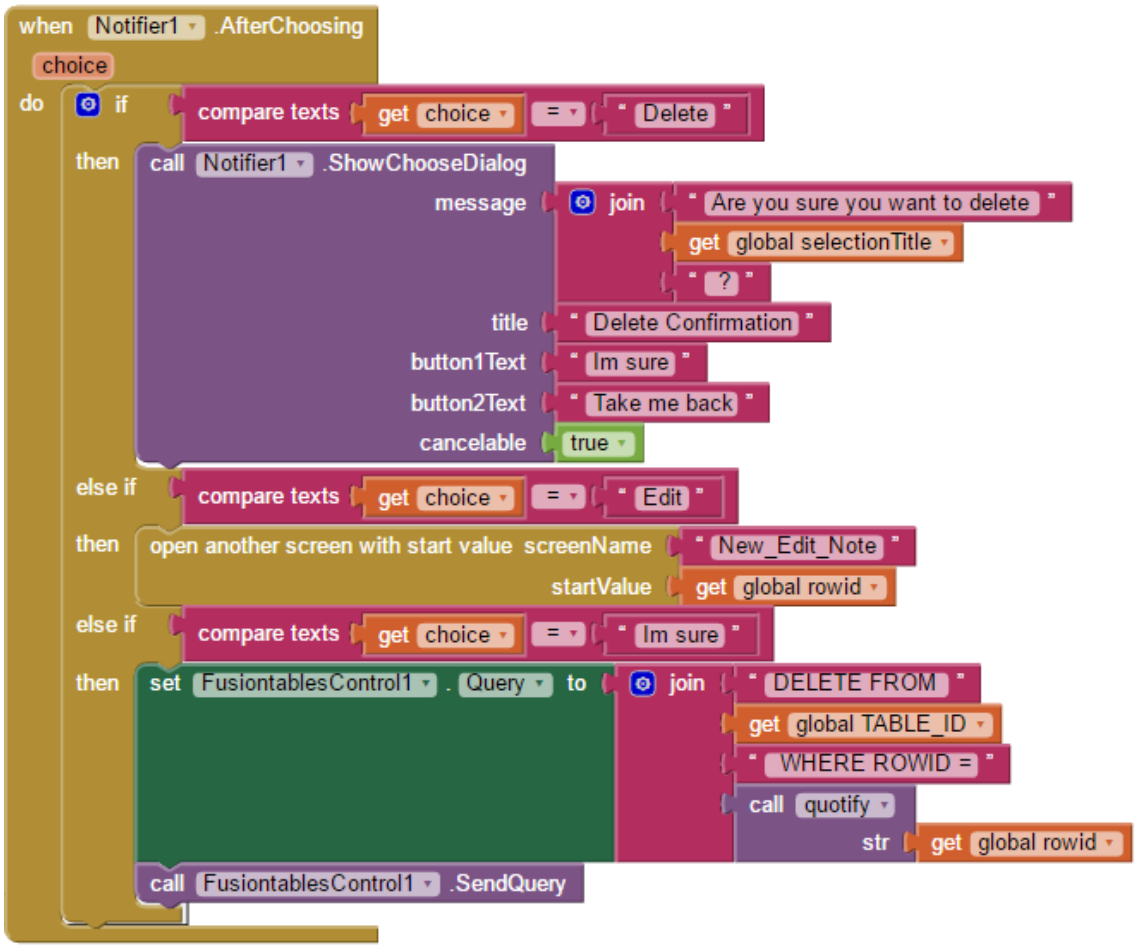


Παρακάτω φαίνεται η συνάρτηση που χειρίζεται τα selection events, όταν δηλαδή επιλεχθεί κάποια καταχώρηση από τη λίστα. Τότε κρατάει την επιλογή στην μεταβλητή rowid και ανοίγει έναν Notifier με τις διαθέσιμες ενέργειες(διπλανή εικόνα). Σκοπός είναι να γνωρίζουμε τι επιλέχθηκε ώστε σε συνδυασμό με την επιθυμητή ενέργεια (edit ή delete) να εκτελέσουμε ακριβώς ότι χρειάζεται για να γίνουν οι παραπάνω διαδικασίες.





Τέλος φαίνεται η συνάρτηση που διαχειρίζεται τις επιλογές που εμφανίζει ο notifier. Πχ αν επιλέξουμε delete εμφανίζει εκ νέου έναν ακόμα Notifier με διαφορετικό μήνυμα αυτή τη φορά το οποίο φαίνεται δίπλα και προτρέπει το χρήστη να επιβεβαιώσει την επιλογή του (Im sure). Συνεχίζοντας τώρα, αν επιλεχθεί πλέον το Im sure στέλνεται ένα έτοιμα delete στη βάση για την συγκεκριμένη καταχώρηση και έπειτα ανανεώνεται η οθόνη. Τέλος αν πατηθεί το edit ανοίγει την οθόνη New_Edit_Note περνώντας ως αρχικό value το RowID της σημείωσης.

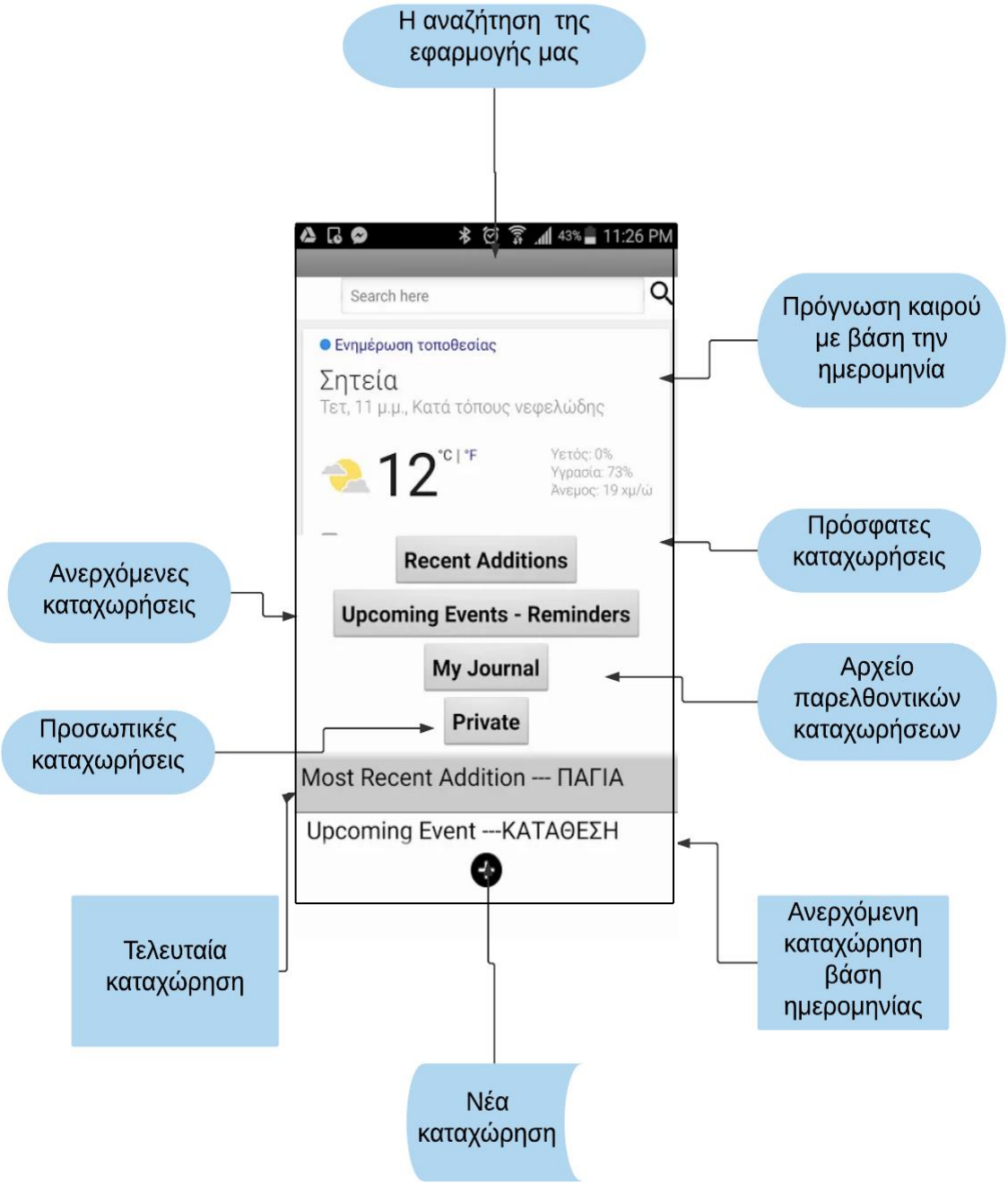

when Notifier1 .AfterChoosing
choice
do if compare texts get choice = " Delete "
then call Notifier1 .ShowChooseDialog
message join " Are you sure you want to delete "
get global selectionTitle
" ? "
title " Delete Confirmation "
button1Text " Im sure "
button2Text " Take me back "
cancelable true
else if compare texts get choice = " Edit "
then open another screen with start value screenName " New_Edit_Note "
startValue get global rowid
else if compare texts get choice = " Im sure "
then set FusiontablesControl1 . Query to join " DELETE FROM "
get global TABLE_ID
" WHERE ROWID = "
call quotify
str get global rowid
call FusiontablesControl1 .SendQuery

ΚΕΦΑΛΑΙΟ 5: ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Παρακάτω γίνεται παρουσίαση της εφαρμογής και των λειτουργιών της. Δίδονται επεξηγήσεις για τη σημασία κάθε στοιχείου και για όλες τις οθόνες και επιλογές που διαθέτει η εφαρμογή.

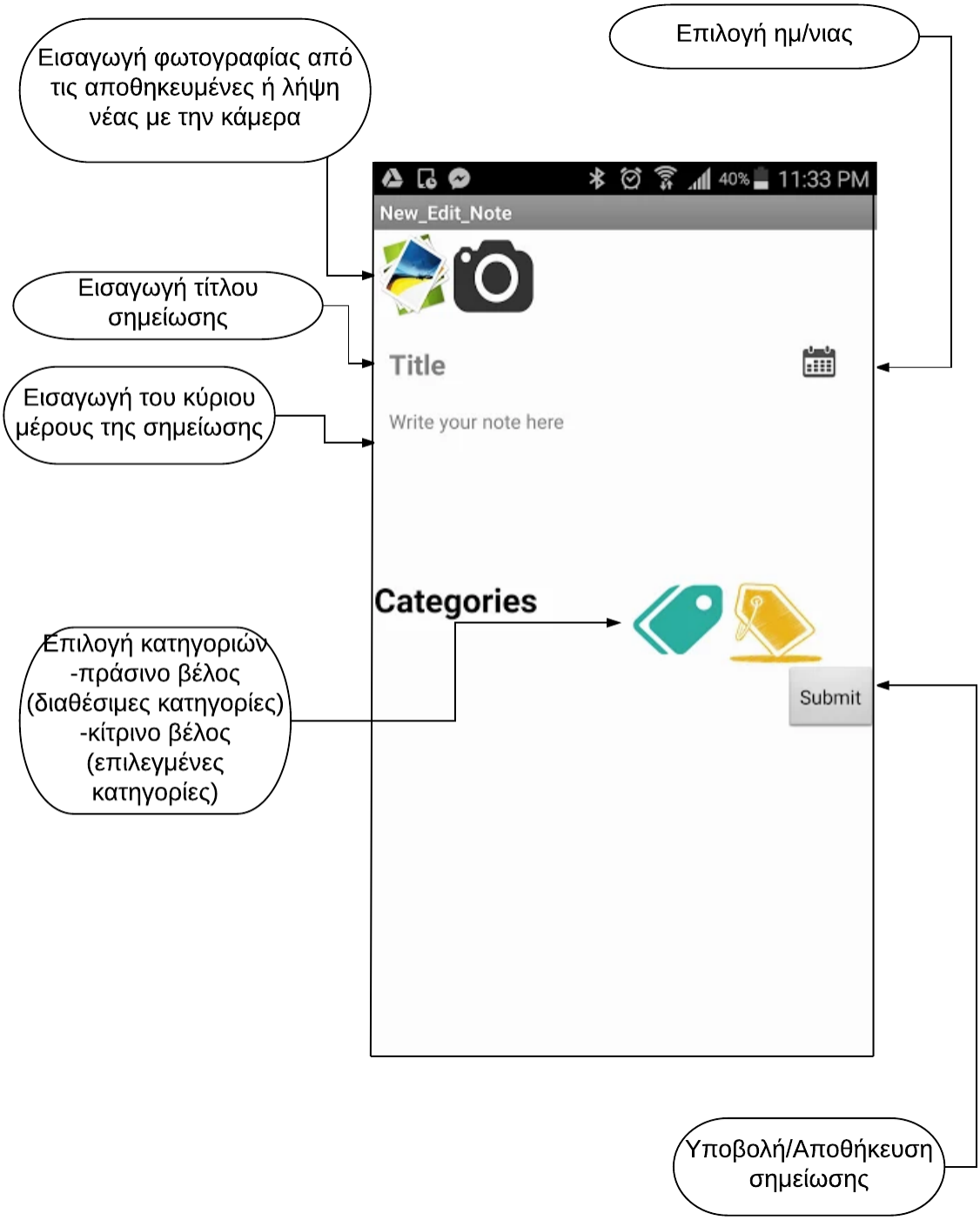
5.1 ΚΕΝΤΡΙΚΗ ΟΘΟΝΗ

Αυτή είναι η κεντρική οθόνη που εμφανίζεται με την έναρξη της εφαρμογής. Είναι η βασική οθόνη που περιέχει όλες τις δυνατότητες και επιλογές που έχει ο χρήστης.



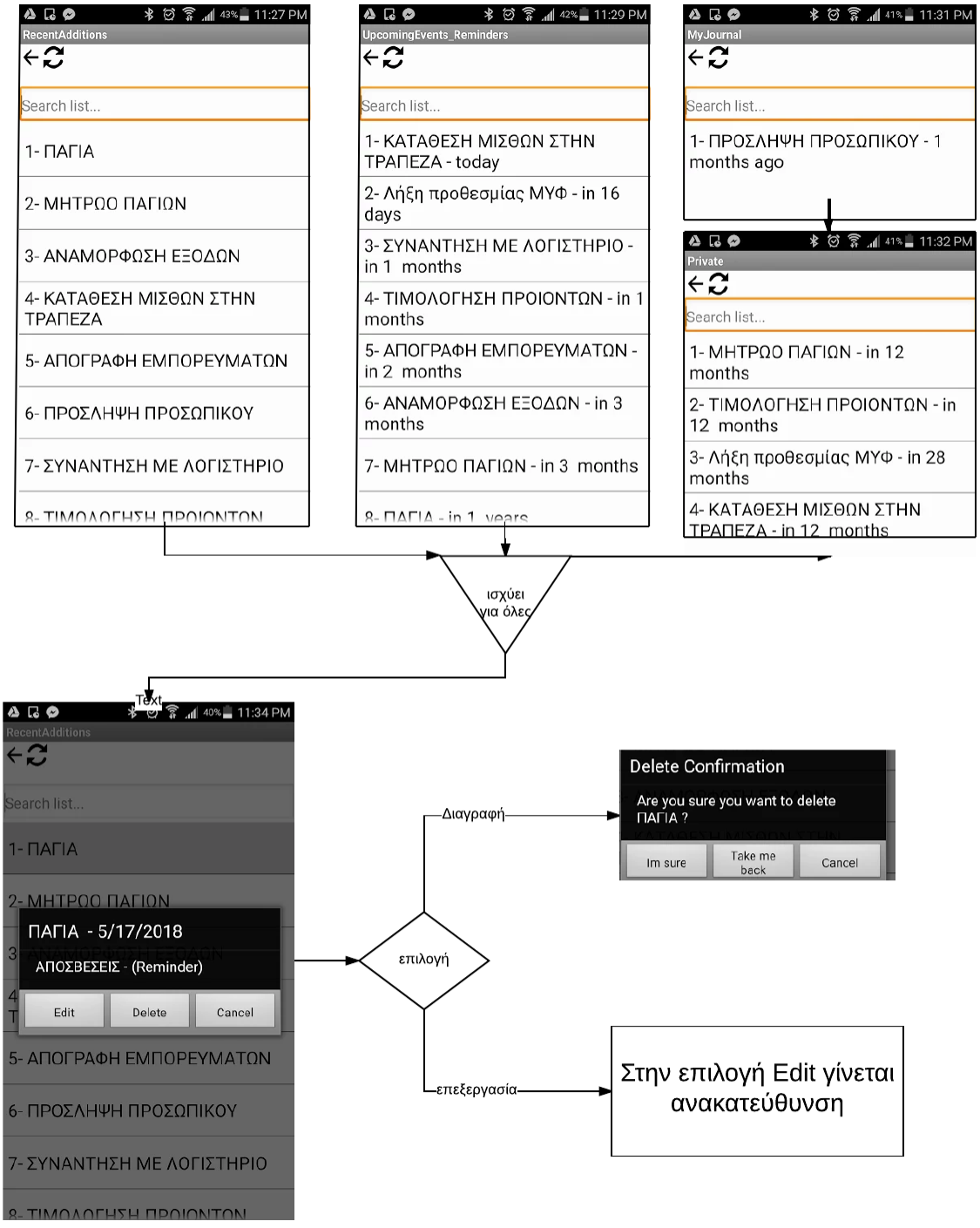
5.2 ΠΡΟΣΘΗΚΗ ΝΕΑΣ ΚΑΤΑΧΩΡΗΣΗΣ

Αυτή η οθόνη εμφανίζεται όταν ο χρήστης επιθυμεί την προσθήκη μίας νέας καταχώρησης. Η ίδια οθόνη παρουσιάζεται και για την επεξεργασία μίας καταχώρησης μόνο που όλα τα πεδία θα φαίνονται συμπληρωμένα.



5.3 ΟΙ ΟΘΟΝΕΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Παρακάτω φαίνονται οι οθόνες για τις κατηγοριοποιημένες καταχωρήσεις που έχει κάνει ο χρήστης καθώς και τα παράθυρα που παρουσιάζονται σε περίπτωση που επιλεγεί μία καταχώρηση προς επεξεργασία ή διαγραφή.



ΚΕΦΑΛΑΙΟ 6:ΣΥΜΠΕΡΑΣΜΑΤΑ

6.1 ΔΥΣΚΟΛΙΕΣ

Οι δυσκολίες που αντιμετώπισα στην ανάπτυξη της εφαρμογής είναι:

- Έλλειψη πηγών/παραδειγμάτων για την κατανόηση και την σωστή χρήση του App Inventor
- Το App Inventor μου φάνηκε δύσχρηστο σε μερικά κομμάτια και με κάποιες ενοχλητικές ατέλειες(bugs) όπως κλείσιμο και άνοιγμα της εφαρμογής κάθε φορά που γινόταν αλλαγή στον κώδικα
- Επίσης υπήρχαν προβλήματα με την διαχείριση και το συντονισμό των ημερομηνιών μεταξύ του App Inventor και του fusion tables
- Τέλος η ανάλυση των απαιτήσεων ήταν δύσκολη καθώς έπρεπε να λάβω υπόψη μου και τις οργανωτικές ανάγκες των επιχειρήσεων αλλά και τις απλές καθημερινές απαιτήσεις ενός απλό χρήστη

6.2 ΑΔΥΝΑΤΑ ΣΗΜΕΙΑ

Το App Inventor παρέχει συγκεκριμένες δυνατότητες στο σχεδιαστή της εφαρμογής έναντι άλλων γλωσσών προγραμματισμού με τα modules που διαθέτει πράγμα που οδηγεί στα παρακάτω αδύνατα σημεία

- Ο περιορισμένος όγκος δεδομένων που μπορεί να διαχειριστεί η εφαρμογή
- Η αδυναμία καταχώρησης ηχητικών σημειώσεων
- Δεν είναι δυνατή η κοινή χρήση
- Η καταχώρηση μίας εικόνας ανά σημείωση
- Δεν είναι δυνατή η χρήση pen για χειρόγραφες σημειώσεις
- Αδυναμία διαμόρφωσης του κειμένου

6.3 ΑΞΙΟΛΟΓΗΣΗ

Στην εν λόγω πτυχιακή εργασία περιγράφεται αναλυτικά η διαδικασία κατασκευής της εφαρμογής στο App Inventor και αυτό μπορεί να βοηθήσει τους αναγνώστες στη δημιουργία αντίστοιχων εφαρμογών καθώς και στην καλύτερη κατανόησή της.
Συμπερασματικά πρέπει να αναφερθεί ότι κατά τη διάρκεια υλοποίησης της συγκεκριμένης εφαρμογής χρησιμοποίησα με τον καλύτερο δυνατό τρόπο τις δυνατότητες του MIT App Inventor παρόλο τους περιορισμούς της πλατφόρμας. Φτάνοντας λοιπόν στο τέλος της πτυχιακής εργασίας μπορώ να πω με βεβαιότητα ότι το περιβάλλον του App Inventor μας επιτρέπει τη δημιουργία λειτουργικών και πρακτικών εφαρμογών. Τέλος η εφαρμογή που υλοποιήθηκε μπορεί να χαρακτηριστεί ως ένα ικανοποιητικό πρόγραμμα διευκόλυνσης των επιχειρηματικών διαδικασιών που συμβαδίζει με τη συνεχή εξέλιξη στον τεχνολογικό κλάδο και αποτελεί σημαντικό βοήθημα για μία επιχείρηση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Massachusetts Institute of Technology (2015), **App of the month winners**, διαθέσιμο στην ιστοσελίδα http://appinventor.mit.edu/explore/app-month-gallery.html, ημερομηνία πρόσβασης 15/03/2017

2. Massachusetts Institute of Technology (2015), **Explore about us**, διαθέσιμο στην ιστοσελίδα http://appinventor.mit.edu/explore/about-us.html , ημερομηνία πρόσβασης 15/03/2017

3. Massachusetts Institute of Technology (2015), **Tutorials for App Inventor**, Διαθέσιμο στην ιστοσελίδα http://appinventor.mit.edu/explore/ai2/tutorials.html, ημερομηνία πρόσβασης 15/03/2017

4. Massachusetts Institute of Technology (2015), **Pizza Party with Fusion Tables for App Inventor 2**, διαθέσιμο στην ιστοσελίδα http://appinventor.mit.edu/explore/ai2/pizzaparty.html, ημερομηνία πρόσβασης 15/03/2017

5. Edward M (2014), **Display "Warning" and "Alert" box messages in App Inventor apps**, διαθέσιμο στην ιστοσελίδα http://appinventor.pevest.com/?p=81, ημερομηνία πρόσβασης 15/03/2017

6. Βασίλης Ορφανάκης (2017), **App Inventor**, διαθέσιμο στην ιστοσελίδα http://users.sch.gr/vorfan/index.php/app-inventor, ημερομηνία πρόσβασης 15/03/2017

7. Wikipedia (2017), **App Inventor for Android**, διαθέσιμο στην ιστοσελίδα https://en.wikipedia.org/wiki/App_Inventor_for_Android, ημερομηνία πρόσβασης 15/03/2017

8. Pura Vida Apps (2016), **App Inventor Tutorials and Advanced Examples**, διαθέσιμο στην ιστοσελίδα https://puravidaapps.com/tutorials.php, ημερομηνία πρόσβασης 15/03/2017

9. Marcus L Endicott (2017), **100 Best App Inventor Videos**, διαθέσιμο στην ιστοσελίδα http://meta-guide.com/videography/100-best-appinventor-videos, ημερομηνία πρόσβασης 15/03/2017

10. Sajal Dutta (2013), **Advanced Components On App Inventor**, διαθέσιμο στην ιστοσελίδα http://www.imagnity.com/tutorials/app-inventor/advanced-components-on-app-inventor/, ημερομηνία πρόσβασης 15/03/2017

11. Google Developers (2016), **Fusion Tables REST API,** διαθέσιμο στην ιστοσελίδα https://developers.google.com/fusiontables/docs/v2/getting_started, ημερομηνία πρόσβασης 15/03/2017

12. Shaileen Crawford Pokress, José Juan Dominguez Veiga (2013), **MIT App Inventor: Enabling Personal Mobile Computing**, διαθέσιμο στις ιστοσελίδες http://appinventor.mit.edu/explore/resources/personal_mobile_computing.html
http://arxiv.org/abs/1310.2830, ημερομηνία πρόσβασης 15/03/2017

13. Yu-Chang Hsu, Yu-Hui Ching, (2013), **Mobile app design for teaching and learning: Educators experience in an online graduate course**, διαθέσιμο στην ιστοσελίδα http://appinventor.mit.edu/explore/resources/hsu-ching-2013.html, ημερομηνία πρόσβασης 15/03/2017

14. D. Wolder, H. Abelson, E. Spertus, L. Looney (2011), **App Inventor**, διαθέσιμο στην ιστοσελίδα https://books.google.gr/books?hl=el&lr=&id=WcMBVlXltSsC&oi=fnd&pg=PR4&dq=app+inventor&ots=Wn_AbQW2AO&sig=4kclStEPI15L88IYKM0S4HGz01g&redir_esc=y#v=onepage&q=app%20inventor&f=false, ημερομηνία πρόσβασης 15/03/2017