

Εξώφυλλο Αναφοράς Πτυχιακής Εργασίας
Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Μηχανικών Πληροφορικής



Πτυχιακή Εργασία

**Τίτλος: Εφαρμογή για καθοδήγηση χρηστών στις καθημερινές
τους δραστηριότητες σε περιβάλλον Android**

Γκεντζογλάνης Κωνσταντίνος (3056)

Επιβλέπων καθηγητής : **Παπαδάκης Νικόλαος**

Επιτροπή Αξιολόγησης : **Παπαδάκης Νικόλαος, Ρουσάκης, Τσάμης**

Ημερομηνία παρουσίασης : **19/11/2014**

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή Δρ. Νικόλαο Παπαδάκη για την πρόταση και ανάθεση αυτής της πτυχιακής σε μένα. Εν συνεχεία, θα ήθελα να ευχαριστήσω την οικογένεια μου για την κατανόηση και υποστήριξη που μου έδειξαν κατά την περίοδο της δημιουργίας της εφαρμογής αυτής αλλά και για την συγγραφή αυτής της πτυχιακής εργασίας.

Abstract

The objective of this thesis is to develop an application that is compatible with the latest versions of the operating system Android. This application has been developed focusing to the simplicity of the user interface, so that anybody can use this application without any specific knowledge.

This application helps the user in his daily activities. Such an activity is the management of appointments, the user is able to enter, modify and delete his appointments. It also helps the user in finding the nearest gas station / supermarket available. To find the nearest gas station or supermarket, the application must consider what time the user wants to go to the gas station / supermarket, the store hours, the location of the user and the traffic. If the user has an appointment at the time, the application finds the nearest gas station / supermarket based in the appointment's time and location.

For the purposes of this thesis, we used existent gas stations and supermarkets of Heraklion city in Crete, Greece, but we didn't implement all gas stations and supermarkets of the city, since the purpose of the thesis is to show how such an application works.

Σύνοψη

Ο στόχος αυτής της πτυχιακής εργασίας είναι η ανάπτυξη μίας εφαρμογής η οποία θα είναι συμβατή με τις πιο πρόσφατες εκδόσεις του λειτουργικού συστήματος Android. Η εφαρμογή αυτή έχει αναπτυχθεί με ιδιαίτερη προσοχή στην απλότητα της Διεπαφής Χρήστη (User Interface), έτσι ώστε οποιοσδήποτε να μπορεί να χρησιμοποιήσει την εφαρμογή αυτή χωρίς να έχει ιδιαίτερες γνώσεις.

Η εφαρμογή αυτή βοηθάει τον χρήστη στις καθημερινές του δραστηριότητες. Μία τέτοια δραστηριότητα είναι η διαχείριση των ραντεβού του. Πιο συγκεκριμένα, ο χρήστης έχει τη δυνατότητα να εισάγει, να τροποποιήσει και να διαγράψει τα ραντεβού του. Επίσης, βοηθάει τον χρήστη στην εύρεση του πλησιέστερου διαθέσιμου βενζινάδικου και σούπερ μάρκετ. Η εύρεση του πλησιέστερου βενζινάδικου ή σούπερ μάρκετ πραγματοποιείται με βάση την ώρα που ο χρήστης θέλει να πάει στο βενζινάδικο – σούπερ μάρκετ, τις ώρες καταστημάτων, την τοποθεσία του χρήστη και την κίνηση στους δρόμους. Ακόμα, αν υπάρχει κάποιο ραντεβού, η εφαρμογή βρίσκει το πλησιέστερο βενζινάδικο – σούπερ μάρκετ με βάση την ώρα του ραντεβού και την τοποθεσία του.

Για τις ανάγκες της συγκεκριμένης πτυχιακής εργασίας χρησιμοποιήθηκαν υπαρκτά βενζινάδικα και σούπερ μάρκετ της πόλης του Ηρακλείου Κρήτης, όμως δεν τοποθετήθηκαν όλα τα βενζινάδικα και τα σούπερ μάρκετ της πόλης στην εφαρμογή, αφού, ο σκοπός της πτυχιακής εργασίας είναι να δείξουμε πως μία τέτοια εφαρμογή δουλεύει.

Πίνακας Περιεχομένων

1	Εισαγωγή.....	11
1.1	Περίληψη.....	11
1.2	Κίνητρο για την Διεξαγωγή της Εργασίας.....	11
1.3	Σκοπός και Στόχοι Εργασίας.....	11
1.4	Δομή Εργασίας.....	12
2	Μεθοδολογία Υλοποίησης.....	13
2.1	Μέθοδος Ανάλυσης & Ανάπτυξης Πτυχιακής.....	13
2.1.1	Θεωρίες.....	13
2.1.1.1	Eclipse ADT with Android SDK.....	13
2.1.1.2	Android Studio.....	14
2.1.1.3	MIT App Inventor.....	14
2.1.1.4	Intel XDK.....	14
2.1.1.5	PhoneGap.....	14
2.1.1.6	Xamarin.....	15
2.1.1.7	PHP.....	15
2.1.1.8	SQL.....	15
2.1.1.9	XML.....	15
2.1.2	Αλγόριθμοι και Πολυπλοκότητα.....	15
2.1.2.1	Login Activity.....	17
2.1.2.2	Register Activity.....	19
2.1.2.3	My Account Activity.....	21
2.1.2.4	Appointment Activity.....	23
2.1.2.5	Set Appointment Activity.....	25
2.1.2.6	Station Activity.....	29
2.1.2.7	Map Activity.....	32
3	Σχέδιο Δράσης για την εκπόνηση της εργασίας.....	40
3.1	State of the Art.....	40
3.1.1	Android.....	40
4	Κύριο μέρος Πτυχιακής.....	43
4.1	Ανάλυση Προβλήματος.....	43
4.1.1	Απαιτήσεις Συστήματος.....	43
4.2	Σχεδιασμός Υλοποίησης.....	44
4.3	Αρχιτεκτονική.....	45

4.4	Υλοποίηση	46
4.4.1	Προετοιμασία Ηλεκτρονικού Υπολογιστή.....	46
4.4.2	Κατοχύρωση Ονόματος Τομέα (Domain Name) και Φιλοξενίας Χώρου (Hosting)	53
4.4.3	Η Εφαρμογή	60
5	Αποτελέσματα.....	86
5.1	Συμπεράσματα.....	86
5.2	Μελλοντική Εργασία και Επεκτάσεις.....	86
	Βιβλιογραφία	87

Πίνακας Εικόνων

Εικόνα 1 – Αρχιτεκτονική Εφαρμογής.....	45
Εικόνα 2 – Εγκατάσταση του JDK No.1.....	47
Εικόνα 3 – Εγκατάσταση του JDK No.2.....	48
Εικόνα 4– Εγκατάσταση του JDK No.3.....	49
Εικόνα 5 – Εγκατάσταση του JDK No.4.....	49
Εικόνα 6 – Εγκατάσταση του JDK No.5.....	50
Εικόνα 7 – Εγκατάσταση του JDK No.6.....	50
Εικόνα 8 – Εγκατάσταση του JDK No.7.....	51
Εικόνα 9 – Εγκατάσταση του JDK No.8.....	51
Εικόνα 10 – Εγκατάσταση του Eclipse No.1	52
Εικόνα 11 – Εγκατάσταση του Eclipse No.2	52
Εικόνα 12 – Κατοχύρωση Ονόματος Τομέα (Domain Name) No.1	53
Εικόνα 13 – Κατοχύρωση Ονόματος Τομέα (Domain Name) No.2	54
Εικόνα 14 – Κατοχύρωση Ονόματος Τομέα (Domain Name) No.3	55
Εικόνα 15 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.1	55
Εικόνα 16 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.2.....	56
Εικόνα 17 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.3	56
Εικόνα 18 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.4.....	57
Εικόνα 19 – Μεταφόρτωση Αρχείων No.1	58
Εικόνα 20 – Μεταφόρτωση Αρχείων No.2	58
Εικόνα 21 – Δημιουργία Βάσης Δεδομένων No.1	59
Εικόνα 22 – Δημιουργία Βάσης Δεδομένων No.2	59
Εικόνα 23 – Πρόσβαση στη Βάση Δεδομένων No.1	60
Εικόνα 24 – Πρόσβαση στη Βάση Δεδομένων No.2	60
Εικόνα 25 – Πρόσβαση στη Βάση Δεδομένων No.3	60
Εικόνα 26 – Αρχική Δραστηριότητα Εφαρμογής No.1.....	61
Εικόνα 27 – Δραστηριότητα Σύνδεσης No.1	61
Εικόνα 28 – Δραστηριότητα Εγγραφής No.1.....	62
Εικόνα 29 – Δραστηριότητα Εγγραφής No.2.....	63
Εικόνα 30 – Δραστηριότητα Εγγραφής No.3.....	63
Εικόνα 31 – Αρχική Δραστηριότητα Εφαρμογής No.2.....	64
Εικόνα 32 – Δραστηριότητα Σύνδεσης No.2	65
Εικόνα 33 – Δραστηριότητα Σύνδεσης No.3	65
Εικόνα 34 – Δραστηριότητα Σύνδεσης No.4	66
Εικόνα 35 – Δραστηριότητα Σύνδεσης No.5	67
Εικόνα 36 – Δραστηριότητα Λογαριασμού No.1.....	67
Εικόνα 37 – Δραστηριότητα Λογαριασμού No.2.....	68

Εικόνα 38 – Δραστηριότητα Λογαριασμού Νο.3.....	68
Εικόνα 39 – Δραστηριότητα Λογαριασμού Νο.4.....	69
Εικόνα 40 – Δραστηριότητα Λογαριασμού Νο.5.....	69
Εικόνα 41 – Δραστηριότητα Λογαριασμού Νο.6.....	70
Εικόνα 42 – Δραστηριότητα Λογαριασμού Νο.7.....	71
Εικόνα 43 – Δραστηριότητα Λογαριασμού Νο.8.....	71
Εικόνα 44 – Δραστηριότητα Ραντεβού Νο.1	72
Εικόνα 45 – Δραστηριότητα Ραντεβού Νο.2	72
Εικόνα 46 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.1	73
Εικόνα 47 – Δραστηριότητα Ραντεβού Νο.3	73
Εικόνα 48 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.2	74
Εικόνα 49 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.3	75
Εικόνα 50 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.4	75
Εικόνα 51 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.5	76
Εικόνα 52 – Δραστηριότητα Σταθμού Νο.1.....	77
Εικόνα 53 – Δραστηριότητα Σταθμού Νο.2.....	77
Εικόνα 54 – Δραστηριότητα Σταθμού Νο.3.....	78
Εικόνα 55 – Δραστηριότητα Χάρτη Νο.1	79
Εικόνα 56 – Δραστηριότητα Χάρτη Νο.2.....	79
Εικόνα 57 – Δραστηριότητα Σταθμού Νο.4.....	80
Εικόνα 58 – Δραστηριότητα Χάρτη Νο.3	81
Εικόνα 59 – Δραστηριότητα Χάρτη Νο.4	81
Εικόνα 60 – Δραστηριότητα Χάρτη Νο.5	82
Εικόνα 61 – Δραστηριότητα Χάρτη Νο.6	82
Εικόνα 62 – Δραστηριότητα Χάρτη Νο.7.....	83
Εικόνα 63 – Δραστηριότητα Χάρτη Νο.8.....	83
Εικόνα 64 – Δραστηριότητα Χάρτη Νο.9.....	84
Εικόνα 65 – Δραστηριότητα Χάρτη Νο.10.....	84
Εικόνα 66 – Δραστηριότητα Χάρτη Νο.11	85

Λίστα Πινάκων

Πίνακας 1 – Σχετικές Μεθοδολογίες Υλοποίησης.....	13
Πίνακας 2 – Απαιτήσεις Συστήματος Ηλεκτρονικών Υπολογιστών	44
Πίνακας 3 – Απαιτήσεις Συστήματος Έξυπνου Κινητού	44

Πίνακας Γραφημάτων

Γράφημα 1 – Πληροφορίες εκδόσεων Android που χρησιμοποιούνται.....	41
--	----

1 Εισαγωγή

Η συγκεκριμένη πτυχιακή εργασία αφορά την ανάπτυξη και υλοποίηση μιας εφαρμογής σε περιβάλλον Android. Αυτή η εφαρμογή έχει ως σκοπό την διευκόλυνση του χρήστη στις καθημερινές του δραστηριότητες. Οι δραστηριότητες που περιλαμβάνονται στην εφαρμογή αυτή είναι η εύρεση βενζινάδικων, σούπερ μάρκετ και η διαχείριση των ραντεβού του χρήστη.

1.1 Περίληψη

Ο σκοπός της πτυχιακής αυτής είναι να διευκολύνει τον χρήστη στην επιλογή που έχει να κάνει αν αυτός θελήσει να βάλει βενζίνη στο όχημά του ή να πραγματοποιήσει τα ψώνια της βδομάδας. Όλα αυτά πραγματοποιούνται σε ένα πλαίσιο όπου τα ραντεβού του χρήστη έχουν τον πρώτο ρόλο και όλες οι λειτουργίες της εφαρμογής αυτής συνεργάζονται αρμονικά έτσι ώστε ο χρήστης να έχει μία ολοκληρωμένη εμπειρία χρησιμοποιώντας την εφαρμογή αυτή.

Η εφαρμογή αυτή έχει υλοποιηθεί σε περιβάλλον Android, όμως χρησιμοποιήθηκαν και άλλες τεχνολογίες για την ολοκλήρωσή της. Πιο συγκεκριμένα χρησιμοποιήθηκαν οι PHP, SQL και XML τεχνολογίες. Αρχικά, μία Βάση Δεδομένων ήταν απαραίτητη για να υπάρχει η δυνατότητα να αποθηκεύονται οι χρήστες. Ο χρήστης έχει την δυνατότητα να εγγραφεί στην εφαρμογή και μετέπειτα να κάνει είσοδο σε αυτήν, αν και η εγγραφή και η είσοδος του χρήστη στην εφαρμογή δεν είναι απαραίτητη για να λειτουργήσει οποιοσδήποτε χρήστης την εφαρμογή αυτή. Στην διαδικασία δημιουργίας της Βάσης Δεδομένων χρειάστηκαν οι PHP και SQL τεχνολογίες καθώς και ένας Server ο οποίος φιλοξενεί την Βάση Δεδομένων. Επίσης, ο Server φιλοξενεί και XML αρχεία τα οποία είναι απαραίτητα για να λειτουργήσουν οι υπόλοιπες λειτουργίες της εφαρμογής, όπως τα βενζινάδικα και τα σούπερ μάρκετ. Δηλαδή, στα αρχεία XML υπάρχουν αποθηκευμένες διάφορες πληροφορίες για τα βενζινάδικα και τα σούπερ μάρκετ οι οποίες είναι απαραίτητες για την εύρεσή τους.

Ακόμα, ο χρήστης έχει την δυνατότητα να εισάγει, να τροποποιήσει αλλά και να διαγράψει τα ραντεβού του. Τα ραντεβού έχουν τον μεγαλύτερο ρόλο στην εφαρμογή αυτή καθώς οι υπόλοιπες λειτουργίες της εφαρμογής λειτουργούν με βάση αυτά. Πιο συγκεκριμένα αν ο χρήστης επιλέξει να πάει σε ένα βενζινάδικο ή σε ένα σούπερ μάρκετ, η επιλογή για στο πιο συγκεκριμένο σούπερ μάρκετ ή βενζινάδικο θα πάει γίνεται με βάση το ραντεβού που έχει την ώρα που ο χρήστης θα κάνει την επιλογή αυτή.

1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Στην εποχή που ζούμε, όπου τα έξυπνα κινητά τηλέφωνα και οι εφαρμογές τους έχουν κατακλίσει τον κόσμο, υπάρχουν διάφορες εφαρμογές οι οποίες βοηθούν τον χρήστη να διαχειριστεί τα ραντεβού του, να βρει ένα κοντινό βενζινάδικο ή σούπερ μάρκετ. Όμως, είναι δύσκολο να βρεθεί μία εφαρμογή η οποία θα ασχολείται με τα ραντεβού του κάθε χρήστη καθώς και την ανάγκη που έχει να επισκεφθεί ένα βενζινάδικο ή ένα σούπερ μάρκετ, όπου όλα αυτά να λειτουργούν σε μία εφαρμογή και όχι σε συνδυασμό πολλών και διαφορετικών εφαρμογών. Η παρούσα πτυχιακή εργασία στοχεύει στην δημιουργία αυτής της εφαρμογής, έτσι ώστε ο χρήστης να μην χρειάζεται να εναλλάσσει πολλές και διάφορες εφαρμογές στο κινητό του για να βγάλει εις πέρας τις καθημερινές του δραστηριότητες.

1.3 Σκοπός και Στόχοι Εργασίας

Οι στόχοι της πτυχιακής εργασίας είναι τέσσερις και περιγράφονται σύντομα παρακάτω.

1. Αρχικά, ο πρώτος στόχος είναι η δυνατότητα που δίνεται στον χρήστη να δημιουργήσει ένα λογαριασμό στην εφαρμογή, να πραγματοποιεί είσοδο στον λογαριασμό του από την εφαρμογή καθώς και να τον διαχειρίζεται.

2. Ο χρήστης να έχει την δυνατότητα να εισάγει ένα ραντεβού του στην εφαρμογή, να μπορεί να το τροποποιήσει αλλά και να το διαγράψει.
3. Ο χρήστης να έχει την δυνατότητα να ζητήσει από την εφαρμογή να του βρει το πλησιέστερο βενζινάδικο και η εφαρμογή ανάλογα με τις υπόλοιπες υποχρεώσεις του χρήστη καθώς και την τοποθεσία του να του βρίσκει το κατάλληλο βενζινάδικο.
4. Τέλος, να δίνεται η δυνατότητα στον χρήστη να ζητάει από την εφαρμογή να του βρει το πλησιέστερο σούπερ μάρκετ και η εφαρμογή ανάλογα με τις υπόλοιπες υποχρεώσεις του χρήστη καθώς και την τοποθεσία του να του βρίσκει το κατάλληλο σούπερ μάρκετ.

Σύμφωνα με τους παραπάνω στόχους καταλήγουμε στον πρώτο σκοπό αυτής της πτυχιακής εργασίας και δεν είναι άλλος από την δημιουργία μίας ενιαίας εφαρμογής η οποία θα ασχολείται με τις καθημερινές δραστηριότητες του χρήστη. Επίσης, η εφαρμογή πρέπει να είναι εύκολα κατανοητή από τον χρήστη. Τέλος, η διεπαφή χρήστη (User Interface) της εφαρμογής πρέπει να είναι απλή και να ακολουθεί τα πρότυπα τα οποία έχει θέσει η Google στο λογισμικό της Android.

1.4 Δομή Εργασίας

Τα κεφάλαια που ακολουθούν ασχολούνται με διάφορα θέματα αυτής της πτυχιακής εργασίας. Το Κεφάλαιο 2, **Μεθοδολογία Υλοποίησης**, ασχολείται με τις μεθόδους που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής καθώς και θεωρίες στις οποίες στηρίζεται η εφαρμογή αυτή. Επίσης, αναλύονται και οι αλγόριθμοι που χρησιμοποιήθηκαν και για ποιο λόγο. Το Κεφάλαιο 3, **Σχέδιο Δράσης για την εκπόνηση της εργασίας**, έχει θέμα την τεχνολογία αιχμής (state of the art) και την βιβλιογραφική αναζήτηση που έγινε πάνω σε αυτήν. Το Κεφάλαιο 4, **Κύριο μέρος Πτυχιακής**, έχει ως θέμα το κύριο μέρος της πτυχιακής εργασίας όπως την ανάλυση του προβλήματος, σχεδίαση της λύσης του προβλήματος και την υλοποίηση της λύσης του προβλήματος. Τέλος, το Κεφάλαιο 5, **Αποτελέσματα**, ασχολείται με τα αποτελέσματα που προκύπτουν από την πραγματοποίηση αυτής της πτυχιακής εργασίας.

2 Μεθοδολογία Υλοποίησης

Στο κεφάλαιο αυτό αναλύονται οι μέθοδοι που είναι διαθέσιμοι για την δημιουργία εφαρμογών Android και ποια χρησιμοποιήθηκε για την δημιουργία της εφαρμογής αυτής, οι θεωρίες και οι τεχνολογίες στις οποίες στηρίζεται η εφαρμογή, οι αλγόριθμοι που χρησιμοποιήθηκαν και για ποιο λόγο.

2.1 Μέθοδος Ανάλυσης & Ανάπτυξης Πτυχιακής

Το πρόβλημα που αντιμετωπίζουμε είναι η μη ύπαρξη μίας ενιαίας εφαρμογής όπου ο χρήστης θα έχει το δικό του προσωπικό λογαριασμό και την δυνατότητα να διαχειρίζεται τα ραντεβού του. Επίσης, η εφαρμογή θα βρίσκει το κοντινότερο βενζινάδικο και σούπερ μάρκετ στην τοποθεσία του χρήστη. Για να αντιμετωπιστεί το πρόβλημα αυτό, αρχικά πρέπει να πραγματοποιηθεί η επιλογή της μεθοδολογίας για την δημιουργία μιας εφαρμογής Android.

Υπάρχουν πολλοί και διάφοροι μέθοδοι για να δημιουργήσει κανείς εφαρμογές για το λειτουργικό σύστημα Android με την χρήση ενός Η/Υ. Η πρώτη μέθοδος, η οποία χρησιμοποιήθηκε στη δημιουργία της Android εφαρμογής στη συγκεκριμένη πτυχιακή εργασία είναι το Eclipse ADT (Android Development Tool) μαζί με το Android SDK (Software Development Kit) για περιβάλλον Windows. Όμως, αν κάποιος χρήστης έχει ήδη εγκατεστημένο το Eclipse IDE (Integrated Development Environment “Ολοκληρωμένο Περιβάλλον Ανάπτυξης”) ή θέλει να χρησιμοποιήσει ένα διαφορετικό IDE μπορεί να κατεβάσει από την επίσημη ιστοσελίδα της Google για το Android μία αυτόνομη έκδοση του Android SDK Tools και να την εγκαταστήσει στον IDE που προτιμάει.

Μία άλλη μέθοδος για την δημιουργία Android εφαρμογών είναι το Android Studio. Η τρίτη μέθοδος είναι το MIT App Inventor και συνεχίζουμε με τις μεθόδους Intel XDK, PhoneGap και Xamarin. Όλες αυτές οι μέθοδοι φαίνονται στον πίνακα 1 και θα αναλυθούν παρακάτω.

ΜΕΘΟΔΟΛΟΓΙΑ	ΑΝΑΦΟΡΑ ΣΤΗΝ ΒΙΒΛΙΟΓΡΑΦΙΑ
Eclipse ADT with Android SDK	http://developer.android.com/sdk/index.html , http://en.wikipedia.org/wiki/Eclipse_(software)
Android Studio	http://developer.android.com/sdk/installing/studio.html , http://en.wikipedia.org/wiki/Android_Studio
MIT App Inventor	http://appinventor.mit.edu/explore/ , http://en.wikipedia.org/wiki/App_Inventor_for_Android
Intel XDK	https://software.intel.com/en-us/html5/tools
PhoneGap	http://phonegap.com/ , http://en.wikipedia.org/wiki/PhoneGap
Xamarin	http://xamarin.com/ , http://en.wikipedia.org/wiki/Xamarin

Πίνακας 1 – Σχετικές Μεθοδολογίες Υλοποίησης

2.1.1 Θεωρίες

Παρακάτω γίνεται αναφορά για τις μεθόδους που μπορούν να χρησιμοποιηθούν για την ανάπτυξη εφαρμογών Android, πια μέθοδος χρησιμοποιήθηκε για την συγκεκριμένη πτυχιακή εργασία και για πιο λόγο. Επίσης, αναφέρονται θεωρητικά και οι τεχνολογίες που χρησιμοποιήθηκαν για την δημιουργία της εφαρμογής αυτής.

2.1.1.1 Eclipse ADT with Android SDK

Το Eclipse ADT with Android SDK μας επιτρέπει να γράφουμε κώδικα για την δημιουργία εφαρμογών χρησιμοποιώντας την Java ως γλώσσα προγραμματισμού. Έπειτα, μπορούμε να

εγκαταστήσουμε την εφαρμογή αυτή σε μία συσκευή με λειτουργικό σύστημα Android ή σε ένα προσομοιωτή του λειτουργικού συστήματος Android, ο οποίος προσομοιωτής μας παρέχεται από το Eclipse ADT with Android SDK, για να ελέγξουμε αν η εφαρμογή λειτουργεί σωστά όταν την χρησιμοποιεί ο χρήστης.

Επιλέξαμε να δημιουργήσουμε την εφαρμογή μας με τη χρήση αυτής της μεθόδου επειδή το Android SDK συνεργάζεται πολύ καλά με το Eclipse IDE και η μέθοδος αυτή είναι κατάλληλη για περιβάλλοντα παραγωγής. Επίσης, το Eclipse ADT with Android SDK χρησιμοποιείται ευρέως από διάφορους προγραμματιστές που θέλουν να δημιουργήσουν εφαρμογές για το οικοσύστημα του Android και μπορούν να βρεθούν εύκολα λύσεις για τυχόν προβλήματα που μπορεί να παρουσιαστούν.^{(1), (2)}

2.1.1.2 Android Studio

Το Android Studio είναι ένα IDE παρόμοιο με το Eclipse ADT with Android SDK το οποίο όμως παρέχει νέα χαρακτηριστικά και βελτιώσεις έναντι του Eclipse ADT και θα είναι το επίσημο Android IDE μόλις ολοκληρωθεί η ανάπτυξή του. Επειδή το IDE αυτό είναι ακόμα σε δοκιμαστικό στάδιο επιλέξαμε το Eclipse ADT with Android SDK έναντι του Android Studio, γιατί το Android Studio είναι ακόμα ένα ημιτελές προϊόν.^{(1), (2)}

2.1.1.3 MIT App Inventor

Το MIT App Inventor είναι μία διαδικτυακή εφαρμογή ανοιχτού κώδικα που δημιουργήθηκε από την Google αλλά τώρα συντηρείται από το Ινστιτούτο Τεχνολογίας της Μασαχουσέτης (MIT). Αποτελείται από ένα περιβάλλον μεταφοράς στοιχείων (drag-and-drop) που μπορεί να χρησιμοποιηθεί για την δημιουργία νέων εφαρμογών που αποτελούνται από τμήματα, τα οποία τμήματα αποτελούνται από κώδικα, και ενώνονται το ένα με το άλλο. Με τη μέθοδο αυτή οποιοσδήποτε μπορεί να δημιουργήσει εφαρμογές Android και ας μην είναι προγραμματιστής.^{(2), (3)}

2.1.1.4 Intel XDK

Το Intel XDK έχει δημιουργηθεί από την Intel και είναι ένα εργαλείο ανάπτυξης εφαρμογών για έξυπνες συσκευές, το οποίο μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα (Windows, OS X, Ubuntu Linux). Το Intel XDK βασίζεται πάνω σε τεχνολογίες Ιστού (Web), όπως τις γλώσσες HTML, CSS και JavaScript, για την δημιουργία εφαρμογών έξυπνων συσκευών ανεξαρτήτως λειτουργικού συστήματος. Οι εφαρμογές αυτές μπορούν να εγκατασταθούν σε συσκευές με λειτουργικά συστήματα Android, iOS, Windows 8 και Windows Phone 8 χωρίς ιδιαίτερες τροποποιήσεις στον πηγαίο κώδικα της κάθε εφαρμογής.^{(2), (4)}

2.1.1.5 PhoneGap

Το PhoneGap ένα πλαίσιο (framework) ανάπτυξης εφαρμογών έξυπνων κινητών που δημιουργήθηκε από την εταιρία Nitobi και αγοράστηκε από την Adobe Systems το 2011. Επιτρέπει στους προγραμματιστές λογισμικού να δημιουργήσουν εφαρμογές για έξυπνες συσκευές χρησιμοποιώντας την JavaScript, την HTML5 και την CSS3. Το PhoneGap επεκτείνει τις δυνατότητες της HTML και JavaScript με σκοπό να λειτουργήσουν με τις έξυπνες συσκευές. Οι εφαρμογές που προκύπτουν από το PhoneGap είναι υβριδικές. Δηλαδή, οι εφαρμογές αυτές έχουν πρόσβαση στα APIs της συσκευής αλλά χρησιμοποιούνται οι τεχνολογίες Ιστού (Web), (HTML5, CSS3, JavaScript), για την δημιουργία των εφαρμογών. Τα API (Application Programming Interface), στα Ελληνικά Διεπαφή Προγραμματισμού Εφαρμογών, είναι η διεπαφή των προγραμματιστικών διαδικασιών που ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή παρέχει προκειμένου να επιτρέπει να γίνονται προς αυτό αιτήσεις από άλλα προγράμματα ή / και ανταλλαγή δεδομένων.^{(2), (5)}

2.1.1.6 Xamarin

Η Xamarin είναι μία εταιρία λογισμικού που δημιουργήθηκε τον Μάιο του 2011. Με την πλατφόρμα της Xamarin, οι προγραμματιστές μπορούν να δημιουργήσουν εφαρμογές για έξυπνες συσκευές χρησιμοποιώντας την γλώσσα προγραμματισμού C#. Αυτές οι εφαρμογές είναι συμβατές με τα λειτουργικά συστήματα έξυπνων συσκευών Android, iOS και Windows. Μία εφαρμογή που έχει δημιουργηθεί με την πλατφόρμα της Xamarin χρησιμοποιεί τον ίδιο κώδικα στα διαφορετικά λειτουργικά συστήματα έξυπνων συσκευών, ταυτόχρονα όμως χρησιμοποιείται η Διεπαφή Χρήστη (User Interface) του κάθε λειτουργικού συστήματος.^{(2), (6)}

2.1.1.7 PHP

Η PHP (PHP: Hypertext Preprocessor) είναι μία γλώσσα η οποία είναι γραμμένη σε αρχεία (scripts) και εκτελείται στο Διακομιστή (Server). Είναι σχεδιασμένη για την ανάπτυξη εφαρμογών Ιστού (Web) η οποία μπορεί να ενσωματωθεί εύκολα στην HTML. Επίσης, χρησιμοποιείται πολύ συχνά και ως γλώσσα προγραμματισμού γενικής χρήσης.⁽²⁾

2.1.1.8 SQL

Η SQL (Structured Query Language) είναι μία γλώσσα προγραμματισμού ειδικού σκοπού που έχει σχεδιαστεί για τη διαχείριση των δεδομένων που τηρούνται σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System).⁽²⁾

2.1.1.9 XML

Η XML (Extensible Markup Language) είναι μια γλώσσα σήμανσης που καθορίζει ένα σύνολο κανόνων για την κωδικοποίηση των εγγράφων σε μια μορφή που είναι αναγνώσιμη τόσο από τον άνθρωπο όσο και από τους Η/Υ.⁽²⁾

2.1.2 Αλγόριθμοι και Πολυπλοκότητα

Στη διαδικασία υλοποίησης της πτυχιακής εργασίας χρησιμοποιήθηκαν διάφοροι αλγόριθμοι για την πραγματοποίησή της. Τέτοιοι αλγόριθμοι, οι οποίοι θα αναλυθούν παρακάτω σε ψευδοκώδικα, εντοπίζονται μέσα στο πρόγραμμα στις εξής Δραστηριότητες (Μία Δραστηριότητα (Activity) είναι ένα στοιχείο της εφαρμογής που παρέχει μια οθόνη με την οποία οι χρήστες μπορούν να αλληλεπιδρούν με σκοπό να κάνουν κάτι, όπως να τραβήξουν μια φωτογραφία, να στείλουν ένα e-mail ή να δουν ένα χάρτη):

1. **Login Activity**, όπου πραγματοποιείται η διαδικασία είσοδος του χρήστη στην εφαρμογή.^{(1), (7), (8), (9), (10)}
2. **Register Activity**, όπου πραγματοποιείται η διαδικασία εγγραφής του χρήστη στην εφαρμογή.^{(1), (8), (9), (10)}
3. **My Account Activity**, εδώ ο χρήστης μπορεί να τροποποιήσει τα στοιχεία του λογαριασμού του.^{(1), (7), (8), (9), (10)}
4. **Appointment Activity**, σε αυτή τη διαδικασία ο χρήστης βλέπει όλα τα ραντεβού του.^{(1), (7) (11), (12), (13)}
5. **Set Appointment Activity**, σε αυτή τη διαδικασία ο χρήστης εισάγει τα ραντεβού του αλλά και τα τροποποιεί.^{(1), (7), (8), (12), (14), (15)}

6. **Station Activity**, εδώ ο χρήστης βλέπει όλα τα βενζινάδικα ή τα σούπερ μάρκετ της πόλης. ^{(1), (7), (8), (11), (12), (13), (16)}
7. **Map Activity**, εδώ ο χρήστης βλέπει την τοποθεσία του, όλα τα βενζινάδικα - σούπερ μάρκετ της πόλης ή τα κοντινότερα και τον δρόμο καθοδήγησης του χρήστη στο κοντινότερο βενζινάδικο – σούπερ μάρκετ. ^{(1), (7), (8), (9), (10), (17), (18)}

2.1.2.1 Login Activity

ΑΡΧΗ

```
AN (loginButton == pressed) ΤΟΤΕ
    cancel = false
    choice = "sign"
    AN (usernameText == null) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Username is required!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    AN (usernameText <= 4) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Username is too short!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    AN (passwordText == null) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Password is required!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    AN (passwordText <= 4) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Password is too short!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (forgotPasswordButton == pressed) ΤΟΤΕ
    cancel = false
    choice = "reset"
    AN (resetButton == pressed) ΤΟΤΕ
        AN (emailText == null) ΤΟΤΕ
            ΕΜΦΑΝΙΣΕ "E-mail is required!"
            cancel = true
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (cancel == false) ΤΟΤΕ
    Επικοινωνία με Βάση Δεδομένων
    AN (choice == "sign") ΤΟΤΕ
        ΔΙΑΒΑΣΕ databaseUsername, databasePassword
        AN (username <> databaseUsername) ΤΟΤΕ
            ΕΜΦΑΝΙΣΕ "Incorrect username!"
            Διακοπή διαδικασίας εισόδου
        ΑΛΛΙΩΣ AN (password <> databasePassword) ΤΟΤΕ
            ΕΜΦΑΝΙΣΕ "Incorrect password!"
            Διακοπή διαδικασίας εισόδου
        ΑΛΛΙΩΣ
            Είσοδος του χρήστη στην εφαρμογή
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ AN (choice == "reset") ΤΟΤΕ
        ΔΙΑΒΑΣΕ databaseEmail
        AN (email <> databaseEmail) ΤΟΤΕ
            ΕΜΦΑΝΙΣΕ "This e-mail does not exist!"
            Διακοπή διαδικασίας νέου κωδικού
```

ΑΛΛΙΩΣ

Δημιουργία νέου κωδικού για τον χρήστη με e-mail:
databaseEmail και αποστολή του νέου κωδικού του στο
e-mail του

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ

Ο αλγόριθμος αυτός ελέγχει τα στοιχεία που έχει εισάγει ο χρήστης για να συνδεθεί στο λογαριασμό του. Το πρώτο στάδιο ελέγχου πραγματοποιείται από την ίδια την εφαρμογή, τέτοιοι έλεγχοι αφορούν την συμπλήρωση των απαραίτητων πεδίων της φόρμας εισόδου του χρήστη στο λογαριασμό του.

Το δεύτερο στάδιο ελέγχου πραγματοποιείται από την εφαρμογή σε συνεργασία με τη Βάση Δεδομένων, δηλαδή η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και επαληθεύει τα στοιχεία που έχει δώσει ο χρήστης. Αν τα στοιχεία που έχει δώσει ο χρήστης στην εφαρμογή συμφωνούν με τα στοιχεία που έχει η Βάση Δεδομένων ο χρήστης εισέρχεται επιτυχώς στο λογαριασμό του, αλλιώς πρέπει να επαναλάβει τη διαδικασία αυτή από την αρχή.

Επίσης, ο αλγόριθμος αυτός βοηθάει τον χρήστη να δημιουργήσει ένα νέο κωδικό πρόσβασης στη περίπτωση που έχει ξεχάσει τον δικό του. Αρχικά, γίνεται έλεγχος από την εφαρμογή για την συμπλήρωση των απαραίτητων πεδίων και έπειτα επικοινωνεί η εφαρμογή με τη Βάση Δεδομένων για τον έλεγχο της εγκυρότητας των στοιχείων. Αν τα στοιχεία είναι έγκυρα δημιουργείται ένας νέος κωδικός πρόσβασης για τον χρήστη και του αποστέλλεται μέσω email.

Η πολυπλοκότητα αυτού του αλγορίθμου είναι $f(n) = 1$, εφόσον δεν υπάρχει κάποια διαδικασία επανάληψης και ο αριθμός των οδηγιών που χρειάζεται ο αλγόριθμος είναι απλά μια σταθερά.

2.1.2.2 Register Activity

APXH

```
AN (registerButton == pressed) TOTE
    cancel = false

AN (emailText == null) TOTE
    ΕΜΦΑΝΙΣΕ "E-mail is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (usernameText == null) TOTE
    ΕΜΦΑΝΙΣΕ "Username is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (usernameText < 4) TOTE
    ΕΜΦΑΝΙΣΕ "Username is too short!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (passwordText == null) TOTE
    ΕΜΦΑΝΙΣΕ "Password is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (passwordText < 4) TOTE
    ΕΜΦΑΝΙΣΕ "Password is too short!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (confirmPasswordText == null) TOTE
    ΕΜΦΑΝΙΣΕ "Confirm Password is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (passwordText <> confirmPasswordText) TOTE
    ΕΜΦΑΝΙΣΕ "Passwords don't match!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (firstNameText == null) TOTE
    ΕΜΦΑΝΙΣΕ "First name is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN (lastNameText == null) TOTE
    ΕΜΦΑΝΙΣΕ "Last name is required!"
    cancel = true
ΤΕΛΟΣ_ΑΝ

AN ((phoneNumberText <> null) && (phoneNumberText < 10)) TOTE
    ΕΜΦΑΝΙΣΕ "Invalid phone number!"
    cancel = true
ΤΕΛΟΣ_ΑΝ
```

```

AN ((creditCardNumberText <> null) && (creditCardNumberText <
15)) ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ "Invalid credit card number!"
    cancel = true
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (cancel == false) ΤΟΤΕ
    Επικοινωνία με Βάση Δεδομένων
    ΔΙΑΒΑΣΕ databaseUsername, databaseEmail
    AN (email == databaseEmail) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "E-mail is already in use. Please choose another
e-mail or recover lost password!"
        Διακοπή διαδικασίας εγγραφής
    ΤΕΛΟΣ_ΑΝ

    AN (username == databaseUsername) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Username is already in use. Please choose
another one!"
        Διακοπή διαδικασίας εγγραφής
    ΑΛΛΙΩΣ
        Είσοδος του χρήστη στη βάση δεδομένων
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ

```

Ο αλγόριθμος αυτός ελέγχει τα στοιχεία που έχει εισάγει ο χρήστης για να δημιουργήσει το λογαριασμό του. Το πρώτο στάδιο ελέγχου πραγματοποιείται από την ίδια την εφαρμογή, τέτοιοι έλεγχοι αφορούν την συμπλήρωση των απαραίτητων πεδίων της φόρμας εγγραφής του χρήστη.

Το δεύτερο στάδιο ελέγχου πραγματοποιείται από την εφαρμογή σε συνεργασία με τη Βάση Δεδομένων, δηλαδή η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και ελέγχει τα στοιχεία που έχει δώσει ο χρήστης. Αν το email ή username που έχει δώσει ο χρήστης χρησιμοποιούνται από κάποιον άλλο χρήστη, τότε ο χρήστης πρέπει να δώσει κάποιο άλλο email ή username, αλλιώς ο χρήστης δημιουργεί επιτυχώς το λογαριασμό του.

Η πολυπλοκότητα αυτού του αλγορίθμου είναι $f(n) = 1$, εφόσον δεν υπάρχει κάποια διαδικασία επανάληψης και ο αριθμός των οδηγιών που χρειάζεται ο αλγόριθμος είναι απλά μια σταθερά.

2.1.2.3 My Account Activity

ΑΡΧΗ

```
AN (changePasswordButton == pressed) TOTE
    cancel = false
    choice = "password"
    AN (changeButton == pressed) TOTE
        AN (oldPasswordText == null) TOTE
            ΕΜΦΑΝΙΣΕ "Old Password is required!"
            cancel = true
        ΤΕΛΟΣ_ΑΝ
    AN (newPasswordText <= 4) TOTE
        ΕΜΦΑΝΙΣΕ "New Password is too short!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ
    AN (confirmPasswordText <> newPasswordText) TOTE
        ΕΜΦΑΝΙΣΕ "Passwords don't match!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (changePhoneNumberButton == pressed) TOTE
    cancel = false
    choice = "pnumber"
    AN (changeButton == pressed) TOTE
        AN ((phoneNumberText <> null) && (phoneNumberText < 10))
            TOTE
                ΕΜΦΑΝΙΣΕ "Invalid phone number!"
                cancel = true
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (changeCreditCardNumberButton == pressed) TOTE
    cancel = false
    choice = "crdtnumber"
    AN (changeButton == pressed) TOTE
        AN ((creditCardNumberText <> null) &&
            (creditCardNumberText < 15)) TOTE
            ΕΜΦΑΝΙΣΕ "Invalid credit card number!"
            cancel = true
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

AN (cancel == false) TOTE
    Επικοινωνία με Βάση Δεδομένων
    ΔΙΑΒΑΣΕ databasePassword, databasePhoneNumber,
    databaseCreditCardNumber, databaseAddress
```

```

AN ((choice == "password") && (oldPasswordText <>
databasePassword)) TOTE
    ΕΜΦΑΝΙΣΕ "Wrong old password, please retype your old
password!"
    Διακοπή διαδικασίας αλλαγής password
ΑΛΛΙΩΣ
    Γίνεται αλλαγή των πεδίων στη Βάση Δεδομένων
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ

```

Αυτός ο αλγόριθμος αφορά την τροποποίηση των στοιχείων του λογαριασμού του χρήστη. Με την βοήθεια αυτού του αλγόριθμου ο χρήστης μπορεί να αλλάξει τον κωδικό πρόσβασής του, την διεύθυνσή του, το τηλέφωνο επικοινωνίας του και τον αριθμό της πιστωτικής του κάρτας. Εφόσον γίνει έλεγχος στα απαραίτητα πεδία συμπλήρωσης, η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και πραγματοποιείται η αλλαγή των στοιχείων του χρήστη.

Η πολυπλοκότητα αυτού του αλγόριθμου είναι $f(n) = 1$, εφόσον δεν υπάρχει κάποια διαδικασία επανάληψης και ο αριθμός των οδηγιών που χρειάζεται ο αλγόριθμος είναι απλά μια σταθερά.

2.1.2.4 Appointment Activity

ΑΡΧΗ

```
ΔΙΑΒΑΣΕ extras, πλήθοςΡαντεβούI, ΛίσταΡαντεβού[]

ΑΝ (πλήθοςΡαντεβούI < 1) ΤΟΤΕ
    ΛίσταΡαντεβού[0] = "No Appointments"
ΑΛΛΙΩΣ
    ΛίσταΡαντεβού[0] = "Your Appointments"
    ΑΝ (extras <> null) ΤΟΤΕ
        ΔΙΑΒΑΣΕ ραντεβού, θέσηI, κατάσταση
        ΑΝ (κατάσταση == "change") ΤΟΤΕ
            ΛίσταΡαντεβού[θέσηI] = ραντεβού
        ΑΛΛΙΩΣ ΑΝ (κατάσταση == "remove") ΤΟΤΕ
            ΓΙΑ k ΑΠΟ θέσηI ΜΕΧΡΙ πλήθοςΡαντεβούI ΜΕ ΒΗΜΑ 1
                ΑΝ (k == πλήθοςΡαντεβούI) ΤΟΤΕ
                    ΛίσταΡαντεβού[k] = null
                    πλήθοςΡαντεβούI = πλήθοςΡαντεβούI - 1
                ΑΠΟΘΗΚΕΥΣΕ πλήθοςΡαντεβούI
            ΑΛΛΙΩΣ
                ΛίσταΡαντεβού[k] = ΛίσταΡαντεβού[k + 1]
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
    ΔΙΑΒΑΣΕ data, πληροφορίαΡαντεβού
    ΛίσταΡαντεβού[πλήθοςΡαντεβούI] = πληροφορίαΡαντεβού
ΤΕΛΟΣ_ΑΝ

ΑΠΟΘΗΚΕΥΣΕ ΛίσταΡαντεβού[]
ΓΙΑ k ΑΠΟ 0 ΜΕΧΡΙ πλήθοςΡαντεβούI ΜΕ ΒΗΜΑ 1
    ΕΜΦΑΝΙΣΕ ΛίσταΡαντεβού[k]
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ

ΑΝ (newAppointmentButton == pressed) ΤΟΤΕ
    ΑΠΟΘΗΚΕΥΣΕ ΛίσταΡαντεβού[], πλήθοςΡαντεβούI
    Ανακατεύθυνση του χρήστη στη κλάση SetAppointmentActivity
ΤΕΛΟΣ_ΑΝ

ΑΝ (deleteAllButton == pressed) ΤΟΤΕ
    ΕΜΦΑΝΙΣΕ "Are you sure you want to delete all of your
appointments?"
    ΑΝ (yesButton == pressed) ΤΟΤΕ
        ΓΙΑ k ΑΠΟ 0 ΜΕΧΡΙ πλήθοςΡαντεβούI ΜΕ ΒΗΜΑ 1
            ΛίσταΡαντεβού[k] = null
        ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
        ΛίσταΡαντεβού[0] = "No Appointments"
        πλήθοςΡαντεβούI = 0
        ΑΠΟΘΗΚΕΥΣΕ πλήθοςΡαντεβούI, ΛίσταΡαντεβού[]
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
```

```

AN (ραντεβού == pressed) TOTE
    πληροφορίαΡαντεβού = ραντεβού
    extras = πληροφορίαΡαντεβού + "" + θέσηΡαντεβού
ΑΠΟΘΗΚΕΥΣΕ extras
    Ανακατεύθυνση του χρήστη στη κλάση SetAppointmentActivity
ΤΕΛΟΣ_AN
ΤΕΛΟΣ

```

Αυτός ο αλγόριθμος αφορά την πρόσθεση ραντεβού, τη τροποποίηση και τη διαγραφή τους. Αρχικά, ο αλγόριθμος εμφανίζει όλα τα ραντεβού του χρήστη στην οθόνη της συσκευής, αν υπάρχουν. Αν κάποιο ραντεβού έχει τροποποιηθεί εμφανίζεται τροποποιημένο, αν κάποιο ραντεβού έχει διαγραφεί τότε δεν εμφανίζεται στη λίστα με τα ραντεβού.

Αν ο χρήστης επιλέξει να προσθέσει ένα ραντεβού, η εφαρμογή ανακατευθύνει τον χρήστη στη Δραστηριότητα Δημιουργίας Ραντεβού όπου ο χρήστης συμπληρώνει όλες τις απαραίτητες λεπτομέρειες. Αν ο χρήστης επιλέξει ένα συγκεκριμένο ραντεβού, η εφαρμογή τον ανακατευθύνει ξανά στη Δραστηριότητα Δημιουργίας Ραντεβού για να τροποποιήσει το ραντεβού του. Αν ο χρήστης επιλέξει να διαγράψει όλα τα ραντεβού τότε αυτά διαγράφονται.

Η πολυπλοκότητα αυτού του αλγορίθμου είναι $f(n) = n$, εφόσον υπάρχουν τρεις διαδικασίες επανάληψης οι οποίες όμως δεν είναι η μία μέσα στην άλλη, όπου n είναι το πλήθος των επαναλήψεων όλων των ραντεβού του χρήστη στη λίστα.

2.1.2.5 Set Appointment Activity

ΑΡΧΗ

ΔΙΑΒΑΣΕ extras, πλήθοςΡαντεβούI, ΛίσταΡαντεβού[]
bool = false

ΑΝ (extras <> null) **ΤΟΤΕ**

ΔΙΑΒΑΣΕ πληροφορίαΡαντεβού, θέσηΡαντεβού

ΑΝ (πληροφορίαΡαντεβού <> null) **ΤΟΤΕ**

ΔΙΑΒΑΣΕ πληροφορίαΡαντεβούSubject,
πληροφορίαΡαντεβούDescription,
πληροφορίαΡαντεβούLocation, πληροφορίαΡαντεβούDate,
πληροφορίαΡαντεβούHour, πληροφορίαΡαντεβούMin

ΕΜΦΑΝΙΣΕ πληροφορίαΡαντεβούSubject,
πληροφορίαΡαντεβούDescription,
πληροφορίαΡαντεβούLocation, πληροφορίαΡαντεβούDate,
πληροφορίαΡαντεβούHour, πληροφορίαΡαντεβούMin

ΕΜΦΑΝΙΣΕ deleteButton

ΤΕΛΟΣ_ΑΝ

bool = true

ΑΛΛΙΩΣ

ΕΜΦΑΝΙΣΕ τρέχουσα ημερομηνία και ώρα συστήματος

ΤΕΛΟΣ_ΑΝ

ΑΝ (cancelButton == pressed) **ΤΟΤΕ**

Ανακατεύθυνση του χρήστη στη κλάση AppointmentActivity

ΤΕΛΟΣ_ΑΝ

ΑΝ (deleteButton == pressed) **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ "Are you sure you want to delete this appointments?"

ΑΝ (yesButton == pressed) **ΤΟΤΕ**

θέσηI = θέσηΡαντεβού

κατάσταση = "remove"

extras = θέσηI + "" + κατάσταση

ΑΠΟΘΗΚΕΥΣΕ extras

Ανακατεύθυνση του χρήστη στη κλάση AppointmentActivity

ΤΕΛΟΣ_ΑΝ

ΤΕΛΟΣ_ΑΝ

ΑΝ (doneButton == pressed) **ΤΟΤΕ**

cancel = false

ΑΝ (subjectText == null) **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ "This field is required!"

cancel = true

ΤΕΛΟΣ_ΑΝ

ΑΝ (locationText == null) **ΤΟΤΕ**

ΕΜΦΑΝΙΣΕ "This field is required!"

cancel = true

ΤΕΛΟΣ_ΑΝ

ΑΝ (selectedYear < currentYear) **ΤΟΤΕ**

```

        ΕΜΦΑΝΙΣΕ "Invalid year!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    ΑΝ ((selectedYear == currentYear) && (selectedMonth <
currentMonth)) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Invalid month!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    ΑΝ ((selectedYear == currentYear) && (selectedMonth ==
selectedMonth) && (selectedDay < selectedDay)) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Invalid day!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    ΑΝ ((selectedYear == currentYear) && (selectedMonth ==
selectedMonth) && (selectedDay == selectedDay) && (selectedHour
< selectedHour)) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "Invalid hour!"
        cancel = true
    ΤΕΛΟΣ_ΑΝ

    ΑΝ (πλήθοςΡαντιεβούΙ >= 1) ΤΟΤΕ
        ΓΙΑ k ΑΠΟ 0 ΜΕΧΡΙ πλήθοςΡαντιεβούΙ ΜΕ ΒΗΜΑ 1

            ΑΝ (ΛίσταΡαντιεβού[k] <> "Your Appointments") ΤΟΤΕ
                ΔΙΑΒΑΣΕ ΛίσταΡαντιεβού[k]
                ΔΙΑΒΑΣΕ curAppointHour, curAppointMin

                ΑΝ (bool == true) ΤΟΤΕ
                    ΑΝ ((πληροφορίαΡαντιεβούHour ==
curAppointHour) &&
(πληροφορίαΡαντιεβούMin ==
curAppointMin)) ΤΟΤΕ
                        continue
                    ΤΕΛΟΣ_ΑΝ
                ΤΕΛΟΣ_ΑΝ

                ΑΝ ((selectedYear == currentYear) &&
(selectedMonth == selectedMonth) &&
(selectedDay == selectedDay)) ΤΟΤΕ
                    diffHour = selectedHour -
curAppointHour
                    ΑΝ (diffHour == 0) ΤΟΤΕ
                        diffMint = selectedMin -
curAppointMin
                        ΑΝ (((diffMint >= 0) && (diffMint
<= 30)) || ((diffMint <= 0) &&
(diffMint >= -30))) ΤΟΤΕ
                            ΕΜΦΑΝΙΣΕ "There is another
appointment in the next 30
minutes, please change the
start hour!"

```

```

        cancel = true
        break
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ ΑΝ (diffHour == 1) ΤΟΤΕ
    diffMint = (59 - curAppointMin) +
    selectedMin
    ΑΝ ((diffMint >= 0) && (diffMint
    <= 30)) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "There is another
        appointment in the next 30
        minutes, please change the
        start hour!"
        cancel = true
        break
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ ΑΝ (diffHour == -1) ΤΟΤΕ
    diffMint = (59 - selectedMin) +
    curAppointMin
    ΑΝ ((diffMint >= 0) && (diffMint
    <= 30)) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "There is another
        appointment in the next 30
        minutes, please change the
        start hour!"
        cancel = true
        break
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
ΑΝ (cancel == false) ΤΟΤΕ
    ΑΝ (extras <> null) ΤΟΤΕ
        θέσηI = θέσηΡαντιεβού
        κατάσταση = "change"
        extras = θέσηI + "" + κατάσταση
        data = πλήθοςΡαντιεβούI + "" + πληροφορίαΡαντιεβού
        ΑΠΟΘΗΚΕΥΣΕ extras
    ΑΛΛΙΩΣ
        πλήθοςΡαντιεβούI = πλήθοςΡαντιεβούI + 1
        data = πλήθοςΡαντιεβούI + "" + πληροφορίαΡαντιεβού
        ΑΠΟΘΗΚΕΥΣΕ πλήθοςΡαντιεβούI
    ΤΕΛΟΣ_ΑΝ
    ΑΠΟΘΗΚΕΥΣΕ data
    Ανακατεύθυνση του χρήστη στη κλάση AppointmentActivity
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ

```

Αυτός ο αλγόριθμος βοηθάει στη συμπλήρωση των λεπτομερειών στα ραντεβού κατά την εισαγωγή νέων ραντεβού και κατά την τροποποίησή τους. Επίσης, δίνει τη δυνατότητα στο χρήστη να διαγράψει τα ραντεβού του ένα προς ένα. Κατά την εισαγωγή των ραντεβού γίνεται έλεγχος για το αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία. Αν δεν υπάρξει κάποιο πρόβλημα τότε το ραντεβού εισέρχεται επιτυχώς στη λίστα με τα ραντεβού, ο χρήστης ανακατευθύνεται στη Δραστηριότητα Ραντεβού όπου βλέπει όλα τα ραντεβού του και αυτό που μόλις εισήγαγε.

Κατά την τροποποίηση των ραντεβού, ο χρήστης μπορεί να τροποποιήσει οποιοδήποτε πεδίο θέλει και να αποθηκεύσει εκ νέου το ραντεβού αυτό. Επίσης, στη διαδικασία της τροποποίησης των ραντεβού, ο χρήστης μπορεί να διαγράψει το συγκεκριμένο ραντεβού. Μετά από αυτή τη διαδικασία ο χρήστης ανακατευθύνεται στη Δραστηριότητα Ραντεβού με τροποποιημένο το ραντεβού του ή διαγραμμένο.

Κατά την εισαγωγή ή τροποποίηση ενός ραντεβού, ο χρήστης δεν μπορεί να επιλέξει ως ώρα έναρξης ραντεβού μικρότερη του μισάωρου από το προηγούμενο ή το επόμενο ραντεβού του, εφόσον τα δύο ραντεβού έχουν την ίδια ημερομηνία.

Η πολυπλοκότητα αυτού του αλγορίθμου είναι $f(n) = n$, εφόσον υπάρχει τουλάχιστον μία διαδικασία επανάληψης, όπου n είναι το πλήθος των επαναλήψεων όλων των ραντεβού του χρήστη στη λίστα.

2.1.2.6 Station Activity

ΑΡΧΗ

```
ΔΙΑΒΑΣΕ extras, currentTime  
exist = false
```

```
AN (extras <> null) ΤΟΤΕ
```

```
    ΔΙΑΒΑΣΕ κατάσταση
```

```
ΤΕΛΟΣ_ΑΝ
```

```
AN (κατάσταση == "gas_station.xml") ΤΟΤΕ
```

```
    διαχωρισμός = "gas station"
```

```
    StationActivityTitle = "Gas Stations"
```

```
    StationActivityIcon = gasStationIcon
```

```
    StationActivityBackgroundColor = orange
```

```
    nearButtonName = "Nearest Gas Station"
```

```
ΑΛΛΙΩΣ AN (κατάσταση == "supermarket.xml") ΤΟΤΕ
```

```
    διαχωρισμός = "supermarket"
```

```
    StationActivityTitle = "Supermarkets"
```

```
    StationActivityIcon = supermarketIcon
```

```
    StationActivityBackgroundColor = green
```

```
    nearButtonName = "Nearest Supermarket"
```

```
ΤΕΛΟΣ_ΑΝ
```

```
ΔΙΑΒΑΣΕ αρχείο(κατάσταση)
```

```
AN (αρχείο(κατάσταση) <> null) ΤΟΤΕ
```

```
    exist = true
```

```
    ΔΙΑΒΑΣΕ αρχείο(κατάσταση)I
```

```
    ΓΙΑ k ΑΠΟ 0 ΜΕΧΡΙ αρχείο(κατάσταση)I ΜΕ ΒΗΜΑ 1
```

```
        ΕΜΦΑΝΙΣΕ αρχείο(κατάσταση)[k]
```

```
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
```

```
ΑΛΛΙΩΣ
```

```
    exist = false
```

```
    ΕΜΦΑΝΙΣΕ "Press refresh button to see the list."
```

```
ΤΕΛΟΣ_ΑΝ
```

```
AN (refreshButton == pressed) ΤΟΤΕ
```

```
    Επικοινωνία με Server
```

```
    ΔΙΑΒΑΣΕ αρχείο(κατάσταση)
```

```
    exist = true
```

```
ΤΕΛΟΣ_ΑΝ
```

```
AN (row == pressed) ΤΟΤΕ
```

```
    ΔΙΑΒΑΣΕ latitudeRowPressed, longitudeRowPressed,
```

```
    titleRowPressed
```

```
    extras = list + "" + κατάσταση + "" + latitudeRowPressed + "" +
```

```
    longitudeRowPressed + "" + titleRowPressed
```

```
    ΑΠΟΘΗΚΕΥΣΕ extras
```

```
    Ανακατεύθυνση του χρήστη στη κλάση MapActivity
```

```
ΤΕΛΟΣ_ΑΝ
```

```
AN (nearButton == pressed) ΤΟΤΕ
```

```
    AN (exist == true) ΤΟΤΕ
```

```
        ΔΙΑΒΑΣΕ statusOfGPS
```

```

AN (statusOfGPS == true) ΤΟΤΕ
    AN (διαχωρισμός == "supermarket") ΤΟΤΕ
        AN ((currentTime >= 800) && (currentTime <
            2100)) ΤΟΤΕ
            AN ((currentTime >= 2030) &&
                (currentTime < 2100)) ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "Supermarkets are
                    closing at 21:00 and you may not
                    have enough time to complete your
                    shopping!"
                AN (okButton == pressed) ΤΟΤΕ
                    extras = nearest + "" +
                        κατάσταση
                    ΑΠΟΘΗΚΕΥΣΕ extras
                    Ανακατεύθυνση του χρήστη
                    στη κλάση MapActivity
                ΤΕΛΟΣ_ΑΝ
            ΑΛΛΙΩΣ
                extras = nearest + "" + κατάσταση
                ΑΠΟΘΗΚΕΥΣΕ extras
                Ανακατεύθυνση του χρήστη στη
                κλάση MapActivity
            ΤΕΛΟΣ_ΑΝ
        ΑΛΛΙΩΣ
            ΕΜΦΑΝΙΣΕ "There are no supermarkets
                open at this hour."
            ΤΕΛΟΣ_ΑΝ
        ΑΛΛΙΩΣ AN (διαχωρισμός == "gas station") ΤΟΤΕ
            AN ((currentTime >= 600) && (currentTime <
                2230)) ΤΟΤΕ
                extras = nearest + "" + κατάσταση
                ΑΠΟΘΗΚΕΥΣΕ extras
                Ανακατεύθυνση του χρήστη στη κλάση
                MapActivity
            ΑΛΛΙΩΣ
                ΕΜΦΑΝΙΣΕ "Some gas stations are closed
                    at this hour. The nearest open will be
                    found."
                AN (okButton == pressed) ΤΟΤΕ
                    extras = nearest + "" + κατάσταση
                    ΑΠΟΘΗΚΕΥΣΕ extras
                    Ανακατεύθυνση του χρήστη στη
                    κλάση MapActivity
                ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΕΜΦΑΝΙΣΕ "GPS is disabled!"
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΕΜΦΑΝΙΣΕ "Refresh the list to find the closest " +
            διαχωρισμός
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ

```

ΤΕΛΟΣ

Ο αλγόριθμος στη Δραστηριότητα Σταθμός είναι υπεύθυνος για την εμφάνιση της λίστας όλων των βενζινάδικών – σούπερ μάρκετ στην οθόνη της συσκευής. Αν η λίστα δεν υπάρχει αποθηκευμένη στη συσκευή, ενημερώνεται ο χρήστης και πατώντας το κουμπί ανανέωσης ο αλγόριθμος φροντίζει να επικοινωνήσει με τον Διακομιστή (Server) και να κατεβάσει και να αποθηκεύσει στη συσκευή τη λίστα με τα βενζινάδικα και τα σούπερ μάρκετ. Αν λίστα υπάρχει στη συσκευή, ο αλγόριθμος εμφανίζει επιτυχώς όλα τα βενζινάδικα – σούπερ μάρκετ στην οθόνη της συσκευής. Επίσης, ο χρήστης μπορεί να πατήσει το κουμπί ανανέωσης ακόμα και αν η λίστα υπάρχει στη συσκευή, αυτό είναι εφικτό επειδή ο χρήστης μπορεί να χρειαστεί να ανανεώσει τη λίστα με τις πιο πρόσφατες πληροφορίες.

Ακόμα, ο χρήστης πατώντας σε ένα βενζινάδικο – σούπερ μάρκετ από τη λίστα, ανακατευθύνεται στη Δραστηριότητα Χάρτη όπου εμφανίζεται το συγκεκριμένο βενζινάδικο – σούπερ μάρκετ. Αν ο χρήστης πατήσει το κουμπί για την εύρεση του πλησιέστερου βενζινάδικου – σούπερ μάρκετ, πραγματοποιείται έλεγχος της ώρας. Αν η τρέχουσα ώρα είναι μεγαλύτερη από την ώρα καταστημάτων, τότε δεν γίνεται προσπάθεια εύρεσης του πλησιέστερου σταθμού. Αν η τρέχουσα ώρα είναι μεγαλύτερη από την ώρα καταστημάτων μερικών σταθμών, τότε ενημερώνεται ο χρήστης γι' αυτό, ανακατευθύνεται στη Δραστηριότητα Χάρτη και πραγματοποιείται η εύρεση του πλησιέστερου σταθμού. Αλλιώς, αν η τρέχουσα ώρα είναι στα πλαίσια των ωρών καταστημάτων, ανακατευθύνεται ο χρήστης στη Δραστηριότητα Χάρτη και γίνεται η εύρεση του πλησιέστερου σταθμού χωρίς κανένα πρόβλημα.

Η πολυπλοκότητα αυτού του αλγορίθμου είναι $f(n) = n$, εφόσον υπάρχει τουλάχιστον μία διαδικασία επανάληψης, όπου n είναι το πλήθος των επαναλήψεων όλων των σταθμών (βενζινάδικων – σούπερ μάρκετ) στη λίστα.

2.1.2.7 Map Activity

ΑΡΧΗ

```
ΔΙΑΒΑΣΕ extras, currentDate, currentTime, HER_CENTER_LATITUDE,  
HER_CENTER_LONGITUDE  
waitingForLocation = false  
navigation = false  
traffic = false
```

```
ΑΝ (extras <> null) ΤΟΤΕ
```

```
    ΔΙΑΒΑΣΕ κατάσταση
```

```
ΤΕΛΟΣ_ΑΝ
```

```
ΑΝ ((κατάσταση[0] == "map") || (κατάσταση[0] == "list") ||  
(κατάσταση[0] == "nearest")) ΤΟΤΕ
```

```
    ΑΝ (κατάσταση[1] == "gas_station.xml") ΤΟΤΕ
```

```
        seperation = "gas station"
```

```
        StationActivityTitle = "Gas Stations"
```

```
        StationActivityIcon = gasStationIcon
```

```
        StationActivityBackgroundColor = orange
```

```
    ΑΛΛΙΩΣ ΑΝ (κατάσταση[1] == "supermarket.xml") ΤΟΤΕ
```

```
        seperation = "supermarket"
```

```
        StationActivityTitle = "Supermarkets"
```

```
        StationActivityIcon = supermarketIcon
```

```
        StationActivityBackgroundColor = green
```

```
    ΤΕΛΟΣ_ΑΝ
```

```
ΤΕΛΟΣ_ΑΝ
```

```
ΔΙΑΒΑΣΕ map
```

```
ΑΡΧΗ_ΕΝΩ
```

```
    ΔΙΑΒΑΣΕ myPosition, myLatitude, myLongitude
```

```
    myPosition1 = myPosition
```

```
    ΑΝ (waitingForLocation == true) ΤΟΤΕ
```

```
        ΑΝ (myLatitude <> null) ΤΟΤΕ
```

```
            waitingForLocation = false
```

```
            findNearest(myLatitude, myLongitude)
```

```
        ΤΕΛΟΣ_ΑΝ
```

```
    ΤΕΛΟΣ_ΑΝ
```

```
    myLastLatitude = myLatitude
```

```
    myLastLongitude = myLongitude
```

```
    ΑΠΟΘΗΚΕΥΣΕ myLastLatitude, myLastLongitude
```

```
    ΑΝ (navigation == true) ΤΟΤΕ
```

```
        mapCameraPosition = (myLatitude, myLongitude)
```

```
    ΤΕΛΟΣ_ΑΝ
```

```
    ΔΙΑΒΑΣΕ myPosition
```

```
    myPosition2 = myPosition
```

```
ΕΝΩ (myPosition1 <> myPosition2)
```

```
ΑΝ (κατάσταση[0] == "nearest") ΤΟΤΕ
```

```
    exist = true
```

```
    ΔΙΑΒΑΣΕ αρχείο(κατάσταση)
```

```
    ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ αρχείο(κατάσταση[1]).Size ΜΕ ΒΗΜΑ 1
```



```

stationLat = αρχείο(κατάσταση[1])[i]Latitude
stationLong = αρχείο(κατάσταση[1])[i]Longitude
stationClosingHour = αρχείο(κατάσταση[1])[i]ClosingHour
stationName = αρχείο(κατάσταση[1])[i]Name
stationAddress = αρχείο(κατάσταση[1])[i]Address
TOTE
    AN ((currentTime >= 2230) || (currentTime < 30)) TOTE
        AN (stationClosingHour <= 600) TOTE
            ΕΜΦΑΝΙΣΕ drawMarkerOnMap(stationLat,
                stationLong, stationName, stationAddress)
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ AN ((currentTime >= 30) || (currentTime < 600))
        AN ((stationClosingHour > 30) &&
            (stationClosingHour <= 600)) TOTE
            ΕΜΦΑΝΙΣΕ drawMarkerOnMap(stationLat,
                stationLong, stationName, stationAddress)
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        ΕΜΦΑΝΙΣΕ drawMarkerOnMap(stationLat, stationLong,
            stationName, stationAddress)
        ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
AN (myLatitude == 0) TOTE
    AN (lastKnownPositionButton == pressed) TOTE
        ΔΙΑΒΑΣΕ myLastLatitude, myLastLongitude
        AN (myLastLatitude == null) TOTE
            ΕΜΦΑΝΙΣΕ "Last known position is not
                available!"
        ΑΛΛΙΩΣ
            findNearest(myLastLatitude, myLastLongitude)
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
    waitingForLocation = true
ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ AN (κατάσταση[0] == "list") TOTE
    ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ αρχείο(κατάσταση[1])Size ΜΕ ΒΗΜΑ 1
        stationLat = αρχείο(κατάσταση[1])[i]Latitude
        stationLong = αρχείο(κατάσταση[1])[i]Longitude
        stationName = αρχείο(κατάσταση[1])[i]Name
        stationAddress = αρχείο(κατάσταση[1])[i]Address
        ΕΜΦΑΝΙΣΕ drawMarkerOnMap(stationLat, stationLong,
            stationName, stationAddress)
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    stationLat = κατάσταση[2]
    stationLong = κατάσταση[3]
    stationName = κατάσταση[4]
    ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ markerOnMapSize ΜΕ ΒΗΜΑ 1
        title = markerOnMap[i]Name
        AN (stationName == title) TOTE
            mapCameraPosition = (stationLat, stationLong)
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΑΛΛΙΩΣ
    ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ αρχείο(κατάσταση[1])Size ΜΕ ΒΗΜΑ 1
        stationLat = αρχείο(κατάσταση[1])[i]Latitude
        stationLong = αρχείο(κατάσταση[1])[i]Longitude
        stationName = αρχείο(κατάσταση[1])[i]Name

```

```

        stationAddress = αρχείο(κατάσταση[1])[i]Address
        ΕΜΦΑΝΙΣΕ drawMarkerOnMap(stationLat, stationLong,
        stationName, stationAddress)
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
    mapCameraPosition = (HER_CENTER_LATITUDE, HER_CENTER_LONGITUDE)
ΤΕΛΟΣ_ΑΝ

ΑΝ (navigationButton == pressed) ΤΟΤΕ
    ΑΝ (marker <> null) ΤΟΤΕ
        ΑΝ (gpsEnabled == true) ΤΟΤΕ
            navigation = true
            ΑΝ (waitingForLocation == true) ΤΟΤΕ
                navigateFromTo(myLastLatitude,
                myLastLongitude, destLatitude, destLongitude)
            ΑΛΛΙΩΣ
                navigateFromTo(myLatitude, myLongitude,
                destLatitude, destLongitude)
            ΤΕΛΟΣ_ΑΝ
        ΑΛΛΙΩΣ
            ΕΜΦΑΝΙΣΕ "GPS is disabled!"
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ

findNearest(Latitude, Longitude)
ΑΡΧΗ
    appointment = checkForAppointment
    ΑΝ (appointment <> "no") ΤΟΤΕ
        ΑΝ (traffic == true) ΤΟΤΕ
            ΑΝ (seperation == "gas station") ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "You have an appointment at " +
                appointment[1] + " at " + appointment[0] + ".
                Do you want to find a "
                + seperation + " near that location? Keep in
                mind that there is traffic on the road at
                this hour!"
            ΑΛΛΙΩΣ ΑΝ (seperation == "supermarket") ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "You have an appointment at " +
                appointment[1] + " at " + appointment[0] + ".
                Do you want to find a " + seperation +
                " near that location? Keep in mind that there
                is traffic on the road at this hour and you
                may not have enough time to complete your
                shopping!"
            ΤΕΛΟΣ_ΑΝ
        ΑΛΛΙΩΣ
            ΑΝ (seperation == "gas station") ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "You have an appointment at " +
                appointment[1] + " at " + appointment[0] + ".
                Do you want to find a "
                + seperation + " near that location?"
            ΑΛΛΙΩΣ ΑΝ (seperation == "supermarket") ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "You have an appointment at " +
                appointment[1] + " at " + appointment[0]
                + " and you may not have enough time to
                complete your shopping. Do you want to find a
                " + seperation + " near that location?"
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ

```

```

        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
    ΑΝ (yesButton == pressed) ΤΟΤΕ
        ΑΝ (internetConnection == true) ΤΟΤΕ
            locationAddress = appointment[0]
            Επικοινωνία με Server για μετατροπή της
            διεύθυνσης locationAddress σε Latitude και
            Longitude
            ΔΙΑΒΑΣΕ locationAddressLatitude,
            locationAddressLongitude
            marker =
            nearestMarker(locationAddressLatitude,
            locationAddressLongitude)
            ΑΝ (markerKey > 2500) ΤΟΤΕ
                ΕΜΦΑΝΙΣΕ "There isn't a " + seperation
                + " near your appointment. The nearest
                "
                + seperation + " in your location will
                be found."
                ΑΝ (okButton == pressed) ΤΟΤΕ
                    marker = nearestMarker(Latitude,
                    Longitude)
                ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΑΝ
        ΑΛΛΙΩΣ
            ΕΜΦΑΝΙΣΕ "There isn't internet connection to
            determine the location of your appointment.
            The nearest " + seperation + " in your
            location will be found."
            ΑΝ (okButton == pressed) ΤΟΤΕ
                marker = nearestMarker(Latitude,
                Longitude)
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ ΑΝ (noButton == pressed) ΤΟΤΕ
        marker = nearestMarker(Latitude, Longitude)
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
    ΑΝ (traffic == true) ΤΟΤΕ
        ΕΜΦΑΝΙΣΕ "There is traffic on the road and
        transportation will be slower."
        ΑΝ (okButton == pressed) ΤΟΤΕ
            marker = nearestMarker(Latitude, Longitude)
        ΤΕΛΟΣ_ΑΝ
    ΑΛΛΙΩΣ
        marker = nearestMarker(Latitude, Longitude)
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ

nearestMarker(Latitude, Longitude)
ΑΡΧΗ
    ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ αρχείο(κατάσταση[1])Size ΜΕ ΒΗΜΑ 1
        stationLat = αρχείο(κατάσταση[1])[i]Latitude
        stationLong = αρχείο(κατάσταση[1])[i]Longitude
        stationClosingHour = αρχείο(κατάσταση[1])[i]ClosingHour
        stationName = αρχείο(κατάσταση[1])[i]Name

```

```

AN ((currentTime >= 2230) || (currentTime < 30)) TOTE
    AN (stationClosingHour <= 600) TOTE
        Επικοινωνία με Server για τον υπολογισμό του
        distanceBetween(stationLat, stationLong,
        Latitude, Longitude)
        distance = distanceBetween(stationLat,
        stationLong, Latitude, Longitude)
        distanceHashMap = (distance, stationName)
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ AN ((currentTime >= 30) && (currentTime < 600))
TOTE
    AN ((stationClosingHour > 30) &&
    (stationClosingHour <= 600)) TOTE
        Επικοινωνία με Server για τον υπολογισμό του
        distanceBetween(stationLat, stationLong,
        Latitude, Longitude)
        distance = distanceBetween(stationLat,
        stationLong, Latitude, Longitude)
        distanceHashMap = (distance, stationName)
    ΤΕΛΟΣ_ΑΝ
ΑΛΛΙΩΣ
        Επικοινωνία με Server για τον υπολογισμό του
        distanceBetween
        (stationLat, stationLong, Latitude, Longitude)
        distance = distanceBetween(stationLat, stationLong,
        Latitude, Longitude)
        distanceHashMap = (distance, stationName)
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
treeMap = distanceHashMap
key = treeMap[0]Key
value = treeMap[0]Value
ΓΙΑ i ΑΠΟ 0 ΜΕΧΡΙ markerOnMapSize ΜΕ ΒΗΜΑ 1
    title = markerOnMap[i]Name
    AN (value == title) TOTE
        destinationLatitude = markerOnMap[i]Latitude
        destinationLongitude = markerOnMap[i]Longitude
        mapCameraPosition = (destinationLatitude,
        destinationLongitude)
        break
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
return destinationLatitude + "" + destinationLongitude + "" +
key
ΤΕΛΟΣ

navigateFromTo(startLatitude, startLongitude, destLatitude,
destLongitude)
ΑΡΧΗ
AN (navigation == true) TOTE
    AN (internetConnection == true) TOTE
        mapCameraPosition = (startLatitude, startLongitude)
        Επικοινωνία με Server για την εμφάνιση οδηγιών
κατεύθυνσης από fromPosition σε toPosition στον
χάρτη
        fromPosition = (startLatitude, startLongitude)
        toPosition = (destLatitude, destLongitude)

```

```

        getDirections(fromPosition, toPosition)
    ΑΛΛΙΩΣ
        navigation = false
        ΕΜΦΑΝΙΣΕ "No Network Available!"
    ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ

```

```

checkForAppointment
ΑΡΧΗ

```

```

    ΔΑΙΒΑΣΕ λίσταΡαντιεβού[], πλήθοςΡαντιεβούI
    appoint = false
    currentDate, currentTime
    ΑΝ (πλήθοςΡαντιεβούI < 1) ΤΟΤΕ
        appoint = false
    ΑΛΛΙΩΣ
        ΓΙΑ k ΑΠΟ 0 ΜΕΧΡΙ πλήθοςΡαντιεβούI ΜΕ ΒΗΜΑ 1
            ΑΝ (λίσταΡαντιεβού[k] <> "Your Appointments") ΤΟΤΕ
                ΔΙΑΒΑΣΕ λίσταΡαντιεβού[k], curAppointHour,
                curAppointMin, curAppointLocation
                ΑΝ ((selectedYear == currentYear) &&
                (selectedMonth == selectedMonth) &&
                (selectedDay == selectedDay)) ΤΟΤΕ
                    diffHour = selectedHour -
curAppointHour

                    ΑΝ (diffHour == 0) ΤΟΤΕ
                        diffMint = selectedMin -
                        curAppointMin
                        ΑΝ ((diffMint >= 0) && (diffMint
                        <= 30)) ΤΟΤΕ
                            appoint = true
                            location =
                            curAppointLocation
                            startHour = curAppointHour
                            + ":" + curAppointMin
                            break
                    ΤΕΛΟΣ_ΑΝ
                ΑΛΛΙΩΣ ΑΝ (diffHour == 1) ΤΟΤΕ
                    diffMint = (59 - curAppointMin) +
                    selectedMin
                    ΑΝ ((diffMint >= 0) && (diffMint
                    <= 30)) ΤΟΤΕ
                        appoint = true
                        location =
                        curAppointLocation
                        startHour = curAppointHour
                        + ":" + curAppointMin
                        break
                    ΤΕΛΟΣ_ΑΝ
            ΤΕΛΟΣ_ΑΝ
        ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΑΝ
    ΤΕΛΟΣ_ΕΠΑΝΑΛΗΨΗΣ
ΤΕΛΟΣ_ΑΝ
    ΑΝ (appoint == true) ΤΟΤΕ
        return location + "" + startHour

```

```

ΑΛΛΙΩΣ
    return "no"
ΤΕΛΟΣ_ΑΝ
ΤΕΛΟΣ
ΤΕΛΟΣ

```

Ο αλγόριθμος στη Δραστηριότητα Χάρτη αποτελείται από το κύριο μέρος του αλγορίθμου και από τέσσερις διαφορετικές συναρτήσεις, οι οποίες βοηθάνε το κύριο πρόγραμμα για τη σωστή λειτουργία του. Το κύριο μέρος του αλγορίθμου ασχολείται με την εύρεση της τρέχουσας τοποθεσίας του χρήστη και την αποθήκευσή της για να μπορεί να χρησιμοποιηθεί στο μέλλον σαν τελευταία γνωστή τοποθεσία του χρήστη, στη περίπτωση που η τρέχουσα τοποθεσία δεν είναι διαθέσιμη. Επίσης, εμφανίζει στο χάρτη όλους τους διαθέσιμους σταθμούς, βενζινάδικα ή σούπερ μάρκετ, ανάλογα με τι έχει επιλέξει ο χρήστης να βρει. Ακόμα, αν ο χρήστης έχει επιλέξει από την προηγούμενη Δραστηριότητα Σταθμού ένα συγκεκριμένο σταθμό, τότε ο αλγόριθμος αυτός εμφανίζει τον συγκεκριμένο σταθμό στο χάρτη.

Όταν βρεθεί η τρέχουσα τοποθεσία του χρήστη, ο αλγόριθμος καλεί τη συνάρτηση `findNearest()` για να βρεθεί ο πλησιέστερος σταθμός. Επίσης, η συνάρτηση `findNearest()` καλείται και όταν ο χρήστης έχει επιλέξει να βρεθεί ο πλησιέστερος σταθμός με βάση τη τελευταία γνωστή τοποθεσία του χρήστη. Μόλις έχει βρεθεί ο πλησιέστερος σταθμός καλείται η συνάρτηση `navigateFromTo()`, η οποία δίνει οδηγίες στο χρήστη να πάει από τη τοποθεσία του στη τοποθεσία του πλησιέστερου σταθμού. Πιο συγκεκριμένα η συνάρτηση `navigateFromTo()` επικοινωνεί με τους Διακομιστές (Servers) της Google για να βρεθούν οι οδηγίες καθοδήγησης του χρήστη από την τοποθεσία του στη τοποθεσία του πλησιέστερου σταθμού και να εμφανιστούν στην οθόνη της εφαρμογής.

Η συνάρτηση `findNearest()` καλεί αρχικά μία άλλη συνάρτηση την `checkForAppointment()`, η οποία ελέγχει αν ο χρήστης έχει κάποιο ραντεβού την ώρα που πάτησε το κουμπί για να βρεθεί ο πλησιέστερος σταθμός. Δηλαδή, η συνάρτηση `checkForAppointment()` χρησιμοποιεί τη λίστα με τα ραντεβού του χρήστη και ελέγχει αν κάποιο από αυτά έχουν ώρα έναρξης μικρότερη του μισάωρου από την ώρα που επέλεξε ο χρήστης να βρει τον πλησιέστερο σταθμό. Ανάλογα με το αποτέλεσμα η συνάρτηση `checkForAppointment()` επιστρέφει αν υπάρχει ή όχι κάποιο ραντεβού.

Έπειτα, η συνάρτηση `findNearest()` ελέγχει για το αν υπάρχει κίνηση στους δρόμους και ενημερώνεται ο χρήστης με κατάλληλο μήνυμα. Στη συνέχεια, αν υπάρχει κάποιο ραντεβού ενημερώνεται ο χρήστης για το πιο ραντεβού έχει εκείνη την ώρα και σε πιο μέρος και η εφαρμογή ρωτάει τον χρήστη αν θέλει να βρεθεί ο πλησιέστερος σταθμός στη διεύθυνση του ραντεβού του. Αν ο χρήστης επιλέξει να μη βρεθεί ο πλησιέστερος σταθμός στη διεύθυνση του ραντεβού του, ο αλγόριθμος καλεί τη συνάρτηση `nearestMarker()` για να βρεθεί ο πλησιέστερος σταθμός στη τοποθεσία του χρήστη. Αλλιώς, ο αλγόριθμος, χρησιμοποιώντας τη συνάρτηση `nearestMarker()`, βρίσκει τον πλησιέστερο σταθμό στη διεύθυνση του ραντεβού του. Αν η διεύθυνση του ραντεβού του χρήστη δεν είναι διαθέσιμη για οποιοδήποτε λόγο, η συνάρτηση `nearestMarker()` βρίσκει τον πλησιέστερο σταθμό στη τοποθεσία του χρήστη.

Η συνάρτηση `nearestMarker()` χρησιμοποιεί τη λίστα με όλους τους σταθμούς και βρίσκει τον πλησιέστερο διαθέσιμο σταθμό για τον χρήστη. Αυτή η συνάρτηση επιστρέφει τις συντεταγμένες του πλησιέστερου σταθμού.

Για να υπολογίσουμε την πολυπλοκότητα αυτού του αλγορίθμου πρέπει να υπολογίσουμε την πολυπλοκότητα των αλγορίθμων των συναρτήσεων που τον απαρτίζουν, οπότε πρέπει να βρεθεί η πολυπλοκότητα των αλγορίθμων των συναρτήσεων `findNearest()` και `navigateFromTo()`. Όμως, επειδή η συνάρτηση `findNearest()` αποτελείται από επιμέρους συναρτήσεις, όπως τη συνάρτηση `checkForAppointment()` και τη `nearestMarker()`, πρέπει να βρεθεί η πολυπλοκότητα αυτών των αλγορίθμων των συναρτήσεων για να βρεθεί η πολυπλοκότητα του αλγορίθμου της συνάρτησης `findNearest()`.

Αρχίζοντας από τη συνάρτηση `navigateFromTo()` η πολυπλοκότητα του αλγορίθμου της είναι $f(n) = 1$, εφόσον δεν υπάρχει κάποια διαδικασία επανάληψης και ο αριθμός των οδηγιών που χρειάζεται είναι απλά μια σταθερά.

Στη συνέχεια, για τη συνάρτηση `findNearest()` υπολογίζεται η πολυπλοκότητα των αλγορίθμων των συναρτήσεων `checkForAppointment()` και `nearestMarker()`. Η πολυπλοκότητα του αλγορίθμου της συνάρτησης `checkForAppointment()` είναι $f(n) = n$, εφόσον υπάρχει τουλάχιστον μία διαδικασία επανάληψης, όπου n είναι το πλήθος των επαναλήψεων όλων των ραντεβού στη λίστα. Η πολυπλοκότητα του αλγορίθμου της συνάρτησης `nearestMarker()` είναι $f(n) = n$, εφόσον υπάρχουν δύο διαδικασίες επανάληψης οι οποίες όμως δεν είναι εμφωλευμένες, όπου n είναι το πλήθος των επαναλήψεων όλων των σταθμών (βενζινάδικων – σούπερ μάρκετ) στη λίστα για την πρώτη επανάληψη και όλων των σταθμών που βρίσκονται στο χάρτη για τη δεύτερη επανάληψη. Οπότε, η πολυπλοκότητα του αλγορίθμου της συνάρτησης `findNearest()` είναι $f(n) = n$, εφόσον οι επιμέρους συναρτήσεις δεν βρίσκονται μέσα σε κάποια διαδικασία επανάληψης και δεν είναι εμφωλευμένες.

Τέλος, η πολυπλοκότητα ολόκληρου του αλγορίθμου του προγράμματος της Δραστηριότητας Χάρτη είναι $f(n) = n^2 + n$. Αυτό συμβαίνει διότι η συνάρτηση `findNearest()` καλείται σε δύο διαφορετικές περιπτώσεις. Στην πρώτη περίπτωση, η συνάρτηση `findNearest()` καλείται μέσα σε μία διαδικασία επανάληψης, όπου προκύπτει το n^2 . Στη δεύτερη περίπτωση, η ίδια συνάρτηση καλείται επίσης εκτός κάποιας διαδικασίας επανάληψης, όπου προκύπτει το n . Ακόμα, η πολυπλοκότητα του αλγορίθμου της συνάρτησης `navigateFromTo()` είναι $f(n) = 1$, όπου έχουμε το προηγούμενο n από τη συνάρτηση `findNearest()` συν 1, δηλαδή $n + 1 = n$. Το τελικό αποτέλεσμα είναι $f(n) = n^2 + n$.

3 Σχέδιο Δράσης για την εκπόνηση της εργασίας

Σε αυτό το κεφάλαιο θα μιλήσουμε γενικά για το λειτουργικό σύστημα Android, για το State of the Art του Android και για το State of the Art της διαδικασίας ανάπτυξης εφαρμογών Android.

3.1 State of the Art

Ο όρος “State of the Art” αναφέρεται στο υψηλότερο επίπεδο της γενικής ανάπτυξης μίας συσκευής, μιας τεχνικής ή ενός επιστημονικού πεδίου που επιτυγχάνεται σε μία συγκεκριμένη στιγμή. Επίσης, αναφέρεται στο επίπεδο της ανάπτυξης μιας συσκευής, διαδικασίας, μεθόδου, τεχνικής ή στην επιστήμη που επιτεύχθηκε σε συγκεκριμένο χρόνο χρησιμοποιώντας κοινές μεθοδολογίες. Ουσιαστικά, το State of the Art είναι μία μέθοδος που χρησιμοποιήθηκε για την δημιουργία ενός προϊόντος, μίας συσκευής, μίας τεχνικής, μιας διαδικασίας σε τεχνολογικό επίπεδο, στην επιστήμη, στη διαδικασία παραγωγής τροφίμων και σε πολλούς άλλους τομείς, η οποία μέθοδος αυτή αποτελεί την τελευταία λέξη της τεχνολογίας ή την τεχνολογία αιχμής.⁽²⁾

3.1.1 Android

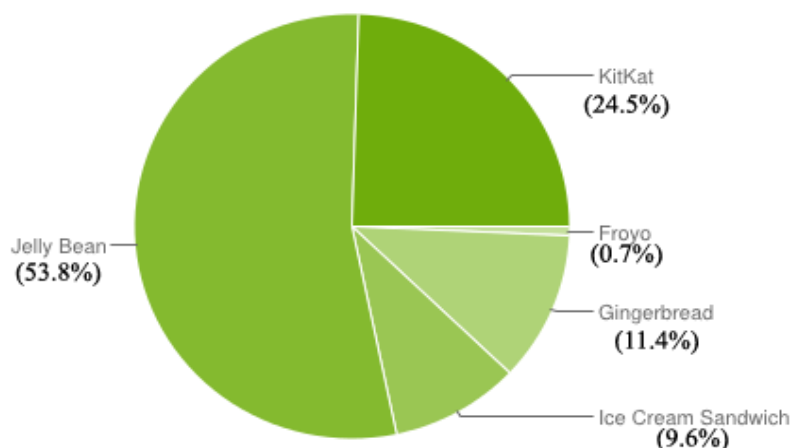
Το Android είναι ένα λειτουργικό σύστημα για έξυπνες κινητές συσκευές το οποίο βασίζεται στον πυρήνα του λειτουργικού Linux και αναπτύσσεται σήμερα από τη Google. Η Διεπαφή Χρήστη (User Interface) του Android βασίζεται κυρίως σε συσκευές με οθόνες αφής, όπως έξυπνες κινητές συσκευές (smartphones) και υπολογιστές ταμπλέτες (tablets). Επίσης, το Android χρησιμοποιείται και σε άλλες συσκευές, όπως τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear), όπου η Διεπαφή Χρήστη (User Interface) του Android έχει προσαρμοστεί ειδικά γι’ αυτές.

Το λειτουργικό σύστημα Android χρησιμοποιεί εισροές αφής που αντιστοιχούν σε πραγματικές δράσεις, όπως το σύρσιμο του δαχτύλου στην οθόνη (swiping), άγγιγμα του δαχτύλου στην οθόνη (tapping), σύρσιμο δύο δαχτύλων ταυτόχρονα στην οθόνη προς τα μέσα αλλά και προς τα έξω (pinching, reverse pinching) για να χειραγωγηθούν τα αντικείμενα στην οθόνη, επίσης χρησιμοποιείται και ένα εικονικό πληκτρολόγιο. Παρά το γεγονός ότι το Android έχει σχεδιαστεί κυρίως για συσκευές με οθόνες αφής, έχει επίσης χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, σταθερούς υπολογιστές και άλλα ηλεκτρονικά είδη.

Ο πηγαίος κώδικας κάθε έκδοσης του Android δημοσιεύεται από την Google βάσει των αδειών ανοικτού κώδικα, αν και οι περισσότερες συσκευές Android κυκλοφορούν από τις κατασκευάστριες εταιρίες με ένα συνδυασμό ανοικτού και κλειστού λογισμικού. Το Android αναπτύχθηκε αρχικά από την εταιρία Android Inc., η οποία αργότερα υποστηρίχθηκε οικονομικά από την Google και τελικά αγοράστηκε το 2005 από αυτήν. Το Android παρουσιάστηκε το 2007 μαζί με την ίδρυση του οργανισμού Open Handset Alliance, μια κοινοπραξία υλικού, λογισμικού, και εταιρειών τηλεπικοινωνιών που διατίθενται για την προώθηση ανοιχτών προτύπων για τις συσκευές κινητής τηλεφωνίας.

Το Android είναι δημοφιλής με τις εταιρείες τεχνολογίας οι οποίες απαιτούν ένα έτοιμο, χαμηλού κόστους και προσαρμόσιμο λειτουργικό σύστημα για συσκευές υψηλής τεχνολογίας. Ακόμα, όσον αφορά τα έξυπνα κινητά τηλέφωνα, το Android είναι ένα από τα πιο ευρέως χρησιμοποιούμενα λειτουργικά συστήματα.

Στο παρακάτω γράφημα 1, το οποίο έχει αναρτηθεί από την Google στην επίσημη ιστοσελίδα του Android, φαίνονται πληροφορίες για τις εκδόσεις του λογισμικού Android που χρησιμοποιούνται από τους χρήστες σε μία περίοδο έρευνας επτά ημερών, η οποία έρευνα πραγματοποιείται με βάση τις συσκευές που έχουν πρόσβαση στο Google Play Store. Το Google Play Store είναι μία εφαρμογή – υπηρεσία για τις συσκευές με λειτουργικό σύστημα Android η οποία επιτρέπει στους χρήστες να κατεβάζουν άλλες εφαρμογές, παιχνίδια, μουσική, βιβλία, περιοδικά, ταινίες, και τηλεοπτικά προγράμματα.



Γράφημα 1 – Πληροφορίες εκδόσεων Android που χρησιμοποιούνται

Όπως βλέπουμε από το γράφημα 1, οι εκδόσεις Android Ice Cream Sandwich, Jelly Bean και KitKat καταλαμβάνουν το 87.9 % των εκδόσεων που χρησιμοποιούνται από τους χρήστες αυτού του λειτουργικού συστήματος, οι οποίες είναι οι πιο πρόσφατες εκδόσεις, γι' αυτό το λόγο επιλέξαμε η εφαρμογή μας να είναι συμβατή με αυτές τις εκδόσεις Android.

Αυτή τη στιγμή, το Android χρησιμοποιεί το Dalvik ως περιβάλλον εκτέλεσης (Runtime Environment) για να εκτελούνται οι εφαρμογές της συσκευής. Το Dalvik Runtime Environment είναι μία εικονική μηχανή (Virtual Machine) στο Android το οποίο αναλαμβάνει τη διαδικασία εκτέλεσης εφαρμογών που είναι γραμμένες για το Android. Αυτό έχει σαν αποτέλεσμα το Dalvik να είναι αναπόσπαστο κομμάτι της στοίβας λογισμικού του Android. Το Dalvik χρησιμοποιεί τη σύνταξη Just-in-time (JIT compilation) η οποία είναι μία σύνταξη που γίνεται κατά τη διάρκεια της εκτέλεσης και λειτουργίας μιας εφαρμογής. Επίσης, το Dalvik είναι σχεδιασμένο για συσκευές που περιορίζονται όσον αφορά τη μνήμη και την ταχύτητα του επεξεργαστή. Τέλος, το Dalvik χρησιμοποιείται σε φορητές συσκευές όπως τα έξυπνα κινητά τηλέφωνα, τους υπολογιστές ταμπλέτες (tablets), έξυπνες τηλεοράσεις και ρολόγια χειρός.

Όμως, State of the Art για το Android είναι το περιβάλλον εκτέλεσης Android Runtime (ART) το οποίο είναι ένα εναλλακτικό περιβάλλον εκτέλεσης για το Android έναντι του Dalvik. Το ART συμπεριλήφθηκε αρχικά στην έκδοση Android 4.4 "KitKat" ως επιλογή για προγραμματιστές μόνο και όχι για καθημερινή χρήση καθώς είναι ακόμα υπό ανάπτυξη. Το Android Runtime (ART) θα αντικαταστήσει πλήρως το Dalvik περιβάλλον εκτέλεσης στην επόμενη έκδοση του Android η οποία θα είναι η Android 5.0 "Lollipop".

Το Android Runtime (ART) εισάγει τη χρήση της Ahead-Of-Time (AOT) σύνταξης η οποία πραγματοποιείται κατά την εγκατάσταση μιας εφαρμογής. Με αυτό τον τρόπο μειώνεται το συνολικό ποσό σύνταξης που πρέπει να εκτελεστεί κατά τη διάρκεια λειτουργίας μιας εφαρμογής και κατά συνέπεια μειώνεται η χρήση του επεξεργαστή της συσκευής και βελτιώνεται ο χρόνος λειτουργίας της μπαταρίας της. Επίσης, το Android Runtime (ART) φέρνει βελτιώσεις στις επιδόσεις μιας συσκευής, στη συλλογή σκουπιδιών (garbage collection) και στον εντοπισμό σφαλμάτων μιας εφαρμογής. Το μειονέκτημα του Android Runtime (ART) είναι ότι οι εφαρμογές θα χρειάζονται μεγαλύτερο αποθηκευτικό χώρο στη συσκευή και περισσότερο χρόνο εγκατάστασης.

Όσον αφορά την ανάπτυξη εφαρμογών για το λειτουργικό σύστημα Android υπάρχουν πολλοί και διάφοροι τρόποι, όπως προαναφερθήκαμε στο Κεφάλαιο 2. Όμως, το State of the Art στη διαδικασία δημιουργίας εφαρμογών Android είναι το Android Studio. Το Android Studio έχει πολλά πλεονεκτήματα έναντι του Eclipse ADT with Android SDK και των υπόλοιπων IDE, όμως είναι ακόμα σε δοκιμαστικό στάδιο και ορισμένες λειτουργίες δεν έχουν ακόμη εφαρμοστεί, αυτό έχει σαν αποτέλεσμα να υπάρχει αρκετά μεγάλη πιθανότητα να προκύπτουν ενίοτε σφάλματα κατά τη διάρκεια δημιουργίας εφαρμογών από τους προγραμματιστές.

Όμως, υπάρχει και μία διαφορετική άποψη πάνω στο θέμα State of the Art για την ανάπτυξη εφαρμογών έξυπνων συσκευών γενικά. Όπως προαναφερθήκαμε, το Android Studio αποτελεί το State of the Art για την ανάπτυξη εφαρμογών ειδικά για το λειτουργικό σύστημα Android. Παρ' όλα αυτά, υπάρχουν μέθοδοι για την δημιουργία εφαρμογών, οι οποίες εφαρμογές μπορούν να εγκατασταθούν σε οποιαδήποτε έξυπνη συσκευή ανεξαρτήτως λειτουργικού συστήματος. Μία τέτοια μέθοδος είναι η χρήση των γλωσσών HTML, JavaScript και CSS.

Η HTML (HyperText Markup Language) είναι η τυπική γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία ιστοσελίδων. Μπορεί να ενσωματώσει σενάρια (scripts) γραμμένα σε γλώσσες προγραμματισμού όπως η JavaScript, που επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML.

Η JavaScript είναι μια δυναμική γλώσσα προγραμματισμού ηλεκτρονικών υπολογιστών. Χρησιμοποιείται συνήθως ως μέρος των προγραμμάτων περιήγησης Ιστού (Web Browsers), των οποίων οι υλοποιήσεις επιτρέπουν τα σενάρια από την πλευρά του πελάτη (client-side scripts) να επικοινωνούν με τον χρήστη. Επίσης, χρησιμοποιείται για τον προγραμματισμό δικτύων από την πλευρά του Διακομιστή (server-side), για την ανάπτυξη παιχνιδιών και τη δημιουργία εφαρμογών για σταθερούς ηλεκτρονικού υπολογιστές αλλά και για έξυπνες συσκευές.

Η CSS (Cascading Style Sheets) είναι μια γλώσσα που χρησιμοποιείται για την περιγραφή της εμφάνιση και τη μορφοποίηση ενός εγγράφου γραμμένο σε μια γλώσσα σήμανσης όπως η HTML. Δηλαδή, η CSS χρησιμοποιείται για να αλλαχθεί το στυλ των ιστοσελίδων καθώς και της Διεπαφής Χρήστη (User Interface) που είναι γραμμένα σε HTML. Μαζί με την HTML και τη JavaScript, η CSS είναι μία θεμελιώδης τεχνολογία που χρησιμοποιείται από τις περισσότερες ιστοσελίδες για την δημιουργία οπτικά ελκυστικών ιστοσελίδων, για τη δημιουργία Διεπαφής Χρήστη για δικτυακές εφαρμογές αλλά και για εφαρμογές έξυπνων συσκευών.

Εν ολίγοις, με τη χρήση των γλωσσών HTML, JavaScript και CSS μπορούμε να δημιουργήσουμε εφαρμογές οι οποίες να είναι συμβατές για όλες τις έξυπνες συσκευές ανεξαρτήτου λειτουργικού συστήματος. Για την δημιουργία τέτοιων εφαρμογών μπορούμε να χρησιμοποιήσουμε ένα IDE όπως το Intel XDK ή ένα πλαίσιο (framework) ανάπτυξης εφαρμογών έξυπνων κινητών όπως το PhoneGap, τα οποία έχουμε αναλύσει προηγουμένως στο Κεφάλαιο 2 και μπορούν να θεωρηθούν State of the Art για την ανάπτυξη εφαρμογών έξυπνων συσκευών.^{(1), (2)}

4 Κύριο μέρος Πτυχιακής

Το κεφάλαιο αυτό ασχολείται με το κύριο μέρος της συγκεκριμένης πτυχιακής εργασίας. Το κύριο μέρος της πτυχιακής εργασίας εμπεριέχει την ανάλυση του προβλήματος, όπου πραγματοποιείται η περιγραφή του προβλήματος που πρέπει να αντιμετωπισθεί από τη μεριά του χρήστη και του προγραμματιστή. Επίσης, γίνεται αναφορά στον τρόπο με τον οποίο σχεδιάστηκε η υλοποίηση της εφαρμογής, την αρχιτεκτονική που ακολουθήθηκε αλλά και την υλοποίησή της.

4.1 Ανάλυση Προβλήματος

Για τον χρήστη, το πρόβλημα που υπάρχει είναι η μη ύπαρξη μίας ενιαίας εφαρμογής στο οικοσύστημα του λογισμικού συστήματος Android η οποία να διαχειρίζεται τις καθημερινές του δραστηριότητες. Τέτοιες δραστηριότητες είναι η διαχείριση των ραντεβού, η εύρεση του κοντινότερου βενζινάδικου ή σούπερ μάρκετ ανάλογα με την τοποθεσία του χρήστη, την ώρα αιχμής, την ώρα καταστημάτων, την ώρα και την τοποθεσία των ραντεβού του χρήστη και η δρομολόγηση του χρήστη σε αυτό.

Για τον προγραμματιστή, το πρόβλημα που έχει να αντιμετωπίσει είναι η δημιουργία μίας τέτοιας εφαρμογής η οποία θα καλύπτει τις ανάγκες των χρηστών. Για την ανάπτυξη μιας τέτοιας εφαρμογής Android χρειάζονται για αρχή βασικές γνώσεις προγραμματισμού. Έπειτα, είναι απαραίτητο η εκμάθηση προγραμματισμού σε περιβάλλον Android. Σε αυτό το σημείο, αν η γλώσσα προγραμματισμού JAVA είναι ήδη γνωστή τότε η εκμάθηση προγραμματισμού σε περιβάλλον Android είναι πιο εύκολη, μιας και ο προγραμματισμός σε Android βασίζεται πάνω στην γλώσσα προγραμματισμού JAVA.

Έπειτα, είναι απαραίτητο η εκμάθηση των γλωσσών προγραμματισμού PHP και SQL. Αυτές οι γλώσσες προγραμματισμού χρειάζονται για την επικοινωνία της εφαρμογής Android με ένα Server και κατά συνέπεια με μία Βάση Δεδομένων. Τέλος, η γλώσσα σήμανσης XML είναι απαραίτητη για την διαχείριση αρχείων XML από την εφαρμογή Android αλλά και από τον Server.

Όσον αφορά την εφαρμογή, θα μπορούν να την χρησιμοποιούν εγγεγραμμένοι και μη-εγγεγραμμένοι χρήστες σε αυτήν, θα πρέπει να είναι εύκολα κατανοητή και διαχειρίσιμη από τον χρήστη και η Διεπαφή Χρήστη (User Interface) θα πρέπει να είναι απλή για να μη μπερδεύονται άσκοπα οι χρήστες. Επίσης, η εφαρμογή θα πρέπει να επικοινωνεί με ένα Server για την ανανέωση των πληροφοριών των βενζινάδικων και των σούπερ μάρκετ, θα πρέπει ακόμα να επικοινωνεί με μία Βάση Δεδομένων για την εγγραφή νέων χρηστών αλλά και την διαχείριση των υπαρχόντων και τέλος θα πρέπει να διαχειρίζεται τα ραντεβού του χρήστη και με βάση αυτά να βρίσκει και να δρομολογεί τον χρήστη στο κοντινότερο σούπερ μάρκετ – βενζινάδικο.

4.1.1 Απαιτήσεις Συστήματος

Οι απαιτήσεις συστήματος ενός ηλεκτρονικού υπολογιστή για την δημιουργία εφαρμογών Android χρησιμοποιώντας το Eclipse ADT with Android SDK φαίνονται στον παρακάτω πίνακα 2, όπως περιγράφονται από την επίσημη ιστοσελίδα της Google για το κατέβασμα του λογισμικού Eclipse ADT with Android SDK.

Λειτουργικό Σύστημα	Απαιτήσεις Συστήματος	Εργαλεία Ανάπτυξης
Windows	XP (32-bit), Vista (32- or 64-bit), 7 (32- or 64-bit), 8 (32- or 64-bit)	JDK 6, Apache Ant 1.8 ή νεότερη έκδοση
Mac OS X	10.8.5 ή νεότερη έκδοση	JDK 6, Apache Ant 1.8 ή νεότερη έκδοση

Linux	64-bit διανομή ικανό να εκτελεί 32-bit εφαρμογές, GNU C Library (glibc) 2.11 ή νεότερη έκδοση, Tested on Ubuntu 12.04, Precise Pangolin	JDK 6, Apache Ant 1.8 ή νεότερη έκδοση
--------------	---	---

Πίνακας 2 – Απαιτήσεις Συστήματος Ηλεκτρονικών Υπολογιστών

Οι απαιτήσεις συστήματος ενός έξυπνου κινητού τηλεφώνου με λειτουργικό σύστημα Android για να μπορεί να τρέξει μία τέτοια εφαρμογή όπως αυτή που αναλύουμε στη συγκεκριμένη πτυχιακή εργασία φαίνονται στον παρακάτω πίνακα 3.

Χαρακτηριστικά	Ελάχιστες Απαιτήσεις Συστήματος	Συνηστώμενες Απαιτήσεις Συστήματος
Επεξεργαστής	1 GHz	1 GHz
Μνήμη RAM	512 MB	1 GB
Android	4.0.0 έκδοση	4.4.2 ή νεότερη έκδοση

Πίνακας 3 – Απαιτήσεις Συστήματος Έξυπνου Κινητού

Επειδή κάθε εφαρμογή Android έχει διαφορετικές απαιτήσεις συστήματος έξυπνου κινητού, οι συγκεκριμένες απαιτήσεις που περιγράφονται στον πίνακα 3 αφορούν την εφαρμογή που περιγράφουμε σε αυτή τη πτυχιακή εργασία.⁽¹⁾

4.2 Σχεδιασμός Υλοποίησης

Αρχικά, για να αρχίσουμε να σχεδιάζουμε και να υλοποιούμε εφαρμογές Android, ποιο συγκεκριμένα την εφαρμογή που περιγράφεται σε αυτή τη πτυχιακή εργασία, πρέπει να έχουμε έναν ηλεκτρονικό υπολογιστή ο οποίος να καλύπτει τις απαιτήσεις συστήματος όπου περιγράφηκαν στο κεφάλαιο 4.1 πίνακας 2. Έπειτα, πρέπει να δημιουργήσουμε τις κατάλληλες συνθήκες για να έχουμε την δυνατότητα δημιουργίας εφαρμογών Android.

Στη περίπτωση αυτή, όπου το λειτουργικό σύστημα του ηλεκτρονικού υπολογιστή μας είναι Windows 8 64-bit, κατεβάζουμε και εγκαθιστούμε τα εργαλεία ανάπτυξης JDK 6 και Apache Ant 1.8 ή νεότερες εκδόσεις τους. Εφόσον τα δύο παραπάνω εργαλεία ανάπτυξης έχουν εγκατασταθεί σωστά και για το JDK 6 έχουμε ορίσει σωστά τις μεταβλητές περιβάλλοντος, τότε κατεβάζουμε το Eclipse ADT with Android SDK από την επίσημη ιστοσελίδα της Google.

Μετά την επιτυχημένη εγκατάσταση του Eclipse ADT with Android SDK είμαστε πλέον έτοιμοι να αρχίσουμε να προγραμματίζουμε σε περιβάλλον Android. Όμως, επειδή η εφαρμογή η δική μας πρέπει να επικοινωνεί με έναν Server, χρειαζόμαστε ένα Όνομα Τομέα (Domain Name) και ένα μέρος το οποίο θα φιλοξενεί τα απαραίτητα αρχεία που χρειάζεται η εφαρμογή μας (Hosting). Εφόσον έχουμε καταχωρήσει ένα Όνομα Τομέα (Domain Name) και ένα μέρος φιλοξενίας των αρχείων μας (Hosting), ο Διακομιστής (Server) είναι έτοιμος για την αποθήκευση των απαραίτητων αρχείων και διαχείριση της Βάσης Δεδομένων των χρηστών.

Από τη στιγμή που ο ηλεκτρονικός υπολογιστής μας είναι έτοιμος για την δημιουργία μιας τέτοιας εφαρμογής, το μόνο που μας χρειάζεται πλέον είναι ένα έξυπνο κινητό τηλέφωνο με λειτουργικό σύστημα Android, το οποίο όμως να έχει τις απαραίτητες απαιτήσεις συστήματος όπως περιγράφονται στον πίνακα 3 του κεφαλαίου 4.1, για να έχουμε τη δυνατότητα να ελέγχουμε κάθε φορά την εφαρμογή μας πάνω σε αυτό.

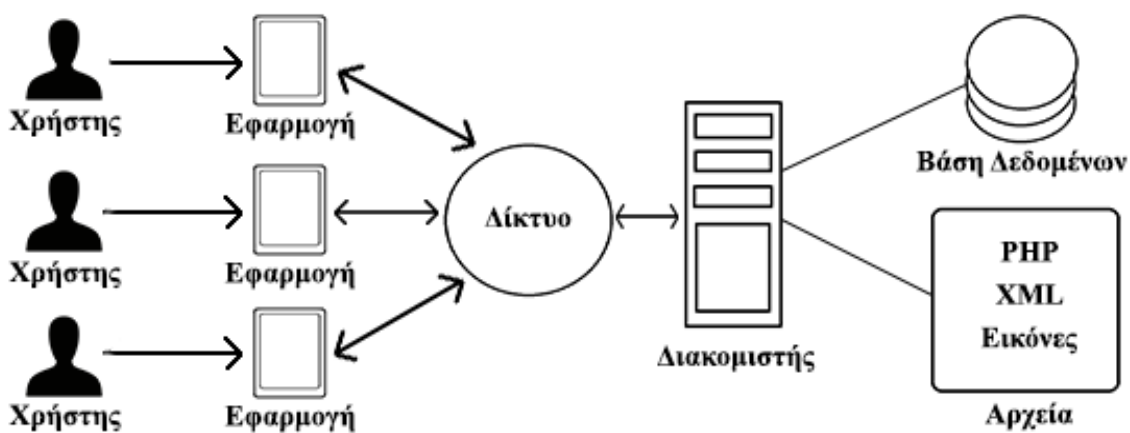
Όμως, η κατοχή ενός έξυπνου κινητού τηλεφώνου με λειτουργικό σύστημα Android δεν είναι απαραίτητο για τον έλεγχο της εφαρμογής μας, αφού το Eclipse ADT with Android SDK μας παρέχει έναν προσομοιωτή ενός τέτοιου κινητού τηλεφώνου στον οποίο μπορούμε να αλλάξουμε τα χαρακτηριστικά του έτσι ώστε κάθε φορά να προσομοιώνουμε ένα κινητό τηλέφωνο με διαφορετικά χαρακτηριστικά. Με αυτό τον τρόπο μπορούμε να προσομοιώσουμε ένα κινητό τηλέφωνο το οποίο να καλύπτει τις ελάχιστες απαιτήσεις συστήματος και ένα δεύτερο που να καλύπτει τις συνηστώμενες

απαιτήσεις συστήματος. Το αρνητικό του προσομοιωτή είναι ότι είναι πολύ αργός, ενώ με το έξυπνο κινητό τηλέφωνο μπορούμε να ελέγχουμε την εφαρμογή μας με πολύ πιο γρήγορο ρυθμό.

Τέλος, εφόσον ο ηλεκτρονικός υπολογιστής και το έξυπνο τηλέφωνο ή ο προσομοιωτής είναι έτοιμα για την δημιουργία της συγκεκριμένης εφαρμογής, πρέπει να σκεφτούμε για το πώς ακριβώς θα δουλεύει η εφαρμογή μας, την αρχιτεκτονική της και έπειτα να περάσουμε στην υλοποίησή της.

4.3 Αρχιτεκτονική

Αρχικά, ο χρήστης αλληλοεπιδρά μόνο με τη Διεπαφή Χρήστη (User Interface) της εφαρμογής. Έπειτα η εφαρμογή, ανάλογα με τις πληροφορίες που της έχει δώσει ο χρήστης μέσω της Διεπαφής Χρήστη (User Interface), επικοινωνεί με τον Εξυπηρετητή (Server) και με την Βάση Δεδομένων. Αυτή η αλληλεπίδραση φαίνεται στην παρακάτω εικόνα 1. Με βάση αυτή την εικόνα, αναλύεται παρακάτω πώς δουλεύει η εφαρμογή σε κάθε περίπτωση.



Εικόνα 1 – Αρχιτεκτονική Εφαρμογής

Στην περίπτωση που ο χρήστης θέλει να δημιουργήσει λογαριασμό στην εφαρμογή, συμπληρώνει τα απαραίτητα πεδία στην φόρμα εγγραφής που του παρέχει η εφαρμογή. Έπειτα, η εφαρμογή δοκιμάζει αν υπάρχει δίκτυο στην συσκευή και αν αυτή μπορεί να συνδεθεί στο διαδίκτυο. Εφόσον η σύνδεση στο δίκτυο είναι εφικτή, η εφαρμογή επικοινωνεί με τον Διακομιστή και στη συνέχεια με την Βάση Δεδομένων. Εκεί γίνεται ο έλεγχος αν τα στοιχεία που εισήγαγε ο χρήστης κατά τη διαδικασία συμπλήρωσης της φόρμας εγγραφής είναι σωστά. Δηλαδή, ελέγχει αν το e-mail του είναι έγκυρο και αν είναι ήδη κατοχυρωμένο, ακόμα ελέγχει αν το όνομα χρήστη είναι ήδη κατοχυρωμένο από κάποιον άλλον χρήστη.

Ανάλογα με τα αποτελέσματα, ο Διακομιστής επικοινωνεί με την εφαρμογή και εμφανίζεται το κατάλληλο μήνυμα στην οθόνη της συσκευής. Ο χρήστης βλέποντας το μήνυμα πράττει αναλόγως δηλαδή, αν η εγγραφή του δεν ήταν επιτυχής τότε διορθώνει τα λάθη που επισημαίνονται στην οθόνη της συσκευής και προσπαθεί ξανά. Αν η εγγραφή του ήταν επιτυχής τότε ο χρήστης έχει πλέον δημιουργήσει επιτυχώς τον λογαριασμό του στην εφαρμογή.

Αν ο χρήστης επιλέξει να συνδεθεί στον λογαριασμό του, τότε συμπληρώνει τα απαραίτητα πεδία στην φόρμα σύνδεσης που του παρέχει η εφαρμογή. Στη συνέχεια, η εφαρμογή ελέγχει αν μπορεί να επικοινωνήσει με τον Διακομιστή. Αν η επικοινωνία είναι εφικτή γίνεται έλεγχος στη Βάση Δεδομένων για το αν το όνομα χρήστη και ο κωδικός πρόσβασης που έδωσε νωρίτερα ο χρήστης ταιριάζουν με αυτά που είναι αποθηκευμένα σε αυτή. Έπειτα, ο Διακομιστής επικοινωνεί με την εφαρμογή και εμφανίζεται το κατάλληλο μήνυμα στην οθόνη της συσκευής. Αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά τότε ο χρήστης έχει συνδεθεί επιτυχώς στον λογαριασμό του και εμφανίζεται το κατάλληλο μήνυμα στην οθόνη της συσκευής. Αν τα στοιχεία είναι λανθασμένα τότε ο χρήστης τα διορθώνει και προσπαθεί ξανά.

Ακόμα, ο χρήστης μπορεί να ζητήσει ένα νέο κωδικό πρόσβασης από την εφαρμογή, στη περίπτωση που έχει ξεχάσει το δικό του. Εφόσον ο χρήστης έχει εισάγει τη διεύθυνση email του, η οποία είναι απαραίτητη, η εφαρμογή επικοινωνεί με τον Διακομιστή και αν το email υπάρχει στη Βάση Δεδομένων τότε ο Διακομιστής δημιουργεί ένα νέο κωδικό πρόσβασης για το χρήστη και τον στέλνει στη διεύθυνση email του. Έπειτα ο χρήστης πρέπει να εισέλθει στο email του για να δει τον νέο κωδικό πρόσβασης έτσι ώστε να μπορεί να εισέλθει στο λογαριασμό του στην εφαρμογή, από όπου αργότερα μπορεί να αλλάξει τον κωδικό του.

Ο χρήστης έχει επίσης τη δυνατότητα να τροποποιήσει κάποια στοιχεία του λογαριασμού του. Εφόσον ο χρήστης τροποποιήσει τα στοιχεία του λογαριασμού του όπως εκείνος θέλει, μέσω της φόρμας τροποποίησης που του παρέχει η εφαρμογή, η εφαρμογή επικοινωνεί με τον Διακομιστή και τη Βάση Δεδομένων, αν και μόνο αν η επικοινωνία είναι εφικτή, δηλαδή η εφαρμογή έχει πρόσβαση στο δίκτυο. Έπειτα, ο Διακομιστής επικοινωνεί με την εφαρμογή και εμφανίζεται το κατάλληλο μήνυμα στην οθόνη της συσκευής.

Αν ο χρήστης επιλέξει από την εφαρμογή να του βρει το κοντινότερο βενζινάδικο ή σούπερ μάρκετ τότε η εφαρμογή βλέπει αν υπάρχουν τα απαραίτητα αρχεία αποθηκευμένα στο κινητό. Αν τα αρχεία αυτά δεν υπάρχουν ή αν ο χρήστης επιλέξει να τα ανανεώσει, τότε η εφαρμογή επικοινωνεί με το Διακομιστή, εφόσον η συσκευή μπορεί να συνδεθεί στο δίκτυο, και του ζητάει τα αρχεία αυτά, τα οποία είναι αποθηκευμένα σε μορφή XML. Με τη σειρά του ο Διακομιστής στέλνει τα αρχεία που ζήτησε η εφαρμογή στην συσκευή και πλέον η εφαρμογή μπορεί να χρησιμοποιήσει τα συγκεκριμένα αρχεία για να βρει το κοντινότερο βενζινάδικο - σούπερ μάρκετ.

Κατά τη διάρκεια του υπολογισμού του κοντινότερου βενζινάδικου – σούπερ μάρκετ, η εφαρμογή λαμβάνει υπόψην της αν ο χρήστης έχει εισάγει ραντεβού στην εφαρμογή και αν η ώρα κάποιου ραντεβού από αυτά που έχει εισάγει συμπίπτει με την ώρα που επέλεξε ο χρήστη να βρει το κοντινότερο βενζινάδικο ή σούπερ μάρκετ. Ακόμα, στην έρευνα αυτή, η εφαρμογή λαμβάνει υπόψην της την ώρα που ο χρήστης επέλεξε να γίνει αυτή η έρευνα, την ώρα αιχμής, την ώρα καταστημάτων, την τοποθεσία του χρήστη αλλά και την τοποθεσία του ραντεβού του.

Τέλος, η εφαρμογή δίνει την δυνατότητα στον χρήστη να προσθέσει ραντεβού σε αυτήν, να τα τροποποιήσει και να τα διαγράψει.

4.4 Υλοποίηση

4.4.1 Προετοιμασία Ηλεκτρονικού Υπολογιστή

Αρχικά, πρέπει να ρυθμίσουμε τον ηλεκτρονικό υπολογιστή μας έτσι ώστε να μπορεί να υποστηρίξει την δημιουργία εφαρμογών Android. Για αρχή, πραγματοποιούμε την εγκατάσταση της πιο πρόσφατης έκδοσης του εργαλείου ανάπτυξης Java JDK μεταβαίνοντας στην ιστοσελίδα <http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html> και κατεβάζοντας το κατάλληλο αρχείο ανάλογα με την έκδοση του λειτουργικού συστήματος που έχει ο ηλεκτρονικός υπολογιστής μας. Στην περίπτωση αυτή κατεβάζουμε το αρχείο για το λειτουργικό σύστημα Windows 32-bit, εφόσον έχουμε αποδεχτεί τους όρους χρήσης, όπως φαίνεται στις παρακάτω εικόνες 2 και 3.


ORACLE® Sign In/Register Help Country Communities I am a... I want to... Search

Products Solutions Downloads Store Support Training Partners About OTN

Oracle Technology Network > Articles > Java Platform, Standard Edition


- Application Development Framework
- Application Express
- Big Data
- Business Intelligence
- Cloud Computing
- Communications
- Database Performance & Availability
- Data Warehousing
- Database
- .NET
- Dynamic Scripting Languages
- Embedded
- Digital Experience
- Enterprise Architecture
- Enterprise Management

Java SE Downloads



Java Platform (JDK) 8u25

[DOWNLOAD](#)



JDK 8u25 & NetBeans 8.0.1

[DOWNLOAD](#)

Java Platform, Standard Edition

Java SE 8u25
 This release includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release.
[Learn more](#)

• [Installation Instructions](#)
[JDK](#)

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)

Εικόνα 2 – Εγκατάσταση του JDK No.1

Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#) (tick the checkbox under Subscription Center > Oracle Technology News)
- [Java Developer Day hands-on workshops](#) (free) and other events
- [Java Magazine](#)

JDK MD5 Checksum












Looking for JDK 8 on ARM?

JDK 8 for ARM downloads have moved to the [JDK 8 for ARM download page](#).

Java SE Development Kit 8u25

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the [Oracle Binary Code License Agreement for Java SE](#); you may now download this software.

Product / File Description	File Size	Download
Linux x86	135.24 MB	 jdk-8u25-linux-i586.rpm
Linux x86	154.88 MB	 jdk-8u25-linux-i586.tar.gz
Linux x64	135.6 MB	 jdk-8u25-linux-x64.rpm
Linux x64	153.42 MB	 jdk-8u25-linux-x64.tar.gz
Mac OS X x64	209.13 MB	 jdk-8u25-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	137.01 MB	 jdk-8u25-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	97.14 MB	 jdk-8u25-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	137.11 MB	 jdk-8u25-solaris-x64.tar.Z
Solaris x64	94.24 MB	 jdk-8u25-solaris-x64.tar.gz
Windows x86	157.26 MB	 jdk-8u25-windows-i586.exe
Windows x64	169.62 MB	 jdk-8u25-windows-x64.exe

Εικόνα 3 – Εγκατάσταση του JDK No.2

Έπειτα πραγματοποιούμε την εγκατάσταση του αρχείου τρέχοντας το συγκεκριμένο αρχείο και επιλέγοντας τι ακριβώς θέλουμε να εγκαταστήσουμε και πού, όπως φαίνεται στις παρακάτω εικόνες 4, 5, 6, 7 και 8.



Εικόνα 4– Εγκατάσταση του JDK No.3



Εικόνα 5 – Εγκατάσταση του JDK No.4



Εικόνα 6 – Εγκατάσταση του JDK Νο.5

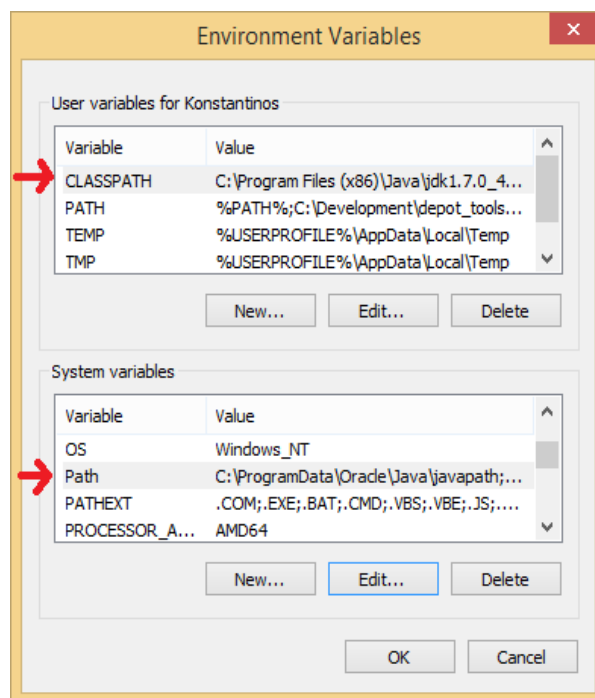


Εικόνα 7 – Εγκατάσταση του JDK Νο.6



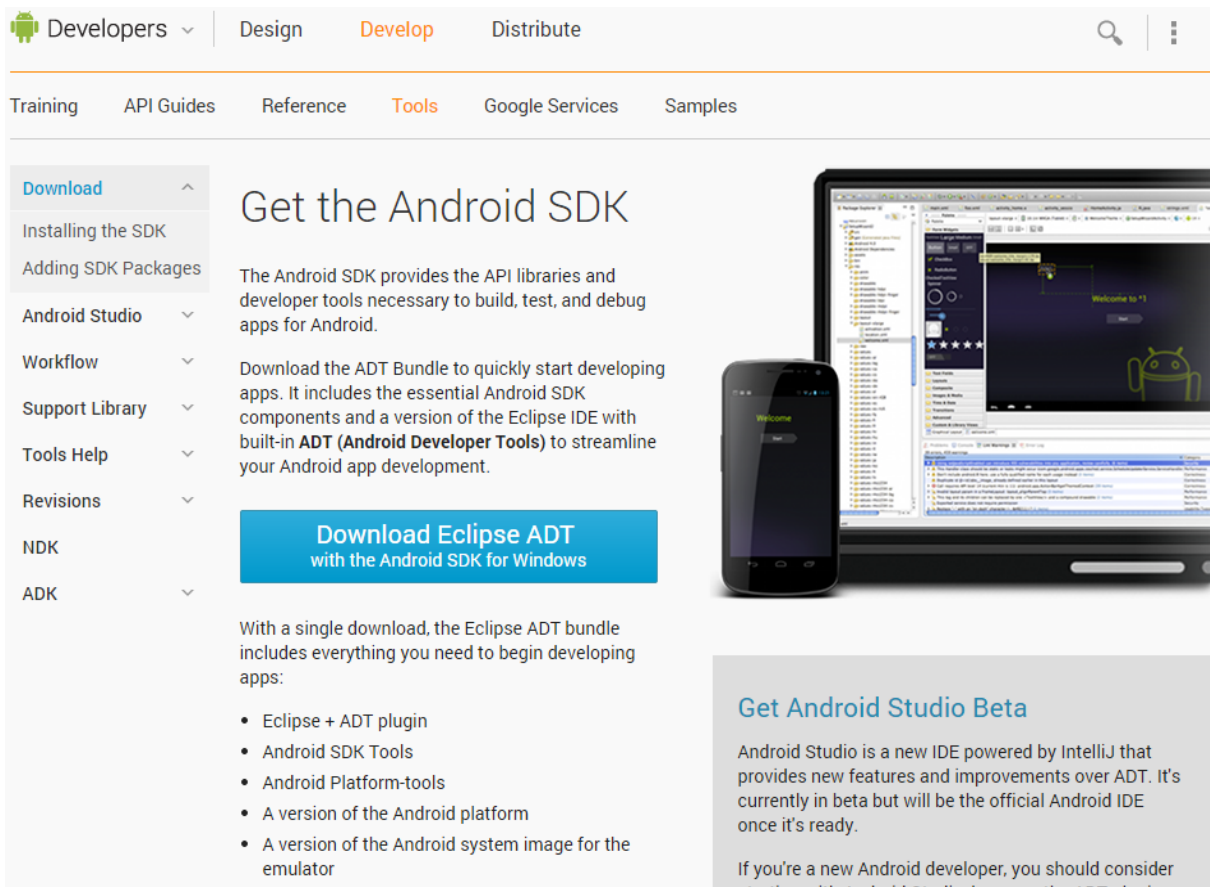
Εικόνα 8 – Εγκατάσταση του JDK No.7

Αφού ολοκληρωθεί η εγκατάσταση του JDK επιτυχώς, πρέπει να ορίσουμε τις μεταβλητές περιβάλλοντος. Οπότε πηγαίνοντας στον πίνακα ελέγχου, σύστημα και ασφάλεια, σύστημα, προηγμένες ρυθμίσεις συστήματος, μεταβλητές περιβάλλοντος δημιουργούμε μία νέα μεταβλητή στις μεταβλητές χρήστη με όνομα CLASSPATH και τιμή την τοποθεσία αποθήκευσης του JDK στον υπολογιστή μας. Ακόμα, στις μεταβλητές συστήματος βρίσκουμε την μεταβλητή Path ή την δημιουργούμε αν δεν υπάρχει και προσθέτουμε στην τιμή την τοποθεσία της JAVA. Όλα αυτά φαίνονται στην παρακάτω εικόνα 9.



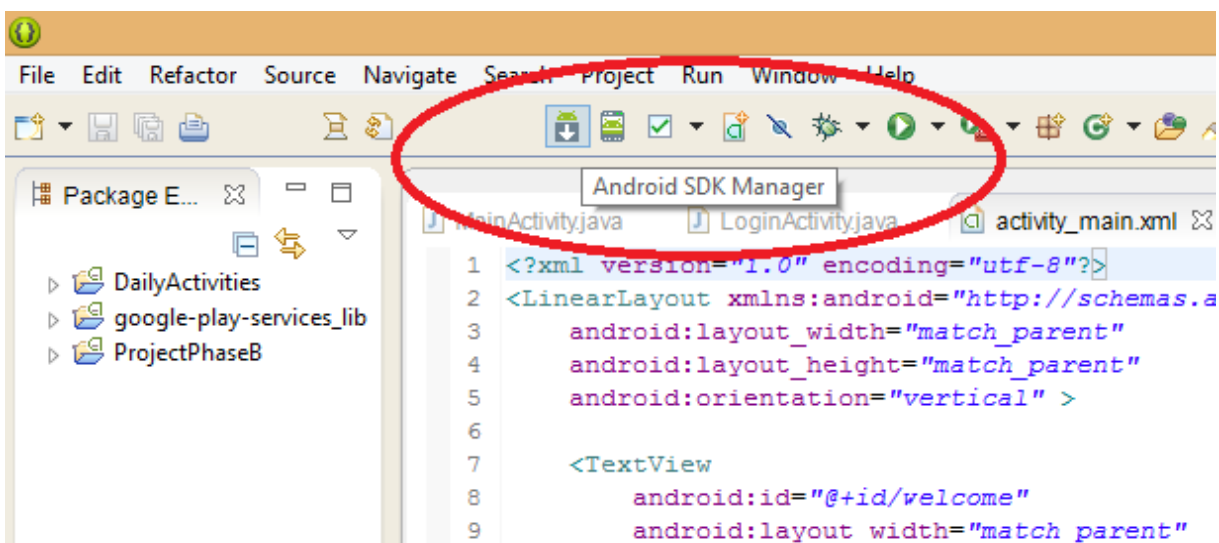
Εικόνα 9 – Εγκατάσταση του JDK No.8

Στη συνέχεια, κατεβάζουμε το Eclipse ADT with Android SDK από την επίσημη ιστοσελίδα της Google <https://developer.android.com/sdk/index.html?hl=i> αφού συμφωνήσουμε με τους όρους χρήσης και διαλέξουμε πια έκδοση να κατεβάσουμε, 32-bit ή 64-bit.



Εικόνα 10 – Εγκατάσταση του Eclipse No.1

Έπειτα κάνουμε αποσυμπίεση του αρχείου που κατεβάσαμε στον σκληρό δίσκο του συστήματος και δεν το τοποθετούμε στον φάκελο Αρχεία Προγράμματος, τρέχουμε το Eclipse, ανοίγουμε το Android SDK Manager και κατεβάζουμε τις εκδόσεις Android για τις οποίες θέλουμε η εφαρμογή μας να είναι συμβατή.^{(1), (19)}



Εικόνα 11 – Εγκατάσταση του Eclipse No.2

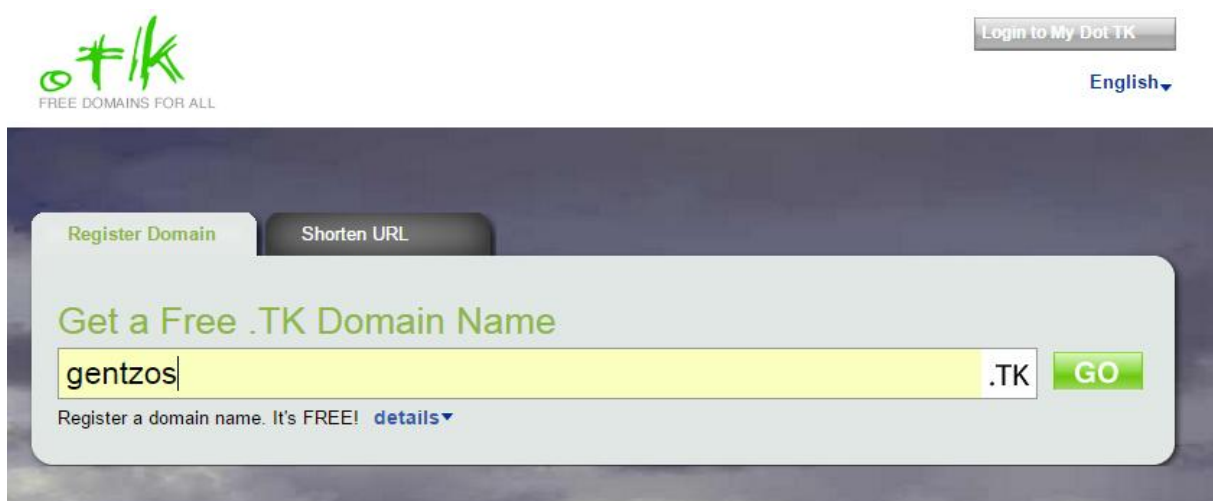
4.4.2 Κατοχύρωση Ονόματος Τομέα (Domain Name) και Φιλοξενίας Χώρου (Hosting)

Πλέον είμαστε έτοιμοι να αρχίσουμε να προγραμματίζουμε σε περιβάλλον Android. Όμως, για να επικοινωνήσει η εφαρμογή μας με έναν Εξυπηρετητή (Server) χρειαζόμαστε ένα Όνομα Τομέα (Domain Name) και ένα μέρος το οποίο θα φιλοξενεί τα αρχεία μας (Hosting). Για την κατοχύρωση του Ονόματος Τομέα (Domain Name) επιλέξαμε το .tk Τομέα (Dot TK Domain). Το Dot TK είναι μια κοινοπραξία της κυβέρνησης του Τοκελάου, μία χώρα στο Νότιο Ειρηνικό Ωκεανό, και των εταιρειών επικοινωνίας της χώρας Teletok και BV Dot TK. Επιλέξαμε το συγκεκριμένο Τομέα γιατί μας επιτρέπει να κατοχυρώσουμε δωρεάν οποιοδήποτε Όνομα Τομέα (Domain Name) είναι διαθέσιμο στον Τομέα .tk.

Για να φιλοξενήσουμε τα αρχεία μας (Hosting) επιλέξαμε το 000webhosting.com, το οποίο μας προσφέρει δωρεάν χώρο στους Εξυπηρετητές (Servers) της εταιρίας. Επίσης, μας προσφέρει:

- 1.5 GB χώρο φιλοξενίας αρχείων,
- 100 GB εύρος ζώνης,
- μπορεί να υποστηρίξει απεριόριστα Ονόματα Τομέα (Domain Names),
- 5 Emails,
- PHP MySQL υποστήριξη,
- FTP υποστήριξη κ.α.

Αρχικά, μεταβαίνουμε στην ιστοσελίδα <http://www.dot.tk/en/index.html> για να ελέγξουμε την διαθεσιμότητα του Ονόματος Τομέα (Domain Name) που θέλουμε να κατοχυρώσουμε (βλέπε εικόνα 12).



Εικόνα 12 – Κατοχύρωση Ονόματος Τομέα (Domain Name) No.1

Εφόσον το Όνομα Τομέα (Domain Name) είναι διαθέσιμο, η ιστοσελίδα μας ανακατευθύνει στο επόμενο βήμα (βλέπε εικόνα 13), αλλιώς πρέπει να επιλέξουμε ένα άλλο όνομα. Στο βήμα αυτό, επιλέγουμε να χρησιμοποιήσουμε το δικό μας DNS και πρέπει να συμπληρώσουμε τα ονόματα των Διακομιστών (Server) μας όπως φαίνεται παρακάτω. Τα ονόματα των Διακομιστών (Server) τα αποκτούμε με την δημιουργία λογαριασμού στην ιστοσελίδα <http://www.000webhost.com/order.php> όπου περιγράφουμε αργότερα.

REGISTRATION OF A FREE .TK DOMAIN

You are registering the following domain name...

GENTZOS.TK

Use your new domain

Forward this domain to **Use DNS** Build a new website with [Imcreator.com](#)

Use your own DNS Services

Server name	<input type="text" value="ns01.000webhost.com"/>	IP address	<input type="text"/>
Server name	<input type="text" value="ns02.000webhost.com"/>	IP address	<input type="text"/>

Registration length

Domains can be registered from 1 to 12 months. Registered Dot TK users are allowed an unlimited number of domain renewals.

Type the characters you see in this picture

GC Q C H T O B

Type the characters you see in this picture

You are seconds away from your FREE domain name...

Becoming a registered Dot TK user allows you to:

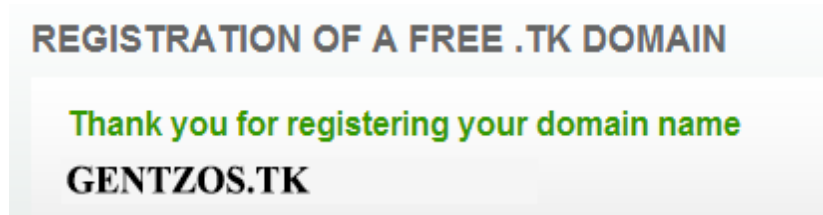
- Update your domain
- Create Tikilinks
- Use Dot TK Apps
- Access TK Stats
- Use the Ambassadors Program
- Renew domains for free

Sign Up ➔

Εικόνα 13 – Κατοχύρωση Ονόματος Τομέα (Domain Name) Νο.2

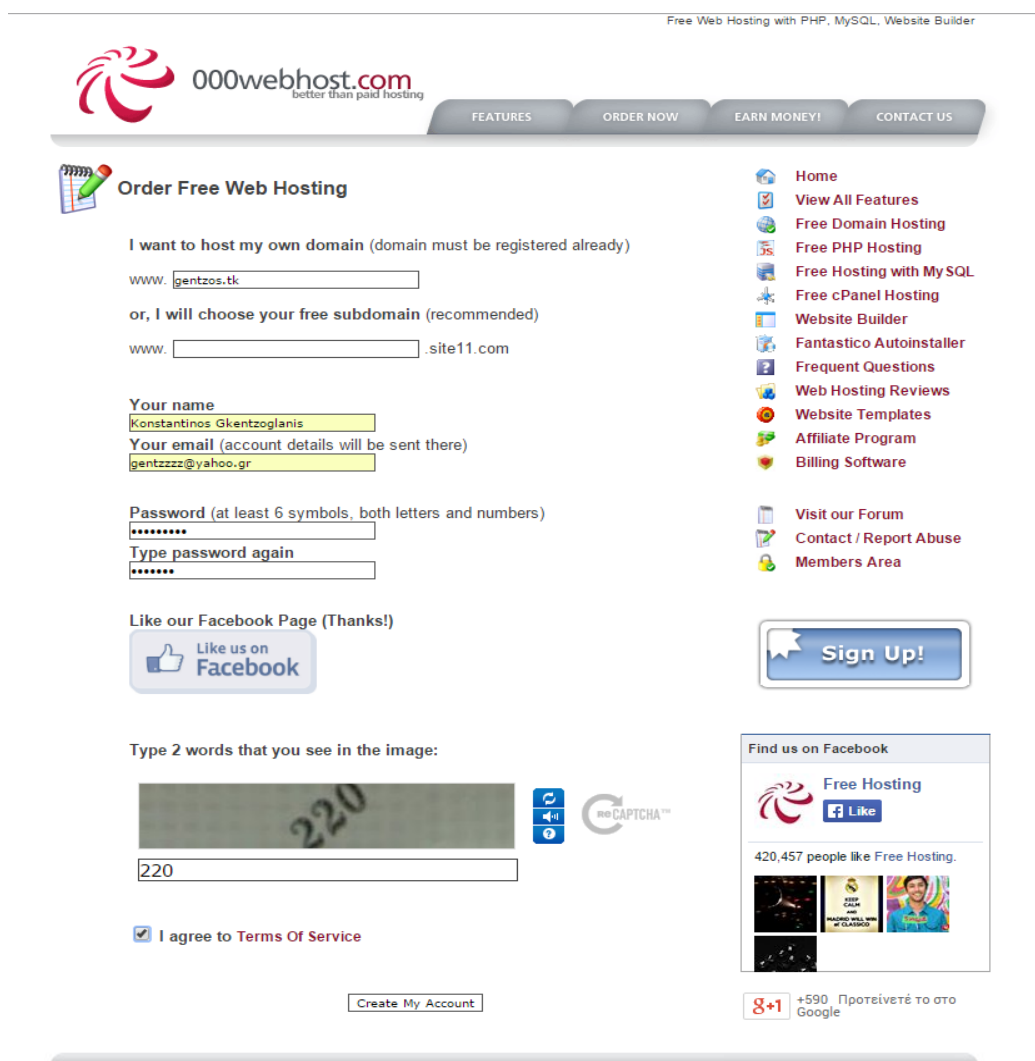
Επίσης, επιλέγουμε το χρονικό όριο που θέλουμε το Όνομα Τομέα (Domain Name) να είναι κατοχυρωμένο, με ελάχιστο όριο ένα μήνα και μέγιστο όριο τους δώδεκα μήνες. Συμπληρώνουμε τον κωδικό επιβεβαίωσης που εμφανίζεται στην οθόνη και πατάμε το κουμπί εγγραφής. Η εγγραφή μπορεί να πραγματοποιηθεί με οποιονδήποτε λογαριασμό κοινωνικής δικτύωσης, όπως Facebook, Google + κ.α., ή με έναν λογαριασμό email. Εφόσον η εγγραφή πραγματοποιήθηκε επιτυχώς,

εμφανίζεται ένα μήνυμα επιβεβαίωσης για την επιτυχή κατοχύρωση του Ονόματος Τομέα (Domain Name).



Εικόνα 14 – Κατοχύρωση Ονόματος Τομέα (Domain Name) Νο.3

Για την απόκτηση των ονομάτων των Διακομιστών (Server) μεταβαίνουμε στην ιστοσελίδα <http://www.000webhost.com/order.php> για να δημιουργήσουμε ένα νέο λογαριασμό με τον οποίο θα έχουμε πρόσβαση στον χώρο φιλοξενίας (Hosting) και κατά συνέπεια στους Διακομιστές (Servers). Συμπληρώνουμε τα στοιχεία που είναι απαραίτητα όπως το Όνομα Τομέα (Domain Name) που επιλέξαμε στο προηγούμενο βήμα, το ονοματεπώνυμό μας, το email μας και έναν κωδικό, όλα αυτά φαίνονται στην παρακάτω εικόνα 15.



Εικόνα 15 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) Νο.1

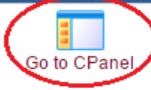

Έπειτα, πρέπει να επιβεβαιώσουμε την διεύθυνση email που δώσαμε στο 000webhost.com. Μετά την επιβεβαίωση του email μας, πατάμε το CPanel όπως φαίνεται παρακάτω (βλέπε εικόνα 16).

Manage another domain

gentzos.tk



List of your domains

» Domain	» Status	» Action
gentzos.tk	Active	 Go to CPanel  Website Builder

» Customer Details	
Name	Konstantinos Gentzoglakis
Email	gentzzzz@yahoo.gr
Registration	2014-05-22 06:23
Account Status	Active
Last Login From	94.71.117.113
Location	Greece

If you like our free hosting services you can help us in any of these 3 ways:



Put a link from your website

You can help us by putting a link from your website to <http://www.000webhost.com/>, get HTML code [here](#).



Donate by Paypal

If you found our services useful - [send a donation](#). Each dollar will help us to run this free web hosting service.



Upgrade your account!

Want a faster more reliable and truly **unlimited** web hosting service? [Upgrade your account](#) and you will get the best available web hosting service for just \$4.84 per month.



Earn Money!

Εικόνα 16 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.2

Στη συνέχεια πατάμε το View Account Details όπως φαίνεται στην επόμενη εικόνα 17.

Manage another domain

gentzos.tk






Email Passwords Updated

As per Norton Security recommendations web hosts should reset Email passwords at least once every 12 months.



We follow these recommendations and changed your password to a random string. You can change this password to anything else by going to "CPanel" -> "Manage Email Accounts" -> "Change Password" (do not use your old password).

» Main Menu

 View Account Details	 Getting Started Information	 Frequently Asked Questions	 Change Account Password	 Upgrade Account!
--	---	--	---	--

» Domains

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

» Account Information	
Domain	gentzos.tk
Username	a2765732
Password	*****
Disk Usage	0.18 / 1500.0 MB
Bandwidth	100000 MB (100GB)
Home Root	/home/a2765732
Server Name	server21.000webhost.com
IP Address	31.170.160.85
Apache ver.	2.2.19 (Unix)
PHP version	5.2.*
MySQL ver.	5.1
Activated On	2014-10-18 04:47
Status	Active
Plan	Free (Upgrade!)

Εικόνα 17 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.3

Εδώ βλέπουμε τα ονόματα του Διακομιστή (Server) τα οποία μπορούμε να χρησιμοποιήσουμε για να αποκτήσουμε το Όνομα Τομέα (Domain Name) που περιγράψαμε νωρίτερα (βλέπε εικόνα 18).



Manage another domain

gentzos.tk

Account Details Overview

[[Enter Control Panel](#)]

This is a short overview of your account details. To learn more about each feature browse icons inside control panel.

Don't know how to get started? Check "[Getting Started](#)" information.

Nameserver details (only if you host your own domain)

ns01.000webhost.com	31.170.167.6
ns02.000webhost.com	31.170.164.248

You can also point your domain ("A" DNS record) to IP **31.170.163.51**

Website Details

Access your website at	http://gentzos.tk
Access your website with www.	http://www.gentzos.tk
Website IP address	31.170.160.85
Server name	server21.000webhost.com

File Upload Details ([learn more](#))


FTP host name	gentzos.tk
FTP user name	a2765732
FTP password	*****
File system path	/home/a2765732
Folder to upload files	public_html
Forgot FTP password?	Change account password
Recommended FTP clients	SmartFTP or FileZilla
Web File Manager	Click here

Εικόνα 18 – Κατοχύρωση Χώρου Φιλοξενίας (Hosting) No.4

Με το τέλος της διαδικασίας αυτής, έχουμε αποκτήσει το δικό μας Όνομα Τομέα (Domain Name), το οποίο είναι gentzos.tk, και έναν Εξυπηρετητή (Server) στον οποίο μπορούμε να αποθηκεύουμε τα αρχεία μας αλλά και να διαχειριζόμαστε τη Βάση Δεδομένων μας. Όλα αυτά θα είναι έτοιμα για να τα χρησιμοποιήσουμε στις επόμενες 24 ώρες.

Για να ανεβάσουμε στον Διακομιστή (Server) τα αρχεία που θέλουμε, όπως XML αρχεία, PHP αρχεία και εικόνες, μπορούμε να χρησιμοποιήσουμε διάφορα προγράμματα, όπως το SmartFTP και το FileZilla. Επίσης, μπορούμε να χρησιμοποιήσουμε ένα διαχειριστή αρχείων μέσω της ιστοσελίδας

<http://members.000webhost.com/login.php> στην οποία πρέπει να συνδεθούμε με τον λογαριασμό που δημιουργήσαμε νωρίτερα. Έπειτα, μεταβαίνουμε στο CPanel, πηγαίνουμε στο View Account Details και πατάμε το Web File Manager (βλέπε εικόνα 19).

 **File Upload Details** ([learn more](#))

FTP host name	genzos.tk
FTP user name	a2765732
FTP password	*****
File system path	/home/a2765732
Folder to upload files	public_html
Forgot FTP password?	Change account password
Recommended FTP clients	SmartFTP or FileZilla
Web File Manager	Click here

Εικόνα 19 – Μεταφόρτωση Αρχείων Νο.1

Πατώντας το Web File Manager μεταβαίνουμε σε μία νέα ιστοσελίδα στην οποία μας δίνεται η δυνατότητα να μεταφορτώσουμε τα αρχεία μας στον Διακομιστή. Τα αρχεία τα μεταφορτώνουμε στον φάκελο public_html αναγκαστικά, αφού έτσι ορίζει η εταιρία 000webhost.com (βλέπε εικόνα 20).

Είναι απαραίτητο να μπορούμε να ανεβάζουμε αρχεία στον Διακομιστή για την εφαρμογή, επειδή εκεί θα βρίσκονται όλες οι πληροφορίες για τα βενζινάδικα και τα σούπερ μάρκετ σε μορφή XML καθώς και τα λογότυπά τους σε εικόνες, τις οποίες πληροφορίες θα κατεβάζει η εφαρμογή για να μπορέσει να λειτουργήσει σωστά. Επίσης, τα αρχεία PHP είναι απαραίτητα διότι αυτά μας βοηθάνε στο να διαχειριστούμε την Βάση Δεδομένων, δηλαδή μας βοηθάνε στο να πραγματοποιούνται οι εγγραφές των χρηστών, να γίνεται είσοδος στον λογαριασμό τους και να είναι εφικτή η τροποποίηση του λογαριασμού τους.

■■■■■■■■■■ 100% Script finished in 0.10 seconds


genzos.tk

Language: English

Directory Tree: root /public_html

All	Name	Type	Size	Owner	Group	Perms	Mod Time	Actions
<input type="checkbox"/>	Up..							
<input type="checkbox"/>	genzos_login_api	Directory	4096	a2765732	a2765732	rw-r-xr-x	Sep 15 05:23	
<input type="checkbox"/>	icons	Directory	4096	a2765732	a2765732	rw-r-xr-x	Sep 15 04:15	
<input type="checkbox"/>	xml	Directory	4096	a2765732	a2765732	rw-r-xr-x	Sep 15 04:14	
<input type="checkbox"/>	.htaccess	HTACCESS File	91	a2765732	a2765732	rw-r--r--	Sep 14 15:51	View Edit Open

Directories: 3
Files: 1 / 91 B
Symlinks: 0

Εικόνα 20 – Μεταφόρτωση Αρχείων Νο.2

Τώρα, πρέπει να δημιουργήσουμε την Βάση Δεδομένων την οποία θα χρησιμοποιήσουμε για να αποθηκεύουμε τους χρήστες. Οπότε, πηγαίνοντας στο CPanel και μετά στο View Account Details πατάμε το MySQL Management (βλέπε εικόνα 21).

Quick Links

Upgrade account
Change account password
MySQL management
phpMyAdmin
Enter helpdesk
Visit our forum
www.000webhost.com

Εικόνα 21 – Δημιουργία Βάσης Δεδομένων Νο.1

Έπειτα, συμπληρώνουμε τα πεδία ονόματος της Βάσης Δεδομένων, του ονόματος χρήστη και τον κωδικό χρήστη (βλέπε εικόνα 22).

Manage MySQL Databases

MySQL databases are required by many web applications including bulletin boards, content management systems, and others. To use MySQL, you need to create database and user, which will be automatically assigned to this database. Click for [phpMyAdmin](#) when database is created.

Important: MySQL Host for any database in this account is **mysql5.000webhost.com**, do not use localhost!

Create new database and user

MySQL database name:	a2765732_	<input type="text"/>
MySQL user name:	a2765732_	<input type="text"/>
Password for MySQL user:	<input type="password"/>	
Enter password again:	<input type="password"/>	
<input type="button" value="Create database"/>		

Εικόνα 22 – Δημιουργία Βάσης Δεδομένων Νο.2

Στη συνέχεια, εμφανίζονται στην οθόνη μας τα στοιχεία τα οποία τα αντιγράφουμε στο αρχείο config.php για να μπορούμε να συνδεόμαστε στη Βάση Δεδομένων που μόλις δημιουργήσαμε από την εφαρμογή μας.

Αν θέλουμε να διαχειριστούμε την Βάση Δεδομένων, τότε πηγαίνοντας στο CPanel, View Account Details, phpMyAdmin, βλέπουμε όλες τις Βάσεις Δεδομένων που έχουμε δημιουργήσει και επιλέγουμε αυτή που θέλουμε να διαχειριστούμε (βλέπε εικόνες 23, 24).

Quick Links

Upgrade account
Change account password
MySQL management
phpMyAdmin
Enter helpdesk
Visit our forum
www.000webhost.com

Εικόνα 23 – Πρόσβαση στη Βάση Δεδομένων Νο.1

phpMyAdmin Access

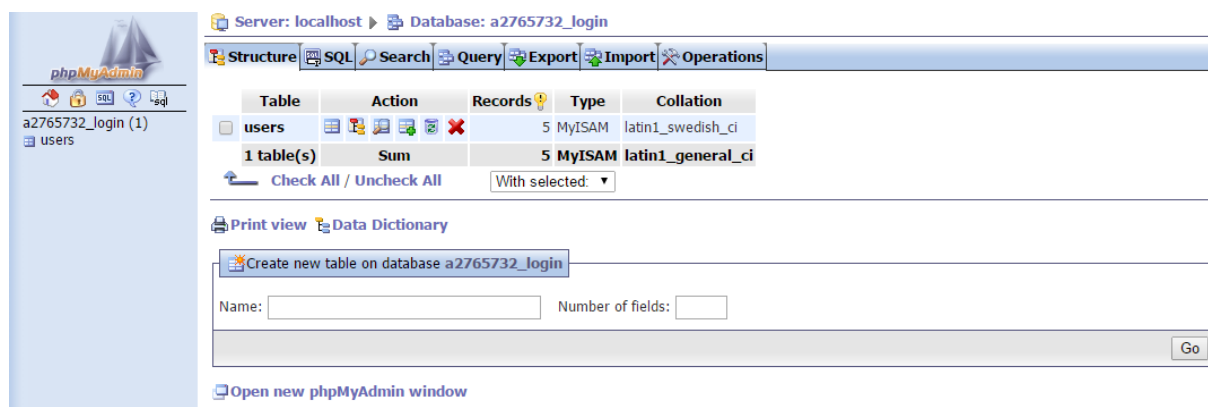
phpMyAdmin is an open source tool written in PHP intended to handle the administration of MySQL over the Internet. Currently it can create/drop/alter tables, delete/edit/add fields, execute any SQL statement, and manage keys on fields. To create / drop MySQL databases go to **MySQL management**. **Important:** if you have just created new database it can take about 1 minute until phpMyAdmin starts working!

List of current databases:

» MySQL Database	» MySQL User	» phpMyAdmin
a2765732_login	a2765732_gentzos	Enter phpMyAdmin

Εικόνα 24 – Πρόσβαση στη Βάση Δεδομένων Νο.2

Τέλος, μεταβαίνουμε στη Βάση Δεδομένων που επιλέξαμε και μπορούμε να διαχειριστούμε αυτήν αλλά και όλους τους πίνακες που εμπεριέχει (βλέπε εικόνα 25).^{(20), (21)}



Server: localhost Database: a2765732_login

Structure SQL Search Query Export Import Operations

Table	Action	Records	Type	Collation
users		5	MyISAM	latin1_swedish_ci
1 table(s) Sum		5	MyISAM	latin1_general_ci

Check All / Uncheck All With selected: ▾

Print view Data Dictionary

Create new table on database a2765732_login

Name: Number of fields:

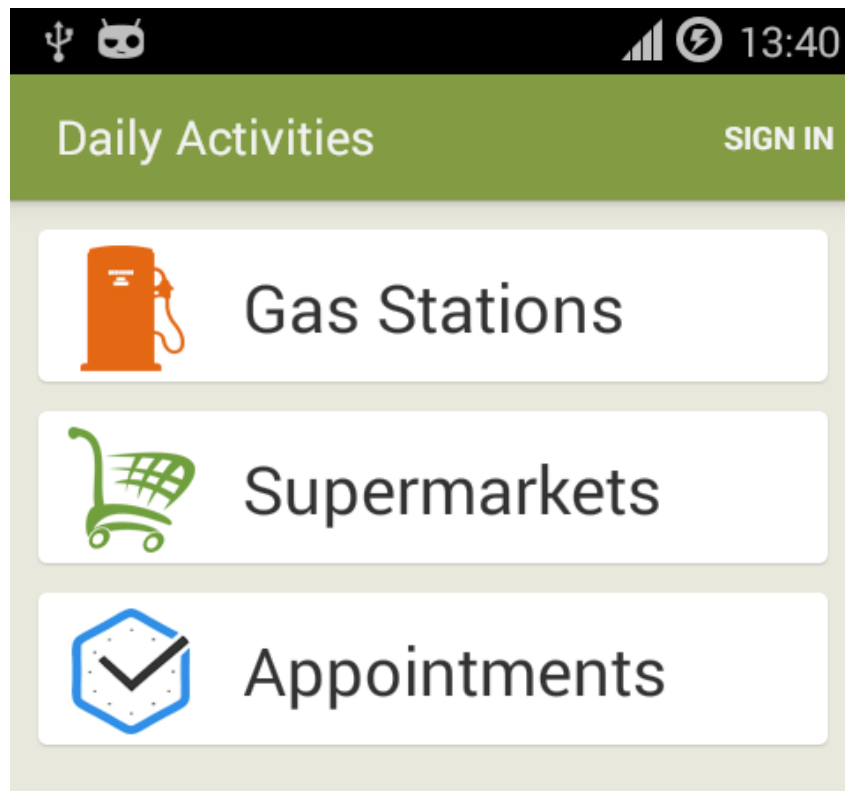
Go

Open new phpMyAdmin window

Εικόνα 25 – Πρόσβαση στη Βάση Δεδομένων Νο.3

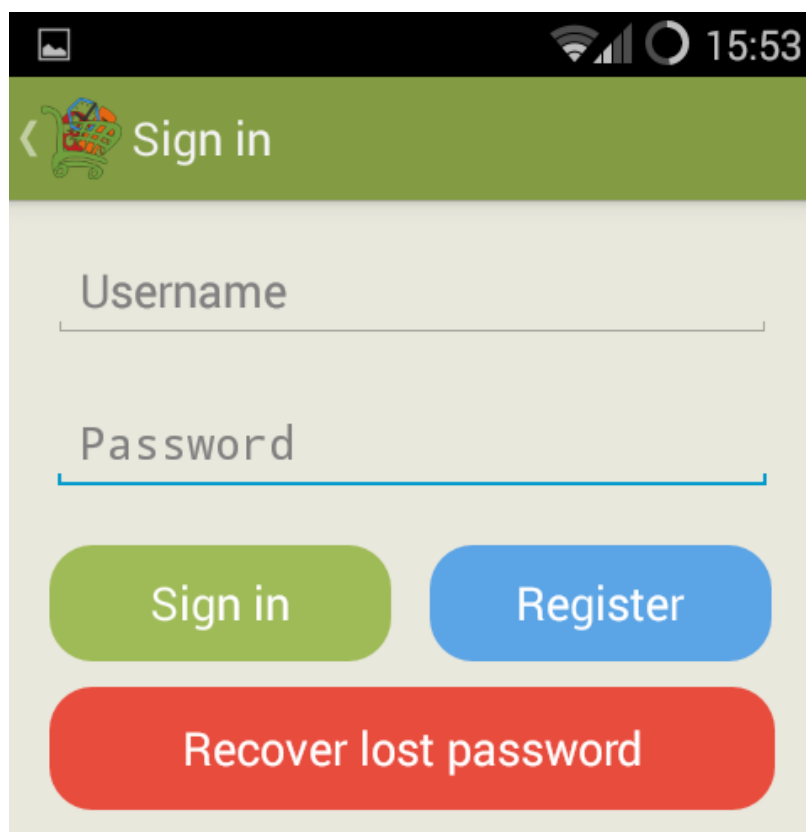
4.4.3 Η Εφαρμογή

Τώρα που ο ηλεκτρονικός υπολογιστής είναι έτοιμος και μπορούμε να δημιουργήσουμε εφαρμογές Android, οι οποίες μπορούν να επικοινωνήσουν με ένα Εξυπηρετητή (Server) αλλά και με μία Βάση Δεδομένων, μπορούμε να προχωρήσουμε στην υλοποίηση της εφαρμογής αυτής. Αρχικά, όταν ο χρήστης ανοίγει την εφαρμογή βλέπει μία λίστα με τις δραστηριότητες που μπορεί να τον βοηθήσει η εφαρμογή, όπως τα βενζινάδικα, τα σούπερ μάρκετ και τα ραντεβού (βλέπε εικόνα 26).



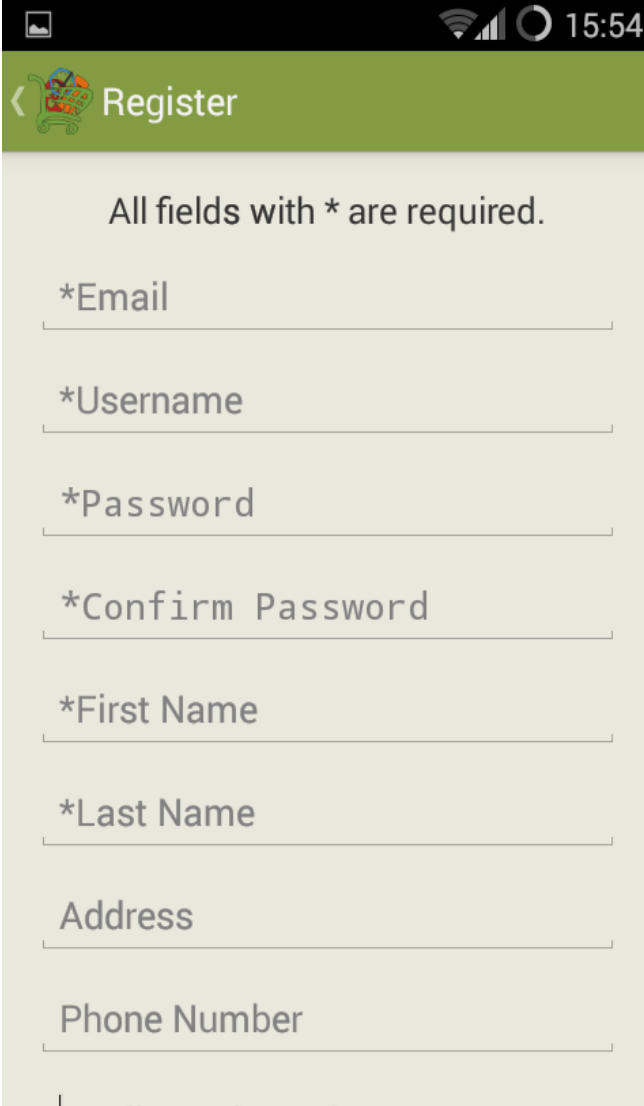
Εικόνα 26 – Αρχική Δραστηριότητα Εφαρμογής No.1

Από εδώ ο χρήστης μπορεί να επιλέξει να εισάγει ένα ραντεβού, να του βρει η εφαρμογή το κοντινότερο βενζινάδικο – σούπερ μάρκετ ή να συνδεθεί στην εφαρμογή. Αν ο χρήστης επιλέξει να συνδεθεί στην εφαρμογή μεταβαίνει στην Δραστηριότητα Σύνδεσης (βλέπε εικόνα 27).



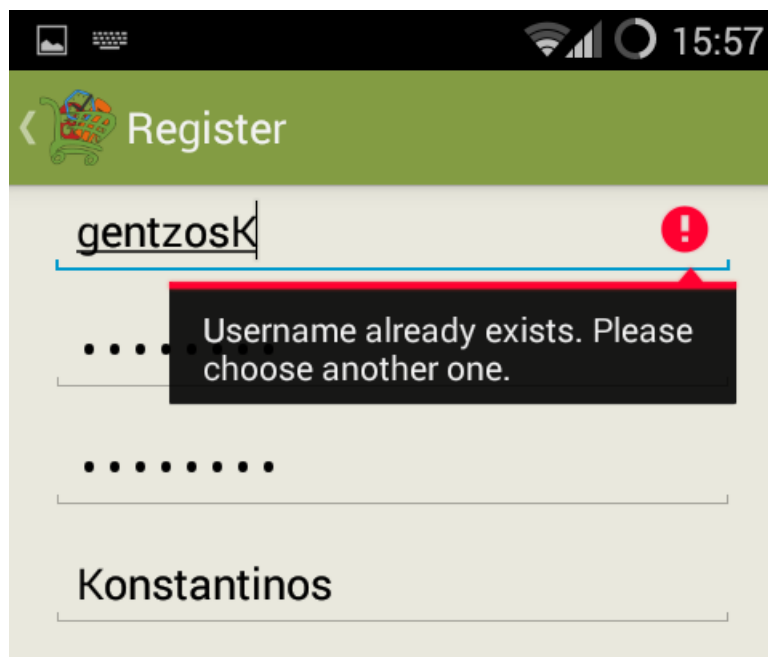
Εικόνα 27 – Δραστηριότητα Σύνδεσης No.1

Από αυτή τη Δραστηριότητα ο χρήστης μπορεί να συνδεθεί στην εφαρμογή, να κάνει εγγραφή σε αυτή ή να αποκτήσει νέο κωδικό πρόσβασης αν έχει ξεχάσει τον παλιό του κωδικό. Αν αυτή είναι η πρώτη επίσκεψη του χρήστη στην εφαρμογή και θέλει να δημιουργήσει λογαριασμό, τότε πατάει το κουμπί εγγραφής και μεταβαίνει στην επόμενη Δραστηριότητα (βλέπε εικόνα 28).



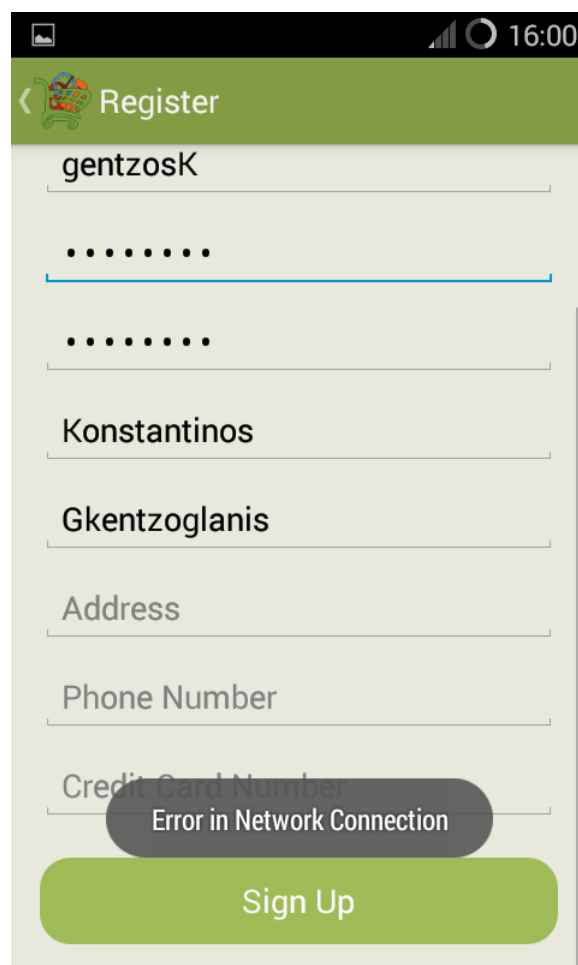
Εικόνα 28 – Δραστηριότητα Εγγραφής No.1

Σε αυτή τη Δραστηριότητα ο χρήστης πρέπει να συμπληρώσει όλα τα απαραίτητα πεδία, τα οποία είναι αυτά με το σύμβολο *, και πατάει το κουμπί εγγραφής. Αν υπάρξει κάποιο πρόβλημα με τα πεδία συμπλήρωσης στοιχείων τότε η διαδικασία εγγραφής διακόπτεται και εμφανίζεται ένα κόκκινο θαυμαστικό δίπλα στο πεδίο με το πρόβλημα μαζί με μία επεξήγηση του προβλήματος. Τέτοια προβλήματα είναι αν ο χρήστης έχει ξεχάσει να συμπληρώσει ένα απαραίτητο πεδίο, αν το όνομα χρήστη είναι πολύ μικρό, αν ο κωδικός πρόσβασης είναι πολύ μικρός, αν στα δύο πεδία κωδικών οι κωδικοί δεν ταιριάζουν μεταξύ τους, αν η διεύθυνση email δεν είναι έγκυρη ή αν υπάρχει ήδη λογαριασμός με αυτή τη διεύθυνση και τέλος αν το όνομα χρήστη χρησιμοποιείται από κάποιον άλλο χρήστη. Ένα τέτοιο παράδειγμα προβλήματος φαίνεται στην παρακάτω εικόνα 29.



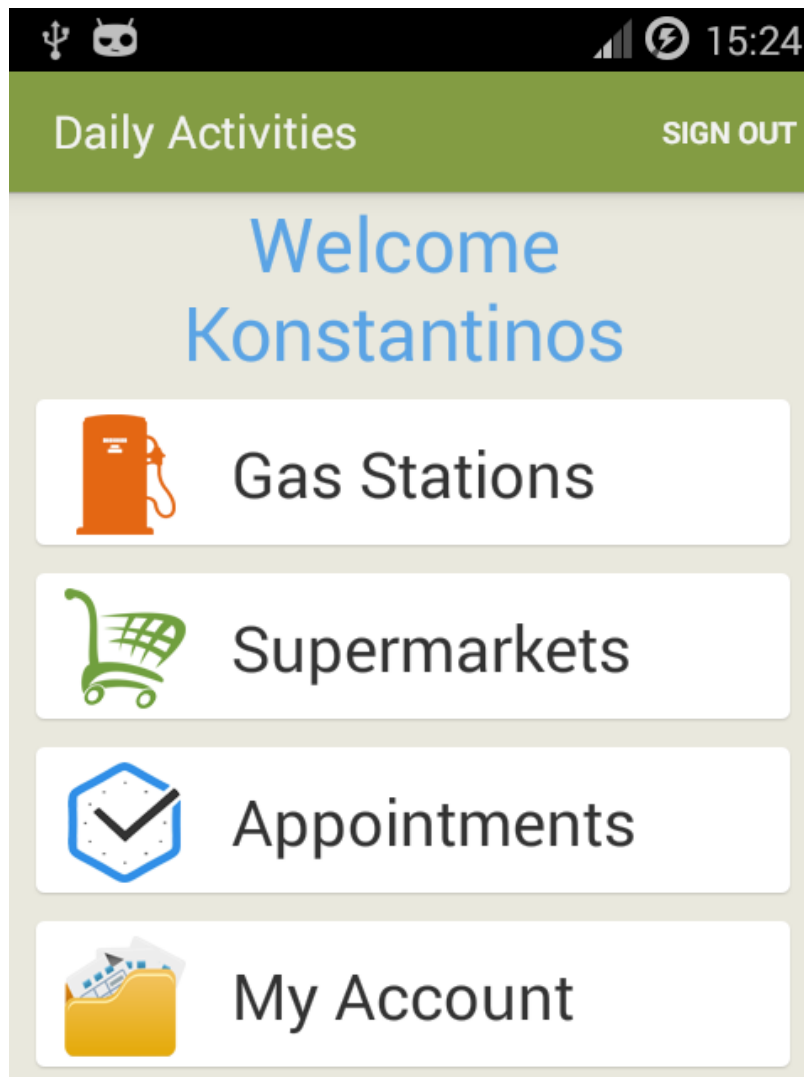
Εικόνα 29 – Δραστηριότητα Εγγραφής Νο.2

Ένα τελευταίο πρόβλημα που μπορεί να υπάρξει είναι αν δεν υπάρχει δίκτυο στην εφαρμογή για να επικοινωνήσει με τη Βάση Δεδομένων και τότε εμφανίζεται μήνυμα λάθους όπως στην παρακάτω εικόνα 30.



Εικόνα 30 – Δραστηριότητα Εγγραφής Νο.3

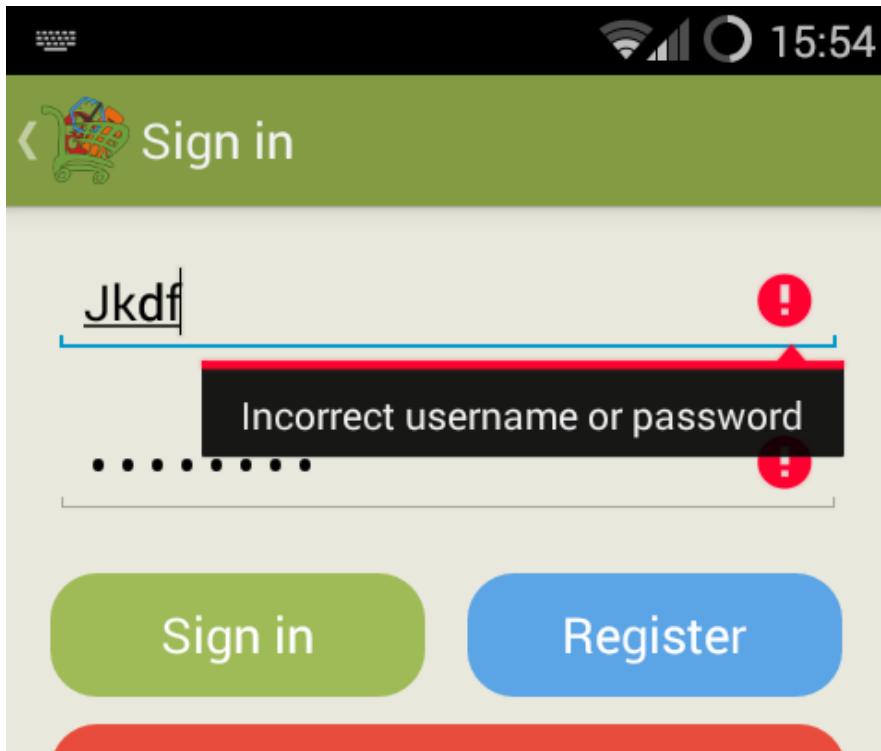
Αν δεν υπάρξει κάποιο μήνυμα λάθους τότε η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων, η εγγραφή του χρήστη πραγματοποιείται με επιτυχία και ο χρήστης μεταβαίνει αυτόματα στην Αρχική Δραστηριότητα της εφαρμογής καλωσορίζοντας τον με το μικρό του όνομα. Βέβαια στην λίστα δραστηριοτήτων εμφανίζεται τώρα άλλο ένα πεδίο το οποίο μπορεί να επιλέξει ο χρήστης για να δει τα στοιχεία του λογαριασμού του αλλά και να τροποποιήσει. Επίσης, το κουμπί σύνδεσης έχει αντικατασταθεί από το κουμπί αποσύνδεσης (βλέπε εικόνα 31).



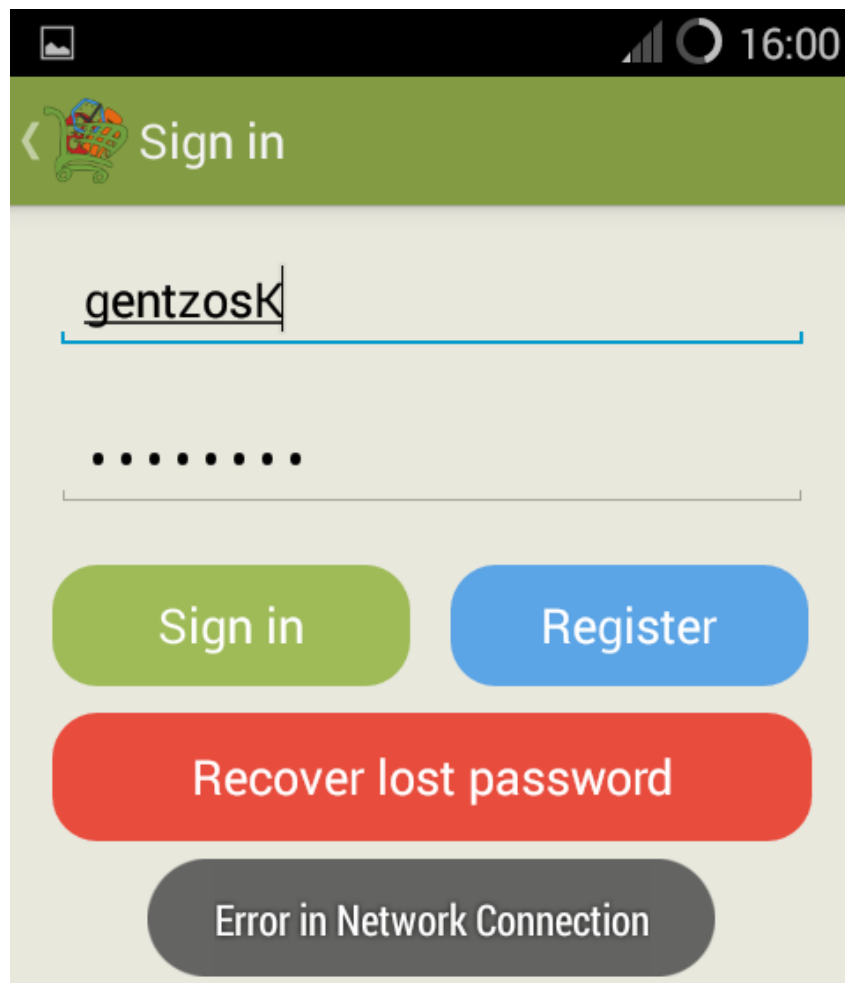
Εικόνα 31 – Αρχική Δραστηριότητα Εφαρμογής Νο.2

Όμως, αν ο χρήστης έχει ήδη λογαριασμό στην εφαρμογή αυτή, τότε μπορεί στην Δραστηριότητα Σύνδεσης να συμπληρώσει το όνομα χρήστη και τον κωδικό πρόσβασης του για να συνδεθεί στην εφαρμογή. Σε αυτή τη Δραστηριότητα και τα δύο διαθέσιμα πεδία είναι απαραίτητα και αν υπάρξει κάποιο πρόβλημα με αυτά τότε εμφανίζεται ένα κόκκινο θαυμαστικό δίπλα στο πεδίο με μία επεξήγηση του προβλήματος. Τέτοια προβλήματα μπορεί να είναι η μη συμπλήρωση κάποιου από τα πεδία, η εισαγωγή εσφαλμένου ονόματος χρήστη ή εσφαλμένου κωδικού πρόσβασης. Επίσης, υπάρχει και το πρόβλημα μη ύπαρξης δικτύου στην συσκευή όπου εμφανίζεται κατάλληλο μήνυμα.

Αν δεν υπάρχει κάποιο σφάλμα, η εφαρμογή επικοινωνεί με την Βάση Δεδομένων και ο χρήστης συνδέεται επιτυχώς στην εφαρμογή. Έπειτα, ο χρήστης μεταβαίνει αυτόματα στην Αρχική Δραστηριότητα της εφαρμογής και η εφαρμογή τον καλωσορίζει με το μικρό του όνομα. Παραδείγματα των πιθανόν σφαλμάτων φαίνονται στις παρακάτω εικόνες 32 και 33. Η επιτυχής σύνδεση στην εφαρμογή έχει το ίδιο αποτέλεσμα με την επιτυχή εγγραφή του χρήστη στην εφαρμογή και φαίνεται στην παραπάνω εικόνα 31.

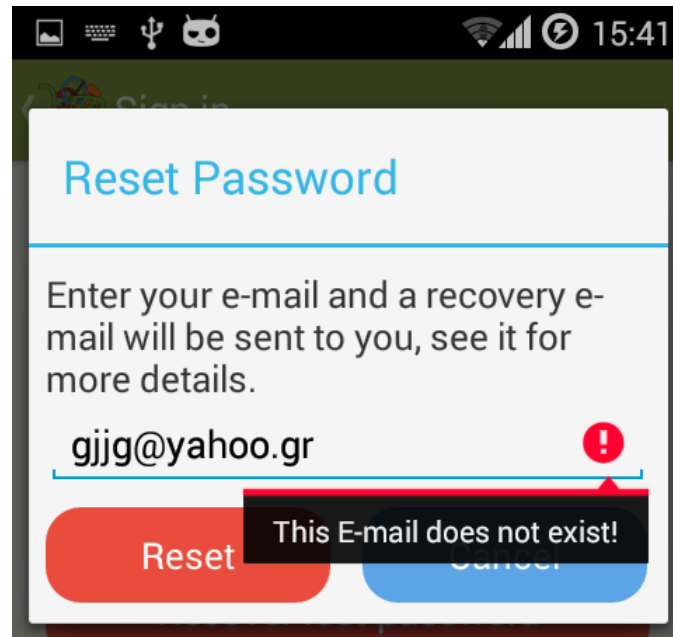


Εικόνα 32 – Δραστηριότητα Σύνδεσης Νο.2



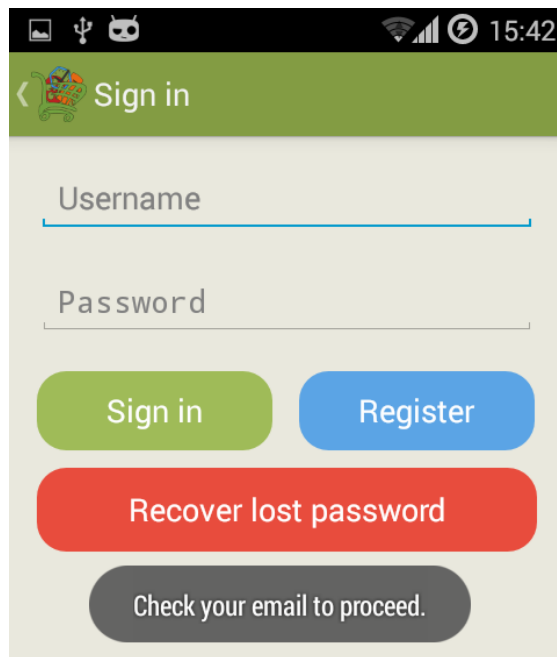
Εικόνα 33 – Δραστηριότητα Σύνδεσης Νο.3

Ακόμα, στην Δραστηριότητα σύνδεσης μπορεί ο χρήστης να πατήσει το κουμπί ανάκτησης χαμένου κωδικού πρόσβασης αν έχει ξεχάσει τον κωδικό του. Τότε εμφανίζεται ένα νέο παράθυρο στο οποίο ζητείται από τον χρήστη να συμπληρώσει το email του. Το πεδίο του email είναι απαραίτητο να συμπληρωθεί για να προχωρήσει η διαδικασία ανάκτησης του χαμένου κωδικού, επίσης το email πρέπει να είναι έγκυρο και να υπάρχει στη Βάση Δεδομένων. Σε κάθε άλλη περίπτωση εμφανίζεται το κόκκινο θαυμαστικό δίπλα από το πεδίο μαζί με μία επεξήγηση του προβλήματος. Επίσης, αν δεν υπάρχει σύνδεση στο δίκτυο τότε εμφανίζεται μήνυμα λάθους (βλέπε εικόνα 34).



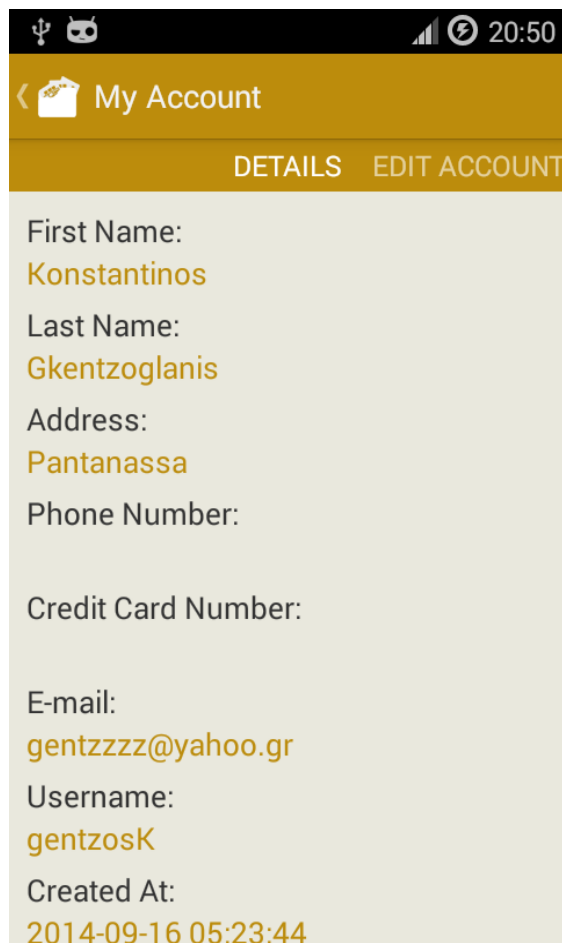
Εικόνα 34 – Δραστηριότητα Σύνδεσης Νο.4

Αν δεν υπάρξει κανένα πρόβλημα τότε η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και εμφανίζεται ένα μήνυμα για να ελέγξει ο χρήστης τα εισερχόμενα email του για το email που του έστειλε η εφαρμογή με το νέο του κωδικό. Στο email αυτό συμβουλευεται ο χρήστης να αλλάξει τον νέο κωδικό πρόσβασης άμεσα με ένα δικό του μέσα από την Δραστηριότητα Λογαριασμού της εφαρμογής (βλέπε εικόνα 35).



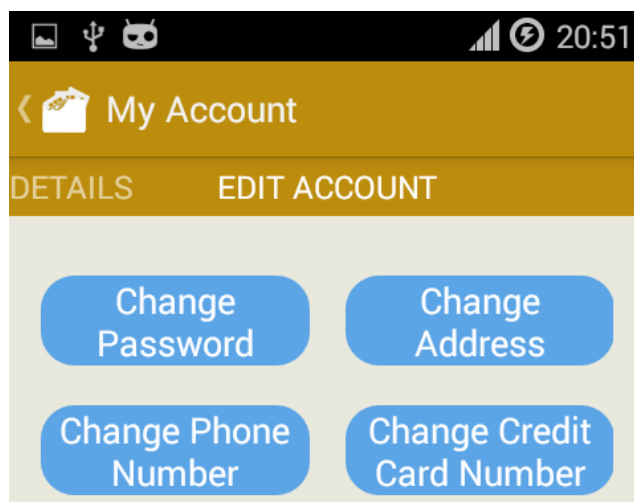
Εικόνα 35 – Δραστηριότητα Σύνδεσης Νο.5

Εφόσον ο χρήστης είναι συνδεδεμένος στην εφαρμογή, έχει πρόσβαση στην Δραστηριότητα Λογαριασμού όπου μπορεί να δει λεπτομέρειες για τον λογαριασμό του αλλά και να αλλάξει τον κωδικό του πρόσβασης, την διεύθυνσή του, το τηλέφωνό του και τον αριθμό της πιστωτικής του κάρτας. Η Δραστηριότητα Λογαριασμού φαίνεται στην παρακάτω εικόνα 36 όπου το πρώτο πράγμα που βλέπει ο χρήστης είναι οι λεπτομέρειες του λογαριασμού του.



Εικόνα 36 – Δραστηριότητα Λογαριασμού Νο.1

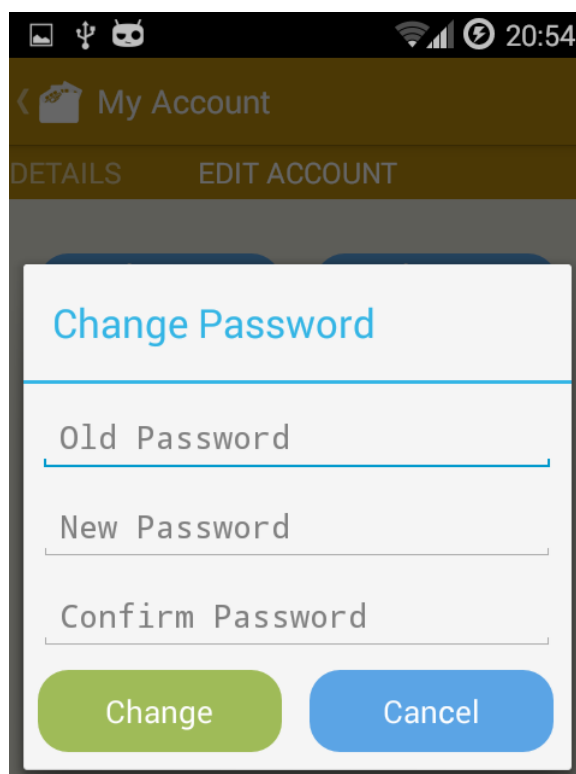
Σέρνοντας ο χρήστης το δάκτυλό του προς τα αριστερά μεταβαίνει στην επόμενη καρτέλα της Δραστηριότητας Λογαριασμού όπου μπορεί να τροποποιήσει τα στοιχεία του (βλέπε εικόνα 37).



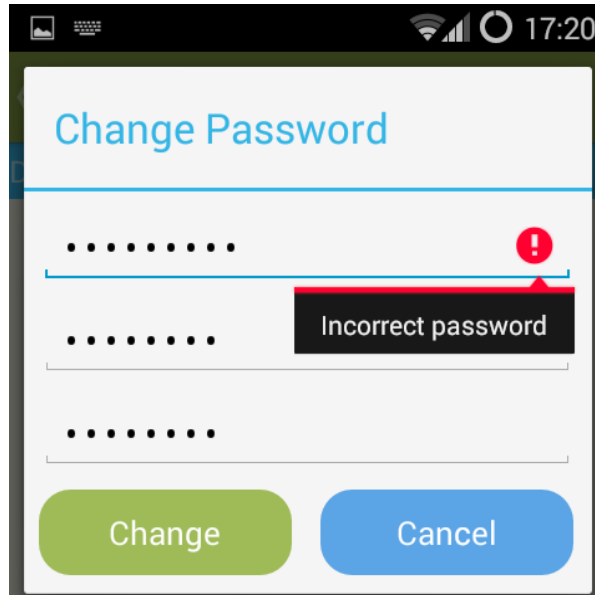
Εικόνα 37 – Δραστηριότητα Λογαριασμού Νο.2

Σέρνοντας ο χρήστης το δάκτυλό του προς τα δεξιά μεταβαίνει στην προηγούμενη καρτέλα της Δραστηριότητας Λογαριασμού όπου μπορεί να δει τα στοιχεία του (βλέπε εικόνα 36).

Στην καρτέλα τροποποίησης των στοιχείων του, αν ο χρήστης πατήσει το κουμπί αλλαγής κωδικού τότε εμφανίζεται ένα παράθυρο το οποίο ζητάει από τον χρήστη να εισάγει τον παλιό του κωδικό πρόσβασης, τον νέο κωδικό καθώς και την επαλήθευσή του. Πατώντας το κουμπί αλλαγής πρέπει όλα τα πεδία να είναι συμπληρωμένα, ο παλιός κωδικός να ταιριάζει με τον τρέχον κωδικό πρόσβασης, ο νέος κωδικός να μην είναι πολύ μικρός και να ταιριάζει με τον κωδικό στο πεδίο επαλήθευσης. Αν κάτι από αυτά δεν είναι σωστά, τότε προκύπτει σφάλμα και εμφανίζεται ένα κόκκινο θαυμαστικό δίπλα στο το πεδίο με το σφάλμα μαζί με μία επεξήγηση (βλέπε εικόνες 38, 39).

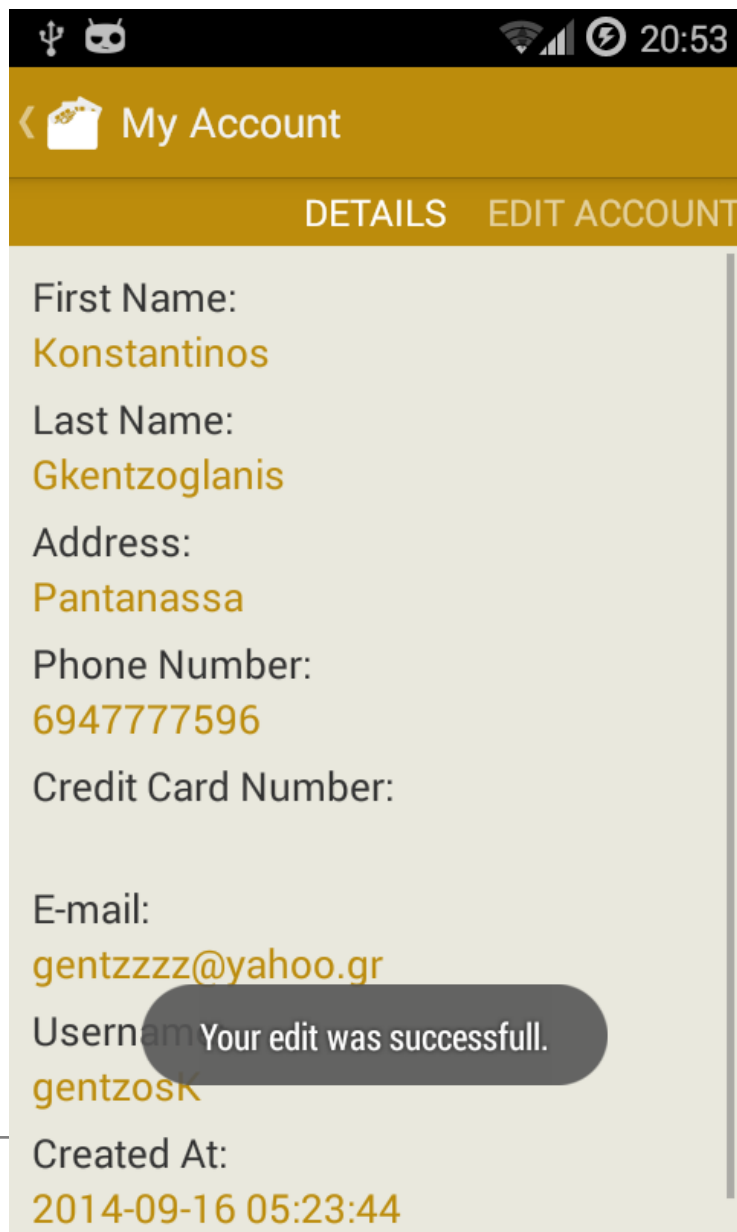


Εικόνα 38 – Δραστηριότητα Λογαριασμού Νο.3

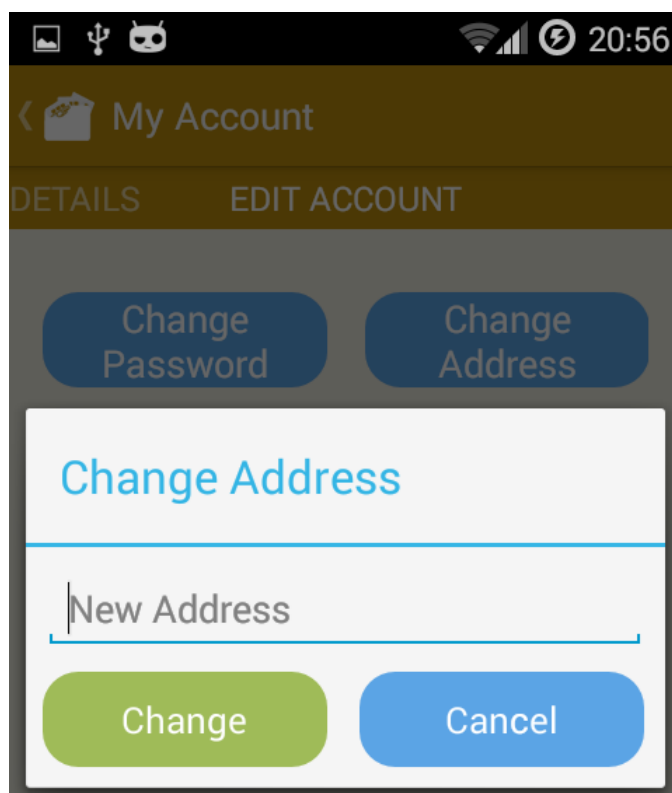


Εικόνα 39 – Δραστηριότητα Λογαριασμού Νο.4

Αν δεν υπάρξει κάποιο σφάλμα τότε η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων, πραγματοποιείται η αλλαγή κωδικού, ο χρήστης μεταβαίνει αυτόματα στην Δραστηριότητα Λογαριασμού στην καρτέλα με τα στοιχεία του λογαριασμού και εμφανίζεται ένα μήνυμα επιτυχίας (βλέπε εικόνα 40).

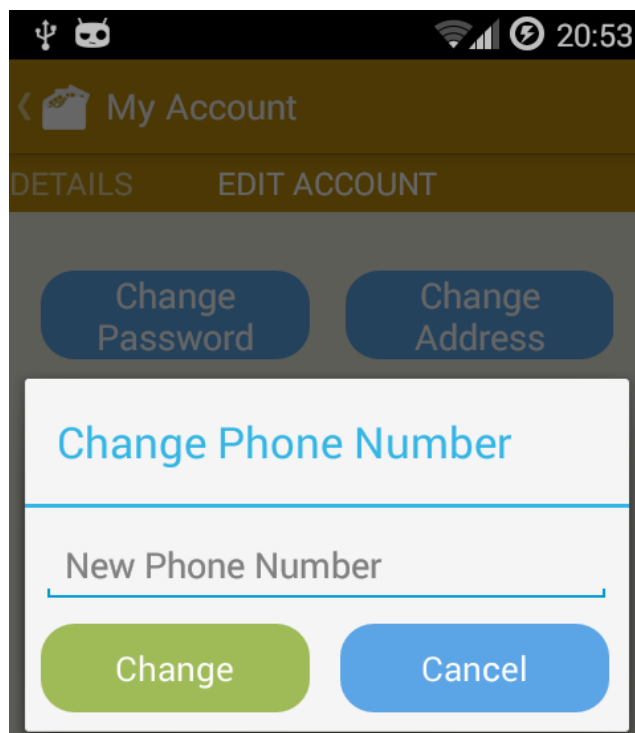


Αν ο χρήστης πατήσει το κουμπί αλλαγής διεύθυνσης εμφανίζεται ένα παράθυρο το οποίο ζητάει τη νέα διεύθυνση του χρήστη (βλέπε εικόνα 41). Αν ο χρήστης αφήσει κενό αυτό το πεδίο τότε η διεύθυνση θα διαγραφεί. Αν το πεδίο δεν είναι κενό τότε η τρέχουσα διεύθυνση θα αντικατασταθεί με την καινούργια. Όταν ο χρήστης πατήσει το κουμπί αλλαγής, η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και αλλάζει τη διεύθυνση ή τη διαγράφει. Ο χρήστης ανακατευθύνεται αυτόματα στην Δραστηριότητα Λογαριασμού στην καρτέλα με τα στοιχεία του λογαριασμού και εμφανίζεται ένα μήνυμα επιτυχίας όπως στην παραπάνω εικόνα 40.



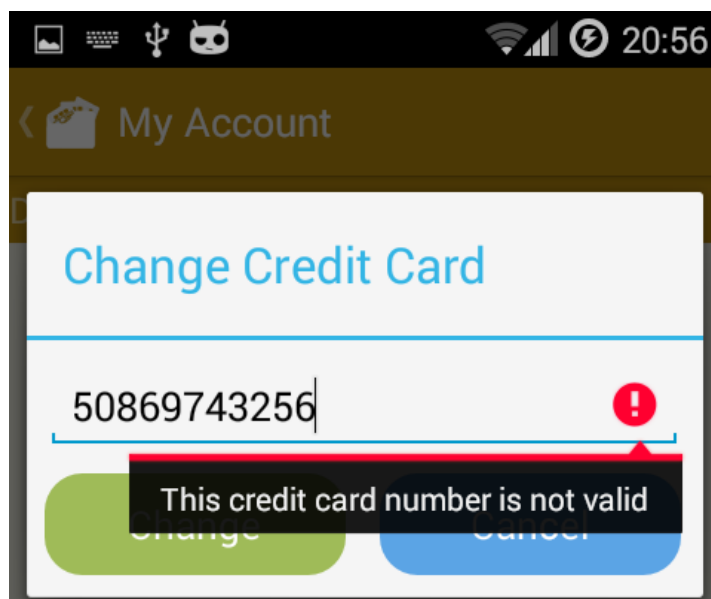
Εικόνα 41 – Δραστηριότητα Λογαριασμού Νο.6

Η ίδια διαδικασία πραγματοποιείται αν ο χρήστης πατήσει το κουμπί αλλαγής τηλεφώνου (βλέπε εικόνα 42). Η μόνη αλλαγή που υπάρχει εδώ είναι ότι αν το πεδίο δεν είναι κενό γίνεται έλεγχος για το αν το τηλέφωνο είναι έγκυρο ή όχι. Αν το τηλέφωνο δεν είναι έγκυρο τότε ο χρήστης πρέπει να πληκτρολογήσει ένα νέο τηλέφωνο, αλλιώς η εφαρμογή επικοινωνεί με τη Βάση Δεδομένων και πραγματοποιεί την αλλαγή τηλεφώνου.



Εικόνα 42 – Δραστηριότητα Λογαριασμού Νο.7

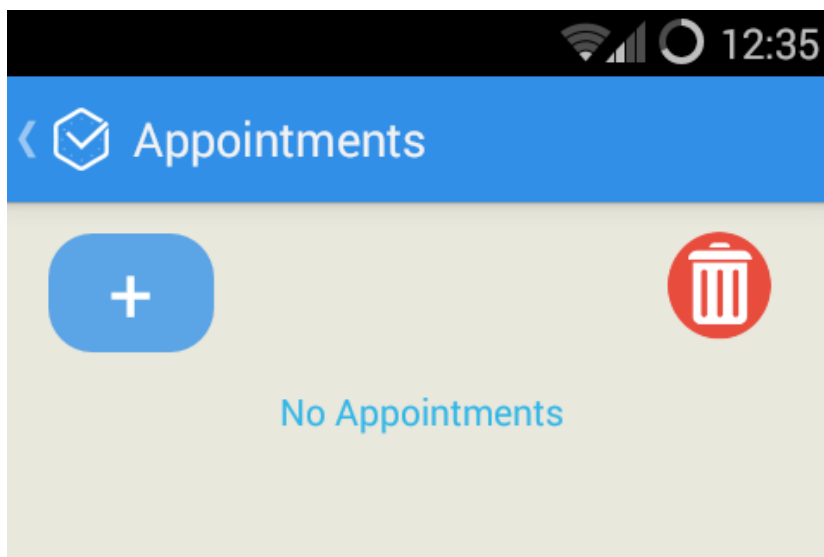
Αν ο χρήστης επιλέξει να αλλάξει τον αριθμό πιστωτικής κάρτας (βλέπε εικόνα 43) τότε επαναλαμβάνεται ακριβώς η ίδια διαδικασία με την διαδικασία αλλαγής του τηλεφώνου.



Εικόνα 43 – Δραστηριότητα Λογαριασμού Νο.8

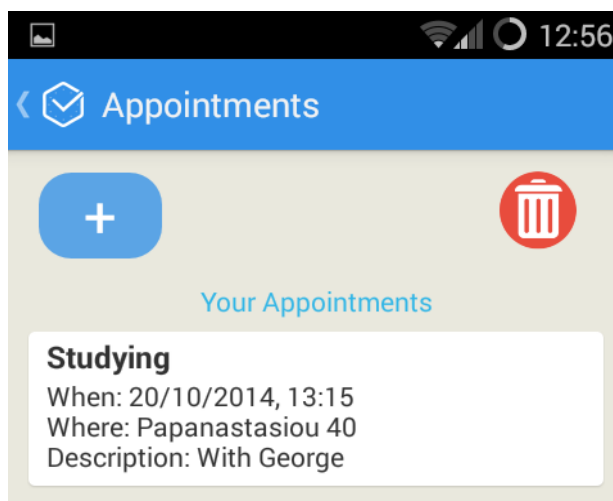
Από την Αρχική Δραστηριότητα της εφαρμογής (βλέπε εικόνα 31), αν ο χρήστης πατήσει το κουμπί αποσύνδεσης τότε ο χρήστης αποσυνδέεται από την εφαρμογή και δεν μπορεί πλέον να διαχειριστεί τον λογαριασμό του. Μπορεί όμως να συνεχίζει να χρησιμοποιεί τις υπόλοιπες Δραστηριότητες της εφαρμογής κανονικά.

Για να δει ο χρήστης τα ραντεβού του, από την Αρχική Δραστηριότητα της εφαρμογής (βλέπε εικόνα 26) πατάει τη Δραστηριότητα Ραντεβού και μεταβαίνει σε αυτήν (βλέπε εικόνα 44).



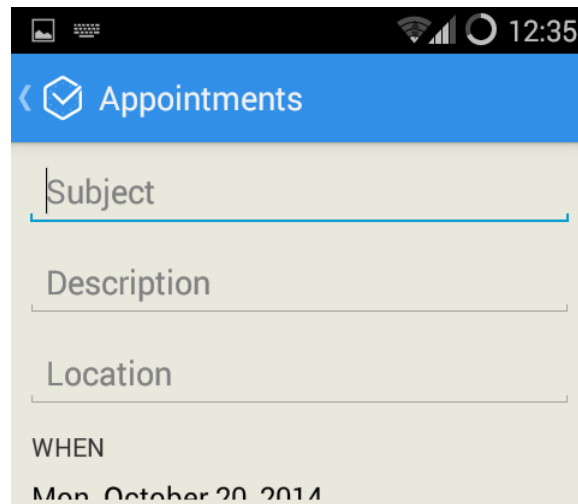
Εικόνα 44 – Δραστηριότητα Ραντεβού Νο.1

Εδώ ο χρήστης μπορεί να δει όλα τα ραντεβού του. Όταν δεν υπάρχουν ραντεβού εμφανίζεται ένα κείμενο το οποίο αναγράφει ότι δεν υπάρχουν ραντεβού (βλέπε εικόνα 44). Όταν υπάρχουν ραντεβού, το κείμενο αυτό αλλάζει και αναγράφει πλέον “τα ραντεβού σας” και ο χρήστης βλέπει όλα τα ραντεβού του (βλέπε εικόνα 45).

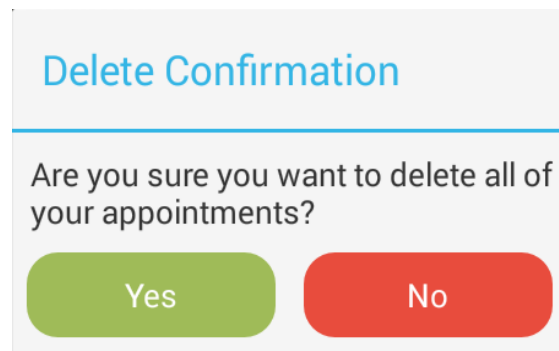


Εικόνα 45 – Δραστηριότητα Ραντεβού Νο.2

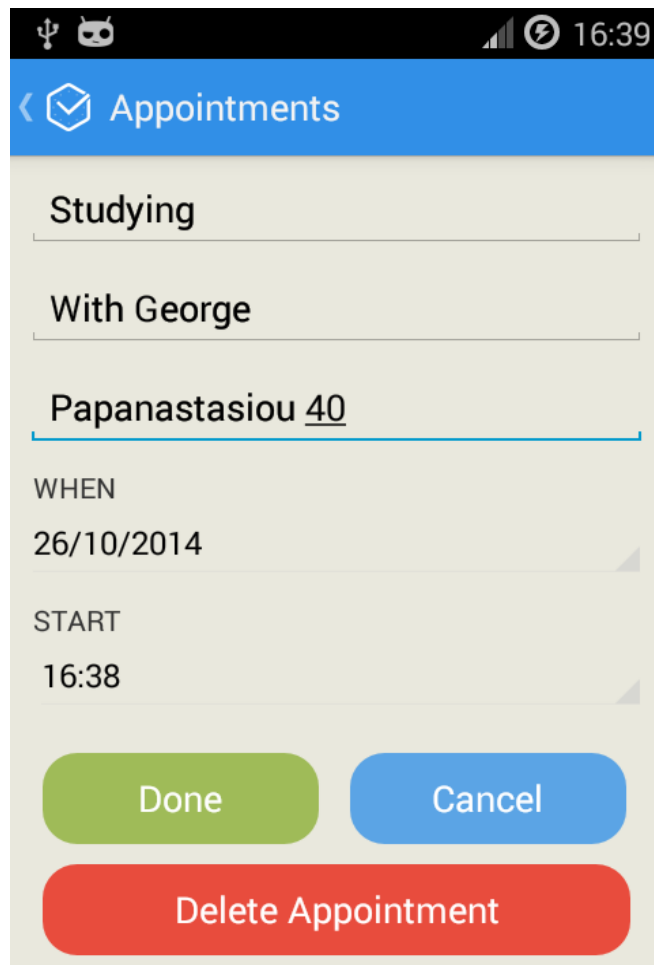
Από αυτή τη Δραστηριότητα, αν ο χρήστης πατήσει το κουμπί συν “+” ανακατευθύνεται στη Δραστηριότητα Δημιουργίας Ραντεβού όπου προσθέτει ένα ραντεβού (βλέπε εικόνα 46). Αν ο χρήστης πατήσει το κουμπί με τον κάδο ως εικόνα, τότε εμφανίζεται ένα παράθυρο το οποίο ρωτάει τον χρήστη αν θέλει να διαγράψει όλα τα ραντεβού του (βλέπε εικόνα 47). Αν ο χρήστης πατήσει το κουμπί ναι τότε όλα τα ραντεβού διαγράφονται από τη συσκευή, αν πατήσει το κουμπί όχι δεν συμβαίνει τίποτα. Επίσης, αν ο χρήστης πατήσει πάνω σε ένα ραντεβού, τότε ανακατευθύνεται στη Δραστηριότητα Δημιουργίας Ραντεβού και από εκεί μπορεί να τροποποιήσει το συγκεκριμένο ραντεβού ή να το διαγράψει (βλέπε εικόνα 48).



Εικόνα 46 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.1



Εικόνα 47 – Δραστηριότητα Ραντεβού Νο.3



Εικόνα 48 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.2

Στη Δραστηριότητα Δημιουργίας Ραντεβού, στη περίπτωση που ο χρήστης έχει επιλέξει να τροποποιήσει ένα συγκεκριμένο ραντεβού (βλέπε εικόνα 48), κατά την τροποποίηση πρέπει ο χρήστης υποχρεωτικά να μην αφήσει κενό το πρώτο πεδίο και το τρίτο, το πεδίο με το θέμα του ραντεβού και το πεδίο με την τοποθεσία του ραντεβού αντίστοιχα. Αν αφήσει τα δύο πεδία αυτά κενά εμφανίζεται ένα κόκκινο θαυμαστικό δίπλα από αυτά μαζί με ένα μήνυμα λάθους.

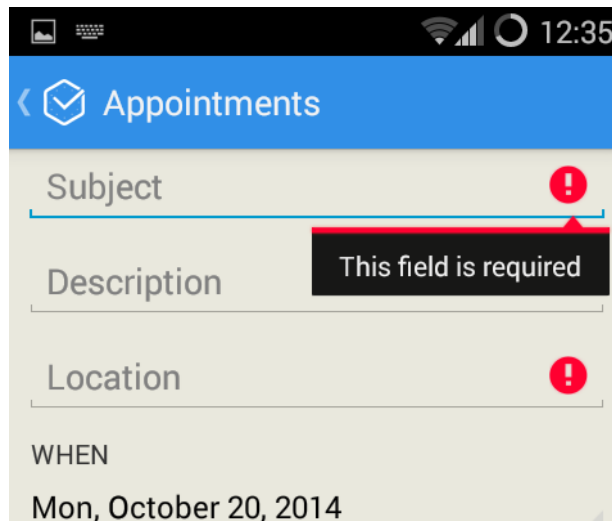
Επίσης δεν πρέπει να βάλει ημερομηνία ραντεβού μικρότερη από την τρέχουσα ημερομηνία αλλιώς εμφανίζονται μηνύματα λάθους. Τέλος, αν η ημερομηνία του ραντεβού είναι η ίδια με τη τρέχουσα ημερομηνία, τότε η ώρα του ραντεβού δεν πρέπει να είναι μικρότερη από την τρέχουσα ώρα. Αν την ημέρα που εισάγει ο χρήστης ένα ραντεβού υπάρχουν και άλλα ραντεβού, τότε η ώρα του ραντεβού αυτού δεν πρέπει να συμπίπτει με την ώρα των άλλων ραντεβού κατά 30 λεπτά πριν ή μετά.

Αν ο χρήστης πατήσει το κουμπί διαγραφής ραντεβού, τότε διαγράφεται το συγκεκριμένο ραντεβού και ο χρήστης ανακατευθύνεται αυτόματα στη Δραστηριότητα Ραντεβού (βλέπε εικόνα 44) με το ραντεβού να μην εμφανίζεται στη λίστα.

Στη Δραστηριότητα Δημιουργίας Ραντεβού, στη περίπτωση που ο χρήστης έχει πατήσει το κουμπί συν "+" (βλέπε εικόνα 46), ο χρήστης πρέπει να συμπληρώσει υποχρεωτικά τα πεδία θέμα και τοποθεσία και προαιρετικά το πεδίο περιγραφή. Αν δεν συμπληρώσει τα πεδία θέμα και τοποθεσία τότε εμφανίζεται το κόκκινο θαυμαστικό με ένα μήνυμα λάθους δίπλα από κάθε πεδίο. Επίσης, το πεδίο τοποθεσία πρέπει να περιλαμβάνει μια υπαρκτή τοποθεσία γιατί χρησιμοποιείται αργότερα στη Δραστηριότητα Σταθμός, που περιλαμβάνει τα βενζινάδικα και τα σούπερ μάρκετ, για τον υπολογισμό του κοντινότερου σταθμού (βενζινάδικο – σούπερ μάρκετ).

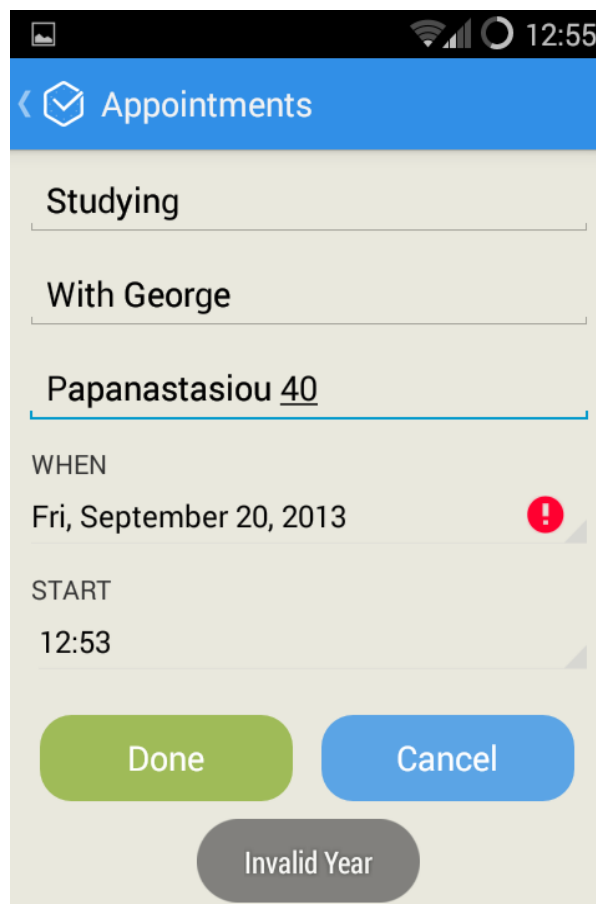
Επίσης, ο χρόνος, ο μήνας και η ημέρα που βάζει ο χρήστης πρέπει να είναι μεγαλύτερα από την τρέχουσα ημερομηνία αλλιώς εμφανίζονται μηνύματα λάθους. Ακόμα, η ώρα του ραντεβού πρέπει να

είναι μεγαλύτερη από την τρέχουσα ώρα στη περίπτωση που η ημερομηνία του ραντεβού είναι ίδια με τη τρέχουσα ημερομηνία. Τέλος, η ώρα του ραντεβού δεν πρέπει να συμπίπτει με την ώρα οποιουδήποτε άλλου ραντεβού την ίδια ημερομηνία κατά 30 λεπτά πριν ή μετά. Κάποια από τα σφάλματα που περιγράφηκαν παραπάνω φαίνονται στις παρακάτω εικόνες 49, 50 και 51.



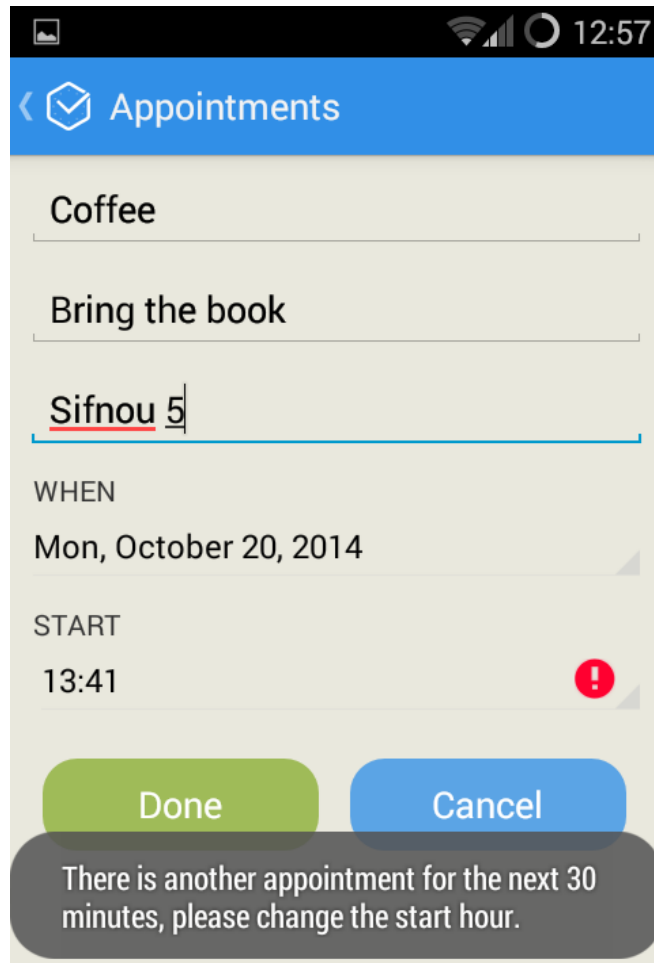
The screenshot shows the 'Appointments' app interface. At the top, there's a blue header with a back arrow and the title 'Appointments'. Below it, there are three input fields: 'Subject', 'Description', and 'Location'. The 'Subject' and 'Location' fields have a red exclamation mark icon to their right, indicating a validation error. A black tooltip with white text 'This field is required' is positioned over the 'Description' field. Below the input fields, there's a section labeled 'WHEN' with the date 'Mon, October 20, 2014'.

Εικόνα 49 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.3



The screenshot shows the 'Appointments' app interface. At the top, there's a blue header with a back arrow and the title 'Appointments'. Below it, there are three input fields: 'Studying', 'With George', and 'Papanastasiou 40'. Below the input fields, there's a section labeled 'WHEN' with the date 'Fri, September 20, 2013' and a red exclamation mark icon to its right. Below that, there's a section labeled 'START' with the time '12:53'. At the bottom, there are three buttons: 'Done' (green), 'Cancel' (blue), and 'Invalid Year' (grey).

Εικόνα 50 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.4



Εικόνα 51 – Δραστηριότητα Δημιουργίας Ραντεβού Νο.5

Όσον αφορά την εμφάνιση των βενζινάδικων - σούπερ μάρκετ και των πληροφοριών τους, διαχειρίζονται από μία μόνο Δραστηριότητα, τη Δραστηριότητα Σταθμού. Από την Αρχική Δραστηριότητα της εφαρμογής, επιλέγοντας ο χρήστης την κατηγορία βενζινάδικα ή σούπερ μάρκετ ανακατευθύνεται στη Δραστηριότητα Σταθμού. Σε αυτή τη Δραστηριότητα γίνεται ο διαχωρισμός με βάση μια κατηγορία επέλεξε ο χρήστης στην Αρχική Δραστηριότητα. Η Δραστηριότητα Σταθμού υλοποιήθηκε με αυτό τον τρόπο διότι η διαδικασία εύρεσης του κοντινότερου βενζινάδικου ή σούπερ μάρκετ είναι ίδια, το μόνο που διαφέρει είναι οι παραλλαγές των μηνυμάτων που εμφανίζονται για το καθένα.

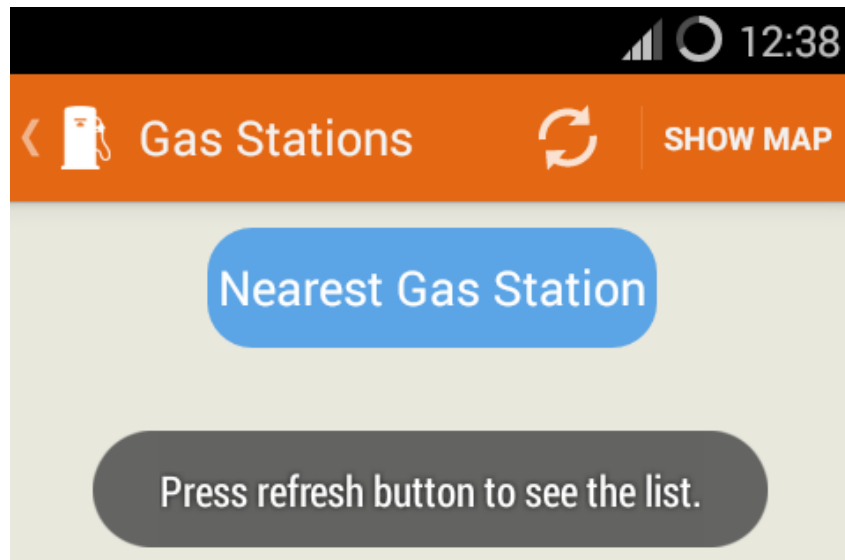
Οπότε, στην ουσία έχουμε μία Δραστηριότητα και έναν αλγόριθμο ο οποίος κατεβάζει από τον Διακομιστή (Server) τα απαραίτητα αρχεία και εμφανίζει τους σταθμούς (βενζινάδικα – σούπερ μάρκετ) με τις πληροφορίες τους. Η ίδια διαδικασία συμβαίνει με την Δραστηριότητα Χάρτη, όπου χρησιμοποιώντας τα αρχεία που κατέβασε η προηγούμενη Δραστηριότητα βρίσκει τον πλησιέστερο σταθμό (βενζινάδικο – σούπερ μάρκετ) από την τοποθεσία του χρήστη, ανάλογα με τα ραντεβού που έχει εκείνη τη στιγμή, το ωράριο των καταστημάτων και την κυκλοφορία στους δρόμους.

Έχοντας μία Δραστηριότητα Σταθμού για την εμφάνιση των σταθμών (βενζινάδικα – σούπερ μάρκετ) και μία Δραστηριότητα Χάρτη για την εύρεση του πλησιέστερου σταθμού (βενζινάδικα – σούπερ μάρκετ), παρακάτω θα δείξουμε πως δουλεύουν αυτές οι δύο Δραστηριότητες παίρνοντας ως παράδειγμα την εύρεση του πλησιέστερου βενζινάδικου, διότι είναι ακριβώς η ίδια διαδικασία με την εύρεση του πλησιέστερου σούπερ μάρκετ.

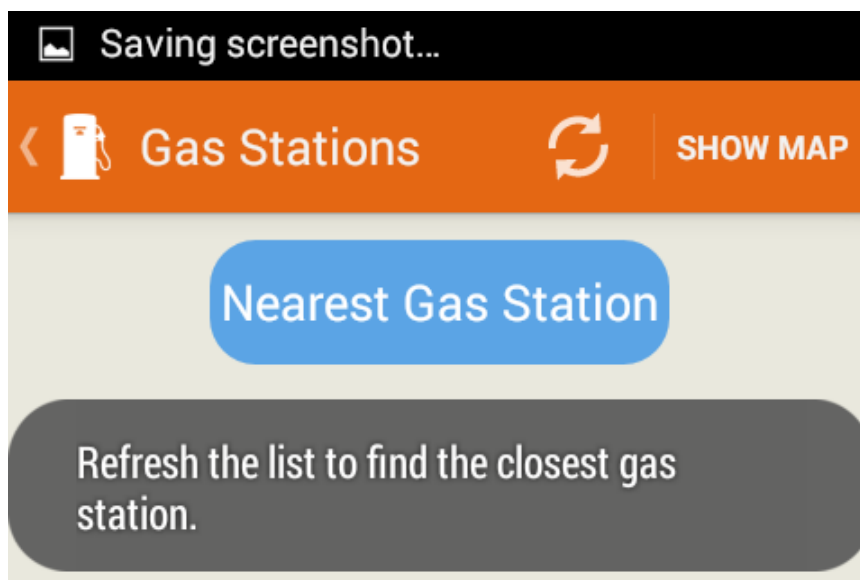
Όταν ο χρήστης επιλέγει στην Αρχική Δραστηριότητα την κατηγορία βενζινάδικα, ανακατευθύνεται στη Δραστηριότητα Σταθμού. Αν ο χρήστης έχει επιλέξει αυτή τη κατηγορία για πρώτη φορά τότε του εμφανίζεται μήνυμα ότι πρέπει να πατήσει το κουμπί ανανέωσης έτσι ώστε να του εμφανιστούν τα βενζινάδικα στην οθόνη (βλέπε εικόνα 52). Εκείνη τη στιγμή, ο χρήστης έχει τη

δυνατότητα να επιλέξει επίσης την εμφάνιση του χάρτη και την εύρεση του πλησιέστερου βενζινάδικου.

Αν τα βενζινάδικα δεν είναι ορατά στην οθόνη, δηλαδή ο χρήστης δεν έχει πατήσει το κουμπί ανανέωσης για να επικοινωνήσει η εφαρμογή με τον Διακομιστή (Server) και να κατεβάσει το αρχείο με τα βενζινάδικα, τότε το πάτημα του κουμπιού εύρεσης του πλησιέστερου βενζινάδικου υπενθυμίζει στο χρήστη ότι πρέπει να πατήσει το κουμπί ανανέωσης και δεν προχωράει περαιτέρω (βλέπε εικόνα 53). Το πάτημα του κουμπιού του χάρτη τον ανακατευθύνει στη Δραστηριότητα Χάρτη αλλά δεν δείχνει τα βενζινάδικα αφού ο χρήστης δεν έχει πατήσει αρχικά το κουμπί ανανέωσης.

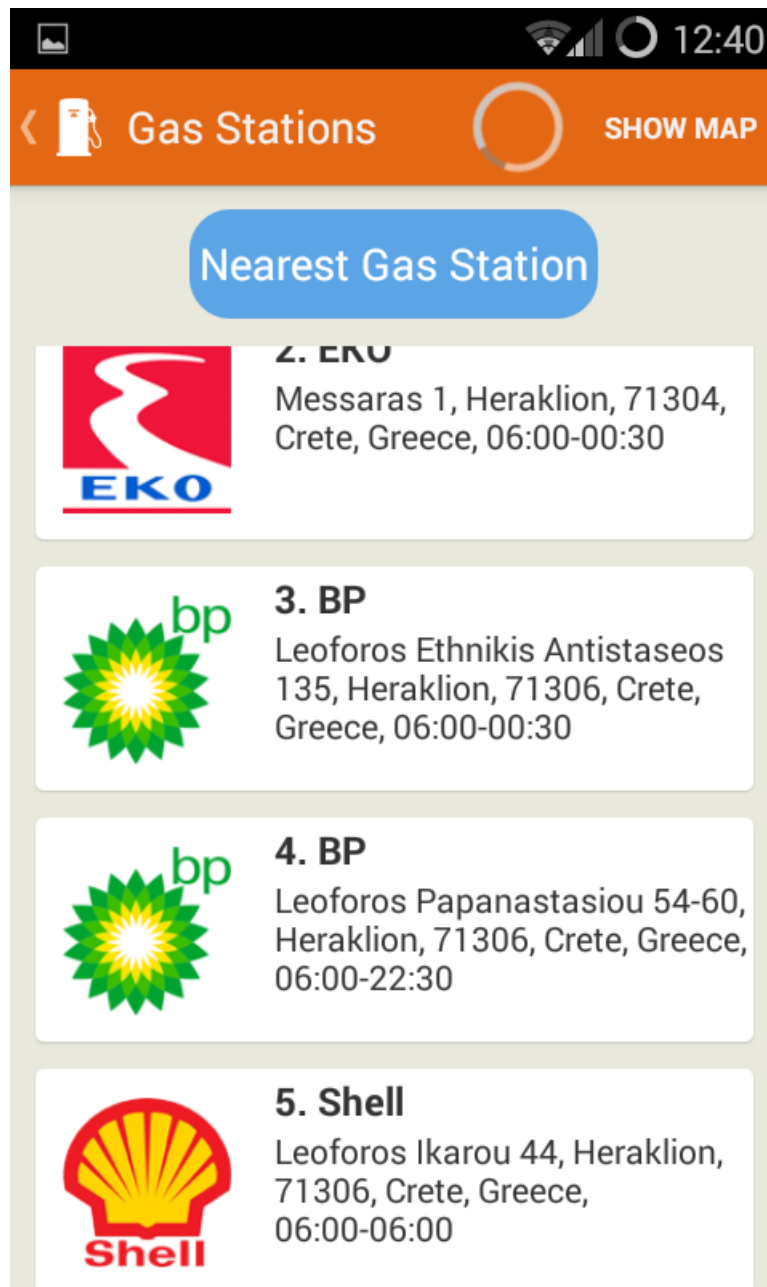


Εικόνα 52 – Δραστηριότητα Σταθμού Νο.1



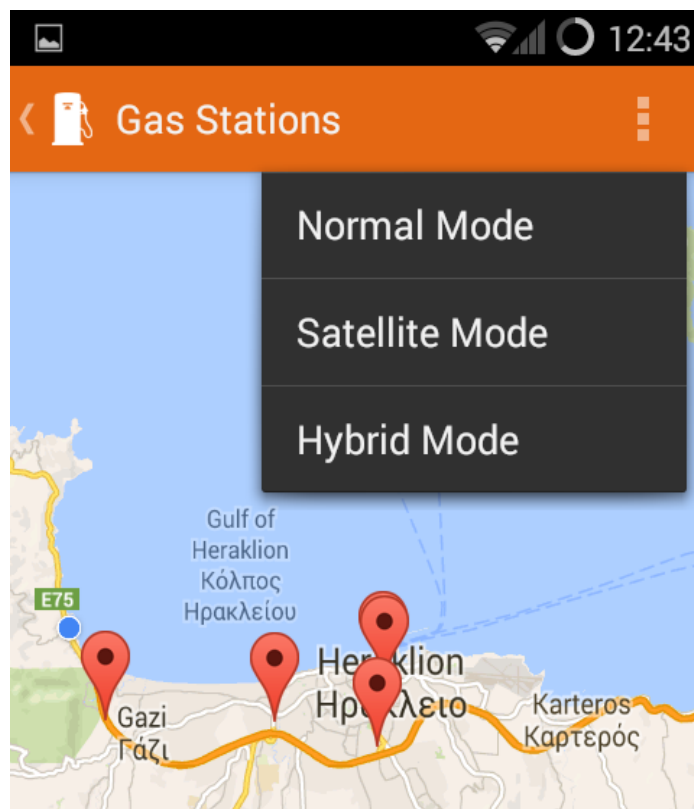
Εικόνα 53 – Δραστηριότητα Σταθμού Νο.2

Όταν ο χρήστης πατήσει το κουμπί ανανέωσης, αν υπάρχει σύνδεση στο δίκτυο επικοινωνεί η εφαρμογή με τον Διακομιστή (Server) και κατεβάζει το αρχείο XML με όλα τα βενζινάδικα, στη συνέχεια τα βενζινάδικα εμφανίζονται στην οθόνη της συσκευής (βλέπε εικόνα 54). Αν δεν υπάρχει σύνδεση στο δίκτυο τότε εμφανίζεται το κατάλληλο μήνυμα.



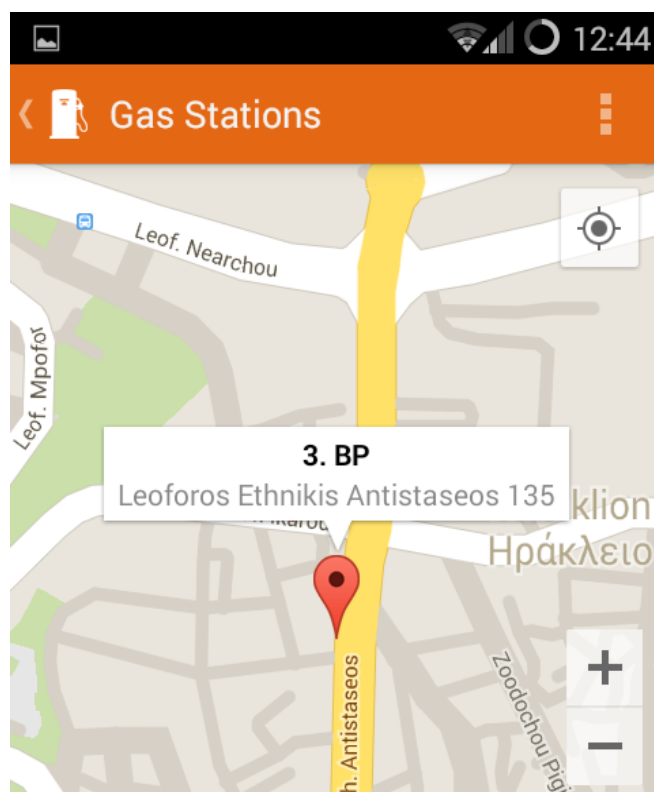
Εικόνα 54 – Δραστηριότητα Σταθμού Νο.3

Στη συνέχεια, αν ο χρήστης πατήσει το κουμπί του χάρτη ανακατευθύνεται στη Δραστηριότητα Χάρτη και μπορεί να δει τον χάρτη της πόλης με όλα τα βενζινάδικα. Ο χρήστης έχει την επιλογή να δει τον χάρτη σε κανονική μορφή, σε δορυφορική μορφή αλλά και σε υβριδική μορφή. Ο χάρτης με όλα τα βενζινάδικα αλλά και οι μορφές του χάρτη που είναι διαθέσιμες στο χρήστη για να επιλέξει φαίνονται στη παρακάτω εικόνα 55.



Εικόνα 55 – Δραστηριότητα Χάρτη No.1

Επίσης, ο χρήστης μπορεί να επιλέξει ένα βενζινάδικο από τη λίστα της Δραστηριότητας Σταθμού. Ο χρήστης ανακατευθύνεται πάλι στη Δραστηριότητα Χάρτη, όπου εμφανίζεται ο χάρτης της πόλης με όλα τα βενζινάδικα αλλά ο χάρτης είναι εστιασμένος στο βενζινάδικο που επέλεξε ο χρήστης νωρίτερα (βλέπε εικόνα 56).

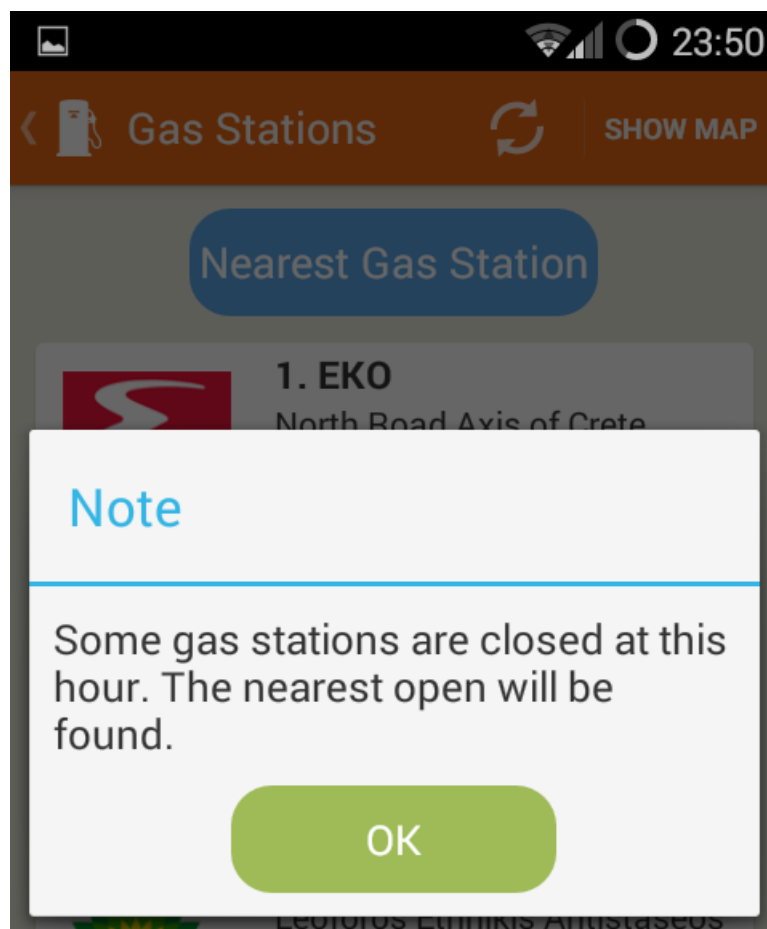


Εικόνα 56 – Δραστηριότητα Χάρτη No.2

Ο χρήστης μπορεί οποιαδήποτε στιγμή να πατήσει το κουμπί ανανέωσης για να ανανεωθεί η λίστα με τα βενζινάδικα της συσκευής και να συγχρονιστεί με τη λίστα των βενζινάδικων του Διακομιστή (Server). Με αυτό τον τρόπο η λίστα των βενζινάδικων θα είναι πάντα ενημερωμένη.

Αν ο χρήστης επιλέξει να πατήσει το κουμπί εύρεσης πλησιέστερου βενζινάδικου, αρχικά γίνεται έλεγχος για το αν το GPS είναι ενεργοποιημένο. Αν δεν είναι ενεργοποιημένο το GPS εμφανίζεται κατάλληλο μήνυμα στον χρήστη, αλλιώς πραγματοποιείται έλεγχος με βάση την ώρα που πάτησε ο χρήστης αυτό το κουμπί. Ο έλεγχος αυτός πραγματοποιείται για να εμφανίσει στον χρήστη μόνο τα βενζινάδικα τα οποία είναι ανοιχτά εκείνη την ώρα. Αν δεν υπάρχει κανένα βενζινάδικο ανοιχτό εκείνη την ώρα, τότε δεν πραγματοποιείται η ανακατεύθυνση του χρήστη στη Δραστηριότητα Χάρτη και εμφανίζει μήνυμα ότι όλα τα βενζινάδικα είναι κλειστά αυτή την ώρα.

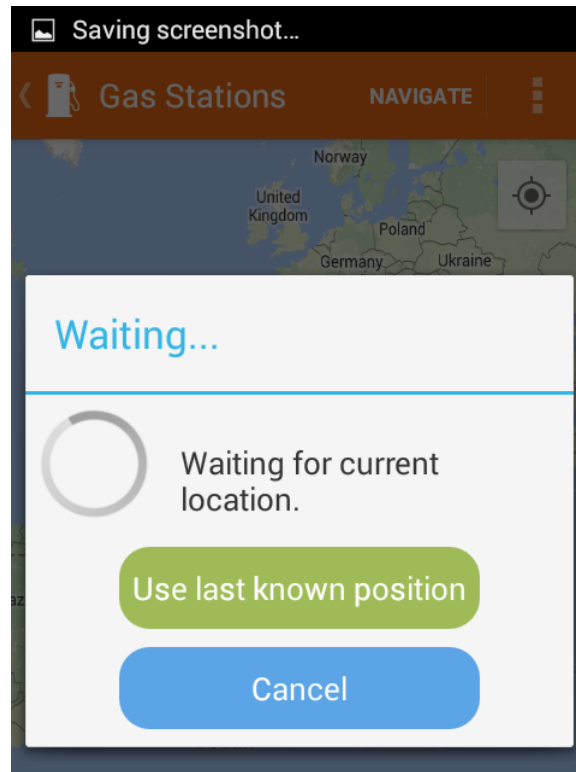
Αν όμως κάποια από τα βενζινάδικα είναι ακόμα ανοιχτά, τότε ο χρήστης ανακατευθύνεται στη Δραστηριότητα Χάρτη και γίνεται ο υπολογισμός του πλησιέστερου βενζινάδικου με βάση αυτά που είναι ανοιχτά εκείνη την ώρα. Ένα τέτοιο παράδειγμα φαίνεται στην παρακάτω εικόνα 57.



Εικόνα 57 – Δραστηριότητα Σταθμού Νο.4

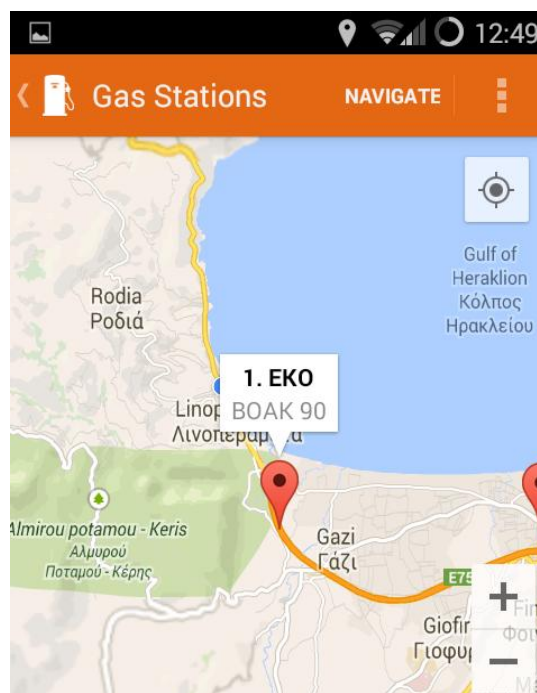
Εφόσον ο χρήστης έχει φτάσει στη Δραστηριότητα Χάρτη, με όλα τα βενζινάδικα ανοιχτά ή μερικά από αυτά ανοιχτά, η εφαρμογή περιμένει να βρει την τοποθεσία του χρήστη (βλέπε εικόνα 58). Αν ο χρήστης θέλει μπορεί να επιλέξει να χρησιμοποιηθεί η τελευταία του τοποθεσία που είναι γνωστή στην εφαρμογή. Αν η τελευταία τοποθεσία του χρήστη δεν είναι γνωστή στην εφαρμογή, εμφανίζεται ένα μήνυμα στο χρήστη ότι η εφαρμογή δεν γνωρίζει την τελευταία τοποθεσία του και εξακολουθεί να περιμένει να βρει την τρέχουσα τοποθεσία του χρήστη. Αν η τελευταία τοποθεσία του χρήστη είναι γνωστή στην εφαρμογή, τότε υπολογίζεται το πλησιέστερο βενζινάδικο με βάση αυτή.

Στην περίπτωση που ο χρήστης δεν επιλέξει να χρησιμοποιηθεί η τελευταία του γνωστή τοποθεσία, τότε η εφαρμογή περιμένει να βρεθεί η τρέχουσα τοποθεσία του χρήστη και να χρησιμοποιηθεί αυτή η τοποθεσία για την εύρεση του πλησιέστερου βενζινάδικου.

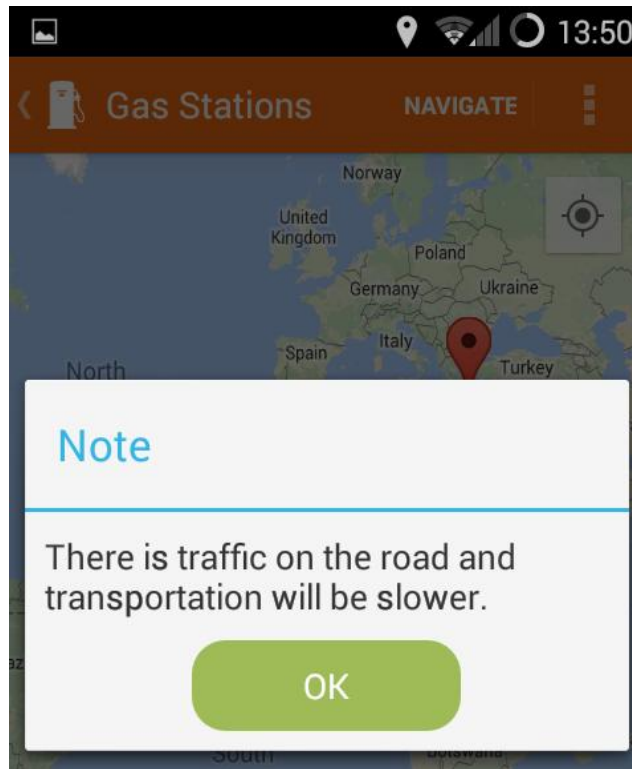


Εικόνα 58 – Δραστηριότητα Χάρτη Νο.3

Εφόσον η εφαρμογή έχει καταλήξει ποια τοποθεσία θα χρησιμοποιήσει, τρέχουσα ή τελευταία γνωστή τοποθεσία, πραγματοποιείται έλεγχος για το αν ο χρήστης έχει κάποιο ραντεβού την ώρα που πάτησε να βρεθεί το πλησιέστερο βενζινάδικο, επίσης γίνεται έλεγχος για το αν υπάρχει κυκλοφοριακό πρόβλημα στους δρόμους. Στην απλή περίπτωση όπου ο χρήστης δεν έχει κανένα ραντεβού εκείνη την ώρα, αν δεν υπάρχει κυκλοφοριακό πρόβλημα στους δρόμους τότε ο χάρτης εστιάζει στο βενζινάδικο που βρέθηκε ως το πλησιέστερο. Αν υπάρχει κυκλοφοριακό πρόβλημα, τότε εμφανίζεται κατάλληλο μήνυμα για την ενημέρωση του χρήστη και ύστερα εστιάζει ο χάρτης στο πλησιέστερο βενζινάδικο. Όλα αυτά φαίνονται στις παρακάτω εικόνες 59 και 60.

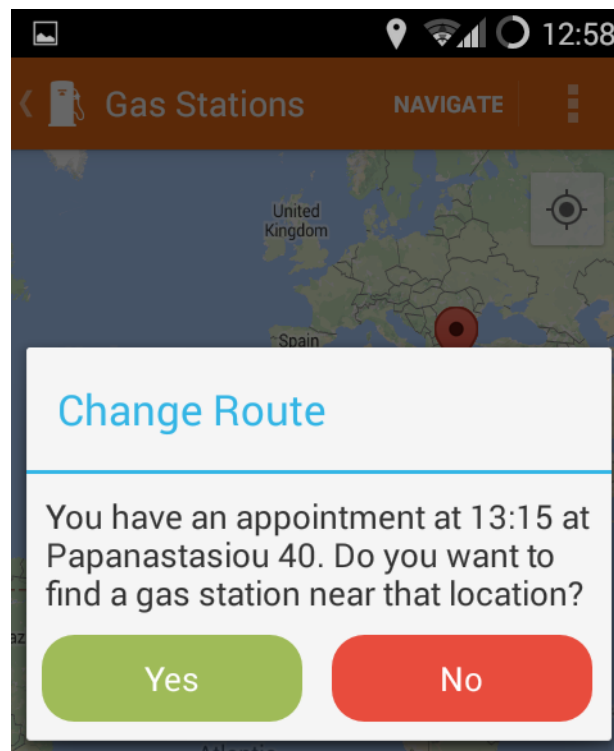


Εικόνα 59 – Δραστηριότητα Χάρτη Νο.4

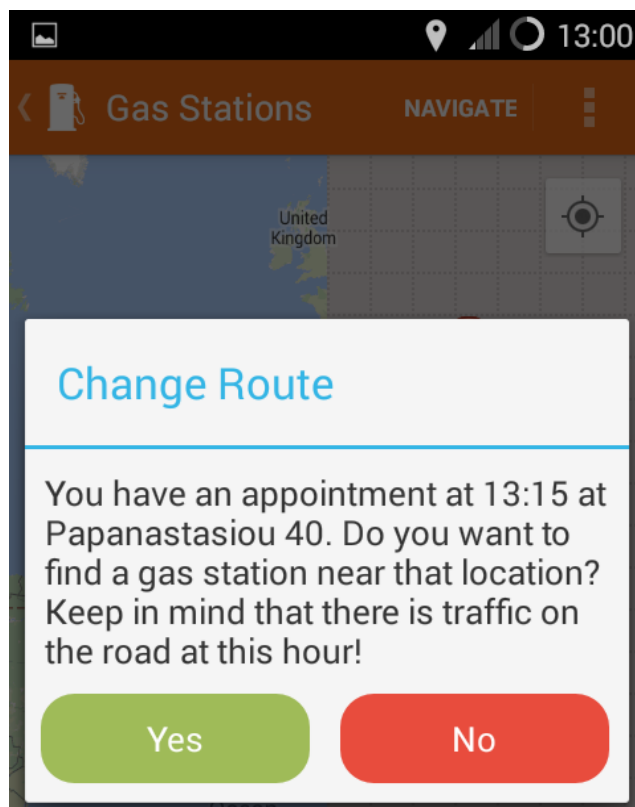


Εικόνα 60 – Δραστηριότητα Χάρτη Νο.5

Στην περίπτωση που ο χρήστης έχει ραντεβού εκείνη την ώρα, τότε η εφαρμογή ρωτάει τον χρήστη αν θέλει να βρεθεί το πλησιέστερο βενζινάδικο κοντά στην τοποθεσία του ραντεβού του χρήστη. Αν ο χρήστης πατήσει το κουμπί ναι, η εφαρμογή θα ψάξει για το πλησιέστερο βενζινάδικο στην τοποθεσία του ραντεβού του. Αν ο χρήστης πατήσει το κουμπί όχι, η εφαρμογή θα βρει το πλησιέστερο βενζινάδικο στην τοποθεσία του χρήστη. Επίσης, γίνεται έλεγχος για κυκλοφοριακό πρόβλημα στους δρόμους και ενημερώνεται ο χρήστης με κατάλληλο μήνυμα. (βλέπε εικόνες 61, 62).

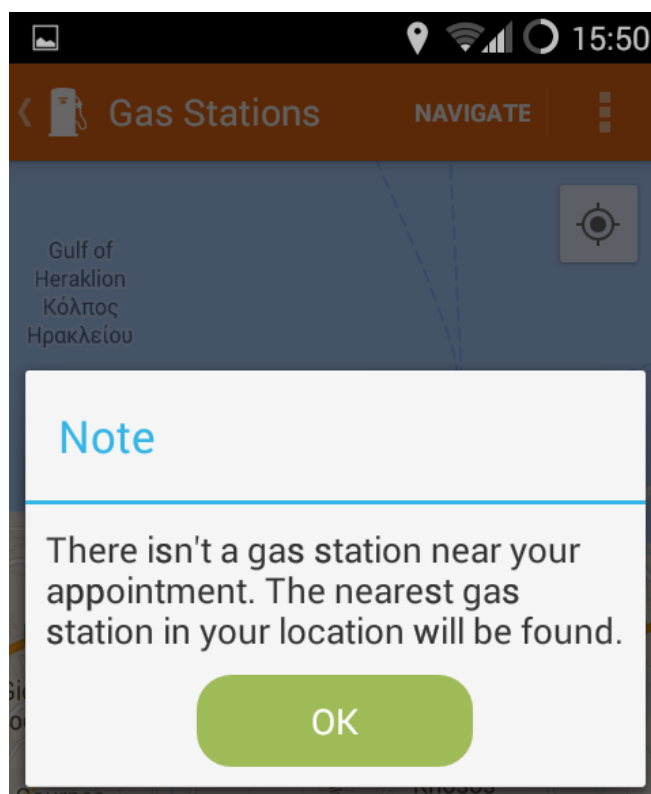


Εικόνα 61 – Δραστηριότητα Χάρτη Νο.6



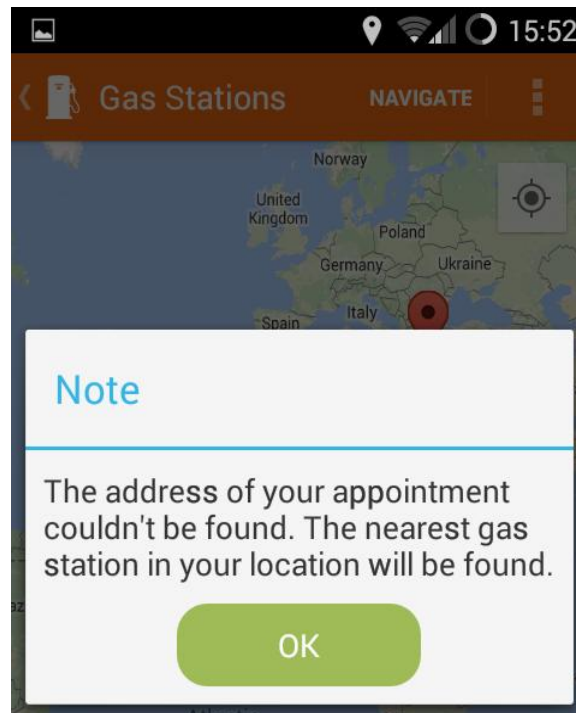
Εικόνα 62 – Δραστηριότητα Χάρτη Νο.7

Στη περίπτωση που ο χρήστης πατήσει το κουμπί ναι, αν στην τοποθεσία του ραντεβού του χρήστη δεν υπάρχει βενζινάδικο σε ακτίνα 2.5 km τότε εμφανίζεται στον χρήστη μήνυμα ότι δεν υπάρχει βενζινάδικο κοντά στην τοποθεσία του ραντεβού του και θα βρεθεί το πλησιέστερο στην τοποθεσία του χρήστη (βλέπε εικόνα 63).



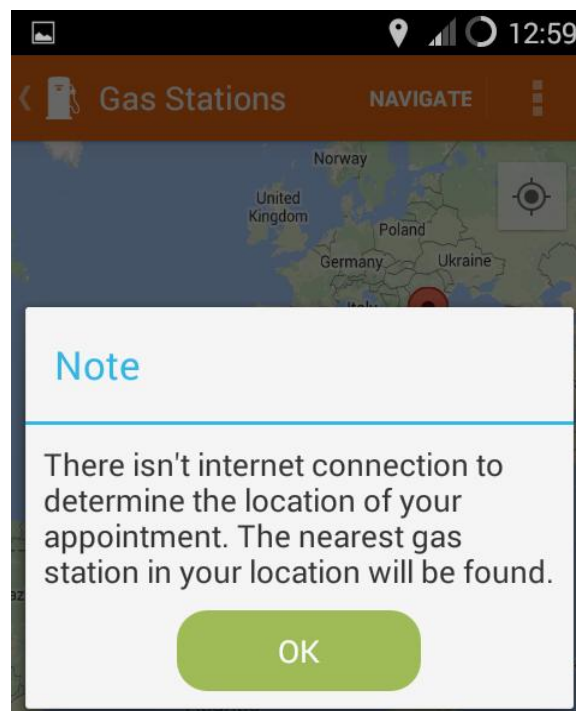
Εικόνα 63 – Δραστηριότητα Χάρτη Νο.8

Άλλη μία περίπτωση είναι η εφαρμογή να μην μπορεί να προσδιορίσει την τοποθεσία του ραντεβού του χρήστη και τον ενημερώνει με μήνυμα ότι θα βρεθεί το πλησιέστερο βενζινάδικο στην τοποθεσία του (βλέπε εικόνα 64). Σε αυτή τη περίπτωση, πιθανό σφάλμα είναι ο χρήστης να μην έχει δώσει έγκυρη διεύθυνση όταν πρόσθετε το συγκεκριμένο ραντεβού στην λίστα με τα ραντεβού του.



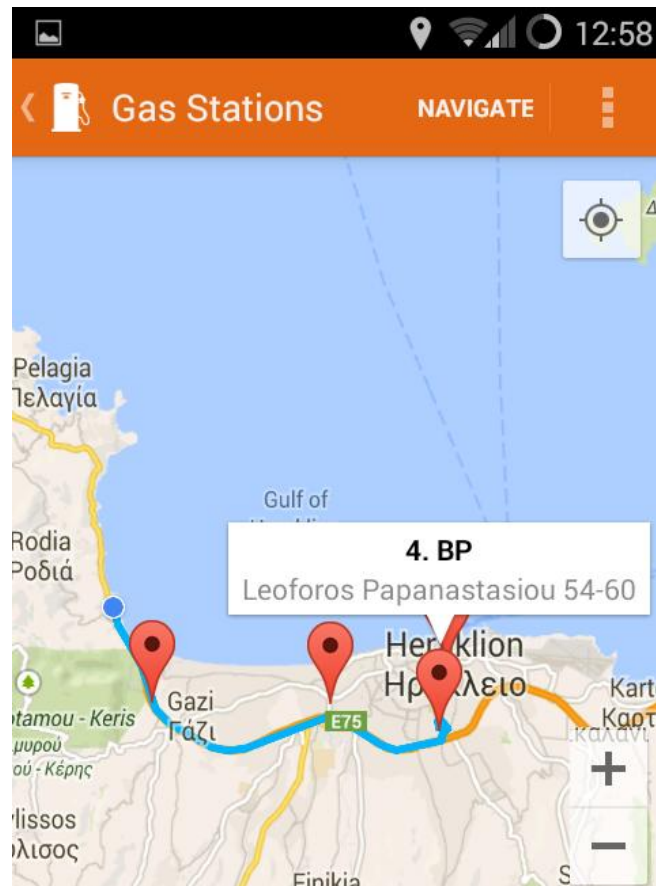
Εικόνα 64 – Δραστηριότητα Χάρτη No.9

Η τελευταία περίπτωση είναι να μην είναι η συσκευή συνδεδεμένη στο δίκτυο έτσι ώστε να μην μπορεί να προσδιοριστή η διεύθυνση του ραντεβού του χρήστη. Σε αυτή τη περίπτωση εμφανίζεται μήνυμα στον χρήστη ότι θα βρεθεί το πλησιέστερο βενζινάδικο στην τοποθεσία του και όχι στην τοποθεσία του ραντεβού του (βλέπε εικόνα 65).



Εικόνα 65 – Δραστηριότητα Χάρτη No.10

Εφόσον έχει βρεθεί ένα βενζινάδικο ως το πλησιέστερο, ο χάρτης εστιάζει σε αυτό. Έπειτα, ο χρήστης μπορεί να πατήσει το κουμπί πλοήγησης έτσι ώστε ο χάρτης να του δείξει τον δρόμο για να κατευθυνθεί σε αυτό. Όμως, αν δεν υπάρχει δίκτυο στην συσκευή τότε η καθοδήγηση του χρήστη στο πλησιέστερο βενζινάδικο δεν είναι εφικτή και εμφανίζεται κατάλληλο μήνυμα στην οθόνη της συσκευής. Αλλιώς, η εφαρμογή επικοινωνεί με του Διακομιστές (Servers) της Google για να βρεθεί ο συντομότερος δρόμος για το πλησιέστερο βενζινάδικο και αποτυπώνεται στην οθόνη της συσκευής. (βλέπε εικόνα 66).



Εικόνα 66 – Δραστηριότητα Χάρτη Νο.11

Κατά τη διάρκεια της πλοήγησης, ο χάρτης είναι εστιασμένος στην τοποθεσία του χρήστη. Ακόμα και αν ο χρήστης αλλάξει την εστίαση του χάρτη, ο χάρτης θα ξανά εστιάσει στην τοποθεσία του χρήστη.

5 Αποτελέσματα

Με την ολοκλήρωση αυτής της εφαρμογής και κατά συνέπεια της συγκεκριμένης πτυχιακής εργασίας αποκομίσαμε γνώσεις αλλά και εμπειρία πάνω στην ανάπτυξη εφαρμογών για έξυπνες συσκευές και ιδιαίτερα για συσκευές με λειτουργικό σύστημα Android. Δημιουργώντας αυτήν την εφαρμογή, πραγματοποιήθηκε έρευνα γύρω από το αντικείμενο ανάπτυξης εφαρμογών για την πλατφόρμα Android και μάθαμε καλύτερα την γλώσσα προγραμματισμού Java την οποία χρησιμοποιήσαμε για την δημιουργία της εφαρμογής μας.

5.1 Συμπεράσματα

Για την δημιουργία της εφαρμογής μας χρησιμοποιήσαμε το Eclipse ADT with Android SDK. Το Eclipse ADT with Android SDK χρησιμοποιείται ευρέως από προγραμματιστές σε όλο τον κόσμο και μπορεί να χρησιμοποιηθεί για την δημιουργία εφαρμογών Android διαφορετικών ειδών. Μέσω αυτού, μπορούν να δημιουργηθούν εφαρμογές για διαφορετικές εκδόσεις του λογισμικού Android. Βασικό πλεονέκτημά του είναι ότι βρίσκεται στο χώρο δημιουργίας εφαρμογών Android πολύ καιρό ώστε να υπάρχουν λύσεις σε οποιοδήποτε πρόβλημα μπορεί να συναντήσει ο προγραμματιστής από άλλους συναδέλφους στις διάφορες ιστοσελίδες κοινότητας Android.

5.2 Μελλοντική Εργασία και Επεκτάσεις

Για αρχή, η εργασία αυτή θα μπορούσε να βοηθήσει άλλους προγραμματιστές, κυρίως φοιτητές, που δεν έχουν ασχοληθεί στο παρελθόν με την δημιουργία εφαρμογών Android. Πιο συγκεκριμένα, αυτή η πτυχιακή εργασία μπορεί να χρησιμοποιηθεί ως ένας οδηγός για το πώς μπορεί κάποιος να αρχίσει να δημιουργεί εφαρμογές στο οικοσύστημα του Android αλλά και ποια εργαλεία μπορεί να χρησιμοποιήσει.

Όσον αφορά την επέκταση της εφαρμογής αυτής, θα μπορούσε για αρχή να προστεθούν περισσότερες δραστηριότητες καθημερινής χρήσης. Μία τέτοια καθημερινή δραστηριότητα είναι η εύρεση του κοντινότερου κινηματογράφου ανάλογα την τοποθεσία του χρήστη αλλά και την ταινία που θα ήθελε να παρακολουθήσει. Μία βελτίωση της συγκεκριμένης εφαρμογής θα μπορούσε να είναι η προεπισκόπηση της τιμής υγρών καυσίμων των βενζινάδικων και ο χρήστης να έχει την επιλογή να δρομολογείται με βάση την χαμηλότερη τιμή υγρών καυσίμων αντί για την δρομολόγηση στο πλησιέστερο βενζινάδικο. Μία ακόμα βελτίωση της εφαρμογής αυτής θα ήταν η διάθεση στον χρήστη μιας λίστας στην οποία θα μπορεί να σημειώνει τα ψώνια που θα ήθελε να πραγματοποιήσει από το σούπερ μάρκετ.

Βιβλιογραφία

1. **Google.** Android. *Development*. [Ηλεκτρονικό] <http://developer.android.com/index.html>.
2. **Wikipedia.** [Ηλεκτρονικό] http://en.wikipedia.org/wiki/Main_Page.
3. **Massachusetts Institute of Technology.** MIT App Inventor. [Ηλεκτρονικό] 2012 - 2014. <http://appinventor.mit.edu/explore/>.
4. **Intel.** Intel Developer Zone. *Intel XDK*. [Ηλεκτρονικό] <https://software.intel.com/en-us/html5/tools>.
5. **Adobe.** PhoneGap. [Ηλεκτρονικό] <http://phonegap.com/>.
6. **Xamarin.** [Ηλεκτρονικό] <http://xamarin.com/>.
7. **Android Developers Blog.** Android Development. *Blog*. [Ηλεκτρονικό] <http://android-developers.blogspot.gr/>.
8. **StackOverflow.** [Ηλεκτρονικό] <http://stackoverflow.com/>.
9. **Amal, Raj.** Learn2Crack. [Ηλεκτρονικό] <http://www.learn2crack.com/>.
10. **Tamada, Ravi.** AndroidHive. [Ηλεκτρονικό] <http://www.androidhive.info/>.
11. **Vogel, Lars.** Vogella. [Ηλεκτρονικό] <http://www.vogella.com/>.
12. **Azzola, Francesco.** SurvivingWithAndroid. [Ηλεκτρονικό] <http://www.survivingwithandroid.com/>.
13. **M, Adnan A.** MyAndroidTuts Blogspot. [Ηλεκτρονικό] <http://myandroidtuts.blogspot.gr/>.
14. **Fet, Ngewi.** Coding User. [Ηλεκτρονικό] <http://www.codinguser.com/>.
15. **Ahmad, Tofeeq Abdul.** Android Training Center Blogspot. [Ηλεκτρονικό] <http://androidtrainingcenter.blogspot.gr/>.
16. **FoamyGuy και nostra13.** GitHub. [Ηλεκτρονικό] <https://github.com/>.
17. **Mathew, George.** wpTrafficAnalyzer Blog. [Ηλεκτρονικό] <http://wptrafficanalyzer.in/blog/>.
18. **Google.** Google Developers. [Ηλεκτρονικό] <https://developers.google.com/>.
19. **Oracle.** Java. *JDK*. [Ηλεκτρονικό] <http://www.oracle.com/technetwork/articles/javase/index-jsp-138363.html>.
20. **DOT TK.** [Ηλεκτρονικό] <http://www.dot.tk/en/index.html>.
21. **000 Web Host.** [Ηλεκτρονικό] <http://www.000webhost.com/order.php>.