ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Θ Ε Μ Α ''ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΚΟΙΝΩΝΙΚΗΣ ΔΙΚΤΥΩΣΗΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ ANDROID''

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ: ΑΓΓΕΛΑΚΗΣ ΦΩΤΙΟΣ

ΕΠΙΒΛΕΠΩΝ: ΑΣΗΜΑΚΟΠΟΥΛΟΣ ΓΕΩΡΓΙΟΣ, ΚΑΘΗΓΗΤΗΣ ΕΦΑΡΜΟΓΩΝ

ΑΝΤΙΡΡΙΟ 2015

Περίληψη

Την εποχή που διανύουμε, τα έξυπνα κινητά ή smartphones είναι ιδαίτερα δημοφιλή και αποτελούν αναπόσπαστο κομμάτι της καθημερινότητάς μας. Σε ότι αφορά τα λειτουργικά συστήματα των κινητών τηλεφώνων, το Android κατέχει εξέχουσα θέση μιάς και από το 2013 κατέχει το μεγαλύτερο μερίδιο της αγοράς. Συνεπώς, η ανάπτυξη επιχειρηματικής δραστηριότητας στην αγορά των εφαρμογών για Android είναι πολλά υποσχόμενη. Σκοπός της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη μιας διαδραστικής εφαρμογής για το λειτουργικό σύστημα Android. Συγκεκριμένα, η εφαρμογή αυτή επιτρέπει στο χρήστη κινητού τηλεφώνου με λειτουργικό σύστημα Android να εντοπίσει άλλους χρήστες που βρίσκονται γεωγραφικά κοντά του και να επικοινωνήσει μαζί τους. Για να το καταφέρει αυτό, η εφαρμογή αξιοποιεί μια σειρά από υπηρεσίες όπως ο εντοπισμός της γεωγραφικής του θέσης μέσω GPS.

Abstract

In the recent times, smartphones have become particularly popular and are playing an important role in our daily lives. As regards the operating systems of mobile phones, Android holds a prominent position and from 2013 holds the largest market share. Therefore, the development of business activity in the market of Android applications is extremely promising. The purpose of this thesis is the development of an interactive application for the Android operating system. Specifically, this application allows the users of mobile phones running the Android operating system to locate other users who are geographically close to them and to communicate with them. To accomplish this, the application uses a range of services such as the detection of its location using GPS services.

Περιεχόμενα

Εισαγωγή

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη μιας διαδραστικής εφαρμογής για κινητά τηλέφωνα που τρέχουν το λειτουργικό σύστημα Android. Η εφαρμογή επιτρέπει στον χρήστη να ανιχνέυσει άλλους χρήστες που βρίσκονται γεωγραφικά κοντά του και να επικοινωνήσει μαζί τους μέσω γραπτών μηνυμάτων.

Στο πρώτο κεφάλαιο, κάνουμε αρχικά μια εισαγωγή στα έξυπνα κινητά τηλέφωνα και στη συνέχεια, παρουσιάζεται αναλυτικά η πλατφόρμα Android. Συγκεκριμένα, παρουσιάζεται η ιστορία του και η πορεία εξέλιξής του, το γιατί η ανάπτυξη σε Android είναι ελκυστική καθώς και η αρχιτεκτονική του. Τέλος περιγράφεται η τεχνολογία ανίχνευσης θέσης (GPS) καθώς και η δομή και λειτουργία της.

Στο δεύτερο κεφάλαιο, περιγράφεται η διαδικασία ανάπτυξης εφαρμογών για την πλατφόρμα Android και συγκεκριμένα, τα απαραίτητα προγραμματιστικά εργαλεία για την ανάπτυξη μιας εφαρμογής, τα τμήματα από τα οποία αποτελείται μια εφαρμογή καθώς και τα απαραίτητα αρχεία και οι θέσεις που πρέπει να βρίσκονται αυτά.

Στο τρίτο κεφάλαιο παρουσιάζεται η πορεία ανάπτυξης της εφαρμογής καθώς και τα κύρια σημεία της εφαρμογής, δηλαδή τα διάφορα κομμάτια κώδικα που συνθέτουν την εφαρμογή και η λειτουργία που επιτελούν έτσι ώστε να επιτευχθεί το τελικό επιθυμητό αποτέλεσμα.

Στο τελευταίο κεφάλαιο της εργασίας περιγράφονται τα συμπεράσματα που προέκυψαν από την εκπόνησή της καθώς και μελλοντικές επεκτάσεις.

Τέλος, παρατίθεται η Βιβλιογραφία καθώς και ένα Παράρτημα το οποίο περιλαμβάνει τον κώδικα της εφαρμογής.

1

Εισαγωγή στο Λειτουργικό Σύστημα Android

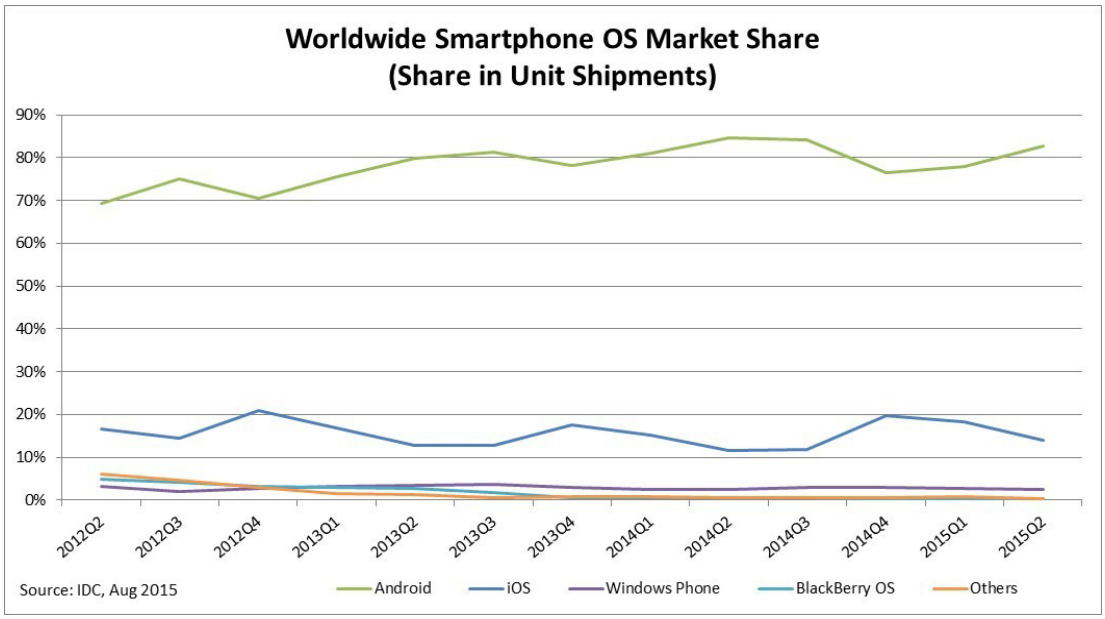
Στο κεφάλαιο αυτό κάνουμε μια εισαγωγή στα smartphones και στις δυνατότητές τους και στη συνέχεια, περιγράφουμε διάφορες πτυχές της πλατφόρμας Android όπως την ιστορία, την εξέλιξή της και την αρχιτεκτονική της.

1.1 Smartphones

Η κινητή τηλεφωνία έδωσε στον άνθρωπο τη δυνατότητα να επικοινωνεί με άλλους ανθρώπους οπουδήποτε κι αν βρίσκεται. Τα πρώτα κινητά τηλέφωνα επέτρεπαν μόνο την πραγματοποίηση κλήσεων προς άλλους χρήστες, ενώ αργότερα προστέθηκε και η δυνατότητα αποστολής γραπτών μηνυμάτων. Με την εξέλιξη της τεχνολογίας δόθηκε και η δυνατότητα αποστολής μηνυμάτων εμπλουτισμένου κειμένου, τα λεγόμενα mms. Κανένας όμως δεν περίμενε ότι τα κινητά τηλέφωνα θα αποκτούσαν δυνατότητες όπως αυτές που ζήσαμε κατά την έλευση των smartphones και μετά. Τα smartphoneseίναι κινητάτηλέφωνατα οποία διαθέτουν προηγμέναλειτουργικάσυστήματα πουσυνδυάζουντα χαρακτηριστικά των λειτουργικών συστημάτων προσωπικώνυπολογιστών μεάλλα χαρακτηριστικάχρήσιμαγια την φορητήχρήση. Κατά κανόνα,συνδυάζουν τα χαρακτηριστικάενόςκινητού τηλεφώνουμε εκείνα άλλωνδημοφιλώνφορητών συσκευών, όπωςπροσωπικών ψηφιακών βοηθών(PDA), συσκευών αναπαραγωγής πολυμέσων καιμονάδων πλοήγησης GPS. Τα περισσότερασmartphonesμπορούν να έχουν πρόσβασηστο Internet, διαθέτουν οθόνηαφής και κάμερα, μπορούν να

τρέξουνεφαρμογές τρίτων, καθώς και να αναπαράγουν μουσική. Το 2014, οι πωλήσειστων smartphonesπαγκοσμίωςέφτασαν τα1,2 δισεκατομμύρια [1].

Αν και υπήρχαν κινητά τηλέφωνα που προσέγγιζαν τα χαρακτηριστικά των smartphones και νωρίτερα, το μεγάλο μπάμ έγινε το 2007, όταν η Apple παρουσίασε το iPhone, ένα από τα πρώτα smartphones που χρησιμοποιούσαν μια διεπαφή πολλαπλών επαφών. Το iPhone έγινε γνωστό λόγω της μεγάλης οθόνης αφής του η οποία αποτελούσε το κύριο μέσο αλληλεπίδρασης με το χρήστη, αντί για μια γραφίδα ή πληκτρολόγιο τα οποία αποτελούσαν τους τυπικούς τρόπους αλληλεπίδρασης εκείνη την εποχή. Το 2008 εμφανίστηκε το HTC Dream (επίσης γνωστό ως το T-Mobile G1), το πρώτο τηλεφώνο που χρησιμοποιούσε το λειτουργικό σύστημα Android ονομάζεται HTC Dream (επίσης γνωστό ως το T-Mobile G1). Το Android είναι ένα open-source λειτουργικό σύστημα που δημιουργήθηκε από τον Andy Rubin και τώρα ανήκει στη Google. Αν και η αποδοχή του Android ήταν σχετικά αργή στην αρχή, άρχισε σιγά σιγά να κερδίζει δημοτικότητα στην αγορά το 2010 και σήμερα αποτελεί την κυρίαρχη πλατφόρμα. Αυτές οι δυο νέες πλατφόρμες οδήγησαν στην παρακμή των παλαιότερων. Η Microsoft, για παράδειγμα, ξεκίνησε ένα νέο λειτουργικό σύστημα από το μηδέν, που ονομάζεται Windows Phone. Η Nokia εγκατέλειψε το λειτουργικό Symbian και συνεργάζεται με τη Microsoft χρησιμοποιώντας Windows Phone στα smartphones της με αποτέλεσμα το Windows Phone να γίνει το τρίτο πιο δημοφιλές λειτουργικό σύστημα (Εικόνα 1.1).



Εικόνα 1.1 Μερίδιο αγοράς λειτουργικού συστήματος smartphones.

1.2 Η Πλατφόρμα Android

Το Android είναι ένα λειτουργικό σύστημα για κινητά τηλέφωνα που βασίζεται στον πυρήνα του Linux και αναπτύσσεται από την Google. Με μια διεπαφή χρήστη που βασίζεται στον άμεσο χειρισμό, το Android έχει σχεδιαστεί κυρίως για φορητές συσκευές με οθόνη αφής όπως smartphones και tablets, ενώ σε συνδυασμό με εξειδικευμένες διεπαφές χρήστη μπορεί να χρησιμοποιηθεί σε τηλεοράσεις (Android TV), αυτοκίνητα (Auto Android), και ρολόγια χειρός (Android φθορά). Το λειτουργικό σύστημα είναι ιδιαίτερα δημοφιλές στην αγορά, πράγμα που φαίνεται από την ανακοίνωση της Google το Σεπτέμβρη του 2013ότι περισσότερες από ένα δισεκατομμύριοενεργοποιημένες συσκευές Androidήταν σε χρήσησε όλο τον κόσμο [2].

1.2.1 Η Ιστορία του Android

Η εταιρεία Android ιδρύθηκε στο Πάλο Άλτο της Καλιφόρνια, τον Οκτώβριο του 2003 από τους Andy Rubin, Rich Miner, Nick Sears και Chris. Οι ιδρυτές της εταιρείας είχαν σκοπό να αναπτύξουν έξυπνες φορητές συσκευές που χρησιμοποιούν πληροφορίες σχετικές με την τοποθεσία και τις προτιμήσεις των χρηστών. Οι αρχικές προθέσεις της εταιρείας ήταν να αναπτύξει ένα προηγμένο λειτουργικό σύστημα για ψηφιακές φωτογραφικές μηχανές. Ωστόσο, λόγω του ότι η αγορά των φωτογραφικών μηχανών είναι αρκετά περιορισμένη, η εταιρεία έθεσε ως στόχο την παραγωγή ενός λειτουργικού συστήματος smartphone που θα ανταγωνιζόταν τα Symbian και Microsoft Windows Mobile. Τον Ιούλιο του 2005, η Google εξαγόρασετην AndroidInc και πολλοί υπέθεσαν ότι με την κίνησή της αυτή, η Google σχεδιάζεινα εισέλθει στην αγοράκινητής τηλεφωνίας.

Στις 5 Νοεμβρίου2007, διάφορες εταιρείες συμπεριλαμβανομένου τηςGoogle συνεργάστηκαν για να δημιουργήσουν τηνOpen Handset Alliance, μια κοινοπραξίαεταιρειώντεχνολογίας μεστόχο την ανάπτυξηανοιχτών προτύπωνγια τις κινητές συσκευές. Εκτός της Google συμμετείχανκατασκευαστές κινητών συσκευών, όπως η HTC, η Sonyκαι η Samsung, ασύρματοιφορείς όπωςη SprintNextelκαι η T-Mobile, καικατασκευαστέςchipset, όπως η QualcommκαιTexas Instruments. Εκείνη την ημέρα, το Androidπαρουσιάστηκεως το πρώτο προϊόντης κοινοπραξίας, μια πλατφόρμαγια κινητές συσκευές η οποίακατασκευάστηκεαπό την έκδοση τουπυρήνα Linux2.6.25. Η πρώτη κινητή συσκευή που έτρεχε AndroidήταντοHTC Dream

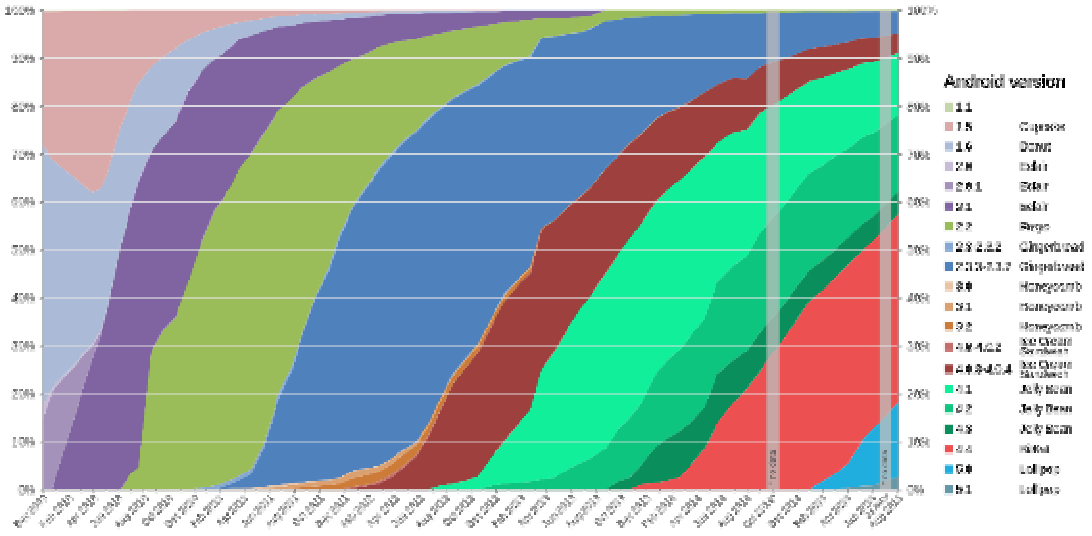
(Εικόνα 1.2), που κυκλοφόρησε στις22 Οκτωβρίου 2008. Το 2010, η Google λάνσαρε τις συσκευές Nexus, μια σειρά από smartphones και tablets που τρέχουν το λειτουργικό σύστημα Android, και κατασκευάστηκαν από συνεργάτες της. Το 2014, η Google διέθεσε στην αγορά το AndroidOne, ένα τυποποιημένοsmartphone, με στόχο κυρίωστους πελάτεςστον αναπτυσσόμενο κόσμο.

Εικόνα 1.2 Το κινητό τηλέφωνο HTC Dream.

1.2.2 Εξέλιξη του Android

Η ιστορία του λειτουργικού συστήματος Android ξεκίνησε με την έκδοση Android beta το Νοέμβριο του 2007. Η πρώτη εμπορική έκδοση, το Android 1.0, κυκλοφόρησε το Σεπτέμβριο του 2008. Η πιο πρόσφατη σημαντική ενημέρωση του Android είναι το Android 5.0 "Lollipop", το οποίο κυκλοφόρησε στις 3 Νοεμβρίου 2014. Στην Εικόνα 1.3 την παγκόσμια διανομή των εκδόσεων του Android από τον Δεκέμβριο του2009. Το Android είναι υπό συνεχή ανάπτυξη από την Google και την Open Handset Alliance. Η εξέλιξή του είναι ραγδαία και αυτό αντικατοπτρίζεται από το σύντομο διάστημα που παρεμβάλλεται μεταξύ των διάφορων εκδόσεών του. Όσον αφορά τον Αύγουστο του2015,το Android4.4"KitKat" είναι ηπιοευρέως χρησιμοποιούμενηέκδοση και λειτουργεί στο39,3% του συνόλου

τωνAndroidσυσκευών. Η δεύτερηπιοευρέως χρησιμοποιούμενηέκδοση είναι διαφορετικέςAndroid"Jelly Bean" εκδόσεις(4.1-4.3.1), με συνολικό μερίδιο33,6%.



Εικόνα 1.3 Παγκόσμια διανομή εκδόσεων του Android.

1.2.3 Γιατί Ανάπτυξη σε Android;

Το Android είναι μια ολοκληρωμένη, ανοιχτή και ελεύθερη πλατφόρμα για κινητά τηλέφωνα που περιλαμβάνει ένα λειτουργικό σύστημα, το απαραίτητο ενδιάμεσο λογισμικό, βιλιοθήκες και βασικές εφαρμογές. Επιπλέον, παρέχει στους προγραμματιστές όλα τα εργαλεία και APIs για να αρχίσουν να αναπτύσσουν λογισμικό για την πλατφόρμα Android χρησιμοποιώντας τη γλώσσα προγραμματισμού Java. Τα κύρια χαρακτηριστικά του είναι τα παρακάτω:

1) Λειτουργικότητα και Ευελιξία

Το Android είναι μια μοναδική πλατφόρμα που επιτρέπει την ανάπτυξη εφαρμογών λογισμικού το οποίο εκμεταλλεύεται πλήρως τις δυνατότητες μιας συμβατής συσκευής. Για παράδειγμα, οι προγραμματιστές εφαρμογών είναι ελεύθεροι να δημιουργήσουν εφαρμογές που χρησιμοποιούν οποιαδήποτε από τις βασικές λειτουργίες του τηλεφώνου όπως η αποστολή SMS, τηλεφωνικές κλήσεις, τη λήψη φωτογραφιών, το GPS κτλ. Έτσι διευκολύνονται στην ανάπτυξη πιο περίπλοκου και με περισσότερες λειτουργίες λογισμικού. Αυτό το λειτουργικό σύστημα κινητών τηλεφώνων (ή άλλων μικρών φορητών συσκευών που συνδέονται στο διαδίκτυο) στηρίζεται στον ελεύθερο πυρήνα του Linux. Επιπλέον, η πλατφόρμα ανάπτυξης Android είναι μια πλατφόρμα multi tasking, πράγμα που σημαίνει ότι κάθε εφαρμογή μπορεί να τρέξει στο τηλέφωνο ταυτόχρονα κάποια άλλη χωρίς να επηρεαστεί η απόδοσή τους, και αυτό είναι καλύτερο από το να περιορίζεται σε μία εφαρμογή κάθε

φορά. Το Android είναι μια πλατφόρμα ανοικτού κώδικα, πράγμα που σημαίνει ότι μπορεί εύκολα να επεκταθεί και να τροποποιηθεί για να συμβαδίζει και να υιοθετεί τις τελευταίες τεχνολογίες και εξελίξεις. Το γεγονός ότι και η πηγή της πλατφόρμας είναι ανοικτή διασφαλίζει ότι η ανάπτυξη το Android θα έχει συνεχή πρόοδο και θα εξελίσσεται αφού ένας μεγάλος αριθμός ικανών android προγραμματιστών εργάζεται για τη δημιουργία ελεύθερων για χρήση προηγμένων εργαλείων λογισμικού.

2) Πλήρη Παραμετροποίηση

Δεν υπάρχει διαφορά μεταξύ των λειτουργιών/εφαρμογών οι οποίες είναι ενσωματωμένες στο τηλεφώνου απο τις εφαρμογές που δημιουργούνται και προστίθενται από τρίτους προγραμματιστές Android. Οι τελευταίες μπορούν και έχουν την ίδια πρόσβαση σε όλες τις κύριες λειτουργίες της συσκευής κάτι που επιτρέπει στους τελικούς χρήστες να απολαμβάνουν ένα ευρύ φάσμα εφαρμογών Android που μπορούν να χρησιμοποιηθούν για τη σχεδόν απεριόριστους σκοπούς. Με συσκευές χτισμένες στην πλατφόρμα Android, οι χρήστες έχουν τη δυνατότητα να προσαρμόσουν πλήρως τη συσκευή τους ανάλογα με τις ανάγκες και τις απαιτήσεις τους. Τυχόν εφαρμογές ακόμα και οι βασικές λειτουργίες μπορεί να τροποποιηθούν ή να αντικατασταθούν πλήρως απο άλλες. Για παράδειγμα, ο χρήστης μπορεί να χρησιμοποιήσει την επιθυμητή του εφαρμογή για να εμφανίσει τις φωτογραφίες που είναι αποθηκευμένες στο τηλέφωνό του, ή για να έχει πρόσβαση στην αλληλογραφία του.

3) Διαδραστικότητα

Οι προγραμματιστές Android μπορούν να δημιουργήσουν πολύπλοκες καινοτόμες εφαρμογές με σχεδόν απεριόριστη λειτουργικότητα. Για παράδειγμα, μια εφαρμογή μπορεί να μεταδώσει τα δεδομένα από το κινητό σας με το διαδίκτυο (κάτι που μπορεί να περιλαμβάνει το ημερολόγιο σας και τις προγραμματισμένες εκδηλώσεις, λίστα με τις επαφές, τις φωτογραφίες σας και ακόμη και την τρέχουσα θέση σας, αλλά και παραγγελίες, τιμολόγια κτλ) και να λάβει όλα όσα μπορεί να χρειαστεί online και να εμφανίζονται στην οθόνη της συσκευής.

4) Απλότητα στην ανάπτυξη εφαρμογών

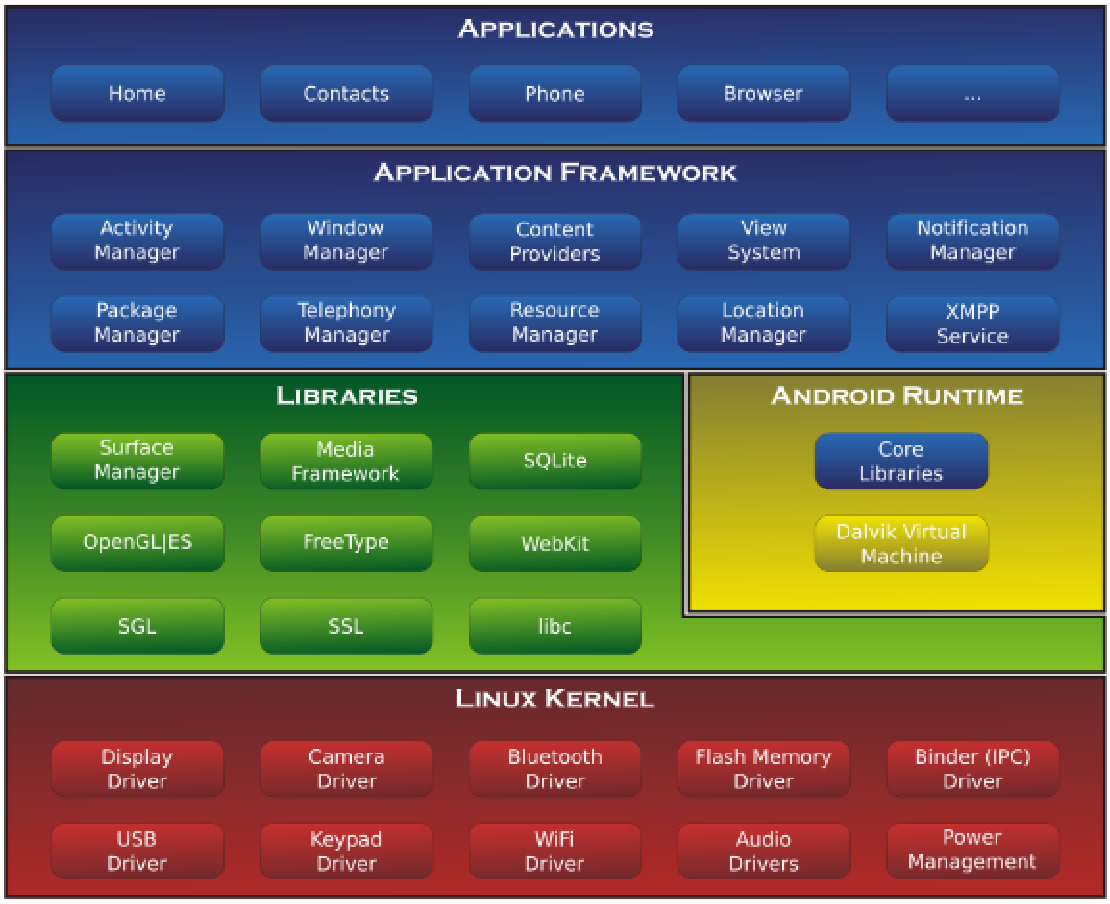
Η πλατφόρμα παρέχει στο καθένα που ασχολείται με την ανάπτυξη εφαρμογών τη δυνατότητα χρησιμοποίησης μια μεγάλης ποικιλίας από βιβλιοθήκες και τα χρήσιμα εκείνα εργαλεία που μπορούν να χρησιμοποιηθούν για τη δημιουργία του πιο εξελιγμένου λογισμικού. Αυτή η ολοκληρωμένη δέσμη από έτοιμα εργαλεία αυξάνει σημαντικά την παραγωγικότητα των προγραμματιστών Android εφαρμογών και τους

βοηθά να δημιουργήσουν εκπληκτικά πλούσιο λογισμικό γρηγορότερα και με λιγότερα λάθη.

1.2.4 Αρχιτεκτονική Android

Η αρχιτεκτονική του Android περιλαμβάνει τα εξής επίπεδα [3] τα οποία φαίνονται στην Εικόνα 1.4:

- **Επίπεδο Εφαρμογών (Applications):** Το Android περιλαμβάνει ένα σύνολο από βασικές εφαρμογές που έχουν ένα email client, ένα πρόγραμμα για SMS μηνύματα, ημερολόγιο, χάρτες (Google Maps), περιηγητή ιστού, πρόγραμμα για δομημένη αποθήκευση των επαφών και άλλα. Όλες οι εφαρμογές είναι γραμμένες στην γλώσσα προγραμματισμού Java.
- **Επίπεδο Πλαισίου Εφαρμογών (Applications Framework):** Το Android προσφέρει στους προγραμματιστές την δυνατότητα να κατασκευάσουν πλούσιες και καινοτόμες εφαρμογές. Οι προγραμματιστές αφήνονται ελεύθεροι να εκμεταλλευτούν πλήρως το hardware της συσκευής, να έχουν πρόσβαση σε υπηρεσίες εντοπισμού θέσης, να θέσουν χρονοδιακόπτες για εμφάνιση ειδοποιήσεων και πολλά άλλα. Επίσης έχουν πλήρη πρόσβαση στο ίδιο πλαίσιο από APIs που έχουν οι βασικές εφαρμογές του Android.
- **Επίπεδο Βιβλιοθηκών (Libraries):** Το οποίο περιλαμβάνει ένα σύνολο από βιβλιοθήκες γραμμένες σε C/C++ οι οποίες χρησιμοποιούνται από διάφορα στοιχεία του συστήματος του Android. Οι δυνατότητες που προσφέρουν οι βιβλιοθήκες αυτές είναι προσβάσιμες στους προγραμματιστές.
- **Επίπεδο Εκτέλεσης (Android Runtime):** Το οποίο αποτελείται από ένα σύνολο από βασικές βιβλιοθήκες και την Dalvik Virtual Machine.
- **Πυρήνας του Linux:** Το Android βασίζεται στον πυρήνα του Linux για βασικές υπηρεσίες συστήματος όπως ασφάλεια, διαχείριση μνήμης, διαχείριση διεργασιών, στοίβα δικτύου, και οδηγούς συσκευών. Από τον Απρίλιο του2014, οι συσκευέςAndroid χρησιμοποιούνκυρίωςτις εκδόσεις3.4ή3.10του πυρήνα του Linux.



Εικόνα 1.4 Αρχιτεκτονική του Android.

1.3 Τεχνολογία Ανίχνευσης Θέσης GPS

Ο προσδιορισμός της θέσης αποτέλεσε έναν πρωταρχικό σκοπό των δραστηριοτήτων (οικονομικών, επιστημονικών, κοινωνικών, κ.α.) του ανθρώπου. Για το σκοπό αυτό, κατά το πέρασμα των αιώνων, επινοήθηκαν και εφαρμόστηκαν διάφοροι μέθοδοι προσδιορισμού θέσης.

Το GPS (Global Positioning System) είναι ένα παγκόσμιο σύστημα εντοπισμού θέσης, το οποίο για την λειτουργία του χρησιμοποιεί ένα σύνολο 24 δορυφόρων οι οποίοι μεταδίδουν ακριβή σήματα μικροκυμάτων τα οποία επιτρέπουν στους δέκτες GPS να καθορίσουν την τρέχουσα θέση τους, την τρέχουσα ώρα και την τρέχουσα ταυτότητα τους. Ένας δέκτης GPS υπολογίζει την τρέχουσα θέση του μετρώντας προσεκτικά τους χρόνους αφίξεων των σημάτων τα οποία στέλνονται από τους GPS δορυφόρους.

Κάθε δορυφόρος μεταδίδει διαρκώς μηνύματα που περιέχουν πληροφορία για την ακριβή ώρα αποστολής του μηνύματος, για την ακριβή θέση του την στιγμή της αποστολής, για την συνολική κατάσταση του συστήματος και για τις τροχιές όλων

των υπόλοιπων δορυφόρων. Ο δέκτης μετράει τον χρόνο που χρειάζεται κάθε σήμα να φτάσει σε αυτόν και υπολογίζει την απόσταση του από κάθε δορυφόρο. Για να προσδιοριστεί η ακριβής θέση του δέκτη συνδυάζονται οι μετρούμενες αποστάσεις με τις θέσεις των δορυφόρων με την γεωμετρική μέθοδο της τριγωνοποίησης.

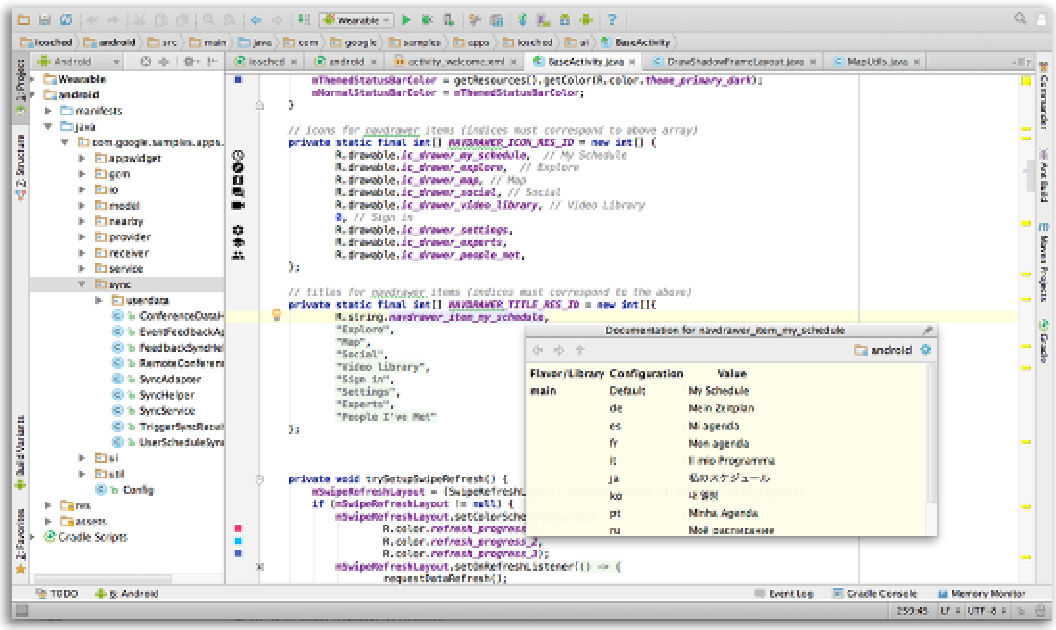
2

Προγραμματισμός σε Περιβάλλον Android

Όπως αναφέρθηκε προηγουμένως, το Android αποτελεί λογισμικό ανοιχτού κώδικα με συνέπεια οι κατασκευαστές κινητών τηλεφώνων και άλλων ηλεκτρονικών συσκευών να μπορούν να το χρησιμοποιούν στα προϊόντα τους ελεύθερα και χωρίς κόστος. Αυτό συνέβαλλε σε μεγάλο βαθμό στην ταχύτατη διάδοση του λειτουργικού αυτού συστήματος στην αγορά των smartphones. Εκτός όμως από το ίδιο το Android, διατίθενται ελεύθερα και προγραμματιστικά εργαλεία για την ανάπτυξη εφαρμογών σε αυτό, καθώς και αναλυτικοί οδηγοί ανάπτυξης εφαρμογών. Ακολουθεί μία σύντομη περιγραφή των κυριότερων προγραμματιστικών εργαλείων καθώς και μία ανάλυση των βασικών στοιχείων που αφορούν στην ανάπτυξη εφαρμογών σε λειτουργικό σύστημα Android.

2.1 Προγραμματιστικά Εργαλεία

Η γλώσσα προγραμματισμού που χρησιμοποιείται για την κατασκευή εφαρμογών Android είναι η Java, η οποία έχει πλήρη πρόσβαση στο API του Android. Συνεπώς, βασική προϋπόθεση για την ανάπτυξη εφαρμογών Android είναι η εγκατάσταση του Java Development Kit (JDK). Επίσης, είναι βολικό να χρησιμοποιήσουμε ένα περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) για τη συγγραφή του κώδικα και την εκτέλεση του προγράμματός μας. Αρχικά, το περιβάλλον ανάπτυξηςπου υποστήριζε η Google ήταν το Eclipseχρησιμοποιώντας το pluginAndroidDevelopment Tools (ADT). Τον Δεκέμβριο του2014, η ίδια η Google κυκλοφόρησετο AndroidStudio (Εικόνα 2.1), το οποίο βασίζεται στο IntelliJIDEA, και το κατοχύρωσε ως το κύριοπεριβάλλον ανάπτυξη εφαρμογών Android.



Εικόνα 2.1 Το περιβάλλον ανάπτυξης Android Studio

Το πιο βασικό εργαλείο για την ανάπτυξη Android εφαρμογών είναι το Android SDK (software development kit) το οποίο μας παρέχει τα επιπλέον εργαλεία ώστε να μπορούμε να γράψουμε κώδικα συγκεκριμένα για την κατασκευή εφαρμογής Android. Συγκεκριμένα, το SDKπεριλαμβάνει μια πλήρη σειράαπό εργαλεία ανάπτυξης, συμπεριλαμβανομένων ενός προγράμματος εντοπισμού σφαλμάτων, βιβλιοθηκών λογισμικού, ενόςεξομοιωτήβασισμένου στοQEMU, εγγράφων, δειγμάτων κώδικα, και παραδειγμάτων.

Για να τρέξουμε την Android εφαρμογή μας, είτε θα την εγκαταστήσουμε σε ένα smartphone ή θα χρησιμοποιήσουμε κάποιον εξομοιωτή κινητού τηλεφώνου. Το AndroidSDK περιλαμβάνει έναν εξομοιωτή κινητών συσκευών, δηλαδήμια εικονικήφορητή συσκευήπου τρέχειστον υπολογιστή (Εικόνα 1.2). Ο εξομοιωτήςεπιτρέπει τηνανάπτυξη και τη δοκιμήεφαρμογώντου Androidχωρίς τη χρήσηφυσικής συσκευής.

Εικόνα 2.2 Εικονική συσκευή με λειτουργικό σύστημα Android.

2.2 Δομικά Μέρη μιας Εφαρμογής Android

Η κάθε εφαρμογή Android περιέχει κάποια απαραίτητα δομικά στοιχεία. Κάθεστοιχείοείναι ένα διαφορετικόσημείο από το οποίοτοσύστημαμπορεί να εισέλθειστην εφαρμογή μας. Υπάρχουντέσσερις διαφορετικοί τύποιδομικών στοιχείων: δραστηριότητες (activities), υπηρεσίες (services), πάροχοι περιεχομένου (content providers) και παραλήπτες εκπομπών (broadcast receivers). Κάθε τύποςεξυπηρετεί ένασυγκεκριμένο σκοπόκαιέχει ένα ξεχωριστόκύκλο ζωήςπου καθορίζειτο πώςτο στοιχείοδημιουργείται καικαταστρέφεται.

Δραστηριότητες

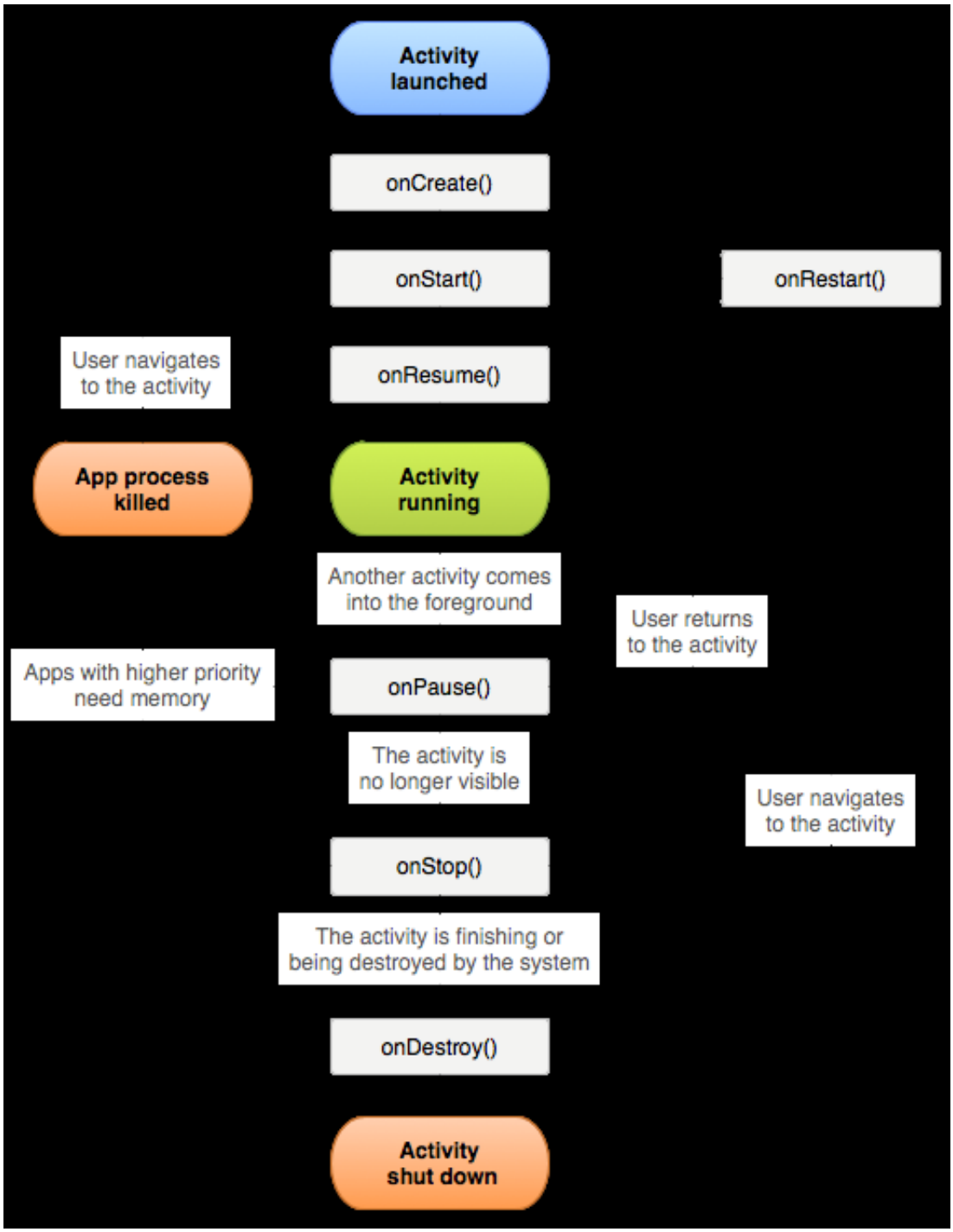
Πρόκειται ίσως για το κύριο δομικό στοιχείο μιας εφαρμογής. Μια δραστηριότητααντιπροσωπεύειένα στιγμιότυπο της οθόνηςμε μια διεπαφήχρήστη. Γιαπαράδειγμα, μίαεφαρμογή ηλεκτρονικού ταχυδρομείουμπορεί να έχειμίαδραστηριότητα πουεμφανίζει μια λίστατωνμηνυμάτων ηλεκτρονικού ταχυδρομείου, μια άλλη δραστηριότηταγια να συνθέσειένα μήνυμα ηλεκτρονικού

ταχυδρομείου, καθώς και μια τρίτη δραστηριότηταγια την ανάγνωση ενόςe-mail. Αν και οιδραστηριότητεςδουλεύουνμαζί για να δημιουργήσουν μια ομαλήεμπειρία στο χρήστητης εφαρμογής,κάθε μιαείναι ανεξάρτητη απότις άλλες.Ως εκ τούτου, μιαδιαφορετικήεφαρμογή μπορεί ναξεκινήσειοποιαδήποτε απότις δραστηριότητες αυτές. Γιαπαράδειγμα, μια εφαρμογήτης κάμεραςμπορεί να ξεκινήσειτη δραστηριότηταπουσυνθέτει ένανέο e-mail, έτσι ώστεο χρήστης ναμοιράσει μια εικόνα που έχει βγάλει.

Κάθε δραστηριότηταυλοποιείται ωςμιαυποκλάση της κλάσης Activity και υλοποιεί τη μέθοδο onCreate() η οποία καλείται τη στιγμή που ξεκινάει η δραστηριότητα. Επίσης, μέσα στη μέθοδο αυτή πρέπει να δημιουργηθεί η διεπαφή χρήστη η οποία δηλώνεται με την εντολή setContentView() και η findViewById() για να ανακτήσει τα διάφορα widgets προγραμματιστικά. Για να ξεκινήσουμε μια δραστηριότητα MyActivity μέσα από μία άλλη δραστηριότητα χρησιμοποιούμε τις παρακάτω γραμμές κώδικα:

```
Intent intent = new Intent (this, MyActivity.class);
startActivity(intent);
```

Για να τερματίσουμε μια δραστηριότητα χρησιμοποιούμε τη μέθοδο finish(). Το Android διαχειρίζεται από μόνο του τον κύκλο ζωής των δραστηριοτήτων, οπότε δεν χρειάζεται να τις τερματίσουμε εμείς. Ο κύκλος ζωής και οι αντίστοιχες μέθοδοι των δραστηριοτήτων φαίνονται στην Εικόνα 2.3



Εικόνα 2.3 Κύκλος ζωής δραστηριοτήτων και αντίστοιχες μέθοδοι [4].

Υπηρεσίες

Μια υπηρεσίαείναι ένα στοιχείο πουτρέχει στο παρασκήνιογια να εκτελέσει μακροχρόνιεςεργασίες ήεργασίεςγια απομακρυσμένεςδιαδικασίες. Οι υπηρεσίες δεν παρέχουνδιεπαφή χρήστη. Γιαπαράδειγμα, μια υπηρεσίαμπορεί να παίξειμουσικήστο παρασκήνιο ενώο χρήστης βρίσκεταισε μιαδιαφορετικήεφαρμογή, ή θα μπορούσε ναφέρει δεδομέναμέσω δικτύουχωρίς να εμποδίζει τηναλληλεπίδραση του χρήστημε

κάποια δραστηριότητα. Οι υπηρεσίες εκκινούνται από άλλα στοιχεία όπως δραστηριότητες, οι οποίες στη συνέχεια τις αφήνουν να τρέξουνή συνδέονται μεαυτές, ούτως ώστεναλληλεπιδράσουν μαζί τους.

Κάθε υπηρεσίαυλοποιείται ωςμιαυποκλάση της κλάσης Service και έχει δύο βασικές μεθόδου: την startService() και την bindService(). Η πρώτη καλείται από κάποια δραστηριότητα για να ξεκινήσει την υπηρεσία η οποία συνεχίζει να εκτελείται ακόμα και αν η δραστηριότητα που την ξεκίνησε τερματιστεί. Για το λόγο αυτό οι υπηρεσίες δεν επιστρέφουν κάποιο αποτέλεσμα στις δραστηριότητες που τις κάλεσαν αλλά εκτελούν κάποια λειτουργία όπως το κατέβασμα κάποιου αρχείου από το διαδίκτυο όπως αναφέρθηκε παραπάνω. Η δεύτερη «δένει» την υπηρεσία με κάποιο άλλο στοιχείο της εφαρμογής, όπως με μία δραστηριότητα, έτσι ώστε να υπάρχει αλληλεπίδραση μεταξύ των στοιχείων αυτών, όπως η αποστολή και λήψη αποτελεσμάτων. Έτσι αν τερματιστεί η δραστηριότητα τερματίζεται και η υπηρεσία. Σε αντίθεση με τις δραστηριότητες, το Android δεν διαχειρίζεται τον κύκλο ζωής των υπηρεσιών. Επομένως, κάθε υπηρεσία θα πρέπει να δηλώνει το πότε θα σταματήσει μέσω της μεθόδου stopSelf().

Πάροχοι Περιεχομένου

Ένας πάροχοςπεριεχομένουδιαχειρίζεταιένα κοινόσύνολο δεδομένων μιας εφαρμογής. Τα δεδομένα μπορούν να αποθηκευτούνστοσύστημα αρχείων, σε μιαβάση δεδομένων SQLite, στο διαδίκτυοή σε οποιαδήποτε άλληθέση αποθήκευσηςστην οποία έχει πρόσβαση η εφαρμογή. Μέσω του παρόχου περιεχομένου, άλλες εφαρμογέςμπορούν να προσπελάσουνήακόμα και να τροποποιήσουνταδεδομένα(αν ο πάροχος περιεχομένου το επιτρέπει). Γιαπαράδειγμα, τοσύστημα Androidπαρέχειμια υπηρεσία παροχήςπεριεχομένου πουδιαχειρίζεταιτις επαφές του χρήστη. Ως εκ τούτου, οποιαδήποτεεφαρμογή έχειτακατάλληλα δικαιώματα μπορεί να μέσω του παρόχου περιεχομένουναδιαβάσει και να γράψειπληροφορίες σχετικά μεένα συγκεκριμένο πρόσωπο.

Κάθε πάροχοςπεριεχομένουυλοποιείται ωςμιαυποκλάσητης κλάσηςContentProviderκαιπρέπει να υλοποιεί στο εσωτερικό τουένα τυποποιημένο σύνολοαπό APIsπου επιτρέπουν σε άλλες εφαρμογέςνα πραγματοποιούν συναλλαγές.

Παραλήπτες Εκπομπών

Ένας παραλήπτης εκπομπώνείναι ένα στοιχείο πουανταποκρίνεται σεόλο το σύστημαμετάδοσηςανακοινώσεων. Πολλέςεκπομπέςπροέρχονταιαπότοσύστημα, για

παράδειγμα, μία εκπομπή που ανακοινώνει ότιη οθόνηέχεισβήσει, ότι η μπαταρίαείναι χαμηλή, ή ότι μια φωτογραφία τραβήχτηκε. Οιεκπομπές μπορείνα προέρχονται κι από εφαρμογές. Για παράδειγμα, για να ενημερώσειάλλες εφαρμογέςότι έχει κατεβάσει κάποια δεδομένα και είναι διαθέσιμα να τα χρησιμοποιήσουν. Αν καιοι παραλήπτες εκπομπώνδενεμφανίζουν διεπαφέςχρήστη, μπορεί να δημιουργήσουνμια ειδοποίησηστη γραμμή κατάστασηςγια ναειδοποιήσουντο χρήστη ότανσυμβεί μιαεκπομπή.Πιο συχνάόμως, ένας παραλήπτηςεκπομπών είναιαπλά μια «πύλη» προς άλλα στοιχεία καιέχει ως στόχονα κάνει έναπολύ ελάχιστοποσό της εργασίας. Για παράδειγμα, θα μπορούσε ναξεκινήσει μιαυπηρεσίαγια να εκτελέσεικάποιες εργασίεςπου βασίζονταιστο γεγονός.

Κάθε παραλήπτηςεκπομπώνυλοποιείται ωςμιαυποκλάση της κλάσηςBroadcastReceiverκαι κάθεεκπομπήπαραδίδεταιως έναντικείμενο της κλάσης Intent.

2.3 Το Αρχείο Manifest

Κάθε εφαρμογή περιέχει ένα αρχείο με τίτλο AndroidManifest.xml στο οποίο βρίσκονται καταχωρημένες οι σημαντικότερες πληροφορίες της. Το αρχείο αυτό περιέχει πληροφορίες τις οποίες το λειτουργικό σύστημα πρέπει να γνωρίζει για να μπορέσει να τρέξει την εφαρμογή. Ο προγραμματιστής καταχωρεί στο αρχείο αυτό τις σημαντικότερες πληροφορίες της εφαρμογής. Κάποιες από αυτές τις πληροφορίες είναι:

- Η ονομασία του πακέτου της Java εφαρμογής.
- Η έκδοση της εφαρμογής.
- Η έκδοση του λειτουργικού συστήματος Android που απαιτεί η εφαρμογή.
- Το όνομα της εφαρμογής καθώς και το εικονίδιο της.
- Οι άδειες που χρειάζονται για να εκτελεστούν ορισμένες λειτουργίες της εφαρμογής.
- Όλα τα συστατικά στοιχεία(δραστηριότητες, υπηρεσίες, πάροχοι περιεχομένου, παραλήπτες εκπομπών) της εφαρμογής.
- Οι εξωτερικές βιβλιοθήκες που χρησιμοποιεί η εφαρμογή.

2.4 Οι Φάκελοι src και res

Ο φάκελος src περιέχει τα αρχεία Java που αντιστοιχούν σε όλες τις δραστηριότητες, υπηρεσίες, παρόχους περιεχομένου, βοηθητικά αρχεία, κλπ της εφαρμογής. Ο φάκελος αυτός συνεπώς περιέχει το πακέτο ή τα πακέτα της εφαρμογής τα οποία περιέχουν τα αρχεία Java, και αποτελεί τον μοναδικό φάκελο στο project στον οποίο αποθηκεύονται αρχεία κώδικα.

Από την άλλη μεριά, ο φάκελος res περιέχει όλα τα αρχεία εικόνας, κειμένου, xml layout, κλπ τα οποία χρησιμοποιούνται από τις δραστηριότητες που αντιστοιχούν σε κλάσεις στον φάκελο src. Φυσικά, δεν βρίσκονται όλα τα αρχεία πόρων, σε έναν αποκλειστικά φάκελο, αλλά είναι χωρισμένα και ταξινομημένα σε υποφακέλους ανάλογα με το είδος τους. Συνηθισμένοι υποφάκελοι του κύριου φακέλου res, είναι ο φάκελος drawable ο οποίος περιέχει τα αρχεία εικόνας (.png, .jpg, .gif) τα οποία χρησιμοποιεί η εφαρμογή, ο φάκελος layout ο οποίος περιέχει όλα τα αρχεία xml τα οποία ορίζουν τα διάφορα layouts που υπάρχουν στην εφαρμογή, και τέλος ο φάκελος values στον οποίο αποθηκεύονται τιμές για διάφορες μεταβλητές της εφαρμογής.

2.5 Υπόλοιποι Φάκελοι

Εκτός από τους δυο φακέλους που περιγράφηκαν παραπάνω, υπάρχουν και αρκετοί επιπλέον φάκελοι. Κάποιοι από αυτούς είναι ο ο φάκελος με τα διαθέσιμα APIs, ο φάκελος με τις διαθέσιμες βιβλιοθήκες που έχουμε εισάγει στο project μας, ο φάκελος bin ο οποίος παράγεται από τον compiler και περιέχει όλα τα αρχεία με κατάληξη .class για να παράγει το αρχείο apk , το οποίο είναι το εκτελέσιμο αρχείο στο λειτουργικό σύστημα android, καθώς και ο φάκελος gen ο οποίος περιέχει Java κλάσεις που παράγονται αυτόματα από το σύστημα.

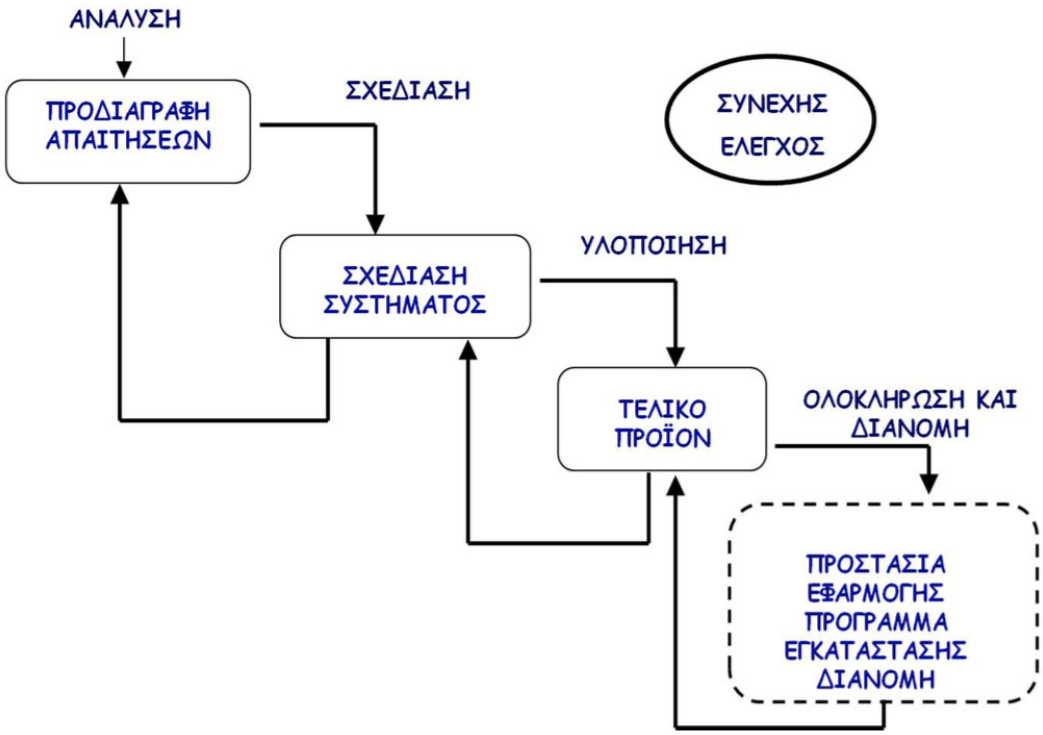
3

Η Εφαρμογή Κοινωνικής Δικτύωσης Socializer

Στα πλαίσια της παρούσας πτυχιακής εργασίας αναπτύχθηκε μια εφαρμογή κοινωνικής δικτύωσης για το περιβάλλον Android. Συγκεκριμένα, η εφαρμογή αυτή δίνει τη δυνατότητα στο χρήστη να επικοινωνήσει με άλλους χρήστες που βρίσκονται γεωγραφικά κοντά του και για αυτό της δόθηκε το όνομα Socializer. Οι τρόποι χρήσης της εφαρμογής είναι πολλοί και παρουσιάζουν μεγάλη ποικιλλία. Για παράδειγμα, η εφαρμογή μπορεί να φανεί ιδαίτερα χρήσιμη σε χρήστες που βρίσκονται σε ένα άγνωστο για αυτούς μέρος και θέλουν να μάθουν κάποια τοπική πληροφορία ή σε χρήστες που θέλουν να γνωρίσουν ανθρώπους με τους οποίους κατοικούν στην ίδια γειτονιά. Στην αρχή αυτού του κεφαλαίου, θα περιγραφούν οι τέσσερις φάσεις ανάπτυξης της εφαρμογής, ενώ στο υπόλοιπο αυτού του κεφαλαίου, θα παρουσιάσουμε τα κύρια σημεία της εφαρμογής Socializer καθώς και τα κύρια σημεία του server που χρησιμοποιείται για να συντονίσει τους διάφορους χρήστες της εφαρμογής. Ουσιαστικά, η εφαρμογή αποτελείται από μια σειρά από δραστηριότητες όπου κάθε μία καλεί κάποια άλλη ανάλογα με τις ενέργειες του χρήστη. Για αυτό το λόγο θα αφιερωσουμε το μεγαλύτερο μέρος του κεφαλαίου στην ανάλυση των κύριων δραστηριοτήτων.

3.1 Φάσεις Ανάπτυξης Εφαρμογής

Σε αυτό το κεφάλαιο θα αναλύσουμε τις φάσεις ανάπτυξης της εφαρμογής. Αναπτύξαμε την εφαρμογή με βάση το μοντέλο καταρράκτη. Γενικά, πριν ξεκινήσει η ανάπτυξη οποιασδήποτε εφαρμογής, πρέπει να πραγματοποιείται η ανάλυση απαιτήσεων χρήστη και συστήματος. Μόνο μετά από αυτά λαμβάνει χώρα η σχεδίαση και εν συνεχεία η υλοποίηση. Τελος, έχουμε τη φάση ελέγχου του συστήματος και διανομής. Οι φάσεις ανάπτυξης μιας εφαρμογής σύμφωνα με το μοντέλο καταρράκτη φαίνεται στην Εικόνα 3.1.



Εικόνα 3.1 Μοντέλο καταρράκτη [7].

3.1.1 Ανάλυση

Στην ενότητα αυτή παρουσιάζεται η φάση της ανάλυσης του συστήματος καθώς και μια αρχική προσέγγιση της εφαρμογής που πρόκειται να αναπτυχθεί. Αποτελεί την πρώτη φάση και μια από τις σημαντικότερες ώστε να προχωρήσουμε ορθά στη σχεδίαση και στην υλοποιηση.

Στην παρούσα φάση της εφαρμογής θα καθοριστούν οι κύριοι στόχοι της καθώς και θα συγκεντρωθούν οι απαιτήσεις χρηστών. Παρακάτω ακολουθεί ένας βοηθητικός πίνακας ερωταπαντήσεων που θα συντελέσει στην περαιτέρω οργάνωση της δομής του έργου.

Ανάλυση Χρηστών

Ερώτηση	Απάντηση
Ποιά είναι η ηλικία χρηστών;	Άνω των 12
Τι γλώσσα μιλούν;	Ελληνικά
Έχουν χρησιμοποιήσει παλιότερα smartphone;	Ναι
Έχουν χρησιμοποιήσει παρόμοιες εφαρμογές;	Όχι
Υφίσταται φυλετικός διαχωρισμός;	Όχι
Σε ποιά βαθμίδα εκπαίδευσης ανήκουν;	Από πρωτοβάθμια και άνω

Ανάλυση Τεχνολογίας και Χρόνου Υλοποίησης

Ερώτηση	Απάντηση
Χρειάζεται πρόσβαση στο Διαδίκτυο;	Ναι
Απαιτείται πρόσβαση σε τοποθεσία χρήστη;	Ναι
Τι τεχνολογία πρέπει να χρησιμοποιηθει;	3G, 4G, Wi-Fi, GPS
Τι πολυμεσικά στοιχεία απαιτούνται;	Smartphone με λειτουργικό σύστημα Android από έκδοση 3 και πάνω
Πώς θα γίνει η διανομή του τελικού προιόντος;	Προσωπική χρήση στα πλαίσια πτυχιακής εργασίας
Πόσο χρόνο διαθέτουμε για την ανάπτυξη του έργου;	Πέντε μήνες

Έπειτα καθορίστηκαν τα τμήματα υλοποίησης του έργου και αποφασίστηκε ο χρονοπρογραμματισμός αυτού. Παράλληλα, τα παραπάνω τμήματα τοποθετήθηκαν σε χρονική σειρά και καθορίστηκε η χρονική τους διάρκεια. Στον παρακάτω πίνακα ακολουθεί το χρονοδιάγραμμα εκπόνησης έργου.

Πλάνο και Χρονοπρογραμματισμός Έργου

Πλάνο Εργασιών	**Μάιος**	**Ιούνιος**	**Ιούλιος**	**Αύγουστος**	**Σεπτέμβριος**
Φάση Α: Ανάλυση					
Καταγραφή βασικών ιδεών					
Χρονο προγραμματισμός και πλάνο έργου					
Φάση Β: Σχεδίαση					
Κατηγοριοποίηση και επιλογή περιεχομένου					
Φάση Γ: Υλοποίηση					
Συγκέντρωση και επεξεργασία πρωτογενούς υλικού					
Φάση Δ: Έλεγχος					
Έλεγχος					

λειτουργίας εφαρμογής και απιτήσεων χρηστών					
Διορθώσεις για την παραγωγή του τελικού προιόντος					

3.1.2 Σχεδίαση

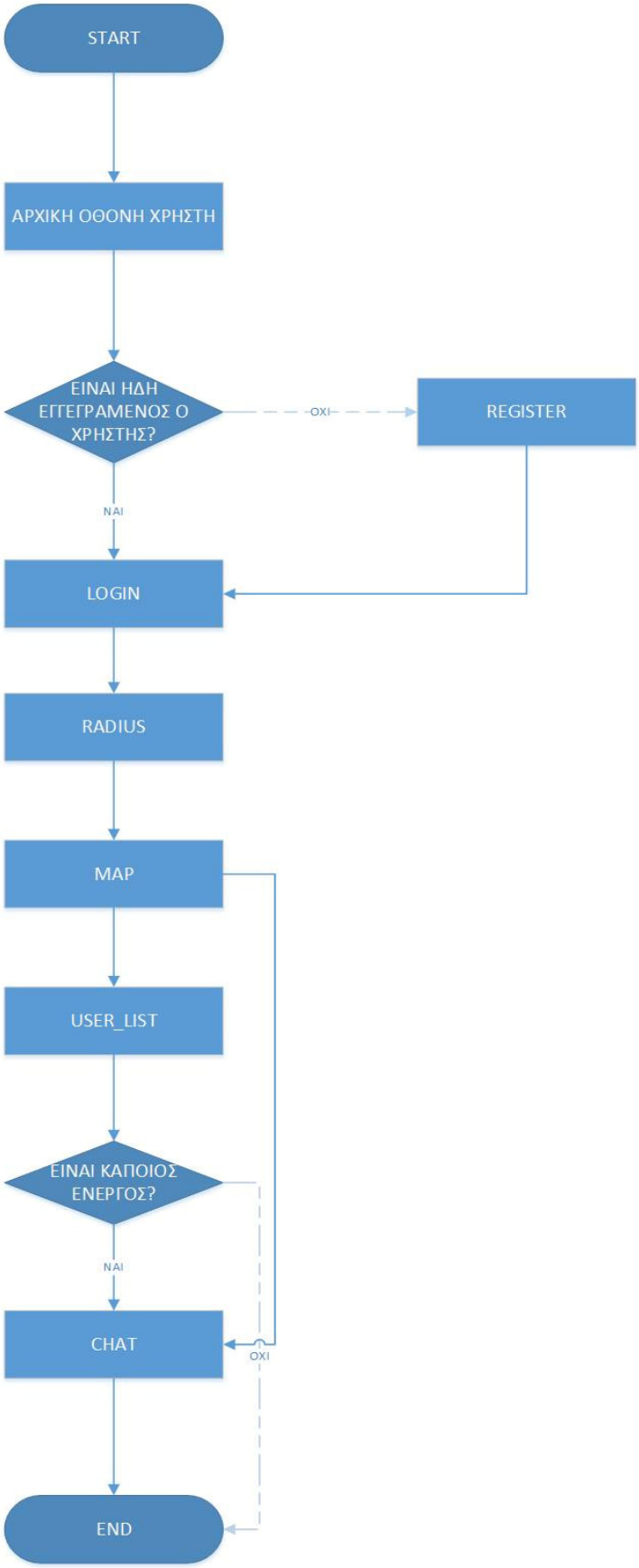
Η φάση της σχεδίασης αποτελεί τη δεύτερη φάση ανάπτυξης της εφαρμογής. Οι γενικοί στόχοι και οι αρχές που καθορίστηκαν στη φάση της ανάλυσης μετατρέπονται σε μια ολοκληρωμένη αναλυτική περιγραφή της εφαρμογής. Σε αυτή τη φάση καθορίζεται και ή αρχιτεκτονική του συστήματος.

Απαιτήσεις σχεδιασμού:

Κατά το σχεδιασμό μιας εφαρμογής υπάρχουν κάποιες απαιτήσεις που θέλουμε να έχει. Τα εργαλεία που χρησιμοποιήθηκαν είναι όλα open source. Για όλους τους χρήστες που θα χρησιμοποιήσουν αυτή την εφαρμογή το μόνο που απαιτείται είναι ένα smartphone με λειτουργικό Android, δυνατότητα εντοπισμού γεωγραφικής θέσηςκαι σύνδεση στο Διαδίκτυο. Για την ανάλυση απαιτήσεων έγινε αρχικά μια έρευνα,όπου μελετήθηκαν και παρόμοια συστήματα που υπαρχουν στο Διαδίκτυο. Εδώ ας υπογραμμίσουμε ότι κατά το σχεδιασμό του λογισμικού προέκυψε η εξής απαίτηση: Το περιβάλλον διεπαφής με τους χρηστες πρέπει να είναι απλό, προσβάσιμο και να μην απαιτεί ιδιαίτερες γνώσεις.

Δομή εφαρμογής:

Η ανάπτυξη της εφαρμογής έγινε με τη γλώσσα προγραμματισμού Java. Για τη διαγραμματική απεικόνιση της δομής της εφαρμογής κατά τη φάση της σχεδίασηςχρησιμοποιήθηκε το εργαλείο λογισμικού DiagramDesignerόπου σχεδιάσαμε το αρχικό διάγραμμα ροής δεδομένων (flowchart) που υπήρχε στο μυαλό μας για τη δομή της εφαρμογής. Για την ευκολότερη κατανόηση της δομής της εφαρμογής βλέπουμε την παρακάτω εικόνα:



Εικόνα 3.2 Διάγραμμα ροής εφαρμογής.

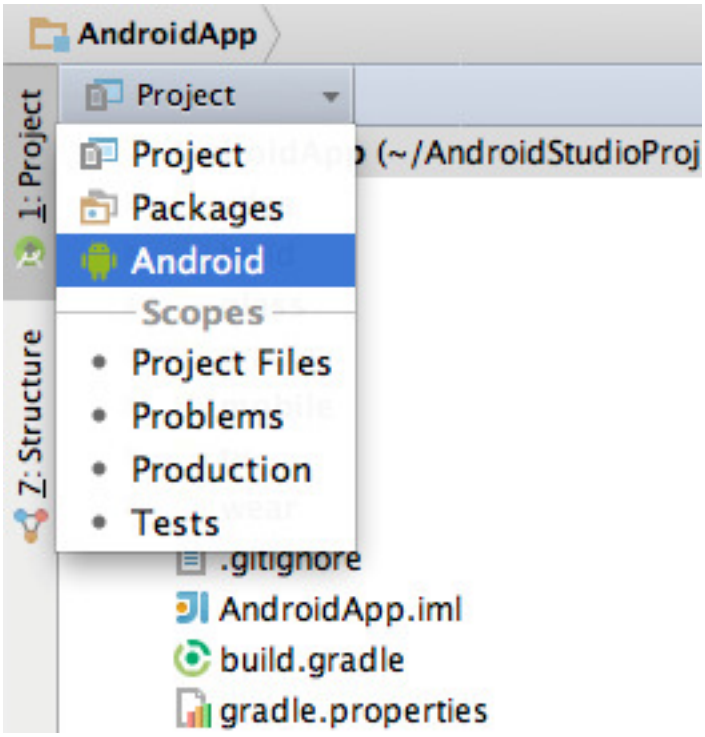
3.1.3 Υλοποίηση

Η φάση της υλοποίησης αποτελεί την τρίτη φάση στην οποία εισαγόμαστε. Στα προηγούμενα κεφάλαια παρουσιάστηκαν η φάση της ανάλυσης και της σχεδίασης του συστήματος της εφαρμογής μας. Στη φάση αυτή θα βασιστούμε στα δεδομένα που προέκυψαν από τις 2 προηγούμενες φάσεις, ώστε να προχωρήσουμε στην υλοποίηση της εφαρμογής. Ο ολοκληρωμένος κώδικας της εφαρμογής παρουσιάζεται στο τέλος της παρούσας πτυχιακής εργασίας.

Πλατφόρμα και εργαλεία ανάπτυξης:

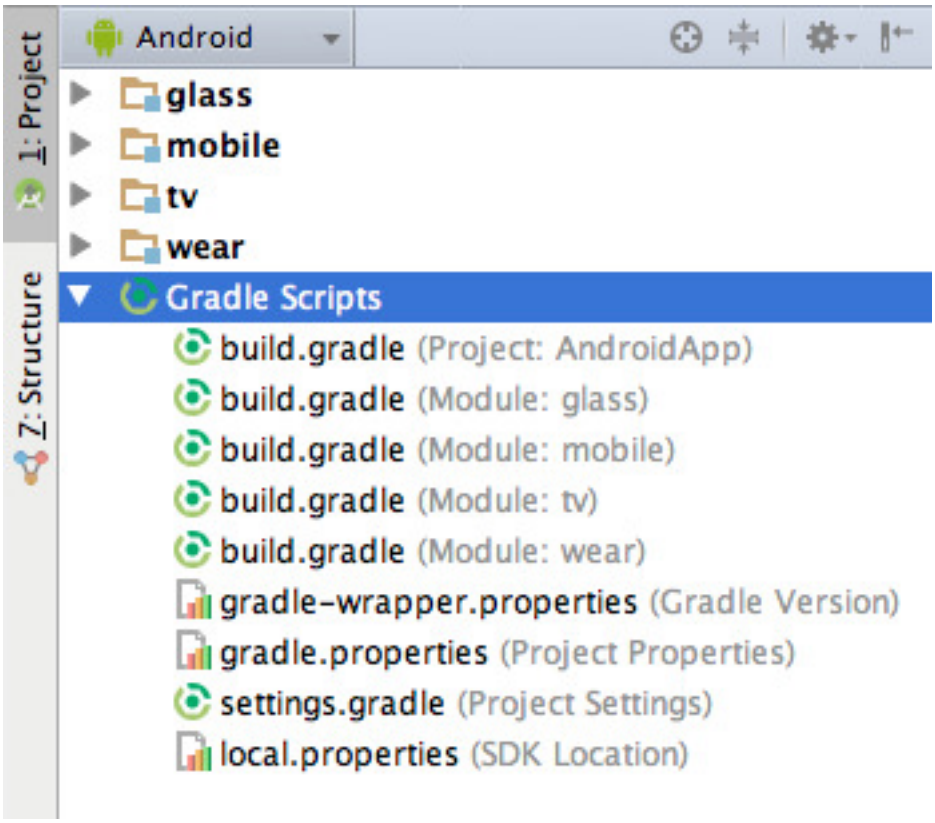
Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε ένας υπολογιστής με λειτουργικό σύστημα Windows 8. Σε αυτόν τον υπολογιστή εγκαταστάθηκε το περιβάλλον ανάπτυξης λογισμικού Android Studio το οποίο περιλαμβάνει ένα ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών βασισμένο στο IntelliJ IDEA.

Από προεπιλογή, το AndroidStudio εμφανίζει τα αρχεία του έργου σας στο Androidprojectview. Μέσω αυτής της προβολής βλέπουμε μια πεπλατυσμένη μορφή της δομής του έργου μας, που παρέχει γρήγορη πρόσβαση στα βασικά αρχεία προέλευσης των έργων Android και μας βοηθά να εργάζομαστε με το Gradle-based build system.



Εικόνα 3.3 Όψη του πρότζεκτ Android.

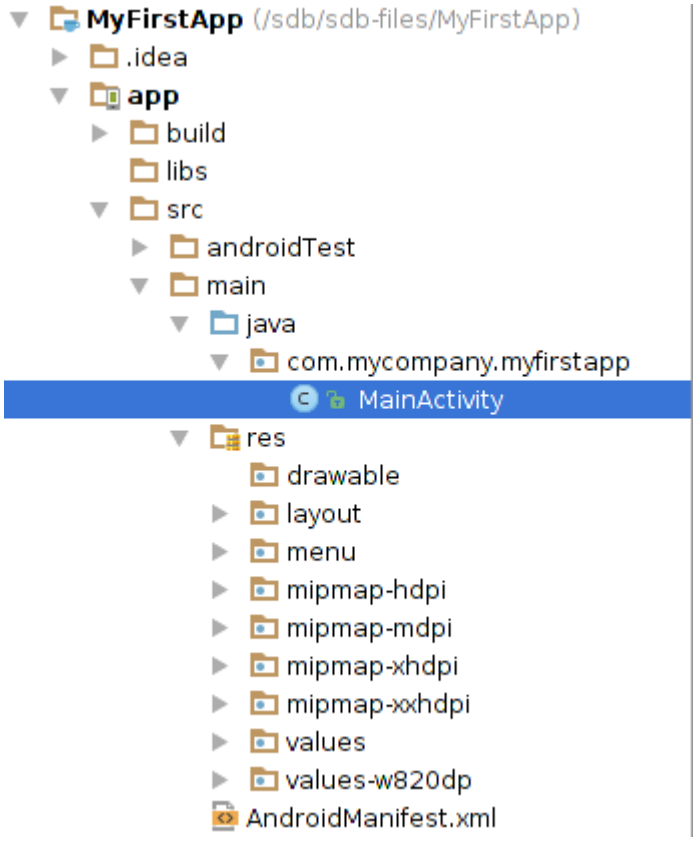
Για να γίνει πιο κατανοητό ας δούμε τις παρακάτω εικόνες:



Εικόνα 3.4 Ταbuild αρχεία του πρότζεκτ

Όταν χρησιμοποιούμε την προβολή του έργου στο Android Studio, θα πρέπει να παρατηρήσουμε ότι η δομή του σχεδίου εμφανίζεται διαφορετική από αυτή που μπορεί να χρησιμοποιηθεί στο Eclipse. Κάθε στιγμιότυπο του Android Studio περιέχει ένα έργο με μία ή περισσότερες ενότητες της εφαρμογής. Κάθε φάκελος της εφαρμογής περιλαμβάνει τα πλήρη sourcesets για την εν λόγω ενότητα, συμπεριλαμβανομένων src /main και / src / androidTest / καταλόγους, πόρους, την κατασκευή και το αρχείο Android manifest. Για το μεγαλύτερο μέρος, θα πρέπει να τροποποιήσουμε τα αρχεία σύμφωνα με τα src / main/ directory για ενημερώσεις κώδικα, το αρχείο gradle.build για τις προδιαγραφές κατασκευής και τα αρχεία κάτω από src / androidTest / directory για testcasecreation.

Ας το δούμε στην παρακάτω εικόνα:



Εικόνα 3.5 Δομή ενός πρότζεκτ.

3.1.4 Έλεγχος

Η φάση αυτή ακολουθεί μετά από τη φάση της υλοποίησης και σε περίπτωση που εντοπιστούν παραλείψεις ή δυσλειτουργίες, επιστρέφουμε στη φάση της υλοποίησης ή ακόμα και σε κάποια προγενέστερη φάση αν κρίνεται απαραίτητο. Σκοπός μας είναι πάντοτε η αποδοτικότερη τελική λειτουργία του συστήματος.

Στη συγκεκριμένη εφαρμογή ο έλεγχος έγινε πρακτικά, χρησιμοποιώντας την εφαρμογή σε διαφορετικές συσκευές(διαφορετικών εταιρειών). Σε όλες τις περιπτώσεις η εφαρμογή λειτούργησε ορθά και δεν παρουσιάστηκαν σφάλματα

3.1.5 Διανομή

Η παρούσα εφαρμογή δε θα διατίθεται κάπου προς πώληση(πχ AndroidMarket). Σχεδιάστηκε και υλοποιήθηκε μόνο στα πλάισια της πτυχιακής εργασίας.

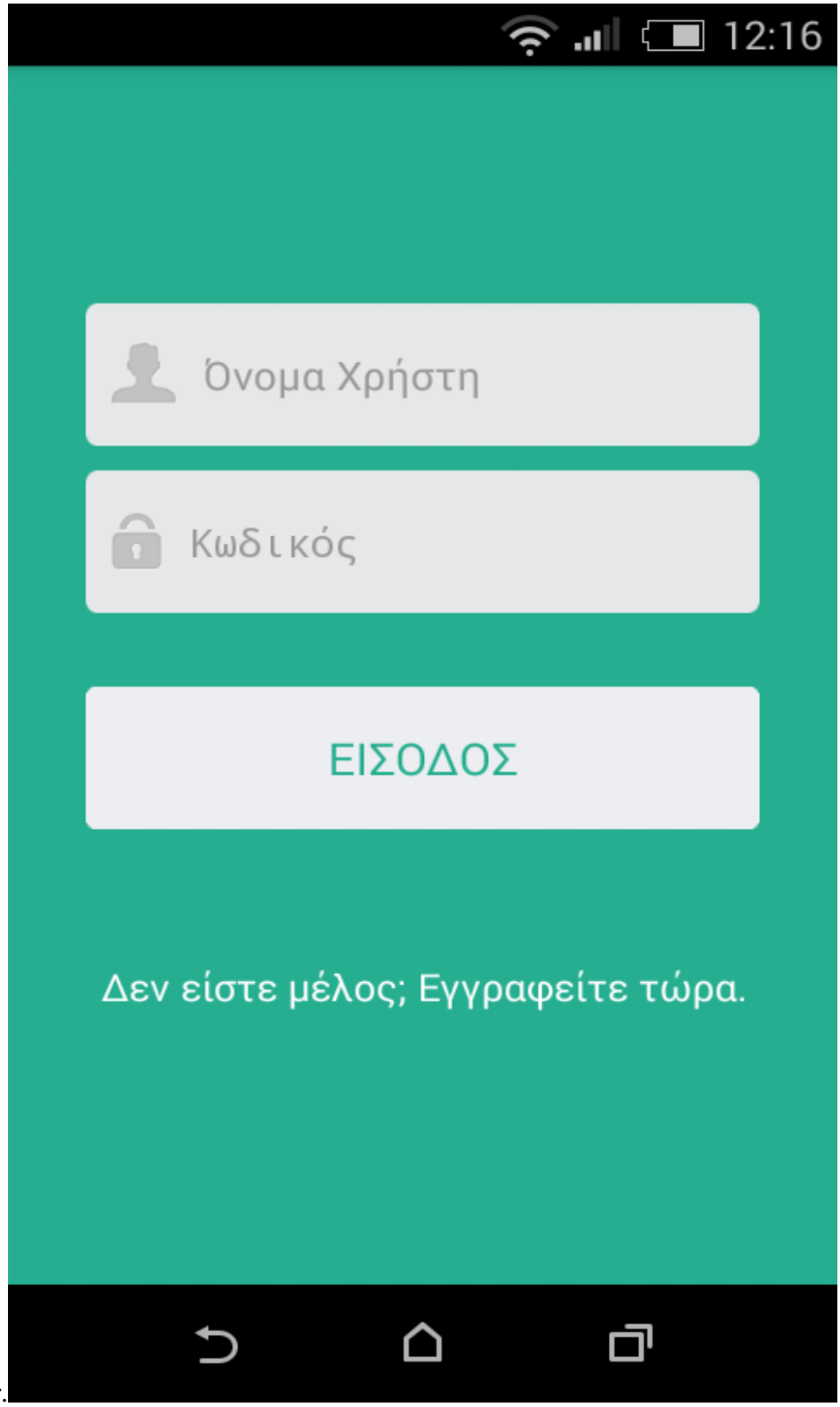
3.2 Λειτουργία της Εφαρμογής

Στην ενότητα αυτή θα δώσουμε μια αναλυτική περιγραφή της εφαρμογής Socializer καθώς και του τρόπου που μπορεί να την αξιοποιήσει ένας χρήστης.Η εφαρμογή αποτελείται από μια αλληλουχία δραστηριοτήτωντις οποίες παρουσιάζουμε παρακάτω.

3.2.1 Η Δραστηριότητα Login

Η δραστηριότητα Login είναι η αρχική οθόνη της εφαρμογής μας. Επιτρέπει σε ένα χρήστη να συνδεθεί με την εφαρμογή χρησιμοποιώντας τα αναγνωριστικά του ή να εγγραφεί σε αυτή αν δεν το έχει κάνει ήδη. Η διεπαφή χρήστη της δραστηριότητας Login φαίνεται στην Εικόνα 3.6. Αν ο χρήστης έχει εγγραφεί ήδη, μπορεί να πληκτρολογήσει το όνομα χρήστη του και τον κωδικό του και να πατήσει το πλήκτρο "ΕΙΣΟΔΟΣ" ώστε να συνδεθεί. Διαφορετικά, μπορεί να πατήσει το "Δεν είστε μέλος; Εγγραφελιτε τώρα." ώστε να δημιουργήσει έναν καινούριο λογαριασμό. Για την εγγραφή των χρηστών έχει χρησιμοποιηθεί η βιβλιοθήκη Parse. Με το πάτημα του κουμπιού "ΕΙΣΟΔΟΣ" η δραστηριότητα αρχικά κρυπτογραφεί τον κωδικό πουκαταχώρησε ο χρήστης με βάση τον αλγόριθμο SHA-512 και στη συνέχεια επικοινωνεί με τον server του Parse για τη συγκεκριμένη εφαρμογή και του στέλνει το e-mail και τον κρυπτογραφημένο κωδικό του χρήστη ώστε να εξετάσει αν τα στοιχεία που έδωσε ο χρήστης αντιστοιχούν πράγματι σε κάποιον εγγεγραμμένο χρήστη. Σε αυτή την περίπτωση, η δραστηριότητα Login ξεκινά την δραστηριότητα Radius. Αν κάποιο από τα στοιχεία είναι λάθος, η Login ενημερώνει το χρήστη και του ζητάει να ξαναδώσει τα αναγνωριστικά του. Αν ο χρήστης επιθυμεί να δημιουργήσει νέο λογαριασμό και πατήσει επομένως στο "Δεν είστε

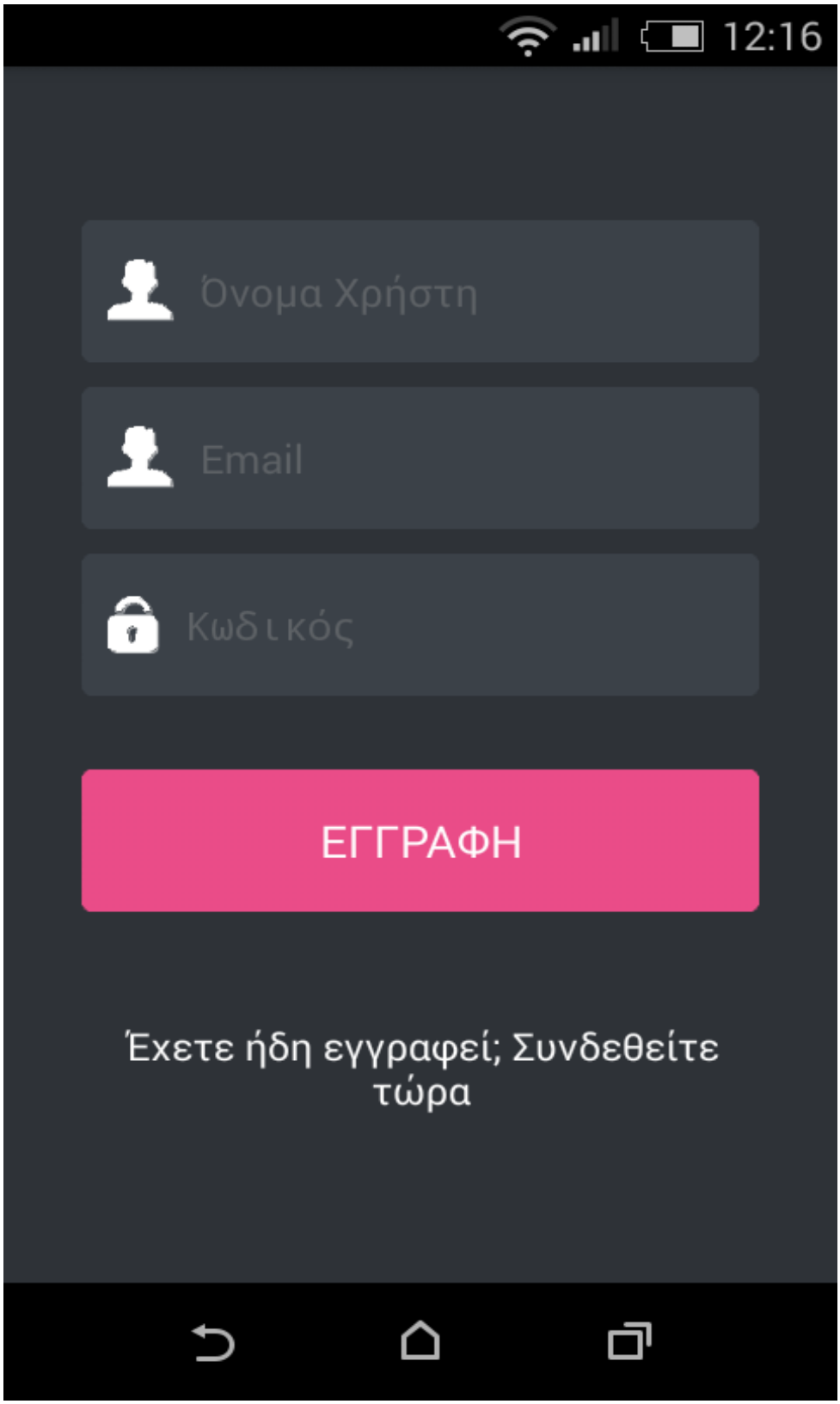
μέλος;Εγγραφελιτε τώρα.", η δραστηριότητα Login ξεκινά την δραστηριότητα Register.



Εικόνα 3.6 Η δραστηριότητα Login.

3.2.2 Η Δραστηριότητα Register

Σε περίπτωση που ο χρήστης θελήσει να εγγραφεί στη βάση δεδομένων της

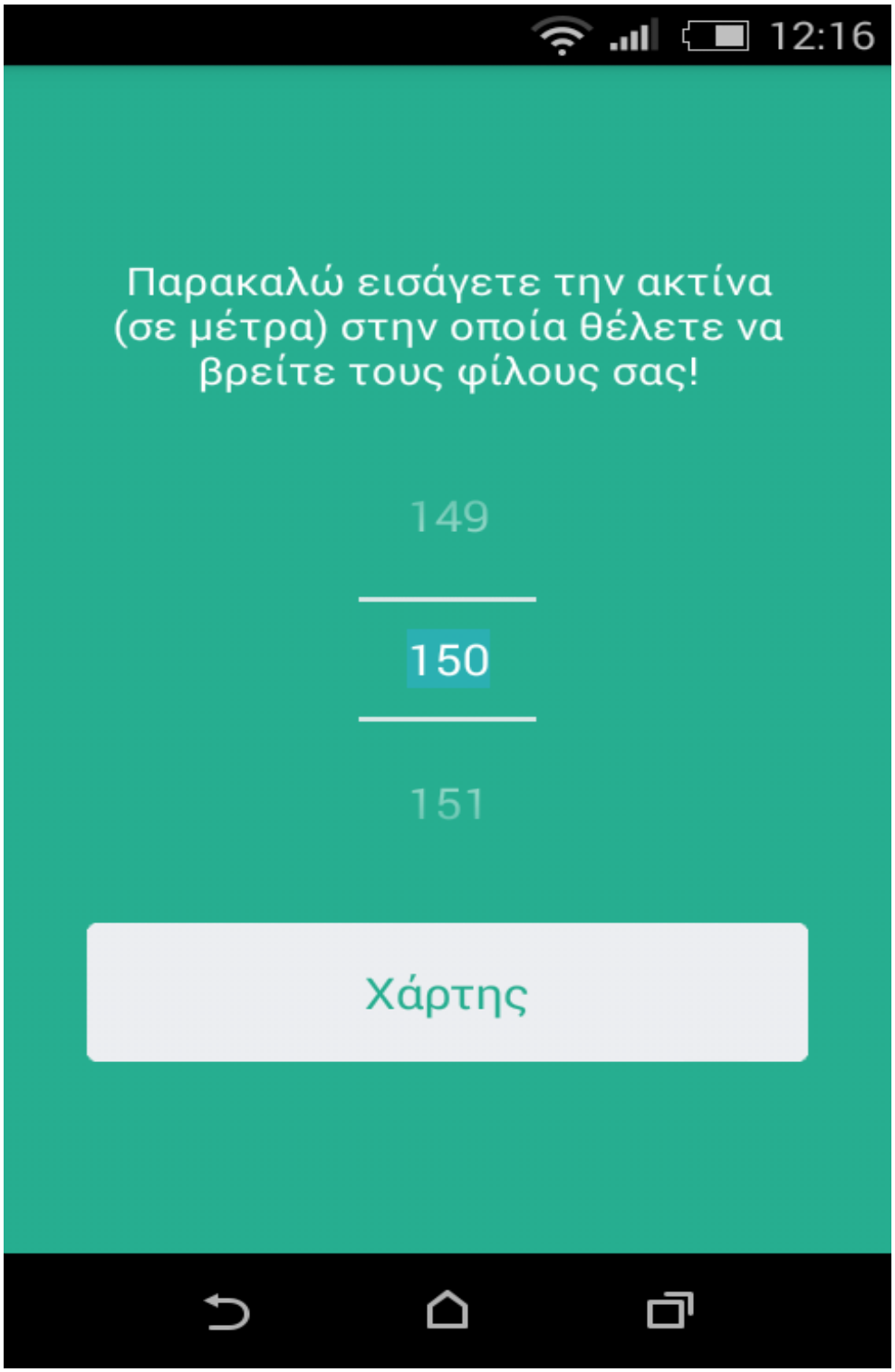


Εικόνα 3.7 Η δραστηριότητα Register.

εφαρμογής, η δραστηριότητα μέσω της οποίας γίνεται αυτό είναι η Register. Στην Εικόνα 3.7 δείχνουμε την διεπαφή χρήστη της δραστηριότητας Register. Όπως και στη δραστηριότητα Login, έτσι και στην Register υπάρχουν δυο σημεία στα οποία μπορεί να πατήσει ο χρήστης. Το ένα είναι το πλήκτρο "ΕΓΓΡΑΦΗ" και το άλλο είναι το "Έχτετε ήδη εγγραφεί; Συνδεθείτε τώρα". Πατώντας στο δεύτερο, η δραστηριότητα ξεκινά μια δραστηριότητα Login που θα επιτρέψει στο χρήστη να συνδεθεί στην εφαρμογή. Πατώντας στο πρώτο πλήκτρο, αφού έχουμε συμπληρώσει τα πεδία για όνομα χρήστη, e-mail και κωδικό, η εφαρμογή συνδέεται με το server του Parse για την εφαρμογή και καταχωρεί τον χρήστη στέλνωντας το περιεχόμενο των τριών παραπάνω πεδίων. Και σε αυτή την περίπτωση, ο κωδικός που καταχωρεί ο χρήστης κρυπρογραφείται με τον αλγόριθμο SHA-512. Σε περίπτωση που κάποιο πρόβλημα προκύψει, όπως για παράδειγμα υπάρχων e-mail, ο χρήστης ενημερώνεται ότι η εγγραφή δεν ήταν επιτυχής.

3.2.3 Η Δραστηριότητα Radius

Όπως αναφέρθηκε προηγουμένως, σε περίπτωση που ο χρήστης συνδεθεί επιτυχώς, η δραστηριότητα Login ξεκινά τη δραστηριότητα Radius. Η δραστηριότητα αυτή επιτρέπει στο χρήστη να καθορίσει μέσω ενός κυλιόμενου επιλογέα αριθμών την ακτίνα μέσα στην οποία επιθυμεί να εντοπίσει η εφαρμογή άλλους χρήστες. Η προκαθορισμένη τιμή είναι τα 150 μέτρα, ενώ οι τιμές κυμαίνονται από 0 έως 5 χιλιόμετρα. Η διεπαφή χρήστη της δραστηριότητας φαίνεται στην Εικόνα 3.3. Αν ο χρήστης θέσει την τιμή 300, τότε η εφαρμογή θα επιστρέψει όλους τους χρήστες οι οποίοι βρίσκονται σε απόσταση μικρότερη από 300 μέτρα σε σχέση με τη θέση του χρήστη. Όταν ο χρήστης καταχωρίσει την επιλογή του, μπορεί να πατήσει το πλήκτρο"Χάρτης" με το οποίο ξεκινά η δραστηριότητα Map.

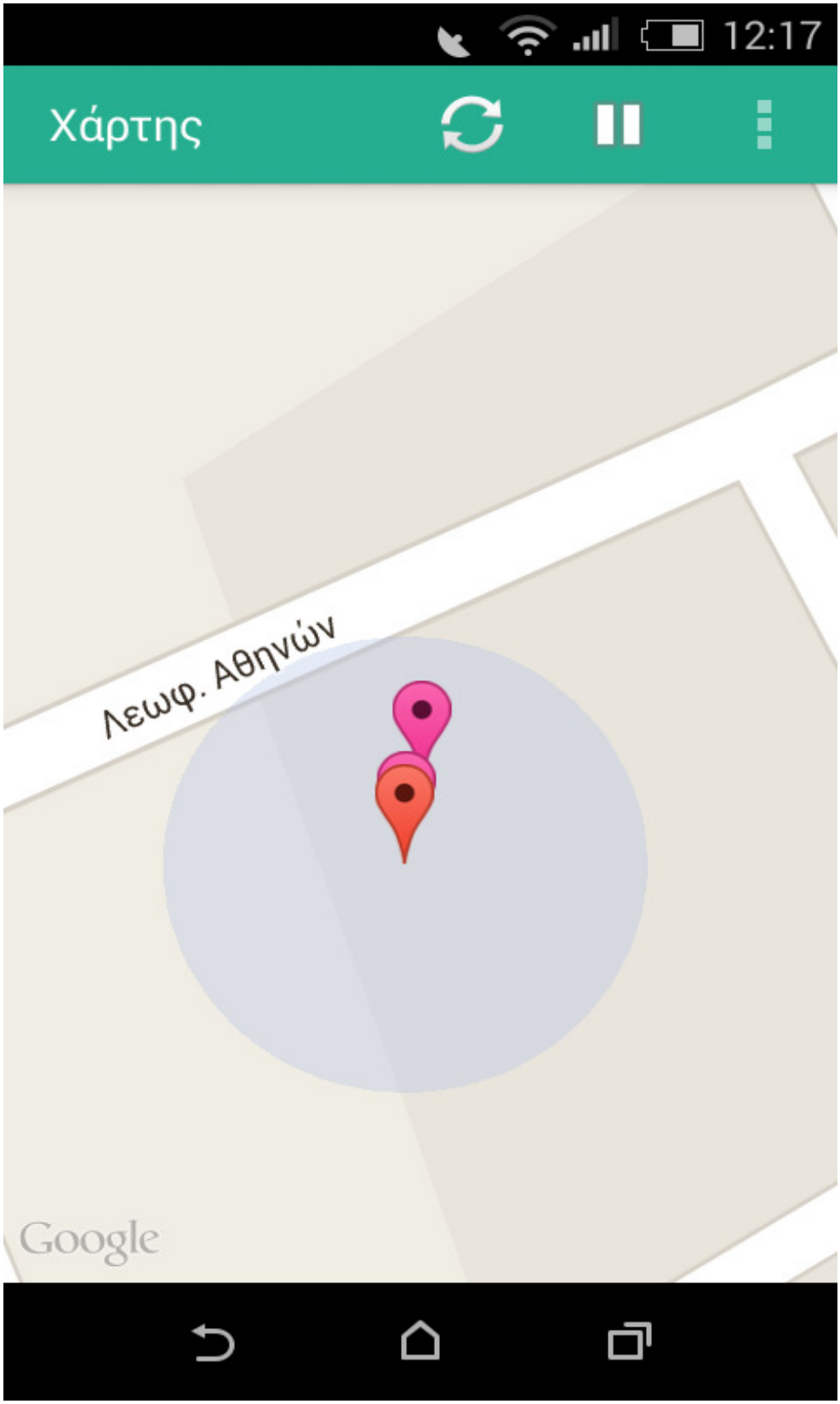


Εικόνα 3.8 Η δραστηριότητα Radius.

3.2.4 Η Δραστηριότητα Map

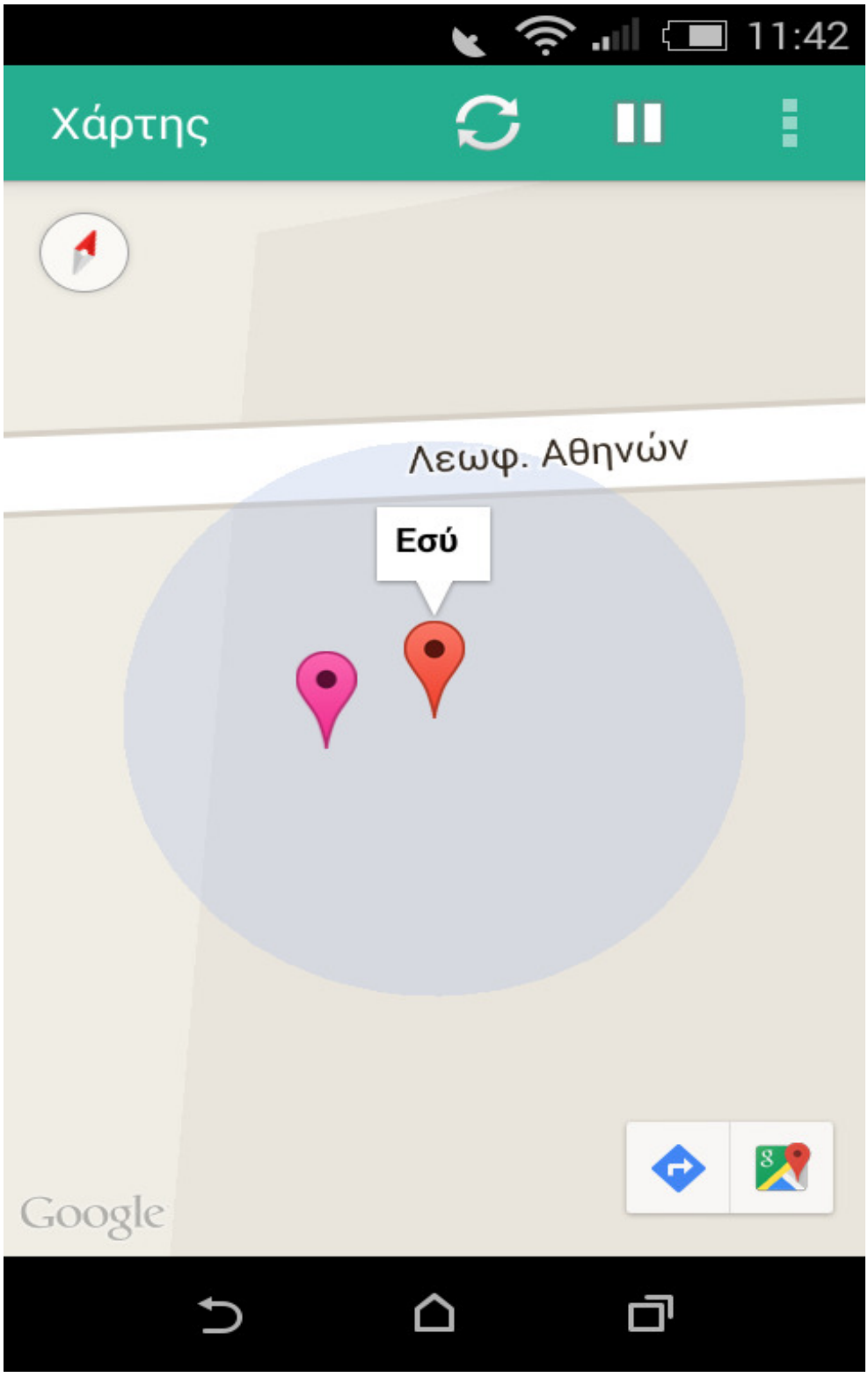
Η δραστηριότητα Map χρησιμοποιεί την υπηρεσία τοποθεσίας ώστε να εντοπίσει τις ακριβείς γεωγραφικές συντεταγμένες στις οποίες βρίσκεται ο χρήστης και δημιουργεί τον χάρτη και τοποθετεί ένα σημάδι στο σημείο που βρίσκεται ο χρήστης. Επίσης,

εμφανίζει με μπλέ αδιαφανές χρώμα την ακτίνα την οποία έχει θέσει ο χρήστης. Επιπλέον, επικοινωνεί με τον server τον οποίο θα περιγράψουμε αργότερα και στον



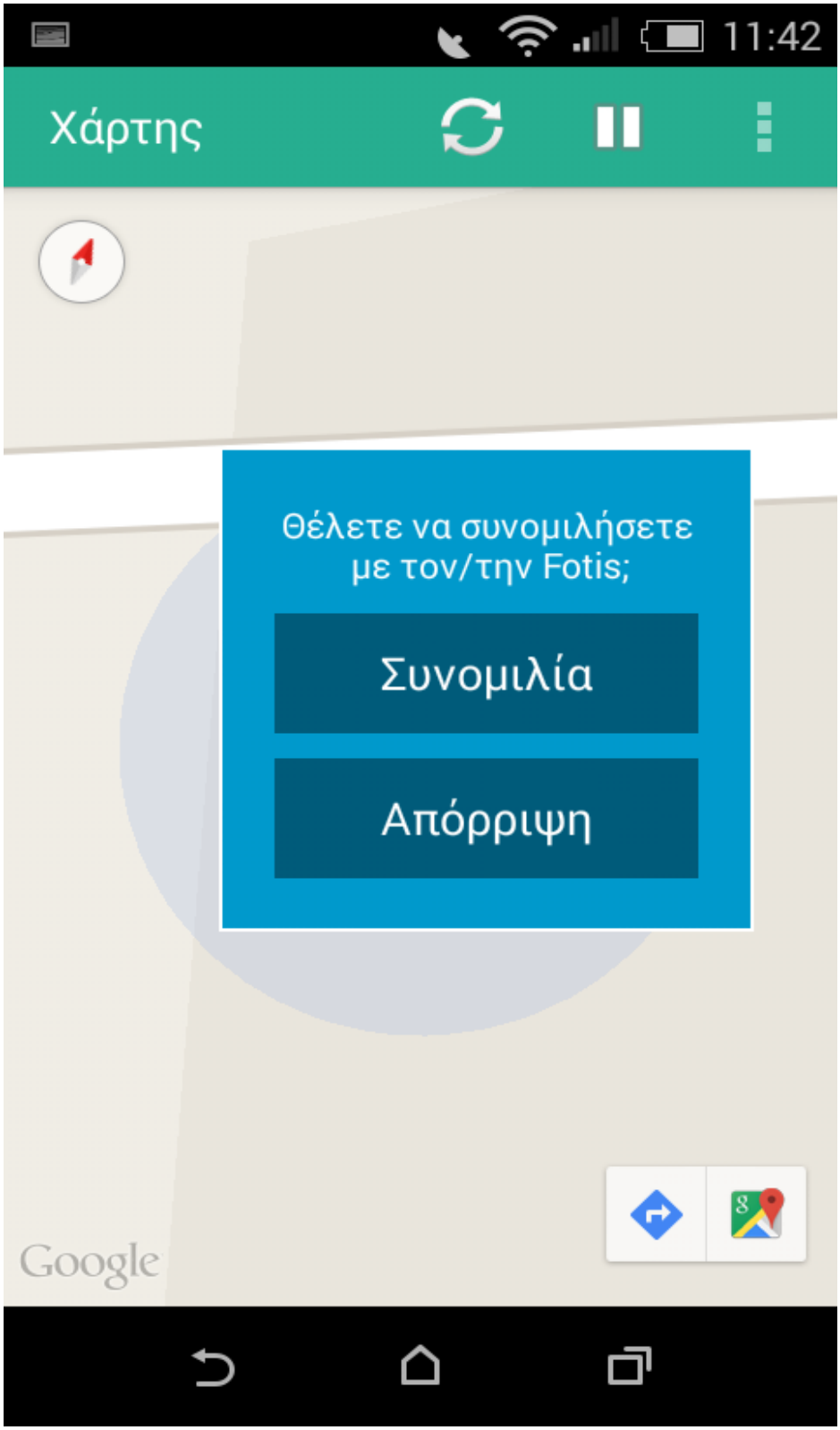
Εικόνα 3.9 Η δραστηριότητα Map.

οποίο αποθηκεύουμε τις τελευταίες γεωγραφικές συντεταγμένες κάθε χρήστη και στέλνει τις γεωγραφικές συντεταγμένες του μαζί με το όνομα



Εικόνα 3.10 Η δραστηριότητα Map.

χρήστη του. Ο server στέλνει πίσω στον χρήστη τις συντεταγμένες όλων των χρηστών και στη συνέχεια, η εφαρμογή υπολογίζει ποιοί από αυτούς τους χρήστες



Εικόνα 3.11 Η δραστηριότητα Map.

βρίσκονται σε απόσταση από το χρήστη μικρότερη από την ακτίνα που έθεσε ο χρήστης. Για αυτούς τους χρήστες η εφαρμογή δημιουργεί επίσης σημάδια διαφορετικού χρώματος και τα τοποθετεί στις αντίστοιχες θέσεις στο χάρτη. Ένα

παράδειγμα του χάρτη που εμφανίζεται και στον οποίο εκτός του χρήστη υπάρχουν και δυο επιπλέον χρήστες φαίνεται στην Εικόνα 3.9.Αν πατήσουμε πάνω στο εικονίδιο που εμφανίζει τον χρήστη της εφαρμογής εμφανίζεται το μήνυμα "Εσύ" (Εικόνα 3.10). Αν αντίθετα πατήσουμε στο εικονίδιο κάποιου άλλου χρήστη, εμφανίζεται ένα παράθυρο, το οποίο φαίνεται στην Εικόνα 3.11 και το οποίο μας δίνει τη δυνατότητα είτε να συνομιλήσουμε με το χρήστη αυτό ή να τον απορρίψουμε. Σε περίπτωση που πατήσουμε στο κουμπί "Απόρριψη" επιστρέφουμε στην προηγούμενη οθόνη, ενώ σε περίπτωση που πατήσουμε στην επιλογή ¨Συνομιλία¨ ξεκινά μια δραστηριότητα Chat η οποία επιτρέπει τη συνομιλία μεταξύ δυο χρηστών. Επιπλέον, στην πάνω μπάρα υπάρχουν διαθέσιμα ένα πλήκτρο Refresh το οποίο ανανεώνει τις γεωγραφικές συντεταγμένες του χρήστη καθώς και τη θέση του στο χάρτη και ένα πλήκτρο Pause που σταματάει τις αναβαθμίσεις τοποθεσίας των χρηστών και το οποίο αντικαθίσταται μόλις πατηθεί από ένα πλήκτρο Play το οποίο τις ξαναξεκινά. Υπάρχει επίσης μια επιλογή για την εμφάνιση όλων των εγγεγραμμένων στην εφαρμογή χρηστών η οποία αφού πατηθεί ξεκινά μια δραστηριότητα UserList.

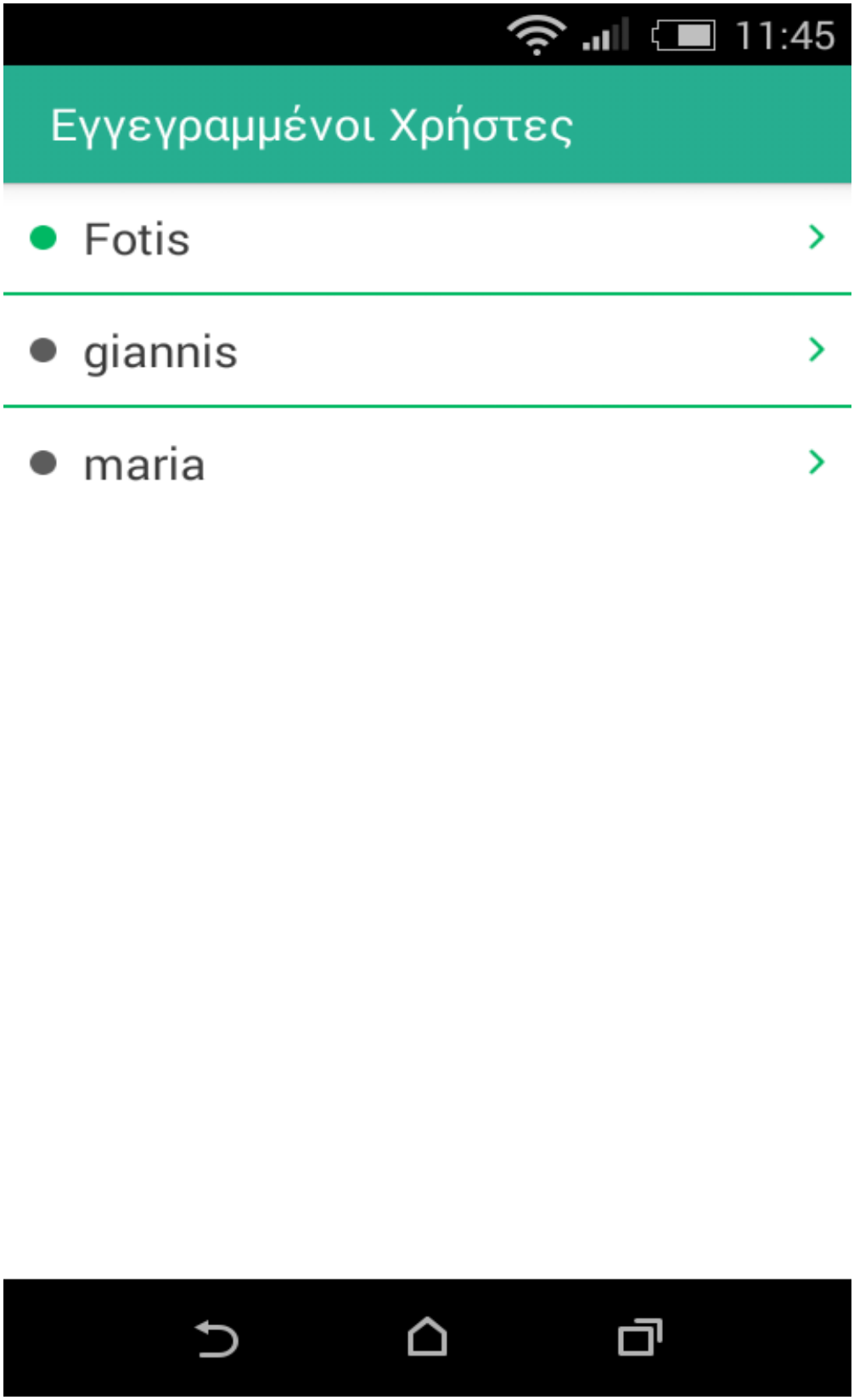
3.2.5 Η Δραστηριότητα UserList

Όπως αναφέρθηκε παραπάνω, η δραστηριότητα UserList εμφανίζει μια λίστα με όλους τους εγγεγραμμένους στην εφαρμογή χρήστες από όπου μπορούμε να επιλέξουμε κάποιον χρήστη ώστε να επικονωνήσουμε μαζί του. Για κάθε χρήστη υπάρχει ένδειξη αν είναι ενεργός ή όχι μέσω ενός κύκλου ο οποίος είναι φωτισμένος με μπλέ χρώμα σε περίπτωση που ο χρήστης είναι ενεργός. Η διεπαφή χρήστη της δραστηριότητας UserList φαίνεται στην Εικόνα 3.12. Αν επιλέξουμε έναν χρήστη ώστε να συνομιλήσουμε μαζί του, η εφαρμογή ξεκινά μια δραστηριότητα Chat η οποία δημιουργεί την απαραίτητη διεπαφή μέσω της οποίας οι δυο χρήστες μπορούν να συνομιλήσουν μεταξύ τους.

3.2.6 Η Δραστηριότητα Chat

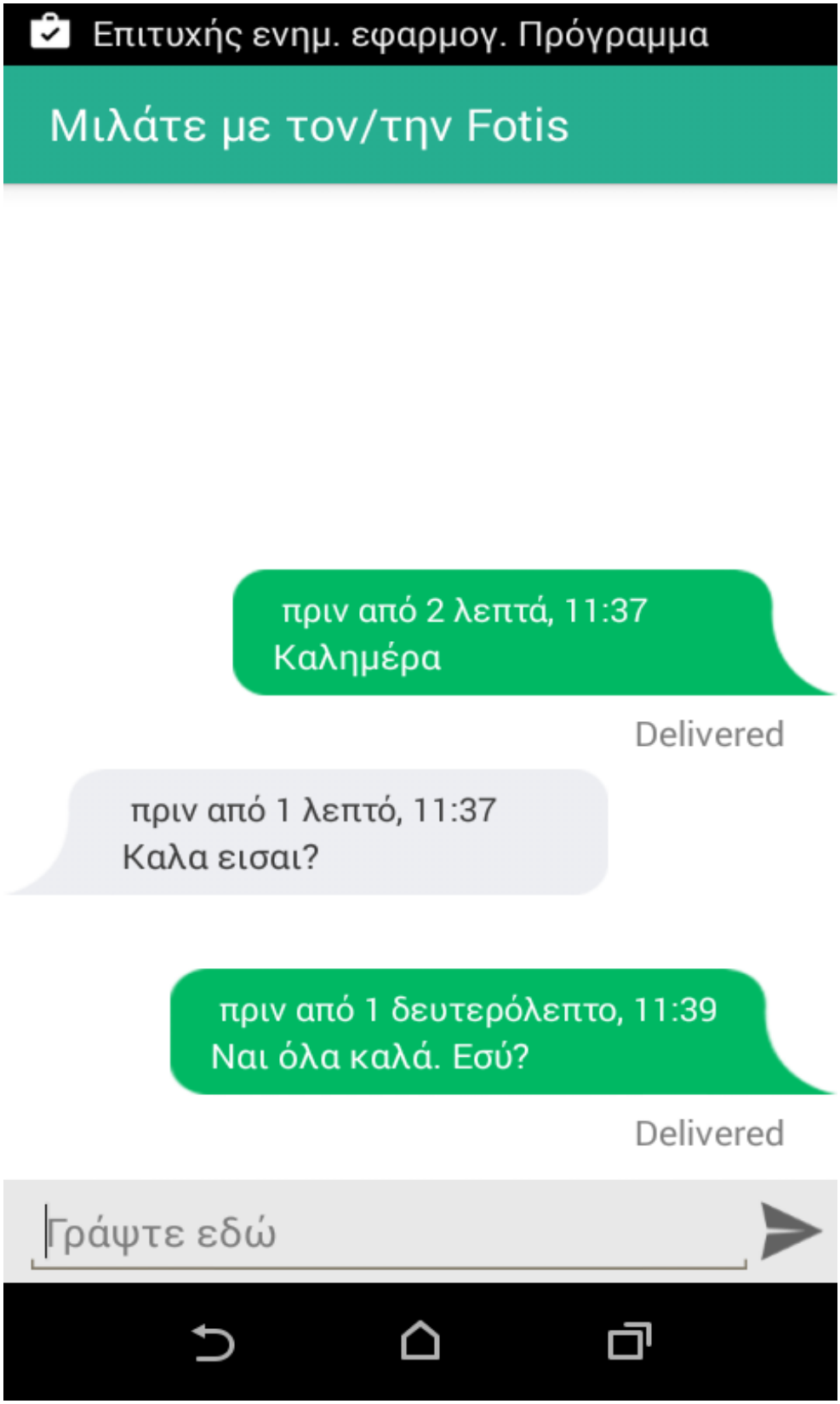
Η δραστηριότητα Chat επιτρέπει σε δυο χρήστες να μιλήσουν μεταξύ τους. Η διεπαφή χρήστη της δραστηριότητας Chat φαίνεται στην Εικόνα 3.13. Είναι αντίστοιχη με αυτές που παρέχουν οι δημοφιλείς εφαρμογές κοινωνικής δικτύωσης. Μπορούμε να δούμε το όνομα του χρήστη με τον οποίο συνομιλούμε, τα μηνύματα

που στείλαμε, την ώρα που τα στείλαμε, αν τα έλαβε ο συνομιλητής καθώς και τα



Εικόνα 3.12 Η δραστηριότητα UserList.

μηνύματα που λάβαμε και την ώρα που τα λάβαμε. Η δραστηριότητα Chat



Εικόνα 3.13 Η δραστηριότητα Chat.

χρησιμοποιεί για την ανταλλαγή μηνυμάτων μεταξύ των χρηστών διεπαφές της βιβλιοθήκης Parse.

3.2.7 Άλλες Κλάσεις της Εφαρμογής

Η εφαρμογή χρησιμοποιεί μια σειρά από άλλες κλάσεις για να υλοποιήσει τις διάφορες λειτουργίες της. Οι κλάσεις TouchEffect, TouchEffect2 και Toucheffect3 δημιουργούν το εφέ που βλέπουμε όταν πατάμε κάποια πλήκτρα σε όλες τις δραστηριότητες εκτός της Map. Συγκεκριμένα, όλες αυτές οι κλάσεις κληρονομούν την κλάση Custom η οποία καθορίζει την εκτέλεση των παραπάνω εφφέ. Η κλάση ChatApp συνδέει την εφαρμογή με το αντίστοιχο πρότζεκτ το οποίο έχει δημιουργηθεί στο Parse. Η κλάση SHA512 χρησιμοποιείται για να κρυπτογραφήσει ένα αλφαριθμητικό με βάση τον αλγόριθμο SHA-512. Η κλάση Conversation αντιπροσωπεύει ένα μήνυμα και περιέχει το κείμενο του μηνύματος, την ημερομηνία και ώρα που στάλθηκε και το όνομα του χρήστη που το έστειλε. Η κλάση User αντιπροσωπεύει έναν χρήστη και περιέχει το όνομά του και το σημάδι στο χάρτη που αντιστοιχεί στο συγκεκριμένο χρήστη. Η διεπαφή Config περιέχει κάποιες παραμέτρους της εφαρμογής όπως την διεύθυνση στην οποία βρίσκεται ο server στον οποίο αποθηκεύονται οι γεωγραφικές συντεταγμένες των χρηστών. Τέλος, η κλάση NumberPicker βοηθάει στη δημιουργία της διεπαφής χρήστη της δραστηριότητας Radius στην οποία επιλέγουμε μέσω ενός κυλιόμενου παραθύρου την επιθυμητή ακτίνα.

3.2.8 Βάση Δεδομένων Locations

Η εφαρμογή χρησιμοποιεί μια βάση δεδομένων στην οποία αποθηκεύει το όνομα χρήστη και το γεωγραφικό μήκος και πλάτος στο οποίο βρέθηκε τελευταία φορά ο χρήστης. Συνεπώς, απαιτεί πρόσβαση, εισαγωγή, ενημέρωση και αποθήκευση δεδομένων. Χρησιμοποιήσαμε ένα δωρεάν, ανοικτού λογισμικού σύστημα διαχείρισης βάσεων δεδομένων, την MySQL. Η MySQL είναι γρήγορη, εξαιρετικά αξιόπιστη και χρησιμοποιείται από πολλές εταιρείες και οργανισμούς σε όλο τον κόσμο, ενώ όπως και η Java, έτσι και η MySQL αναπτύσσεται, διανέμεται και υποστηρίζεται από την Oracle. Για να κάνουμε ερωτήματα στη βάση δεδομένων χρησιμοποιούμε τη γλώσσα SQL. Η SQL είναι η πιο κοινή τυποποιημένη γλώσσα που χρησιμοποιείται για πρόσβαση σε βάσεις δεδομένων. Ξεκίνησε να αναπτύσσεται από την IBM στις αρχές της δεκαετίας του 1970. Αν και από τότε έχει τροποποιηθεί

σημαντικά, ο σκοπός της ως μια γλώσσα διαχείρισης βάσεων δεδομένων παραμένει ο ίδιος. Η SQL υποστηρίζεται από σχεδόν όλα τα συστήματα διαχείρισης βάσεων δεδομένων καθώς δεν είναι αποκλειστική ιδιοκτησία κάποιας εταιρείας, πράγμα που την κάνει ιδιαίτερα προσφιλή.

3.2.9 Σύνδεση της Εφαρμογής με τη Βάση Δεδομένων

Για τη σύνδεση της εφαρμογής με τη βάση δεδομένων locations χρησιμοποιήσαμε scripts γραμμένα σε γλώσσα php. Η γλώσσα προγραμματισμού php σχεδιάστηκε με στόχο την ανάπτυξη και τη διαχείριση ιστοσελίδων. Χρησιμοποιείται όμως πλέον και ως μια γενική γλώσσα προγραμματισμού. Στην εφαρμογή μας η php χρησιμοποιείται για να παρέχει πρόσβαση στα δεδομένα της βάσης του MySQL server. Αυτό επιτυγχάνεται ενσωματώνοντας εντολές SQL σε κώδικα php. Έτσι, μέρος του περιεχομένου της εφαρμογής παράγεται από τις πληροφορίες που εξάγουμε από την βάση δεδομένων. Πιο συγκεκριμένα, για να αποκτήσουμε πρόσβαση στο περιεχόμενο της βάσης δεδομένων μας πρέπει να δημιουργήσουμε μια σύνδεση στην MySQL μέσω ενός php script. Χρησιμοποιούμε php scripts για να καταχωρήσουμε στη βάση έναν νέο χρήστη και τις γωγραφικές του συντεταγμένες όταν συνδέεται για πρώτη φορά, να τροποποιήσουμε τις συντεταγμένες ενός εγγεγραμμένου χρήστη όταν ανανεώσει την τοποθεσία στην οποία βρίσκεται και για να λάβουμε τις γεωγραφικές θέσεις όλων των άλλων χρηστών.

4

Συμπεράσματα και Μελλοντικές Επεκτάσεις

Στη συγκεκριμένη εργασία αναπτύχθηκε μια εφαρμογή για έξυπνα κινητά τηλέφωνα που τρέχουν λογισμικό Android. Η εφαρμογή δίνει τη δυνατότητα στους χρήστες να βρούν άλλους χρήστες της εφαρμογής οι οποίοι βρίσκονται γεωγραφικά κοντά τους και να μιλήσουν μαζί τους εάν το επιθυμούν. Βασικό δομικό στοιχείο της εφαρμογής είναι η τεχνολογία GPS η οποία επιτρέπει τον ακριβή καθορισμό της γεωγραφικής θέσης του χρήστη.

Μέσα από την ανάπτυξη της παραπάνω εφαρμογής αποκτήθηκαν σημαντικές γνώσεις. Συγκεκριμένα, εξερευνήσαμε σε μεγάλο βάθος την πλατφόρμα Android και εξοικειωθήκαμε μαζί της. Παράλληλα, κατανοήσαμε σε μεγάλο βαθμό τη γλώσσα προγραμματισμού Java η οποία χρησιμοποιείται για την ανάπτυξη Android εφαρμογών. Επίσης, αποκτήθηκαν σημαντικές γνώσεις για τις τεχνολογίες των έξυπνων κινητών τηλεφώνων και τον τρόπο ανάπτυξης εφαρμογών σε αυτά. Ωστόσο, σημαντικότερη από όλα είναι η ευχαρίστηση και η ικανοποίηση που νιώσαμε για την ανάπτυξη μιας λειτουργικής εφαρμογής που δεν έχει να ζηλέψει πολλά από άλλες εμπορικές εφαρμογές, πράγμα που μας έδειξε ότι αρκεί να υπάρχει η κατάλληλη διάθεση και ιδέα ώστε να δημιουργήσουμε κάτι πρωτότυπο.Συνολικά, παρόλες τις δυσκολίες, καταφέραμε να φέρουμε εις πέρας την εργασία και να αποκτήσουμε στην πορεία πολλά εφόδια τα οποία αδιαμφισβήτητα θα αξιοποιήσουμε στο μέλλον.

Όσον αφορά μελλοντικές επεκτάσεις της εφαρμογής, είναι αλήθεια ότι θα μπορούσαμε να τη βελτιώνουμε συνεχώς προσθέτοντας καινούρια χαρακτηριστικά κάθε φορά. Κάποια από αυτά είναι η μετάφραση της διεπαφής χρήστη της εφαρμογής στις πιο δημοφιλείς ξένες γλώσσες, ώστε να στοχεύσουμε και σε χρήστες εκτός

Ελλάδας, η δυνατότητα σύνδεσης μέσω λογαριασμού άλλης εφαρμογής κοινωνικής δικτύωσης όπως το Facebook, και η δυνατότητα μπλοκαρίσματος ανεπιθύμητων χρηστών.

Αναφορές

[1] B. Molina and M. Cava. "Apple beats Samsung in Q4 smartphone sales". *USA TODAY*

[2] "Android device activation numbers reach 1 billion worldwide". Phandroid.com, 3 Σεπτεμβρίου 2013

[3] Darcey L. and Conder S. "*Sams Teach Yourself Android Application Development in 24Hours*". Pearson Education, 2011

[4] "Activity", http://developer.android.com/reference/android/app/Activity.html

[5] Darcey L. and Shane C., "*Ανάπτυξη Εφαρμογών με το Android-Δεύτερη Έκδοση*", Μ.Γκιούρδας, Αθήνα, 2011

[6] Meier R. "*Professional Android 4 application development*". John Wiley & Sons, 2012

[7] Δημητριάδης Σ., Πομπόρτσης, Α., Τριανταφύλλου Ε., *'"Τεχνολογία πολυμέσων Θεωρια και Πραξη"*, Τζιόλα, Θεσσαλονίκη, 2004

[8] Deitel P. and Deitel H. "*Java How to program"*.Prentice Hall Press, 2011.

[9] Steele J. and To N. "*The Android Developer's Cookbook: Building Applications with the Android SDK"*. Pearson Education, 2010.

[10] Murphy M. "*Android Programming Tutorials"*.CommonsWare, 2010.

[11] Lee W. "*Beginning Android 4 Application Development"*. John Wiley & Sons, 2012.

[12] "Global Positioning System", Wikipedia, https://en.wikipedia.org/wiki/Global_Positioning_System

[13] "Android Studio", https://developer.android.com/intl/zh-cn/sdk/index.html

[14] "Android", https://www.android.com/

[15] "Smartphone", Wikipedia, https://en.wikipedia.org/wiki/Smartphone

[16] "Waterfall model", Wikipedia, https://en.wikipedia.org/wiki/Waterfall_model

Παράρτημα – Κώδικας Εφαρμογής

ChatApp.java

```
publicclass ChattApp extends Application {

        @Override
        publicvoid onCreate() {
            super.onCreate();
            Parse.initialize(this, "16lXwJMrLYYWTsRUUYCYTbFOGWNdbrXKBsxo4ogH",
"o72MwKoKlNBXMzuyj2ak7WM1ndFVTQMaH7e7gpFZ");
        }

}
```

Login.java

```
publicclass Login extends Custom {

private String inputUsername, inputPassword;
private EditText userNameEditText;
private EditText passwordEditText;

    @Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.login);
setTouchNClick(R.id.btnLogin);
setTouchNClick2(R.id.btnLinkToRegisterScreen);
userNameEditText = (EditText) findViewById(R.id.username);
passwordEditText = (EditText) findViewById(R.id.password);
    }

publicvoid showAlertDialog(Context context, String title, String message, Boolean status) {
        AlertDialog alertDialog = newAlertDialog.Builder(context).create();
alertDialog.setTitle(title);
alertDialog.setMessage(message);
if (status != null)
alertDialog.setIcon((status) ? R.drawable.success : R.drawable.fail);
alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
publicvoid onClick(DialogInterface dialog, int which) {}
        });
alertDialog.show();
    }

publicboolean isConnectingToInternet() {
        ConnectivityManager connectivity = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
if (connectivity != null) {
NetworkInfo[] info = connectivity.getAllNetworkInfo();
if (info != null)
for (NetworkInfo anInfo : info)
if (anInfo.getState() == NetworkInfo.State.CONNECTED)
returntrue;
        }
returnfalse;
    }

    @Override
publicvoid onClick(View v)
    {
super.onClick(v);
if ( v.getId() == R.id.btnLogin ) {
if ( !isConnectingToInternet() ) {
showAlertDialog(this, "Δεν υπάρχει σύνδεση στο internet", "Παρακαλώ συνδεθείτε στο internet", false);
                }
```

```
                else {
inputUsername = userNameEditText.getText().toString();
inputPassword = passwordEditText.getText().toString();
if (inputUsername.trim().length() >0&& inputPassword.trim().length() >0) {
try {
checkLogin(inputUsername, SHA512.sha512(inputPassword));
                    }
catch ( NoSuchAlgorithmException | UnsupportedEncodingException e ) {
e.printStackTrace();
                    }
                }
else {
Toast.makeText(getApplicationContext(), "Παρακαλώ εισάγετε τα στοιχεία σύνδεσης!",
Toast.LENGTH_LONG).show();
                }
            }
        }
else {
startActivity(new Intent(this, Register.class));
finish();
        }
    }

privatevoid checkLogin(final String username, final String password) {
final ProgressDialog progressDialog = ProgressDialog.show(this, null, getString(R.string.pistopoiisi));
ParseUser.logInInBackground(username, password, new LogInCallback() {
        @Override
publicvoid done(ParseUser pu, ParseException e) {
progressDialog.dismiss();
if (pu != null) {
                    UserList.username = username;
                    Chat.user = pu;
startActivity(new Intent(Login.this, Radius.class).putExtra("username", inputUsername));
finish();
                } else {
Toast.makeText(getApplicationContext(), "Τα στοιχεία σύνδεσης είναι λαθεμένα!", Toast.LENGTH_LONG).show();
                }
            }
        });
    }

}
```

Register.java

```
publicclass Register extends Custom {

        private EditText fullName;
        private EditText email;
        private EditText password;

    @Override
        publicvoid onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
setContentView(R.layout.register);
setTouchNClick(R.id.btnRegister);
setTouchNClick3(R.id.btnLinkToLoginScreen);
            fullName = (EditText) findViewById(R.id.name);
            email = (EditText) findViewById(R.id.email);
password = (EditText) findViewById(R.id.password);
        }

publicboolean isConnectingToInternet() {
      ConnectivityManager connectivity = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
if (connectivity != null) {
NetworkInfo[] info = connectivity.getAllNetworkInfo();
if (info != null)
for (NetworkInfo anInfo : info)
if (anInfo.getState() == NetworkInfo.State.CONNECTED)
returntrue;
        }
```

```
            returnfalse;
    }

publicvoid showAlertDialog(Context context, String title, String message, Boolean status) {
        AlertDialog alertDialog = newAlertDialog.Builder(context).create();
alertDialog.setTitle(title);
alertDialog.setMessage(message);
if (status != null)
alertDialog.setIcon((status) ? R.drawable.success : R.drawable.fail);
alertDialog.setButton("OK", new DialogInterface.OnClickListener() {

publicvoid onClick(DialogInterface dialog, int which) {

            }

        });
alertDialog.show();
    }

    @Override
publicvoid onClick(View v) {
super.onClick(v);
if ( v.getId() == R.id.btnRegister ) {
if ( !isConnectingToInternet() ) {
showAlertDialog(Register.this, "Δεν υπάρχει σύνδεση στο internet", "Παρακαλώ συνδεθείτε στο internet", false);
            }
else {
                String nameString = fullName.getText().toString();
                String emailString = email.getText().toString();
                String passwordString = password.getText().toString();
try {
passwordString = SHA512.sha512(passwordString);
            } catch (NoSuchAlgorithmException e) {
Log.e(Config.TAG, "Ο sha512 δεν μπόρεσε να εκτελεσθεί");
            } catch (UnsupportedEncodingException e) {
Log.e(Config.TAG, "Δεν υποστηρίζεται το encoding του κωδικού");
            }
if ( nameString.trim().length() >0&& emailString.trim().length() >0&& passwordString.trim().length() >0 ) {
register(emailString, nameString, passwordString);
            }
else {
Toast.makeText(getApplicationContext(), "Παρακαλώ εισάγετε τα στοιχεία εγγραφής!",
Toast.LENGTH_SHORT).show();
            }
        }
    }
else {
startActivity(new Intent(this, Login.class));
finish();
    }
    }

privatevoid register(String email,final String name, String password) {
final ProgressDialog progressDialog = ProgressDialog.show(this, null, getString(R.string.eggrafi));
final ParseUser pu = new ParseUser();
pu.setEmail(email);
pu.setPassword(password);
pu.setUsername(name);
pu.signUpInBackground(new SignUpCallback() {
        @Override
publicvoid done(ParseException e) {
progressDialog.dismiss();
if (e == null) {
                UserList.username = name;
                Chat.user = pu;
startActivity(new Intent(Register.this, Radius.class).putExtra("username", name));
finish();
            } else {
Toast.makeText(getApplicationContext(), e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        }
    });
    }
}
```

Radius.java

```
package gr.teiwest.cied.fotis.socializer.activity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import gr.teiwest.cied.fotis.socializer.utils.NumberPicker;
import gr.teiwest.cied.fotis.socializer.R;

publicclass Radius extends Custom {

private NumberPicker numberPicker;
private String userName;

    @Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.radius);
        Intent intent = getIntent();
setTouchNClick(R.id.map);
userName = intent.getStringExtra("username");
numberPicker = (NumberPicker) findViewById(R.id.numberPicker);
numberPicker.setMinValue(0);
numberPicker.setMaxValue(5000);
numberPicker.setValue(150);
numberPicker.setWrapSelectorWheel(false);
    }

    @Override
publicvoid onClick(View v) {
int value = numberPicker.getValue();
        Intent intent = newIntent(getApplicationContext(), Map.class);
intent.putExtra("username", userName);
intent.putExtra("radius", value);
startActivity(intent);
    }

}
```

Map.java

```
publicclass Map extends Activity implements GoogleMap.OnMarkerClickListener,
GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener, LocationListener {
publicstaticfinallong UPDATE_INTERVAL_IN_MILLISECONDS = 10000;
publicstaticfinallong FASTEST_UPDATE_INTERVAL_IN_MILLISECONDS =
UPDATE_INTERVAL_IN_MILLISECONDS / 2;
protectedfinalstatic String REQUESTING_LOCATION_UPDATES_KEY = "requesting-location-updates-key";
protectedfinalstatic String LOCATION_KEY = "location-key";
protectedfinalstatic String LAST_UPDATED_TIME_STRING_KEY = "last-updated-time-string-key";
protected GoogleApiClient mGoogleApiClient;
protected LocationRequest mLocationRequest;
protected Location mCurrentLocation;
protected Boolean mRequestingLocationUpdates;
protected String mLastUpdateTime;
private String userName;
private GoogleMap googleMap;
private String longitude;
private String latitude;
private ArrayList<User> users;
privateint radius;
private Menu optionsMenu;
private MarkerOptions myMarker;
private CircleOptions circle;
privatestaticint refreshCount;
privatestaticboolean firstTimeInitialized;
```

```
    @Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.map);
        Intent intent = getIntent();
userName = intent.getStringExtra("username");
radius = intent.getIntExtra("radius", 0);
users = new ArrayList<>();
refreshCount = 0;
firstTimeInitialized = true;
try {
initializeMap();
        }
catch (Exception e) {
e.printStackTrace();
        }
mRequestingLocationUpdates = false;
mLastUpdateTime = "";
updateValuesFromBundle(savedInstanceState);
buildGoogleApiClient();
    }

privatevoid initializeMap() {
if (googleMap == null) {
googleMap = ((MapFragment) getFragmentManager().findFragmentById(R.id.map)).getMap();
if (googleMap == null) {
Toast.makeText(getApplicationContext(), "Ανεπιτυχής δημιουργία του χάρτη!", Toast.LENGTH_SHORT).show();
            }
        }
googleMap.setOnMarkerClickListener(this);
myMarker = new MarkerOptions().title("Εσύ");
circle = new CircleOptions().radius(radius).fillColor(Color.argb(80, 173, 192, 232)).strokeWidth(0);
if ( firstTimeInitialized ) {
googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(37.984124, 23.728013), 10.0f));
firstTimeInitialized = false;
        }
    }

publicvoid sendLocation() {
        String serverURL = Config.SERVER_URL + "store_location.php";
        LongOperation2 serverRequest = newLongOperation2();
serverRequest.execute(serverURL, longitude, latitude, userName);
    }

privatevoid updateValuesFromBundle(Bundle savedInstanceState) {
if (savedInstanceState != null) {
// Update the value of mRequestingLocationUpdates from
// the Bundle, and make sure that
// the Start Updates and Stop Updates buttons are correctly
// enabled or disabled.
if (savedInstanceState.keySet().contains(REQUESTING_LOCATION_UPDATES_KEY)) {
mRequestingLocationUpdates = savedInstanceState.getBoolean(REQUESTING_LOCATION_UPDATES_KEY);
            }
// Update the value of mCurrentLocation from
// the Bundle and update the UI to show the
// correct latitude and longitude.
if (savedInstanceState.keySet().contains(LOCATION_KEY)) {
// Since LOCATION_KEY was found in the
// Bundle, we can be sure that mCurrentLocation
// is not null.
mCurrentLocation = savedInstanceState.getParcelable(LOCATION_KEY);
            }
// Update the value of mLastUpdateTime from the Bundle and update the UI.
if (savedInstanceState.keySet().contains(LAST_UPDATED_TIME_STRING_KEY)) {
mLastUpdateTime=savedInstanceState.getString(LAST_UPDATED_TIME_STRING_KEY);
            }
updateValues();
        }
    }
```

```
protectedsynchronizedvoid buildGoogleApiClient() {
mGoogleApiClient = new GoogleApiClient.Builder(this)
.addConnectionCallbacks(this)
.addOnConnectionFailedListener(this)
.addApi(LocationServices.API)
.build();
createLocationRequest();
   }

protectedvoid createLocationRequest() {
mLocationRequest = new LocationRequest();
mLocationRequest.setInterval(UPDATE_INTERVAL_IN_MILLISECONDS);
mLocationRequest.setFastestInterval(FASTEST_UPDATE_INTERVAL_IN_MILLISECONDS);
mLocationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
   }

publicvoid startUpdates() {
if (!mRequestingLocationUpdates) {
mRequestingLocationUpdates = true;
startLocationUpdates();
      }
   }

publicvoid stopUpdates() {
if (mRequestingLocationUpdates) {
mRequestingLocationUpdates = false;
stopLocationUpdates();
      }
   }

protectedvoid startLocationUpdates() {
LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient, mLocationRequest, this);
   }

privatevoid updateValues() {
if (mCurrentLocation != null) {
latitude = String.valueOf(mCurrentLocation.getLatitude());
longitude = String.valueOf(mCurrentLocation.getLongitude());
      }
   }

protectedvoid stopLocationUpdates() {
LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, this);
   }

   @Override
protectedvoid onStart() {
super.onStart();
mGoogleApiClient.connect();
   }

   @Override
protectedvoid onResume() {
super.onResume();
initializeMap();
if (mGoogleApiClient.isConnected() && mRequestingLocationUpdates) {
startLocationUpdates();
      }
   }

   @Override
protectedvoid onPause() {
super.onPause();
if (mGoogleApiClient.isConnected()) {
stopLocationUpdates();
      }
   }

   @Override
protectedvoid onStop() {
super.onStop();
mGoogleApiClient.disconnect();
   }

   @Override
```

```
publicvoid onConnected(Bundle connectionHint) {
if (mCurrentLocation == null) {
mCurrentLocation = LocationServices.FusedLocationApi.getLastLocation(mGoogleApiClient);
mLastUpdateTime = DateFormat.getTimeInstance().format(new Date());
updateValues();
startUpdates();
        }
if
```