



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

Πτυχιακή Εργασία

Ανάπτυξη εφαρμογών γονικού ελέγχου για
κινητές συσκευές Android

Μανδαλενάκη Μαρία-Ευφραιμία

ΑΜ: 21020

Επιβλέπουσα Καθηγήτρια

Μαρία Νικολαΐδη

Μέλη της Εξεταστικής Επιτροπής

Βασίλειος Δαλάκας

Παναγιώτης Ριζομυλιώτης

Αθήνα, Ιούλιος 2014

2014

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΜΑΤΙΚΗΣ

Μανδαλενάκη Μαρία-Ευφραιμία



**ΧΑΡΟΚΟΠΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ**



[ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΩΝ ΓΟΝΙΚΟΥ ΕΛΕΓΧΟΥ ΓΙΑ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ ANDROID]

Αθήνα, Ιούλιος 2014

Επιβλέπουσα Καθηγήτρια: Νικολαΐδη Μαρία

Περιεχόμενα

Περιεχόμενα	3
Κατάλογος εικόνων	5
Κατάλογος σχημάτων	7
Ευχαριστίες	9
Πρόλογος	10
Περίληψη	11
Abstract	12
Κεφάλαιο 1	13
Εισαγωγή	13
1.1 Σκοπός της διπλωματικής	13
1.2 Έξυπνα τηλέφωνα	14
1.3 Λειτουργικά Συστήματα	14
1.4 GPS (Global Positioning System)	17
1.5 Εφαρμογές	19
Κεφάλαιο 2	23
Τεχνολογίες	23
2.1 MySQL	23
2.2 PhpMyAdmin	24
2.3 Eclipse	25
2.4 Java	26
2.5 https://oceanos.grnet.gr/	27
2.6 GlassFish	28
2.7 Netbeans	29
2.8 RESTful Web Service	30
Κεφάλαιο 3	33
Ανάλυση και Σχεδίαση	33
3.1 Αρχιτεκτονική	33
3.2 Ρόλοι του συστήματος	34
3.3 Απαιτήσεις	35

3.4 Βάση δεδομένων	36
3.5 Πλατφόρμα ανάπτυξης βάσης δεδομένων	42
3.6 Διάγραμμα οντοτήτων-συσχετίσεων	42
3.7 Προσχέδιο βάσης δεδομένων.....	43
3.8 Περιπτώσεις χρήσης	44
Κεφάλαιο 4.....	51
Εφαρμογή MapLocation.....	51
4.1 Εισαγωγικά	51
4.2 Ανατομία μίας εφαρμογής Android.....	54
4.3 Διαγράμματα λειτουργίας MapLocation	58
4.4 Προγραμματισμός-Λειτουργία της εφαρμογής	63
4.5 Στοιχεία κώδικα εφαρμογής.....	87
4.6 Στοιχεία Web Service	87
4.7 Ασφάλεια	87
Κεφάλαιο 5.....	95
Εφαρμογή Doodlz.....	95
5.1 Διαγράμματα λειτουργίας Doodlz	95
5.2 Οθόνη ζωγραφικής.....	97
5.3 Προσθήκη συσκευής.....	100
5.4 Επεξεργασία κωδικού ασφαλείας	101
5.5 Στοιχεία κώδικα εφαρμογής.....	103
5.6 Ασφάλεια	104
Κεφάλαιο 6.....	107
Επίλογος	107
6.1 Συμπεράσματα	107
6.2 Δυνατότητες επέκτασης	108
Βιβλιογραφία.....	111

Κατάλογος εικόνων

Εικόνα 1.1	Λειτουργικά συστήματα έξυπνων κινητών.....	15
Εικόνα 1.2	Επερχόμενα λειτουργικά συστήματα.....	15
Εικόνα 1.3	Λογότυπο Android.....	16
Εικόνα 1.4	Χρήση OS στην Αμερική.....	17
Εικόνα 1.5	Παράδειγμα Geo-fencing.....	19
Εικόνα 2.1	Το λογότυπο της MySQL	24
Εικόνα 2.2	Το λογότυπο του phpMyAdmin	25
Εικόνα 2.3	Το λογότυπο του Eclipse	26
Εικόνα 2.4	Το λογότυπο της γλώσσας Java.....	27
Εικόνα 2.5	Το λογότυπο της υπηρεσίας Okeanos.....	28
Εικόνα 2.6	Το λογότυπο του GlassFish	29
Εικόνα 2.7	Το λογότυπο του προγράμματος Netbeans.....	30
Εικόνα 2.8	Τρόπος λειτουργίας ενός RESTful WS	31
Εικόνα 3.1	Αρχιτεκτονική συστήματος	34
Εικόνα 3.2	Login.....	44
Εικόνα 3.3	Επιλογή Προσθήκης Συσκευής.....	44
Εικόνα 3.4	Εύρεση θέσης του παιδιού στο χάρτη.....	45
Εικόνα 3.5	Καρτέλα ορισμού security hint.....	46
Εικόνα 3.6	Email ανάκτησης λογαριασμού	48
Εικόνα 4.1	Εξομοιωτής Genymotion	53
Εικόνα 4.2	Sony Xperia P	54
Εικόνα 4.3	Δομή Android εφαρμογής.....	55
Εικόνα 4.4	Κύκλος ζωής ενός activity	57
Εικόνα 4.5	Διάρκεια ζωής ενός service	58
Εικόνα 4.6	Είσοδος χρήστη.....	63
Εικόνα 4.7	Επικοινωνία Android-Mysql Database	64
Εικόνα 4.8	Εγγραφή χρήστη	67
Εικόνα 4.9	Ανάκτηση λογαριασμού	70
Εικόνα 4.10	Αποτέλεσμα ανάκτησης λογαριασμού	71

Εικόνα 4.11 Αρχική σελίδα MapLocation	74
Εικόνα 4.12 Πληροφορίες θέσης συσκευής.....	75
Εικόνα 4.13 Προσθήκη συσκευής.....	79
Εικόνα 4.14 Επεξεργασία φράσης ασφαλείας	81
Εικόνα 4.15 Αποστολή σχολίων	84
Εικόνα 4.16 Αποκρυπτογράφηση δεδομένων με AES	91
Εικόνα 5.1 Κεντρική οθόνη Doodlz.....	97
Εικόνα 5.2 Οθόνη συγχρονισμού Doodlz	100
Εικόνα 5.3 Ταυτοποίηση χρήστη	102
Εικόνα 5.4 Κρυπτογράφηση δεδομένων με AES.....	105

Κατάλογος σχημάτων

Σχήμα 3.1 PCA_users	38
Σχήμα 3.2 PCA_devices.....	39
Σχήμα 3.3 PCA_linkCodes	41
Σχήμα 3.4 Διάγραμμα οντοτήτων-συσχετίσεων	42
Σχήμα 3.5 Προσχέδιο βάσης δεδομένων	43
Σχήμα 4.1 Σύνδεση χρήστη.....	59
Σχήμα 4.2 Εγγραφή χρήστη	60
Σχήμα 4.3 Ανάκτηση λογαριασμού χρήστη.....	60
Σχήμα 4.4 Κεντρική οθόνη εφαρμογής.....	61
Σχήμα 4.5 Προσθήκη συσκευής.....	62
Σχήμα 4.6 Απενεργοποίηση λογαριασμού	62
Σχήμα 5.1 Κεντρική οθόνη εφαρμογής.....	96
Σχήμα 5.2 Προσθήκη συσκευής.....	96

Ευχαριστίες

Στα πλαίσια της παρούσας διπλωματικής εργασίας ήρθα σε επαφή με τεχνολογίες νέες προς εμένα, αλλά κυρίως με ένα σύνολο από ανθρώπους τους οποίους θα ήθελα να ευχαριστήσω για την τεχνική και την ηθική υποστήριξη που μου παρείχαν σε όλη τη διάρκεια αυτής της προσπάθειας.

Αρχικά, θα ήθελα να ευχαριστήσω την κα. Μαρία Νικολαΐδη, καθηγήτρια του τμήματος και επιβλέπουσα καθηγήτρια της πτυχιακής εργασίας μου, καθώς και τους Δρ. Βασίλειο Δαλάκα, μέλος του Κέντρου Πληροφορικής & Δικτύων, και Δρ. Παναγιώτη Ριζομυλιώτη, επίκουρο καθηγητή του τμήματος Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων στο Πανεπιστήμιο Αιγαίου, για την εμπιστοσύνη που μου έδειξαν δίνοντάς μου τη δυνατότητα να ασχοληθώ με θέμα της επιλογής μου, καθώς και για την πολύτιμη βοήθεια καθ' όλη τη διάρκεια της δημιουργίας και της συγγραφής.

Στην προσπάθεια αυτή συνέβαλαν επίσης οι κύριοι Κωνσταντίνος Τσερπές, Λέκτορας του τμήματος Πληροφορικής και Τηλεματικής και Ανάργυρος Τσαδίμας, υποψήφιος Διδάκτωρ του τμήματος, τους οποίους θα ήθελα να ευχαριστήσω θερμά για την υποστήριξη που μου παρείχαν με τις χρήσιμες υποδείξεις και τα σχόλιά τους.

Τέλος, η εργασία αυτή δε θα μπορούσε να ολοκληρωθεί δίχως την αμέριστη συμπαράσταση και στήριξη της οικογένειας και των φίλων μου, που ήταν πρόθυμοι να ελέγξουν τη λειτουργικότητα της εφαρμογής μου.

Μανδαλενάκη Μαρία

Πρόλογος

Η παρούσα διπλωματική εργασία υλοποιήθηκε στα πλαίσια του Προγράμματος Προπτυχιακών Σπουδών του τμήματος Πληροφορικής και Τηλεματικής στο Χαροκόπειο Πανεπιστήμιο Αθηνών από τον Οκτώβριο του 2013 έως και τον Ιούλιο του 2014.

Σκοπός της συγκεκριμένης διπλωματικής είναι η ανάπτυξη εφαρμογών γονικού ελέγχου σε προγραμματιστικό περιβάλλον που βασίζονται στον προσδιορισμό της θέσης του κινητού με βάση τις συντεταγμένες που λαμβάνονται από το GPS. Έτσι, δημιουργήθηκαν δύο εφαρμογές για έξυπνα τηλέφωνα από τις οποίες κεντρική είναι αυτή που χρησιμοποιεί ο γονέας και η άλλη εγκαθίσταται στην εξαρτώμενη συσκευή του παιδιού.

Παρακάτω γίνεται μία αναφορά στη διάρθρωση του κειμένου που θα ακολουθηθεί στη διπλωματική.

- Στο **Κεφάλαιο 1** γίνεται μια αναφορά στην τεχνολογία των έξυπνων κινητών, σε ήδη υπάρχουσες εφαρμογές με παρόμοιο θέμα και στο λόγο επιλογής της παρούσας διπλωματικής.
- Το **Κεφάλαιο 2** αναλύει τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής της διπλωματικής.
- Στο **Κεφάλαιο 3** αναφέρεται ο τρόπος σχεδιασμού της βάσης δεδομένων και αναλύεται το σύστημα για την καλύτερη κατανόηση του τελικού αποτελέσματος.
- Στο **Κεφάλαιο 4** παρουσιάζεται η κεντρική εφαρμογή που εκπονήθηκε στα πλαίσια της παρούσας διπλωματικής, η οποία ονομάζεται MapLocation και πραγματοποιήθηκε σε έξυπνη συσκευή με λειτουργικό Android και παρέχει τη δυνατότητα του γονικού ελέγχου στο γονέα. Επιπλέον, παρουσιάζεται ο τρόπος υλοποίησής της και οι δυνατότητες που παρέχει στο χρήστη. Επίσης, περιέχονται και κάποια σημαντικά αποσπάσματα κώδικα. Τέλος, αναλύεται η ασφάλεια που χρησιμοποιήθηκε.
- Στο **Κεφάλαιο 5** παρουσιάζεται η δεύτερη εφαρμογή με την ονομασία Doodlz που εκπονήθηκε για την έξυπνη συσκευή του παιδιού, ο τρόπος υλοποίησής της και οι δυνατότητες που παρέχει στο χρήστη. Επίσης, περιέχονται τα αποσπάσματα κώδικα που επιβεβαιώνουν τη λειτουργία του γονικού ελέγχου.
- Το **Κεφάλαιο 6** αποτελεί σύνοψη της διπλωματικής και παρουσιάζονται τα συμπεράσματα και οι μελλοντικές εξελίξεις.

Περίληψη

Ο σύγχρονος αστικός τρόπος ζωής έχει ενισχύσει τις φοβίες των γονέων σχετικά με την ανατροφή των παιδιών τους δημιουργώντας νέες ανάγκες. Μία από τις πιο σημαντικές είναι ο γονικός έλεγχος. Η σύγχρονη τεχνολογία μπορεί να προσφέρει λύσεις μέσω του Συστήματος Εντοπισμού Θέσης (GPS) σε συνδυασμό με ένα σύγχρονο τηλέφωνο (smartphone).

Αντικείμενο της παρούσας διπλωματικής είναι να προσφέρει τη δυνατότητα στους γονείς να γνωρίζουν ανά πάσα στιγμή που βρίσκονται τα παιδιά τους χρησιμοποιώντας κάποιο σύγχρονο τηλέφωνο με λειτουργικό Android.

Ειδικότερα, ο κύριος στόχος αυτής της εργασίας ήταν η υλοποίηση δύο εφαρμογών, που σε αντίθεση με τις ήδη υπάρχουσες να παρέχουν διακριτικότητα και εχεμύθεια στο γονέα. Η βασική εφαρμογή εγκαθίσταται στην έξυπνη συσκευή του γονέα και του εμφανίζει ένα χάρτη με τη γεωγραφική τοποθεσία των εξαρτώμενων συσκευών που έχουν στην κατοχή τους τα παιδιά του, ενώ του παρέχεται και η δυνατότητα διαχείρισής τους. Η δεύτερη εφαρμογή χρησιμοποιείται από τα ανήλικα άτομα της οικογένειας και αποτελεί μία εφαρμογή ζωγραφικής, η οποία έχει υλοποιηθεί ώστε να τρέχει στο υπόβαθρο μία υπηρεσία που θα χρησιμοποιεί το GPS για να στέλνει σε ένα διακομιστή τη θέση της συσκευής.

Σημαντικό γνώρισμα για μία τέτοια εφαρμογή που χρησιμοποιεί την τοποθεσία του χρήστη της, διαδραματίζει η ασφάλεια των δεδομένων του. Έτσι, έχει γίνει προσπάθεια στις δύο εφαρμογές που εκπονήθηκαν να διασφαλιστεί η μεταδιδόμενη πληροφορία με σύγχρονες τεχνικές κωδικοποίησης, καθώς είναι αναγκαίο να αποτρέπεται η πρόσβαση σε οποιονδήποτε.

Καθώς στις μέρες μας το λογισμικό ανοιχτού κώδικα (ΕΛ/ΛΑΚ) είναι ευρέως διαδεδομένο και υπάρχει μεγάλη ποικιλία προγραμμάτων, επιλέχθηκε η χρήση τέτοιων τεχνολογιών για την υλοποίηση των εφαρμογών της διπλωματικής. Τα χαρακτηριστικά και η αποδοτικότητά τους μπορούν να ανταγωνιστούν τα εμπορικά λογισμικά που υπάρχουν. Ειδικότερα, όσον αφορά τα λογισμικά που ειδικεύονται στην υλοποίηση Android εφαρμογών, υπάρχουν πολλές διαθέσιμες επιλογές σε προγράμματα ανοιχτού κώδικα. Έτσι το σύνολο του κώδικα που αναπτύχθηκε βασίστηκε σε ΕΛ/ΛΑΚ.

Λέξεις κλειδιά: Smartphone, γονικός έλεγχος, παρακολούθηση παιδιού, γονείς, παιδιά, GPS, Android.

Abstract

Modern urban lifestyle has increased the fears of parents regarding the upbringing of their children by creating new needs. One of the most significant is parental control. Modern technology can provide solutions through the Global Positioning System (GPS) in combination with a smartphone.

The subject of this thesis is to provide the opportunity for parents to be aware at all times of the position of their children by using a smartphone running Android.

In particular, the main objective of this work was the implementation of two applications which, unlike the existing, can provide discretion and confidentiality to parents. The master application (installed on parent's device) displays a map of the geographical location of the devices held by their children. The parent has also the ability to manage every child's device attached to his/her account. The second application is installed on the children's device and it consists of a painting canvas. A service is running in the background that uses GPS in order to report the location of the device to a server.

A key issue for applications that use the location of their users is data security. Thus, efforts have been made in both applications to ensure that the information transmitted would be encoded, which is a necessity in order to prevent unauthorized access.

As open source software is widespread and there is a great variety of open source programs available we selected open source technologies in order to implement both applications presented here. The characteristics and the efficiency of open source programs can compete with success those of the corresponding commercial software. In particular, regarding software related with the implementation of Android applications, there are many possibilities coming from the open source world. Hence, the entire code was developed with the aid of open source applications.

Keywords: Smartphone, parental control, child tracking, parents, children, GPS, Android.

Κεφάλαιο 1

Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία εισαγωγή στα έξυπνα τηλέφωνα και σε εφαρμογές σχετικές με το γονικό έλεγχο και θα αναλυθεί ο λόγος που επιλέχθηκε το συγκεκριμένο θέμα.

1.1 Σκοπός της διπλωματικής

Ο σκοπός της συγκεκριμένης διπλωματικής είναι η ανάπτυξη εφαρμογών γονικού ελέγχου, σε προγραμματιστικό περιβάλλον, που βασίζεται στις πτυχές του geolocation και η οποία περιλαμβάνει την υπηρεσία των χαρτών της Google. Για να επιτευχθεί αυτό δημιουργήθηκαν δύο εφαρμογές για έξυπνα τηλέφωνα από τις οποίες η κεντρική είναι αυτή που χρησιμοποιεί ο γονέας κι η άλλη εγκαθίσταται στην εξαρτώμενη συσκευή του παιδιού του.

Η εφαρμογή έχει ως στόχο να προσφέρει τη δυνατότητα στους γονείς να εντοπίζουν τα νεαρά άτομα της οικογένειάς τους. Σε αντίθεση με τις περισσότερες από τις εφαρμογές που κυκλοφορούν σήμερα, τα παιδιά δεν έχουν καμία εμπλοκή στη λειτουργικότητα της εφαρμογής, ενώ ταυτόχρονα δεν γνωρίζουν ότι εντοπίζονται. Συγκεκριμένα, το παιδί ανοίγει την εφαρμογή της ζωγραφικής, η οποία όμως χρησιμοποιώντας την τεχνολογία του GPS, ενεργοποιεί στο υπόβαθρο μία υπηρεσία που στέλνει κρυπτογραφημένες περιοδικές ενημερώσεις της τρέχουσας τοποθεσίας του παιδιού στη βάση. Ο γονέας στη συνέχεια έχει τη δυνατότητα να εντοπίζει στη δική του εφαρμογή τις εξαρτώμενες από τη δική του συσκευές, να βλέπει τη χρονική στιγμή που έγινε η τελευταία μέτρηση και την απόσταση που απέχει από το αντίστοιχο παιδί του.

Οι δύο εφαρμογές έχουν δημιουργηθεί με κύριο γνώμονα τον ίδιο το χρήστη και την ασφάλεια των προσωπικών δεδομένων του ίδιου αλλά και των παιδιών του. Έτσι, τα δεδομένα που συναλλάσσονται μεταξύ των συσκευών και της βάσης είναι κρυπτογραφημένα και μόνο η συσκευή που έχει οριστεί από το γονέα μπορεί να τα αποκρυπτογραφήσει με κλειδί που ορίζει ο χρήστης και αποθηκεύεται κρυπτογραφημένο τοπικά. Επίσης, προς αποφυγή κλοπής των δεδομένων των παιδιών από οποιονδήποτε κατά την είσοδο ο χρήστης πρέπει να δώσει τα αναγνωριστικά εισόδου και του παρέχεται πρόσβαση αν είναι σωστά. Τέλος, προασπίζεται η ανωνυμία των χρηστών, καθώς το μόνο που απαιτείται για την ομαλή λειτουργία της εφαρμογής

είναι ένα όνομα χρήστη ως αναγνωριστικό κατά τη σύνδεση και ένα email για την ανάκτηση του λογαριασμού σε περίπτωση αλλαγής της συσκευής ή κλοπής.

1.2 Έξυπνα τηλέφωνα

Το έξυπνο τηλέφωνο είναι συσκευή βασισμένη σε ένα λειτουργικό σύστημα και παρουσιάζει περισσότερη προηγμένη υπολογιστική ικανότητα και συνδεσιμότητα σε σχέση με ένα συμβατικό κινητό τηλέφωνο. Τα πρώτα έξυπνα τηλέφωνα συνδύαζαν τις λειτουργίες ενός προσωπικού ψηφιακού βοηθού (PDA) και ενός κινητού τηλεφώνου, ενώ στα μεταγενέστερα μοντέλα προστέθηκαν και άλλες λειτουργίες, όπως ψηφιακές κάμερες και συστήματα πλοήγησης, με αποτέλεσμα σήμερα να διαμορφωθεί μια πολυχρηστική συσκευή με δυνατότητες αντίστοιχες ενός υπολογιστή. Ο όρος “έξυπνο τηλέφωνο” πρωτοεμφανίστηκε το 1997 όταν η Ericsson περιέγραψε το GS 88 “Penelope” ως “Smart Phone” (“Έξυπνο Τηλέφωνο”).

1.3 Λειτουργικά Συστήματα

Όπως αναφέρθηκε και παραπάνω, ένα έξυπνο τηλέφωνο χρησιμοποιεί κάποιο λειτουργικό σύστημα για να λειτουργήσει.

Το Λειτουργικό Σύστημα (Operating System ή OS) ονομάζεται στην επιστήμη της πληροφορικής το λογισμικό του υπολογιστή που είναι υπεύθυνο για τη διαχείριση και το συντονισμό των εργασιών, καθώς και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει ένα υπόβαθρο, ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή. Μία από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση του υλικού, απαλλάσσοντας έτσι το λογισμικό του χρήστη από το άμεσο και επίπονο χειρισμό του υπολογιστή και καθιστώντας εύκολο τον προγραμματισμό του. Οι πρώτοι υπολογιστές στερούνταν λειτουργικού συστήματος, ενώ τώρα σχεδόν όλοι οι υπολογιστές χρησιμοποιούν ένα τύπο αυτού.

Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων μπορούν να εγκατασταθούν σε πολλά διαφορετικά μοντέλα κινητών τηλεφώνων και συνήθως κάθε συσκευή μπορεί να λάβει πολλές ενημερωμένες εκδόσεις λογισμικού λειτουργικού συστήματος κατά τη διάρκεια ζωής της.

Κάποια από τα σύγχρονα λειτουργικά συστήματα που χρησιμοποιούνται σε smartphones είναι:

- Android της Google
- iOS της Apple
- Windows Phone της Microsoft
- Blackberry OS της RIM
- webOS της Hewlett-Packard
- Bada της Samsung

Επίσης, υπάρχουν και οι ενσωματωμένες διανομές Linux όπως το MeeGo.



Εικόνα 1.1 Λειτουργικά συστήματα έξυπνων κινητών

Μερικά άλλα αναπτυσσόμενα και επερχόμενα λειτουργικά συστήματα είναι:

- Firefox OS της Mozilla
- Ubuntu Phone της Canonical Ltd.'s
- Tizen



Εικόνα 1.2 Επερχόμενα λειτουργικά συστήματα

1.3.1 Android

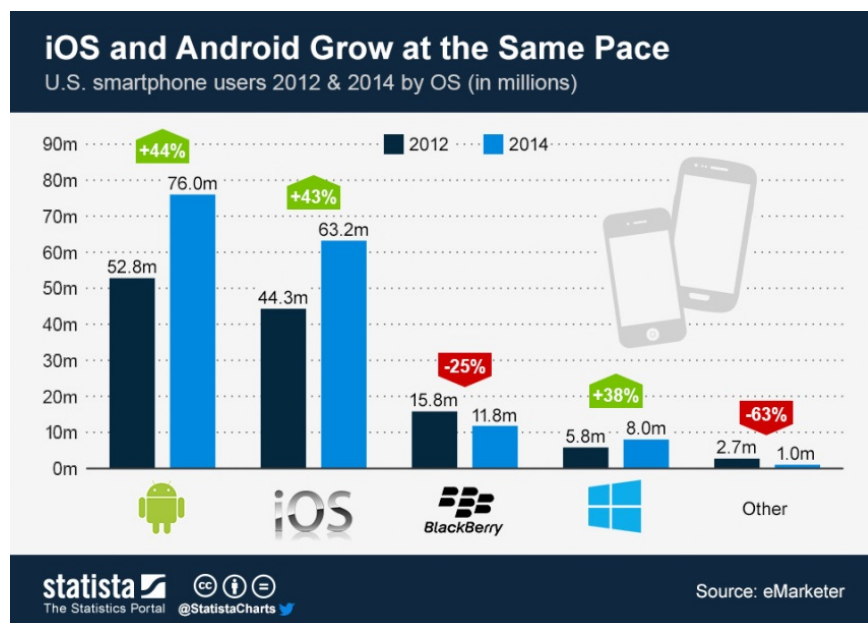
Ένα από τα πιο διαδεδομένα λειτουργικά συστήματα σήμερα είναι το Android. Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από τη Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με τη χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας τη συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από τη Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας σύμπραξης εταιρειών-κολοσσών στον τομέα της τεχνολογίας. Το λογότυπο για το λειτουργικό Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου. Στις 30 Απριλίου 2009 κυκλοφόρησε η επίσημη ενημέρωση έκδοσης 1.5 για το Android, ενώ η τελευταία μέχρι στιγμής έκδοση είναι η 4.4 με κωδική ονομασία KitKat και ανακοινώθηκε στις 3 Οκτωβρίου 2013.



Εικόνα 1.3 Λογότυπο Android

Τα τελευταία χρόνια, το Android μαζί με το iOS της Apple μάχονται για την κατάκτηση της πρώτης θέσης στην αγορά με τα Windows Phone της Microsoft να ακολουθούν.



Εικόνα 1.4 Χρήση OS στην Αμερική

1.4 GPS (Global Positioning System)

Οι δύο εφαρμογές που υλοποιήθηκαν στα πλαίσια της παρούσας διπλωματικής βασίστηκαν στο GPS. Το GPS (Παγκόσμιο Σύστημα Θεσιθεσίας) είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο βασίζεται σε ένα “πλέγμα” εικοσιτεσσάρων (24) δορυφόρων της Γης, στους οποίους υπάρχουν ειδικές συσκευές που ονομάζονται “δέκτες GPS”. Οι δέκτες αυτοί παρέχουν ακριβείς πληροφορίες για τη θέση ενός σημείου, το υψόμετρό του, την ταχύτητα και την κατεύθυνση της κίνησής του. Επίσης, σε συνδυασμό με ειδικό λογισμικό χαρτογράφησης (Google Maps) μπορούν να απεικονίσουν γραφικά τις πληροφορίες αυτές.

1.4.1 Geolocation

Οι εφαρμογές που δημιουργήθηκαν στηρίζονται στο geolocation για να εμφανίζουν στο χρήστη τις συντεταγμένες που λαμβάνουν από το GPS. Ο όρος geolocation είναι ο προσδιορισμός της γεωγραφικής θέσης ενός αντικειμένου, όπως ένα ραντάρ ή ένα κινητό στον πραγματικό κόσμο. Το geolocation μπορεί πρακτικά να εκτιμήσει την τοποθεσία ή αντιπροσωπεύει την ήδη εκτιμημένη θέση. Το geolocation είναι στενά συνδεδεμένο με τα συστήματα εντοπισμού θέσης,

αλλά μπορεί να διακριθεί από αυτά καθώς δίνεται μεγαλύτερη έμφαση στον καθορισμό μιας γεωγραφικής θέσης (πχ διεύθυνση) και όχι απλά σε μια σειρά από γεωγραφικές συντεταγμένες.

1.4.2 Location-based services (LBS)

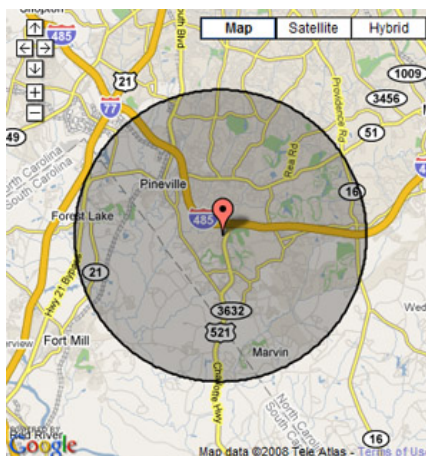
Όλες οι υπηρεσίες που χρησιμοποιήθηκαν στις δύο εφαρμογές της παρούσας διπλωματικής ορίζονται ως υπηρεσίες με βάση τη θέση. Οι υπηρεσίες με βάση τη θέση (LBS) είναι μια γενική κατηγορία των προγραμμάτων πληροφορικής που χρησιμοποιούν δεδομένα τοποθεσίας για τον έλεγχο των χαρακτηριστικών. Σήμερα, χρησιμοποιείται ιδιαίτερα στις κινητές συσκευές μέσω του δικτύου κινητής τηλεφωνίας και λαμβάνει ως παράμετρο τις πληροφορίες σχετικά με τη γεωγραφική θέση της συσκευής. Τα LBS περιλαμβάνουν υπηρεσίες που προσδιορίζουν τη θέση ενός αντικειμένου ή ενός ατόμου, όπως η εύρεση του κοντινότερου μηχανήματος ανάληψης ή τα ίχνη ενός φίλου ή εργαζόμενου. Επίσης, περιλαμβάνουν κινητό εμπόριο, όταν παίρνουν τη μορφή κουπονιών ή διαφημιστικών απευθυνόμενων σε πελάτες με βάση την τρέχουσα θέση τους. Τέλος, εμπεριέχουν εξατομικευμένες μετεωρολογικές υπηρεσίες, ακόμα και location-based παιχνίδια.

1.4.3 Geo-fencing

Στην περίπτωση που ο γονέας θέλει μέσα από την εφαρμογή του να θέσει μία περίμετρο στις περιοχές που μπορεί να κινηθεί ένα ανήλικό άτομο της οικογένειάς του, χρησιμοποιεί τις υπηρεσίες geo-fencing. Ο όρος geo-fencing είναι μία εικονική περίμετρος για μία γεωγραφική περιοχή του πραγματικού κόσμου. Ουσιαστικά, ένας γεω-φράκτης μπορεί να παραχθεί δυναμικά με βάση μία ακτίνα γύρω από ένα σημείο αναφοράς ή στατικά με κάποιο προκαθορισμένο σύνολο ορίων. Όταν μία συσκευή με επίγνωση θέσης με τη βοήθεια των LBS εισέρχεται ή εξέρχεται από ένα γεω-φράκτη, τότε λαμβάνεται μία ειδοποίηση που περιλαμβάνει πληροφορίες σχετικά με τη θέση της συσκευής.

Το geo-fencing είναι ένα από τα βασικά στοιχεία για το υλικό και το λογισμικό της τηλεματικής, καθώς επιτρέπει στους χρήστες του συστήματος να ορίσουν ζώνες γύρω από τόπους εργασίας ή ασφαλείς χώρους.

Το geo-fencing είναι επίσης ιδιαίτερα διαδεδομένο σε υπηρεσίες εντοπισμού των παιδιών, καθώς μπορούν να ειδοποιήσουν τους γονείς αν το παιδί αφήνει μία προκαθορισμένη περιοχή.



Εικόνα 1.5 Παράδειγμα Geo-fencing

1.4.4 Google Maps

Οι χάρτες που παρουσιάζονται στη μία από τις εφαρμογές που εκπονήθηκαν λαμβάνονται από τη Google μέσα από τη χρήση κάποιας Διεπαφής Προγραμματισμού Εφαρμογών (API) πακέτου. Οι χάρτες της Google είναι μία τεχνολογία χαρτογράφησης που παρέχεται από τη Google και είναι προσβάσιμη από φυλλομετρητή αλλά και από εφαρμογή κινητού τηλεφώνου. Η υπηρεσία παρέχει δορυφορικές εικόνες, χάρτες οδών, καθώς και λειτουργίες όπως το σχεδιασμό διαδρομής για μετακινήσεις με τα πόδια, το αυτοκίνητο, το ποδήλατο ή τα μέσα μεταφοράς. Υποστηρίζουν επίσης την ενσωμάτωσή τους σε ιστοσελίδες τρίτων μέσω του Google Maps API, καθώς και τον εντοπισμό των αστικών επιχειρήσεων και άλλων οργανώσεων σε πολλές χώρες του κόσμου. Οι δορυφορικές εικόνες των χαρτών της Google δεν ενημερώνονται σε πραγματικό χρόνο, ωστόσο η Google προσθέτει στοιχεία στην Κύρια Βάση Δεδομένων της σε τακτά διαστήματα καταφέροντας έτσι οι περισσότερες από τις εικόνες της να μην είναι περισσότερο από τριών ετών. Τέλος, οι χάρτες παρέχονται δωρεάν για την ανάπτυξη εφαρμογών που θα διατίθενται χωρίς κάποιο κόστος στο κοινό, ενώ είναι έναντι κάποιου ποσού για τις εταιρίες ή τους προγραμματιστές που θέλουν να δημιουργήσουν εφαρμογές με εμπορικό σκοπό.

1.5 Εφαρμογές

Με την ανάπτυξη των Λειτουργικών Συστημάτων για έξυπνα τηλέφωνα ξεκίνησε και η δημιουργία εφαρμογών που υποστηρίζονται από το αντίστοιχο λειτουργικό και παρέχουν

διάφορες δυνατότητες ή διασκέδαση στο χρήστη. Επίσης, όλες οι υπηρεσίες που αναφέρθηκαν παραπάνω για να χρησιμοποιηθούν από τα έξυπνα τηλέφωνα, πρέπει να συμπεριληφθούν σε κάποια εφαρμογή.

Μία εφαρμογή για κινητά τηλέφωνα είναι μία εφαρμογή λογισμικού σχεδιασμένη για να τρέχει σε έξυπνα τηλέφωνα, ταμπλέτες και άλλες φορητές συσκευές. Οι εφαρμογές είναι συνήθως διαθέσιμες μέσω κάποιων πλατφόρμων διανομής εφαρμογών, οι οποίες άρχισαν να εμφανίζονται από το 2008 και λειτουργούν συνήθως από τον πάροχο του λειτουργικού συστήματος. Οι βασικότερες πλατφόρμες μέχρι στιγμής είναι το Apple App Store, το Google Play, το Windows Phone World και το BlackBerry App World. Ορισμένες εφαρμογές διατίθενται δωρεάν, ενώ άλλες πρέπει να αγοραστούν. Σχεδιάζονται έχοντας ως στόχο ένα συγκεκριμένο λειτουργικό σύστημα, αλλά κάποιες μπορούν να χρησιμοποιηθούν και σε φορητούς ή επιτραπέζιους υπολογιστές.

Αρχικά, οι εφαρμογές για κινητά τηλέφωνα προσφέρθηκαν για τη γενική παραγωγικότητα και την ανάκτηση των πληροφοριών, συμπεριλαμβανομένων του ηλεκτρονικού ταχυδρομείου (e-mail), του ημερολογίου, των επαφών, διαφόρων πληροφοριών για τη χρηματιστηριακή αγορά και του καιρού. Στην πορεία όμως, η μεγάλη ζήτηση και η διαθεσιμότητα των εργαλείων ανάπτυξης οδήγησαν στην ταχεία επέκταση των εφαρμογών και σε άλλες κατηγορίες, όπως τα παιχνίδια, το GPS με τις υπηρεσίες εντοπισμού θέσης, οι τραπεζικές εφαρμογές και οι αγορές εισιτηρίων.

Σήμερα, υπάρχουν διάφορες εφαρμογές για Android και iOS συσκευές σχετικά με το γονικό έλεγχο και την ακριβή εύρεση της θέσης των παιδιών, οι οποίες χρησιμοποιούνται και παρουσιάζουν ιδιαίτερο ενδιαφέρον. Στον παρακάτω πίνακα, παρουσιάζονται κάποιες από αυτές τις εφαρμογές και γίνεται σύγκριση με συγκεκριμένες λειτουργίες της εφαρμογής που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής.

		MapLocation	Footprints	Family Tracker	Sprint Family Locator	GPS Phone Tracker	Follow Mee	Life 360
Συμβατότητα	Android	✓		✓	✓	✓	✓	✓
	iOS		✓	✓		✓	✓	✓
Εμπλοκή παιδιών		✗	✓	✗	✗	✓	✗	✓
Γνώση παιδιού ότι εντοπίζεται		✗	✓	✓	✗	✓	✗	✓
Τοποθεσία πραγματικού χρόνου		✗	✗	✓	✓	✗	✗	✓
Κρυπτογράφηση δεδομένων θέσης		✓	✓	✗	✓	✓	✓	✓
Ανωνυμία γονέα		✓	✗	✓	✗	✓	✓	✗
Είσοδος με συνθηματικό στην εφαρμογή		✓	✗	✗	✓	✗	✓	✓
Κόστος		Δωρεάν	Μηνιαία συνδρομή για την υπηρεσία	Δωρεάν, συνδρομή για επιπλέον λειτουργίες	15 μέρες δωρεάν Μετά, \$5 μηνιαία	Δωρεάν δοκιμή, συνδρομή για όλες τις λειτουργίες	Δωρεάν δοκιμή, συνδρομή για όλες τις λειτουργίες	Δοκιμή ενός μήνα, μετά συνδρομή

Κεφάλαιο 2

Τεχνολογίες

Σε αυτό το κεφάλαιο θα γίνει μία αναφορά στις τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση των εφαρμογών της παρούσας διπλωματικής εργασίας.

2.1 MySQL

Η βάση δεδομένων που αποθηκεύονται όλες οι πληροφορίες που στέλνονται από τις δύο εφαρμογές που εκπονήθηκαν στην παρούσα διπλωματική, δημιουργήθηκε με MySQL.

Η MySQL είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων. Μια βάση δεδομένων επιτρέπει την αποτελεσματική αποθήκευση, αναζήτηση, ταξινόμηση και ανάκληση των δεδομένων. Ο MySQL διακομιστής ελέγχει την πρόσβαση στα δεδομένα για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς, η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language) ως την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Είναι διαθέσιμη δωρεάν με άδεια ανοιχτού κώδικα (Open Source), δηλαδή οποιοσδήποτε μπορεί να την κατεβάσει και να τη διαμορφώσει ανάλογα με τις ανάγκες του ή με εμπορική άδεια σε πολύ χαμηλό κόστος. Η MySQL μπορεί να χρησιμοποιηθεί σε πολλά διαφορετικά συστήματα Unix, όπως επίσης και σε Microsoft Windows.

Η MySQL έχει πολλά πλεονεκτήματα σε σχέση με τους ανταγωνιστές της:

- Είναι ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων.
- Έχει διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων.
- Παρέχει ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες.
- Είναι οικονομική.
- Έχει ευκολίες στη μάθηση και τη χρήση.
- Είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και σε διάφορα εργαλεία ανάπτυξης.

- Παρέχει ευκολίες στο backup.
- Είναι ανοιχτού κώδικα, επομένως ο κώδικας προέλευσης είναι διαθέσιμος.
- Αναπτύσσεται συνεχώς.

2.1.1 Ιστορική αναδρομή

Η MySQL δημιουργήθηκε από μία σουηδική εταιρία, τη MySQL AB. Ιδρυτές της είναι οι David Axmark, Allan Larsson και Michael Widenius. Η πρώτη έκδοση της MySQL εμφανίστηκε στις αρχές του 1995. Το 2008, η MySQL εξαγοράστηκε από τη Sun Microsystems, με αποτέλεσμα από τις αρχές του 2010 να βρίσκεται στην ιδιοκτησία της Oracle.



Εικόνα 2.1 Το λογότυπο της MySQL

2.2 PhpMyAdmin

Για το σχεδιασμό και τη διαχείριση της βάσης δεδομένων για την αποθήκευση των πληροφοριών των δύο εφαρμογών, χρησιμοποιήθηκε το πρόγραμμα phpMyAdmin.

Το phpMyAdmin είναι ένα ελεύθερο και ανοιχτού κώδικα εργαλείο γραμμένο σε PHP που χρησιμοποιείται για τη διαχείριση της MySQL με τη χρήση ενός web browser. Μπορούν να εκτελεστούν διάφορες εργασίες, όπως τη δημιουργία, τροποποίηση ή διαγραφή βάσεων δεδομένων, πινάκων, πεδίων ή γραμμών. Επίσης, το πρόγραμμα παρέχει τη δυνατότητα δημιουργία και εκτέλεσης SQL ερωτημάτων, καθώς και τη διαχείριση των χρηστών και των δικαιωμάτων τους.

Το πρόγραμμα παρέχει πολλές δυνατότητες σε σχέση με τους ανταγωνιστές του. Κάποιες από τις δυνατότητες του προγράμματος περιλαμβάνουν:

- Διασύνδεση Web
- Διαχείριση MySQL βάσεων δεδομένων
- Εισαγωγή δεδομένων από αρχεία csv και sql

- Εξαγωγή δεδομένων σε διάφορες μορφές
- Χορήγηση πολλαπλών διακομιστών
- Δημιουργία γραφικών σε μορφή pdf της διάταξης βάσης δεδομένων
- Διαγράμματα πραγματικού χρόνου για την παρακολούθηση της δραστηριότητας του διακομιστή MySQL

2.2.1 Ιστορική αναδρομή

Το 1998, ο Tobias Ratschiller, σύμβουλος πληροφορικής και στη συνέχεια ιδρυτής της εταιρίας λογισμικού Maguma, άρχισε να εργάζεται σε ένα php πρόγραμμα για την επεξεργασία της MySQL, εμπνευσμένος από το πρόγραμμα MySQL-Webadmin. Το 2000 εγκατέλειψε το έργο λόγω έλλειψης χρόνου. Μέχρι εκείνη τη στιγμή το phpMyAdmin γνώριζε μεγάλη άνθηση, με αποτέλεσμα το 2001 μία ομάδα τριών προγραμματιστών να το καταχωρήσουν στο SourceForge.net και να αναλάβουν την ανάπτυξή του.



Εικόνα 2.2 Το λογότυπο του phpMyAdmin

2.3 Eclipse

Το Eclipse αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης μέσα στο οποίο μπορεί να εκτελεστεί κώδικας. Είναι σχεδιασμένο κατά κύριο λόγο για την ανάπτυξη Java εφαρμογών, ενώ με τη βοήθεια των διαφόρων επεκτάσεων (plug-ins) το Eclipse μπορεί να χρησιμοποιηθεί και για τον προγραμματισμό σε άλλες γλώσσες.

Είναι το πρόγραμμα που χρησιμοποιήθηκε για το σχεδιασμό και την ανάπτυξη των android εφαρμογών που εκπονήθηκαν στα πλαίσια της παρούσας διπλωματικής. Το Eclipse χρησιμοποιείται ευρέως λόγω των δυνατοτήτων που προσφέρει:

- Έχει απλό περιβάλλον χρήσης.

- Υποστηρίζει πολλές γλώσσες προγραμματισμού μέσω των διαθέσιμων επεκτάσεων που διαθέτει.
- Υπάρχει υποστήριξη σε πολλές διαδικτυακές κοινότητες.
- Θεωρείται προαπαιτούμενο για προγραμματισμό και ανάπτυξη Android εφαρμογών.

2.3.1 Ιστορική αναδρομή

Ξεκίνησε ως ένα εργαλείο ανοιχτού κώδικα από την IBM Καναδά. Εκτιμάται ότι η IBM είχε επενδύσει σχεδόν 40 εκατομμύρια δολάρια προτού δημιουργηθεί μία κοινοπραξία από 80 μέλη μαζί με την IBM για την ίδρυση του ιδρύματος Eclipse τον Ιανουάριο του 2004. Από τότε το Eclipse έχει γίνει ένα πολύ διαδεδομένο εργαλείο προγραμματισμού.



Εικόνα 2.3 Το λογότυπο του Eclipse

2.4 Java

Η ανάπτυξη των δύο εφαρμογών, αλλά και του Restful Web Service που χρησιμοποιήθηκε για την επικοινωνία με το διακομιστή, έγινε με τη γλώσσα προγραμματισμού Java, καθώς είναι η μόνη γλώσσα που μπορεί να χρησιμοποιηθεί για προγραμματισμό Android εφαρμογών.

Η Java είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρία πληροφορικής Sun Microsystems. Ένα από τα βασικά χαρακτηριστικά της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι “κατανοητά” από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή αλλά και λειτουργικού συστήματος, αφού κάθε κεντρική μονάδα επεξεργασίας κατανοεί

διαφορετικό κώδικα μηχανής. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM). Τέλος, ακόμα μία ιδέα που βρίσκεται πίσω από τη Java είναι η ύπαρξη του Συλλέκτη Απορριμμάτων (Garbage Collector), ο οποίος είναι υπεύθυνος για την αυτόματη απελευθέρωση μνήμης κατά τη διάρκεια εκτέλεσης ενός προγράμματος.

2.4.1 Ιστορική αναδρομή

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικροσυσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Ο δημιουργός της γλώσσας ήταν ο James Gosling, ο οποίος εκείνη την εποχή εργαζόταν για τη Sun και της έδωσε το όνομα Oak από το ομώνυμο δέντρο (βελανιδιά) που είχε έξω από το γραφείο του. Όμως το όνομα της νέας γλώσσας ήταν ήδη κατοχυρωμένο, οπότε σύντομα οι υπεύθυνοι ανάπτυξης τη μετονόμασαν σε Java, που εκτός των άλλων ήταν το όνομα της αγαπημένης τους ποικιλίας καφέ.



Εικόνα 2.4 Το λογότυπο της γλώσσας Java

2.5 <https://okeanos.grnet.gr/>

Η υπηρεσία okeanos αποτελεί μέρος μίας στοίβας προγραμμάτων νέφους με τη ονομασία Synnefo που δημιουργήθηκαν από το Εθνικό Δίκτυο Έρευνας και Τεχνολογίας (ΕΔΕΤ). Παρέχεται σε όλα τα μέλη της εκπαιδευτικής και ερευνητικής κοινότητας, τα οποία μπορούν να αξιοποιούν τα κέντρα δεδομένων της ΕΔΕΤ με τη μορφή εικονικών μηχανών για τις δικές τους εφαρμογές. Τα δεδομένα των εικονικών μηχανών αποθηκεύονται σε μία υπηρεσία κεντρικής φιλοξενίας υπολογιστικών και αποθηκευτικών πόρων (Cloud Computing).

Η συγκεκριμένη υπηρεσία χρησιμοποιήθηκε για τη δημιουργία μίας εικονικής μηχανής, μέρος της οποίας θα ήταν και ο διακομιστής για τη λειτουργία των δύο android εφαρμογών και του

Rest Web Service. Η εικονική μηχανή που δημιουργήθηκε χρησιμοποιεί λειτουργικό σύστημα Ubuntu 13.10.

2.5.1 Ιστορική αναδρομή

Το Νοέμβριο του 2006, σε μία προσπάθεια για την παροχή προηγμένων υπηρεσιών νέφους για την ελληνική ακαδημαϊκή και ερευνητική κοινότητα, η ΕΔΕΤ αποφασίζει να ξεκινήσει μία υπηρεσία αποθήκευσης, παρόμοια με το S3 της Amazon, την οποία ονόμασε Pithos. Το έργο προσφέρεται στα μέλη της ακαδημαϊκής κοινότητας το Μάιο του 2009.

Τον Ιούνιο του 2010, η ΕΔΕΤ αποφασίζει το επόμενο βήμα στις υπηρεσίες της, να δημιουργήσει ένα ολοκληρωμένο σύστημα νέφους που να περιλαμβάνει την υλοποίηση μιας εικονικής μηχανής. Αυτή η υπηρεσία ονομάστηκε *okeanos* και παρουσιάστηκε σε δοκιμαστική έκδοση τον Ιούλιο του 2011. Στην αναβαθμισμένη έκδοση του *okeanos* που κυκλοφόρησε το Μάρτιο του 2012, έχει ενσωματωθεί πλήρως η υπηρεσία Pithos ως μέρος του συστήματος *okeanos*.



Εικόνα 2.5 Το λογότυπο της υπηρεσίας Okeanos

2.6 GlassFish

Για την εκτέλεση του Rest Web Service που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής, εγκαταστάθηκε στην εικονική μηχανή που χρησιμοποιήθηκε ο εξυπηρετής GlassFish, ο οποίος παρέχει πλήρη συμβατότητα με την τεχνολογία Restful.

Ο GlassFish είναι ένας εξυπηρετής εφαρμογών (application server) ανοιχτού κώδικα που ξεκίνησε από την εταιρία Sun Microsystems για την πλατφόρμα Java EE και πλέον χρηματοδοτείται από την Oracle Corporation. Η υποστηριζόμενη έκδοση ονομάζεται Oracle GlassFish Server και παρέχεται με διπλή άδεια ελεύθερης χρήσης, την Common Development and Distribution License (CDDL) και τη GNU General Public License (GPL).

2.6.1 Ιστορική αναδρομή

Τον Ιούνιο του 2005, η Sun Microsystems ξεκίνησε την υλοποίηση του GlassFish, με αποτέλεσμα το Μάιο του 2006 να παρουσιαστεί στην αγορά η πρώτη έκδοσή του, η οποία υποστήριζε την έκδοση Java EE 5. Το Μάρτιο του 2010 με την απόκτηση της Sun Microsystems από την Oracle ξεκίνησε η διάθεση των νέων εκδόσεων του GlassFish, οι οποίες είναι απόλυτα συμβατές με τις υπόλοιπες τεχνολογίες της εταιρίας. Στις 12 Ιουνίου του 2013 η Oracle Corporation ανακοίνωσε τη διάθεση του GlassFish 4.0 με την υποστήριξη της έβδομης εμπορικής έκδοσης της Java.



Εικόνα 2.6 Το λογότυπο του GlassFish

2.7 Netbeans

Το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε για τη δημιουργία του Web Service για την επικοινωνία των android εφαρμογών με το διακομιστή είναι το Netbeans IDE.

Το Netbeans είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την ανάπτυξη προγραμμάτων κυρίως με τη Java, αλλά και με άλλες γλώσσες όπως PHP, C/C++ και HTML5. Περιλαμβάνει επίσης και μία πλατφόρμα εκτέλεσης εφαρμογών γραμμένων σε Java ή σε άλλη γλώσσα. Το Netbeans IDE είναι γραμμένο σε Java και μπορεί να τρέξει σε Windows, OS X, Linux, Solaris και άλλων πλατφόρμων που μπορούν να υποστηρίξουν μία συμβατή Java εικονική μηχανή (JVM). Η πλατφόρμα Netbeans επιτρέπει την ανάπτυξη των εφαρμογών από ένα σύνολο βασικών στοιχείων λογισμικού που ονομάζονται ενότητες. Οι εφαρμογές που βασίζονται στην πλατφόρμα Netbeans (συμπεριλαμβανόμενου του ίδιου του Netbeans IDE) μπορούν να επεκταθούν από τρίτους προγραμματιστές.

2.7.1 Ιστορική αναδρομή

Το Netbeans ξεκίνησε το 1996 με την ονομασία Xelfi ως μία Java εργασία ενός φοιτητή υπό την επίβλεψη του τμήματος Μαθηματικών και Φυσικής στο Πανεπιστήμιο Charles στην Πράγα. Το 1997, ο Roman Staněk συγκροτεί μία εταιρία γύρω από το συγκεκριμένο έργο και αρχίζουν να

παράγονται εμπορικές εκδόσεις του Netbeans IDE μέχρι που αγοράστηκε από τη Sun Microsystems το 1999. Τον Ιούνιο του επόμενου έτους, η Sun διέθεσε το Netbeans ως πρόγραμμα ανοιχτού κώδικα. Από τότε, η κοινότητα Netbeans συνεχίζει να αυξάνεται. Το 2010, η Sun και συνεπώς το Netbeans IDE εξαγοράστηκαν από την Oracle.



Εικόνα 2.7 Το λογότυπο του προγράμματος Netbeans

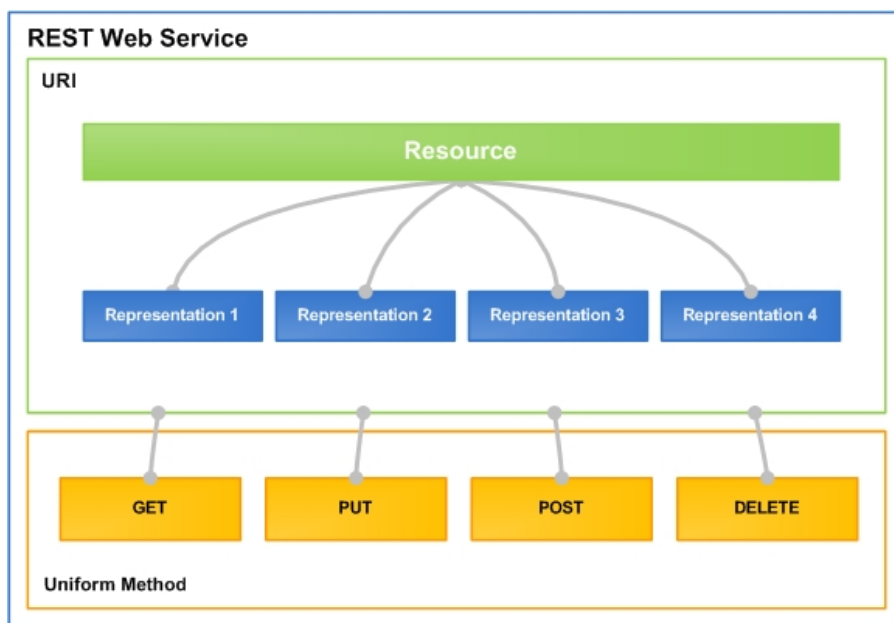
2.8 RESTful Web Service

Η συντομογραφία REST προκύπτει από την παραστατική κατάσταση μεταφοράς (Representational state transfer). Το REST είναι ένα λογισμικού αρχιτεκτονικού στυλ που αποτελείται από ένα συντονισμένο σύνολο από αρχιτεκτονικούς περιορισμούς που εφαρμόζονται σε κατασκευαστικά στοιχεία, συνδέσμους και στοιχεία δεδομένων, μέσα σε ένα καταναμημένο σύστημα υπερμέσων. Το REST αγνοεί τις λεπτομέρειες της υλοποίησης των στοιχείων και της σύνταξης του πρωτοκόλλου ώστε να επικεντρωθεί στους ρόλους των στοιχείων, τους περιορισμούς κατά την αλληλεπίδραση με άλλα στοιχεία και την εξήγησή τους από σημαντικά στοιχεία δεδομένων.

Ο όρος REST καθορίστηκε το 2000 από τον Roy Fielding κατά τη διάρκεια της διδακτορικής διατριβής του στο Πανεπιστήμιο της Καλιφόρνια. Εφαρμόστηκε για να περιγραφεί η επιθυμητή διαδικτυακή (web) αρχιτεκτονική, να εντοπιστούν υπάρχοντα προβλήματα, να συγκριθούν οι εναλλακτικές λύσεις και να εξασφαλιστεί ότι οι επεκτάσεις του πρωτοκόλλου δε θα παραβιάζουν τους βασικούς περιορισμούς που εξασφαλίζουν την επιτυχία του web. Ο Fielding χρησιμοποίησε το REST για να σχεδιάσει το πρωτόκολλο HTTP 1.1 και το αναγνωριστικό πόρων URI (Uniform Resource Identifier). Το αρχιτεκτονικό στυλ του REST εφαρμόστηκε επίσης για την ανάπτυξη διαδικτυακών υπηρεσιών ως εναλλακτική λύση σε άλλες προδιαγραφές καταναμημένων υπολογιστικών συστημάτων, όπως είναι το SOAP (Simple Object Access Protocol).

Τα Web Services που τηρούν τις προδιαγραφές του REST ονομάζονται RESTful και προσδιορίζονται από τα παρακάτω στοιχεία:

- Μία URI διεύθυνση ως βάση, όπως για παράδειγμα `http://example.com/resources/`
- Ένας τύπος πολυμέσων Internet για τα δεδομένα. Ο πιο συχνός τύπος και αυτός που χρησιμοποιήθηκε για την εκπόνηση της παρούσας διπλωματικής είναι JSON, αλλά υπάρχουν και άλλοι, όπως XML, Atom, εικόνες.
- Μία από τις βασικές μεθόδους HTTP (GET, PUT, POST ή DELETE)
- Σύνδεσμοι υπερκειμένου για την αναφορά της κατάστασης
- Σύνδεσμοι υπερκειμένου για την αναφορά των σχετικών πόρων



Εικόνα 2.8 Τρόπος λειτουργίας ενός RESTful WS

Κεφάλαιο 3

Ανάλυση και Σχεδίαση

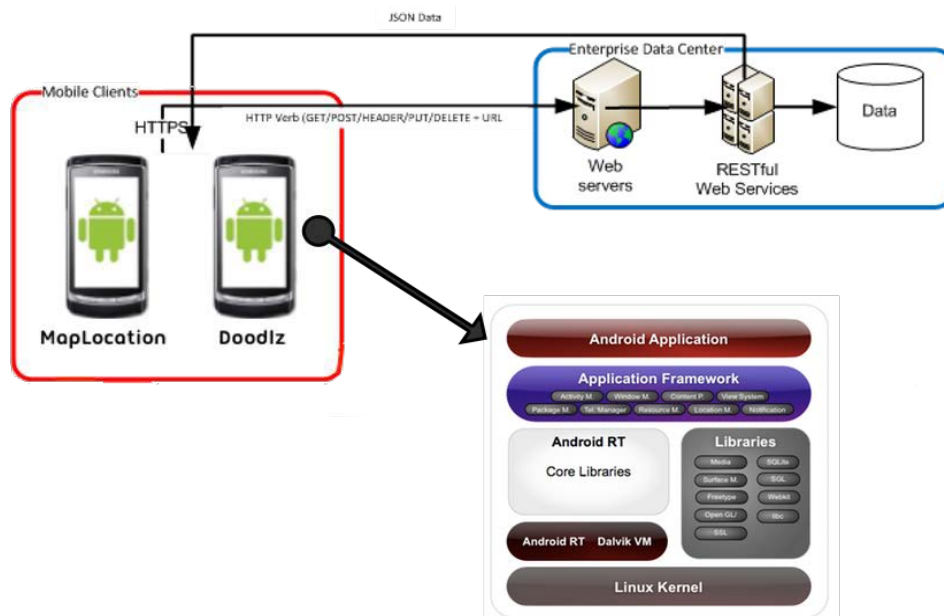
Σε αυτό το κεφάλαιο γίνεται η ανάλυση και ο αρχικός σχεδιασμός του συστήματος ώστε να αποσαφηνιστούν τα βήματα που έγιναν ώστε να φτάσουμε στο τελικό αποτέλεσμα. Συλλέγονται οι απαιτήσεις που προκύπτουν με βάση τις ανάγκες των χρηστών και τις διαθέσιμες λειτουργίες του συστήματος. Στη συνέχεια και λαμβάνοντας υπόψη τις πληροφορίες που συλλέχθηκαν κατά τη φάση της ανάλυσης, οδηγούμαστε στην ανάπτυξη του βασικού αρχιτεκτονικού σχεδιασμού για το σύστημα, ο οποίος περιγράφει το υλικό, λογισμικό και τη δικτυακή υποδομή που θα χρησιμοποιηθούν.

3.1 Αρχιτεκτονική

Οι εφαρμογές έχουν υλοποιηθεί για Smartphones ή tablets συμβατά με λογισμικό Android. Ο χρήστης της εφαρμογής Doodlz (παιδί) χρησιμοποιώντας την εφαρμογή στο κινητό του στέλνει πληροφορίες μέσω της υπηρεσίας GPRS ή του WiFi δικτύου στη βάση δεδομένων. Αντίστοιχα, ο χρήστης (γονέας) της άλλης εφαρμογής (MapLocation) μπορεί να συνδεθεί στη δική του έξυπνη συσκευή (Smartphone) και να δει τις πληροφορίες που στέλνονται από τις εξαρτώμενες από αυτόν συσκευές. Κάθε χρήστης κινητού τερματικού και των δύο εφαρμογών έχει διαφορετικό αναγνωριστικό (id) για να ξεχωρίζει, αφού οι εφαρμογές είναι διαμορφωμένες ώστε να υποστηρίζουν πολλούς χρήστες ταυτόχρονα. Συγχρόνως, η εφαρμογή Doodlz συνεχίζει να στέλνει ανά τακτά διαστήματα στοιχεία για τη θέση της συσκευής, τα οποία μπορεί να δει μόνο ο εγγεγραμμένος χρήστης της εφαρμογής MapLocation που έχει οριστεί ως γονέας του συγκεκριμένου παιδιού.

Ο χρήστης της Doodlz εφαρμογής στέλνει τα δεδομένα μέσω Διαδικτύου στο διακομιστή. Τα δεδομένα αυτά μπορεί να είναι είτε κρυπτογραφημένες πληροφορίες για τη θέση του είτε ο κωδικός συγχρονισμού με άλλη συσκευή που καταχώρησε ο χρήστης. Όλα τα δεδομένα προωθούνται σε έναν εξυπηρετή (server) που περιλαμβάνει τη βάση δεδομένων και καταχωρούνται σε αυτή. Ο χρήστης της MapLocation εφαρμογής με τη σειρά του μπορεί να στείλει δεδομένα στη βάση δεδομένων που σχετίζονται με την είσοδό του στο σύστημα ή με τη διαχείριση των εξαρτώμενων συσκευών των παιδιών. Επίσης, σε αυτή την εφαρμογή ο χρήστης

έχει τη δυνατότητα να λάβει τις πληροφορίες της θέσης που στέλνουν οι εφαρμογές των παιδιών του, να τις αποκρυπτογραφήσει και να τις δει στην τερματική συσκευή του μέσω της υπηρεσίας των χαρτών της Google.



Εικόνα 3.1 Αρχιτεκτονική συστήματος

3.2 Ρόλοι του συστήματος

Κατά τη διάρκεια της ανάλυσης και του σχεδιασμού, καταγράφονται όλοι οι χρήστες που θα αλληλεπιδράσουν με το σύστημα, αλλά και οι ρόλοι τους σε αυτό. Έτσι, καταλήγουμε στους τρεις χρήστες που ακολουθούν.

- Διαχειριστής / administrator
- Εγγεγραμμένος χρήστης της MapLocation εφαρμογής / γονέας
- Χρήστης της Doodlz εφαρμογής / παιδί

Κάθε μία από τις οντότητες του συστήματος αναλύεται ξεχωριστά παρακάτω.

Ο διαχειριστής / administrator είναι η οντότητα που έχει τις περισσότερες δυνατότητες στο σύστημα. Μπορεί να διαχειρίζεται τη βάση δεδομένων και να ελέγχει τους χρήστες. Βέβαια,

λόγω της κρυπτογράφησης που έχουν υποστεί τα βασικότερα δεδομένα που στέλνονται από τις δύο εφαρμογές, ο διαχειριστής δεν έχει την ικανότητα να γνωρίζει τους κωδικούς των χρηστών ή την τρέχουσα θέση μιας συσκευής. Τέλος, αρμοδιότητά του είναι να απαντάει στα ερωτήματα των χρηστών και να επιλύει τα προβλήματα που αντιμετωπίζουν. Γενικά, είναι αυτός που ελέγχει τη σωστή λειτουργία του συστήματος. Όμως, η αλληλεπίδρασή του πρώτου με αυτό είναι λιγότερο συχνή σε σχέση με τις άλλες οντότητες.

Ο εγγεγραμμένος χρήστης της MapLocation εφαρμογής έχει το βασικότερο ρόλο στο σύστημα. Εξ ορισμού είναι ένας ενήλικος χρήστης που χρησιμοποιεί την Android εφαρμογή MapLocation ώστε να λαμβάνει συγκεκριμένες ενημερώσεις για τη θέση των εξαρτώμενων από αυτόν νεαρών χρηστών. Μπορεί να προσθέτει νέες συσκευές παιδιών και να τις διαχειρίζεται. Επίσης, είναι ο μόνος χρήστης που έχει επικοινωνία με το διαχειριστή του συστήματος μέσω της αντίστοιχης φόρμας. Επομένως, είναι ο χρήστης που αλληλεπιδρά πιο συχνά με το σύστημα και αντλεί τις περισσότερες πληροφορίες από αυτό.

Οι χρήστες της Doodlz εφαρμογής είναι ουσιαστικά τα εξαρτώμενα από έναν γονέα παιδιά, που χρησιμοποιούν την εφαρμογή τους για να διασκεδάσουν μέσα από τη ζωγραφική. Είναι οι χρήστες με τη λιγότερη γνώση των ικανοτήτων της εφαρμογής, καθώς δε αντιλαμβάνονται ότι καθ' όλη τη διάρκεια εκτέλεσης της εφαρμογής, ο υπεύθυνος γονέας ενημερώνεται για τη θέση τους.

3.3 Απαιτήσεις

Γνωρίζοντας πλέον ποιοι μπορούν να χρησιμοποιήσουν το σύστημα και να αλληλεπιδράσουν με αυτό, μπορούμε να καθορίσουμε τις απαιτήσεις που έχει κάθε μία από τις οντότητες του συστήματος.

Ο διαχειριστής μπορεί:

- Να διαχειρίζεται τη βάση δεδομένων.
- Να ελέγχει τη σωστή λειτουργία των δύο εφαρμογών από τους χρήστες.
- Να απαντάει στα ερωτήματα των χρηστών.
- Να επιλύει τα προβλήματά τους.

Ο εγγεγραμμένος χρήστης της MapLocation εφαρμογής μπορεί:

- Να κάνει Προσθήκη/Διαχείριση εξαρτώμενων χρηστών (παιδιών).
- Να βρει την τρέχουσα θέση του στο χάρτη.
- Να βρει τη θέση της συσκευής ενός παιδιού.
- Να ελέγξει στοιχεία σχετικά με την τελευταία θέση της συσκευής που τον ενδιαφέρει (χρονική στιγμή που βρισκόταν στην τοποθεσία, ακρίβεια μέτρησης και απόσταση από την τρέχουσα θέση του ίδιου).
- Να κάνει ασφαλή και γρήγορη ανάκτηση του λογαριασμού του σε περίπτωση κάποιου προβλήματος ή αλλαγής της συσκευής του χωρίς να επηρεαστούν οι εξαρτώμενες συσκευές.
- Να απενεργοποιήσει το λογαριασμό του κυρίως λόγω απώλειας της συσκευής, σβήνοντας από τη βάση όσα στοιχεία αφορούν αυτόν και τα παιδιά του.
- Να ορίσει στην εφαρμογή του και στις εφαρμογές των παιδιών του τη φράση ασφαλείας που θα χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων.
- Να επικοινωνήσει με το διαχειριστή του συστήματος για βοήθεια.

Ο χρήστης της Doodlz εφαρμογής μπορεί:

- Να διασκεδάσει χρησιμοποιώντας την εφαρμογή της ζωγραφικής.

3.4 Βάση δεδομένων

Μία σωστά δομημένη βάση δεδομένων παρέχει πρόσβαση σε ενημερωμένες και ακριβείς πληροφορίες. Επομένως, η σωστή σχεδίασή της είναι ουσιαστικής σημασίας για την υλοποίηση της εφαρμογής και την επίτευξη των στόχων. Έτσι, όταν δουλεύουμε με μια βάση δεδομένων, η επένδυση του χρόνου που απαιτείται για την εκμάθηση των αρχών της καλής σχεδίασης είναι σημαντική.

Υπάρχουν ορισμένες αρχές που καθοδηγούν τη διαδικασία της σχεδίασης βάσης δεδομένων. Η πρώτη αρχή είναι ότι οι διπλότυπες πληροφορίες (που επίσης ονομάζονται πλεονάζοντα δεδομένα) είναι κακές, διότι σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Η δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι

σημαντικές. Εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες, οι εκθέσεις που αντλούν πληροφορίες από τη βάση δεδομένων επίσης θα περιέχουν λανθασμένες πληροφορίες. Ως αποτέλεσμα, οι όποιες αποφάσεις σας βασίζονται σε αυτές τις εκθέσεις θα είναι βασισμένες σε παραπληροφόρηση.

Συνεπώς, υπάρχουν κάποια κομβικά σημεία κατά τη διαδικασία σχεδίασης μιας βάσης δεδομένων:

- Ο καθορισμός του σκοπού που θα εξυπηρετεί η βάση δεδομένων.
- Η εύρεση και οργάνωση των απαιτούμενων πληροφοριών.
- Ο χωρισμός των πληροφοριών σε πίνακες.
- Η μετατροπή των στοιχείων πληροφοριών σε στήλες.
- Ο καθορισμός των πρωτευόντων κλειδιών.
- Η δημιουργία σχέσεων μεταξύ των πινάκων.
- Η βελτίωση της σχεδίασης από τυχόν σφάλματα.
- Η εφαρμογή των κανόνων κανονικοποίησης των δεδομένων.

Με βάση όλα τα παραπάνω και μετά το σχεδιασμό σε χαρτί των πινάκων που θα χρησιμοποιηθούν, το επόμενο βήμα είναι ο σχεδιασμός τους. Για τη σχεδίαση της βάσης χρησιμοποιήθηκε το phpMyAdmin. Για τη δημιουργία του διαγράμματος Οντοτήτων-Συσχετίσεων (ER diagram) χρησιμοποιήθηκε το Microsoft Visio 2013.

3.4.1 Πίνακας PCA_users

Ο πίνακας PCA_users δημιουργήθηκε για την καταχώρηση των χρηστών της MapLocation εφαρμογής στο σύστημα.

	Όνομα	Τύπος	Κενό	Προεπιλογή
	username	varchar(10)	Όχι	Καμία
	password	varchar(100)	Όχι	Καμία
	encryptSalt	varchar(128)	Όχι	Καμία
	parentID	varchar(100)	Όχι	Καμία
	email	varchar(100)	Όχι	Καμία
	newParentID	varchar(100)	Ναι	Κενό

Σχήμα 3.1 PCA_users

username

Πρωτεύον κλειδί του πίνακα, αποτελεί το αναγνωριστικό που χρησιμοποιεί ο εγγεγραμμένος χρήστης της MapLocation εφαρμογής κατά την είσοδό του. Ο τύπος δεδομένων είναι Varchar(10) γιατί μπορεί να πάρει μέχρι 10 χαρακτήρες, οι οποίοι μπορούν να είναι αριθμοί και γράμματα.

password

Στο συγκεκριμένο πεδίο αποθηκεύεται ο κωδικός του κάθε χρήστη, ο οποίος είναι προσωπικός για τον καθένα και τον γνωρίζει μόνο αυτός. Για λόγους ασφαλείας, δεν αποθηκεύεται ο κωδικός στη βάση δεδομένων, αλλά το ψηφιακό αποτύπωμα που δημιουργείται από τη συνάρτηση σύνοψης SHA-256. Ο τύπος δεδομένων είναι Varchar(100).

encryptSalt

Το συγκεκριμένο πεδίο αποτελεί την επιπλέον πληροφορία που χρησιμοποιείται από τη μονόδρομη συνάρτηση σύνοψης SHA-256. Χρειάζεται ώστε να προστατεύσει τον κωδικό του χρήστη και από “επιθέσεις λεξικών” (dictionary attacks) πέρα από τις υπόλοιπες επιθέσεις που προστατεύει η ίδια η συνάρτηση σύνοψης. Το salt έχει δυαδική μορφή αλλά έχει οριστεί να αποθηκεύεται στη βάση με μορφή αλφαριθμητικού. Έτσι, ο τύπος δεδομένων του συγκεκριμένου πεδίου είναι Varchar(128).

parentID

Εδώ αποθηκεύεται το αναγνωριστικό της συσκευής που εισέρχεται στην εφαρμογή MapLocation ο χρήστης και χρησιμοποιείται για να ταυτοποιηθεί στις συσκευές των παιδιών

του. Δημιουργείται τυχαία από την εφαρμογή κατά την εγγραφή και έχει τη μορφή αλφαριθμητικού. Έτσι, χρησιμοποιήθηκε ο τύπος Varchar(100).

email

Αποθηκεύει το email του εγγεγραμμένου χρήστη που θα χρησιμοποιηθεί αν προκύψει ανάγκη για την ανάκτηση του λογαριασμού, αλλά και κατά την επικοινωνία του χρήστη με το διαχειριστή του συστήματος. Είναι ξεχωριστό για κάθε χρήστη και επειδή ο τύπος δεδομένων είναι αλφαριθμητικός, χρησιμοποιήθηκε Varchar(100).

newParentID

Στο συγκεκριμένο πεδίο αποθηκεύεται το νέο αναγνωριστικό της συσκευής, το οποίο χρησιμοποιείται κατά τη διαδικασία της ανάκτησης του λογαριασμού. Έχει τις ίδιες ιδιότητες με το πεδίο parentID, επομένως είναι αλφαριθμητικός τύπος και χρησιμοποιείται το Varchar(100). Αρχικά, το newParentID ορίζεται ως κενό (null).

3.4.2 Πίνακας PCA_devices

Ο πίνακας PCA_devices δημιουργήθηκε για την καταχώρηση της θέσης των συσκευών της Doodlz εφαρμογής.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
	locationNo	int(11)	Όχι	Καμία	AUTO_INCREMENT
	parentID	varchar(100)	Ναι	Κενό	
	childID	varchar(100)	Όχι	Καμία	
	positionTime	timestamp	Όχι	CURRENT_TIMESTAMP	ON UPDATE CUR_TIMESTAMP
	location	varchar(400)	Όχι	Καμία	
	encryptIV	varchar(128)	Όχι	Καμία	
	unlinkDate	timestamp	Ναι	Κενό	

Σχήμα 3.2 PCA_devices

locationNo

Πρωτεύον κλειδί του πίνακα PCA_devices, αυξάνεται κατά ένα με την εισαγωγή νέας συσκευής. Ο τύπος δεδομένων είναι Integer(10), καθώς είναι ακέραιος αριθμός.

parentID

Στο συγκεκριμένο πεδίο ορίζεται το αναγνωριστικό της συσκευής του χρήστη από τον οποίο εξαρτάται η συγκεκριμένη συσκευή. Χρησιμοποιείται για να ξεχωρίζουν οι εξαρτώμενες συσκευές του κάθε γονέα. Ο τύπος δεδομένων είναι ίδιος με το αντίστοιχο πεδίο του πίνακα PCA_users και είναι Varchar(100).

childID

Το συγκεκριμένο πεδίο περιλαμβάνει το αναγνωριστικό της συσκευής του παιδιού που τρέχει η εφαρμογή Doodlz. Το αναγνωριστικό της συσκευής δημιουργείται αυτόματα κατά την εγκατάσταση της εφαρμογής και είναι μοναδική για κάθε συσκευή. Ο τύπος δεδομένων είναι Varchar(100), καθώς περιλαμβάνει αλφαριθμητικά δεδομένα.

positionTime

Στο συγκεκριμένο πεδίο ορίζεται η ημερομηνία που βρισκόταν στη συγκεκριμένη θέση η συσκευή. Περιλαμβάνει δεδομένα τύπου Timestamp γιατί αποθηκεύεται η ημερομηνία και η ώρα της εγγραφής.

location

Αποθηκεύονται όλες οι πληροφορίες που αντιπροσωπεύουν τη θέση της συσκευής. Οι πληροφορίες είναι κρυπτογραφημένες, για αυτό και έχει οριστεί ως τύπος δεδομένων Varchar(400). Η εγγραφή location περιλαμβάνει το γεωγραφικό μήκος και πλάτος της θέσης, καθώς και την ακρίβεια του εντοπισμού.

encryptIV

Στο πεδίο αποθηκεύεται το Διάνυσμα Αρχικοποίησης (Initialization Vector – IV), το οποίο αποτελεί ένα κλειδί που χρησιμοποιείται ως είσοδο στον αλγόριθμο AES για την κρυπτογράφηση των δεδομένων. Το IV ορίζεται ως ένα κλειδί δυαδικής μορφής, αλλά έχει επιλεγεί στη βάση δεδομένων να αποθηκεύεται ως αλφαριθμητικό. Έτσι, ο τύπος δεδομένων του συγκεκριμένου πεδίου είναι Varchar(128).

unlinkDate

Αποθηκεύεται η ημερομηνία που έγινε αίτηση από το χρήστη για απενεργοποίηση του λογαριασμού του. Το συγκεκριμένο πεδίο χρησιμοποιείται στη συνέχεια από τη βάση δεδομένων για να διαγραφεί οριστικά ο λογαριασμός μετά το πέρας μίας μέρας. Ο τύπος

δεδομένων είναι Timestamp, καθώς αποθηκεύεται η ημερομηνία και ώρα. Αρχικά, το unlinkDate ορίζεται ως κενό (null).

3.4.3 Πίνακας PCA_linkCodes

Ο πίνακας PCA_devices δημιουργήθηκε για την καταχώρηση της θέσης των συσκευών της Doodlz εφαρμογής.

	Όνομα	Τύπος	Κενό	Προεπιλογή	Επιπλέον
	parentID	varchar(100)	Όχι	Καμία	
	code	int(6)	Όχι	Καμία	
	linkTime	timestamp	Όχι	CURRENT_TIMESTAMP	ON UPDATE CUR_TIMESTAMP

Σχήμα 3.3 PCA_linkCodes

parentID

Αποθηκεύεται το αναγνωριστικό της συσκευής του χρήστη της MapLocation εφαρμογής που έκανε την αίτηση να προσθέσει κάποια συσκευή. Αποτελεί ουσιαστικά παρόμοιο πεδίο με το parentID των υπόλοιπων πινάκων και χρησιμοποιείται ο τύπος δεδομένων Varchar(100).

code

Περιλαμβάνει τον εξαψήφιο κωδικό που χρησιμοποιείται για το συγχρονισμό των συσκευών των δυο εφαρμογών και δημιουργείται μετά από αίτηση του γονέα. Χρησιμοποιήθηκε τύπος δεδομένων Integer(6) γιατί είναι ακέραιος.

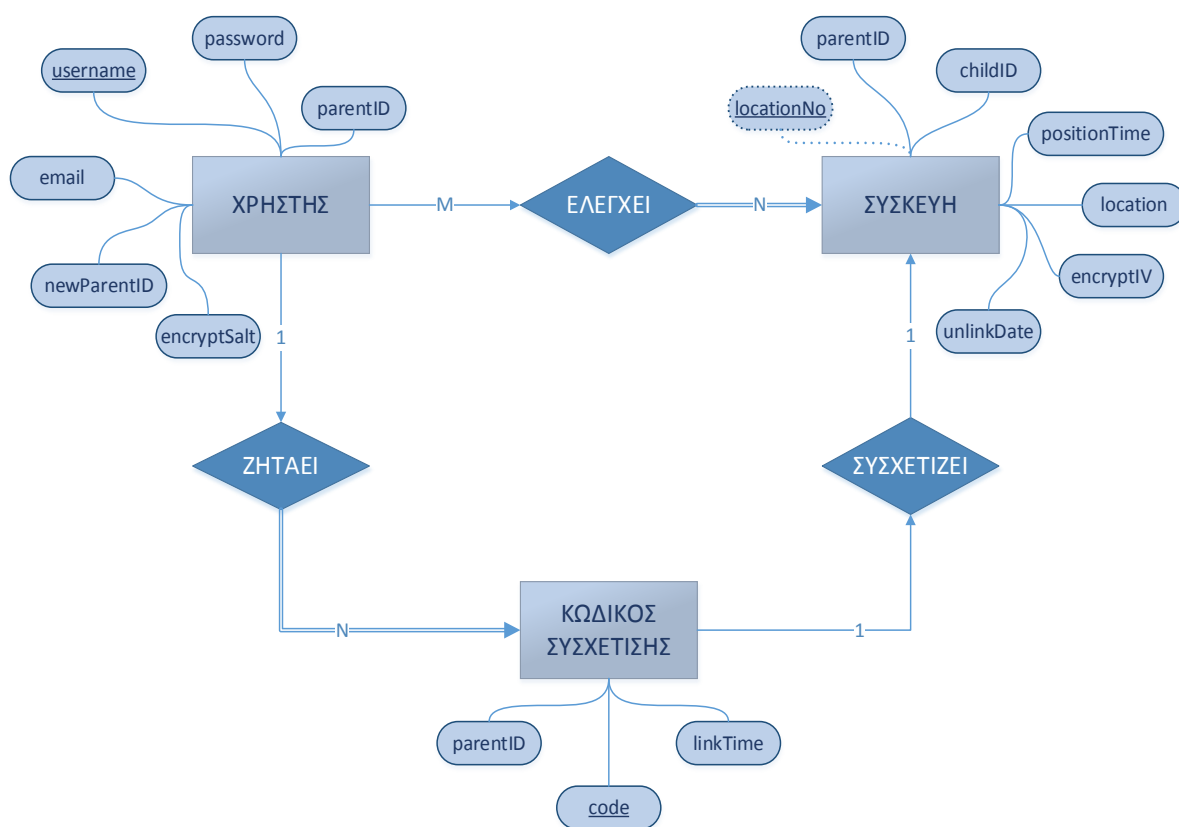
linkTime

Αποθηκεύεται η ημερομηνία και ώρα που έγινε αίτηση για έκδοση κωδικού από το χρήστη της MapLocation εφαρμογής. Χρησιμοποιείται προκειμένου να ελέγχεται η εγγραφή από τη βάση δεδομένων και διαγράφεται αν δεν έχει χρησιμοποιηθεί μετά από 10 λεπτά. Ο τύπος δεδομένων που χρησιμοποιείται είναι Timestamp.

3.5 Πλατφόρμα ανάπτυξης βάσης δεδομένων

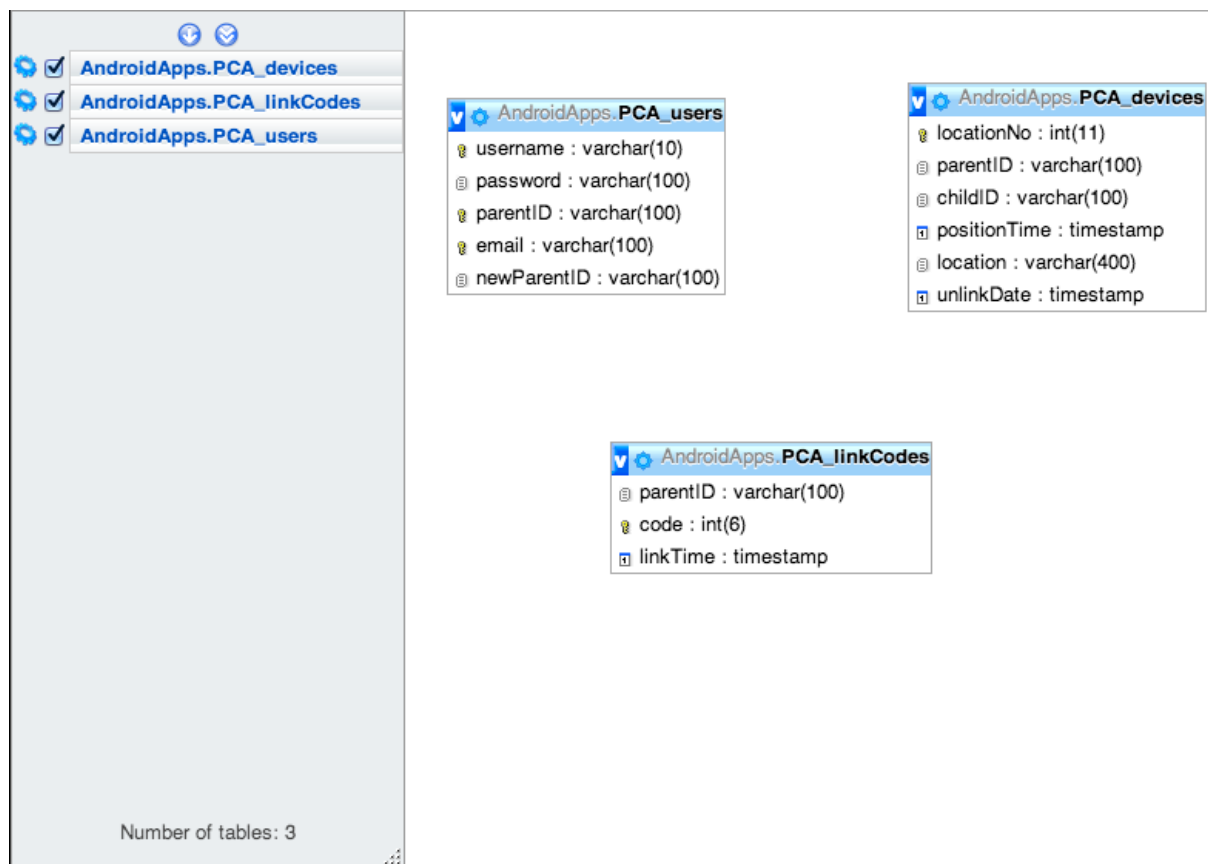
Για τη δημιουργία των πινάκων χρησιμοποιήθηκε αρχικά χαρτί όπως προαναφέρθηκε. Έπειτα, λόγω ευκολίας στη χρήση και των δυνατοτήτων του χρησιμοποιήθηκε το phpMyAdmin, το οποίο παρείχε και έλεγχο στη λειτουργικότητα της βάσης. Τέλος, εφαρμόστηκε σε κάποια σημεία η γλώσσα προγραμματισμού sql που παρέχεται ως δυνατότητα του phpMyAdmin.

3.6 Διάγραμμα οντοτήτων-συσχετίσεων



Σχήμα 3.4 Διάγραμμα οντοτήτων-συσχετίσεων

3.7 Προσχέδιο βάσης δεδομένων



Σχήμα 3.5 Προσχέδιο βάσης δεδομένων

3.8 Περιπτώσεις χρήσης

Παρακάτω αναλύονται οι πιο σημαντικές περιπτώσεις χρήσης του συστήματος.

3.8.1 Προσθήκη/Διαχείριση των εξαρτώμενων χρηστών της Doodlz εφαρμογής

Χρήστης

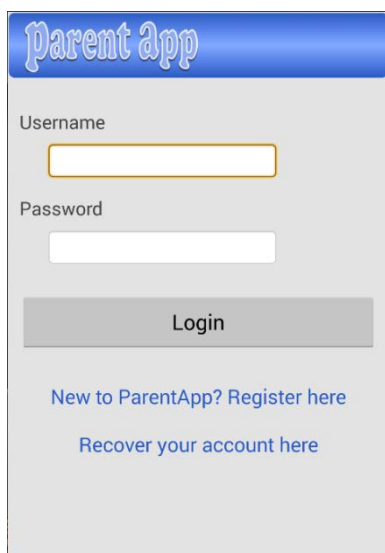
Εγγεγραμμένος χρήστης της MapLocation

Περιγραφή

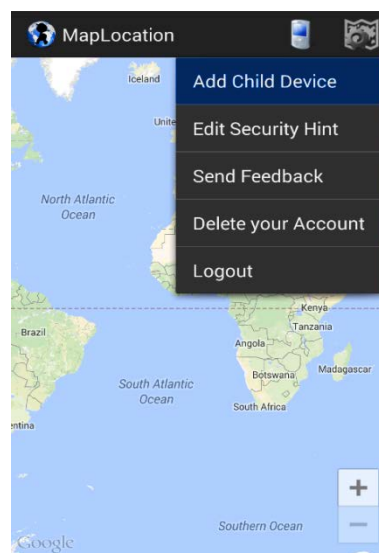
Ο εγγεγραμμένος χρήστης μπορεί να διαχειριστεί τις εξαρτώμενες συσκευές του λογαριασμού του και να εισάγει στο σύστημα κάποιο από τα παιδιά του μέσα από μία διαδικασία συγχρονισμού των δύο συσκευών.

Προϋποθέσεις

Να εισάγει το αναγνωριστικό χρήστη και τον κωδικό του για να εισέλθει στο σύστημα που περιέχονται τα στοιχεία του (Εικόνα 2.1). Να επιλέξει την καρτέλα “Add Child Device” (Εικόνα 2.2).



Εικόνα 3.2 Login



Εικόνα 3.3 Επιλογή Προσθήκης Συσκευής

Σενάριο

Ο γονέας θέλει να προσθέσει τη συσκευή του παιδιού στο λογαριασμό του.

Αποτελέσματα

Πλέον ο γονέας μπορεί να βλέπει την τοποθεσία του παιδιού στην εφαρμογή του.

3.8.2 Εύρεση θέσης παιδιού

Χρήστης

Εγγεγραμμένος χρήστης της MapLocation

Περιγραφή

Ο εγγεγραμμένος χρήστης μπορεί να γνωρίζει τη θέση που βρίσκεται κάποιο από τα παιδιά του ανά πάσα στιγμή.

Προϋποθέσεις

Να γίνει επιλογή του “ίχνους” στο χάρτη που αντιπροσωπεύει το παιδί που επιθυμεί να δει.



Εικόνα 3.4 Εύρεση θέσης του παιδιού στο χάρτη

Σενάριο

Ο γονέας θέλει να μάθει την τελευταία γνωστή τοποθεσία κάποιου από τα παιδιά του και την ακριβή χρονική στιγμή που έγινε. Επίσης, είναι καλό να γνωρίζει την ακριβή χιλιομετρική απόσταση που τον χωρίζει από τα παιδιά του.

Αποτελέσματα

Η θέση του παιδιού εμφανίζεται στο χάρτη της εφαρμογής του με μία πινέζα που το αντιπροσωπεύει και υπολογίζεται και η απόσταση του παιδιού από την τρέχουσα θέση του γονέα. Ο γονέας έχει τη δυνατότητα να πατήσει πάνω στην πινέζα και να παρακολουθήσει στο αναδυόμενο παράθυρο επιπλέον πληροφορίες.

3.8.3 Ορισμός μιας κοινής για τις δύο εφαρμογές φράσης ασφαλείας

Χρήστης

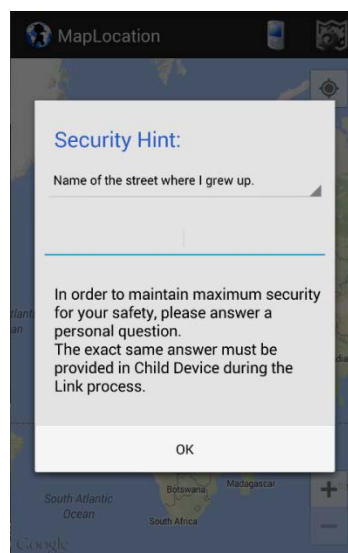
Εγγεγραμμένος χρήστης της MapLocation

Περιγραφή

Ο εγγεγραμμένος χρήστης έχει τη δυνατότητα να ορίσει μία φράση ασφαλείας (security hint), η οποία πρέπει να είναι κοινή για τις δύο εφαρμογές και χρησιμοποιείται για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων.

Προϋποθέσεις

Να έχει εισάγει σωστά αναγνωριστικά εισόδου ώστε να έχει εισέλθει επιτυχώς στο σύστημα και να επιλέξει την καρτέλα “Edit Security Hint”.



Εικόνα 3.5 Καρτέλα ορισμού security hint

Σενάριο

Ο γονέας χρειάζεται να προσθέσει μία νέα εξαρτώμενη συσκευή, αλλά έχει ξεχάσει τη αρχική φράση ασφαλείας που είχε δώσει το σύστημα για την αποκρυπτογράφηση των δεδομένων.

Αποτελέσματα

Επιλέγοντας την καρτέλα “Edit Security Hint” έχει τη δυνατότητα να ορίσει μία φράση ασφαλείας, η οποία θα αντικαταστήσει την προηγούμενη που υπάρχει στη συσκευή.

3.8.4 Ανάκτηση λογαριασμού

Χρήστης

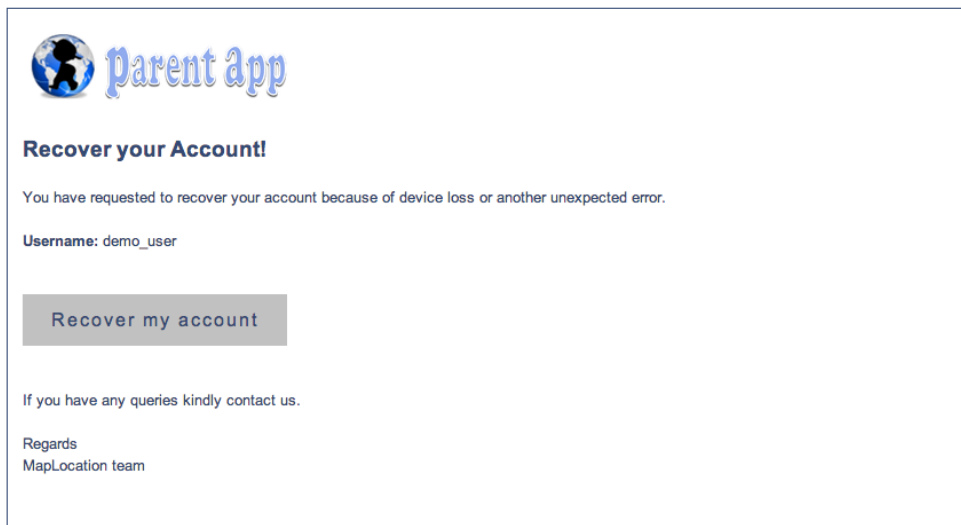
Εγγεγραμμένος χρήστης της MapLocation

Περιγραφή

Ένας εγγεγραμμένος χρήστης μπορεί να ανακτήσει το λογαριασμό πατώντας την αντίστοιχη καρτέλα στην εφαρμογή “MapLocation”. Η συγκεκριμένη λειτουργία είναι χρήσιμη στην περίπτωση που ο χρήστης χρειάζεται να αλλάξει τη συσκευή του και να χρησιμοποιήσει στη νέα την εφαρμογή.

Προϋποθέσεις

Να έχει εισάγει σωστό αναγνωριστικό χρήστη ώστε να του αποσταλεί το email με το σύνδεσμο της ανάκτησης του λογαριασμού. Στη συνέχεια, να επιβεβαιώσει το email ανάκτησης ώστε να ολοκληρωθεί η διαδικασία.



Εικόνα 3.6 Email ανάκτησης λογαριασμού

Σενάριο

Ο γονέας αλλάζει συσκευή και θέλει να εγκαταστήσει την εφαρμογή MapLocation στη νέα χωρίς να χρειάζεται να εγγραφεί και να προσθέσει τις συσκευές από την αρχή.

Αποτελέσματα

Επιλέγοντας την καρτέλα “Recover your account” έχει τη δυνατότητα να ανακτήσει το λογαριασμό του στη νέα συσκευή έχοντας ως μόνη υποχρέωση να εισάγει το αναγνωριστικό χρήστη που είχε ορίσει για την εφαρμογή και να επιβεβαιώσει το σύνδεσμο ανάκτησης από το email που θα του σταλθεί. Στη συνέχεια, μπορεί να συνδεθεί στη νέα συσκευή χωρίς κανένα άλλο πρόβλημα.

3.8.5 Απενεργοποίηση λογαριασμού

Χρήστης

Εγγεγραμμένος χρήστης της MapLocation

Περιγραφή

Ένας εγγεγραμμένος χρήστης θέλει να διαγράψει την εφαρμογή από τη συσκευή του, χωρίς όμως να κρατηθούν τα δικά του στοιχεία αλλά και των παιδιών του στη βάση δεδομένων. Με τη διαδικασία της απενεργοποίησης του λογαριασμού, ο χρήστης επιλέγει να

διαγράψει τις εγγραφές στη βάση που τον αφορούν. Οι εγγραφές σβήνονται μετά από μία ημέρα από την αίτηση της διαγραφής.

Προϋποθέσεις

Να έχει εισάγει σωστά αναγνωριστικά εισόδου ώστε να εισέλθει επιτυχώς στο σύστημα και να επιλέξει την καρτέλα “Delete Account”.

Σενάριο

Ο γονέας επιθυμεί να σταματήσει να χρησιμοποιεί την εφαρμογή MapLocation.

Αποτελέσματα

Επιλέγοντας την καρτέλα “Delete Account” αποστέλλεται το αίτημα στη βάση, το οποίο ολοκληρώνεται μετά από μία ημέρα αδράνειας του λογαριασμού. Μετά το πέρας της μίας ημέρας, τα στοιχεία του χρήστη και όλες οι συσκευές που εξαρτώνται από αυτόν σβήνονται οριστικά.

Κεφάλαιο 4

Εφαρμογή MapLocation

Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της λειτουργίας της εφαρμογής και θα αναλυθούν μερικά από τα πιο σημαντικά στάδια της υλοποίησής της.

4.1 Εισαγωγικά

Η MapLocation δημιουργήθηκε με σκοπό να δώσει στο γονέα μία εφαρμογή εύκολη στη χρήση αλλά και συγχρόνως λειτουργική. Η εφαρμογή απαιτεί να υπάρχει σύνδεση στο Διαδίκτυο είτε μέσω ασύρματου δικτύου (WiFi) είτε μέσω δορυφορικού (3G-4G). Έχει προτιμηθεί και η ενεργοποίηση του GPS (Global Positioning System) προκειμένου να υπάρχει μεγαλύτερη ακρίβεια στη θέση του χρήστη. Τα δεδομένα που στέλνει και λαμβάνει δεν είναι πολλά, επομένως δεν υπάρχει υπερβολική χρέωση. Η κατανάλωση της μπαταρίας είναι μεγάλη, αλλά φτάνει για τη διάρκεια μίας εργάσιμης μέρας.

4.1.1 Συσκευή Smartphone

Η εφαρμογή δημιουργήθηκε χρησιμοποιώντας το πρόγραμμα Eclipse, αφού πρώτα εγκαταστάθηκε το Android Development Toolkit που περιέχει τις δυνατότητες για τον προγραμματισμό σε Android, αλλά και τον εξομοιωτή (Android Virtual Device Manager). Η εφαρμογή είναι διαμορφωμένη να τρέχει σε Android εκδόσεις από 2.2 (API 8) μέχρι και την τελευταία έκδοση πάνω στην οποία και σχεδιάστηκε, την 4.4 (API 19).

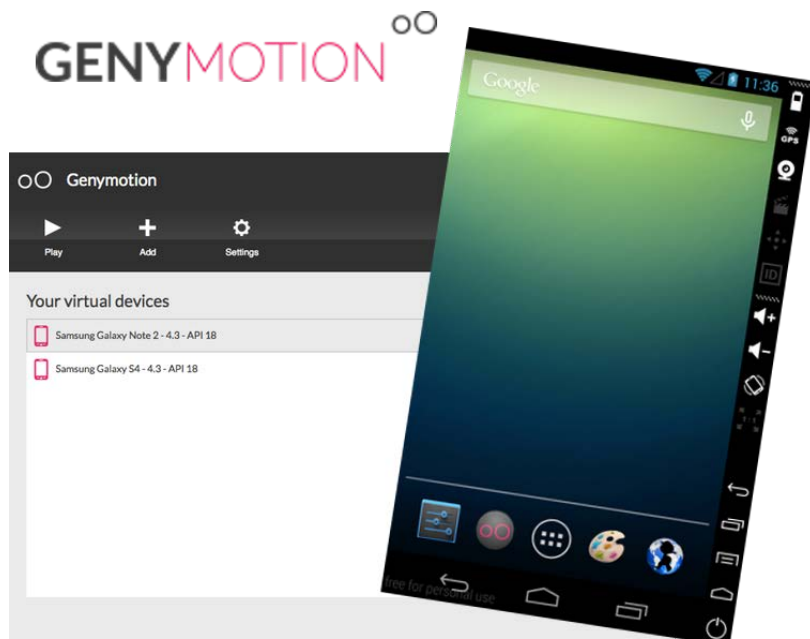
Για τη δοκιμή της, αρχικά χρησιμοποιήθηκε ο εξομοιωτής του Eclipse, αλλά λόγω των διαφόρων προβλημάτων που παρουσιάστηκαν και κυρίως της μη συμβατότητάς του με τους χάρτες της Google, εγκαταλείφθηκε. Στη συνέχεια, χρησιμοποιήθηκε ένας άλλος εξομοιωτής της εταιρίας Genymotion. Η τελική δοκιμή της εφαρμογής πραγματοποιήθηκε σε συσκευή, προκειμένου να ελέγχει η λειτουργικότητα σε πραγματικές συνθήκες. Η συσκευή που χρησιμοποιήθηκε είναι το Sony Xperia P (LT22i).

4.1.1.1 Εξομοιωτής Genymotion

Όπως προαναφέρθηκε, ο συγκεκριμένος εξομοιωτής προτιμήθηκε κυρίως λόγω της συμβατότητάς του με την παρουσίαση των χαρτών της Google. Όπως αποδείχτηκε, ήταν αρκετά γρήγορος και εύκολος στη χρήση. Αρχικά, στην κεντρική σελίδα της εφαρμογής παρουσιάζονται όλοι οι εγκατεστημένοι εξομοιωτές του χρήστη και δίνεται η δυνατότητα να προστεθούν νέοι. Ένας εξομοιωτής περιλαμβάνει τις δυνατότητες διαφόρων μοντέλων που υπάρχουν στην αγορά. Για τις δύο εφαρμογές που δημιουργήθηκαν στα πλαίσια της παρούσας διπλωματικής, ο εξομοιωτής που χρησιμοποιήθηκε είναι σχεδιασμένος με βάση το Samsung Galaxy S4. Η Android έκδοση που τρέχει είναι η 4.3 (API 18).

Πριν την πρώτη χρήση του εξομοιωτή, έπρεπε να εγκατασταθούν οι εφαρμογές της Google μέρος των οποίων είναι και οι χάρτες. Τα βήματα που χρειάστηκε να γίνουν για την εγκατάστασή τους αναλύονται παρακάτω.

1. Να γίνει λήψη δύο .zip αρχείων:
 - Genymotion ARM Translation Installer v1.1
 - Google Apps for Android 4.3 (gapps)
2. Εγκαθίσταται το πρώτο αρχείο ARM Translation Installer στον εξομοιωτή.
3. Γίνεται επανεκκίνηση του εξομοιωτή προκειμένου να ολοκληρωθεί η εγκατάσταση.
4. Εγκαθίσταται το δεύτερο αρχείο gapps στον εξομοιωτή.
5. Γίνεται άλλη μία φορά επανεκκίνηση για την ολοκλήρωση της εγκατάστασης.
6. Μόλις ξεκινήσει ο εξομοιωτής, επιλέγεται η εφαρμογή Google Play Store.
7. Γίνεται είσοδος με τα στοιχεία του λογαριασμού Google του χρήστη.
8. Αναβαθμίζονται τα Google Play Services.
9. Πλέον μπορούν να χρησιμοποιηθούν οι χάρτες της Google.



Εικόνα 4.1 Εξομοιωτής Genymotion

4.1.1.2 Συσκευή Sony Xperia P

Η συσκευή χρησιμοποιήθηκε για την τελική δοκιμή των δύο εφαρμογών, ώστε να ελεγχθεί η λειτουργικότητά τους και να βελτιωθούν τυχόν προβλήματα που θα προέκυπταν. Η συσκευή που χρησιμοποιήθηκε είναι το Sony Xperia P και η έκδοση Android που έτρεχε είναι η 4.1 (API 16).

Τα χαρακτηριστικά της συσκευής είναι τα εξής:

- Λειτουργικό σύστημα: Google Android 4.1.2
- Chipset (SoC): STE NovaThor U8500
- CPU: Διπύρηνος ARM Cortex-A9 στο 1,00 GHz
- GPU: ARM Mali-400 MP
- Μνήμη RAM: 1 GB
- Αποθηκευτικός χώρος: 2 GB
- Διαστάσεις: 122 x 59,5 x 10,6 χιλιοστά
- Βάρος: 122 γραμμάρια
- Οθόνη αφής
- Τύπος οθόνης: LCD

- Ανάλυση: 540 x 960 pixels
- Χρώματα: 16,7 εκατομμύρια
- WiFi: IEEE 802.11 b/g/n
- Εντοπισμός θέσης: GPS, A-GPS
- Δίκτυα:
 - 2G: GSM
 - 3G: HSDPA / UMTS
- Τύπος μπαταρίας: Li-Ion

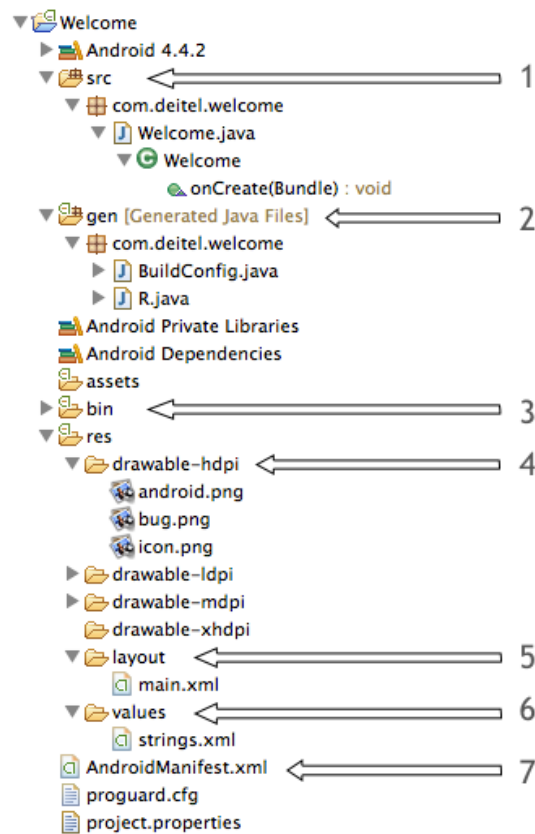


Εικόνα 4.2 Sony Xperia P

4.2 Ανατομία μίας εφαρμογής Android

Για τη δημιουργία μίας εφαρμογής Android είναι απαραίτητη η κατανόηση των αρχείων που την αποτελούν, των εννοιών της δραστηριότητας (activity) και της υπηρεσίας (service) και πώς αυτές αλληλεπιδρούν μεταξύ τους.

4.2.1 Δομή μίας εφαρμογής



Εικόνα 4.3 Δομή Android εφαρμογής

Τα βασικά αρχεία και οι κατάλογοι που θα πρέπει να γνωρίζουμε πριν τη δημιουργία μίας εφαρμογής και θα αναλυθούν παρακάτω, είναι τα εξής:

1. Φάκελος src:

Προέρχεται από το Source files. Περιέχει όλο τον πηγαίο κώδικα της εφαρμογής. Κατά τη δημιουργία μίας εφαρμογής δημιουργείται αυτόματα και το αρχείο MainActivity.java ή όπως αλλιώς το ονομάσει ο χρήστης, το οποίο καλείται κατά την εκτέλεση μόλις πατηθεί το εικονίδιο της εφαρμογής.

2. Φάκελος gen:

Προέρχεται από το Generated files. Περιλαμβάνει αρχεία που δημιουργούνται αυτόματα κατά την πρώτη εκτέλεση του προγράμματος. Το βασικό αρχείο που δημιουργείται είναι το R.java, το οποίο έχει μεταγλωττιστεί και αναφέρει όλους τους πόρους που υπάρχουν στην εφαρμογή.

3. Φάκελος bin:

Το όνομά του προκύπτει από το Binary files. Περιλαμβάνει όλα τα apk πακέτα που χρειάζεται η εφαρμογή για να τρέξει.

4. Φάκελος `res/drawable-hdpi`:

Ο φάκελος `drawable-hdpi` είναι ένας κατάλογος με σχεδιασμένα αντικείμενα για οθόνες υψηλής ευκρίνειας.

5. Φάκελος `res/layout`:

Περιλαμβάνει αρχεία που καθορίζουν το περιβάλλον του χρήστη.

6. Φάκελος `res/values`:

Αποτελεί έναν κατάλογο με αρχεία `xml` που περιέχουν συλλογή από πόρους, όπως για παράδειγμα οι φράσεις που χρησιμοποιούνται στην εφαρμογή (`strings.xml`) ή τα χρώματα (`colors.xml`).

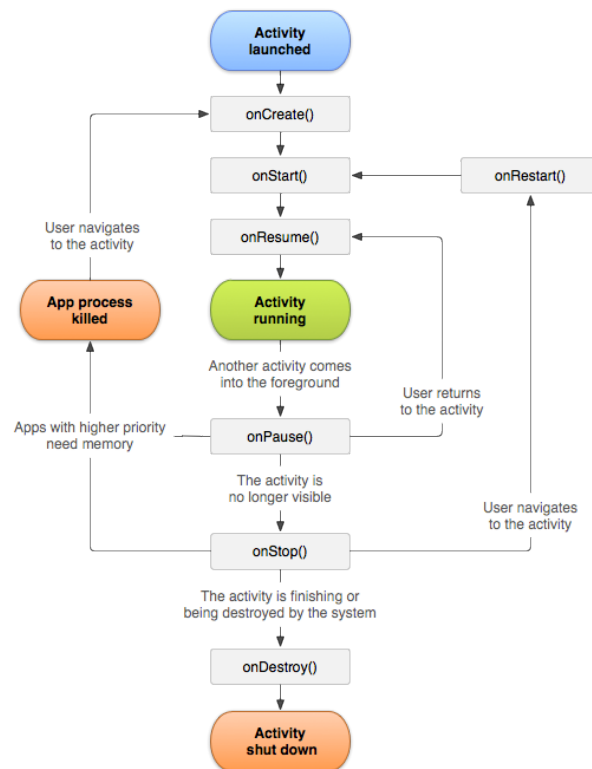
7. Αρχείο `AndroidManifest.xml`:

Αποτελεί ένα `xml` αρχείο που περιγράφει τα θεμελιώδη χαρακτηριστικά της εφαρμογής.

4.2.2 Activity

Η κάθε δραστηριότητα αντιπροσωπεύει μία οθόνη με την οποία μπορεί να αλληλεπιδράσει ο χρήστης. Μία δραστηριότητα καλείται είτε με το άνοιγμα της εφαρμογής είτε μέσα από κάποια άλλη δραστηριότητα. Οι πιο βασικές μέθοδοι που χρησιμοποιούνται σε μία δραστηριότητα είναι οι ακόλουθες:

- `onCreate()`: Καλείται όταν δημιουργείται η εφαρμογή.
- `onStart()`: Καλείται όταν ξεκινάει η εφαρμογή και μπορεί ο χρήστης να δει την κεντρική οθόνη.
- `onResume()`: Καλείται όταν ο χρήστης ξεκινά να αλληλεπιδρά με την εφαρμογή.
- `onStop()`: Καλείται όταν ο χρήστης δεν μπορεί να δει καμία οθόνη της εφαρμογής.
- `onDestroy()`: Καλείται πριν σταματήσει η λειτουργία της εφαρμογής και αποδεσμεύονται οι πόροι.
- `onRestart()`: Καλείται όταν γίνεται επανεκκίνηση της εφαρμογής.

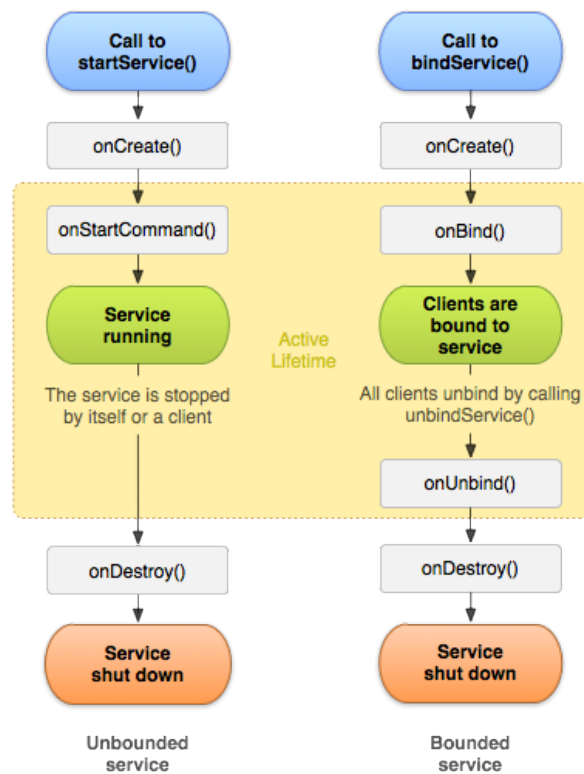


Εικόνα 4.4 Κύκλος ζωής ενός activity

4.2.3 Service

Ένα service αποτελεί ένα στοιχείο που τρέχει στο παρασκήνιο για την εκτέλεση λειτουργιών μακράς διάρκειας χωρίς να χρειάζεται να αλληλεπιδράσει με το χρήστη. Οι πιο βασικές μέθοδοι που χρησιμοποιούνται σε μία υπηρεσία είναι οι ακόλουθες:

- `onCreate()`: Καλείται όταν δημιουργείται για πρώτη φορά η υπηρεσία. Εάν η υπηρεσία είναι ήδη σε λειτουργία, η συγκεκριμένη μέθοδος δεν καλείται.
- `onDestroy()`: Καλείται όταν η υπηρεσία δε χρησιμοποιείται πλέον και καταστρέφεται. Χρησιμοποιείται για να αποδεσμευτούν τυχόν πόροι και αποτελεί την τελευταία κλήση ενός service.
- `onBind()`: Καλείται όταν κάποιο άλλο συστατικό θέλει να συνδεθεί με το συγκεκριμένο service μέσα από τη μέθοδο `bindService()`.



Εικόνα 4.5 Διάρκεια ζωής ενός service

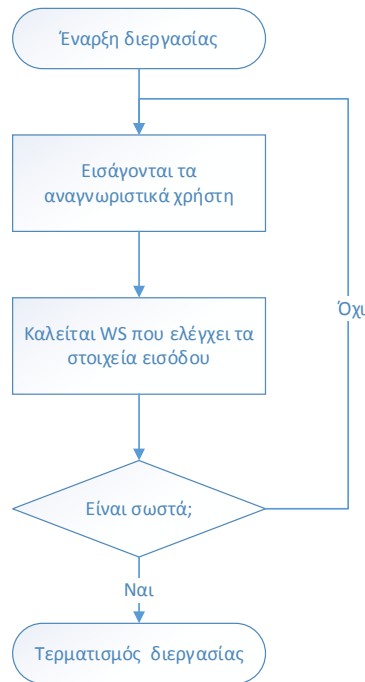
4.3 Διαγράμματα λειτουργίας MapLocation

Οι βασικές λειτουργίες που καλύπτει η εφαρμογή MapLocation αναφέρονται παρακάτω:

- Σύνδεση χρήστη
- Εγγραφή χρήστη
- Ανάκτηση λογαριασμού
- Κεντρική οθόνη της εφαρμογής με την εμφάνιση του χάρτη
- Προσθήκη εξαρτώμενης συσκευής
- Επεξεργασία κωδικού ασφαλείας
- Αποστολή σχολίων
- Απενεργοποίηση του λογαριασμού

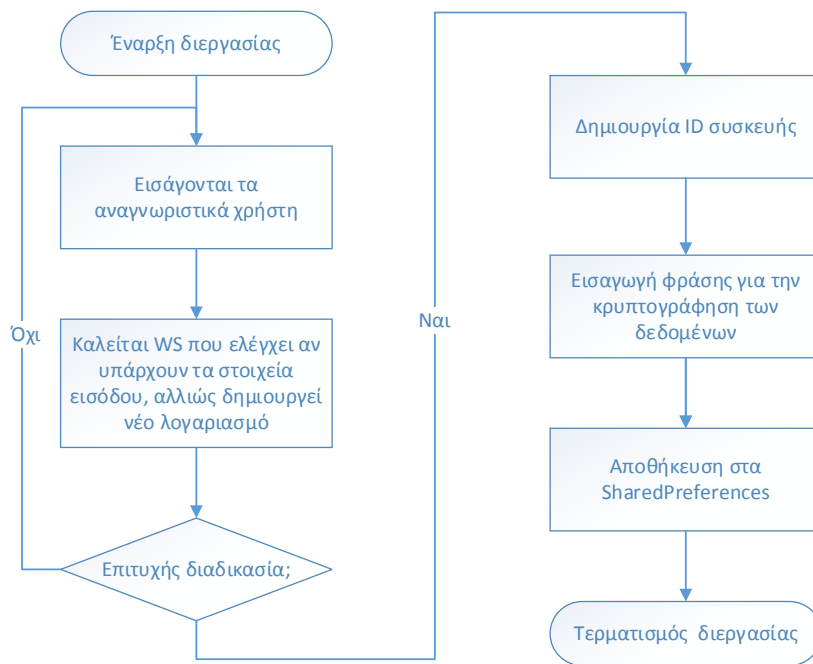
Τα παρακάτω διαγράμματα παρουσιάζουν γραφικά τις βασικές λειτουργίες της εφαρμογής που αναφέρθηκαν παραπάνω και στη συνέχεια θα γίνει μία περιγραφή της κάθε λειτουργίας ξεχωριστά.

4.3.1 Σύνδεση χρήστη



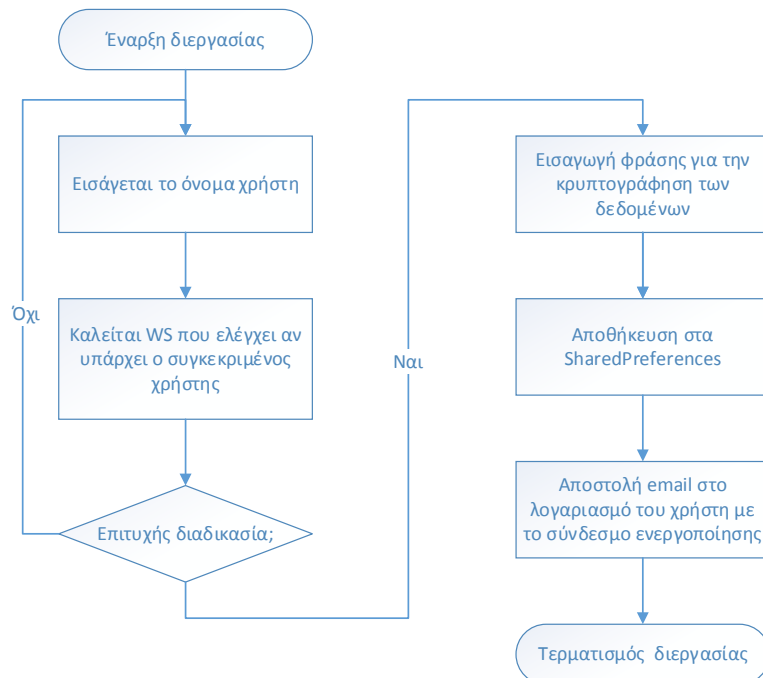
Σχήμα 4.1 Σύνδεση χρήστη

4.3.2 Εγγραφή χρήστη



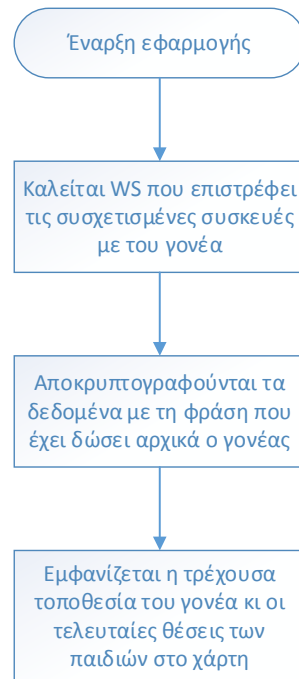
Σχήμα 4.2 Εγγραφή χρήστη

4.3.3 Ανάκτηση λογαριασμού χρήστη



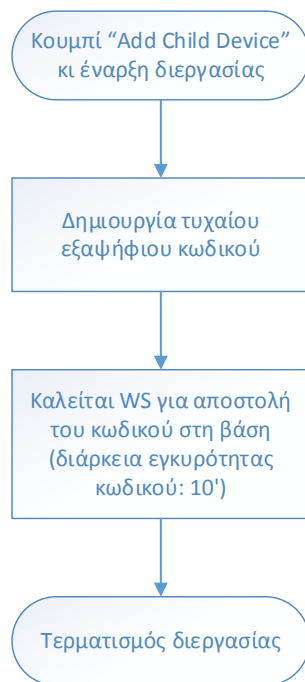
Σχήμα 4.3 Ανάκτηση λογαριασμού χρήστη

4.3.4 Κεντρική οθόνη εφαρμογής



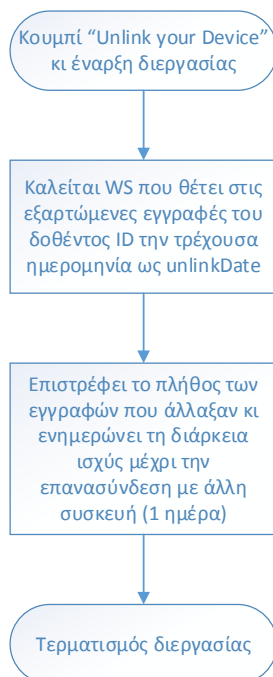
Σχήμα 4.4 Κεντρική οθόνη εφαρμογής

4.3.5 Προσθήκη εξαρτώμενης συσκευής



Σχήμα 4.5 Προσθήκη συσκευής

4.3.6 Απενεργοποίηση λογαριασμού



Σχήμα 4.6 Απενεργοποίηση λογαριασμού

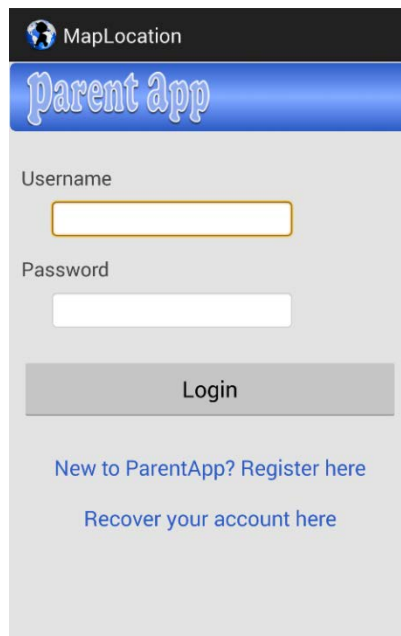
4.4 Προγραμματισμός-Λειτουργία της εφαρμογής

Παρακάτω αναφέρεται αναλυτικά η λειτουργία της εφαρμογής και παραπέμπονται σημαντικά κομμάτια κώδικα, τα οποία και αναλύονται.

4.4.1 Σύνδεση-Αποσύνδεση χρήστη

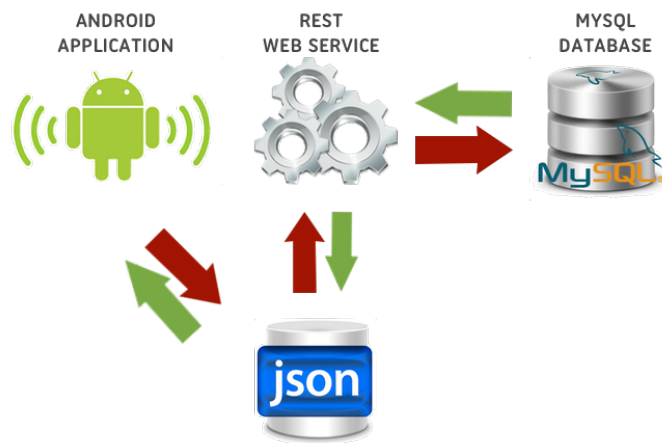
Αποτελεί την πρώτη οθόνη που εμφανίζεται όταν ο χρήστης ενεργοποιεί την εφαρμογή. Για να συνδεθεί θα πρέπει να συμπληρώσει τα αναγνωριστικά πρόσβασης. Στη συνέχεια, πατώντας το κουμπί Login ελέγχεται η εγκυρότητα των στοιχείων που έχουν δοθεί και αν η εισαγωγή είναι σωστή ανακατευθύνεται στην προσωπική αρχική του σελίδα. Σε περίπτωση λάθους εμφανίζεται ανάλογο μήνυμα που ενημερώνει αν ο κωδικός πρόσβασης είναι εσφαλμένος, αν δεν υπάρχει το όνομα χρήστη ή αν εισέρχεται στην εφαρμογή από άγνωστη συσκευή.

Στην καρτέλα login δίνεται επίσης η δυνατότητα για ανακατεύθυνση στις σελίδες εγγραφής χρήστη και ανάκτησης λογαριασμού.



Εικόνα 4.6 Είσοδος χρήστη

Πατώντας το κουμπί Login γίνεται η διαδικασία που περιγράφεται στη συνέχεια. Τα στοιχεία που ορίστηκαν ως όνομα χρήστη και κωδικός μαζί με το αναγνωριστικό της συσκευής στέλνονται στο διακομιστή με τον οποίο είναι συνδεδεμένη η εφαρμογή. Επειδή δεν μπορεί να πραγματοποιηθεί επικοινωνία μεταξύ android εφαρμογής και βάσης δεδομένων, χρησιμοποιείται ενδιάμεσος κώδικας που μετατρέπει τα δεδομένα σε κατανοητή μορφή και για τις δύο πλευρές.



Εικόνα 4.7 Επικοινωνία Android-Mysql Database

Τα δεδομένα στέλνονται από την εφαρμογή με μορφή json, ένα Restful Web Service τα επεξεργάζεται και συνδέεται με τη βάση δεδομένων για να κάνει τον έλεγχο. Αφού γίνει ο έλεγχος, τα στοιχεία αποθηκεύονται σε μεταβλητές με κωδικοποίηση json τα οποία μπορεί να επεξεργαστεί η εφαρμογή.

Συγκεκριμένα, το όνομα χρήστη, ο κωδικός και το id της συσκευής στέλνονται με HTTP_POST στο διακομιστή. Ο διακομιστής λαμβάνει τα δεδομένα και ενεργοποιείται μία συνάρτηση, η οποία αρχικά ελέγχει στη βάση δεδομένων αν υπάρχει ο συγκεκριμένος χρήστης και αν βρεθεί, χρησιμοποιείται η τιμή του πεδίου encryptSalt για να δημιουργηθεί από τη συνάρτηση σύνοψης SHA-256 το ψηφιακό αποτύπωμα του κωδικού. Σε περίπτωση που δεν υπάρχει το δοθέν όνομα χρήστη, επιστρέφεται η αντίστοιχη τιμή στη μεταβλητή result και η συνάρτηση τερματίζεται. Αν υπάρχει ο συγκεκριμένος χρήστης, ελέγχονται τα υπόλοιπα στοιχεία που έχουν δοθεί. Αν είναι σωστά, η μεταβλητή result επιστρέφει την αντίστοιχη τιμή στην android εφαρμογή επιτρέποντας στο χρήστη να εισέλθει στην προσωπική του σελίδα. Οι τιμές που μπορεί να πάρει η μεταβλητή result αναφέρονται παρακάτω:

- Αν είναι όλα σωστά, η μεταβλητή result παίρνει την τιμή 0.
- Αν δεν υπάρχει ο συγκεκριμένος χρήστης, επιστρέφεται η τιμή 105.
- Αν ο κωδικός που δόθηκε είναι λάθος, η result γίνεται 104.
- Αν ο συγκεκριμένος λογαριασμός είναι ανενεργός, η μεταβλητή result έχει τιμή 106.
- Αν ο χρήστης προσπαθεί να συνδεθεί από άγνωστη συσκευή, η result παίρνει την τιμή 110.

Η android εφαρμογή παίρνει την επιστρεφόμενη τιμή και εμφανίζει το αντίστοιχο μήνυμα στο χρήστη. Σε περίπτωση που έχουν δοθεί σωστά στοιχεία εισόδου, ο χρήστης ανακατευθύνεται στην προσωπική του σελίδα.

Αποσύνδεση κάνει από την καρτέλα logout που υπάρχει στις ρυθμίσεις της προσωπικής του σελίδας. Μόλις πατηθεί το logout, η android εφαρμογή τον επιστρέφει στην αρχική σελίδα σύνδεσης χρήστη.

4.4.1.1 Σημαντικά σημεία κώδικα

Ο έλεγχος αν είναι σωστά τα στοιχεία εισαγωγής είναι από τα πιο σημαντικά σημεία. Στη συγκεκριμένη εικόνα παρουσιάζεται η συνάρτηση εισόδου login που έχει υλοποιηθεί στο διακομιστή, η οποία παίρνει ως παράμετρο τα στοιχεία εισόδου (LoginCredentials) και επιστρέφει έναν ακέραιο που είναι το αποτέλεσμα της συγκεκριμένης διαδικασίας (result).

Μόλις ενεργοποιείται το Web Service από την εφαρμογή, η συγκεκριμένη συνάρτηση λαμβάνει από τη βάση δεδομένων όλους τους χρήστες και τους αποθηκεύει σε μία λίστα. Στη συνέχεια, αφού χρησιμοποιηθεί η συνάρτηση σύνοψης για την κωδικοποίηση του κωδικού, γίνεται μία σειρά ελέγχων σχετικά με το αν έχουν δοθεί σωστά στοιχεία εισόδου και επιστρέφεται η μεταβλητή result με τον κωδικό λάθους που προέκυψε ή τον αριθμό 0 αν η απόπειρα σύνδεσης είναι επιτυχής.

```
@POST
@Path("/login")
@Produces({"application/json"})
@Consumes({"application/json"})
public Integer login(LoginCredentials lc) throws Exception {
    Integer result = -1;

    PCAusers user = (PCAusers) em.find(PCAusers.class, lc.username);

    if (user == null) {
        result = 105; // username not exist
        return result;
    }

    BASE64Decoder decoder = new BASE64Decoder();
    String encrSalt = user.getEncryptSalt();
    byte[] salt = decoder.decodeBuffer(encrSalt);
    String encrPass = SHAencrypt(lc.password, salt);

    if (user.getPassword().equals(encrPass)) {
        if (user.getNewParentID() != null) {
            result = 106; // inactive account
            return result;
        }
        else if (!user.getParentID().equals(lc.uuid)) {
            result = 110; // unknown device
            return result;
        }

        result = 0; // success
        return result;
    }

    result = 104; // incorrect password
    return result;
}
```

4.4.2 Εγγραφή χρήστη

Επιλέγοντας από την αρχική καρτέλα login, το σύνδεσμο “New to ParentApp? Register here”, ο χρήστης ανακατευθύνεται στη σελίδα εγγραφής. Στη συγκεκριμένη οθόνη, συμπληρώνονται τα επιθυμητά αναγνωριστικά που θα χρησιμοποιεί για την πρόσβαση του στην εφαρμογή. Στη συνέχεια, πατώντας το κουμπί Register New Account ελέγχεται η εγκυρότητα των στοιχείων που έχουν δοθεί και αν είναι σωστά ενεργοποιείται μία νέα καρτέλα που ζητάει από το χρήστη να ορίσει τη φράση ασφαλείας που θα χρησιμοποιείται για την ασφαλή επικοινωνία με την εφαρμογή των παιδιών του.

Στην καρτέλα register δίνεται η δυνατότητα για ανακατεύθυνση του χρήστη στην οθόνη σύνδεσης.

Εικόνα 4.8 Εγγραφή χρήστη

Πατώντας το κουμπί Register New Account γίνεται η παρακάτω διαδικασία. Τα στοιχεία που ορίστηκαν ως όνομα χρήστη, κωδικός και email μαζί με το αναγνωριστικό της συσκευής στέλνονται στο διακομιστή με τον οποίο είναι συνδεδεμένη η εφαρμογή.

Τα δεδομένα στέλνονται από την εφαρμογή με μορφή json, ένα Restful Web Service τα επεξεργάζεται και συνδέεται με τη βάση δεδομένων για να κάνει τον έλεγχο.

Συγκεκριμένα, το όνομα χρήστη, ο κωδικός, το email και το id της συσκευής στέλνονται με HTTP_POST στο διακομιστή. Ο διακομιστής λαμβάνει τα δεδομένα και ενεργοποιείται μία συνάρτηση, η οποία αρχικά ελέγχει στη βάση δεδομένων αν υπάρχει συσκευή, όνομα χρήστη ή email που να είναι ίδια με τα στοιχεία που δόθηκαν. Αν υπάρχει κάποιο από αυτά, επιστρέφεται η αντίστοιχη τιμή σε μία μεταβλητή result και η συνάρτηση τερματίζεται. Αν κανένα από τα στοιχεία δεν υπάρχουν, δημιουργείται τυχαία η επιπλέον πληροφορία που θα χρησιμοποιηθεί για την κρυπτογράφηση του κωδικού πρόσβασης μέσω της συνάρτησης σύνοψης SHA-256. Τέλος, εισάγεται ο χρήστης στη βάση δεδομένων και επιστρέφεται η μεταβλητή result στην android εφαρμογή. Οι τιμές που μπορεί να πάρει η μεταβλητή result αναφέρονται παρακάτω:

- Αν δημιουργήθηκε ο νέος χρήστης, η μεταβλητή result παίρνει την τιμή 0.
- Αν υπάρχει λογαριασμός συνδεδεμένος με το αναγνωριστικό της συσκευής που δόθηκε, επιστρέφεται η τιμή 101.

- Αν υπάρχει το συγκεκριμένο αναγνωριστικό χρήστη, η result γίνεται 102.
- Αν υπάρχει λογαριασμός με το συγκεκριμένο email, η μεταβλητή result έχει τιμή 103.

Η android εφαρμογή παίρνει την επιστρεφόμενη τιμή και εμφανίζει το αντίστοιχο μήνυμα στο χρήστη. Σε περίπτωση που έχουν δοθεί σωστά στοιχεία εισόδου, ο χρήστης ανακατευθύνεται στη δημιουργία της φράσης ασφαλείας.

4.4.2.1 Σημαντικά σημεία κώδικα

Στη συγκεκριμένη εικόνα παρουσιάζεται η συνάρτηση εγγραφής addUser που αποτελεί μέρος του Web Service, η οποία παίρνει ως παράμετρο τα στοιχεία εγγραφής (SignupCredentials) και επιστρέφει έναν ακέραιο που είναι το αποτέλεσμα της συγκεκριμένης διαδικασίας (result).

Συγκεκριμένα, η συνάρτηση εγγραφής λαμβάνει από τη βάση δεδομένων όλους τους χρήστες και τους αποθηκεύει σε μία λίστα. Στη συνέχεια, ελέγχει αν κάποιο από τα στοιχεία που δόθηκαν χρησιμοποιούνται από ήδη υπάρχων χρήστη και επιστρέφει μήνυμα λάθους. Σε αντίθετη περίπτωση, δημιουργείται τυχαία η επιπλέον πληροφορία για τη συνάρτηση σύνοψης και κρυπτογραφείται ο κωδικός πρόσβασης. Τέλος, καλείται η συνάρτηση create() για να δημιουργήσει και να αποθηκεύσει το χρήστη στη βάση δεδομένων.

```
@POST
@Path("/addUser")
@Produces({"application/json"})
@Consumes({"application/json"})
public int addUser(SignupCredentials sc) throws Exception {
    int result = 0;
    List<PCAusers> users = super.findAll();

    for (PCAusers u:users) {
        if (u.getParentID().equals(sc.parentID)) {
            result = 101; // parentID exists
            return result;
        }
        if (u.getUsername().equalsIgnoreCase(sc.username)) {
            result = 102; // username exists
            return result;
        }
        if (u.getEmail().equalsIgnoreCase(sc.email)) {
            result = 103; // email exists
            return result;
        }
    }

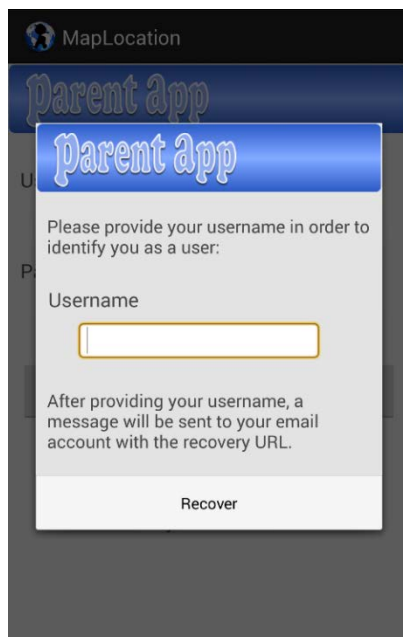
    byte[] salt = getSalt();
    String encrPass = SHAEncrypt(sc.password, salt);

    BASE64Encoder encoder = new BASE64Encoder();
    String encrSalt = encoder.encodeBuffer(salt);
    PCAusers user = new PCAusers(sc.username, encrPass, encrSalt,
    sc.parentID, sc.email);
    super.create(user);

    return result;
}
```

4.4.3 Ανάκτηση λογαριασμού

Επιλέγοντας από την αρχική καρτέλα login, το σύνδεσμο “Recover your account here”, ο χρήστης ανακατευθύνεται στη σελίδα ανάκτησης του λογαριασμού του. Στη συγκεκριμένη οθόνη, πρέπει να συμπληρώσει το αναγνωριστικό χρήστη. Στη συνέχεια, πατώντας το κουμπί Recover, ελέγχεται αν έχει συμπληρωθεί το πεδίο και καλείται το Web Service. Αν η διαδικασία ολοκληρωθεί με επιτυχία, επιστρέφεται μήνυμα στο χρήστη ότι του έχει αποσταλεί email με το σύνδεσμο ενεργοποίησης του λογαριασμού του, αλλιώς του εμφανίζεται το λάθος που προέκυψε.



Εικόνα 4.9 Ανάκτηση λογαριασμού

Αφού ο χρήστης συμπληρώσει στο πεδίο το αναγνωριστικό του και πατήσει το κουμπί Recover, η εφαρμογή στέλνει στο διακομιστή το περιεχόμενο του πεδίου και το αναγνωριστικό της συσκευής που το ζήτησε σε μορφή json. Στη συνέχεια, το Web Service επεξεργάζεται τα δεδομένα που έλαβε και συνδέεται με τη βάση για να κάνει τους απαραίτητους ελέγχους.

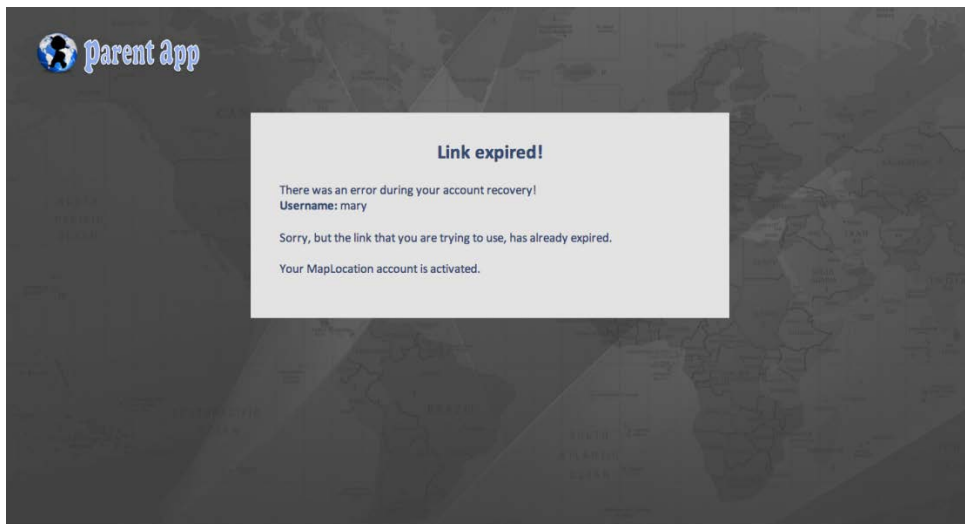
Συγκεκριμένα, το όνομα χρήστη και το id της συσκευής στέλνονται με HTTP_GET στο διακομιστή. Ο διακομιστής λαμβάνει τα δεδομένα και ενεργοποιείται μία συνάρτηση, η οποία αρχικά ελέγχει στη βάση δεδομένων αν υπάρχει ο συγκεκριμένος χρήστης και αν βρεθεί, καλείται μία άλλη συνάρτηση, η οποία χρησιμοποιώντας την τιμή του πεδίου email ως παραλήπτη, θα στείλει στο χρήστη ένα μήνυμα ταχυδρομείου με το σύνδεσμο ενεργοποίησης του. Το αναγνωριστικό της συσκευής αποθηκεύεται στη βάση δεδομένων, καθώς θα χρησιμοποιηθεί μεταγενέστερα στη διαδικασία ενεργοποίησης. Και σε αυτή τη μέθοδο υπάρχει μία μεταβλητή result, η οποία θα επιστραφεί στην android εφαρμογή για να ειδοποιήσει το χρήστη για τα αποτελέσματα της διαδικασίας. Οι τιμές που μπορεί να πάρει η μεταβλητή result αναφέρονται παρακάτω:

- Αν έχει σταλθεί σωστά το email, η μεταβλητή result παίρνει την τιμή 0.
- Αν έχει γίνει ήδη κάποια αίτηση ανάκτησης, επιστρέφεται ο κωδικός 107.

- Αν το δοθέν αναγνωριστικό συσκευής χρησιμοποιείται ήδη, η μεταβλητή result έχει τιμή 108.

Η android εφαρμογή παίρνει την επιστρεφόμενη τιμή και εμφανίζει το αντίστοιχο μήνυμα στο χρήστη. Σε περίπτωση που έχει σταλθεί με επιτυχία ο σύνδεσμος ανάκτησης του λογαριασμού, ο χρήστης ανακατευθύνεται στην καρτέλα εισαγωγής της φράσης ασφαλείας.

Ο σύνδεσμος ενεργοποίησης που στέλνεται στο email του χρήστη αποτελεί ουσιαστικά μία νέα μέθοδο, η οποία καλείται με το πάτημα του συνδέσμου και ολοκληρώνει τη διαδικασία ανάκτησης του λογαριασμού, αφού πρώτα κάνει τους απαραίτητους ελέγχους. Εμφανίζει τα αποτελέσματα της ανάκτησης στο χρήστη σε μορφή html μέσα από κάποιο πρόγραμμα περιήγησης.



Εικόνα 4.10 Αποτέλεσμα ανάκτησης λογαριασμού

4.4.3.1 Σημαντικά σημεία κώδικα

Στη συγκεκριμένη εικόνα παρουσιάζεται η συνάρτηση ανάκτησης recover που έχει υλοποιηθεί ως μέρος του Web Service, η οποία παίρνει ως παραμέτρους τα αναγνωριστικά χρήστη και συσκευής και επιστρέφει έναν ακέραιο που είναι το αποτέλεσμα της συγκεκριμένης διαδικασίας (result).

Αρχικά, γίνονται οι απαραίτητοι έλεγχοι αν υπάρχει ο συγκεκριμένος χρήστης και αν δεν έχει γίνει προηγουμένως κάποια άλλη αίτηση για ανάκτηση του λογαριασμού και στη συνέχεια,

καλείται η συνάρτηση `send()` της κλάσης `SendEmail` που αναλαμβάνει να στείλει το email με το σύνδεσμο ανάκτησης στο χρήστη.

```
@GET
@Path("/recover/{username}/{newpID}")
@Produces({"application/json"})
public Integer recover(@PathParam("username") String username,
@PathParam("newpID") String newpID) throws Exception {
    Integer result = 0;
    PCAusers user = (PCAusers) em.find(PCAusers.class, username);

    if (user == null) {
        result = 105; // username not exist
        return result;
    }

    String messageURL =
"http://83.212.114.35:8080/PCA_RestWS/webresources/gr.hua.webdev.pca.entities
.pcausers/recoveryResults/" + username;

    if (user.getNewParentID() != null) {
        result = 107; // account recovery already in progress
        return result;
    }

    List<PCAusers> users = super.findAll();

    for (PCAusers u:users) {
        if (u.getNewParentID() != null) {
            if (u.getParentID().contentEquals(newpID) ||
u.getNewParentID().contentEquals(newpID)) {
                result = 108; // appID already in use
                return result;
            }
        }
    }

    int temp = em.createQuery("UPDATE PCAusers SET newParentID = '" +
newpID + "' WHERE username = '" + username + "'").executeUpdate();

    String recipient = user.getEmail();
    result = SendEmail.send(messageURL, user);

    return result;
}
```

Η συνάρτηση ενεργοποίησης του λογαριασμού που παρέχεται ως σύνδεσμος στο email του χρήστη ονομάζεται `recoveryResults`, λαμβάνει το όνομα χρήστη με `HTTP_GET` και επιστρέφει αλφαριθμητικά δεδομένα που είναι το αποτέλεσμα της ενεργοποίησης που εμφανίζεται στο χρήστη σε μορφή `html`. Μόλις ενεργοποιηθεί η συγκεκριμένη συνάρτηση από το email του χρήστη ενημερώνεται η βάση δεδομένων για το νέο αναγνωριστικό της συσκευής του χρήστη.

```
@GET
@Path("/recoveryResults/{username}")
@Produces(MediaType.TEXT_HTML)
public String recoveryResults(@PathParam("username") String username)
throws Exception {
    int result = 0;
    PCAusers user = (PCAusers) em.find(PCAusers.class, username);

    if (user == null) {
        result = 105; // username not exist
        return htmlMessage(result, "none");
    }

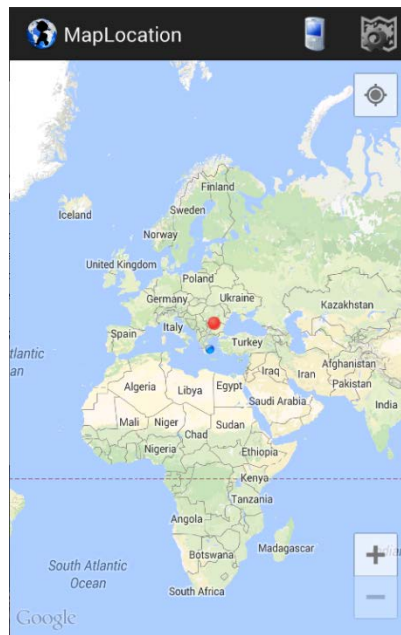
    if (user.getNewParentID() == null) {
        result = 109; //
        return htmlMessage(result, username);
    }

    int temp = em.createQuery("UPDATE PCAdevices SET parentID = '" +
user.getNewParentID() + "' WHERE parentID = '" + user.getParentID() +
"'").executeUpdate();
    temp = em.createQuery("UPDATE PCAusers SET parentID = '" +
user.getNewParentID() + "', newParentID = NULL WHERE username = '" + username
+ "'").executeUpdate();

    return htmlMessage(result, username);
}
```

4.4.4 Κεντρική οθόνη εφαρμογής

Μετά την επιτυχή εισαγωγή των πληροφοριών εισόδου, ο χρήστης εισέρχεται στην κεντρική οθόνη της εφαρμογής. Στη συγκεκριμένη καρτέλα δίνεται η δυνατότητα στο γονέα να παρακολουθήσει την τελευταία τοποθεσία των παιδιών του και άλλες πληροφορίες που σχετίζονται με αυτή. Αυτό καθίσταται δυνατό με τη χρήση της εφαρμογής Doodlz σε smartphone, η λειτουργία της οποίας θα αναφερθεί στο κεφάλαιο 5. Οι θέσεις των παιδιών παρουσιάζονται στο χάρτη ως πινέζες, ενώ παρουσιάζεται και η τρέχουσα θέση του γονέα ώστε να βλέπει και γραφικά την απόστασή του από τα παιδιά του.

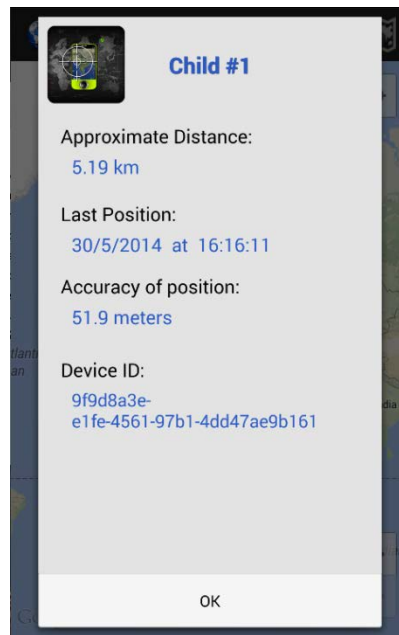


Εικόνα 4.11 Αρχική σελίδα MapLocation

Η εφαρμογή Doodlz στέλνει την τοποθεσία της συσκευής στο διακομιστή, ο οποίος με τη σειρά του αποθηκεύει κρυπτογραφημένα τα στοιχεία της θέσης στον πίνακα PCA_devices της βάσης δεδομένων. Με κατάλληλη επιλογή, από αυτό τα στοιχεία γίνεται δυνατή η αποκρυπτογράφηση και η απεικόνιση των θέσεων στο χάρτη.

Για να εμφανιστούν από τις εγγραφές που υπάρχουν στον πίνακα της βάσης δεδομένων μόνο αυτές που ενδιαφέρουν το γονέα, γίνεται επιλογή με βάση το πεδίο parentID της κάθε εγγραφής. Αν επιλεγούν τα σημεία που εμφανίζονται στο χάρτη, ενεργοποιείται ένα παράθυρο διαλόγου με πληροφορίες όπως η ημερομηνία και ώρα που ήταν η συσκευή σε εκείνο το σημείο, καθώς και η ακρίβεια της μέτρησης. Επίσης, υπολογίζεται η απόσταση της συσκευής από τη συσκευή του γονέα χρησιμοποιώντας τις συντεταγμένες που δίνονται από το GPS.

Τέλος, στην κεντρική οθόνη της εφαρμογής παρέχονται και δύο μενού επιλογών για το χρήστη. Το ένα σχετίζεται με το λογαριασμό και τις συσκευές που μπορεί να διαχειριστεί, ενώ το άλλο με τις ρυθμίσεις εμφάνισης του χάρτη (κανονικός, δορυφορικός, υβριδικός).



Εικόνα 4.12 Πληροφορίες θέσης συσκευής

4.4.4.1 Σημαντικά σημεία κώδικα

Για τη δημιουργία των χαρτών χρησιμοποιήθηκε το Google Maps Android API v2. Χρησιμοποιώντας το Gmail λογαριασμό συνδεθήκαμε στο Google Developers Console και από εκεί πήραμε ένα κλειδί που μπορεί να χρησιμοποιηθεί μόνο στη συγκεκριμένη εφαρμογή και ορίστηκε στο AndroidManifest.xml αρχείο της εφαρμογής.

```
<meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyCT8nueZg7Ly3Io_X9ewN8s8ix51bzpmg0" />
```

Σημαντικό ρόλο στην επιτυχή παρουσίαση των χαρτών στην android εφαρμογή διαδραματίζει και η χρήση μίας βιβλιοθήκης που εισάγεται συμπληρωματικά στην εφαρμογή, της google_play_services_lib.

Ολοκληρώνοντας τις παραπάνω ενέργειες, για να απεικονιστεί ο χάρτης στην εφαρμογή χρησιμοποιείται διάταξη μονάδας (fragment layout), ενώ μέσω της Google Service βιβλιοθήκης “φορτώνεται” ο χάρτης στην εφαρμογή και με τη χρήση των πινεζών (markers) μπορεί να μαρκαριστεί το σημείο στο χάρτη.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_map_location);

    FragmentManager fragmentManager = getFragmentManager();
    MapFragment mapFragment = (MapFragment)
fragmentManager.findFragmentById(R.id.map);
    googleMap = mapFragment.getMap();

    if (googleMap != null) {

        // Get hint from SharedPreferences
        SharedPreferences idSharedPreferences =
getSharedPreferences(LoginActivity.IDPREFS, Activity.MODE_PRIVATE);
        hint = idSharedPreferences.getString("HTEXT", "");

        AESCBC d = new AESCBC();
        try {
            d.getKey(hint);
        } catch (Exception e) {
            e.printStackTrace();
        }

        // connect to Webservice "getParentChildDevices"
        new
DevicesAsyncTask().execute("http://83.212.114.35:8080/PCA_RestWS/webresources
/gr.hua.webdev.pca.entities.pcadevices/findByParentID/" + appID);

        googleMap.getUiSettings().setAllGesturesEnabled(true);
        googleMap.setMyLocationEnabled(true);
        float cameraZoom = 15;

        if (savedInstanceState != null){
            mapType = savedInstanceState.getInt("map_type",
GoogleMap.MAP_TYPE_NORMAL);

            double savedLat = savedInstanceState.getDouble("lat");
            double savedLng = savedInstanceState.getDouble("lng");

            cameraZoom = savedInstanceState.getFloat("zoom", 18);
        }

        googleMap.setMapType(mapType);

        googleMap.setOnMapLongClickListener(new
OnMapLongClickListener(){});

        googleMap.setOnInfoWindowClickListener(listener);
    }
    else {
        int checkGooglePlayServices = GooglePlayServicesUtil
            .isGooglePlayServicesAvailable(this);
        if (checkGooglePlayServices != ConnectionResult.SUCCESS) {
            // google play services is missing!!!!
            /*

```

Για την εύρεση των συσκευών το Web Service λαμβάνει από την εφαρμογή το αναγνωριστικό της συσκευής και αναλαμβάνει να ελέγξει στον πίνακα PCA_device της βάσης δεδομένων για εγγραφές που σχετίζονται με αυτό. Τελικά, επιστρέφεται στην εφαρμογή ένας πίνακας με όλες τις εγγραφές που βρέθηκαν.

Μόλις παραλάβει τον πίνακα από το Web Service, η εφαρμογή αναλαμβάνει να αποκρυπτογραφήσει την τοποθεσία της κάθε συσκευής χρησιμοποιώντας τη φράση ασφαλείας που έχει ορίσει εξ αρχής ο χρήστης, να υπολογίσει την απόστασή της από το χρήστη και να την παρουσιάσει μαζί με τα υπόλοιπα στοιχεία που την αποτελούν ως marker στο χάρτη.

```
private void displayDevices() {
    Date positionTime;
    int counter = 0;
    Marker marker;
    markers = new ArrayList<Marker>();

    myLocation = getLocation();
    if (myLocation == null) {
        AlertDialog.Builder builder = new
AlertDialog.Builder(MapLocationActivity.this);
        builder.setIcon(R.drawable.alert_info);
        builder.setMessage("Your GPS is disabled! Would you like to
enable it?")
            .setCancelable(false)
            .setPositiveButton("Enable GPS", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    Intent gpsOptionsIntent = new
Intent(android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS);
                    startActivity(gpsOptionsIntent);
                }
            });
        builder.setNegativeButton("Do nothing", new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    dialog.cancel();
                }
            });
        AlertDialog alert = builder.create();
        alert.show();
    }

    if (devices.size() == 0 && emptyArray == true) {
        Toast.makeText(this, "There are no child devices attached to your
account!", Toast.LENGTH_LONG).show();
    }

    for (int i=0; i<devices.size(); i++) {
        positionTime = new Date(new SimpleDateFormat("MM/dd/yyyy
k:mm:ss").format(new Date(new Long(devices.get(i).getPositionTime()))));

        counter++;
        LatLng dLatLng = new LatLng(devices.get(i).getmLatitude(),
devices.get(i).getmLongitude());
        googleMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        marker = googleMap.addMarker(new MarkerOptions()
            .position(dLatLng)
            .title("Child #" + counter)
            .snippet("Last position: " + positionTime.getDate() +
"/" + (positionTime.getMonth()+1) + "/" + (positionTime.getYear()+1900) + " "
+ positionTime.getHours() + ":" + positionTime.getMinutes() + ":" +
positionTime.getSeconds()))
            .icon(BitmapDescriptorFactory.fromResource(R.drawable.pin))
            .flat(true));
    }
}
```

Για να εμφανιστούν επιπλέον πληροφορίες σχετικά με τη συσκευή και τη θέση της, ο χρήστης πρέπει να πατήσει πάνω στο marker και θα του εμφανιστεί ένα παράθυρο διαλόγου, όπως παρουσιάστηκε στην εικόνα 3.12. Ο παρακάτω κώδικας περιγράφει πώς πραγματοποιήθηκε αυτή η διαδικασία.

```
private OnInfoWindowClickListener listener = new OnInfoWindowClickListener()
{
    @Override
    public void onInfoWindowClick(Marker marker) {
        PCA_device marker_device = allMarkersMap.get(marker);

        Date positionTime = new Date(new SimpleDateFormat("MM/dd/yyyy
k:mm:ss").format(new Date(new Long(marker_device.getPositionTime()))));

        Location markerLocation = new Location("provider");
        markerLocation.setLatitude(marker_device.getmLatitude());
        markerLocation.setLongitude(marker_device.getmLongitude());

        float distance = markerLocation.distanceTo(myLocation);

        AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(context);

        // get link_code.xml view
        LayoutInflater li = LayoutInflater.from(context);
        View infoView = li.inflate(R.layout.marker_info, null);

        TextView childnoTV = (TextView)
infoView.findViewById(R.id.childNo);
        TextView distanceTV = (TextView)
infoView.findViewById(R.id.distance);
        TextView positiontimeTV = (TextView)
infoView.findViewById(R.id.positiontime);
        TextView maccuracyTV = (TextView)
infoView.findViewById(R.id.maccuracy);
        TextView childidTV = (TextView)
infoView.findViewById(R.id.childid);

        childnoTV.setText(marker.getTitle());
        if (distance/1000 > 0)
            distanceTV.setText(String.format("%.2f km" ,
distance*0.001));
        else
            distanceTV.setText(distance + " meters");

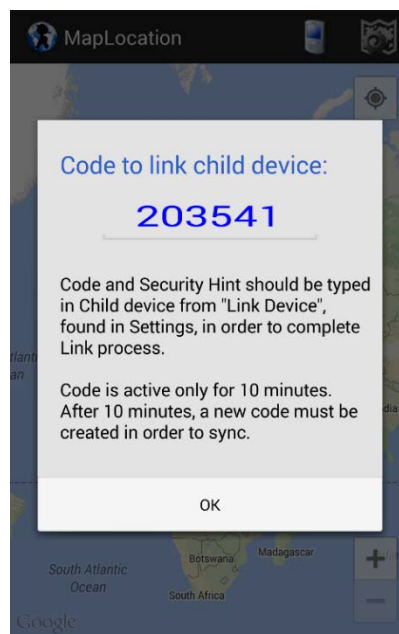
        positiontimeTV.setText(positionTime.getDate() + "/" +
(positionTime.getMonth()+1) + "/" + (positionTime.getYear()+1900) + " at "
+ positionTime.getHours() + ":" + positionTime.getMinutes() + ":" +
positionTime.getSeconds());
        maccuracyTV.setText(String.format("%.1f meters",
marker_device.getmAccuracy()));
        childidTV.setText(marker_device.getChildID());

        // set link_code.xml to alertdialog builder
        alertDialogBuilder.setView(infoView);

        // set dialog message
        alertDialogBuilder
            .setIcon(R.drawable.alert_info)
            .setCancelable(false)
            .setPositiveButton("OK", null);
    }
}
```

4.4.5 Προσθήκη εξαρτώμενης συσκευής

Επιλέγοντας την καρτέλα “Add Child Device” από το αντίστοιχο μενού της συσκευής που βρίσκεται στην κεντρική οθόνη της εφαρμογής MapLocation, ο χρήστης έχει τη δυνατότητα να προσθέσει συσκευές που χρησιμοποιούν την εφαρμογή Doodlz. Για να ολοκληρωθεί η διαδικασία απαιτείται επιβεβαίωση και από την άλλη συσκευή. Γενικά, ακολουθήθηκε το πρότυπο συγχρονισμού που χρησιμοποιεί η Mozilla για το συγχρονισμό των προγραμμάτων περιήγησης Firefox των διαφόρων συσκευών ενός χρήστη. Με άλλα λόγια, στη συσκευή του γονέα δημιουργείται ένας τυχαίος εξαψήφιος κωδικός, ο οποίος αποθηκεύεται στη βάση δεδομένων για συγκεκριμένο χρόνο και μέσα σε αυτό το διάστημα ο χρήστης πρέπει να συμπληρώσει τον ίδιο κωδικό στη συσκευή του παιδιού του για να γίνει ο συγχρονισμός. Ο κωδικός είναι σε ισχύ για δέκα λεπτά από τη δημιουργία του και χρησιμοποιείται μόνο για μία συσκευή.



Εικόνα 4.13 Προσθήκη συσκευής

4.4.5.1 Σημαντικά σημεία κώδικα

Επιλέγοντας ο χρήστης την προσθήκη συσκευής ενεργοποιείται μία συνάρτηση, η οποία χρησιμοποιώντας τη γεννήτρια τυχαίων αριθμών Random δημιουργεί έναν εξαψήφιο αριθμό τον οποίο και παρουσιάζει στο χρήστη μέσα από ένα αναδυόμενο παράθυρο που ανοίγει. Μόλις πατηθεί το κουμπί OK, καλείται το Web Service του διακομιστή το οποίο λαμβάνει

ως παραμέτρους το αναγνωριστικό της συσκευής και τον κωδικό και τα προσθέτει στη βάση ενημερώνοντας το χρήστη αν υπήρξε κάποιο πρόβλημα.

Το παρακάτω απόσπασμα κώδικα περιγράφει τη διαδικασία της προσθήκης νέας συσκευής που πραγματοποιείται από την εφαρμογή.

```
// ----- LINK DEVICE -----

// create random code for link device
public void createCode() {

    Time t = new Time();
    t.setToNow();
    Random mRandom = new Random(t.toMillis(false));

    code = mRandom.nextInt(999999);

    // Display Link Code process
    AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(context);

    // get hint_link_code.xml view
    LayoutInflater li = LayoutInflater.from(context);
    View promptsView = li.inflate(R.layout.link_code, null);

    EditText dialogUserInput =
(EditText)promptsView.findViewById(R.id.dialogUserInput);
    dialogUserInput.setText(String.valueOf(code));

    // set link_code.xml to alertDialog builder
    alertDialogBuilder.setView(promptsView);

    // set dialog message
    alertDialogBuilder
        .setIcon(R.drawable.alert_info)
        .setCancelable(false)
        .setPositiveButton("OK", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,int id) {
                // connect to Webservice "linkDeviceParent"
                new
PostAsyncTask().execute("http://83.212.114.35:8080/PCA_RestWS/webresources/gr
.hua.webdev.pca.entities.pcalinkcodes/linkDeviceParent");
            }
        });

    // create alert dialog
    AlertDialog alertDialog = alertDialogBuilder.create();

    // show it
    alertDialog.show();
}
```

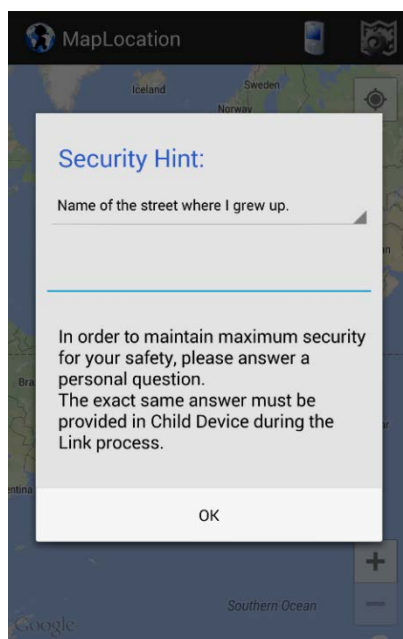
4.4.6 Επεξεργασία κωδικού ασφαλείας

Επιλέγοντας την καρτέλα “Edit Security Hint” από το αντίστοιχο μενού της συσκευής που βρίσκεται στην κεντρική οθόνη της εφαρμογής MapLocation, ο χρήστης έχει τη δυνατότητα

να ορίσει νέα φράση ασφαλείας για την αποκρυπτογράφηση των δεδομένων που λαμβάνει από τις συσκευές που χρησιμοποιούν την εφαρμογή Doodlz.

Η φράση ασφαλείας κρυπτογραφείται και αποθηκεύεται τοπικά στη συσκευή. Για τη διευκόλυνση της μνημόνευσής της από το χρήστη, αποτελεί ουσιαστικά την απάντηση σε κάποια από τις διαθέσιμες ερωτήσεις που του παρέχονται. Για να προασπιστεί η μέγιστη ασφάλεια και να μην μπορεί κάποιος τρίτος πέρα του ίδιου του γονέα να έχει πρόσβαση στη φράση, η εφαρμογή ζητάει τον κωδικό που έχει οριστεί για την είσοδό του στην εφαρμογή για να τον ταυτοποιήσει πριν εμφανιστεί το αναδυόμενο παράθυρο της επεξεργασίας.

Στην περίπτωση αλλαγής της φράσης, η νέα κρυπτογραφείται και αντικαθιστά την προηγούμενη στο φάκελο της εφαρμογής που υπάρχει στην android συσκευή. Η νέα φράση ασφαλείας πρέπει να οριστεί και στις εξαρτώμενες συσκευές που χρησιμοποιούν τη Doodlz εφαρμογή προκειμένου να είναι δυνατή η κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που ανταλλάσσονται.



Εικόνα 4.14 Επεξεργασία φράσης ασφαλείας

4.4.6.1 Σημαντικά σημεία κώδικα

Επιλέγοντας την καρτέλα “Edit Security Hint” ενεργοποιείται μία συνάρτηση, όπως περιγράφει το παρακάτω απόσπασμα κώδικα, η οποία εμφανίζει ένα αναδυόμενο παράθυρο που ζητάει τον κωδικό εισόδου του χρήστη ώστε να τον ταυτοποιήσει. Μόλις ο χρήστης συμπληρώσει τον κωδικό του, καλείται ένα Web Service από το διακομιστή που ελέγχει εάν

έχει δοθεί σωστός κωδικός για το συγκεκριμένο χρήστη και επιστρέφει το αποτέλεσμα στην εφαρμογή. Στη συνέχεια, η εφαρμογή εμφανίζει μήνυμα λάθους αν δεν ολοκληρώθηκε με επιτυχία η ταυτοποίηση ή ενεργοποιεί την κεντρική οθόνη επεξεργασίας της φράσης ασφαλείας.

```
// ----- HINT EDITOR -----

private void editHint() {
    AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(context);

    // get hint_verify_pass.xml view
    LayoutInflater li = LayoutInflater.from(context);
    View promptsView = li.inflate(R.layout.hint_verify_pass, null);

    final EditText passwordEditText =
(EditText)promptsView.findViewById(R.id.edit_hint_password);

    alertDialogBuilder.setView(promptsView);

    alertDialogBuilder
        .setIcon(R.drawable.alert_info)
        .setCancelable(true)
        .setPositiveButton("Verify", new
DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {
                if (passwordEditText.getText().length() > 0) {
                    password = passwordEditText.getText().toString();

                    // connect to Webservice "hintLogin"
                    new
HintAsyncTask().execute("http://83.212.114.35:8080/PCA_RestWS/webresources/gr
.hua.webdev.pca.entities.pcausers/hintLogin");

                }
                else // display message with the error
                {
                    // create a new AlertDialog Builder
                    AlertDialog.Builder builder = new
AlertDialog.Builder(MapLocationActivity.this);

                    builder.setTitle(R.string.errorTitle);
                    builder.setIcon(R.drawable.alert_error);

                    // provide an OK button that simply dismisses the
dialog

                    builder.setPositiveButton(R.string.OK, null);

                    // set the message to display
                    builder.setMessage(R.string.loginErrorMessage);

                    AlertDialog errorDialog = builder.create();
                    errorDialog.show(); // display the Dialog
                }
            }
        });

    AlertDialog dialog = alertDialogBuilder.create();
    dialog.show();
}
```

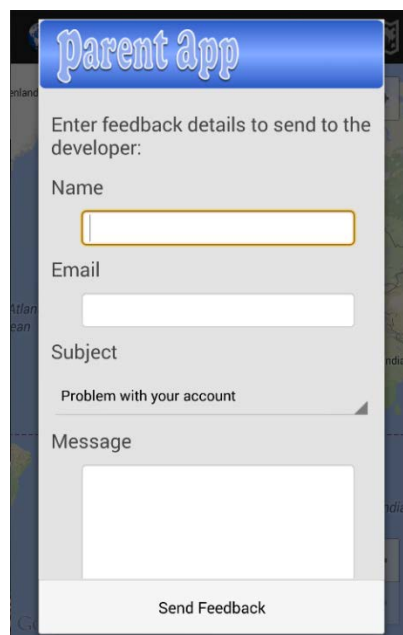
Συμπληρώνοντας ο χρήστης τη φράση που επιθυμεί και επιβεβαιώνοντας την επιλογή του, καλείται η συνάρτηση `displayHint()` που ελέγχει αν είναι αποδεκτά τα στοιχεία που δόθηκαν και στη συνέχεια, χρησιμοποιώντας τη συνάρτηση σύνοψης SHA-256 δημιουργεί το ψηφιακό αποτύπωμα της φράσης ασφαλείας και την αποθηκεύει τοπικά στη συσκευή μέσω της android κλάσης `SharedPreferences`.

```
private void displayHint() {  
    AlertDialog.Builder alertDialogBuilder = new  
AlertDialog.Builder(context);  
  
    // get hint_link_code.xml view  
    LayoutInflater li = LayoutInflater.from(context);  
    View promptsView = li.inflate(R.layout.hint_link_code, null);  
  
    final Spinner hintSpinner =  
(Spinner)promptsView.findViewById(R.id.securitySpinner);  
    ArrayAdapter<CharSequence> adapter =  
ArrayAdapter.createFromResource(  
        this, R.array.content_names,  
        R.layout.spinner_selector);  
  
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item  
);  
  
    hintSpinner.setAdapter(adapter);  
    hintSpinner.setOnItemSelectedListener(new OnItemSelectedListener()  
{});  
  
    final EditText dialogUserInput =  
(EditText)promptsView.findViewById(R.id.hintInput);  
  
    alertDialogBuilder.setView(promptsView);  
  
    alertDialogBuilder  
        .setIcon(R.drawable.alert_info)  
        .setCancelable(true)  
        .setPositiveButton("OK", new DialogInterface.OnClickListener() {  
  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                String hintText = dialogUserInput.getText().toString();  
  
                if (hintText.equals("")) {}  
                else {  
                    // Store hintText in SharedPreferences  
                    SharedPreferences idSharedPreferences =  
getSharedPreferences(LoginActivity.IDPREFS, Activity.MODE_PRIVATE);  
                    SharedPreferences.Editor editor =  
idSharedPreferences.edit();  
  
                    try {  
                        String encryptHint =  
SHAencrypt.SHA256encrypt(hintText);  
                        editor.putString("HTEXT", encryptHint);  
                        editor.commit();  
                    } catch (Exception e) {  
                        e.printStackTrace();  
                    }  
  
                    AlertDialog.Builder builder2 = new  
AlertDialog.Builder(MapLocationActivity.this);  

```

4.4.7 Αποστολή σχολίων

Επιλέγοντας την καρτέλα “Send Feedback” από το αντίστοιχο μενού της συσκευής που βρίσκεται στην κεντρική οθόνη της εφαρμογής MapLocation, ο χρήστης έχει τη δυνατότητα να επικοινωνήσει με το διαχειριστή του συστήματος για οποιοδήποτε σχόλιο ή απορία. Με τη συγκεκριμένη επιλογή ανοίγει ένα αναδυόμενο παράθυρο με μία φόρμα επικοινωνίας, όπως παρουσιάζεται στην εικόνα παρακάτω. Αφού συμπληρώσει ο χρήστης όλα τα στοιχεία και πατήσει το κουμπί “Send Feedback”, στέλνεται ένα email στο διαχειριστή του συστήματος με όλες τις πληροφορίες που έχει δηλώσει ο χρήστης.



Εικόνα 4.15 Αποστολή σχολίων

4.4.7.1 Σημαντικά σημεία κώδικα

Επιλέγοντας την καρτέλα “Send Feedback” καλείται μία συνάρτηση, η οποία εμφανίζει τη φόρμα επικοινωνίας. Με τη συμπλήρωση της φόρμας, ελέγχονται τα στοιχεία που έχουν δηλωθεί και στη συνέχεια καλείται ένα Web Service από το διακομιστή, το οποίο λαμβάνει ως παραμέτρους τις πληροφορίες της φόρμας (FeedbackCredentials) και αναλαμβάνει να στείλει το email στο διαχειριστή.

Τα στοιχεία που πρέπει να δώσει ο χρήστης στη φόρμα επικοινωνίας είναι το όνομά του, το email επικοινωνίας, το θέμα που θα έχει το συγκεκριμένο email και το μήνυμά του.

```
// ----- FEEDBACK -----

private void displayFeedback() {

    AlertDialog.Builder alertDialogBuilder = new
AlertDialog.Builder(context);

    // get feedback.xml view
    LayoutInflater li = LayoutInflater.from(context);
    View promptsView = li.inflate(R.layout.feedback, null);

    final Spinner subjectSpinner =
(Spinner)promptsView.findViewById(R.id.feedbackSpinner);
    ArrayAdapter<CharSequence> adapter =
ArrayAdapter.createFromResource(
        this, R.array.feedback_subjects,
R.layout.spinner_selector);

    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item
);

    subjectSpinner.setAdapter(adapter);
    subjectSpinner.setOnItemSelectedListener(new
OnItemSelectedListener() {

        @Override
        public void onItemSelected(AdapterView<?> parent, View view,
            int position, long id) {
            Object subjectInput = parent.getItemAtPosition(position);
            subject = subjectInput.toString();
        }

        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }

    });

    alertDialogBuilder.setView(promptsView);

    nameInput = (EditText)promptsView.findViewById(R.id.feedName);
    emailInput = (EditText)promptsView.findViewById(R.id.feedEmail);
    messageInput = (EditText)promptsView.findViewById(R.id.feedMessage);

    alertDialogBuilder
        .setCancelable(true)
        .setPositiveButton("Send Feedback", feedbackButtonListener);

    // create alert dialog
    AlertDialog alertDialog = alertDialogBuilder.create();

    // show it
    alertDialog.show();
}
```

4.4.8 Απενεργοποίηση λογαριασμού

Ο χρήστης έχει τη δυνατότητα να απενεργοποιήσει το λογαριασμό του στην περίπτωση που δεν ενδιαφέρεται να χρησιμοποιήσει ξανά τη εφαρμογή. Μέσα από αυτή τη διαδικασία μπορεί να είναι σίγουρος ότι δεν υπάρχουν δικά του στοιχεία ή των παιδιών του στη βάση δεδομένων. Η απενεργοποίηση του λογαριασμού πραγματοποιείται μετά από μία εργάσιμη

ημέρα από την αίτηση του χρήστη και σβήνει όλα τα δεδομένα του χρήστη και των συσκευών που εξαρτώνται από αυτόν.

4.4.8.1 Σημαντικά σημεία κώδικα

Με την επιλογή του χρήστη να απενεργοποιήσει το λογαριασμό του, καλείται η συνάρτηση `unlinkDevice` του Web Service του διακομιστή που παίρνει ως παράμετρο το αναγνωριστικό της συσκευής του χρήστη και αναλαμβάνει να ενημερώσει τις εγγραφές της βάσης δεδομένων θέτοντας στο πεδίο `unlinkDate` του πίνακα `PCA_devices` την τρέχουσα ημερομηνία και ώρα. Αν ο αριθμός των εγγραφών που επιστράφηκαν από τη βάση είναι αρνητικός, ενημερώνεται ο χρήστης ότι υπήρξε κάποιο πρόβλημα κατά τη διαδικασία.

```
@GET
@Path("/unlinkDevice/{parentID}")
@Consumes({"application/json"})
@Produces({"application/json"})
public Integer unlinkDevice(@PathParam("parentID") String uuid) {
    Integer entitiesNo = 0;

    entitiesNo = em.createQuery("UPDATE PCAdevices SET
unlinkDate=CURRENT_TIMESTAMP WHERE parentID = '" + uuid +
"'").executeUpdate();

    if (entitiesNo < 0)
        entitiesNo = 301;

    return entitiesNo;
}
```

Στη βάση δεδομένων έχει δημιουργηθεί ένα συμβάν (event), το οποίο επαναλαμβάνεται κάθε ένα λεπτό και ελέγχει ποιες εγγραφές έχουν ζητήσει να απενεργοποιηθεί ο λογαριασμός τους μία μέρα πριν. Το συγκεκριμένο event είναι γραμμένο σε sql κώδικα και στοχεύει να σβήσει αρχικά τους χρήστες που έχουν ζητήσει να απενεργοποιηθεί ο λογαριασμός τους και στη συνέχεια τις συσκευές που εξαρτώνται από αυτούς.

```
BEGIN

DELETE FROM `PCA_users`
WHERE `parentID` =
    (SELECT `parentID` FROM `PCA_devices`
    WHERE `unlinkDate` < (NOW() - INTERVAL 1 DAY) AND `parentID` IS NOT NULL
);

DELETE FROM `PCA_devices`
WHERE `unlinkDate` < (NOW() - INTERVAL 1 DAY);

END
```

4.5 Στοιχεία κώδικα εφαρμογής

Για την εφαρμογή MapLocation δημιουργήθηκαν 9 αρχεία java με συνολικό μέγεθος 93 KB και συνολικές γραμμές κώδικα 2364. Επίσης, χρησιμοποιήθηκαν 9 εικόνες με συνολικό μέγεθος 187 KB. Τέλος, 21 αρχεία xml μεγέθους 39,3 KB με γραμμές κώδικα 961. Από αυτά τα xml αρχεία, τα 11 δημιουργήθηκαν για την εμφάνιση των διεπαφών του χρήστη (φάκελος layout), τα 2 καθορίζουν τα μενού που εμφανίζονται στην εφαρμογή (φάκελος menu), τα 7 χρησιμοποιούνται για να δηλωθούν οι πόροι που χρησιμοποιούνται από την εφαρμογή (φάκελος values) και το ένα που μένει είναι το AndroidManifest.xml που καθορίζει το ρόλο όλων των αρχείων.

4.6 Στοιχεία Web Service

Όπως αναφέρθηκε και παραπάνω, για την επικοινωνία της εφαρμογής MapLocation με τη βάση δεδομένων χρησιμοποιήθηκε ένα Restful Web Service. Δημιουργήθηκε στο περιβάλλον εργασίας Netbeans IDE και αναπτύχθηκε πάνω στον εξυπηρέτη Glassfish του διακομιστή.

Δημιουργήθηκαν 16 αρχεία java με συνολικό μέγεθος 59,2 KB και συνολικές γραμμές κώδικα 1724. Από αυτά τα αρχεία, τα 11 καθορίζουν τις οντότητες που χρησιμοποιήθηκαν, ενώ τα υπόλοιπα 5 αποτελούν το service με το οποίο επικοινωνεί η εφαρμογή και αναλαμβάνει να επικοινωνήσει με τη βάση.

4.7 Ασφάλεια

Η ασφάλεια διαδραματίζει πολύ σημαντικό ρόλο για τη λειτουργία της εφαρμογής. Πρέπει να προφυλάσσεται από επιθέσεις που έχουν σκοπό να υποκλέψουν ή να αλλοιώσουν στοιχεία χρηστών, δεδομένα, καθώς και να προσπαθήσουν να μουν στο σύστημα ως χρήστες χωρίς να είναι εγγεγραμμένοι.

Πρώτο βήμα για την ασφάλεια είναι η κωδικοποίηση όλων των κωδικών που χρησιμοποιούνται από τους χρήστες πριν την αποστολή τους στη βάση δεδομένων, έτσι ώστε

αν μπορέσει κάποιος να “ψαρέψει” έναν κωδικό με τη μέθοδο του phishing να μην μπορεί να τον αποκωδικοποιήσει. Η συνάρτηση που χρησιμοποιείται για την κρυπτογράφηση είναι η SHA-256, η οποία είναι μονόδρομη. Δηλαδή, αν γνωρίζει κάποιος τον αρχικό κωδικό να μπορεί να βρει το SHA-256 αντίστοιχό του. Αν δεν τον γνωρίζει όμως είναι πρακτικά ακατόρθωτο να τον αποκωδικοποιήσει. Για να διασφαλιστεί επιπλέον ασφάλεια του κωδικού του χρήστη και από “επιθέσεις λεξικών” (dictionary attacks) πέρα από τις υπόλοιπες επιθέσεις που προστατεύονται από την ίδια τη συνάρτηση σύνοψης, χρησιμοποιείται μία πληροφορία (salt) η οποία έχει μέγεθος 16 bits και δυαδική μορφή. Η συγκεκριμένη πληροφορία αποθηκεύεται στη βάση κρυπτογραφημένη με το σύστημα Base64 που είναι ένας τρόπος κωδικοποίησης από δυαδικό σε κείμενο. Κατά τη διαδικασία της εγγραφής του χρήστη, δημιουργείται το salt και κωδικοποιείται για να αποθηκευτεί στη συνέχεια στη βάση δεδομένων. Αντίθετα, για την επαλήθευση των στοιχείων του χρήστη λαμβάνεται το salt από τη βάση δεδομένων και αποκρυπτογραφείται με τη βοήθεια του Base64 decoder για να χρησιμοποιηθεί στη δημιουργία του ψηφιακού ίχνους του κωδικού που έδωσε ο χρήστης και να συγκριθεί με τον κρυπτογραφημένο κωδικό που είναι ήδη αποθηκευμένος. Οι τρεις συναρτήσεις που χρησιμοποιούνται για την κρυπτογράφηση παρουσιάζονται στην παρακάτω εικόνα.

```
private static String SHAencrypt (String password, byte[] salt) throws
Exception {
    String encrPass = null;

    MessageDigest md = MessageDigest.getInstance("SHA-256");
    md.update(salt);
    byte[] hash = md.digest(password.getBytes("UTF-8"));

    encrPass = bytesToHex(hash);

    return encrPass;
}

private static byte[] getSalt() throws Exception {
    SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
    byte[] salt = new byte[16];
    sr.nextBytes(salt);

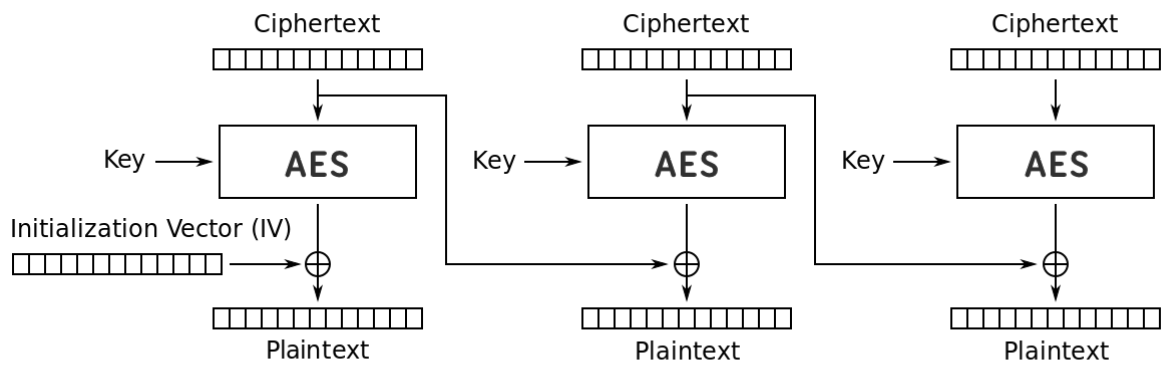
    return salt;
}

public static String bytesToHex(byte[] b) {
    char hexDigit[] = {'0', '1', '2', '3', '4', '5', '6', '7',
                       '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    StringBuffer buf = new StringBuffer();
    for (int j=0; j<b.length; j++) {
        buf.append(hexDigit[(b[j] >> 4) & 0x0f]);
        buf.append(hexDigit[b[j] & 0x0f]);
    }
    return buf.toString();
}
```

Δεύτερο βήμα που πραγματοποιήθηκε για να γίνει πιο ασφαλές το πρόγραμμα είναι η κρυπτογράφηση της φράσης ασφαλείας που ορίζει ο χρήστης για την αποκωδικοποίηση της πληροφορίας που λαμβάνει από τις συσκευές των παιδιών του. Η φράση αποθηκεύεται τοπικά στη συσκευή του γονέα στο φάκελο της εφαρμογής που εμφανίζονται τα SharedPreferences, αλλά έγινε προσπάθεια να είναι σε μορφή μη αναγνωρίσιμη από κάποιο τρίτο. Για αυτό το λόγο δεν είναι δυνατή η αποκρυπτογράφηση της φράσης για την επεξεργασία της, αλλά ζητείται η εισαγωγή νέας φράσης από το χρήστη όταν χρειαστεί. Για την κρυπτογράφηση της φράσης χρησιμοποιήθηκε για άλλη μία φορά η συνάρτηση σύνοψης SHA-256, αλλά χωρίς την επιπλέον πληροφορία salt. Ο κώδικας παρακάτω παρουσιάζει τη διαδικασία κρυπτογράφησης της φράσης μέσα στην android εφαρμογή.

```
public class SHAencrypt {  
    public static String SHA256encrypt(String hint) throws Exception  
    {  
        String hex = "";  
  
        MessageDigest md = MessageDigest.getInstance("SHA-256");  
        md.update(hint.getBytes());  
  
        byte byteData[] = md.digest();  
  
        // convert the byte to hex format  
        StringBuffer hintHex = new StringBuffer();  
        for (int i=0; i<byteData.length; i++) {  
            hex = Integer.toHexString(0xff & byteData[i]);  
  
            if (hex.length() == 1)  
                hintHex.append('0');  
  
            hintHex.append(hex);  
        }  
  
        return hintHex.toString();  
    }  
}
```

Το επόμενο βήμα και το πιο σημαντικό σημείο για την ασφάλεια στην εφαρμογή MapLocation είναι η αποκρυπτογράφηση της θέσης των εξαρτώμενων συσκευών που χρησιμοποιούν τη Doodlz εφαρμογή. Όπως θα αναφερθεί στην ανάλυση της άλλης εφαρμογής που εκπονήθηκε, προκειμένου να προασπιστεί η εμπιστευτικότητα των ευαίσθητων δεδομένων, τα τελευταία κρυπτογραφούνται χρησιμοποιώντας ένα κοινό κρυφό κλειδί που έχει ορίσει ο γονέας (η φράση ασφαλείας που έχει δοθεί εξ αρχής) και μπορούν να αποκρυπτογραφηθούν μόνο από τη συσκευή με το ίδιο μυστικό κλειδί. Έτσι, κανένας τρίτος ακόμα και ο ίδιος ο διαχειριστής δεν έχει πρόσβαση στην πληροφορία που ανταλλάσσεται μεταξύ των δύο εφαρμογών. Για την αποκρυπτογράφηση της θέσης που λαμβάνεται στη συγκεκριμένη εφαρμογή, χρησιμοποιείται το πρότυπο AES (Advanced Encryption Standard), το οποίο αποτελεί έναν από τους αλγόριθμους συμμετρικού κλειδιού (symmetric-key algorithm). Η υλοποίηση του συγκεκριμένου αλγόριθμου στην εφαρμογή περιγράφεται από την παρακάτω εικόνα.



Cipher Block Chaining (CBC) mode decryption

Εικόνα 4.16 Αποκρυπτογράφηση δεδομένων με AES

Για την αποκρυπτογράφηση των δεδομένων των συσκευών που χρησιμοποιούν τη Doodlz εφαρμογή, ενεργοποιείται μία συνάρτηση που αρχικά λαμβάνει την κρυπτογραφημένη θέση μέσω του Web Service και καλεί την κλάση AESCBC. Για την αρχικοποίηση του μυστικού κλειδιού χρησιμοποιείται η φράση ασφαλείας που έχει δηλώσει ο χρήστης, ενώ λαμβάνεται και το IV (Initialization Vector) από τη βάση δεδομένων. Ορίζεται ποιος αλγόριθμος θα χρησιμοποιηθεί και αρχικοποιείται. Τέλος, ξεκινάει η διαδικασία της αποκρυπτογράφησης και επιστρέφεται η πληροφορία στην αρχική συνάρτηση που την κάλεσε. Το παρακάτω κομμάτι κώδικα παρουσιάζει τη λειτουργία του αλγορίθμου AES.

```

public class AESCBC {

    private static byte[] keyValue = null;

    public KeyGenerator keygenerator;
    public SecretKey AESkey;

    public IvParameterSpec ivSpec;

    public String AESdecrypt(String encryptedData, String encryptIV) throws
Exception {
        // Generate the key
        Key key = generateKey();
        // Create the Cipher
        Cipher c = Cipher.getInstance("AES/CBC/PKCS5Padding");

        ivSpec = new IvParameterSpec(Base64.decode(encryptIV,
Base64.DEFAULT));

        // Initialize the cipher for decryption
        c.init(Cipher.DECRYPT_MODE, key, ivSpec);

        byte[] decodedData = Base64.decode(encryptedData, Base64.DEFAULT);
        byte[] decrData = c.doFinal(decodedData);
        String decryptedData = new String(decrData);

        return decryptedData;
    }

    public void getKey(String hintText) throws Exception {

        byte[] b = hintText.getBytes(Charset.forName("UTF-8"));
        keyValue = new byte[b.length/2];

        for (int i=0; i<keyValue.length; i++) {
            keyValue[i] = (byte) (b[i] ^ b[i + keyValue.length]);
        }

    }

    private static Key generateKey() {
        Key key = new SecretKeySpec(keyValue, "AES");
        return key;
    }
}

```

Στο επόμενο πλαίσιο παρουσιάζεται η λειτουργικότητα της συνάρτησης που λαμβάνει τα δεδομένα από το Web Service και τα στέλνει στην κλάση AESCBC για αποκρυπτογράφηση.

```
// convert string to JSONArray
try {
    jsonArray = new JSONArray(bufferString);
    int arrayNum = jsonArray.length();

    for (int i=0; i<arrayNum; i++) {

        JSONObject jsonDevice = jsonArray.getJSONObject(i);

        PCA_device device = new PCA_device();

        String decryptedLoc;
        AESCBC d = new AESCBC();

        device.setChildID(jsonDevice.getString("childID"));
        device.setPositionTime(jsonDevice.getString("positionTime"));

        try {
            decryptedLoc =
d.AESdecrypt(jsonDevice.getString("location"),
jsonDevice.getString("encryptIV"));
            String[] parts = decryptedLoc.split("_");

            device.setmAccuracy(Float.parseFloat(parts[0]));
            device.setmLatitude(Double.parseDouble(parts[1]));
            device.setmLongitude(Double.parseDouble(parts[2]));

            devices.add(device);

        } catch (Exception e) {

            runOnUiThread(new Runnable(){
                @Override
                public void run() {
                    Toast.makeText(MapLocationActivity.this,
                        "Devices of children are found but they
cannot be decrypted. \nPlease verify that you have provided correct Hint to
both devices.",
                        Toast.LENGTH_LONG).show();
                }
            });

            emptyArray = false;

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


Κεφάλαιο 5

Εφαρμογή Doodlz

Η εφαρμογή δημιουργήθηκε με σκοπό να παρέχει στο χρήστη της τη δυνατότητα της ζωγραφικής μέσα από το κινητό του, έχοντας ταυτόχρονα ως στόχο να ενημερώνει το γονέα του για τη θέση της συσκευής. Απαιτείται να υπάρχει σύνδεση στο Διαδίκτυο είτε μέσω ασύρματου δικτύου (WiFi) είτε μέσω δορυφορικού (3G-4G). Για να υπάρχει περισσότερη ακρίβεια στη θέση του χρήστη έχει προτιμηθεί και η ενεργοποίηση του GPS. Δεν υπάρχει υπερβολική χρέωση από τον πάροχο κινητής τηλεφωνίας, καθώς τα δεδομένα που στέλνονται και λαμβάνονται δεν είναι πολλά. Αν και η κατανάλωση της μπαταρίας είναι μεγάλη, φτάνει για τη διάρκεια μιας ημέρας.

Η εφαρμογή της ζωγραφικής που χρησιμοποιήθηκε αποτελεί ένα από τα παραδείγματα που παρέχονται μέσα από το βιβλίο των Deitel Developer Series, *Android for Programmers: An App-Driven Approach*. Η εφαρμογή που παρέχεται στα παραδείγματα ονομάζεται Doodlz, για αυτό και προτιμήθηκε να κρατηθεί ο ίδιος τίτλος.

Σε αυτό το κεφάλαιο θα γίνει η παρουσίαση των σημείων της εφαρμογής που διαμορφώθηκαν ώστε να υποστηρίζεται η εύρεση της θέσης και η αποστολή της στο διακομιστή, και θα αναλυθούν μερικά από τα πιο σημαντικά στάδια της υλοποίησής της.

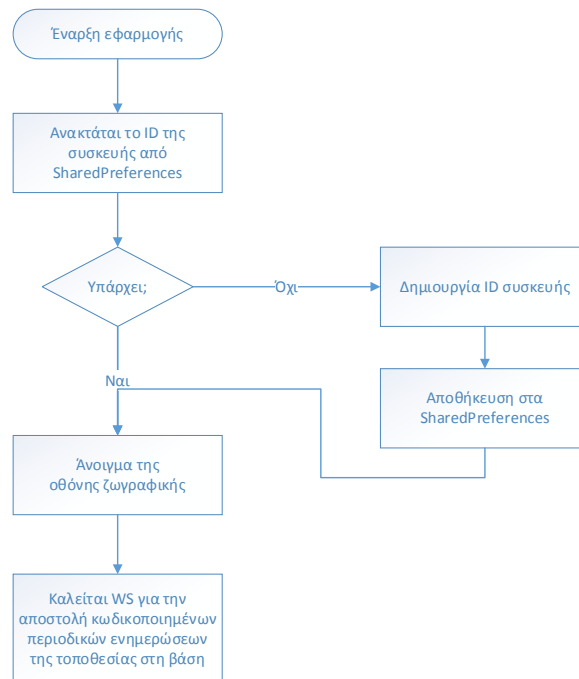
5.1 Διαγράμματα λειτουργίας Doodlz

Οι βασικές λειτουργίες που καλύπτει η εφαρμογή Doodlz αναφέρονται παρακάτω:

- Κεντρική οθόνη της εφαρμογής που σχετίζεται με τη σχεδίαση και το μενού επιλογών
- Προσθήκη συσκευής
- Επεξεργασία κωδικού ασφαλείας

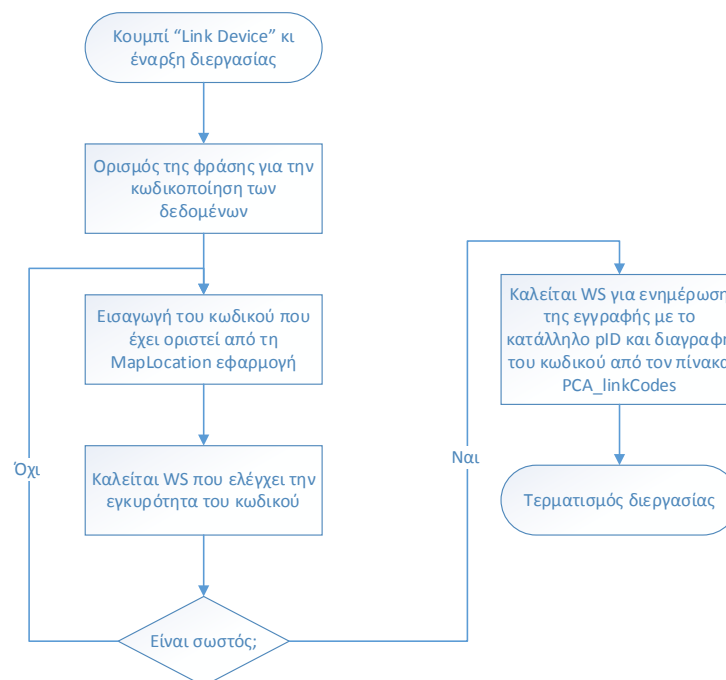
Δημιουργήθηκαν τα παρακάτω διαγράμματα προκειμένου να απεικονιστούν γραφικά οι βασικές λειτουργίες της εφαρμογής και στη συνέχεια θα γίνει μία περιγραφή της κάθε λειτουργίας ξεχωριστά.

5.1.1 Κεντρική οθόνη εφαρμογής



Σχήμα 5.1 Κεντρική οθόνη εφαρμογής

5.1.2 Προσθήκη συσκευής



Σχήμα 5.2 Προσθήκη συσκευής

5.2 Οθόνη ζωγραφικής

Είναι η πρώτη οθόνη που ενεργοποιείται και αποτελεί τον καμβά για τη ζωγραφική. Πέρα από το λευκό πλαίσιο που εμφανίζεται υπάρχουν και δύο μενού επιλογών. Το ένα ενδιαφέρει το κεντρικό χρήστη της εφαρμογής και είναι οι ρυθμίσεις της ζωγραφικής. Αντίθετα, το δεύτερο είναι κρυφό από το χρήστη και χρησιμοποιείται από το γονέα της MapLocation εφαρμογής. Παρουσιάζει τις ρυθμίσεις για τη σωστή αποστολή της κρυπτογραφημένης θέσης στη βάση δεδομένων.



Εικόνα 5.1 Κεντρική οθόνη Doodlz

Μόλις εμφανιστεί η εφαρμογή λαμβάνεται το αναγνωριστικό της συσκευής (appID) και η κωδικοποιημένη φράση ασφαλείας (hint) από τη δεσμευμένη μνήμη της εφαρμογής στη συσκευή (SharedPreferences). Αν το appID είναι κενό σημαίνει πως η εφαρμογή χρησιμοποιείται για πρώτη φορά οπότε με τη συμβολή της συνάρτησης randomUUID δημιουργείται τυχαία το αναγνωριστικό και αποθηκεύεται στα SharedPreferences. Έπειτα, γίνεται ο ίδιος έλεγχος για τη φράση ασφαλείας και στην περίπτωση που είναι κενό, παίρνει την προσωρινή τιμή changeit και αποθηκεύεται με τη σειρά του στα SharedPreferences. Σε αντίθετη περίπτωση καλείται η συνάρτηση getKey της κλάσης AESCBC για να αρχικοποιήσει τον αλγόριθμο συμμετρικού κλειδιού AES που θα χρησιμοποιηθεί για την

κρυπτογράφηση των δεδομένων. Στο τέλος ενεργοποιείται ένα Service, το οποίο αναλαμβάνει να στέλνει περιοδικές ενημερώσεις της θέσης του χρήστη στη βάση δεδομένων.

```
// Get appID & hint from SharedPreferences
SharedPreferences idSharedPreferences = getSharedPreferences(IDPREFS,
Activity.MODE_PRIVATE);
SharedPreferences.Editor editor = idSharedPreferences.edit();

appID = idSharedPreferences.getString("APPID", "");
if (appID.equals("")) {
    appID = UUID.randomUUID().toString();
    editor.putString("APPID", appID);
    editor.commit();
    finish();
}

hint = idSharedPreferences.getString("HTEXT", "");
if (hint == "") {
    editor.putString("HTEXT", "changeit");
    editor.commit();
    finish();
}
else {
    AESCBC d = new AESCBC();
    try {
        d.getKey(hint);
    } catch (Exception e) {
        e.printStackTrace();
    }
}

InitBroadcastReceiver();

Intent serviceIntent = new Intent(this, GPSTrackerService.class);
startService(serviceIntent);
```

Ο πρώτος έλεγχος που γίνεται είναι αν είναι ανοιχτό ή όχι το GPS. Προτείνεται η ενεργοποίηση του GPS για να επιτευχθεί μεγαλύτερη ακρίβεια στη θέση. Στη συνέχεια, ελέγχεται αν η εφαρμογή έχει με κάποιο τρόπο πρόσβαση στο Διαδίκτυο και ειδοποιείται ο χρήστης με μήνυμα σε περίπτωση σφάλματος. Για να ενεργοποιηθεί η λειτουργία πρέπει να υπάρχει ενεργή σύνδεση στο Διαδίκτυο. Όταν ενεργοποιηθεί ξεκινάει να κρυπτογραφεί και να στέλνει τη θέση του παιδιού στο διακομιστή που με τη σειρά του ενημερώνει τον πίνακα PCA_devices κρατώντας μόνο την τελευταία θέση. Ο χρόνος αποστολής είναι καθορισμένος στα 5 λεπτά ή στα 15 μέτρα. Αν δεν ενεργοποιηθεί σύνδεση στο Διαδίκτυο τότε το service δε λειτουργεί. Τα στοιχεία της θέσης που κρυπτογραφούνται και στέλνονται είναι το γεωγραφικό μήκος, το γεωγραφικό πλάτος και η ακρίβεια της μέτρησης. Το service ανά 5 λεπτά κάνει έλεγχο για τη θέση της συσκευής. Στην περίπτωση που μέσα σε αυτό το

διάστημα προηγηθεί αλλαγή της θέσης παραπάνω από 15 μέτρα, τότε το service καλείται να στείλει τη θέση στο διακομιστή.

```

public Location getLocation()
{
    mLM = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    Criteria criteria = new Criteria();
    provider = mLM.getBestProvider(criteria, false);

    mLM.requestLocationUpdates(LocationManager.GPS_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
    location = mLM.getLastKnownLocation(provider);

    if (location != null) {
        onLocationChanged(location);
    }
    return location;
}

@Override
public void onLocationChanged(Location location) {

    longitude = location.getLongitude();
    latitude = location.getLatitude();
    accuracy = location.getAccuracy();
    tracktime = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").format(new
Date());

    System.err.println("Your Location is - \nLat: " + latitude + "\nLong:
" + longitude + "\n\nAcuracy: " + accuracy + "\nDate: " + tracktime);

    if (!isNetworkConnected()) {
        Intent network_intent = new Intent("network_dialog");
        sendBroadcast(network_intent);
    }

    AESCBC aes = new AESCBC();
    try {
        encryptedLoc = aes.AESEncrypt(accuracy + "_" + latitude + "_" +
longitude);
    } catch (Exception e) {
        e.printStackTrace();
    }

    // connect to Webservice "sendLocation"
    new
HttpAsyncTask().execute("http://83.212.114.35:8080/PCA_RestWS/webresources/gr
.hua.webdev.pca.entities.pcadevices/sendLocation");
}

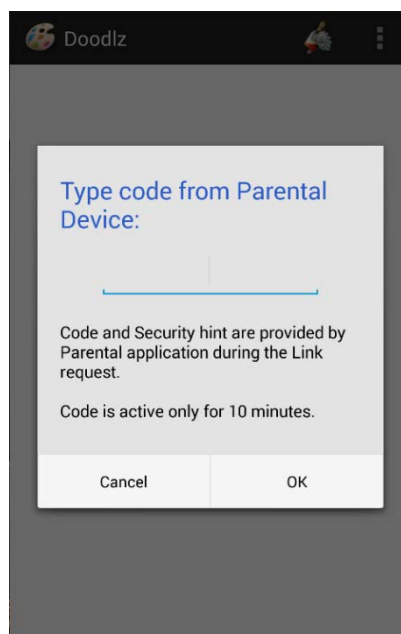
```

Το διάστημα των 5 λεπτών και της απόστασης των 15 μέτρων επιλέχθηκαν μετά από έλεγχο και έρευνα. Τα 5 λεπτά είναι ένα ικανοποιητικό χρονικό διάστημα για το γονέα ώστε να λαμβάνει περιοδικές ενημερώσεις. Τα 15 μέτρα ικανοποιούνται σε περίπτωση που το παιδί βρίσκεται εν κινήσει. Επίσης, διασφαλίζεται ότι η διάρκεια ζωής της μπαταρίας είναι αρκετή για το χρόνο που χρειάζεται η εφαρμογή.

```
// The minimum distance to change Updates in meters
// 15 meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 15;
// The minimum time between updates in milliseconds
// 5 * 60 seconds = 5 minutes
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 5;
```

5.3 Προσθήκη συσκευής

Αποτελεί μία από τις δύο δυνατότητες που έχει ο γονέας στη Doodlz εφαρμογή και βρίσκεται στο μενού Settings της κεντρικής οθόνης. Επιλέγοντας την καρτέλα “Link Device”, ο χρήστης έχει τη δυνατότητα να επιβεβαιώσει τη διαδικασία προσθήκης συσκευής που παρέχεται μέσα από τη MapLocation εφαρμογή. Όπως αναφέρθηκε στην παρουσίαση της MapLocation εφαρμογής, με την κατάλληλη επιλογή δημιουργείται στη συσκευή του γονέα ένας τυχαίος εξαψήφιος κωδικός που αποθηκεύεται στη βάση δεδομένων για ένα συγκεκριμένο χρονικό διάστημα. Μέσα σε αυτό το διάστημα, ο χρήστης έχει την υποχρέωση να επιλέξει από τη Doodlz εφαρμογή τη συγκεκριμένη καρτέλα και να συμπληρώσει τον κωδικό που του δόθηκε στο αντίστοιχο πεδίο.



Εικόνα 5.2 Οθόνη συγχρονισμού Doodlz

Αφού συμπληρώσει τον κωδικό και πατήσει το κουμπί OK, καλείται μία συνάρτηση η οποία επιβεβαιώνει ότι έχει συμπληρωθεί το πεδίο του κωδικού και εμφανίζει μήνυμα σε

περίπτωση λάθους. Στην αντίθετη περίπτωση, ενεργοποιείται το αντίστοιχο Web Service του διακομιστή, το οποίο παίρνει ως παραμέτρους το αναγνωριστικό της συσκευής του παιδιού και τον κωδικό που δόθηκε και αναλαμβάνει να ελέγξει αν υπάρχει ο συγκεκριμένος κωδικός στον πίνακα PCA_linkCodes της βάσης δεδομένων. Αν βρεθεί, λαμβάνει το αναγνωριστικό της συσκευής που τον δημιούργησε και ενημερώνει την εγγραφή του παιδιού με το αντίστοιχο αναγνωριστικό. Στη συνέχεια, διαγράφει το συγκεκριμένο κωδικό από τον πίνακα PCA_linkCodes, καθώς έχει οριστεί να χρησιμοποιείται μόνο για μία εγγραφή. Τέλος, ενημερώνει το χρήστη της εφαρμογής για την πορεία της διαδικασίας με αντίστοιχα μηνύματα.

Το απόσπασμα κώδικα που παρουσιάζεται παρακάτω αποτελεί τη συνάρτηση linkDeviceChild που έχει υλοποιηθεί στο Web Service του διακομιστή.

```
@POST
@Path("/linkDeviceChild")
@Produces({"application/json"})
@Consumes({"application/json"})
public Integer linkDeviceChild(LinkCredentials lcred) {
    Integer result = 0;

    PCALinkCodes link = (PCALinkCodes) em.find(PCALinkCodes.class,
lcred.code);

    if (link == null) {
        result = 201;
        return result;
    }

    String parentID = link.getParentID();

    // Update child row in Devices
    int temp = em.createQuery("UPDATE PCADevices SET parentID='"+
parentID + "', unlinkDate=NULL WHERE childID = '" + lcred.uuid +
"'").executeUpdate();

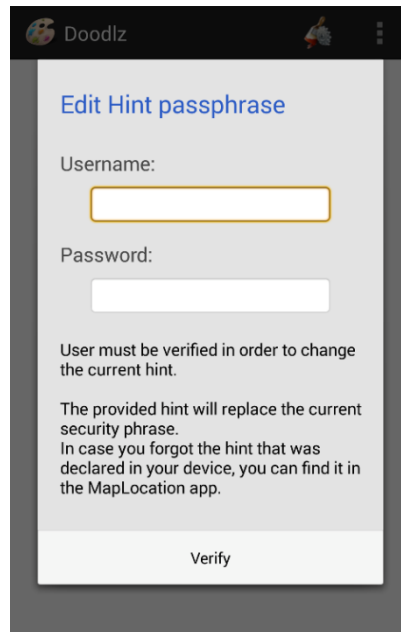
    // Delete row from LinkCodes
    remove(lcred.code);

    return result;
}
```

5.4 Επεξεργασία κωδικού ασφαλείας

Η δεύτερη επιλογή του μενού Settings είναι η καρτέλα “Edit Hint Text”. Σε αυτή, ο γονέας έχει τη δυνατότητα να ορίσει νέα φράση ασφαλείας για την κρυπτογράφηση των δεδομένων που στέλνονται από τη συγκεκριμένη συσκευή και λαμβάνονται από τη δική του.

Η φράση ασφαλείας, όπως και στην εφαρμογή MapLocation, κρυπτογραφείται και αποθηκεύεται τοπικά στη συσκευή. Για να διασφαλιστεί ότι κανένας τρίτος ακόμα και ο βασικός χρήστης της Doodlz εφαρμογής, δεν έχει πρόσβαση στη φράση απαιτείται από το γονέα να συμπληρώσει τα στοιχεία εισόδου που έχει στη MapLocation εφαρμογή.



Εικόνα 5.3 Ταυτοποίηση χρήστη

Για την ταυτοποίηση του χρήστη, καλείται ένα Web Service από το διακομιστή, το οποίο παίρνει ως παραμέτρους τα στοιχεία εισόδου που συμπληρώθηκαν και αναλαμβάνει να ελέγξει στη βάση δεδομένων αν υπάρχει ο συγκεκριμένος χρήστης. Μόλις ταυτοποιηθεί ο χρήστης, εμφανίζεται το αναδυόμενο παράθυρο για την επεξεργασία της φράσης, το οποίο είναι παρόμοιο με το αντίστοιχο της MapLocation εφαρμογής. Σε αντίθετη περίπτωση, εμφανίζεται μήνυμα λάθους που τερματίζει τη διαδικασία.

```
private void editHint() {
    AlertDialog.Builder alertDialogBuilder = new
    AlertDialog.Builder(context);

    // get hint_verify_pass.xml view
    LayoutInflater li = LayoutInflater.from(context);
    View promptsView = li.inflate(R.layout.hint_verify_pass, null);

    final EditText usernameEditText =
    (EditText)promptsView.findViewById(R.id.edit_hint_username);
    final EditText passwordEditText =
    (EditText)promptsView.findViewById(R.id.edit_hint_password);

    alertDialogBuilder.setView(promptsView);

    alertDialogBuilder
        .setIcon(R.drawable.alert_info)
        .setCancelable(true)
        .setPositiveButton("Verify", new
    DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            if (usernameEditText.getText().length() > 0 &&
            passwordEditText.getText().length() > 0) {

                username = usernameEditText.getText().toString();
                password = passwordEditText.getText().toString();

                // connect to Webservice "login"
                new
    PostAsyncTask().execute("http://83.212.114.35:8080/PCA_RestWS/webresources/gr
    .hua.webdev.pca.entities.pcausers/loginChild");

            }
            else // display message with the error
            {}

        }
    });

    AlertDialog dialog = alertDialogBuilder.create();
    dialog.show();
}
```

5.5 Στοιχεία κώδικα εφαρμογής

Για την εφαρμογή Doodlz δημιουργήθηκαν 8 αρχεία java με συνολικό μέγεθος 58 KB και συνολικές γραμμές κώδικα 1681. Επίσης, χρησιμοποιήθηκαν 10 εικόνες με συνολικό μέγεθος 69 KB. Τέλος, ορίστηκαν 16 αρχεία xml μεγέθους 19 KB με γραμμές κώδικα 506. Από αυτά τα xml αρχεία, τα 7 δημιουργήθηκαν για την εμφάνιση των διεπαφών του χρήστη (φάκελος layout), τα 3 καθορίζουν τα μενού που εμφανίζονται στην εφαρμογή (φάκελος menu), τα 5 χρησιμοποιούνται για να δηλωθούν οι πόροι που χρησιμοποιούνται από την εφαρμογή (φάκελος values) και το ένα που μένει είναι το AndroidManifest.xml που καθορίζει το ρόλο όλων των αρχείων.

5.6 Ασφάλεια

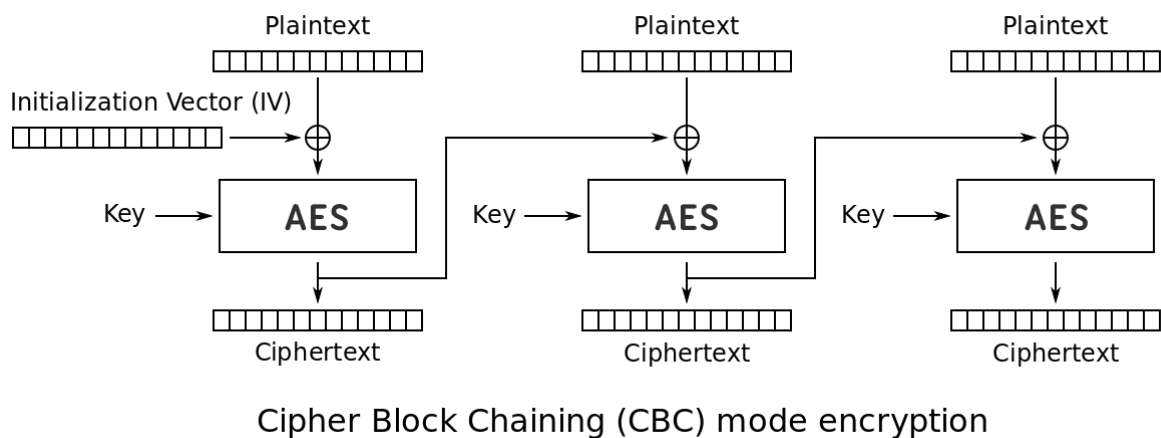
Η ασφάλεια διαδραματίζει πολύ σημαντικό ρόλο και για τη λειτουργία της εφαρμογής Doodlz. Τα δεδομένα που στέλνονται από τη συγκεκριμένη εφαρμογή πρέπει να προφυλάσσονται από επιθέσεις που έχουν σκοπό να τα υποκλέψουν ή να τα αλλοιώσουν.

Πρώτο βήμα για την ασφάλεια είναι η κρυπτογράφηση της φράσης ασφαλείας που ορίζεται από το χρήστη για την κωδικοποίηση της πληροφορίας που στέλνεται από την εφαρμογή. Η φράση αποθηκεύεται τοπικά στη συσκευή στο φάκελο της εφαρμογής που εμφανίζονται τα SharedPreferences, αλλά έγινε προσπάθεια να είναι σε μορφή μη αναγνωρίσιμη από κάποιο τρίτο. Έτσι, δεν είναι δυνατή η επεξεργασία της ήδη ορισμένης φράσης καθώς δεν μπορεί να αποκρυπτογραφηθεί, αλλά με την αντίστοιχη επιλογή ζητείται η εισαγωγή νέας. Για την κρυπτογράφηση της φράσης χρησιμοποιήθηκε η συνάρτηση σύνοψης SHA-256. Ο κώδικας παρακάτω παρουσιάζει τη διαδικασία κρυπτογράφησης της φράσης μέσα στην android εφαρμογή.

```
public class SHAencrypt {  
    public static String SHA256encrypt(String hint) throws Exception  
    {  
        String hex = "";  
  
        MessageDigest md = MessageDigest.getInstance("SHA-256");  
        md.update(hint.getBytes());  
  
        byte byteData[] = md.digest();  
  
        // convert the byte to hex format  
        StringBuffer hintHex = new StringBuffer();  
        for (int i=0; i<byteData.length; i++) {  
            hex = Integer.toHexString(0xff & byteData[i]);  
  
            if (hex.length() == 1)  
                hintHex.append('0');  
  
            hintHex.append(hex);  
        }  
  
        return hintHex.toString();  
    }  
}
```

Το επόμενο βήμα και το πιο σημαντικό σημείο για την ασφάλεια στην εφαρμογή Doodlz είναι η κρυπτογράφηση της θέσης της συσκευής πριν την αποστολή της στο διακομιστή. Όπως αναφέρθηκε παραπάνω, το Service λαμβάνει περιοδικές ενημερώσεις της τοποθεσίας

της συσκευής με τη βοήθεια του δικτύου ή του GPS. Στη συνέχεια, αυτές οι πληροφορίες κρυπτογραφούνται με το κοινό κρυφό κλειδί που έχει ορίσει ο γονέας και στέλνονται μέσω του κατάλληλου Web Service στον πίνακα PCA_devices της βάσης δεδομένων. Έτσι, ο κάθε χρήστης της εφαρμογής MapLocation δεν ανησυχεί μήπως τα δεδομένα των παιδιών του μπορούν να χρησιμοποιηθούν από τρίτους. Για την κρυπτογράφηση της θέσης χρησιμοποιείται το πρότυπο AES (Advanced Encryption Standard), που θεωρείται ένας από τους αλγόριθμους συμμετρικού κλειδιού (symmetric-key algorithm). Η διαδικασία της κρυπτογράφησης με τη συμβολή του συγκεκριμένου αλγόριθμου περιγράφεται στην παρακάτω εικόνα.



Εικόνα 5.4 Κρυπτογράφηση δεδομένων με AES

Για την κρυπτογράφηση των δεδομένων που στέλνονται από την εφαρμογή Doodlz καλείται από το Service η κλάση AESCBC. Σε αυτή στέλνονται το γεωγραφικό μήκος και πλάτος και η ακρίβεια της θέσης ενωμένα με το σύμβολο “_” ως μία φράση. Για την αρχικοποίηση του μυστικού κλειδιού χρησιμοποιείται η φράση ασφαλείας που έχει δηλώσει ο χρήστης, ενώ δημιουργείται και μία τυχαία ακολουθία που αποτελεί το IV (Initialization Vector) και θα σταλθεί έπειτα στη βάση δεδομένων μαζί με την κρυπτογραφημένη τοποθεσία. Στη συνέχεια, ορίζεται ποιος αλγόριθμος θα χρησιμοποιηθεί και αρχικοποιείται. Τέλος, ξεκινάει η διαδικασία της κρυπτογράφησης και επιστρέφεται η κρυπτογραφημένη πληροφορία στο Service για να σταλθεί στο Web Service του διακομιστή. Το παρακάτω κομμάτι κώδικα παρουσιάζει τη λειτουργία του αλγόριθμου AES.

```

public class AESCBC {

    private static byte[] keyValue = null;

    public KeyGenerator keygenerator;
    public SecretKey AESkey;

    private static IvParameterSpec ivSpec;
    public static byte[] iv;

    public String AESEncrypt(String data) throws Exception
    {
        // Generate the key
        Key key = generateKey();
        // Create the Cipher
        Cipher c = Cipher.getInstance("AES/CBC/PKCS5Padding");

        SecureRandom random = new SecureRandom();
        iv = new byte[16];
        random.nextBytes(iv);
        ivSpec = new IvParameterSpec(iv);

        // Initialize the cipher for encryption
        c.init(Cipher.ENCRYPT_MODE, key, ivSpec);

        byte[] encrData = c.doFinal(data.getBytes());
        String encryptedData = Base64.encodeToString(encrData,
Base64.DEFAULT);

        return encryptedData;
    }

    public void getKey(String hintText) throws Exception {

        byte[] b = hintText.getBytes(Charset.forName("UTF-8"));
        keyValue = new byte[b.length/2];

        for (int i=0; i<keyValue.length; i++) {
            keyValue[i] = (byte) (b[i] ^ b[i + keyValue.length]);
        }

    }

    private static Key generateKey() {
        Key key = new SecretKeySpec(keyValue, "AES");
        return key;
    }
}

```

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Οι εφαρμογές που δημιουργήθηκαν έχουν ως κύριο σκοπό αφενός την καταγραφή της θέσης των παιδιών και αφετέρου τον έλεγχο από το γονέα. Πρώτος στόχος ήταν η δημιουργία της εφαρμογής MapLocation για έξυπνο τηλέφωνο με λειτουργικό Android, την οποία θα διαχειρίζεται ο γονέας. Τα στοιχεία που μπορεί να ελέγξει είναι η θέση των εξαρτώμενων συσκευών των παιδιών του, η απόστασή τους από αυτόν, καθώς και η χρονική στιγμή που έγινε η καταγραφή. Επίσης, είναι δυνατή η προσθήκη νέων συσκευών και η διαχείριση της φράσης που θα χρησιμοποιηθεί για την κρυπτογράφηση των δεδομένων, ώστε να προασπιστεί η ασφαλής επικοινωνία μεταξύ των δύο εφαρμογών. Δεύτερος στόχος ήταν η επεξεργασία της ήδη υπάρχουσας εφαρμογής Doodlz ώστε να λειτουργήσει ως εφαρμογή εντοπισμού. Μόλις ενεργοποιείται η εφαρμογή στη συσκευή, το παιδί έχει τη δυνατότητα να σχεδιάσει χρησιμοποιώντας διάφορα εργαλεία που του παρέχει το πρόγραμμα, ενώ ταυτόχρονα στέλνεται το στίγμα της θέσης της συσκευής στο διακομιστή, το οποίο θα μπορεί να δει άμεσα ο γονέας.

Η αρχική ιδέα ήταν η δημιουργία μίας εφαρμογής που θα εξυπηρετούσε την ανάγκη του γονικού ελέγχου που θέλει να ασκεί ένας χρήστης στα προστατευόμενα μέλη της οικογένειας του χωρίς αυτά να το γνωρίζουν και μίας υπηρεσίας που θα τρέχει στο υπόβαθρο των εξαρτώμενων συσκευών και θα έστελνε περιοδικές ενημερώσεις της θέσης. Βέβαια, κατά τη διάρκεια της δημιουργίας κρίθηκε σκόπιμο να χρησιμοποιηθεί μία εφαρμογή που θα ενεργοποιούσε τη συγκεκριμένη υπηρεσία και θα απέκρυπτε από το παιδί τον απώτερο σκοπό του εντοπισμού του. Έτσι επιλέχθηκε η εφαρμογή Doodlz καθώς τηρούσε πλήρως τις προϋποθέσεις που είχαν τεθεί. Βασικό ρόλο στη δημιουργία των δύο εφαρμογών και σημαντικός παράγοντας που τις διαφοροποιεί από ήδη υπάρχουσες είναι η προσπάθεια διασφάλισης των δεδομένων του χρήστη και των παιδιών του από τρίτους, κυρίως με την κρυπτογράφηση των πιο σημαντικών δεδομένων που στέλνονται στο διακομιστή.

Τα αποτελέσματα ήταν ικανοποιητικά πετυχαίνοντας το μεγαλύτερο όγκο των αρχικών στόχων. Οι εφαρμογές δοκιμάστηκαν και σε πραγματικές συνθήκες και κρίθηκαν λειτουργικές με περιθώρια βελτίωσης, κυρίως όσον αφορά τις δυνατότητες που παρέχει η εφαρμογή MapLocation στο χρήστη της. Στα θετικά ήταν η εύκολη χρήση των δύο εφαρμογών σε έξυπνο τηλέφωνο, η δυνατότητα παρακολούθησης των παιδιών χωρίς μεγάλο κόστος και η αποτροπή υποκλοπής των πληροφοριών θέσης με σύγχρονες τεχνικές κρυπτογράφησης. Στα αρνητικά ήταν η πρόσβαση στην υπηρεσία εντοπισμού και αποστολής της τοποθεσίας στο διακομιστή μόνο με την ενεργοποίηση της εφαρμογής Doodlz. Με άλλα λόγια, στην περίπτωση που το παιδί δεν ενεργοποιήσει την εφαρμογή στη συσκευή του δε θα είναι διαθέσιμη η υπηρεσία εντοπισμού και ο γονέας δε θα είναι πλήρως ενημερωμένος για τις εξαρτώμενες από αυτόν συσκευές. Στα αρνητικά συγκαταλέγεται επίσης και η αδυναμία διασφάλισης της ακεραιότητας των δεδομένων που αποθηκεύονται στη βάση δεδομένων, καθώς δεν έχει δημιουργηθεί ένα ασφαλές κανάλι επικοινωνίας μεταξύ των εφαρμογών και του διακομιστή.

6.2 Δυνατότητες επέκτασης

Μελλοντικά είναι απαραίτητη η μετατροπή της εφαρμογής Doodlz του παιδιού σε υπηρεσία που θα τίθεται σε λειτουργία με την ενεργοποίηση της συσκευής και θα λειτουργεί στο υπόβαθρο στέλνοντας περιοδικές ενημερώσεις. Επίσης, είναι εξίσου σημαντική η διασφάλιση της ακεραιότητας των πληροφοριών που αποθηκεύονται στη βάση δεδομένων προκειμένου να αποτρέπεται η υποκλοπή, επεξεργασία ή διαγραφή τους από τρίτους. Με άλλα λόγια, πρέπει να αποφευχθεί κάποιος τρίτος να επεμβαίνει στην επικοινωνία των συσκευών και του διακομιστή ώστε να επεξεργαστεί τις κρυπτογραφημένες πληροφορίες ή να γνωρίζει τη χρονική στιγμή της εγγραφής. Για την επίτευξη του συγκεκριμένου εγχειρήματος χρειάζεται να εδραιωθεί ένα ασφαλές κανάλι επικοινωνίας των συσκευών με το διακομιστή για τη μετάδοση των δεδομένων. Μελλοντικά, καλό θα ήταν να παρέχεται η δυνατότητα στο γονέα να επιλέγει πέρα από την τελευταία γνωστή θέση μίας συσκευής και όλα τα σημεία που βρισκόταν η συγκεκριμένη σε κάποιο ορισμένο χρονικό διάστημα, καθώς και να μπορεί να χρησιμοποιήσει τις τεχνολογίες του geo-fencing ώστε να ορίζει μία περίμετρο στις περιοχές που μπορεί να κινηθεί ένα ανήλικο άτομο της οικογένειάς του και να λαμβάνει ειδοποιήσεις σε περίπτωση που το παιδί αποκλίνει από αυτήν. Επιπλέον, μπορεί να

προστεθεί λειτουργικότητα στην εφαρμογή MapLocation παρέχοντας τη δυνατότητα ανάκτησης της τοποθεσίας από τις εξαρτώμενες συσκευές σε πραγματικό χρόνο. Θα μπορούσε η εφαρμογή MapLocation να σχεδιαστεί και για άλλες πλατφόρμες με διαφορετικά λειτουργικά συστήματα (iOS, Windows Phone). Τέλος, η υπηρεσία εντοπισμού της εξαρτώμενης συσκευής θα μπορούσε να αποτρέπει το χρήστη από την απενεργοποίηση της πρόσβασης στο Διαδίκτυο, καθώς έτσι δεν θα μπορούσε να πραγματοποιηθεί η αποστολή της τοποθεσίας στο διακομιστή.

Βιβλιογραφία

1. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 20 Απρίλιος 2014.]
<http://en.wikipedia.org/wiki/Smartphone>
2. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 20 Απρίλιος 2014.]
http://en.wikipedia.org/wiki/Operating_system
3. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 20 Απρίλιος 2014.]
[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
4. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 20 Απρίλιος 2014.]
http://el.wikipedia.org/wiki/Global_Positioning_System
5. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 20 Απρίλιος 2014.]
<http://en.wikipedia.org/wiki/Geolocation>
6. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 21 Απρίλιος 2014.]
http://en.wikipedia.org/wiki/Location-based_service
7. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 21 Απρίλιος 2014.]
<http://en.wikipedia.org/wiki/Geo-fence>
8. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 21 Απρίλιος 2014.]
http://en.wikipedia.org/wiki/Google_Maps
9. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 6 Ιούνιος 2014.]
<http://en.wikipedia.org/wiki/MySQL>
10. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 6 Ιούνιος 2014.]
<http://en.wikipedia.org/wiki/PhpMyAdmin>
11. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 6 Ιούνιος 2014.]
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software))
12. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 6 Ιούνιος 2014.]
<http://el.wikipedia.org/wiki/Java>
13. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 6 Ιούνιος 2014.]
<http://en.wikipedia.org/wiki/GlassFish>
14. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 7 Ιούνιος 2014.]
<http://en.wikipedia.org/wiki/NetBeans>

15. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 7 Ιούνιος 2014.]
http://en.wikipedia.org/wiki/Representational_state_transfer
16. Google Developers. *Google Maps Android API v2*. [Ηλεκτρονικό]
<https://developers.google.com/maps/documentation/android/>
17. Developers. [Ηλεκτρονικό]
<http://developer.android.com/index.html>
18. AndroidHive. *Android working with Google Maps V2*. [Ηλεκτρονικό]
<http://www.androidhive.info/2013/08/android-working-with-google-maps-v2/>
19. Stackoverflow. *General questions over programming*. [Ηλεκτρονικό]
<http://stackoverflow.com/>
20. Netbeans. *Getting Started with RESTful Web Services*. [Ηλεκτρονικό]
<https://netbeans.org/kb/docs/websvc/rest.html>
21. Mr. Bipin S. Rupadiya. *AES with CBC mode example in Java*. [Ηλεκτρονικό]
<http://www.bipinrupadiya.com/2013/06/aes-with-cbc-mode-example-in-java.html>
22. **Deitel P, Deitel H, Deitel A, Morgano M.** (2012) *Android για Προγραμματιστές*. Εκδόσεις Μ. Γκιούρδας
23. **Reto Meier.** (2012) *Professional Android 4 Application Development*. John Wiley & Sons, Inc.
24. **Zigurd Mednieks, Laird Dormin, G. Blake Meike, Masumi Nakamura.** (2012) *Programming Android 2nd Edition*. O'Reilly