

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ

“ΔΙΔΑΚΤΙΚΗ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΨΗΦΙΑΚΩΝ
ΣΥΣΤΗΜΑΤΩΝ”

ΚΑΤΕΥΘΥΝΣΗ : ΨΗΦΙΑΚΕΣ ΕΠΙΚΟΙΝΩΝΙΕΣ ΚΑΙ ΔΙΚΤΥΑ



ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ ΕΙΔΙΚΕΥΣΗΣ

**Σχεδιασμός και ανάπτυξη μηχανισμού αναζήτησης και
προβολής πληροφοριών βάσει προτιμήσεων χρηστών με
χρήση Android.**

Εισηγητής : Τρίγκα Ρεββέκα

Υπεύθυνος Καθηγητής : Βέρα – Αλεξάνδρα Σταυρουλάκη

ΠΕΙΡΑΙΑΣ , Ιανουάριος 2013

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία αναπτύχθηκε μια εφαρμογή κινητού τηλεφώνου με χρήση της πλατφόρμας Google Android.

Πρόκειται για μια εφαρμογή αναζήτησης δεδομένων. Σε κάθε χρήστη δίνεται η δυνατότητα δημιουργίας προφίλ όπου και θέτει τα κριτήρια της αναζήτησης του, η συγκεκριμένη εφαρμογή έχει σχεδιαστεί για αναζήτηση καταστημάτων, έτσι στο χρήστη δίνετε η δυνατότητα να επιλέξει το είδος του μαγαζιού που ψάχνει, ενώ παράλληλα λαμβάνετε υπ' όψιν μία ακόμα παράμετρος , αυτή των προτιμήσεων των χρηστών. Δηλαδή έχουμε υποθέσει ότι τα καταστήματα αυτά έχουν βαθμολογηθεί και κάθε ένα από αυτά έχει και ένα ranking προτίμησης από τους πελάτες του, έτσι ο χρήστης δημιουργεί το προφίλ του, επιλέγει το είδος καταστήματος προς αναζήτηση και το επιθυμητό ranking για το κατάστημα αυτό. Η εφαρμογή επιστρέφει τα αποτελέσματα της αναζήτησης αυτής δείχνοντας τα καταστήματα στο χάρτη.

Ο κύριος στόχος της πτυχιακής εργασίας ήταν η ανάπτυξη και ο σχεδιασμός ενός λογισμικού κινητού τηλεφώνου που θα αναδείξει τις δυνατότητες που προσφέρει στις νέες εφαρμογές η ραγδαία εξέλιξη του Διαδικτύου που οδήγησε στη νέα γενιά Web ,που ονομάζεται Μελλοντικό διαδίκτυο.

ΠΕΡΙΕΧΟΜΕΝΑ

1 Εισαγωγή	5
1.2 Στόχος.....	5
1.3 Δομη.....	6
2 Υπάρχουσες τεχνολογίες και εφαρμογές	7
2.1 Υπάρχουσες Τεχνολογίες.....	9
2.2 Προοπτικές εφαρμογής.....	12
2.3 Λίστα εφαρμογών.....	15
3 Μηχανισμοί αναζήτησης προτιμήσεων	19
3.1 Μηχανισμοί αναζήτησης προτιμήσεων χρηστών.....	19
3.2 Καταγραφή προτιμήσεων χρηστών στην εφαρμογή που αναπτύχθηκε.....	21
3.3 Υπάρχοντα συστήματα.....	21
4 Επιλογή Android	23
4.1 Επιλογή του Android.....	23
4.2 Τι είναι το Android.....	25
4.3 Αρχιτεκτονική του Android.....	29
4.4 Η εικονική μηχανή Dalvik.....	30
4.5 Πυρήνας Linux (Linux kernel).....	31
4.6 Εγγενείς Βιβλιοθήκες (Native Libraries).....	32
4.6.1 Βιβλιοθήκες Android.....	32
4.6.2 Προηγμένες Βιβλιοθήκες Android.....	34
4.7 Χρόνος Εκτέλεσης (Android Runtime).....	35
4.8 Πλαίσιο Εφαρμογής (Application Framework).....	36
4.9 Βασικά Συστατικά εφαρμογών του Android.....	38

4.10 Γιατί Android.....	43
5. Υλοποίηση εφαρμογής	44
5.1 Κλάσεις Java.....	45
5.2 Use Cases Diagram.....	47
5.2.1 Use Case : Create profil.....	47
5.2.2 Use case :Define preference.....	48
5.2.3 Use case Show.....	59
5.2.4 Use case Back.....	50
6 .Παρουσίαση εφαρμογής	51
6.1 Εγκατάσταση Πλατφόρμας.....	51
7.Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης.....	92
7.1 Μελλοντική επέκταση.....	93
7.2 Βιβλιογραφία.....	94
Παράρτημα.....	96

1 Εισαγωγή

Η εξέλιξη των τηλεπικοινωνιών και των τεχνολογιών που τις απαρτίζουν, έχουν συμβάλει στη διεύρυνση του Διαδικτύου στο σύγχρονο τρόπο ζωής των ανθρώπων και έχουν αλλάξει τον τρόπο που αυτό χρησιμοποιείται. Η καθημερινή χρήση των “έξυπνων” συσκευών βελτιώνει την ποιότητα ζωής του σύγχρονου ανθρώπου καθώς λειτουργούν με γνώμονα τις ανάγκες του και τις συνήθειες του.

Οι νέες τεχνολογίες αλλάζουν τον τρόπο με τον οποίο χρησιμοποιούμε το διαδίκτυο, οι έξυπνες συσκευές, και οι οικιακές συσκευές και μηχανές επικοινωνούν και αλληλεπιδρούν χωρίς την ανθρώπινη παρέμβαση. Αυτό έχει επιφέρει επανάσταση στην καταναλωτική κοινωνία και το διαδίκτυο είναι το βασικό οικονομικό εργαλείο πια. Στο όχι και τόσο μακρινό μέλλον, οι άνθρωποι θα βασίζονται αποκλειστικά στις διαδικτυακές υπηρεσίες και οι συναλλαγές θα πραγματοποιούνται αποκλειστικά στο διαδίκτυο του μέλλοντος που θα τις παρέχει.

Γεννήθηκε λοιπόν η ιδέα του Web 2.0, η δεύτερη γενιά του διαδικτύου που ουσιαστικά οδήγησε στην δημιουργία νέων υπηρεσιών όπως οι υπηρεσίες και τα περιεχόμενα που δημιουργούνται από τους χρήστες, κοινωνικά δίκτυα, συνεργατικά webs , Web-TV, κτλ. Δημιουργείται λοιπόν η ανάγκη για «έξυπνες εφαρμογές» εφαρμογές που θα ταιριάζουν στο νέο αυτό τύπο Διαδικτύου (Μελλοντικό Διαδίκτυο) και θα ικανοποιούν τις συνεχώς αυξανόμενες απαιτήσεις της σημερινής κοινωνίας

1.2 Στόχος

Ο κύριος στόχος της πτυχιακής εργασίας ήταν η ανάπτυξη και ο σχεδιασμός ενός λογισμικού κινητού τηλεφώνου με σκοπό την ανάδειξη των δυνατοτήτων της νέας γενιάς τεχνολογιών.

Πρόκειται για το σχεδιασμό και την ανάπτυξη μια εφαρμογής αναζήτησης δεδομένων λαμβάνοντας υπ’ όψιν τις προτιμήσεις των χρηστών μέσω της δημιουργίας profil.

1.3 Δομή

Η πτυχιακή εργασία ακολουθεί μια δομή η οποία αποτελείται από επτά κεφάλαια.

Στο πρώτο κεφάλαιο περιγράφεται το αντικείμενο της εργασίας ενώ γίνεται ξεκάθαρος ο στόχος της πτυχιακής αυτής.

Στο δεύτερο κεφάλαιο δίνεται ο ορισμός του Μελλοντικού διαδικτύου και αναδεικνύονται οι απεριόριστες δυνατότητες που μπορεί να προσφέρει σε ανθρώπους και εφαρμογές (αντικείμενα) ενώ παράλληλα γίνεται αναφορά σε μια μεγάλη γκάμα τομέων που μπορεί να βρει εφαρμογή η νέα έννοια του Διαδικτύου.

Στο τρίτο κεφάλαιο παρουσιάζεται η αποδοτική αναζήτηση μέσα από τεχνικές δημιουργίας προφίλ και καταγραφής προτιμήσεων χρηστών. Τέλος παρουσιάζονται ήδη υπάρχουσες εφαρμογές που στηρίζονται σε αυτού το είδους τις τεχνικές.

Στο τέταρτο κεφάλαιο αναλύονται οι λόγοι επιλογής της πλατφόρμας android για την υλοποίηση της εφαρμογής .

Στο πέμπτο κεφάλαιο παρουσιάζεται ο σχεδιασμός και οι απαιτήσεις της εφαρμογής που υλοποιήθηκε. Ενώ περιγράφονται όλες οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

Στο έκτο κεφάλαιο γίνεται παρουσίαση του συστήματος και του τρόπου λειτουργίας του μέσω screenshots.

Στο έβδομο κεφάλαιο περιλαμβάνονται συμπεράσματα και προτάσεις για μελλοντική βελτίωση-ανάπτυξη της εφαρμογής.

Τέλος περιλαμβάνεται ένα παράρτημα με τον κώδικα που αναπτύχθηκε κατά την υλοποίηση της εφαρμογής αυτής.

2. Υπάρχουσες τεχνολογίες και εφαρμογές

Σύμφωνα με την ITU, είμαστε στην αρχή μιας νέας επανάστασης αφού , για τις επόμενες δύο δεκαετίες , το ταχύτατα αυξανόμενο πλήθος των ανθρώπων που “σερφάρουν” στο Διαδίκτυο θα ξεπεραστεί από το πλήθος άψυχων αντικειμένων που θα επικοινωνούν “ on line ” μεταξύ τους. Για παράδειγμα, τα ψυγεία θα μπορούν να ανταλλάσσουν πληροφορίες με σούπερ μάρκετ, τα πλυντήρια με το ρουχισμό- ή ακόμα τα ίδια τα ρούχα θα μπορούν να “ μιλούν”. Η σύνδεση του φυσικού κόσμου με τον εικονικό κόσμο του Διαδικτύου ονομάστηκε “Διαδίκτυο των Αντικειμένων”. Στο μελλοντικό Διαδίκτυο των Αντικειμένων ένα κινητό τηλέφωνο, για παράδειγμα , μπορεί να προσαρμοστεί σε μία συσκευή αναγνώρισης που μεταδίδει πληροφορίες για το ποιος είστε και που βρίσκεστε, ενώ ταυτόχρονα μπορείτε να λαμβάνετε πληροφόρηση για την δική σας θέση.

Υπάρχουν πάνω από 2 δισεκατομμύρια κινητά τερματικά που χρησιμοποιούνται σήμερα και πάνω από 1 δισεκατομμύριο χρήστες του Internet παγκοσμίως. Ασύρματες ,κινητές και δικτυακές τεχνολογίες έδωσαν σε πρώτη φάση τη δυνατότητα διάχυτης επικοινωνίας μεταξύ συστημάτων και εφαρμογών και δημιούργησαν πολύ σημαντικές οικονομικές ευκαιρίες. Ωστόσο, αυτό μπορεί να είναι μόνο η αρχή : εξελισσόμενες τεχνολογίες δικτύωσης , όπως η ταυτοποίηση μέσω ραδιοσυχνοτήτων (RFID tags) , ενδεχομένως θα μπορούν να αναπτυχθούν σε εκατοντάδες δισεκατομμύρια , για μια σχεδόν απεριόριστη γκάμα εφαρμογών.

Πιο μακροπρόθεσμα , αναμένεται απλές ετικέτες του σήμερα να εξελιχθούν σε ευφυέστερα αντικείμενα με αυξημένη αποθήκευση , επεξεργασία , καθώς και δυνατότητα προσθήκης αισθητήρων. Αυτό με τη σειρά του θα ανοίξει ένα εντελώς νέο φάσμα σε δικτυακές εφαρμογές , όπως M2M “Man –to-Machine, Machine-to-Man, Machine-to-Mobile and Mobile-to-Machine” και context-aware επικοινωνίες.

Το “Διαδίκτυο των αντικειμένων” είναι ένα δίκτυο δισεκατομμυρίων ή τρισεκατομμυρίων μηχανών οι οποίες επικοινωνούν μεταξύ τους. Είναι πολύ σημαντικό και κυρίαρχο θέμα για την εξέλιξη των

πληροφοριών και των επικοινωνιών τις επόμενες δεκαετίες , καθώς η απλούστερη μορφή αυτού είναι ήδη σε χρήση. Υπάρχουν 1,3 δισεκατομμύρια ταυτοποιήσεις μέσω ραδιοσυχνοτήτων (RFID) και δύο δισεκατομμύρια χρήστες κινητών υπηρεσιών , σε όλων τον κόσμο.

Η ιδέα έχει διευρυνθεί από προηγμένες έννοιες τα τελευταία είκοσι χρόνια:

- Ubiquitous communications
- Pervasive computing
- Ambient intelligence

Από μία γενική σκοπιά, μπορεί να συναχθεί το συμπέρασμα ότι η τάση για όλο και μεγαλύτερο αριθμό συνδεδεμένων ευφυών αντικειμένων είναι αμετάκλητη, διότι η οικονομική αξία ενός συστήματος αντικειμένων και συσκευών είναι άμεσα συνδεδεμένη με το γεγονός ότι είναι «δικτυωμένες».

Το Διαδίκτυο των αντικειμένων θα επιτρέψει μέσω της πληροφορικής την υποστήριξη των οικονομικών , της υγείας , της κοινωνικότητας αλλά και της ιδιωτικότητας.

Παρακάτω αναφέρονται τρία απλά παραδείγματα:

Διαχείριση περιουσιακών στοιχείων. Ηλεκτρονική σήμανση και παρακολούθηση της θέσης των αποσκευών σε έναν αερολιμένα , ή των εμπορευμάτων στο πλαίσιο μιας διαδικασίας παραγωγής στο εργοστάσιο.

Υγειονομική περίθαλψη. Αισθητήρες για αρτηριακή πίεση και καρδιακή συχνότητα, τακτικές μετρήσεις από το σπίτι του ασθενούς θα μπορούν να στέλνονται σε κάποιο κέντρο παρακολούθησης .Ένας υπολογιστής θα εντοπίζει τυχόν ανεπιθύμητες κινήσεις ή σήματα και θα ειδοποιεί τον υπεύθυνο γιατρό να ελέγξει τον ασθενή.

Παρακολούθηση περιβάλλοντος. Ένα δίκτυο αισθητήρων θα είναι σε θέση να παρακολουθεί το ύψος του ποταμού και των βροχοπτώσεων , θα μπορεί να επιτευχθεί πρόβλεψη των

πλημμύρων και στήριξη των μέτρων διαχείρισης των υδάτων για την ανακούφιση από τυχόν πλημμύρες.

Οι πιθανές εφαρμογές της διαδεδομένης δικτύωσης είναι απεριόριστες. Μερικές εφαρμογές φαίνονται απαραίτητες, για παράδειγμα ο εντοπισμός ανθρώπων σε ένα πιθανό σενάριο καταστροφής, όπως φωτιά σε κάποιο τούνελ. Άλλες εφαρμογές μπορεί εκ πρώτης όψεως να φαίνονται μη ρεαλιστικές. Για να θεωρηθεί ότι η τεχνολογία πραγματικά αξίζει τον κόπο, υπάρχει ανάγκη για αντιμετώπιση του μεγάλου όγκου των εφαρμογών προκειμένου να κατανοηθούν καλύτερα οι διάφορες απαιτήσεις (π.χ. πραγματικός χρόνος, ποιότητα υπηρεσιών), που θα οδηγήσουν τελικά στην απαραίτητη “γενική” τεχνολογική εξέλιξη.

- Η προοπτική του συστήματος για το πώς τα δικτυωμένα συστήματα είναι πιθανό να εξελιχθούν.
- Οι επιπτώσεις της διασυνδεσιμότητας της συσκευής για τις μελλοντικές αρχιτεκτονικές δικτύου και των νέων τεχνολογιών.
- Συγκεκριμένη ασφάλεια και προστασία της ιδιωτικότητας.
- Η αρχιτεκτονική που απαιτείται για την υποστήριξη των απαιτήσεων των τρισεκατομμυρίων συσκευών που συνδέονται.
- Η σύνθεση των υπηρεσιών.
- Ο ρόλος των “ανοικτών” (λογισμικών) μοντέλων.
- Ο ρόλος των ασύρματων τεχνολογιών.
- Αυτά καθιερώθηκαν σαν ερωτήσεις που πρέπει να απαντηθούν προκειμένου να κατανοηθούν οι απαιτήσεις για το παγκόσμιο δίκτυο του αύριο, για το πώς μπορεί να οικοδομηθεί και για το πώς θα είναι.

2.2 Υπάρχουσες Τεχνολογίες

Το διαδίκτυο έχει καταστεί η πλατφόρμα ώστε να ξεκινήσουν Information Technology (IT) υπηρεσίες για όλα τα είδη των βιομηχανιών. Τα δίκτυα και οι υπηρεσίες λογισμικού συνδέονται μεταξύ τους με κάθε τρόπο. Οι αρχιτεκτονικές αρχές για τη

δημιουργία προϊόντων λογισμικού και υπηρεσιών έχουν εξελιχθεί. Αρχιτεκτονικές αρχές όπως Representational State Transfer (REST), Event Driven Architecture (EDA), and Service Oriented Architecture (SOA) χρησιμοποιούνται για την δημιουργία υπηρεσιών λογισμικού που χρησιμοποιούνται από εκατομμύρια ανθρώπους και επιχειρήσεις σε όλον τον κόσμο. Για παράδειγμα, το REST έχει αποδείξει την χρησιμότητα του ως μια πετυχημένη αρχή σχεδιασμού αφού όλο το World Wide Web έχει σχεδιαστεί με βάση αυτό. Το SOA χρησιμοποιείται ευρέως για τη δημιουργία centric και web εφαρμογών επιχειρήσεων και υπηρεσιών. Έχει εξελιχθεί σε ένα νέο ώριμο επίπεδο , σε σύγκριση με άλλες τεχνολογίες που ήταν διαθέσιμες στο παρελθόν για την παροχή γρήγορων υπηρεσιών. Όλες αυτές οι αρχές αρχιτεκτονικής συμπληρώνουν επίσης η μία την άλλη για το σχεδιασμό νέων υπηρεσιών.

RFID Systems

Μία από τις πιθανές τεχνολογίες ανάγνωσης που προτείνεται στην έκθεση της ITU είναι η Radio Frequency Identification (RFID). Η RFID είναι μια αυτόματη μέθοδος αναγνώρισης που χρησιμοποιεί τα ραδιοκύματα για να εντοπίσει σηματοδοτημένα στοιχεία όπως ανθρώπους , ζώα ή αντικείμενα. Είναι παρόμοιο με το bar code, αλλά χρησιμοποιεί ένα μικροτσιπ μαζί με ραδιοκύματα. Μια ετικέτα RFID , επίσης είναι γνωστή και ως electronic label, “transponder” ή “ code plate” η οποία αποτελείται από ένα τσιπ RFID που συνδέεται με μία κεραία. Οι ετικέτες μπορούν να διαβαστούν από πολλά μέτρα μακριά και χωρίς να χρειάζεται οπτική επαφή. Η πιο κοινή μέθοδος αναγνώρισης είναι να φυλάσσεται ο αύξοντα αριθμός που προσδιορίζει κάποιο άτομο ή κάποιο αντικείμενο, αλλά μια RFID ετικέτα μπορεί επιπλέον να αποθηκεύσει και άλλες πρόσθετες πληροφορίες. Τα συστήματα RFID χρησιμοποιούν ετικέτες RFID που αποτελούν τη σημαντικότερη και αποδοτικότερη μέχρι στιγμής τεχνολογία για τις υπηρεσίες αναγνώρισης στοιχείου όσο αφορά το «Internet of Things».

Τα συστήματα RFID αποτελούνται από τρία κύρια στοιχεία : ετικέτες RFID , αναγνώστες RFID , και RFID λογισμικό.

Μια ετικέτα RFID ή αναμεταδότης μεταφέρει τα στοιχεία ταυτότητας που μπορεί να είναι μια μοναδική συμβολοσειρά ή ένας κωδικός. Επιπλέον, η ετικέτα μπορεί να αποθηκεύσει πληροφορίες ανάλογα με το μέγεθος της μνήμης της. Μια ετικέτα επισυνάπτεται φυσικά στο αντικείμενο που πρέπει να εντοπιστεί. Πρόκειται για μία ηλεκτρική συσκευή σχεδιασμένη ώστε να λαμβάνει ένα ειδικό σήμα και να στέλνει αυτόματα μια συγκεκριμένη απάντηση. Αποτελείται από ένα στοιχείο σύνδεσης (όπως ένα πηνίο, ή μια κεραία μικροκυμάτων) και ένα ηλεκτρονικό μικροσίπ (λιγότερο από το 1/3 του χιλιοστού σε μέγεθος). Μπορεί να είναι παθητική, ημι - παθητική ή ενεργητική, με βάση την πηγή τροφοδοσίας της και τον τρόπο χρησιμοποίησης τους, και μπορεί να είναι μόνο ανάγνωσης, ανάγνωσης ή εγγραφής και ανάγνωσης / εγγραφής και αναδιατύπωσης, ανάλογα με το πώς τα δεδομένα είναι κωδικοποιημένα. Οι παθητικές ετικέτες RFID λαμβάνουν την ενέργεια από το ηλεκτρομαγνητικό πεδίο που εκπέμπεται από τους αναγνώστες. Οι ετικέτες χρησιμοποιούν συχνότητες εκπομπής και κυμαίνονται στα kilohertz, megahertz, και gigahertz.

Ένας RFID αναγνώστης είναι μία συσκευή που χρησιμοποιείται για να διαβάσει τα δεδομένα που στέλνονται από την ετικέτα. Για ανάγνωση παθητικών ετικετών RFID, ο αναγνώστης πρέπει να χρησιμοποιήσει επιπλέον ενέργεια σε ηλεκτρομαγνητική μορφή. Η συσκευή ανάγνωσης αποτελείται από μια διπολική κεραία και ένα μικροεπεξεργαστή. Η διπολική κεραία χρησιμοποιείται για την μετάδοση των σημάτων και την τροφοδότηση της ετικέτας. Ο μικροεπεξεργαστής ελέγχει και διευθύνει όλες τις σχετικές διεργασίες της συσκευής ανάγνωσης. Ο αναγνώστης μπορεί να είναι είτε μια ειδική συσκευή χειρός είτε να είναι ενσωματωμένος σε μια κινητή συσκευή ή μπορεί ακόμα να ενταχθεί σε ένα τοίχο (σταθερές συσκευές ανάγνωσης). Σε σύγκριση με τις ετικέτες, οι συσκευές ανάγνωσης είναι μεγαλύτερες, πιο ακριβές, και καταναλώνουν περισσότερη ενέργεια ανάλογα με την ιδιότητα τους. Η συσκευή ανάγνωσης πρέπει να έχει επίσης τη δυνατότητα επικοινωνίας (ενσύρματη ή ασύρματη).

Το RFID λογισμικό είναι ένα υλικολογισμικό που τρέχει για την συσκευή ανάγνωσης RFID. Υπάρχουν διάφορες υπηρεσίες που

περιλαμβάνονται σε αυτό. Υπηρεσίες όπως η ανάγνωση, το κλείδωμα, η εγγραφή, και η διαγραφή μιας ετικέτας περιλαμβάνονται στο βασικό σύνολο λειτουργιών. Επιπλέον, το φιλτράρισμα, το άθροισμα, η αναζήτηση, η καταγραφή συμβάντων, και οι υπηρεσίες ασφάλειας περιλαμβάνονται και αυτά στο λογισμικό. Το υλικολογισμικό επίσης χρειάζεται για να επικοινωνεί η συσκευή ανάγνωσης με το δίκτυο, η οποία θα πρέπει να επεξεργάζεται και τις εντολές που λαμβάνει από ένα απομακρυσμένο σύστημα ελέγχου.

Υπάρχουν δύο τύποι RFID: κοντινού πεδίου RFID και μακρινού πεδίου RFID. Με Near Field Communication (NFC) RFID είναι εξοπλισμένα πολλά τηλέφωνα τα τελευταία χρόνια και πολλές NFC υπηρεσίες είναι διαθέσιμες ήδη για NFC τηλέφωνα. Οι NFC συσκευές ανάγνωσης διαβάζουν την ετικέτα εντός εμβέλειας 5 cm ή μερικές φορές χρειάζεται ένα άγγιγμα. Μακρινού πεδίου RFID καλύπτουν απόσταση μεταξύ τριών και έξι μέτρων από σταθερές συσκευές ανάγνωσης.

2.3 Προοπτικές εφαρμογής

Οι πιθανές χρήσεις της τεχνολογίας του Διαδικτύου των πραγμάτων δεν έχουν όριο. Καλύπτουν πολλές ιδέες εφαρμογών, συμπεριλαμβανομένων των:

- Τρεχουσών κερδοφόρων εφαρμογών
- Εφαρμογών δοκιμής και επίδειξης
- Προγραμματισμένων εφαρμογών
- Υποθετικών εφαρμογών

Το Διαδίκτυο των αντικειμένων αποτελεί συγχώνευση του φυσικού και ψηφιακού κόσμου. Δημιουργεί ένα χάρτη του πραγματικού κόσμου μέσα στον εικονικό. Ο υπολογιστής αντιλαμβάνεται το φυσικό κόσμο, ανάλογα με τα χαρακτηριστικά του δικτύου αισθητήρων, με υψηλή χρονική και χωρική ανάλυση. Με άλλα λόγια, μπορεί να κατέχει πολλές αναλυτικές πληροφορίες. Ο υπολογιστής αντιλαμβάνεται τον πραγματικό κόσμο όχι όπως

ακριβώς ο άνθρωπος, αλλά μπορεί να απαλλαγεί από τους περιορισμούς που επιβάλλονται από την ανθρώπινη κατανόηση.

Το Διαδίκτυο των αντικειμένων μπορεί να αντιδρά αυτόνομα στον πραγματικό κόσμο. Αυτό είναι γνωστό ως proactive computing. Το πώς αντιλαμβάνεται ένας υπολογιστής τον κόσμο του επιτρέπει να αλληλεπιδρά με το φυσικό κόσμο και τον επηρεάζει. Μπορεί να τρέχει διαδικασίες χωρίς να χρειάζεται από τον άνθρωπο “να πατάει κάποιο κουμπί”. Το Διαδίκτυο των αντικειμένων δεν είναι απλώς ένα εργαλείο για να επεκταθεί η ανθρώπινη οντότητα. Γίνεται μέρος του περιβάλλοντος στο οποίο οι άνθρωποι ζούν και εργάζονται, και έτσι δημιουργεί ένα οικονομικά, κοινωνικά, και προσωπικά καλύτερο περιβάλλον. Στον τομέα της βιομηχανίας και του εμπορίου, το Διαδίκτυο των αντικειμένων μπορεί να επιφέρει αλλαγή των επιχειρηματικών διαδικασιών.

Οι RFID εφαρμογές έχουν εγκατασταθεί και θα παραμείνουν ενώ τεχνικά πρότυπα αρχίζουν να προκύπτουν. Η πλειονότητα των επιτυχημένων εφαρμογών αυτή την περίοδο χρησιμοποιούν RFID στη διαδικασία της βιομηχανικής και εμπορικής παρακολούθησης και στη διαχείριση περιουσιακών στοιχείων. Πιστεύεται ότι έχει επιτευχθεί απόδοση των επενδύσεων εντός 6-12 μηνών. Οι περισσότερες επιστροφές αναμένεται να γίνουν όταν η παρακολούθηση θα εφαρμόζεται σε υψηλής αξίας κινητά στοιχεία που μέχρι σήμερα δεν έχουν δομημένο χειρισμό. Η απόδοση είναι πιθανό είναι λιγότερο σημαντική σε περιπτώσεις κατά τις οποίες τα RFID αντικαταστήσουν το δομημένο χειρισμό που βασίζεται σε παλαιότερες τεχνολογίες, για παράδειγμα barcodes ή χειροκίνητα συστήματα βασισμένα στο χαρτί.

Ο ανασχεδιασμός της επιχειρηματικής διαδικασίας δεν είναι απαραίτητος απλός. Ορισμένες εφαρμογές μπορεί να λειτουργήσουν χωρίς ανθρώπινη εισαγωγή. Σε άλλες περιπτώσεις, η ανθρώπινη παρουσία και η συμμετοχή θα είναι δύσκολο να παραλειφθεί, ακόμη και στο απλούστερο επίπεδο του πατήματος του κουμπιού ώστε να ξεκινήσει η διαδικασία. Μερικές εφαρμογές μπορεί να αποσκοπούν στη συμπλήρωση, υποβοήθηση και στον εμπλουτισμό και όχι στην αντικατάσταση του ανθρώπινου ελέγχου της διαδικασίας.

Η αυτοκινητοβιομηχανία είναι πιθανόν να αποτελέσει σημαντικό χρήστη του Διαδικτύου των αντικειμένων. Αναμένεται ότι το ποσοστό των αυτοκινήτων που περιέχουν ενσωματωμένο σύστημα τηλεματικής θα αυξηθεί σε 43% το 2013. Τα κίνητρα για τη χρήση ραδιοκυμάτων στα αυτοκίνητα, παρά το γεγονός ότι πρόκειται για ένα δύσκολο ηλεκτρικά περιβάλλον για την ασύρματη μετάδοση, ήταν η παραμετροποίηση, η ευελιξία, η καινοτομία και η απλούστευση της κατασκευής.

Η τηλεματική πλατφόρμα για αυτοκίνητο είναι υπό κατασκευή. Αυτή θα χρησιμοποιεί τις επικοινωνίες για :

- Την παρακολούθηση των αυτοκινήτων
- Τις υποδομές του αυτοκινήτου καθώς και τις υποδομές της επικοινωνίας μεταξύ των αυτοκινήτων
- Τη συνεργασία οχήματος με όχημα

Οι κατασκευαστές μπορούν να χρησιμοποιούν την τεχνολογία αισθητήρων για την δημιουργία επεκτάσιμων προϊόντων. Αυτό προσθέτει αξία στα φυσικά αγαθά παρέχοντας ένα στοιχείο υπηρεσίας. Η ρομποτική σήμερα συνήθως συνδέεται με τις κατασκευές και τις επιχειρήσεις, ωστόσο πιστεύεται πως η αγορά της ρομποτικής που σχετίζεται με το σπίτι μπορεί να υπερβεί τη βιομηχανική αγορά δέκα με είκοσι φορές.

Η εγχώρια αγορά κυρίως στον τομέα της υγειονομικής περίθαλψης, προσθέτει επιπλέον απαιτήσεις για ελαφριές και λειτουργικές συσκευές.

Η επιτυχής ανάπτυξη των εφαρμογών εξαρτάται από την ανάπτυξη των τεχνικών πλατφόρμων και από τις κατάλληλες προδιαγραφές τους. Αυτές περιλαμβάνουν :

- Service object architecture(SOA)
- Ψηφιακή κατασκευή
- Ευέλικτη κατασκευή
- Έξυπνη κατασκευή

- Ανάσυρση δεδομένων
- Διαχείριση δεδομένων
- Δυνατότητα χρησιμοποίησης
- Product representations
- Μέσα εξατομίκευσης

Τα δίκτυα των αισθητήρων και οι κυρίαρχες προσεγγίσεις δικτύωσης παρουσιάζουν, επίσης, μεγάλο ενδιαφέρον για την βιομηχανία κινητών. Το πλαίσιο εφαρμογής που χρησιμοποιούν οι κινητές συσκευές με διάφορες δυναμικά πλατφόρμες παροχής υπηρεσιών αποτελούν αντικείμενο έρευνας από φορείς εκμετάλλευσης της κινητής τηλεφωνίας. Πιλοτικά συστήματα υπάρχουν ήδη δείχνοντας τη δυνατότητα μιας φορητής συσκευής να αλληλεπιδρά με το περιβάλλον, π.χ. σε διάφορα πλαίσια των καταναλωτών. Ένας σημαντικός κατασκευαστής της ΕΕ έχει ξεκινήσει επίσης μια πρωτοβουλία που ονομάζεται “Sensor Planet”, το οποίο αποτελείται από μία πλατφόρμα δοκιμής για την έρευνα κινητών-κεντρικών ασύρματων δικτύων αισθητήρων. Η συντριπτική πλειοψηφία της ερευνητικής πρόκλησης στο πλαίσιο των κινητών επικοινωνιών με τις συσκευές σχετίζεται με την πλατφόρμα παροχής υπηρεσιών και με την απαίτηση διάθεσης “ενήμερων υπηρεσιών” που θα μπορούν να αντιδράσουν δυναμικά και να προσαρμοστούν στο περιβάλλον.

2.4 Λίστα εφαρμογών

Παρακάτω ακολουθεί μια ταξινομημένη λίστα των εφαρμογών του Διαδικτύου των αντικειμένων. Η παρούσα έκθεση δεν σημαίνει ότι έχει συμπεριλάβει κάθε δυνατή εφαρμογή αυτής της τεχνολογίας. Ωστόσο, αυτό δίνει μια εικόνα του φάσματος των δυνατοτήτων της.

Έξυπνα κτήρια

- Αυτοματοποιημένη οικοδόμηση

Μεταφορές

- Παρακολούθηση της θερμοκρασίας που φέρουν τα σιδηροδρομικά οχήματα
- Έγκαιρος έλεγχος της διάρκειας ζωής
- Διοικητικές μέριμνες μεταφορών
- Ναυτιλία
- Επικόλληση ετικετών και πληρωμή
- Κινητά τηλέφωνα που θα αλληλεπιδρούν με τα συστήματα πληρωμών

Αυτοκινητοβιομηχανία

- Ευφυή ελαστικά
- Ταυτότητα οχημάτων
- Σήματα αυτοκινητόδρομων
- Συνεταιριστική οδήγηση οχήματος με όχημα
- Εσωτερική αρχιτεκτονική ελέγχου οχημάτων
- Υποδομές επικοινωνίας μεταξύ οχημάτων: οδικές υπηρεσίες πληροφοριών
- Ανακύκλωση των υλικών των οχημάτων
- Αισθητήρας στάθμευσης
- Ανίχνευση : ζώνης ασφαλείας , άνοιγμα πόρτας , θερμοκρασίας.

Έλεγχος περιβαλλοντικών συνθηκών

- Έλεγχος πληθυσμού πτηνών
- Ανίχνευση ακτινοβολίας
- Μετεωρολογικός έλεγχος
- Έλεγχος ρύπανσης υδάτων

Γεωργία

- Παρακολούθηση της αλυσίδας προσφοράς τροφίμων
- Παρακολούθηση των ζώων
- Φυτοπροστασία

Διαχείριση λιανικής

- Έλεγχος ραφιών
- Καλάθι αγορών
- Εντοπισμός κατεύθυνσης εντός του καταστήματος
- Μηχανές πώλησης
- Ετικέτες εμπορευμάτων
- Κινητά τηλέφωνα που θα αλληλεπιδρούν με τα συστήματα πληρωμών

Ασφάλεια

- Ανίχνευση πλαστών εμπορευμάτων
- Έλεγχος πρόσβασης
- Διαβατήρια

Κυβέρνηση και δημόσιος τομέας

- Διαχείριση καταστροφών
- Διαχείριση πυρκαγιάς

Σπίτι

- Απομακρυσμένη παρακολούθηση κλειδαριών
- Απομακρυσμένες οικιακές μετρήσεις
- Δίκτυο οικιακών συσκευών
- Έλεγχος τροφίμων (Διαδίκτυο- ψυγείο)
- Αυτόματα παραθυρόφυλλα

Υγεία

- Παρακολούθηση ασθενών στο σπίτι
- Ταυτοποίηση και παρακολούθηση φαρμάκων
- Παρακολούθηση της ζωής των ασθενών (εντατική θεραπεία)

3 Μηχανισμοί αναζήτησης προτιμήσεων

Η ραγδαία ανάπτυξη του Διαδικτύου όπως είδαμε προσφέρει ένα ευρύ φάσμα δυνατοτήτων ταυτόχρονα όμως σηματοδοτεί την είσοδο σε μια νέα περίοδο πληροφορίας. Ο παγκόσμιος Ιστός παρέχει ένα καινούργιο μέσο επικοινωνίας το οποίο ξεπερνά κατά πολύ τα παραδοσιακά μέσα ,όπως το ραδιόφωνο , το τηλέφωνο και την τηλεόραση. Ο Ιστός έχει σημαντικό αντίκτυπο τόσο στην ακαδημαϊκή έρευνα όσο και στην καθημερινή ζωή. Έχει φέρει επανάσταση στον τρόπο με τον οποίο η πληροφορία συλλέγεται, αποθηκεύεται, επεξεργάζεται, παρουσιάζεται, μοιράζεται και χρησιμοποιείται. Δεδομένα βρίσκονται άφθονα στο Διαδίκτυο και είναι εύκολα προσβάσιμα από όποιον τα ζητήσει.

Το γεγονός όμως ότι οι πληροφορίες γενικά είναι εύκολα προσβάσιμες δεν σημαίνει πως είναι και εύκολα αναζητήσιμες οι πληροφορίες που αναζητάμε. Η τεράστια ποσότητα πληροφοριών που υπάρχει στο διαδίκτυο είναι δυνατόν να οδηγήσει σε αχρήστευση πολύ σημαντικού μέρους των πληροφοριών εφόσον μένει ανοργάνωτη, καθώς είναι δύσκολο να εντοπισθεί και να χρησιμοποιηθεί από τον χρήστη. Είναι πολύ σημαντικό η αναζήτηση να γίνεται με αποδοτικό τρόπο ώστε να είναι γρήγορη και οικονομική. Για το λόγο αυτό έχουν αναπτυχθεί τεχνικές αναζήτησης που βασίζονται σε ένα συνδυασμό μοντελοποίησης των προτιμήσεων συγκεκριμένων χρηστών , στη δημιουργία προτύπων περιεχομένου και στη μοντελοποίηση κοινωνικών προτύπων .

3.1 Μηχανισμοί καταγραφής προτιμήσεων χρηστών

Για να επιτευχθεί μια αναζήτηση με αποδοτικό τρόπο , είναι πολύ σημαντικό να αξιοποιηθούν αποδοτικά τα ιδιαίτερα χαρακτηριστικά κάθε χρήστη (προφίλ). Τα μοντέλα και οι τεχνικές που ακολουθούνται κάθε φορά εξαρτώνται από τις ιδιαίτερες ανάγκες του κάθε συστήματος. Γενικά μπορούν να αναγνωρισθούν τα ακόλουθα δομικά στοιχεία:

- Αναπαράσταση Προφίλ
- Αρχικό Προφίλ

- Τεχνικές εκμάθησης προφίλ
- Ανατροφοδότηση σχετικότητας
- Τεχνικές προσαρμογής προφίλ
- Μέθοδοι φιλτραρίσματος πληροφοριών
- Ταίριασμα στοιχείων- προφίλ χρήστη
- Ταίριασμα παραμέτρων χρήστη

Η παραγωγή και συντήρηση ενός ακριβούς προφίλ αποτελούν ένα βασικό τμήμα του συστήματος αναζήτησης. Η επιλογή της κατάλληλης αναπαράστασης προηγείται κάθε άλλης ενέργειας, αφού οι άλλες τεχνικές βασίζονται σε αυτή. Άλλωστε, το σύστημα δεν μπορεί να λειτουργήσει αν δεν υπάρχουν παράμετροι χρήστη. Επιπλέον, το σύστημα χρειάζεται να γνωρίζει όσο το δυνατόν περισσότερα για το χρήστη ώστε να είναι σε θέση να του παρέχει ικανοποιητικά αποτελέσματα από την αρχή. Για αυτό το λόγο είναι συνήθως απαραίτητες και τεχνικές που δημιουργούν ένα αρχικό προφίλ.

Η συνεχής αλληλεπίδραση του χρήστη με το σύστημα είναι απαραίτητη όχι μόνο για την αναγνώριση των χαρακτηριστικών γνωρισμάτων του χρήστη και την αξιολόγηση των προτάσεων αλλά και για την προσαρμογή του προφίλ αφού είναι δυνατόν οι προτιμήσεις ενός ατόμου να αλλάζουν με την πάροδο του χρόνου. Οι πληροφορίες αυτές αποτελούν την ανατροφοδότηση σχετικότητας και μπορεί να εξαχθούν είτε με άμεσο τρόπο όπως, ερωτήσεις προς το χρήστη, είτε με έμμεσο, όπως με το χρόνο που αυτός διαθέτει σε μια ιστοσελίδα.

Για να αξιοποιηθούν οι παραπάνω πληροφορίες χρησιμοποιούνται τεχνικές εκμάθησης προφίλ, οι οποίες αναγνωρίζουν τις σχετικές πληροφορίες και τις χρησιμοποιούν για την προσαρμογή των προφίλ, ανάλογα βέβαια με την αναπαράσταση.

Εφόσον λοιπόν έχουν καθοριστεί τα παραπάνω για το προφίλ, ακολουθεί η εκμετάλλευσή του προκειμένου να προταθούν αντικείμενα ή υπηρεσίες. Οι αποφάσεις για τις προτάσεις που θα γίνουν λαμβάνονται σύμφωνα με τις υπάρχουσες πληροφορίες, και για την ακρίβεια, σύμφωνα με το χρήσιμο τμήμα αυτών των πληροφοριών. Υπάρχει δηλαδή η ανάγκη διήθησης των

πληροφοριών ώστε η ποιότητα των συστάσεων να είναι καλύτερη αλλά και να επιτύχουμε οικονομία χώρου και χρόνου. Οι τεχνικές που χρησιμοποιούνται συνήθως είναι η δημογραφική, η συνεργατική και η ανάλυση περιεχομένου. Η δημογραφική αντιστοιχεί αντικείμενα προτάσεων σε τύπους χρηστών. Η συνεργατική λαμβάνει υπόψη την ανατροφοδότηση άλλων χρηστών και η μέθοδος που βασίζεται στο περιεχόμενο αναγνωρίζει τη σχέση ενός συγκεκριμένου χρήστη με το περιεχόμενο των αντικειμένων. Κάθε μέθοδος έχει πλεονεκτήματα και μειονεκτήματα. Συνήθως οι μέθοδοι συνδυάζονται για να εξουδετερωθούν τα μειονεκτήματά τους.

3.2 Καταγραφή προτιμήσεων χρηστών στην εφαρμογή που αναπτύχθηκε

Στην εφαρμογή που έχει υλοποιηθεί και παρουσιάζεται παρακάτω έχει χρησιμοποιηθεί η τεχνική του Αρχικού Προφίλ.

Στο χρήστη δίνεται η δυνατότητα δημιουργίας προφίλ όπου δηλώνει τα αντικείμενα ενδιαφέροντος του. Με αυτόν τον τρόπο η εφαρμογή έχει όλες τις πληροφορίες για τον χρήστη από την αρχή. Επομένως η αναζήτηση γίνεται εξ' αρχής στο χρήσιμο κομμάτι της πληροφορίας κάτι που κάνει την εφαρμογή γρήγορη και αποδοτική.

Τέλος λαμβάνεται υπ' όψιν ότι τα ενδιαφέροντα του χρήστη μπορεί να αλλάξουν για το λόγο αυτό η εφαρμογή είναι σχεδιασμένη έτσι ώστε να δίνει την ευκαιρία να έχει και πάλι πρόσβαση στο προφίλ του για τυχόν αλλαγές και εκ νέου αναζήτηση.

3.3 Υπάρχοντα συστήματα

Όπως αναφέρθηκε παραπάνω στόχος των νέων εφαρμογών είναι να είναι αποδοτικές χρονικά και οικονομικά. Οι εφαρμογές που στηρίζονται στην αναζήτηση χρησιμοποιούν τεχνικές που αναφέρθηκαν παραπάνω ώστε να βελτιώσουν την αναζήτηση τους και να γίνουν και οι ίδιες πιο αποδοτικές. Η εφαρμογή που αναπτύχθηκε και περιγράφεται παρακάτω έχει να κάνει με αναζήτηση καταστημάτων. Αυτή την στιγμή βρίσκεται διαθέσιμο στην αγορά ένα μεγάλο πλήθος εφαρμογών που απευθύνονται σε

χρήστες που θέλουν να βελτιώσουν τον τρόπο των αγορών τους. Ενδεικτικά αναφέρονται μερικές παρακάτω.

Fweebi app : Η εφαρμογή αυτή παρουσιάζει όλες τις διαθέσιμες προσφορές από τα αγαπημένα προϊόντα ή καταστήματα από οποιαδήποτε περιοχή και αν βρίσκεται ο χρήστης. Ο χρήστης επιλέγει τα brands που τον ενδιαφέρουν , την περιοχή στην οποία βρίσκεται και η εφαρμογή του επιστρέφει τις προσφορές που υπάρχουν κοντά μέσα από διαδραστικούς χάρτες.

We Shopper app : Η εφαρμογή αυτή μέσα από μία λίστα που δημιουργεί ο χρήστης κάθε φορά για τις αγορές του, «μαθαίνει» τις συνήθειες του και η λίστα αγορών δημιουργείται αυτόματα από την εφαρμογή όταν χρειάζεται. Όσο πιο πολλές φορές χρησιμοποιείται η εφαρμογή τόσο πιο γρήγορη και αποδοτική γίνεται.

Shopping List app : Η εφαρμογή αυτή βοηθά τον καταναλωτή να οργανώσει τα ψώνια του σε μία ηλεκτρονική λίστα η οποία διατηρείται στο κινητό του. Παράλληλα δίνεται η δυνατότητα προσθήκης εικόνων η και σημειώσεων για περαιτέρω βοήθεια των χρηστών.

Out of Milk Shop List : Η εφαρμογή αυτή κρατά την ηλεκτρονική λίστα για ψώνια «Shopping List» ενήμερη. Έτσι ο χρήστης γνωρίζει τι υπάρχει σπίτι του ανά πάσα στιγμή και τι έχει τελειώσει ώστε να το αγοράσει.

4 Επιλογή Android

Στα πλαίσια αυτής της πτυχιακής εργασίας, μελετήθηκαν διάφορες τεχνολογίες οι οποίες θα μπορούσαν να χρησιμοποιηθούν στον σχεδιασμό και την υλοποίηση μιας εφαρμογής για κινητές συσκευές. Θα δούμε τα βασικά χαρακτηριστικά του Android, μιας στοίβας λογισμικού για κινητές συσκευές.

4.1 Επιλογή του Android

Ένας από τους πρωταρχικούς σκοπούς σε αυτήν την εργασία, ήταν η επιλογή μιας πλατφόρμας, η οποία θα μπορούσε να υποστηρίξει τον σχεδιασμό και την υλοποίηση μιας εφαρμογής. Ταυτόχρονα αναζητούσαμε κάτι το καινούργιο στον χώρο που θα άξιζε την μελέτη του και παράλληλα θα ήταν πολλά υποσχόμενο. Μέσα σε λίγα λεπτά καταλήξαμε σε δύο υποψήφιες πλατφόρμες, το Android και το iPhone. Οι δύο αυτές πλατφόρμες φαίνεται ότι θα πρωταγωνιστήσουν στον χώρο των έξυπνων τηλεφώνων (smart phones), αφού διαθέτουν εξαιρετικά χαρακτηριστικά και απίστευτες δυνατότητες [Εικόνα 1].



Εικόνα 1: Λογότυπο πλατφόρμας Android

Πηγή (<http://socialcompare.com/en/comparison/android-versions-comparison>)

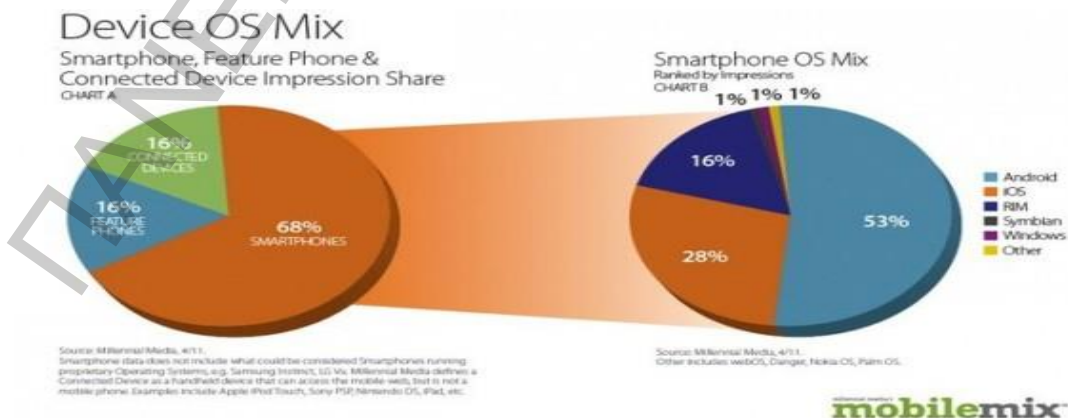
Στις αρχές Οκτωβρίου 2009, το iPhone κατείχε τα σκήπτρα στις πωλήσεις. Παρόλα αυτά, με τη ραγδαία ανάπτυξη που είχε το Android, οι ερευνητές υποστήριζαν ότι τα δεδομένα θα έχουν ανατραπεί μέχρι το 2012 το αργότερο. Προς έκπληξη αρκετών, το Android έχει ήδη ξεπεράσει προ πολλού σε πωλήσεις το iPhone και συνεχίζει να επεκτείνεται. Ωστόσο, είναι μια μάχη η οποία αναμένεται να συνεχιστεί, με τις δύο πλατφόρμες να εκσυγχρονίζουν ανά τακτά χρονικά διαστήματα τα χαρακτηριστικά τους.



Εικόνα 2: Η μάχη μεταξύ Android και Iphone

Πηγή (<http://blog.laptopmag.com/android-vs-ios-and-the-real-winner-is>)

Χαρακτηριστική είναι η έρευνα που ανακοίνωσε η Millennial Media. Τα αποτελέσματα είναι αναμενόμενα, δηλαδή το Android OS κυριαρχεί με παρουσία στο 53% των smartphones παγκοσμίως, με το iOS να ακολουθεί με 28% και το BlackBerry OS με 16% [Εικόνα 4].



Εικόνα 3: Πωλήσεις Λειτουργικών Συστημάτων για τα Smartphones

Πηγή (<http://www.mobilemarketingwatch.com/millennial-medias-may-2011-mobile-mix-report-shows-major-gains-across-the-mobile-landscape-16274/>)

Επίσης, ο Hugo Barra (Product Management Director του Android ανακοίνωσε στο πρόσφατο Google I/O που έγινε στις 10 Μαΐου 2011 στο San Francisco, ορισμένα εντυπωσιακά στατιστικά για την πλατφόρμα. Η Google δεν έχασε την ευκαιρία να ρίξει τα «καρφιά» της σε οποιοδήποτε αμφισβητεί την κυριαρχία του Android OS και ανακοίνωσε τα ιλιγγιώδη στατιστικά στοιχεία που συνοδεύουν το υπέρ-επιτυχημένο λειτουργικό σύστημα, τα οποία παρατίθενται παρακάτω:

- 400.000 νέες ενεργοποιήσεις κάθε ημέρα
- 4.5 δισεκατομμύρια downloads εφαρμογών
- 100 εκατ. ενεργοποιήσεις από την πρώτη ημέρα κυκλοφορίας του Android OS
- 200.000 εφαρμογές στο Android Market
- 215 πάροχοι σε 112 χώρες
- 36 κατασκευαστές συσκευών

Οι δυνατότητες και οι προοπτικές του Android ήταν το έναυσμα, ώστε να γίνει ξεκάθαρη η επιλογή της πλατφόρμας ανάπτυξης της εφαρμογής, έναντι του λειτουργικού iOS. Μεγάλο πλεονέκτημα του λειτουργικού Android είναι ότι παρέχει τη δυνατότητα στον καθένα να γράψει τη δική του εφαρμογή, να μάθει, να πειραματιστεί με το κινητό του και γιατί όχι, να παίξει! Είναι ένα πρόγραμμα ανοικτού κώδικα.

4.2 Τι είναι το Android;

Το Android είναι ένα λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας, το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Οι συνιδρυτές της Android πήγαν να εργαστούν στην Google

συμπεριλαμβανομένων των Andy Rubin (συν-ιδρυτής της Danger), Rich Miner(συν-ιδρυτής της Wildfire Communications, Inc), Nick Sears (πρώην αντιπρόεδρος της T-Mobile), και Chris White (επικεφαλής σχεδιασμού και ανάπτυξης interface στο WebTV). Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας. Στην Google, η ομάδα με επικεφαλής τον Rubin ανέπτυξε μια κινητή πλατφόρμα που στηρίζεται στον πυρήνα του Linux, την οποία προώθησαν με την παροχή ενός ευέλικτου, αναβαθμίσιμου συστήματος. Έχει αναφερθεί ότι η Google έχει ήδη συγκεντρώσει μια σειρά από εταίρους hardware και software και επισήμανε στους παρόχους ότι ήταν ανοικτή σε διάφορους βαθμούς συνεργασίας εκ μέρους της. Περισσότερες εικασίες ότι η Google θα εισέλθει στην αγορά κινητής τηλεφωνίας άρχισαν τον Δεκέμβριο του 2006. Δημοσιεύσεις από το BBC και τη The Wall Street Journal πληροφορούσαν ότι η Google ήθελε την έρευνα και τις εφαρμογές σε κινητά τηλέφωνα και εργάζονται σκληρά για να τις προωθήσουν στην αγορά. Έντυπα και ηλεκτρονικά μέσα ενημέρωσης σύντομα ανέφεραν φήμες ότι η Google ανέπτυξε μια Google-branded συσκευή. Περισσότερες φήμες ακολούθησαν, αναφέροντας ότι η Google καθόριζε τις τεχνικές προδιαγραφές και έδειχνε πρωτότυπα στους κατασκευαστές κινητών τηλεφώνων και τους φορείς δικτύων. Τον Σεπτέμβριο του 2007, η InformationWeek κάλυψε μια μελέτη αξιολόγησης αναφέροντας ότι η Google έχει καταθέσει αρκετές πατέντες στον τομέα της κινητής τηλεφωνίας. Τελικά η Google παρουσίασε το smartphone της Nexus One που χρησιμοποιεί το open source λειτουργικό σύστημα Android. Η συσκευή κατασκευάστηκε από την HTC Corporation της Ταϊβάν, και έγινε διαθέσιμη στις 5 Ιανουαρίου 2010. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών

λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

Ενδεικτικά, αναφέρονται μερικά μέλη του οργανισμού αυτού [Εικόνα 4], για να γίνει ορατή η τεράστια προοπτική που δημιουργείται:

Operator	Handset Makers	Software Companies	Commercialization Companies	Semiconductor Companies
SoftBank 中国移动通信 O-PA MOBILE	htc smart mobility	Ascender Corporation ebay	Aplix Corporation	Audience Broadcom
DoCoMo KDDI	LG Electronics	Google myriad	BORQS	Intel Marvell
Sprint	MOTOROLA	Living Image Ivewire	noser	NVIDIA Qualcomm
Mobile	SAMSUNG	NUANCE pv	tat	SIRF Synaptics
vodafone	GARMIN	SONI VOX great their rocks	TELECA	TEXAS INSTRUMENTS ST ERICSSON
Telefonica	HUAWEI	OMRON SVOX	WIND RIVER	AKM ATHEROS
TELECOM	Sony Ericsson			
China unicom	acer			
	ASUS			
	TOSHIBA			

Εικόνα 4: Εταιρίες ανάπτυξης λογισμικού και κατασκευής υλικού παγκόσμιας εμβέλειας

Πηγή(<http://www.butterscotch.com/news/152/Open-Handset-Alliance-Band-Together-To-Solve-Android-Fragmentation>)

Μερικές από τις εταιρίες αυτές αναγράφονται παρακάτω:

- Intel
- Vodafone
- Motorola
- Samsung
- Sony Ericsson
- ebay
- Google
- Verizon
- Texas Instruments

- Htc

Κατά τα λεγόμενα τους, όπως αναφέρεται και στο επίσημο site, η OHA(Open Handset Alliance) αντιπροσωπεύει: «Μια δέσμευση για την ειλικρίνεια, ένα κοινό όραμα για το μέλλον και συγκεκριμένα σχέδια για να κάνει το όραμα μια πραγματικότητα. Για να επιταχύνει την καινοτομία στο κινητό και να προσφέρει στους καταναλωτές μια πλουσιότερη, λιγότερο ακριβή και καλύτερη εμπειρία κινητών τηλεφώνων.»

Η OHA ελπίζει να παραδώσει μια καλύτερη εμπειρία λογισμικού κινητών στους καταναλωτές, παρέχοντας μια πλατφόρμα για την ανάπτυξη καινοτόμων κινητών εφαρμογών γρηγορότερα και με υψηλότερη ποιότητα, χωρίς τέλη αδείας για τους προγραμματιστές λογισμικού ή τους κατασκευαστές κινητών τηλεφώνων. Η επιτυχία του Android ως πλατφόρμα κινητών τηλεφώνων θα εξαρτηθεί κατά ένα μεγάλο μέρος από την επιτυχία των συνεργατών της OHA στην κυκλοφορία επιθυμητών κινητών τηλεφώνων και κινητών υπηρεσιών που θα ενθαρρύνουν την υιοθέτηση των Android τηλεφώνων. Οι προγραμματιστές έχουν την ευκαιρία να δημιουργήσουν καινοτόμες, νέες εφαρμογές κινητών για Android ώστε να ενθαρρυνθούν περισσότερες επιχειρήσεις κινητής τεχνολογίας να γίνουν μέλη της OHA.



Εικόνα 5: Λογότυπο πλατφόρμας Android

Πηγή(<https://www.google.gr/search?q=green+logo+android+platform&newwindow=1&tbm=isch&tbo=u&source=univ&sa=X&ei=NLYzUZtFw6vRBcKXgcAG&ved=0CG8QsAQ&biw=1366&bih=638#imgsrc=kX0fUiyqQqLhEM%3A%3BCPwGa6r1qkwPLM%3Bhttp%253A%252F%252Fi214.photobucket.com%252Falbums%252Fcc308%252FvNomadv%252FArchos%2525205%252520Android%252FAndroidGreenLogo.jpg%3Bhttp%253A%252F%252Fforum.archosfans.com%252Fviewtopic.php%253Fp%253D178204%3B1023%3B682>)

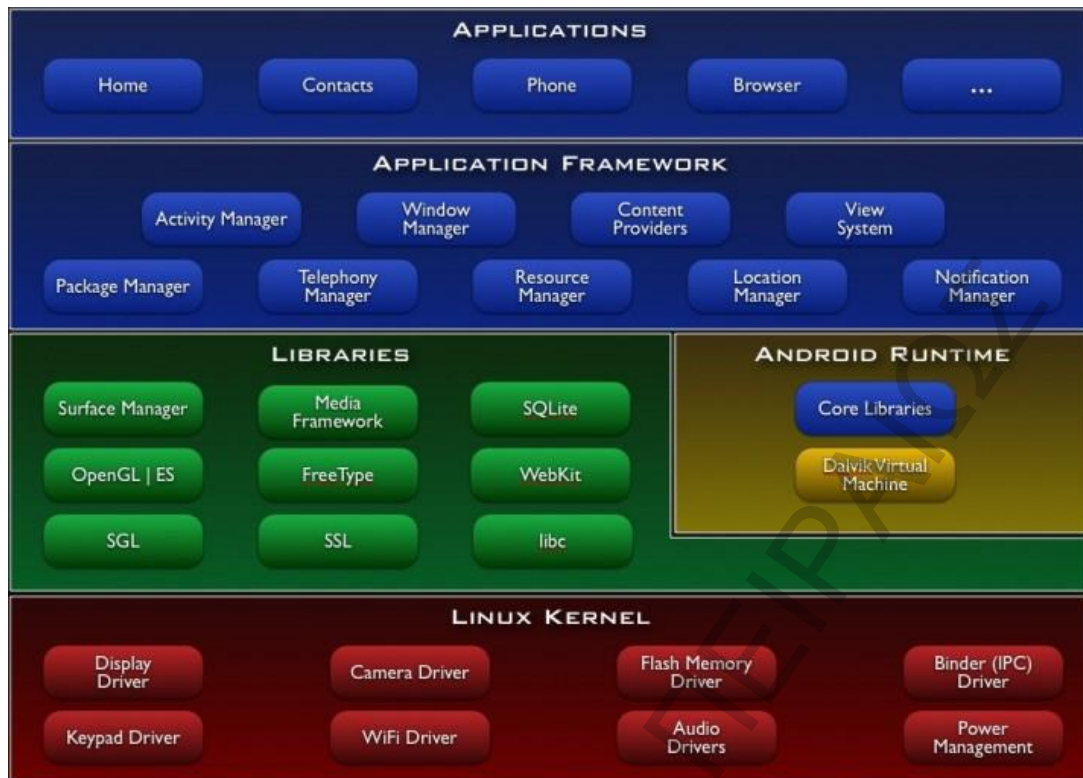
4.3 Αρχιτεκτονική του Android

Όπως αναφέραμε πιο πριν, το Android είναι μια στοίβα λογισμικού. Η λογική πίσω από αυτήν την έκφραση και σε όλη την φιλοσοφία του Android, κρύβεται στο ακόλουθο διάγραμμα με τα βασικά συστατικά του [Εικόνα 7].

Η αρχιτεκτονική του Android ενθαρρύνει την έννοια επαναχρησιμοποίησης κώδικα, που επιτρέπει να δημοσιευτούν και να μοιραστούν δραστηριότητες, υπηρεσίες και δεδομένα με άλλες εφαρμογές με πρόσβαση που ελέγχεται από τους περιορισμούς ασφάλειας που έχουν τεθεί. Έτσι γίνεται δυνατή η επέκταση και βελτιστοποίηση υπάρχοντων εφαρμογών ή δημιουργίας καινούργιων, χρησιμοποιώντας κώδικα από αυτές.

Οι ακόλουθες υπηρεσίες εφαρμογών είναι οι ακρογωνιαίοι λίθοι της αρχιτεκτονικής όλων των εφαρμογών Android, παρέχοντας το πλαίσιο που θα χρησιμοποιηθεί για τη δημιουργία του λογισμικού :

- Διαχειριστής Δραστηριοτήτων (Activity Manager) - Ελέγχει τον κύκλο ζωής των δραστηριοτήτων, συμπεριλαμβανομένης της διαχείρισης του σωρού δραστηριοτήτων.
- Όψεις (Views) - Χρησιμοποιούνται για να κατασκευάσουν διεπαφές χρήστη (user interface) για τις δραστηριότητες (activities).
- Διαχειριστή Ειδοποιήσεων (Notification Manager) - Παρέχει ένα συνεπή και αποτελεσματικό μηχανισμό για να προειδοποιεί τον χρήστη
- Παροχέας Περιεχομένου (Content Providers) - Επιτρέπουν στις εφαρμογές να μοιραστούν δεδομένα μεταξύ τους.
- Διαχειριστής Πόρων (Resource Manager) - Υποστηρίζει πόρους εκτός κώδικα όπως συμβολοσειρές και γραφικά.



Εικόνα 6: Τα βασικά περιεχόμενα του λειτουργικού συστήματος Android

Πηγή (<http://www.android-app-market.com/android-architecture.html>)

Στην στοίβα του Android [Εικόνα 7], παρατηρούμε 4 επίπεδα. Ακολούθως θα περιγράψουμε συνοπτικά τα βασικά αυτά επίπεδα χωρίς να μπούμε σε λεπτομέρειες για όλα τα περιεχόμενα του κάθε επιπέδου. Αν ο αναγνώστης επιθυμεί να μάθει περισσότερα, μπορεί να επισκεφθεί την επίσημη ιστοσελίδα του Android για κατασκευαστές (<http://developer.android.com>). Κάθε επίπεδο στην αρχιτεκτονική αυτή, χρησιμοποιεί τις υπηρεσίες που του προσφέρονται από τα πιο πάνω επίπεδα. Ας δούμε τώρα αυτά τα επίπεδα ξεκινώντας από το πιο χαμηλό.

4.4 Η εικονική μηχανή Dalvik

Ένα από τα στοιχεία κλειδιά του Android είναι η εικονική μηχανή Dalvik. Το Android χρησιμοποιεί την δικιά του εικονική μηχανή και όχι μια παραδοσιακή, με σκοπό να εξασφαλίσει ότι πολλαπλά στιγμιότυπα τρέχουν αποτελεσματικά σε μια ενιαία συσκευή.

Η Dalvik VM (Virtual Machine) χρησιμοποιεί τον πυρήνα Linux της συσκευής για να χειριστεί τις χαμηλού επιπέδου λειτουργίες που περιλαμβάνουν την ασφάλεια, τον πολυνηματισμό και τη

διαχείριση διαδικασιών και μνήμης. Είναι επίσης δυνατό να γραφτούν εφαρμογές C/C++ που τρέχουν άμεσα στο εσωτερικό του λειτουργικού Linux. Αν και μπορεί να γίνει αυτό, στις περισσότερες περιπτώσεις δεν υπάρχει κανένας λόγος.

Μέσω της Dalvik VM επιτυγχάνεται η ρύθμιση της πρόσβασης στο υλικό και στις υπηρεσίες του συστήματος. Με τη χρησιμοποίηση αυτής της εικονικής μηχανής στην εκτέλεση εφαρμογής, η οποία προσφέρει ένα αφαιρετικό στρώμα, οι κατασκευαστές δεν χρειάζεται να ανησυχήσουν για κάποια υλοποίηση υλικού (hardware implementation).

Η Dalvik VM εκτελεί τα Dalvik εκτελέσιμα αρχεία, ένα format βελτιστοποιημένο έτσι ώστε να καταλαμβάνει την ελάχιστη μνήμη. Τα .dex εκτελέσιμα αρχεία δημιουργούνται μετασχηματίζοντας κλάσεις που έχουν μεταγλωττιστεί από Java χρησιμοποιώντας εργαλεία που παρέχονται μέσα στο SDK.

Μια απλή Java VM είναι μια εικονική μηχανή βασισμένη σε στοίβα (stack-based). Η Dalvik VM από την άλλη είναι μια εικονική μηχανή βασισμένη σε μητρώα (registerbased). Με τον τρόπο αυτό αυξάνεται η αποδοτικότητα του επεξεργαστή του κινητού. Επίσης, οι εικονικές μηχανές που είναι βασισμένες σε καταχωρητές (registers) επιτρέπουν ταχύτερους χρόνους εκτέλεσης των μεγάλων προγραμμάτων.

4.5 Πυρήνας Linux (Linux kernel)

Το Android είναι βασισμένο στα γερά θεμέλια του Linux. Ο πυρήνας Linux είναι δοκιμασμένος, σταθερός και πετυχημένος και μπορεί να βρεθεί παντού, από ρολόγια χειρός μέχρι υπερυπολογιστές. Το Linux παρέχει στο Android το αφαιρετικό επίπεδο υλικού, επιτρέποντας του να μπορεί να χρησιμοποιηθεί σε μεγάλη ποικιλία πλατφορμών στο μέλλον. Ειδικότερα, το Android χρησιμοποιεί τον πυρήνα Linux για την διαχείριση μνήμης, την διαχείριση διεργασιών, την δικτύωση και άλλες υπηρεσίες του λειτουργικού συστήματος.

4.6 Εγγενείς Βιβλιοθήκες (Native Libraries)

Στο αμέσως υψηλότερο επίπεδο βρίσκουμε τις Native Libraries – Εγγενείς Βιβλιοθήκες. Όλες αυτές είναι γραμμένες στην γλώσσα προγραμματισμού C και C++ και μεταγλωττίστηκαν για την συγκεκριμένη αρχιτεκτονική υλικού που χρησιμοποιείται από το τηλέφωνο. Οι βιβλιοθήκες αυτές δεν είναι εφαρμογές που μπορούν να στηθούν από μόνες τους. Υπάρχουν για να μπορούν να κληθούν από προγράμματα υψηλότερου επιπέδου. Από την έκδοση Donut και μετά, οι κατασκευαστές μπορούν να γράφουν τις δικές τους τέτοιες βιβλιοθήκες με την χρήση της Εργαλειοθήκης NDK (Native Development Kit).

4.6.1 Βιβλιοθήκες Android

Το Android προσφέρει διάφορα APIs για την ανάπτυξη εφαρμογών. Ο ακόλουθος κατάλογος APIs πυρήνα δίνει μια ιδέα για το τι είναι διαθέσιμο. Όλες οι συσκευές Android υποστηρίζουν το λιγότερο αυτά τα APIs :

- android.util - Το πακέτο βοηθήματος (utility) πυρήνων περιέχει χαμηλού επιπέδου κλάσεις, όπως τα εξειδικευμένα containers, μορφοποιητές (formatters) συμβολοσειρών, και βοηθήματα για parsing XML αρχείων.
- android.os - Το πακέτο λειτουργικού συστήματος παρέχει την πρόσβαση στις βασικές υπηρεσίες του λειτουργικού συστήματος όπως τη διαβίβαση μηνυμάτων, την επικοινωνία μεταξύ των επικοινωνιών, τις λειτουργίες ρολογιών, και την αποσφαλμάτωση (debugging).
- android.graphics - Το API γραφικών παρέχει τις χαμηλού επιπέδου κλάσεις γραφικών που υποστηρίζουν τους καμβάδες (canvases), τα χρώματα, και τα βασικά στοιχεία ζωγραφικής, και παρέχει τη δυνατότητα να ζωγραφίσει κανείς πάνω σε καμβά (canvas).

- `android.text` - Τα εργαλεία επεξεργασίας κειμένων για αναπαράσταση και ανάλυση κειμένου.
- `android.database` - Παρέχει τις χαμηλού επιπέδου κλάσεις που απαιτούνται για το χειρισμό των δρομέων (cursors) κατά τη λειτουργία τους με τις βάσεις δεδομένων.
- `android.content` - Το API περιεχομένου (content) χρησιμοποιείται για να διαχειριστεί την πρόσβαση στα δεδομένα και την έκδοσή τους, παρέχοντας υπηρεσίες για την διαχείριση των πόρων, των Παρόχων Περιεχομένου (Content Provider), και των πακέτων.
- `android.view` - Οι Όψεις (Views) είναι οι κλάσεις πυρήνα διεπαφών χρήστη. Όλες οι διεπαφές χρήστη κατασκευάζονται χρησιμοποιώντας μια σειρά από Όψεις που παρέχουν τα συστατικά της αλληλεπίδρασης χρηστών.
- `android.widget` - Τοποθετημένες στο πακέτο View, οι κλάσεις widget είναι τα στοιχεία της διεπαφής χρήστη (user-interface) που χρησιμοποιούνται στις εφαρμογές. Περιλαμβάνουν τις λίστες, τα κουμπιά, και τις διατάξεις (layouts).
- `com.google.android.maps` - Ένα υψηλού επιπέδου API που παρέχει πρόσβαση στις εγγενείς λειτουργίες χαρτών που μπορούν να χρησιμοποιηθούν μέσα σε εφαρμογές. Περιλαμβάνει τη λειτουργία `MapView`, καθώς επίσης και τις κλάσεις `Overlay` και `MapController` που χρησιμοποιούνται για να διαχειριστούν και να ελέγξουν τους ενσωματωμένους χάρτες.
- `android.app` - Ένα πακέτο υψηλού επιπέδου που παρέχει πρόσβαση στο μοντέλο εφαρμογών. Το πακέτο εφαρμογών περιλαμβάνει τα `Activity` και `Service` APIs που αποτελούν τη βάση για όλες τις Android εφαρμογές.
- `android.provider` - Για να διευκολύνει την πρόσβαση των προγραμματιστών σε ορισμένους τυποποιημένους Παρόχους Περιεχομένου (Content Providers) (όπως η βάση δεδομένων επαφών), προσφέρονται κλάσεις που παρέχουν πρόσβαση σε τυποποιημένες βάσεις δεδομένων που περιλαμβάνονται σε όλες τις εκδόσεις Android.

- `android.telephony` - Τα APIs τηλεφωνίας επιτρέπουν την άμεση αλληλεπίδραση με το τηλεφωνικό σωρό (`phone stack`) της συσκευής, δίνοντας τη δυνατότητα, να γίνει λήψη, και έλεγχος των τηλεφωνημάτων, της κατάστασης του τηλεφώνου, και των μηνυμάτων SMS.
- `android.webkit` - Το πακέτο `WebKit` περιλαμβάνει APIs για την εργασία με περιεχόμενο βασισμένο στο διαδίκτυο, συμπεριλαμβανομένης της λειτουργίας `wapree` για την ενσωμάτωση μηχανών αναζήτησης και ενός διαχειριστή για cookies. Εκτός από το `Android APIs`, ο σωρός του `Android` περιλαμβάνει ένα σύνολο βιβλιοθηκών `C/C++`. Αυτές οι βιβλιοθήκες περιλαμβάνουν:
 - `OpenGL` - Βιβλιοθήκη που χρησιμοποιείται για να υποστηρίξει 3D γραφικά βασισμένη στο `Open GL ES 1.0 API`.
 - `FreeType` - Υποστήριξη για τη `bitmap` και τη `vector` απόδοση γραμματοσειράς.
 - `SGL` - Βιβλιοθήκη πυρήνων που χρησιμοποιείται για να παρέχει μια μηχανή 2D γραφικών.
 - `libc` - Τυποποιημένη βιβλιοθήκη `C` που βελτιστοποιείται για συσκευές βασισμένες σε `Linux`.
 - `SQLite` - Ελαφριά μηχανή σχεσιακών βάσεων δεδομένων που χρησιμοποιείται για να αποθηκεύσει δεδομένα εφαρμογών.
 - `SSL` - Υποστήριξη για χρήση των `Secure Sockets Layer` του κρυπτογραφικού πρωτοκόλλου για ασφαλείς επικοινωνίες μέσω Διαδικτύου.

4.6.2 Προηγμένες Βιβλιοθήκες Android

Οι βιβλιοθήκες πυρήνα παρέχουν όλες τις λειτουργίες που είναι απαραίτητες για να ξεκινήσει η δημιουργία μιας εφαρμογής για `Android`, αλλά οι προηγμένες βιβλιοθήκες `APIs` προσφέρουν πραγματικά συναρπαστικές λειτουργίες.

Επειδή το Android στοχεύει να χρησιμοποιείται από ένα ευρύ φάσμα κινητών, η καταλληλότητα και η εφαρμογή των ακόλουθων APIs ποικίλουν ανάλογα με τη συσκευή επάνω στην οποία εφαρμόζονται.

- `android.location` - Το τοποκεντρικό API (location-based) δίνει πρόσβαση των εφαρμογών σε πληροφορίες για τη τρέχουσα φυσική θέση της συσκευής. Οι υπηρεσίες βασισμένες στη τοποθεσία παρέχουν πρόσβαση σε πληροφορίες τοποθεσίας χρησιμοποιώντας οποιοδήποτε υλικό ή τεχνολογία είναι διαθέσιμη στη συσκευή και βοηθούν στον προσδιορισμό θέσης.
- `android.media` - Τα APIs πολυμέσων παρέχουν υποστήριξη για την αναπαραγωγή και εγγραφή αρχείων ήχου και βίντεο, συμπεριλαμβανομένων των πολυμέσων ροής (streamed).
- `android.opengl` - Το Android προσφέρει μια ισχυρή αποδοτική 3D μηχανή που χρησιμοποιεί το OpenGL ES API με το οποίο δημιουργούνται 3D διεπαφές χρήστη για τις εφαρμογές.
- `android.hardware` - Όπου είναι διαθέσιμο, το API υλικού εκθέτει υλικό αισθητήρων συμπεριλαμβανομένης της φωτογραφικής μηχανής, του επιταχύμετρου (accelerometer) και των αισθητήρων πυξίδας.
- `android.bluetooth`, `android.net.wifi`, και `android.telephony` - Το Android επίσης παρέχει χαμηλού επιπέδου πρόσβαση στην πλατφόρμα υλικού, συμπεριλαμβανομένου του Bluetooth, του WI-FI, και του υλικού τηλεφωνίας.

4.7 Χρόνος Εκτέλεσης (Android Runtime)

Στο ίδιο επίπεδο με τις εγγενείς βιβλιοθήκες, βρίσκουμε και τον χρόνο εκτέλεσης Android.

Εδώ υπάρχουν οι βασικές βιβλιοθήκες της Java και η εικονική μηχανή Dalvik. Η Dalvik είναι μια βελτιστοποιημένη υλοποίηση μιας εικονικής μηχανής Java για φορητές συσκευές από την Google. Η Dalvik τρέχει `.dex` αρχεία, τα οποία είναι `bytecodes` που

προέρχονται από αρχεία .class και .jar. Εν αντιθέσει όμως με τα .class αρχεία, τα .dex είναι πολύ πιο συμπαγή και αποδοτικά, γεγονός σημαντικό για συσκευές με περιορισμένη μνήμη και μπαταρία. Το Android περιλαμβάνει ένα σύνολο βασικών βιβλιοθηκών που παρέχουν τις περισσότερες από τις διαθέσιμες λειτουργίες των βασικών βιβλιοθηκών της Java. Κάποια πακέτα και κλάσεις υπάρχουν και στο Android, κάποια άλλα δεν υποστηρίζονται καθόλου, ενώ ταυτόχρονα το Android παρέχει και επιπρόσθετα προσαρμοσμένα στις δικές του ανάγκες.

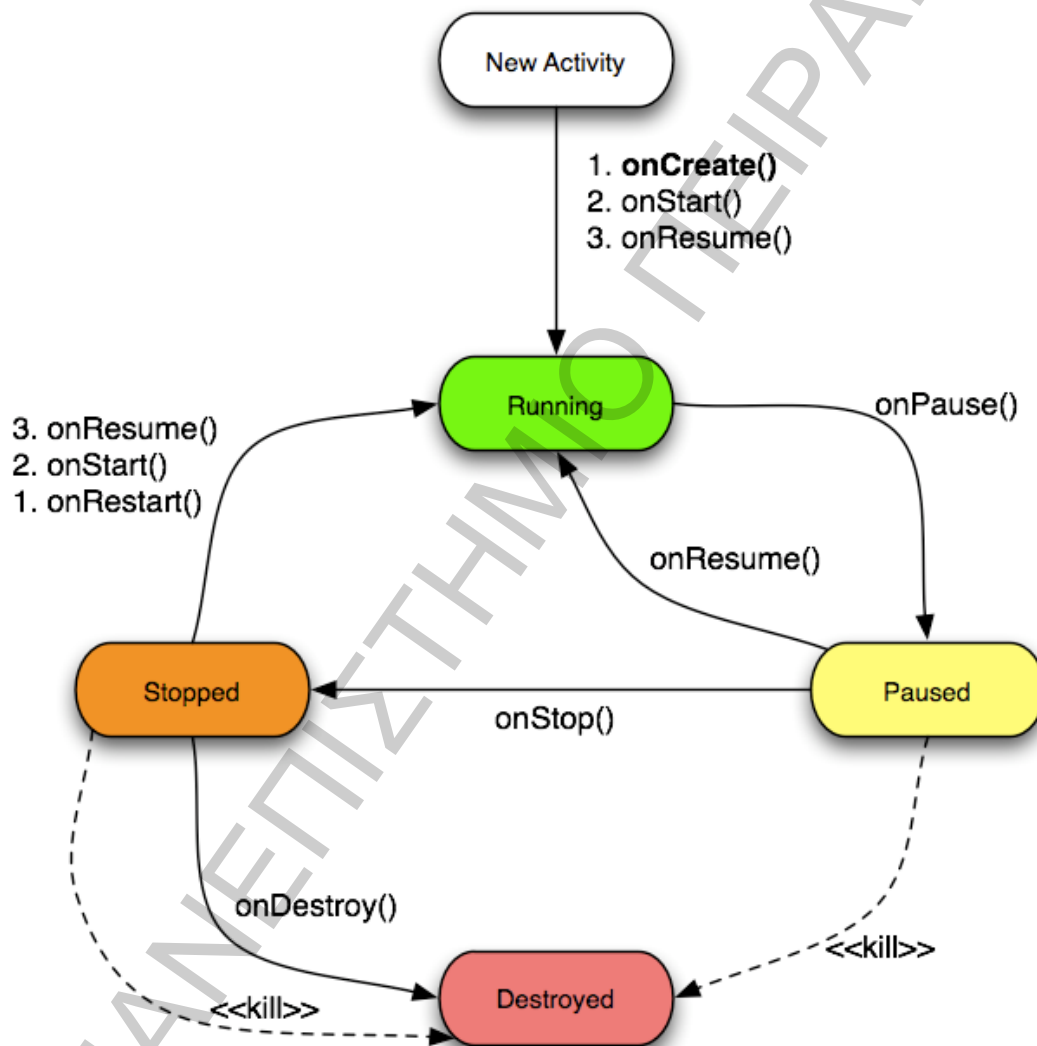
4.8 Πλαίσιο Εφαρμογής (Application Framework)

Πάνω από τις εγγενείς βιβλιοθήκες και το χρόνο εκτέλεσης Android, είναι το πλαίσιο εφαρμογής. Αυτό το επίπεδο παρέχει υψηλού επιπέδου δομικές μονάδες τις οποίες μπορούμε να χρησιμοποιούμε για την κατασκευή των εφαρμογών. Αυτό το πλαίσιο είναι προ-εγκατεστημένο στο Android, αλλά είναι επεκτάσιμο, αφού ο κάθε κατασκευαστής μπορεί να το συμπληρώσει με δικά του κομμάτια. Τα σημαντικότερα δομικά στοιχεία του πλαισίου αυτού είναι:

- Διαχειριστής Δραστηριοτήτων - Activity Manager: Υπεύθυνος για τον έλεγχο του χρόνου ζωής [Εικόνα 7] των εφαρμογών και για την διατήρηση μιας στοίβας που επιτρέπει την πλοήγηση του χρήστη σε προηγούμενες οθόνες.
- Παροχέας Περιεχομένου - Content Providers: Αυτά τα αντικείμενα περιέχουν δεδομένα που μπορούν να διαμοιραστούν μεταξύ εφαρμογών.
- Διαχειριστής Πόρων - Resource Manager: Οι πόροι είναι οτιδήποτε υπάρχει σε ένα πρόγραμμα και δεν είναι κώδικας. Για παράδειγμα μπορεί να είναι κωδικοί χρωμάτων, αλφαριθμητικοί χαρακτήρες ή ακόμα και έτοιμα σχεδιαγράμματα οθονών φτιαγμένα σε XML, τα οποία μπορεί το πρόγραμμα να καλεί.
- Διαχειριστής Τοποθεσίας - Location Manager: Χρησιμοποιείται για να μπορεί να ξέρει το τηλέφωνο που βρίσκεται ανά πάσα στιγμή.

- Διαχειριστής Κοινοποιήσεων - Notification Manager: Ιδανικός τρόπος για την ενημέρωση του χρήστη για γεγονότα που συμβαίνουν, διακριτικά χωρίς να διακόπτεται η εργασία του.

Activity Lifecycle



Εικόνα 7: Ο κύκλος ζωής μιας 3ραστηριότητας Android

Πηγή (<http://manisivapuram.blogspot.gr/2011/07/activity-life-cycle-example.html>)

4.9 Βασικά Συστατικά εφαρμογών του Android

Οι εφαρμογές Android αποτελούνται από κάποια συστατικά, που συνδέονται χρησιμοποιώντας ένα XML αρχείο, το AndroidManifest.xml του κάθε project, το οποίο περιγράφει κάθε συστατικό και πώς αλληλεπιδρά με τα άλλα. Υπάρχουν έξι συστατικά που παρέχουν τις δομικές μονάδες για τις εφαρμογές:

- Δραστηριότητα (Activity) – Είναι το επίπεδο παρουσίασης (presentation layer) της εφαρμογής. Μια δραστηριότητα είναι μια απλή οθόνη της εφαρμογής. Κάθε δραστηριότητα υλοποιείται σαν μια κλάση που επεκτείνει (extends) την βασική κλάση δραστηριότητα (Activity base class). Η συγκεκριμένη κλάση προβάλλει μια διεπαφή χρήστη (user interface) αποτελούμενη από Όψεις (Views) και ανταποκρίνεται σε Συμβάντα (Events). Ένα Activity είναι ισοδύναμο με μια Φόρμα (Form) στην ανάπτυξη desktop εφαρμογών.

- Υπηρεσία (Service) - Μια Υπηρεσία είναι κώδικας που τρέχει για μεγάλο χρονικό διάστημα και χωρίς διεπαφή χρήστη (UI), ενημερώνοντας τις πηγές δεδομένων και τις ορατές δραστηριότητες (Activities), ενεργοποιώντας Ειδοποιήσεις (Notifications). Αυτές τρέχουν και επεξεργάζονται δεδομένα ακόμα και όταν οι δραστηριότητες της εφαρμογής δεν είναι ενεργές ή ορατές. Ένα καλό παράδειγμα είναι μια εφαρμογή που αναπαράγει μουσική από μια λίστα μουσικών κομματιών (media player). Σε μια τέτοια εφαρμογή, θα υπήρχαν κατά πάσα πιθανότητα μία ή και παραπάνω δραστηριότητες που επιτρέπουν στον χρήστη να επιλέξει τραγούδια και να τα αναπαράγει. Ωστόσο, η αναπαραγωγή από μόνη της δεν θα έπρεπε να διαχειρίζεται από την δραστηριότητα γιατί ο χρήστης θα περίμενε την μουσική να συνεχίζει να παίζει ακόμη και μετά την πλοήγησή του σε μια νέα οθόνη. Σε αυτή τη περίπτωση, η δραστηριότητα της αναπαραγωγής μουσικής θα ξεκινούσε μια Υπηρεσία (Service) για να τρέξει στο παρασκήνιο και να συνεχίσει η μουσική να παίζει. Το σύστημα τότε θα κρατά την Υπηρεσία αναπαραγωγής ενεργή μέχρι να τελειώσει το κομμάτι. Όταν πραγματοποιηθεί σύνδεση σε μια Υπηρεσία, μπορεί να υπάρξει επικοινωνία με αυτή μέσω μιας διεπαφής που προσφέρεται από την Υπηρεσία μουσικής. Για την

Υπηρεσία μουσικής, αυτό θα επέτρεπε την παύση ή την επιστροφή πίσω στο κομμάτι (rewind) κλπ.

- Παροχέας Περιεχομένου (Content Provider) - Οι εφαρμογές μπορούν να σώζουν τα δεδομένα τους σε αρχεία, σε μια βάση δεδομένων SQLite ή με οποιοδήποτε άλλο μηχανισμό μπορούν. Ένας Παροχέας Περιεχομένου, ωστόσο, είναι χρήσιμος ώστε τα δεδομένα μιας εφαρμογής να είναι διαθέσιμα και σε άλλες εφαρμογές. Ένας Παροχέας Περιεχομένου είναι μια κλάση που υλοποιεί μια συγκεκριμένη ομάδα μεθόδων που επιτρέπουν σε άλλες εφαρμογές να αποθηκεύουν και να επανακτούν δεδομένα του τύπου που διαχειρίζεται ο Παροχέας Περιεχομένου. Οι συσκευές Android περιλαμβάνουν διάφορους εγγενείς Παροχείς Περιεχομένου (Content Providers) που εκθέτουν τις χρήσιμες βάσεις δεδομένων, όπως για παράδειγμα των στοιχείων των επαφών του χρήστη.

- Προθέσεις (Intents) - Το Android χρησιμοποιεί μια ειδική κλάση που λέγεται Πρόθεση (Intent) για να κινείται από οθόνη σε οθόνη. Η Πρόθεση περιγράφει τι θέλει η εφαρμογή να γίνει στη συνέχεια. Τα δυο πιο σημαντικά μέρη της δομής δεδομένων της Πρόθεσης είναι η δράση (Action) και τα δεδομένα βάσει των οποίων αυτή θα εκτελεστεί. Τυπικές τιμές για μια δράση είναι η MAIN (η κεντρική είσοδος της εφαρμογής), VIEW, PICK, EDIT κλπ. Τα δεδομένα εκφράζονται ως URI (Uniform Resource Indicator). Για παράδειγμα, για να εμφανιστεί μια ιστοσελίδα στον φυλλομετρητή (browser), δημιουργείται μια Πρόθεση με δράση VIEW και τα δεδομένα ως ένα URL.

New Intent (android.content.Intent.VIEW_ACTION);

ContentURI.create("http://addev.org");

Υπάρχει μια σχετική κλάση που λέγεται Φίλτρο Πρόθεσης (IntentFilter). Ενώ μια Πρόθεση είναι στην ουσία ένα αίτημα για να γίνει κάτι, το Φίλτρο Πρόθεσης είναι μια περιγραφή του τι είναι δυνατόν να διαχειριστεί ένας δέκτης Πρόθεσης (Intent Receiver).

Μια δραστηριότητα που είναι σε θέση να προβάλει πληροφορίες επικοινωνίας για ένα άτομο, θα ανακοινώνει με ένα Φίλτρο Πρόθεσης (IntentFilter) ότι γνωρίζει πως να διαχειριστεί την VIEW_ACTION όταν τα δεδομένα αντιπροσωπεύουν ένα άτομο. Οι δραστηριότητες ανακοινώνουν τα Φίλτρα Πρόθεσης (IntentFilters) στο AndroidManifest.xml αρχείο. Η πλοήγηση από οθόνη σε οθόνη πετυχαίνεται με Προθέσεις. Για να πλοηγηθεί κανείς προς τα μπρος, μια δραστηριότητα καλεί την startActivity(myIntent). Το σύστημα τότε κοιτά στα Φίλτρα Προθέσεων (Intent Filters) για όλες τις εγκατεστημένες εφαρμογές και διαλέγει την δραστηριότητα που τα Φίλτρα Πρόθεσης ταιριάζουν καλύτερα με την παράμετρο 'MyIntent' της κλήσης. Η νέα δραστηριότητα ενημερώνεται για την Πρόθεση και ξεκινά. Η διαδικασία της υλοποίησης των Προθέσεων συμβαίνει κατά τον χρόνο εκτέλεσης της εφαρμογής, όταν δηλαδή καλείται η startActivity, πράγμα που προσφέρει 2 πλεονεκτήματα – κλειδιά:

1. Οι δραστηριότητες μπορούν να επαναχρησιμοποιούν κάποια λειτουργικότητα από άλλα τμήματα του κώδικα απλά κάνοντας ένα αίτημα υπό την μορφή μιας πρόθεσης.

2. Οι δραστηριότητες μπορούν να αντικατασταθούν οποιαδήποτε στιγμή από μια νέα δραστηριότητα με ένα αντίστοιχο Φίλτρο Πρόθεσης.

- Δέκτες Πρόθεσης (Intent Receiver) - Χρησιμοποιούνται για να εκτελεστεί μια εφαρμογή σε απάντηση ενός εξωτερικού συμβάντος (external event), για παράδειγμα, όταν το τηλέφωνο χτυπά, ή όταν το δίκτυο είναι διαθέσιμο, ή όταν είναι μεσάνυχτα. Οι δέκτες Πρόθεσης δεν προβάλλουν μια διεπαφή χρήστη (UI), ωστόσο μπορούν να προβάλουν Ειδοποιήσεις (Notifications) για να ειδοποιήσουν τον χρήστη για κάτι σημαντικό που συνέβη. Οι δέκτες Πρόθεσης είναι επίσης καταχωρημένοι στο AndroidManifest.xml, αλλά μπορούν επίσης καταχωρηθούν από τον κώδικα χρησιμοποιώντας την Context.registerReceiver(). Η εφαρμογή δεν χρειάζεται να τρέχει για να κληθούν οι έκτες Πρόθεσης που έχει. Το σύστημα θα κινήσει την εφαρμογή, αν χρειαστεί, όταν ένας δέκτης Πρόθεσης ενεργοποιηθεί. Οι εφαρμογές μπορούν επίσης να στέλνουν τις δικές τους

Ανακοινώσεις Πρόθεσης (Intent Broadcasts) σε άλλους με την `Context.broadcastIntent()`.

• Δέκτες Μετάδοσης (Broadcast Receivers) - Βασική Κλάση για τον κώδικα, που θα λαμβάνει τις Προθέσεις (Intents) που στέλνονται από το `sendBroadcast ()`. Μπορεί είτε δυναμικά να καταχωρηθεί μια περίπτωση αυτής της κατηγορίας με `Context.registerReceiver()`, είτε να καταχωρηθεί ένα στιγμιότυπο αυτής της κλάσης σε μια εφαρμογή μέσω της ετικέτας `<receiver>` στο `AndroidManifest.xml`. Οι έκτες Μετάδοσης αρχίζουν αυτόματα την εφαρμογή αποκρινόμενοι σε μια εισερχόμενη Πρόθεση, που στην ουσία κάνει την εφαρμογή, μια εφαρμογή οδηγούμενη από τα γεγονότα.

Υπάρχουν δύο σημαντικές κατηγορίες Μεταδόσεων (Broadcasts) που μπορούν να παραληφθούν:

1. Οι Κανονικές Μεταδόσεις - Normal broadcasts (που στέλνονται με `Context.sendBroadcast`) είναι απολύτως ασύγχρονες. Όλοι οι δέκτες της Μετάδοσης οργανώνονται σε μια απροσδιόριστη διαταγή, συχνά συγχρόνως. Αυτό είναι αποδοτικότερο, αλλά σημαίνει ότι οι έκτες δεν μπορούν να χρησιμοποιήσουν το αποτέλεσμα ή να αποβάλουν APIs που συμπεριλαμβάνονται εδώ.

2. Οι Μεταδόσεις διαταγής - Ordered broadcasts (που στέλνονται με `Context.sendOrderedBroadcast`) παραδίδονται σε έναν δέκτη τη φορά. Δεδομένου ότι κάθε δέκτης εκτελείται στη συνέχεια, μπορεί να διαδώσει ένα αποτέλεσμα στον επόμενο δέκτη, ή μπορεί να αποβάλει τη Μετάδοση έτσι ώστε να μη περάσει σε άλλους δέκτες. Ο χειρισμός της εκτέλεσης των δεκτών διαταγής μπορεί να γίνει με την ιδιότητα του Φίλτρου Προθέσεων Ταιριάσματος (Matching Intent-Filter). Οι δέκτες με την ίδια προτεραιότητα τρέχουν με τυχαία σειρά.

• Ειδοποιήσεις (Notifications) - Είναι ένα πλαίσιο ειδοποιήσεων χρήστη. Οι Ειδοποιήσεις δίνουν την δυνατότητα ειδοποίησης του χρήστη χωρίς να υπάρξει παρεμβολή στην τρέχουσα δραστηριότητα (Activity) και του παρέχουν τις εξής δυνατότητες :

1. Να δημιουργήσει ένα νέο εικονίδιο status bar.
2. Να επιδείξει τις πρόσθετες πληροφορίες (και να προωθήσουν μια πρόθεση) στο εκτεταμένο παράθυρο status bar.
3. Να ενεργοποιήσει τα φώτα/LEDs.
4. Να δονηθεί το τηλέφωνο.
5. Να ενεργοποιήσει διάφορους ήχους (ringtones, ήχοι του τηλεφώνου κ.τ.λ).

Οι Ειδοποιήσεις είναι ο προτιμότερος τρόπος για τα αόρατα τμήματα της εφαρμογής δέκτες μετάδοσης, Υπηρεσίες, και ανενεργές δραστηριότητες) για να προειδοποιήσουν τους χρήστες ότι κάποια γεγονότα που έχουν εμφανιστεί απαιτούν την προσοχή του χρήστη. Παραδείγματος χάριν, όταν λαμβάνει μια συσκευή ένα μήνυμα κειμένων ή μια εισερχόμενη κλήση, προειδοποιεί με τη λάμψη των φώτων, την παραγωγή ήχων, την επίδειξη εικονιδίων ή με την εμφάνιση μηνυμάτων διαλόγου.

Ο Χειριστής Ειδοποιήσεων (Notification Manager) είναι μια Υπηρεσία συστήματος που χρησιμοποιείται για να χειριστεί τις ειδοποιήσεις. Καλείται μέσω της μεθόδου `getSystemService`, όπως φαίνεται παρακάτω:

```
String svcName = Context.NOTIFICATION_SERVICE;
```

```
NotificationManager notificationManager;
```

```
NotificationManager=(NotificationManager)getSystemService(  
svcName);
```

Χρησιμοποιώντας το Χειριστή Ειδοποιήσεων, μπορούν να ενεργοποιηθούν νέες ειδοποιήσεις, να τροποποιηθούν οι υπάρχουσες ή να αφαιρεθούν εκείνες που δεν είναι απαραίτητες ή επιθυμητές

4.10 Γιατί Android

Εν κατακλείδι, παρατίθενται εδώ κάποια σημεία και σκέψεις για το Android, σαν μια τελευταία απόδειξη της πρωτοπορίας του στο χώρο και της ισχυρής δυναμικής του. Γιατί Android λοιπόν;

- Είναι μια πραγματικά ανοιχτή, ελεύθερη πλατφόρμα ανάπτυξης, βασισμένη στο Linux.
- Διαθέτει αρχιτεκτονική βασισμένη σε δομικά στοιχεία τα οποία μπορούν να τροποποιηθούν, να ολοκληρωθούν και να προσαρμοστούν στις ανάγκες κάθε κατασκευαστή και κατά συνέπεια χρήστη.
- Πάμπολλες ενσωματωμένες υπηρεσίες που μπορούν να κάνουν την εμπειρία του χρήστη μοναδική, όπως υπηρεσίες βασισμένες στην τοποθεσία, πανίσχυρη SQL βάση δεδομένων, μηχανή αναζήτησης και χάρτες.
- Αυτόματη διαχείριση του κύκλου ζωής μιας εφαρμογής, με πολλαπλές δικλείδες ασφαλείας ανάμεσα στα προγράμματα. Βελτιστοποιήσεις στον τομέα διαχείρισης μνήμης και χαμηλής κατανάλωσης σε τέτοιο βαθμό που δεν έχει ξανασυναντηθεί σε άλλο smartphone.
- Υψηλής ποιότητας γραφικά και ήχος.
- Φορητότητα ανάμεσα σε ένα ευρύ φάσμα ήδη υπάρχοντος υλικού αλλά και μελλοντικού. Αυτό έρχεται σαν απόρροια του γεγονότος ότι όλα τα προγράμματα γράφονται σε Java και εκτελούνται από την εικονική μηχανή Dalvik. Επιπρόσθετα οι οθόνες μπορούν να τροποποιηθούν κατάλληλα για να υποστηρίξουν οποιαδήποτε ανάλυση, μέγεθος και προσανατολισμό οθόνης. Αυτοί είναι μερικοί μόνο από τους λόγους για τους οποίους έγινε η επιλογή του Android ως πλατφόρμα ανάπτυξης σε αυτή την πτυχιακή εργασία. Ο οποιοσδήποτε χρήστης θα μπορούσε να επαληθεύσει όλα τα παραπάνω, χρησιμοποιώντας απλά ένα τηλέφωνο ή μια ταμπλέτα που τρέχει το λειτουργικό αυτό σύστημα.

5.Υλοποίηση εφαρμογής

Η εφαρμογή για αναζήτηση καταστημάτων σύμφωνα με τα κριτήρια που ορίζει ο χρήστης υλοποιήθηκε όπως αναφέρθηκε και παραπάνω με χρήση της τεχνολογίας Android σε προγραμματιστικό περιβάλλον Eclipse.Πιο συγκεκριμένα χρησιμοποιήθηκε Eclipse Indigo και η πλατφόρμα Android 2.3.3.

Έχουμε υποθέσει ότι η αναζήτηση γίνεται για καταστήματα που βρίσκονται στην περιοχή της Αθήνας.Τα καταστήματα αυτά έχουν καταχωρηθεί σε μια βάση δεδομένων (MySQL) και βρίσκονται τοπικά στον υπολογιστή μας.

Ο χρήστης δημιουργεί το προφίλ του όπου στην ουσία ορίζει τα κριτήρια της αναζήτησης του.Συγκεκριμένα επιλέγει το είδος του καταστήματος προς αναζήτηση ,στη συγκεκριμένη εφαρμογή επιλέγει ανάμεσα σε δύο ειδών καταστήματα (καταστήματα κινητής τηλεφωνίας και καταστήματα ρουχισμού) επίσης ορίζει το επιθυμητό ranking των καταστημάτων προς αναζήτηση.Τέλος κάνοντας click στο κουμπί show της εφαρμογής έχει την δυνατότητα να δει τα αποτελέσματα της αναζήτησης του στο χάρτη.

Τα ονόματα, οι διευθύνσεις, το είδος , οι γεωγραφικές συντεταγμένες καθώς και το ranking κάθε καταστήματος είναι αποθηκευμένα σε μια βάση δεδομένων (MySQL) τοπικά.Η εφαρμογή έχει πρόσβαση στα δεδομένα αυτά μέσω μίας διαδικασίας , η οποία παίρνει τα δεδομένα αυτά σαν αρχείο xml.Στη συνέχεια μέσω μιας διαδικασίας που ονομάζεται parsing κρατά σε μία λίστα shopNames τα στοιχεία των καταστημάτων που πληρούν τα κριτήρια που έχει ορίσει ο χρήστης στο προφίλ του.Τέλος η εφαρμογή παρουσιάζει τα αποτελέσματα – καταστήματα στο χάρτη σύμφωνα με τις γεωγραφικές συντεταγμένες τους.Για την επίτευξη αυτής της διαδικασίας έχει χρησιμοποιηθεί κώδικας PHP σε προγραμματιστικό περιβάλλον NetBeans IDE 7.2.

5.1 Κλάσεις Java

MainActivity.java: Αυτή η κλάση παρουσιάζει την αρχική οθόνη της εφαρμογής όπου ο χρήστης επιλέγει να δημιουργήσει το προφίλ του. Δηλαδή περιέχει την `main_activity` η οποία επιτρέπει στο χρήστη κάνοντας click στο κουμπί `create your profil` να μεταβεί στην φόρμα για την δημιουργία του προφίλ του.

Activity2.java: Όπως έχουμε ήδη αναφέρει τα στοιχεία των καταστημάτων προς αναζήτηση είναι καταχωρημένα σε μία βάση δεδομένων (`MySQL`) τοπικά. Για να μπορέσει να έχει η εφαρμογή πρόσβαση στα δεδομένα αυτά έχει εφαρμοστεί μια διαδικασία όπου τα δεδομένα αυτά έχουν μετατραπεί σε μορφή `xml`. Μέσω της κλάσης `Activity2` δημιουργείται κάθε φορά το `Uri` ανάλογα με τις επιλογές των χρηστών ώστε η εφαρμογή να έχει πρόσβαση στα αντίστοιχα αποτελέσματα. Η κλάση αυτή περιέχει μια πολύ σημαντική συνάρτηση , την `onClick`. Μέσω αυτής της συνάρτησης καλείται η κλάση `ShopParser` . Τέλος περιέχει την `activity_2` η οποία επιτρέπει στο χρήστη κάνοντας click στο κουμπί `show` να μεταβεί στο χάρτη της εφαρμογής όπου μπορεί να δει τα αποτελέσματα - καταστήματα της αναζήτησης του.

ShopParser.java: Η κλάση αυτή μέσω μιας διαδικασίας που ονομάζεται `parsing` από το αρχείο `xml` όπου βρίσκονται τα αποτελέσματα της αναζήτησης κρατά σε μία λίστα `shopNames` το όνομα, το `ranking` και τις γεωγραφικές συντεταγμένες κάθε μαγαζιού ώστε αργότερα να τα εμφανίσει στο χάρτη.

Activity3.java: Αυτή η κλάση παρουσιάζει τον χάρτη με τα καταστήματα πάνω σε αυτόν. Για την χρήση του χάρτη στην εφαρμογή, χρειάστηκε ένα κλειδί το οποίο είναι μοναδικό και παρέχετε από την `Google` δίνοντας μας την δυνατότητα να χρησιμοποιήσουμε μια πληθώρα υπηρεσιών. Λέγοντας ότι το κλειδί είναι μοναδικό εννοούμε ότι διαφέρει από υπολογιστή σε υπολογιστή και προέρχεται από το `MD5` δαχτυλικό αποτύπωμα .Εκτός από το κλειδί για την χρήση του χάρτη χρειαζόμαστε και μια άδεια πρόσβασης στο `internet` για την εφαρμογή.

OverlayItems.java: Σε αυτήν την κλάση δημιουργούμε τα καταστήματα – αποτελέσματα στο χάρτη. Έχουμε διαλέξει μία εικόνα που θα αντιπροσωπεύει τα καταστήματα – αποτελέσματα στο χάρτη. Όταν ο χρήστης κάνει click πάνω σε αυτήν την εικόνα στο χάρτη τότε εκτελείται η μέθοδος onTap() . Το αποτέλεσμα της μεθόδου αυτής είναι η εμφάνιση ενός κουτιού διαλόγου(AlertDialog box) με ένα μήνυμα. Το μήνυμα ενημερώνει τον χρήστη για το όνομα του καταστήματος που έχει επιλέξει.

Shop.java: Η κλάση αυτή έχει δημιουργηθεί για να μπορέσουμε να χειριστούμε τα στοιχεία της λίστας shopNames σαν objects. Περιλαμβάνει μόνο τους βασικούς getters και setters ώστε να μπορέσουμε να χρησιμοποιήσουμε το όνομα και τα γεωγραφικά στοιχεία κάθε καταστήματος στην κλάση Activity3.

XML Arxeia

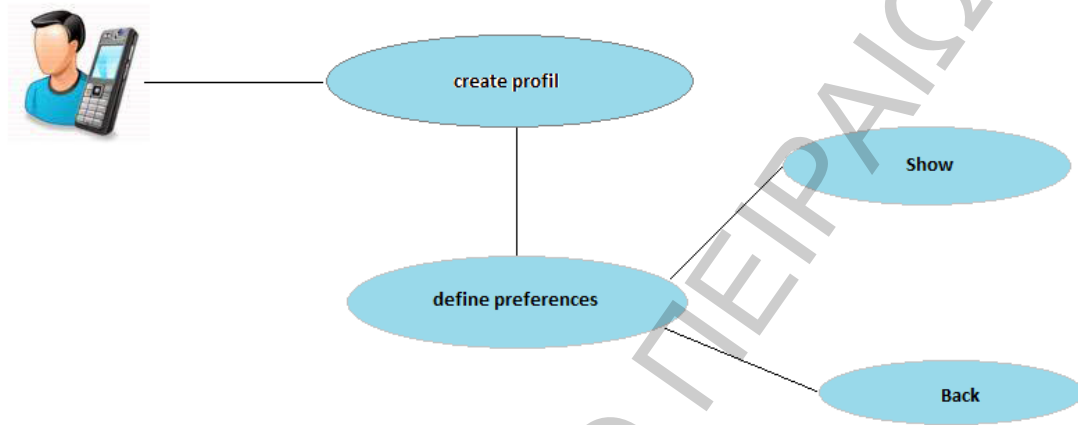
Αυτά τα αρχεία περιέχουν τα layouts για κάθε μια από τις παραπάνω κλάσεις. Έχουμε δημιουργήσει τρία διαφορετικά layouts activity_main.xml , activity_2.xml και activity_3.xml για κάθε μία από τις βασικές κλάσεις.

Manifest

Το αρχείο αυτό είναι ένα πολύ σημαντικό και απαραίτητο κομμάτι για μια εφαρμογή Android, μας βοηθάει να διαχειριστούμε καλύτερα την εφαρμογή μας, ρυθμίζοντας τα δικαιώματα που θέλουμε να έχει, την έκδοση (version) της εφαρμογής μας όπως και άλλες σημαντικές επιλογές. Σε αυτό το αρχείο δηλώνονται όλες οι δραστηριότητες της εφαρμογής ώστε να λειτουργούν σωστά. Επίσης η εφαρμογή από αυτό το αρχείο παίρνει την άδεια για πρόσβαση στο internet ώστε να μπορεί να κάνει χρήση των Google maps.

5.2 Use Cases Diagram

Το παρακάτω διάγραμμα παρουσιάζει όλα τα πιθανά σενάρια του συστήματος και όλους τους πιθανούς χρήστες που εμπλέκονται σε αυτό.



Εικόνα 8: Use Cases Diagram

5.2.1 Use Case : Create profil

Το παρακάτω διάγραμμα αναφέρεται στην δημιουργία προφίλ από τον χρήστη.

Πίνακας 1:Use case create profil

Use case	Create profil
Χρήστες	Χρήστης κινητού τηλεφώνου
Ενέργεια	Ο χρήστης δημιουργεί το προφίλ με τις προτιμήσεις του με στόχο να δει τα αποτελέσματα της αναζήτησης του στο χάρτη
Παρεμβάλλουσες διαδικασίες	Ο χρήστης οδηγείτε στην φόρμα συμπλήρωσης προφίλ

Βασική σειρά εξέλιξης γεγονότων	
Χρήστης	Ανταπόκριση συστήματος
1) Ο χρήστης κάνει click στο κουμπί create your profil 2) Ο χρήστης συμπληρώνει τις προτιμήσεις του και κάνει click στο κουμπί show	3) Η εφαρμογή μεταβαίνει στη φόρμα δημιουργίας προφίλ 4) Η εφαρμογή παρουσιάζει τα αποτελέσματα στο χάρτη
Εναλλακτική διαδρομή	Όχι

5.2.2 Use case :Define preference

Το ακόλουθο διάγραμμα use case αναφέρεται στον καθορισμό προτιμήσεων του χρήστη. Ο χρήστης μπορεί να επιλέξει ανάμεσα σε δύο ειδών καταστήματα και ανάμεσα σε πέντε αστέρια για την βαθμολόγηση τους.

Πίνακας 2:Use case define preference

Use case	Define preference
Χρήστες	Χρήστης κινητού τηλεφώνου
Ενέργεια	Ο χρήστης διαλέγει είδος καταστήματος και αστέρια
Παρεμβάλλουσες διαδικασίες	Καθορισμός προτιμήσεων
Βασική σειρά εξέλιξης γεγονότων	
Χρήστης	Ανταπόκριση συστήματος
1)Ο χρήστης κάνει click στα αντίστοιχα κουμπιά ανάλογα με την προτίμηση του	2) Το σύστημα «κρατά » τις προτιμήσεις του χρήστη ώστε αργότερα να εκτελέσει την αναζήτηση

	με βάση αυτές
Εναλλακτική διαδρομή	Όχι

5.2.3 Use case Show

Πίνακας 3: Use case Show

Use case	Display shop
Χρήστες	Χρήστης κινητού τηλεφώνου
Ενέργεια	Ο χρήστης κάνει click στο κουμπί show ώστε να δει τα αποτελέσματα της αναζήτησης του στο χάρτη.
Παρεμβάλλουσες ενέργειες	Ο χρήστης θα πρέπει πριν να έχει δημιουργήσει το προφίλ του και να έχει καθορίσει τις προτιμήσεις του
Βασική σειρά εξέλιξης γεγονότων	
Χρήστης	Ανταπόκριση συστήματος
1) Ο χρήστης κάνει click στο κουμπί show	2) Παρουσιάζονται τα αποτελέσματα στο χάρτη
Εναλλακτική διαδρομή	Ο χρήστης μπορεί να πάει πίσω και να αλλάξει τις προτιμήσεις του

5.2.4 Use case Back

Πίνακας 4: Use case Back

Use case	Back
Χρήστες	Χρήστης κινητού τηλεφώνου
Ενέργεια	Ο χρήστης κάνει click στο κουμπί back ώστε να αλλάξει τις προτιμήσεις του
Παρεμβάλουσες ενέργειες	Ο χρήστης έχει ήδη δημιουργήσει το προφίλ του και έχει πραγματοποιήσει μία αναζήτηση τουλάχιστον με βάση αυτές.
Βασική σειρά εξέλιξης γεγονότων	
Χρήστης	Ανταπόκριση συστήματος
1) Ο χρήστης κάνει click στο κουμπί back	2) Η εφαρμογή επιστρέφει στη φόρμα δημιουργίας προφίλ
Εναλλακτική διαδρομή	Όχι

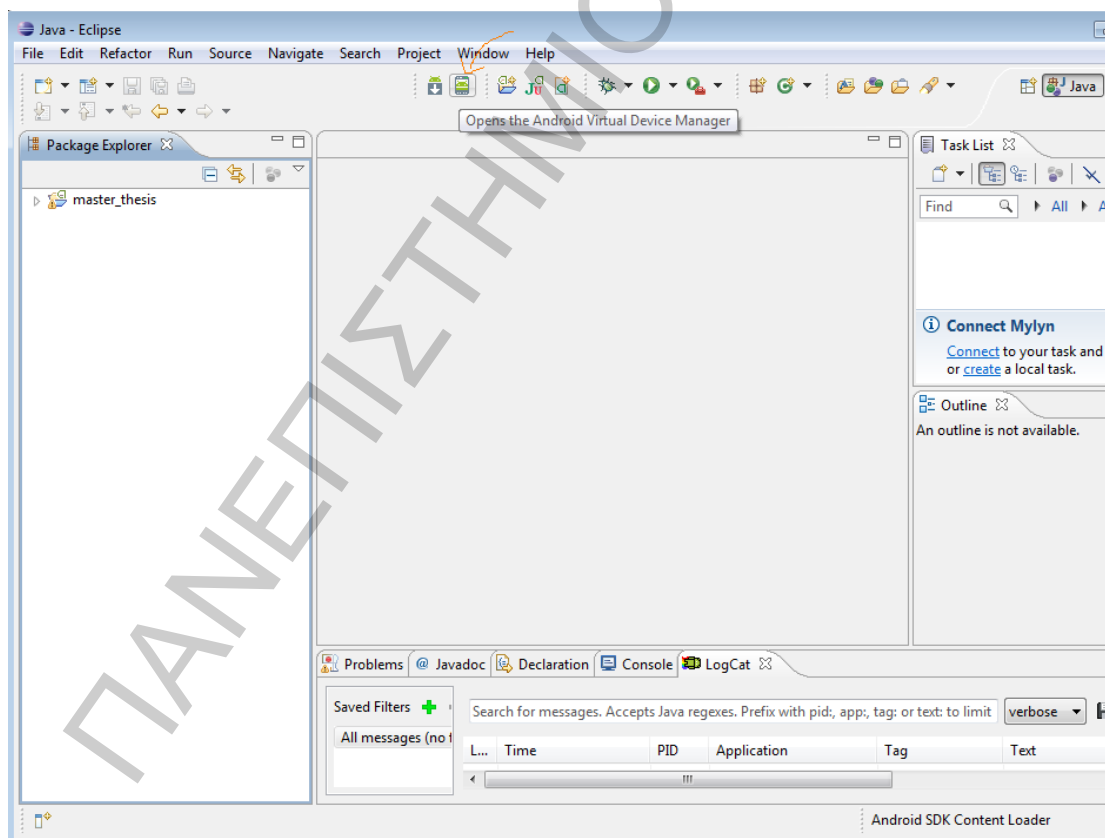
6 Παρουσίαση εφαρμογής

6.1 Εγκατάσταση Πλατφόρμας

Ο πηγαίος κώδικας βρίσκεται μέσα στο προγραμματιστικό περιβάλλον Eclipse. Για την εκτέλεση της εφαρμογής υπάρχει συγκεκριμένη επιλογή (Run Option).

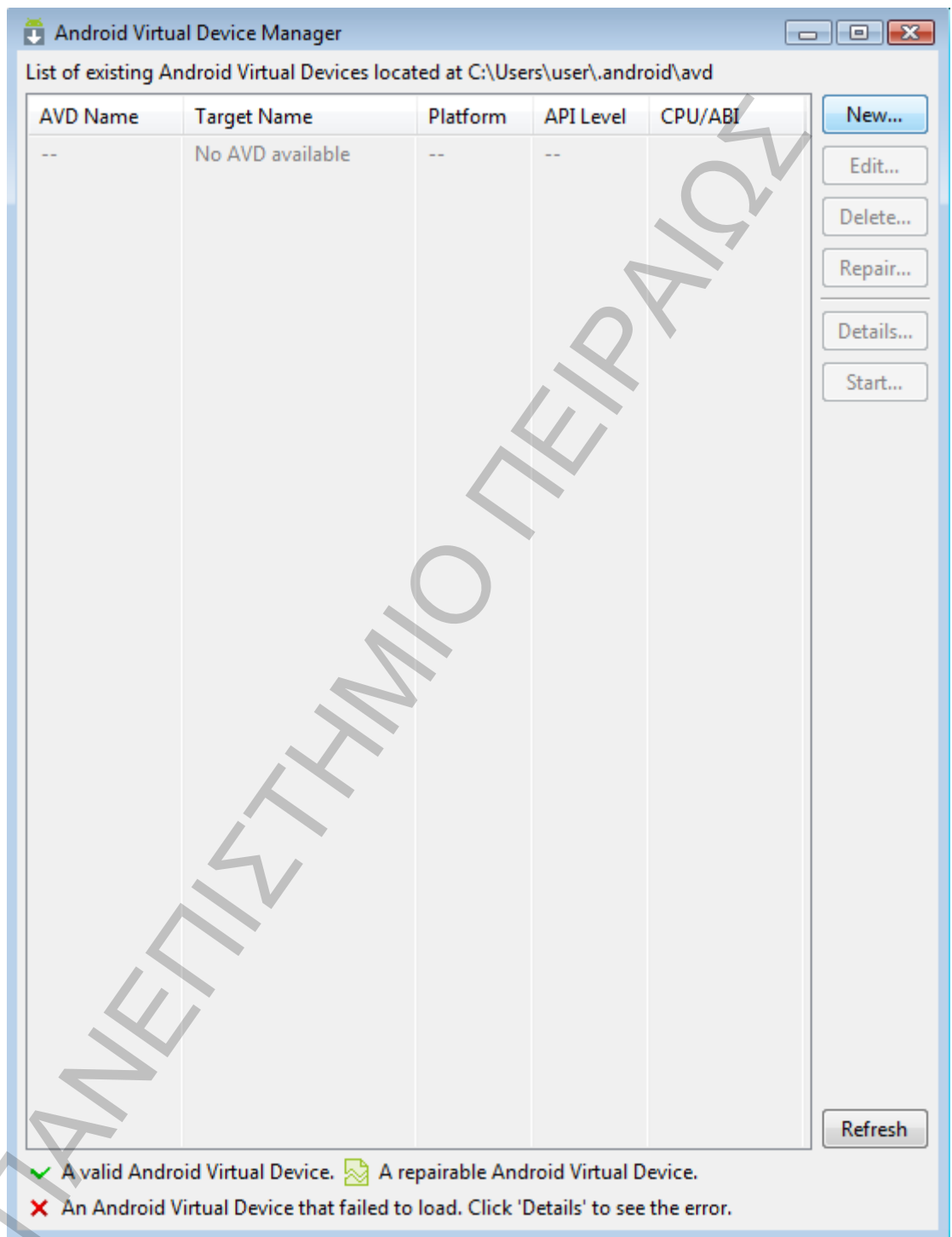
Μερικές φορές δεν υπάρχει διαθέσιμος emulator για αυτό είναι απαραίτητο να δημιουργήσουμε έναν νέο ακολουθώντας την παρακάτω διαδικασία.

1. Κάνουμε click στο εικονίδιο που δείχνει το κόκκινο βέλος της εικόνας.



Εικόνα 9 Προγραμματιστικό περιβάλλον Eclipse

Και εμφανίζεται το επόμενο παράθυρο διαλόγου.



Εικόνα 10 AVD Manager

2.Κάνοντας click στην επιλογή new

Create new Android Virtual Device (AVD)

Name:

Target:

CPU/ABI:

SD Card:

Size:

File:

Snapshot:

Enabled

Skin:

Built-in:

Resolution: x

Hardware:

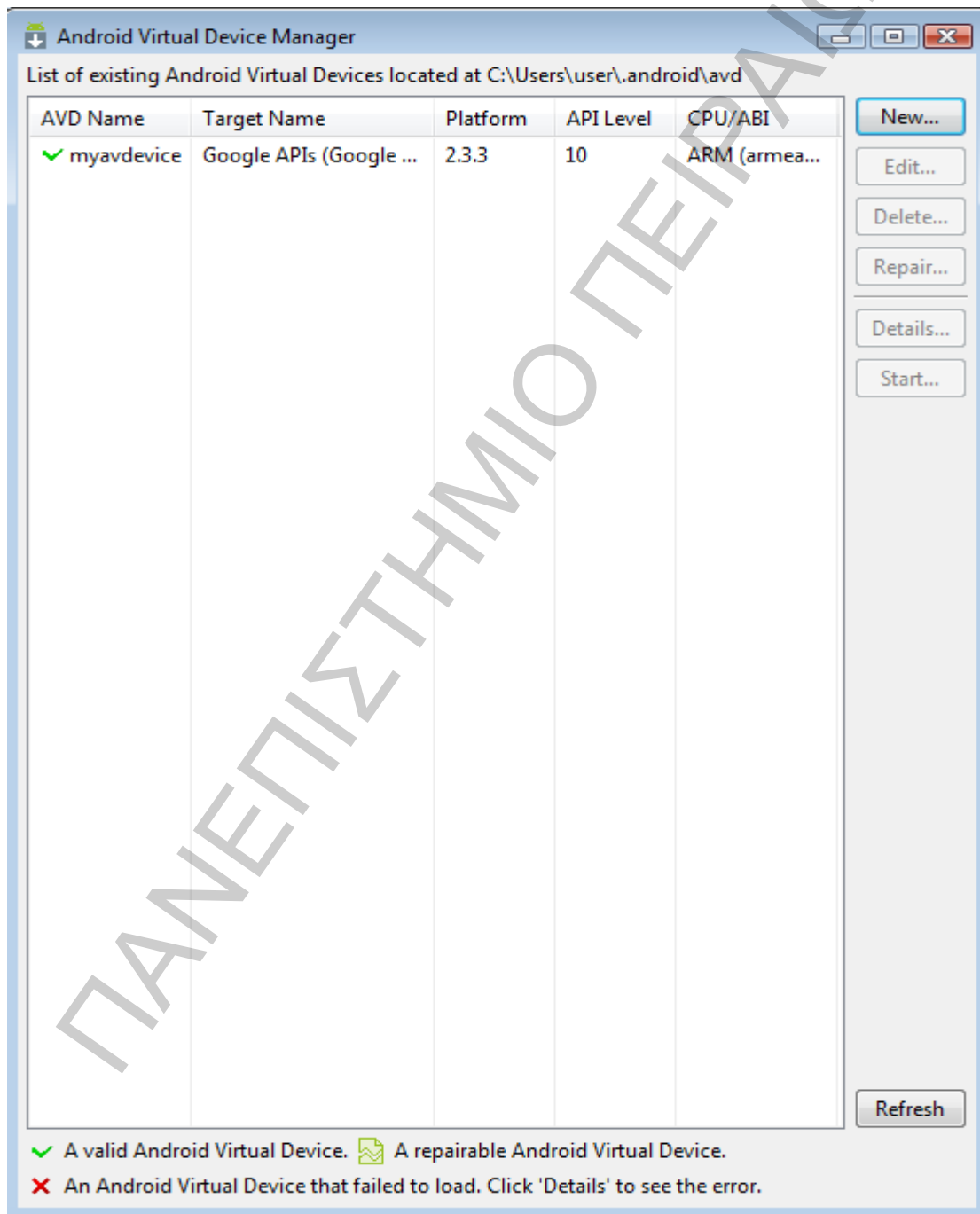
Property	Value
Max VM application hea...	24
Device ram size	256
Abstracted LCD density	240

Override the existing AVD with the same name

Εικόνα 11 AVD Manager fields

3. Συμπληρώνουμε τα πεδία Name και Target. Το API Level του πεδίου target είναι πολύ σημαντικό γιατί η εφαρμογή γιατί δεν θα είναι ικανή να τρέξει σε ένα σύστημα όπου το API Level είναι μικρότερο από αυτό που απαιτείται από την εφαρμογή έτσι όπως αυτό ορίζεται στο minSdk Version attribute στο Manifest αρχείο της εφαρμογής.

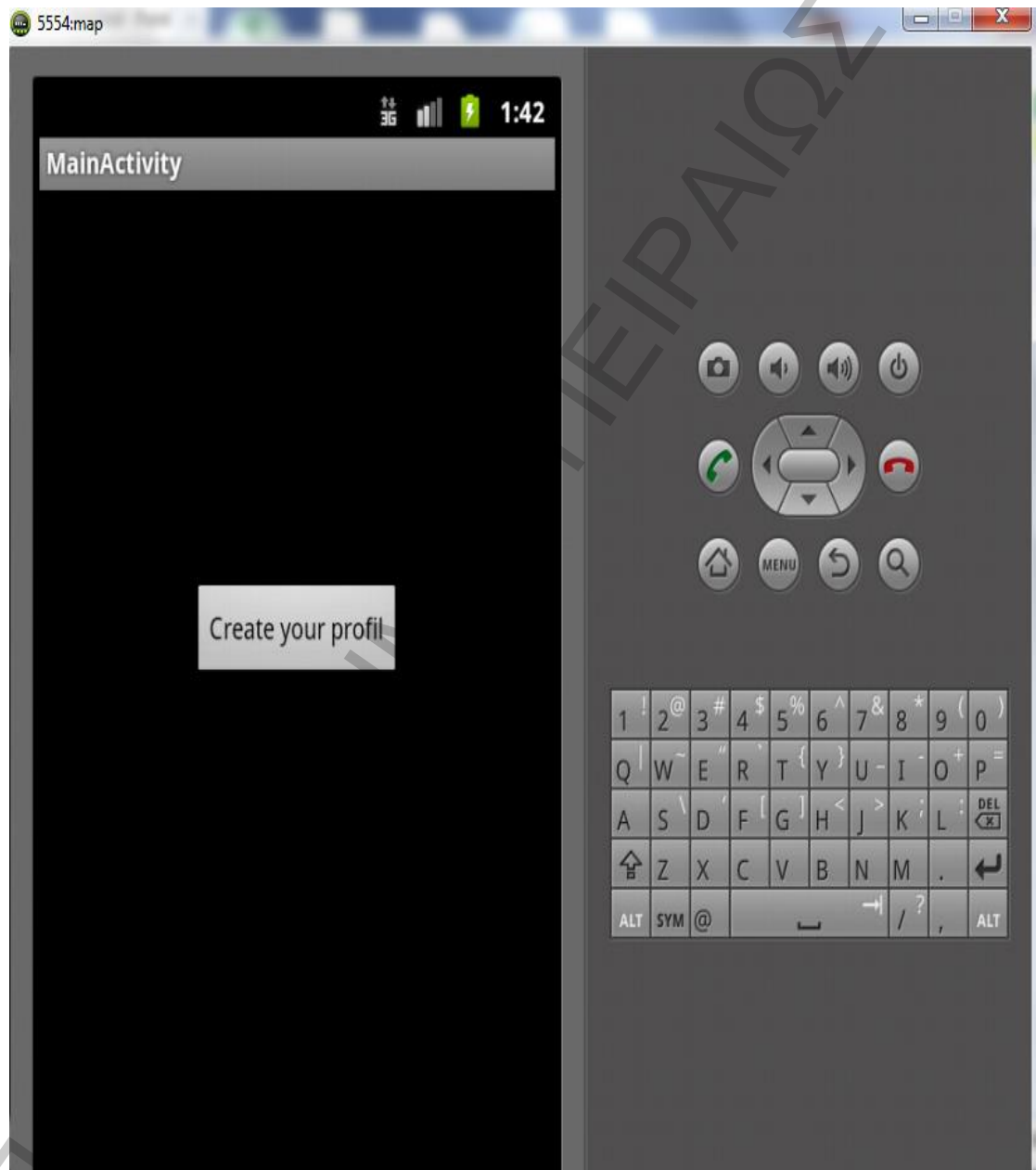
4. Το αποτέλεσμα είναι ένα νέο AVD



Εικόνα 12 New Device

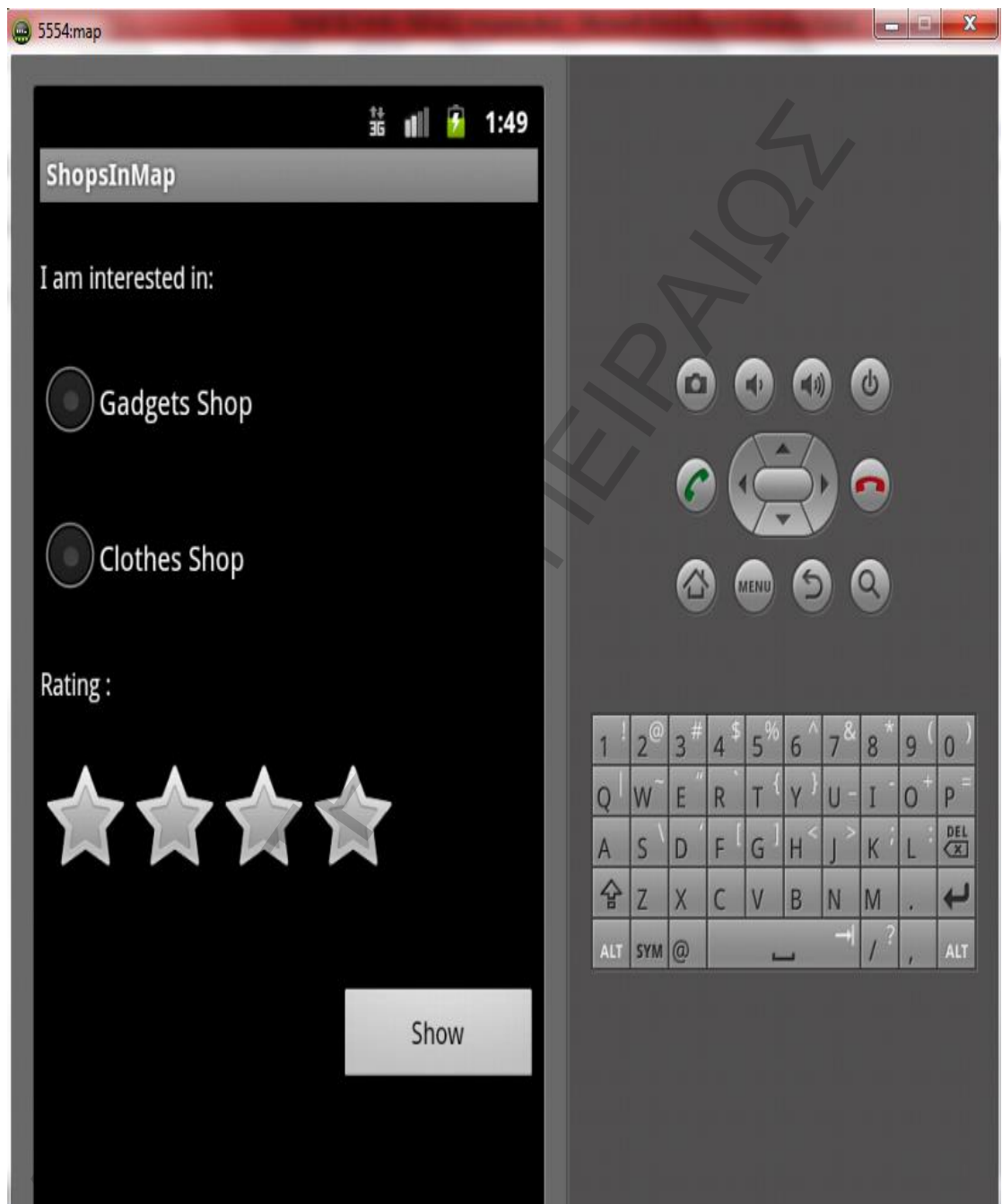
Τώρα είμαστε έτοιμοι να τρέξουμε την εφαρμογή ακολουθώντας τα επόμενα βήματα

1. Αρχικά ο χρήστης βλέπει την `activity_main` της κλάσης `MainActivity.java`



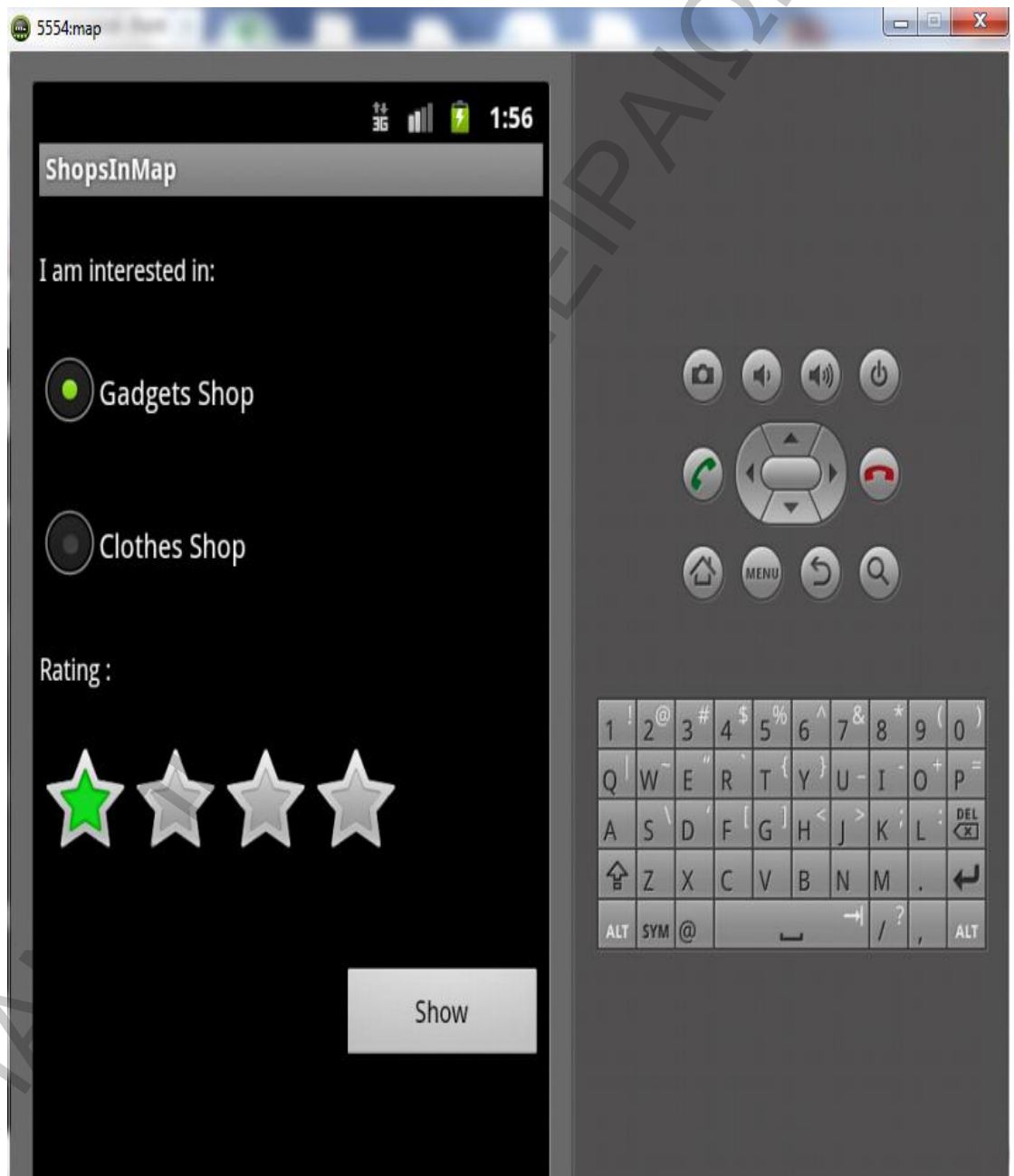
Εικόνα 13 MainActivity.java Activity

2. Στη συνέχεια ο χρήστης πατά το κουμπί Create your profil.



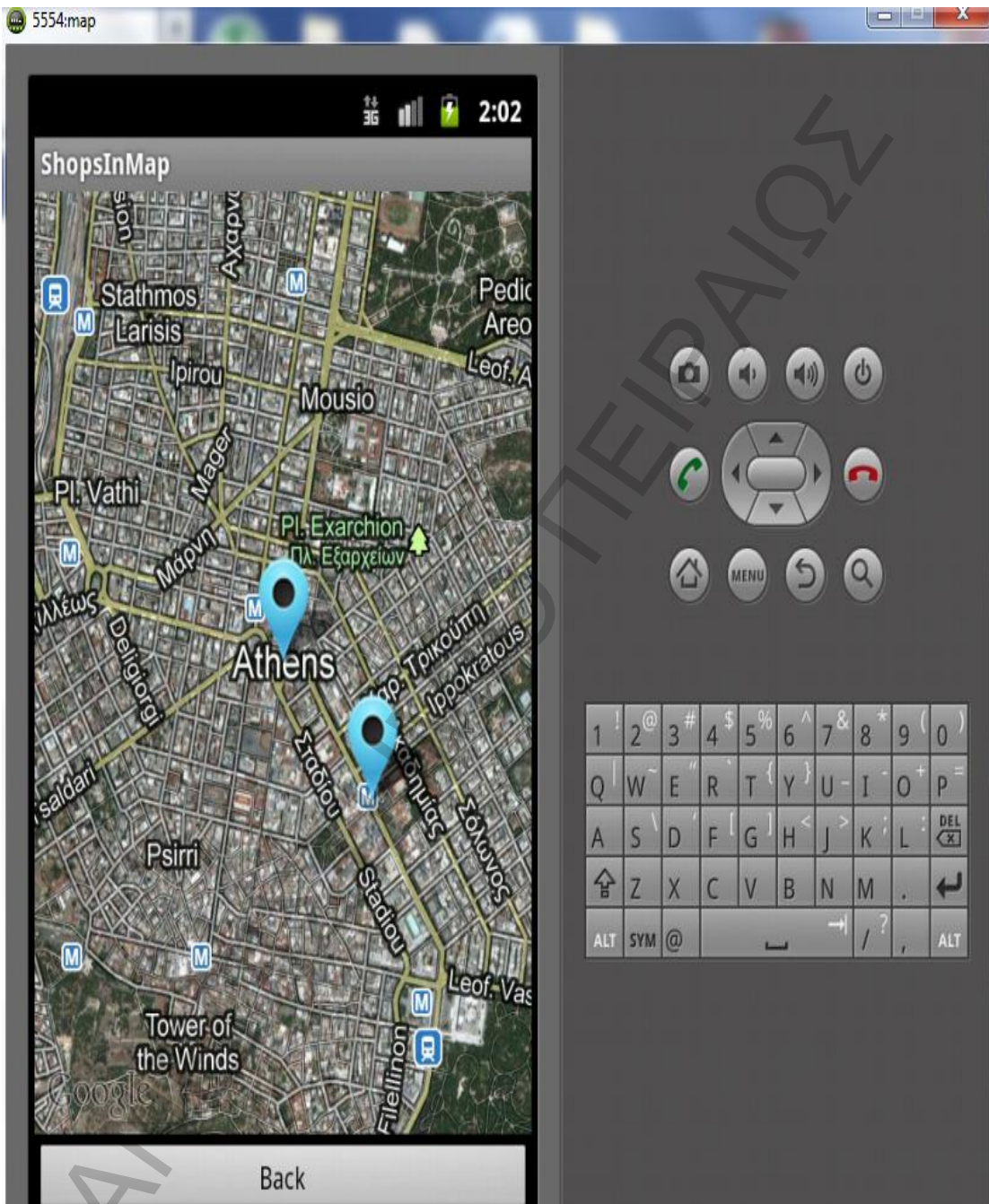
Εικόνα 14 Activity2.java Activity

3. Σε αυτό το βήμα ο χρήστης επιλέγει τα κριτήρια αναζήτησης του.
- Έστω ότι ο χρήστης ενδιαφέρεται για Gadgets Shop που είναι βαθμολογημένο με ένα αστέρι.



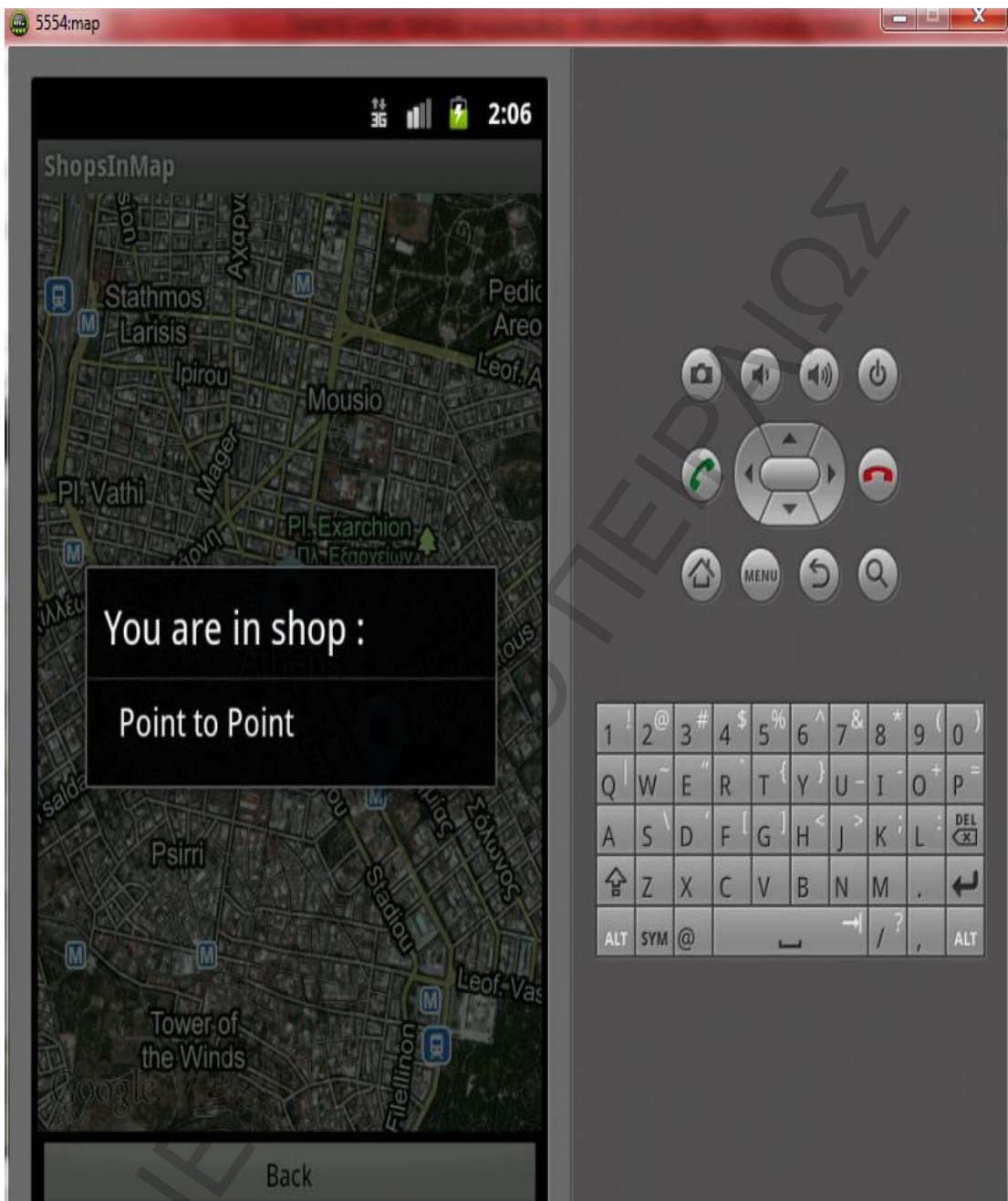
Εικόνα 15 Activity2.java Activity

4. Έπειτα ο χρήστης επιλέγει το κουμπί show



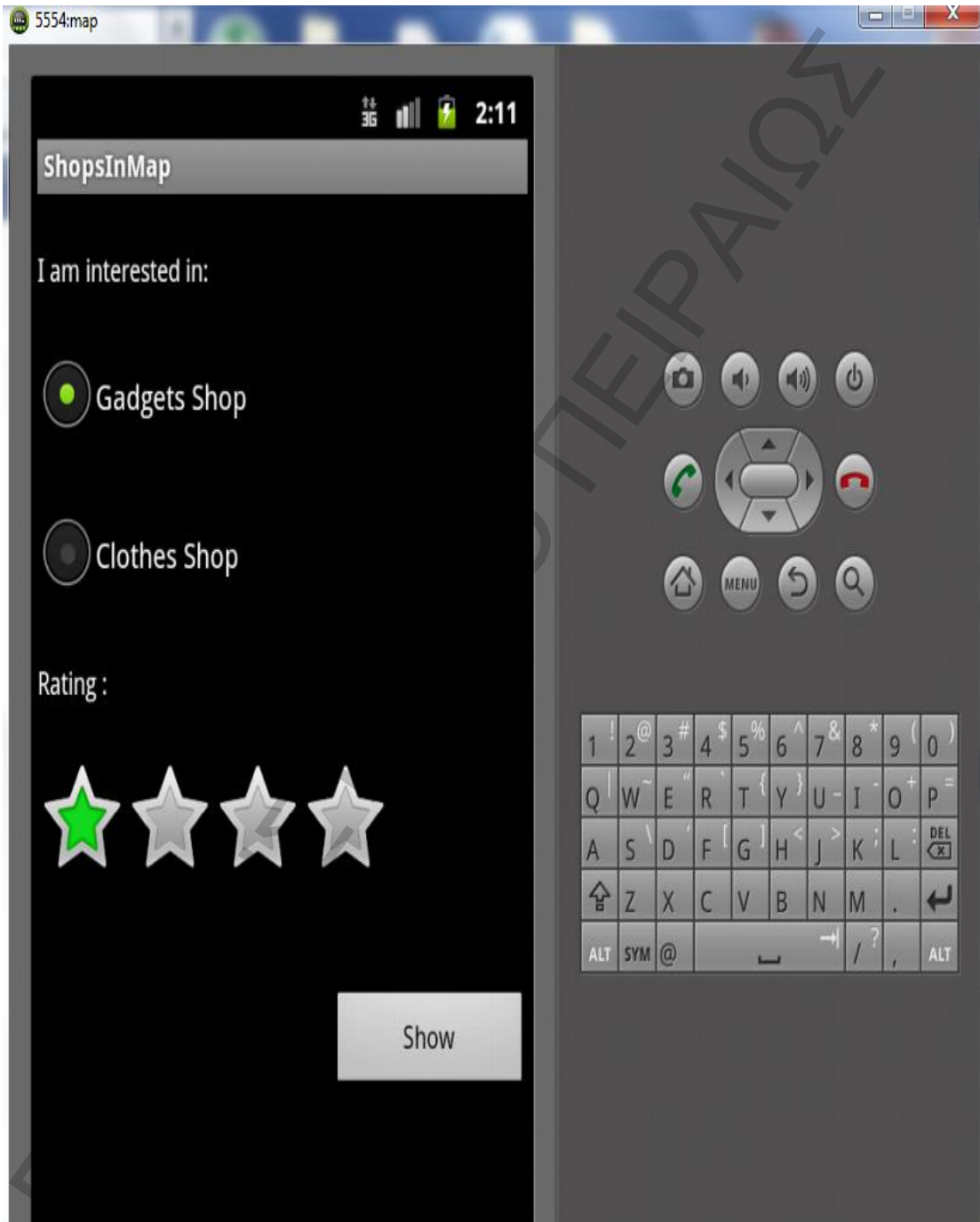
Εικόνα 16 Activity3.java Activity

5. Επιλέγοντας ένα OverlayItem :



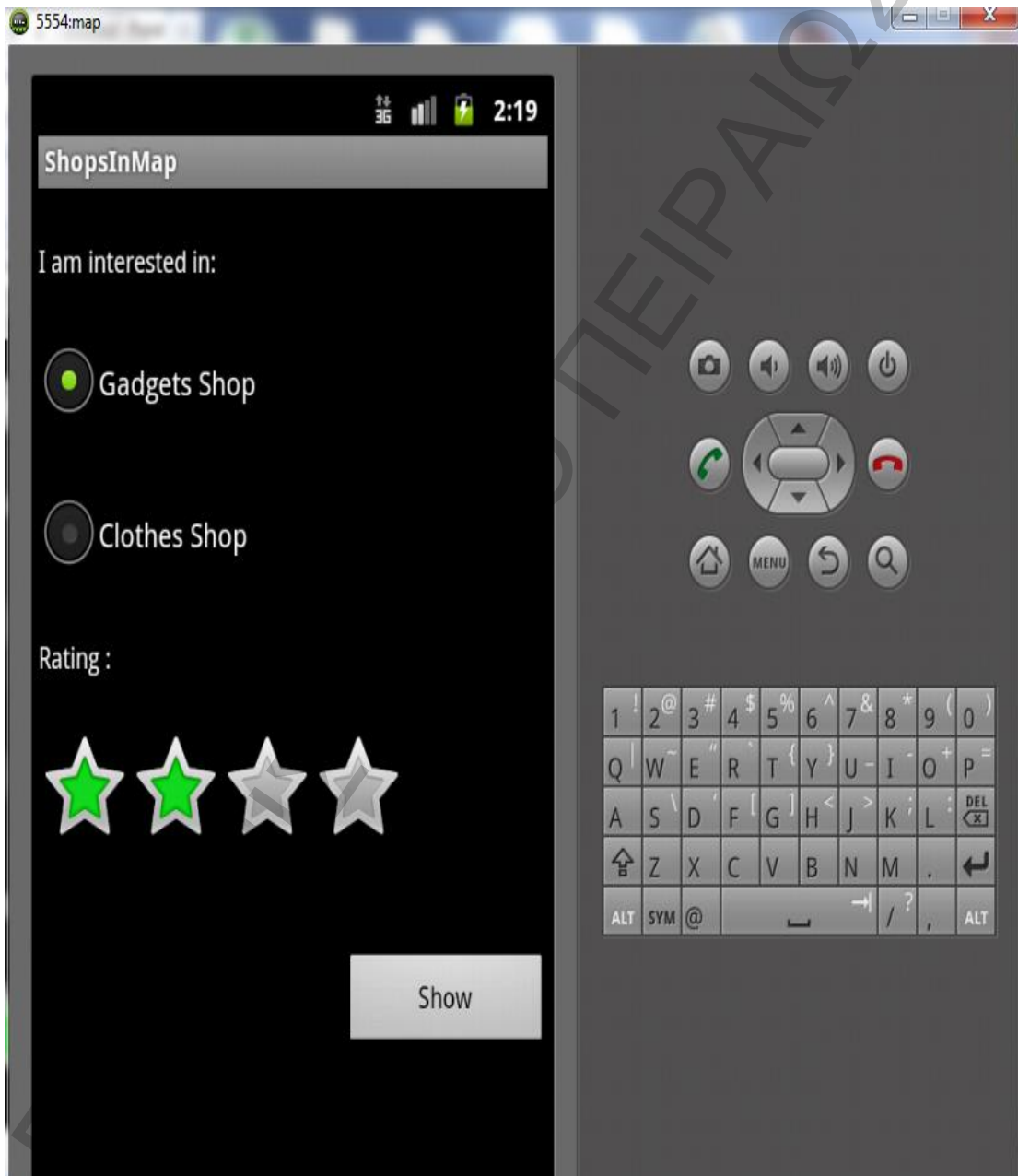
Εικόνα 17 OverlayItems.java Activity

6. Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



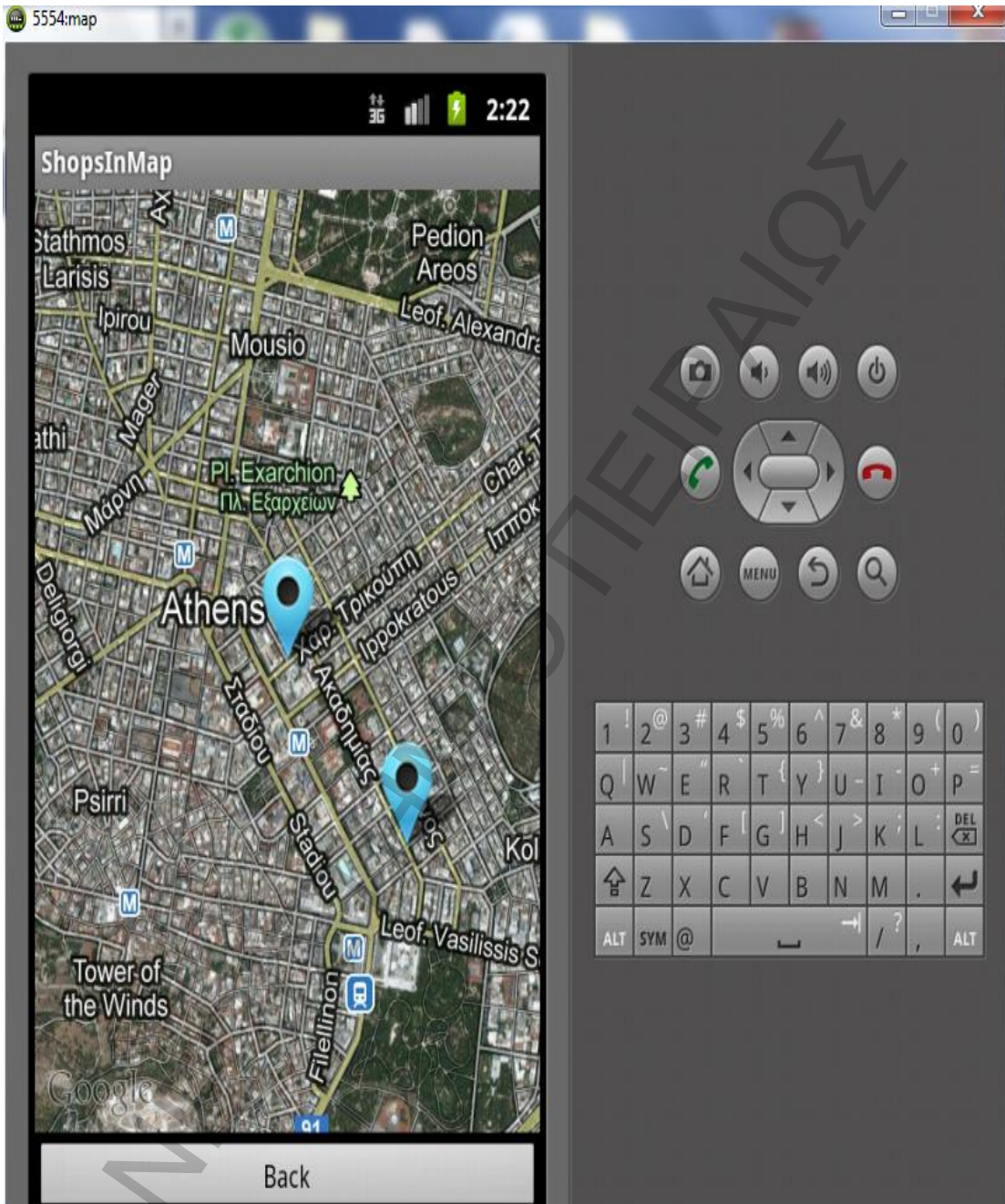
Εικόνα 18 Activity2.java Activity

7. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.
- Έστω ότι ο χρήστης επιλέγει πάλι Gadgets Shop αλλά αυτή την φορά δύο αστέρια ως ranking.



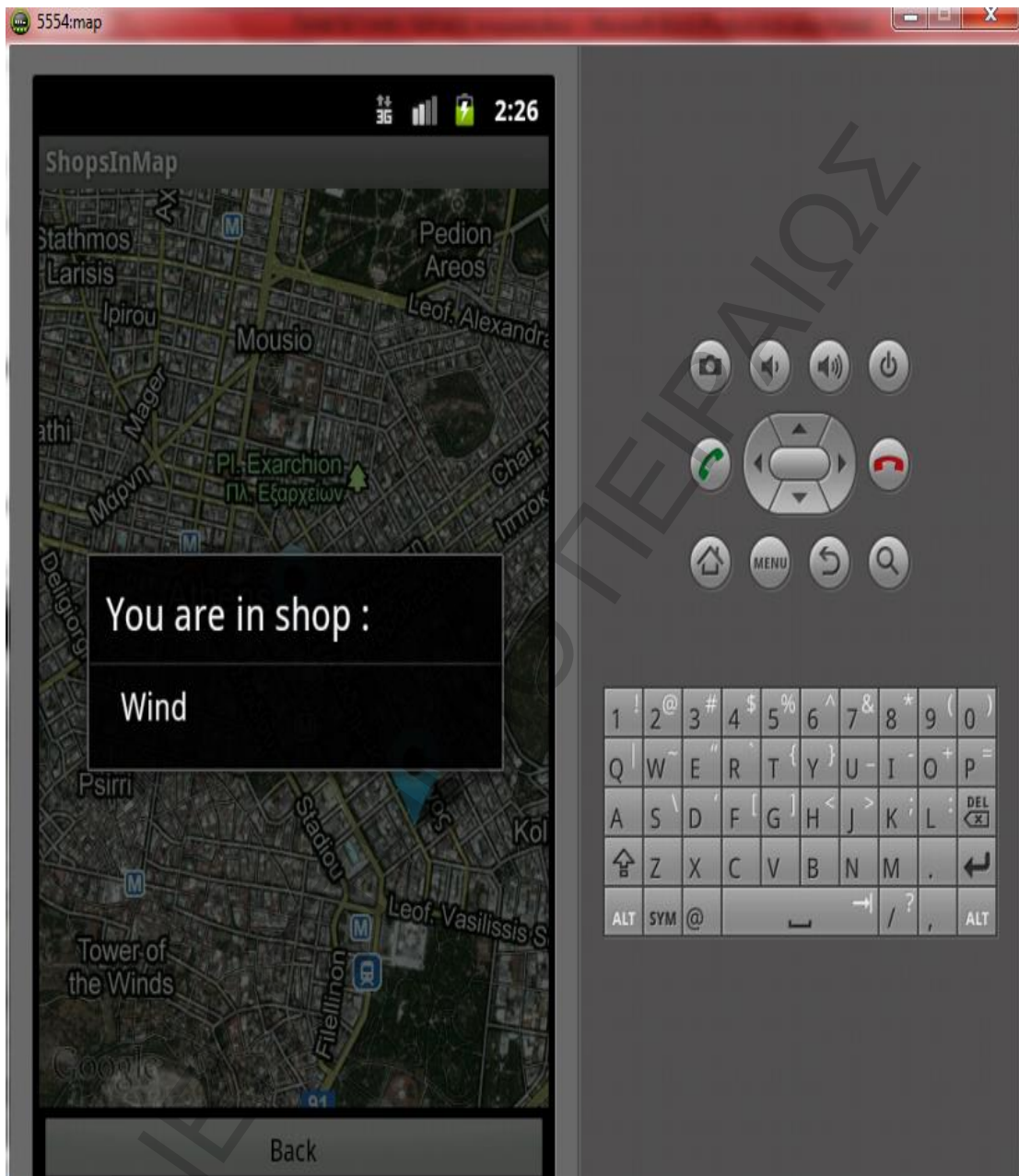
Εικόνα 19 Activity2.java Activity

8. Έπειτα ο χρήστης επιλέγει το κουμπί show



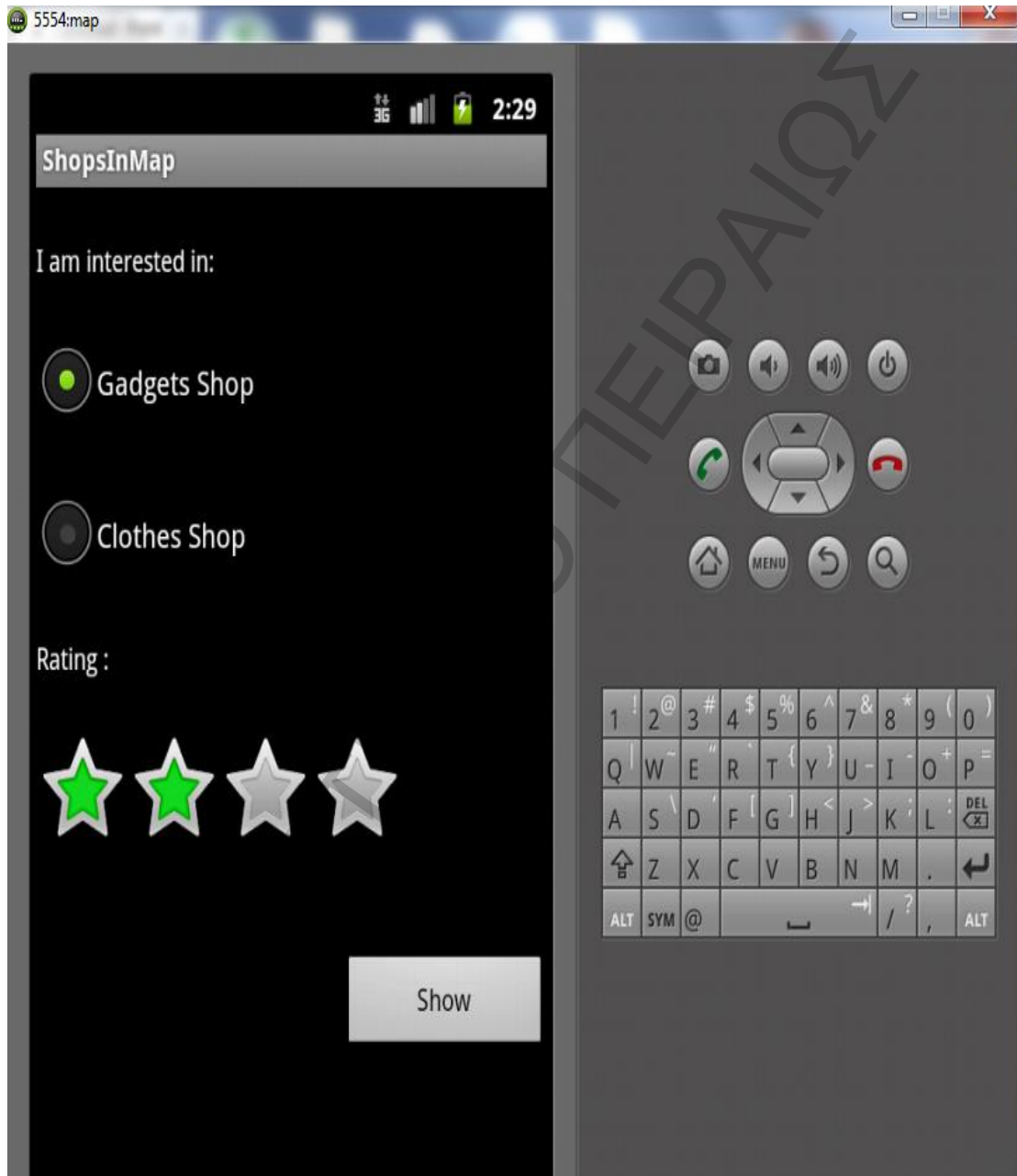
Εικόνα 20 Activity3.java Activity

9. Επιλέγοντας ένα OverlayItem :



Εικόνα 21 OverlayItems.java Activity

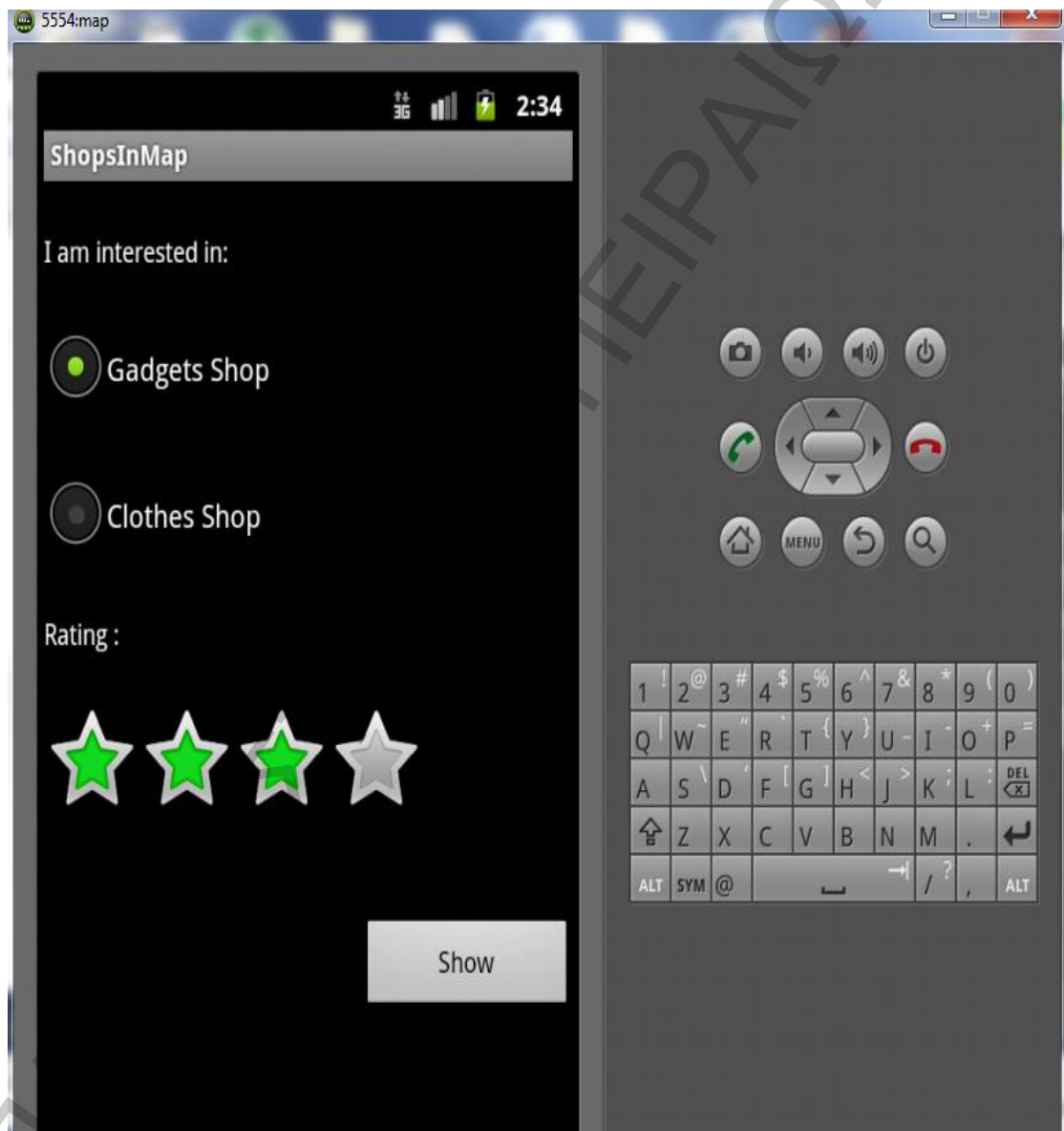
10.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 22 Activity2.java Activity

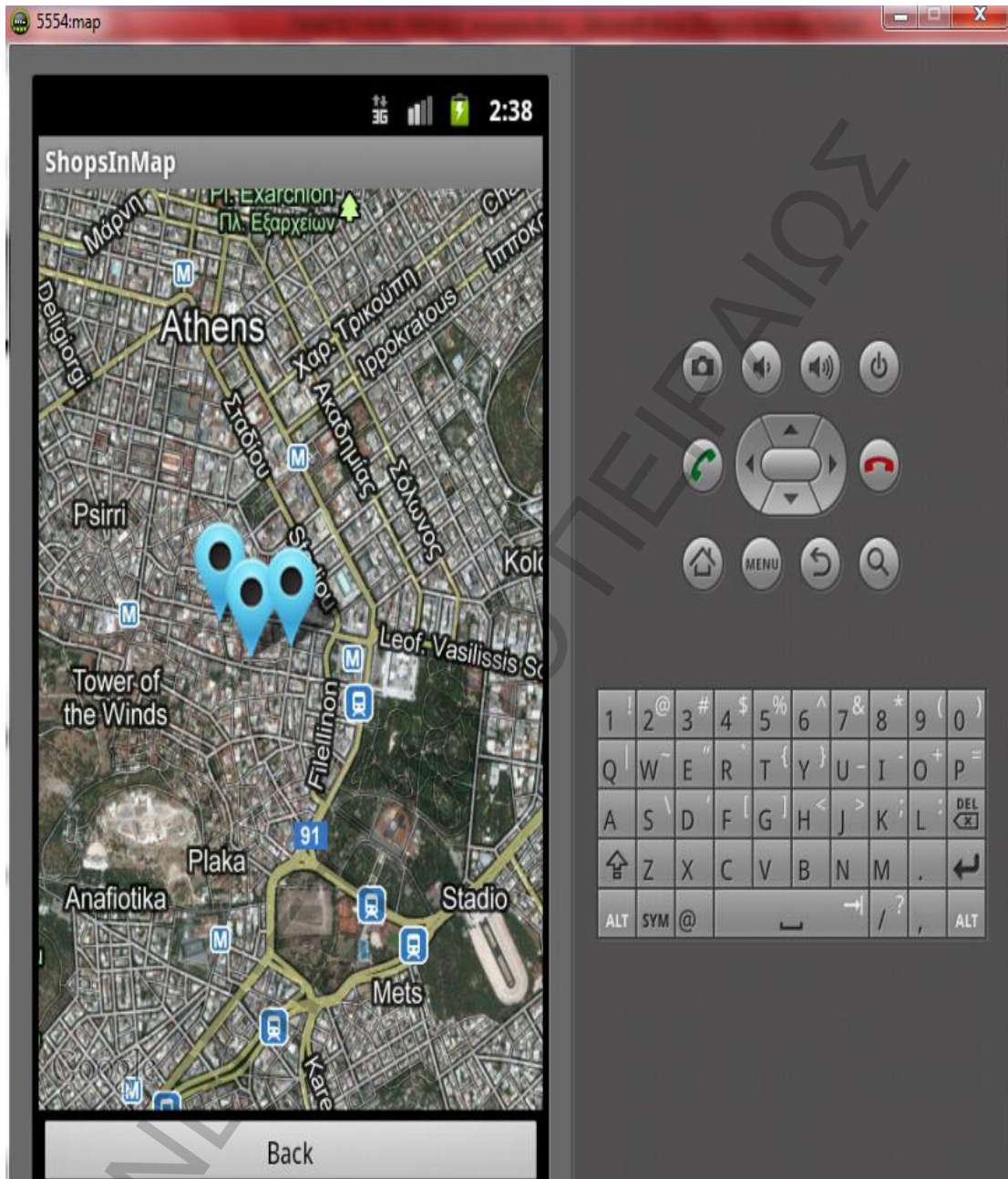
11. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει πάλι Gadgets Shop αλλά αυτή την φορά τρία αστέρια ως ranking.



Εικόνα 23 Activity2.java Activity

12. Έπειτα ο χρήστης επιλέγει το κουμπί show



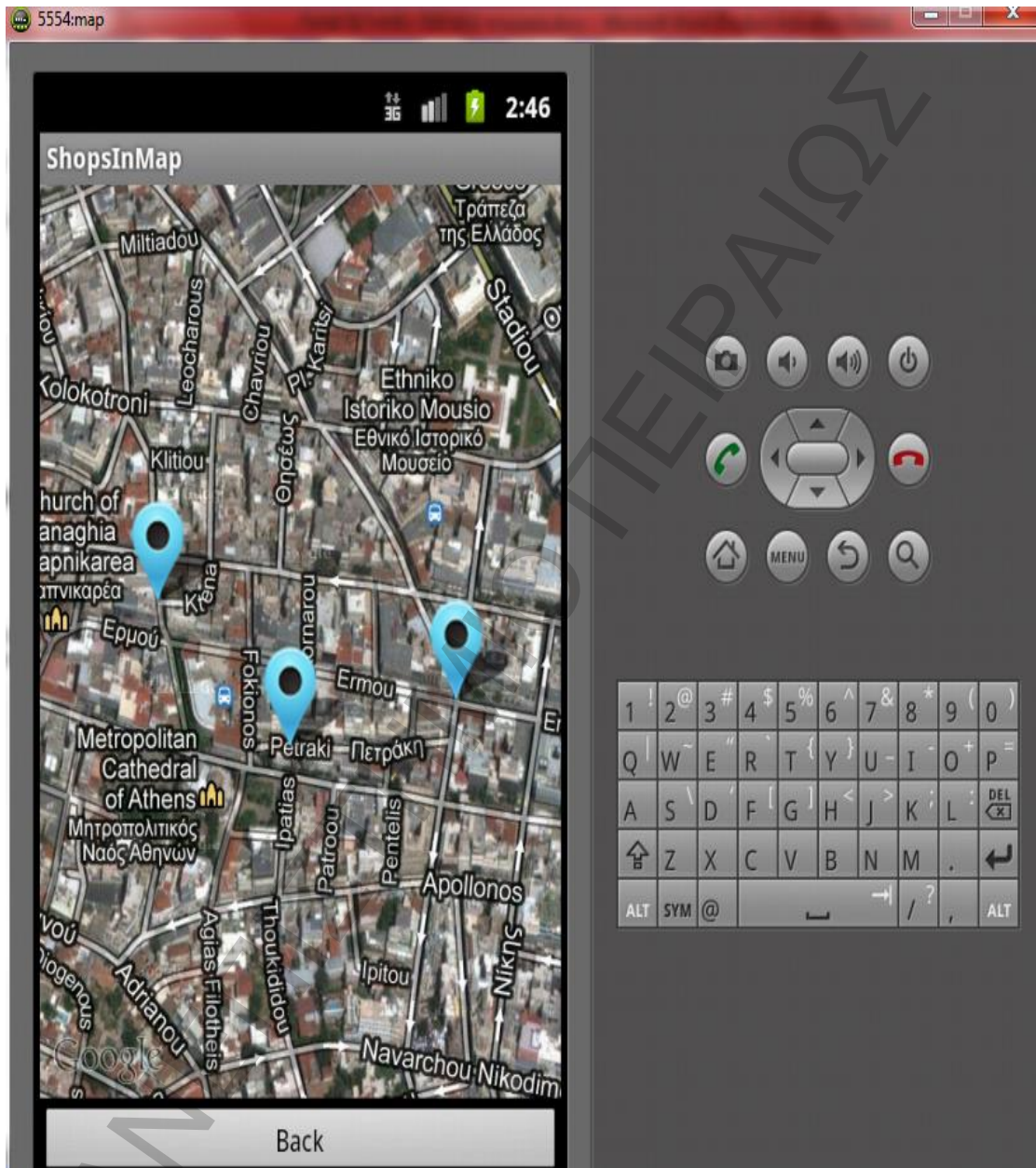
Εικόνα 24 Activity3.java Activity

13. Πατώντας οπουδήποτε πάνω στο χάρτη ενεργοποιούνται οι ιδιότητες zoom in και zoom out του χάρτη.



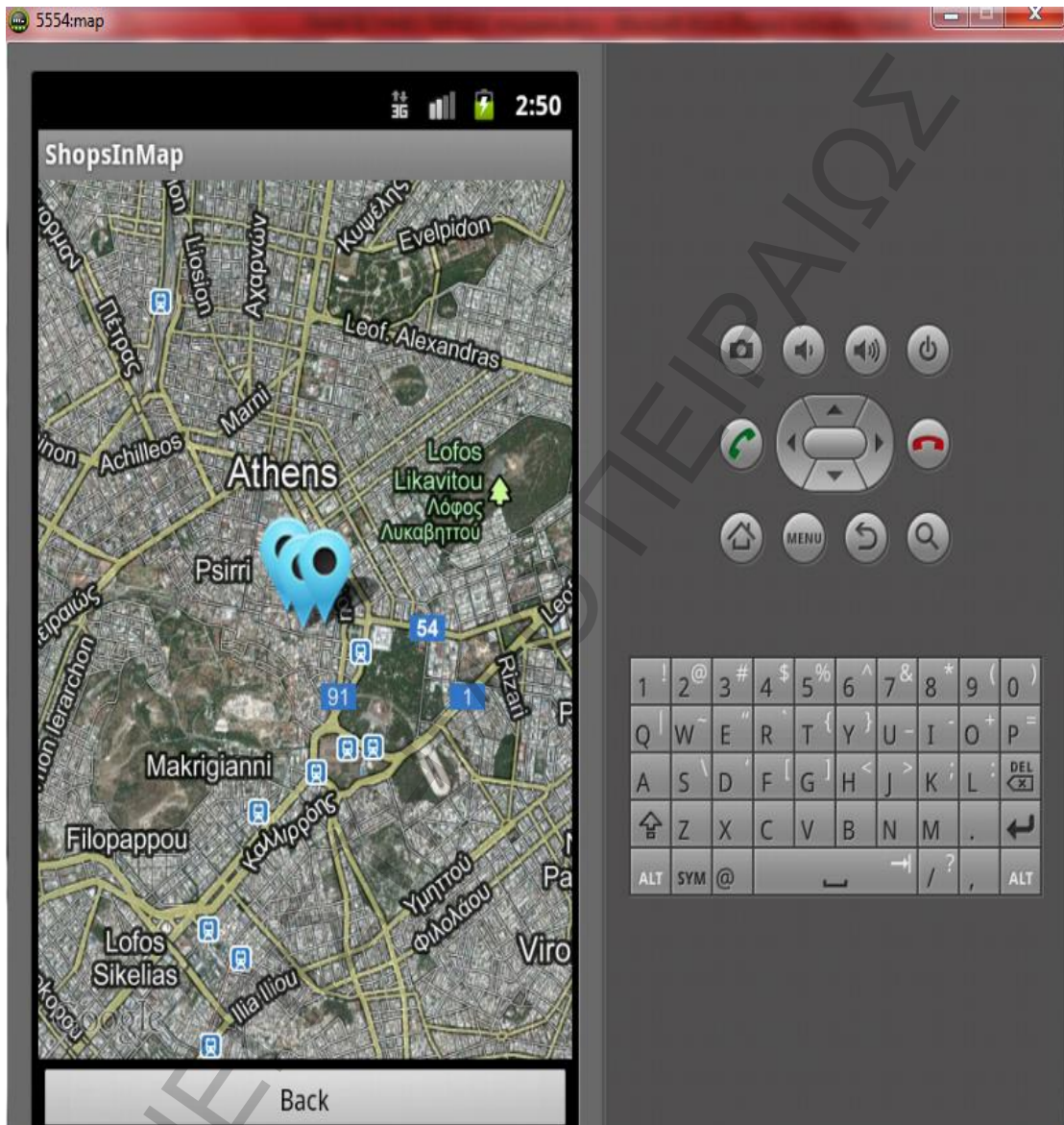
Εικόνα 25 Activity3.java Activity

14. Ο χρήστης μπορεί κάνοντας click στο εικονίδιο με το + να δει τα αποτελέσματα του από πιο κοντά.



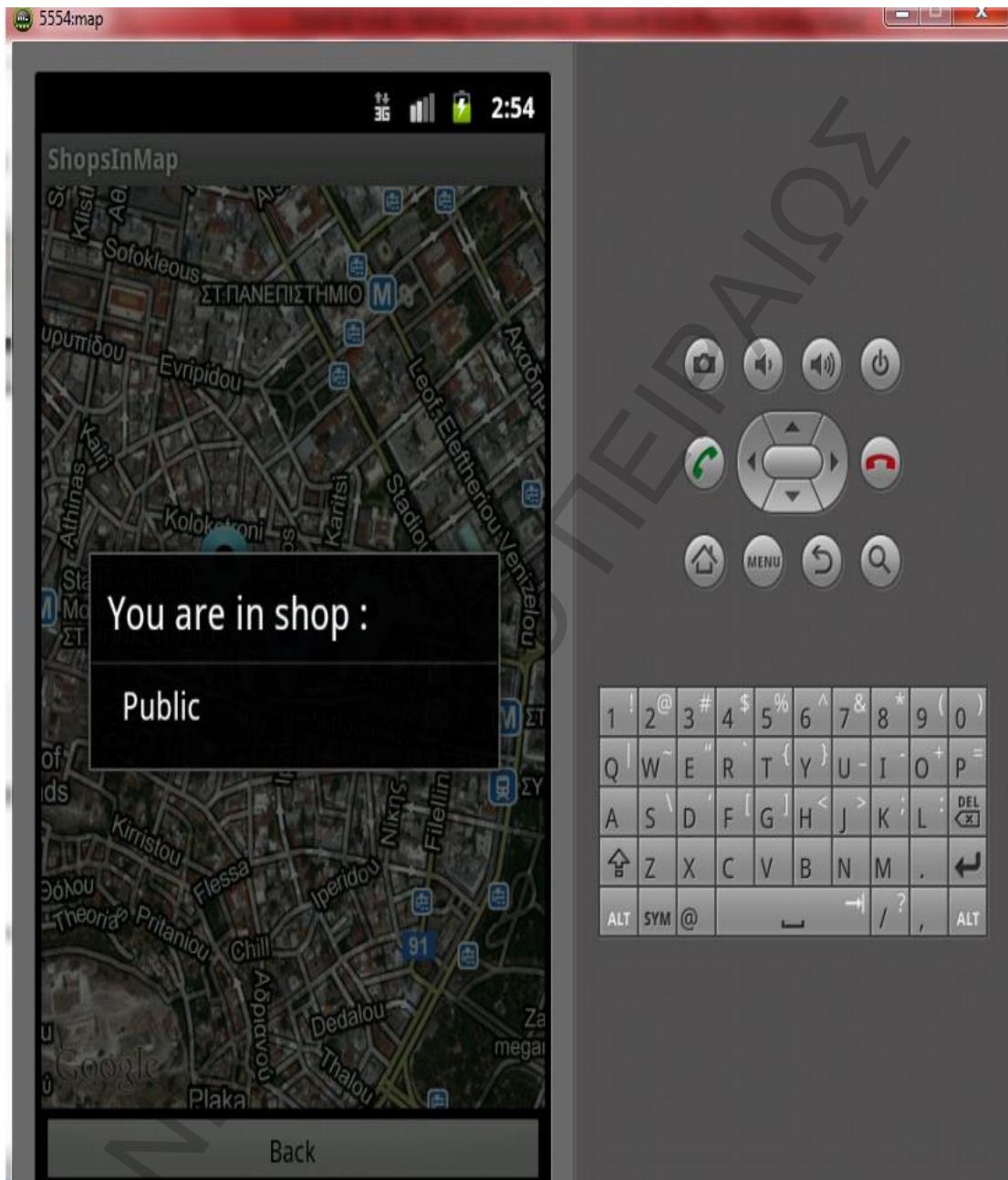
Εικόνα 26 Activity3.java Activity

15.Ο χρήστης μπορεί κάνοντας click στο εικονίδιο με το – να δεί τα αποτελέσματα του από πιο μακριά.



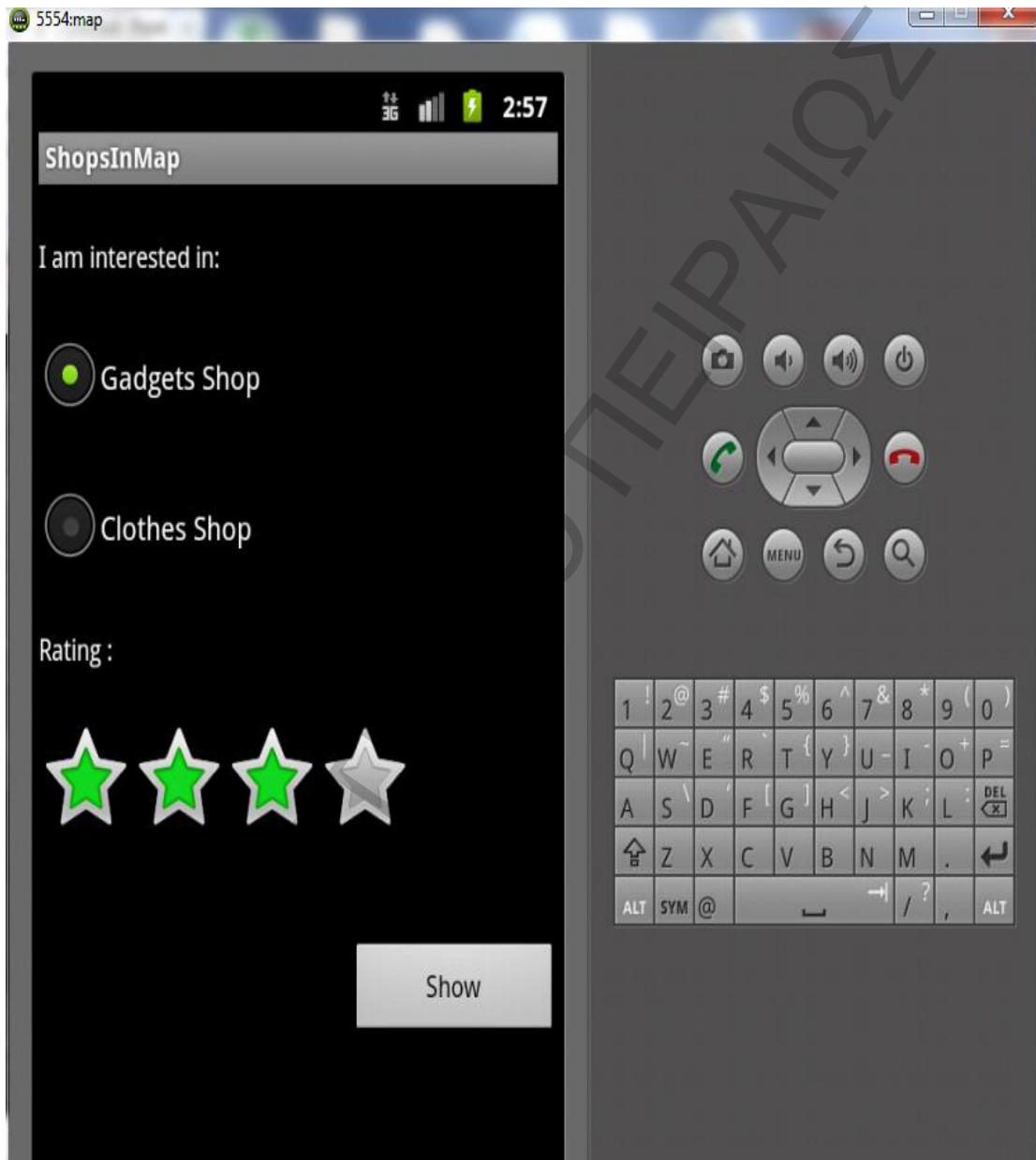
Εικόνα 27 Activity3.java Activity

16. Επιλέγοντας ένα OverlayItem :



Εικόνα 28 OverlayItems.java Activity

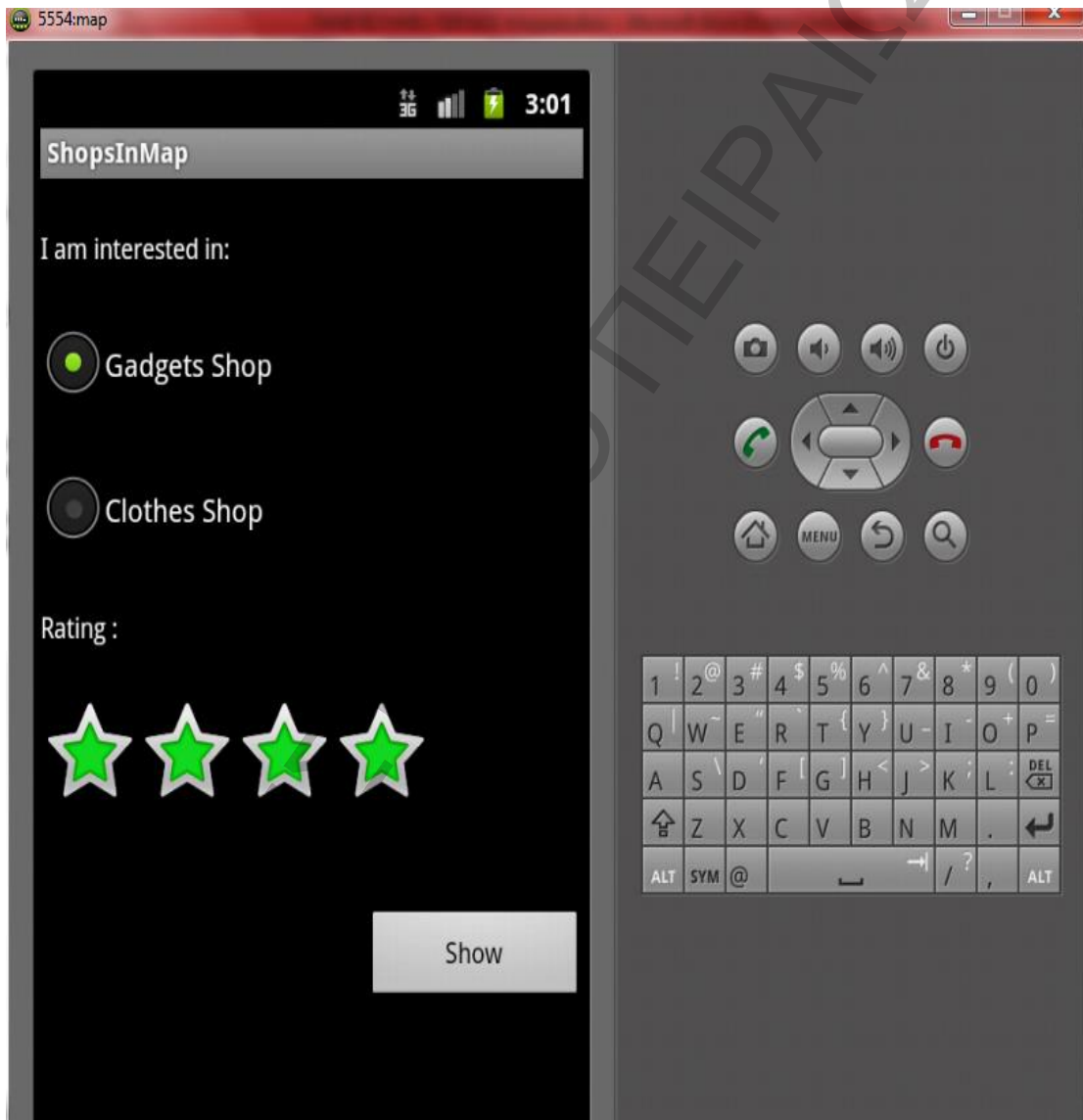
17.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 29 Activity2.java Activity

18. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει πάλι Gadgets Shop αλλά αυτή την φορά τέσσερα αστέρια ως ranking.



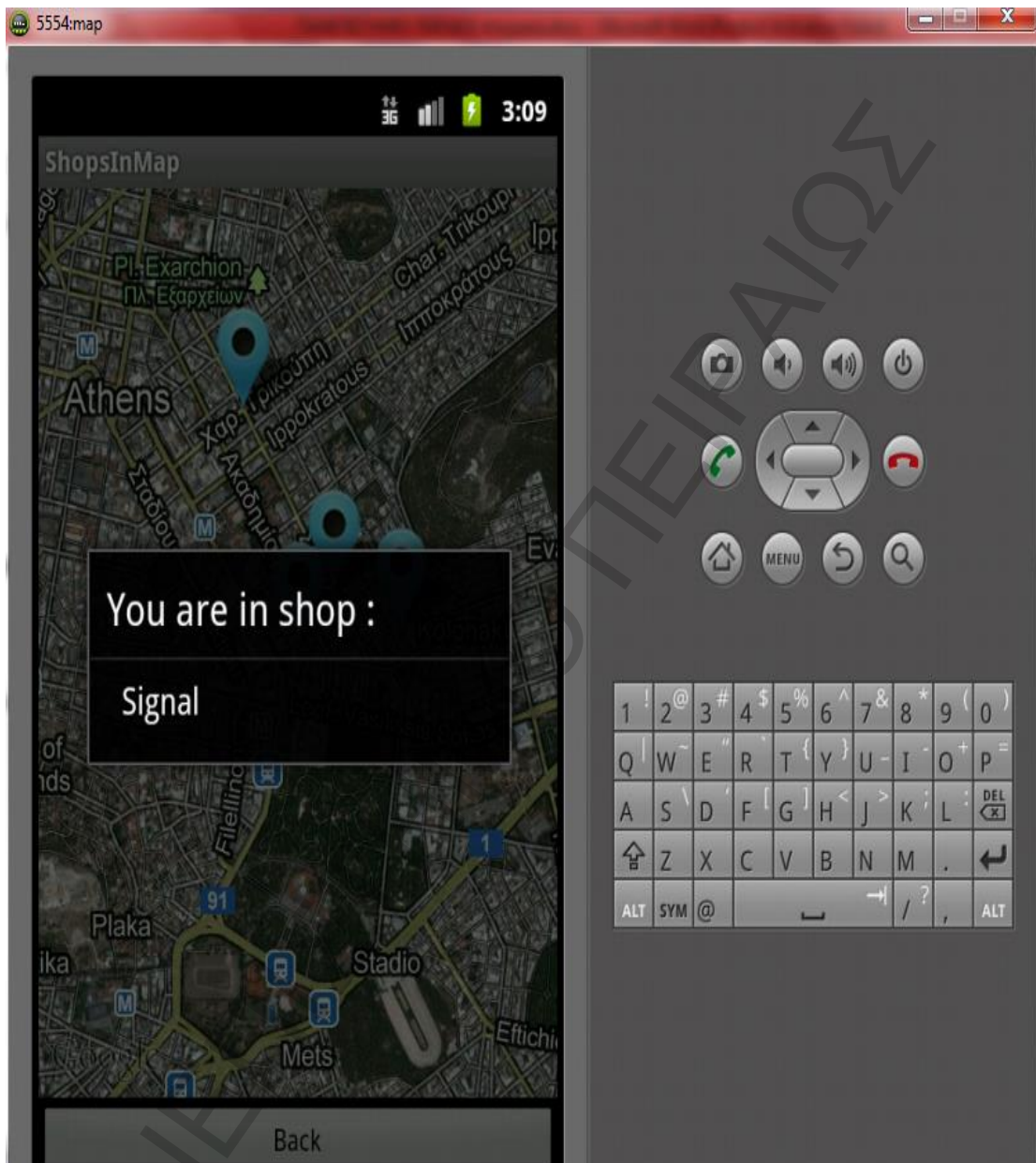
Εικόνα 30 Activity2.java Activity

19. Έπειτα ο χρήστης επιλέγει το κουμπί show



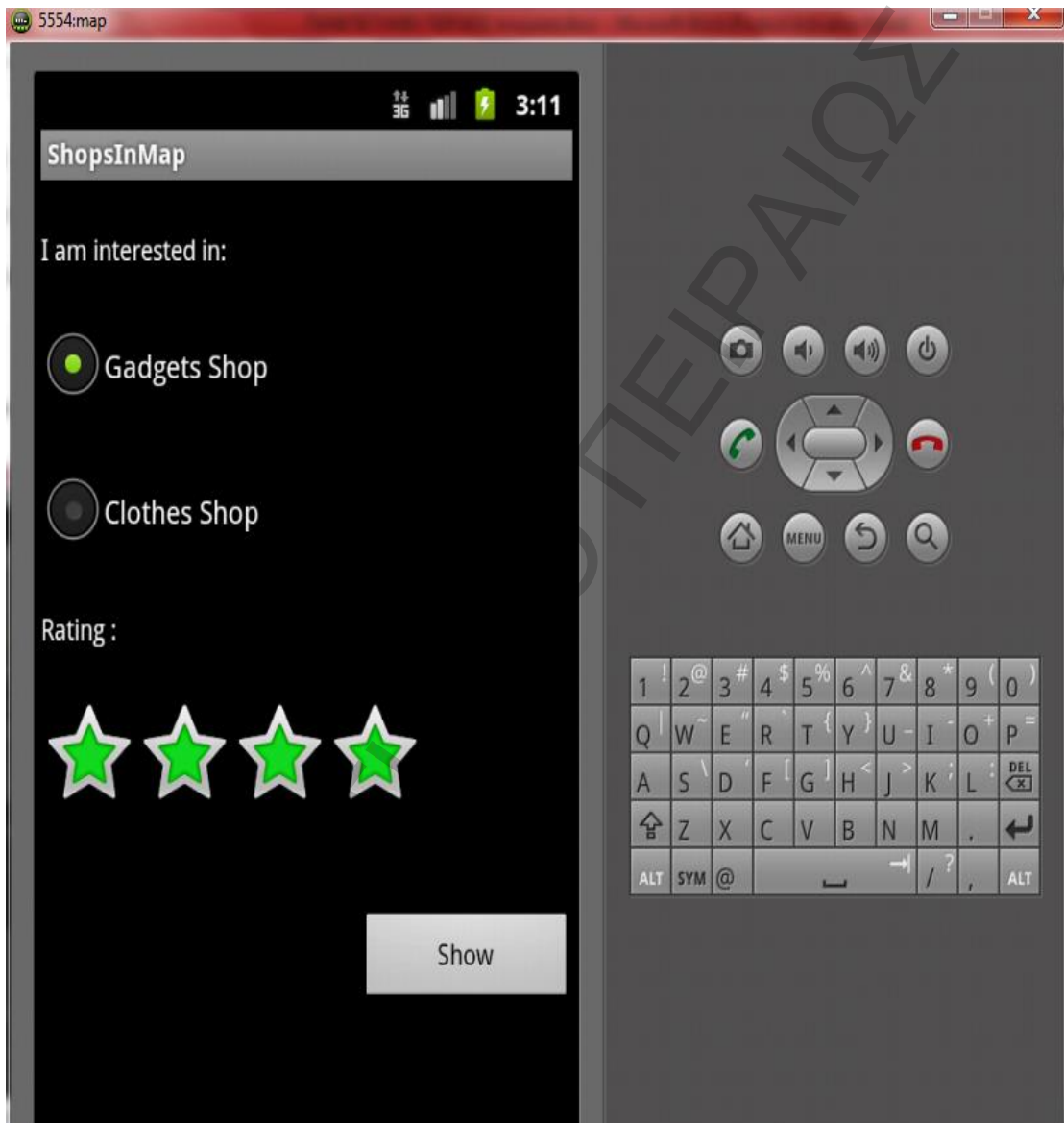
Εικόνα 31 Activity3.java Activity

20. Επιλέγοντας ένα OverlayItem :



Εικόνα 32 OverlayItems.java Activity

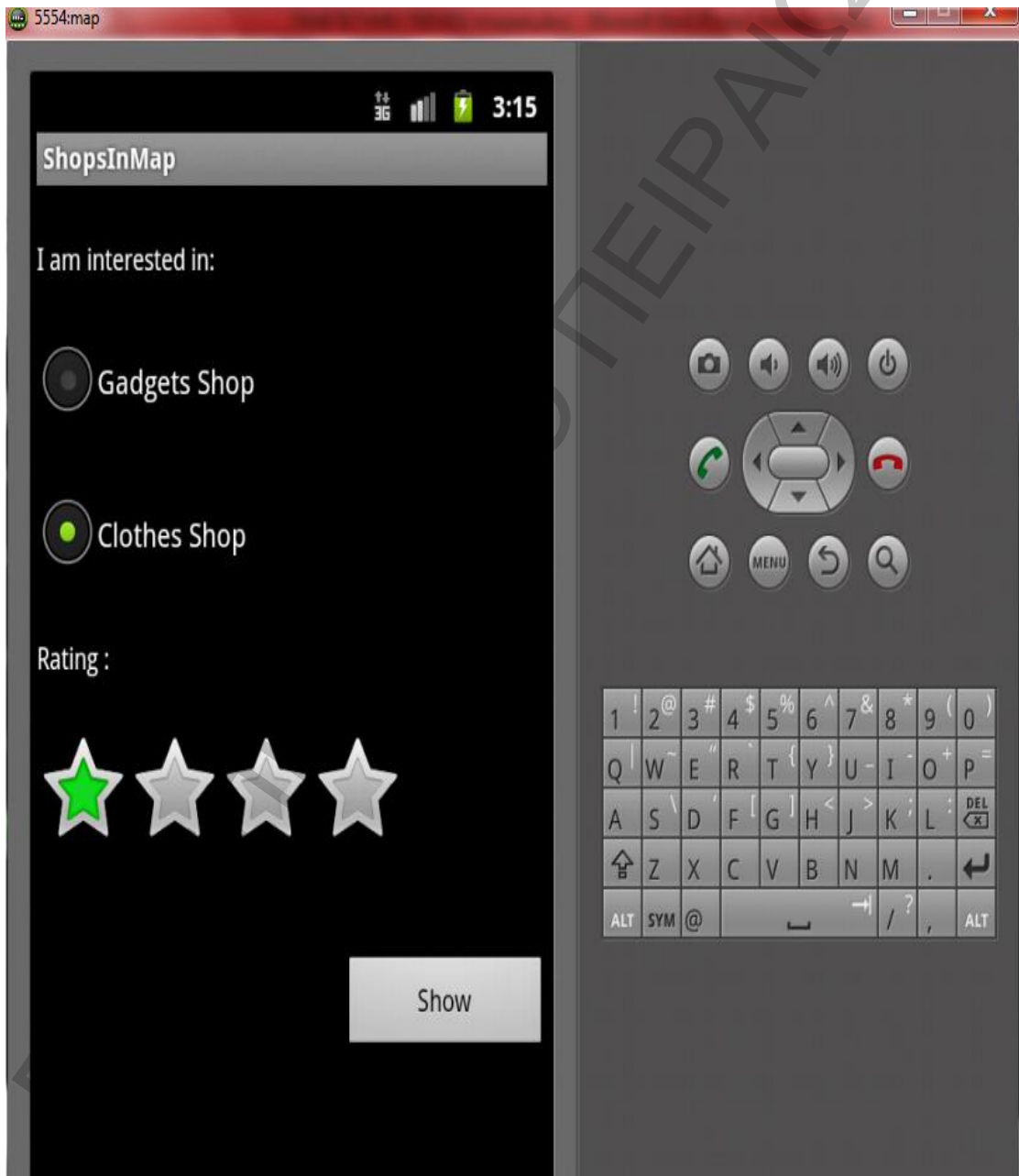
21.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 33 Activity2.java Activity

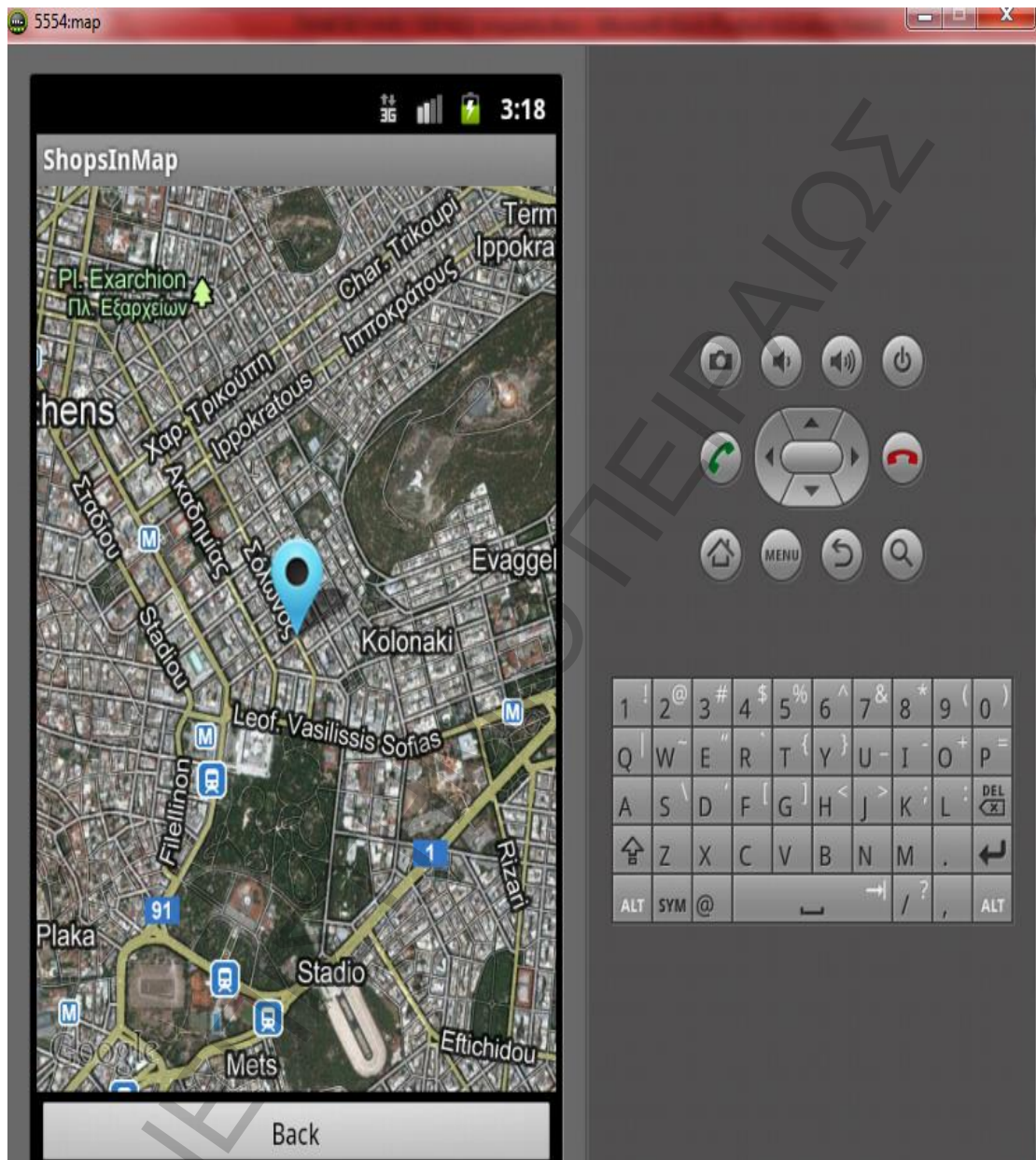
22. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει Clothes Shop αυτή την φορά και ένα αστέρι ως ranking.



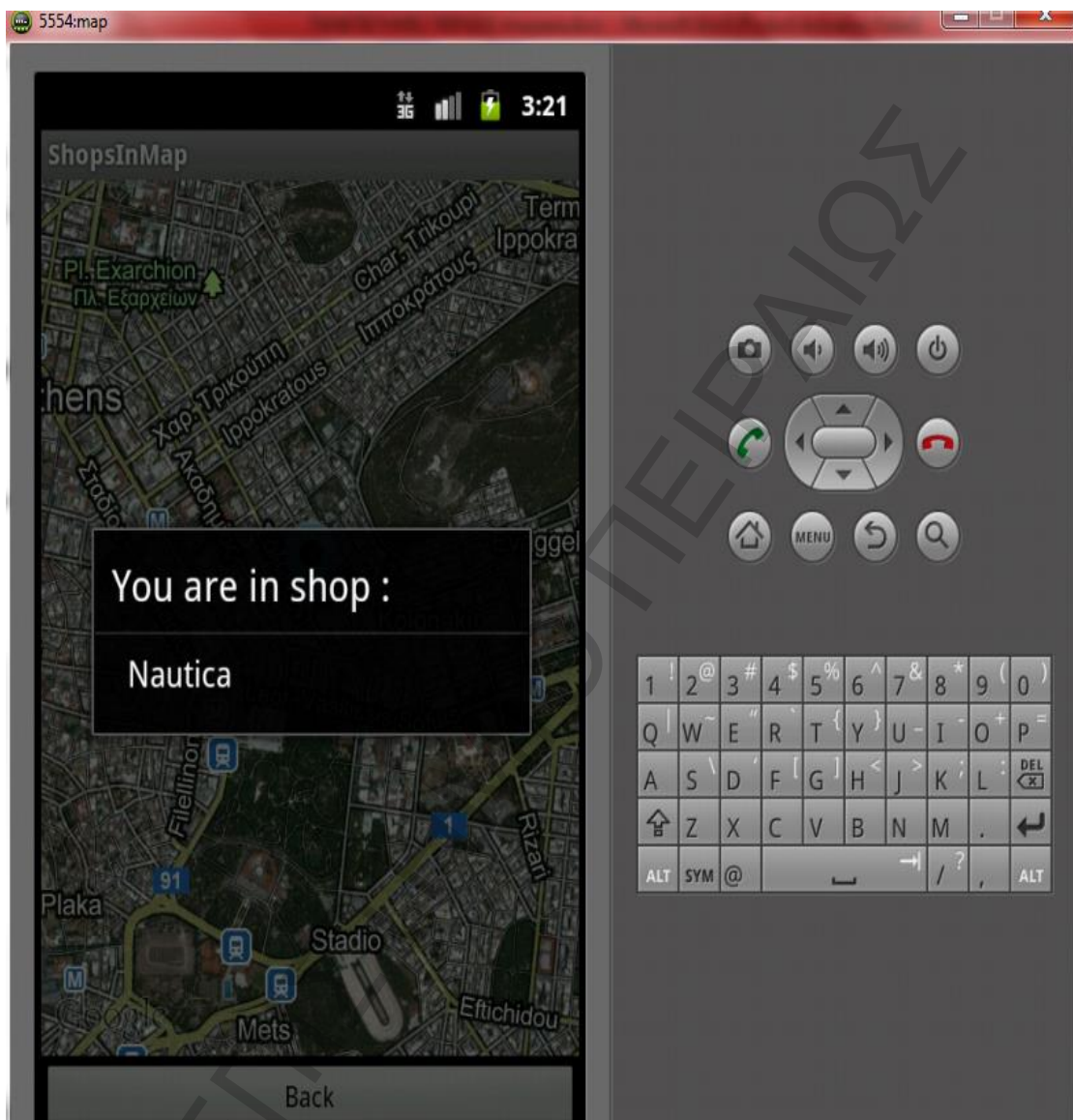
Εικόνα 34 Activity2.java Activity

23. Έπειτα ο χρήστης επιλέγει το κουμπί show



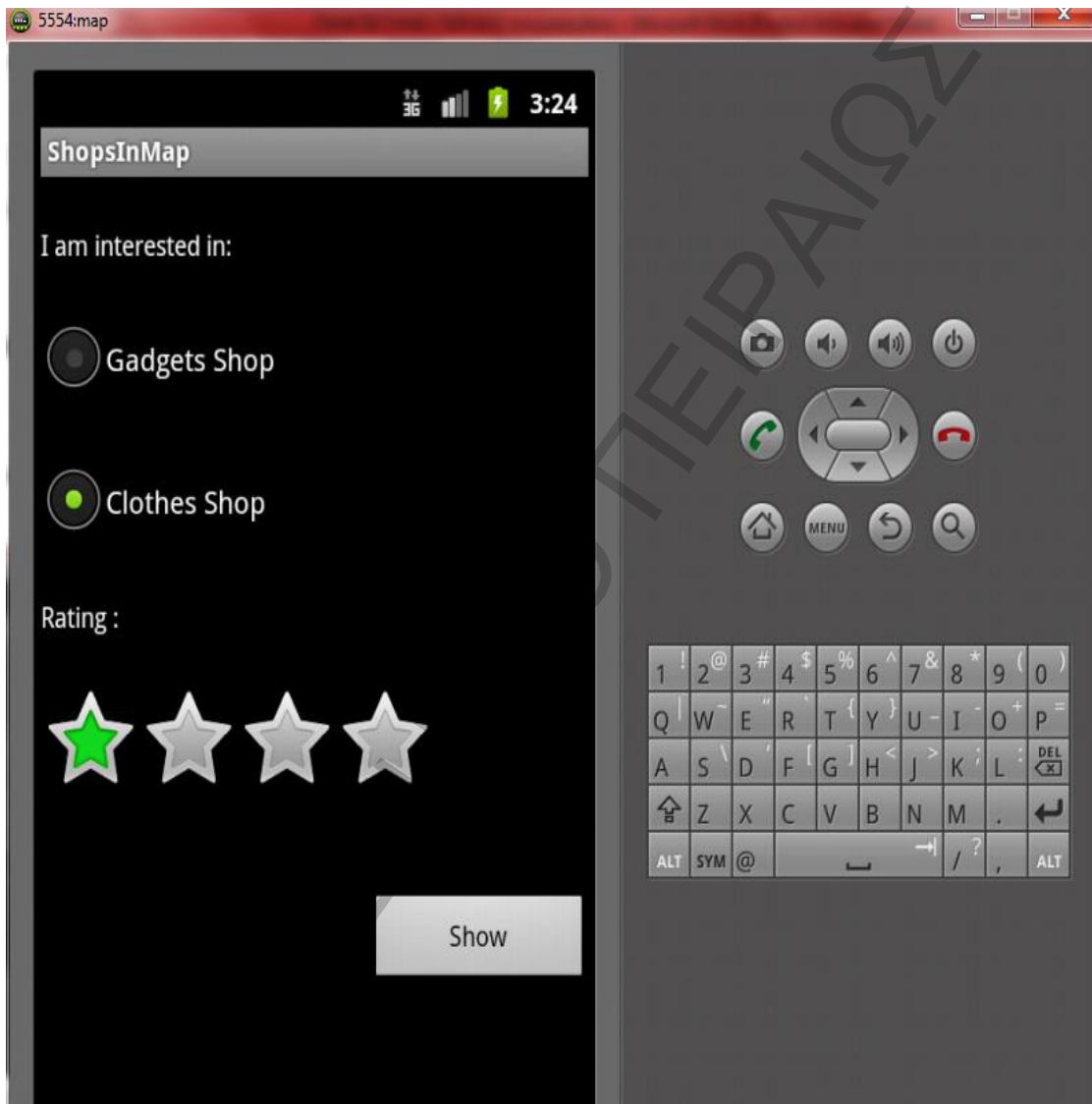
Εικόνα 35 Activity3.java Activity

24. Επιλέγοντας ένα OverlayItem :



Εικόνα 36 OverlayItems.java Activity

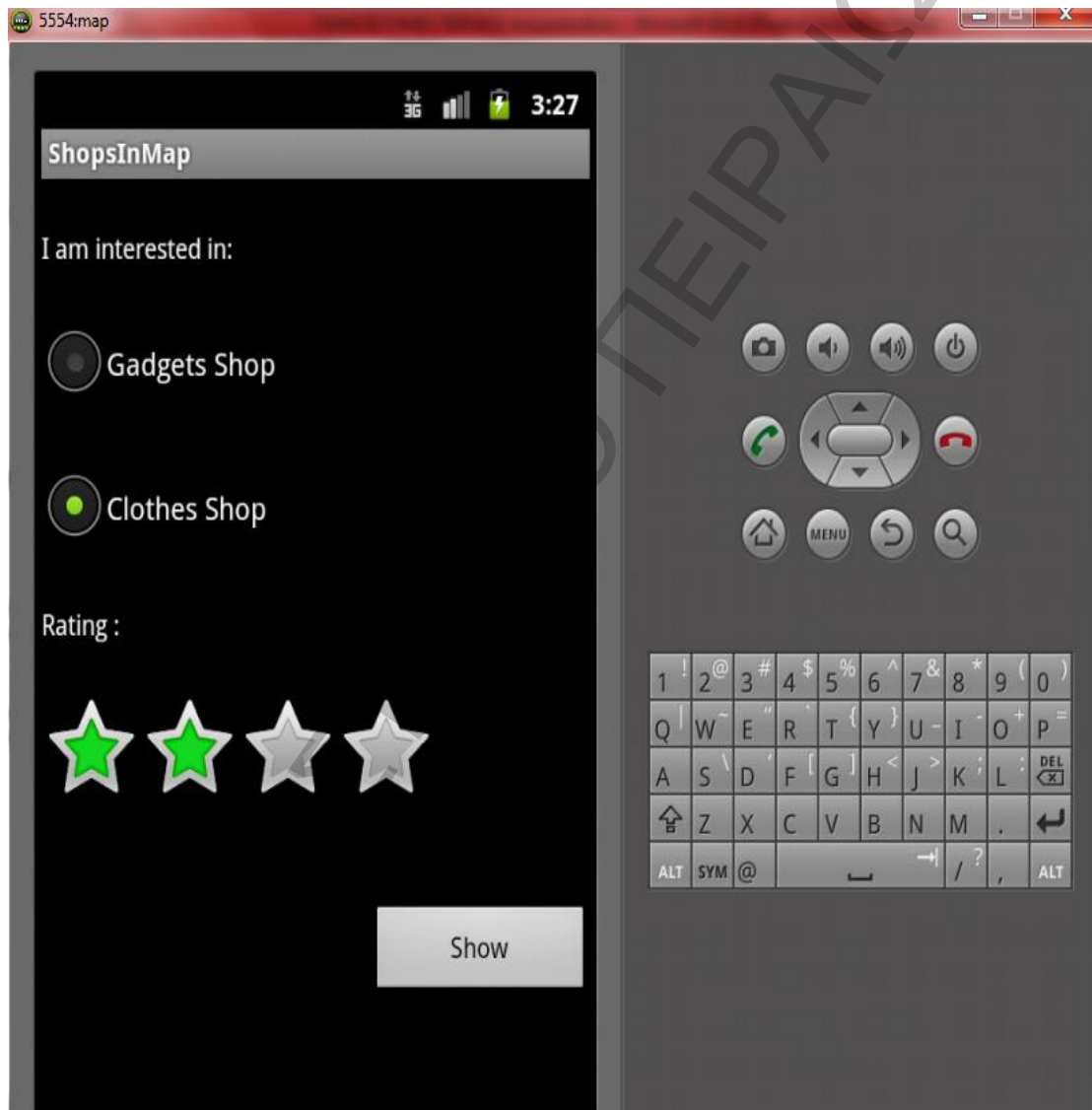
25.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 37 Activity2.java Activity

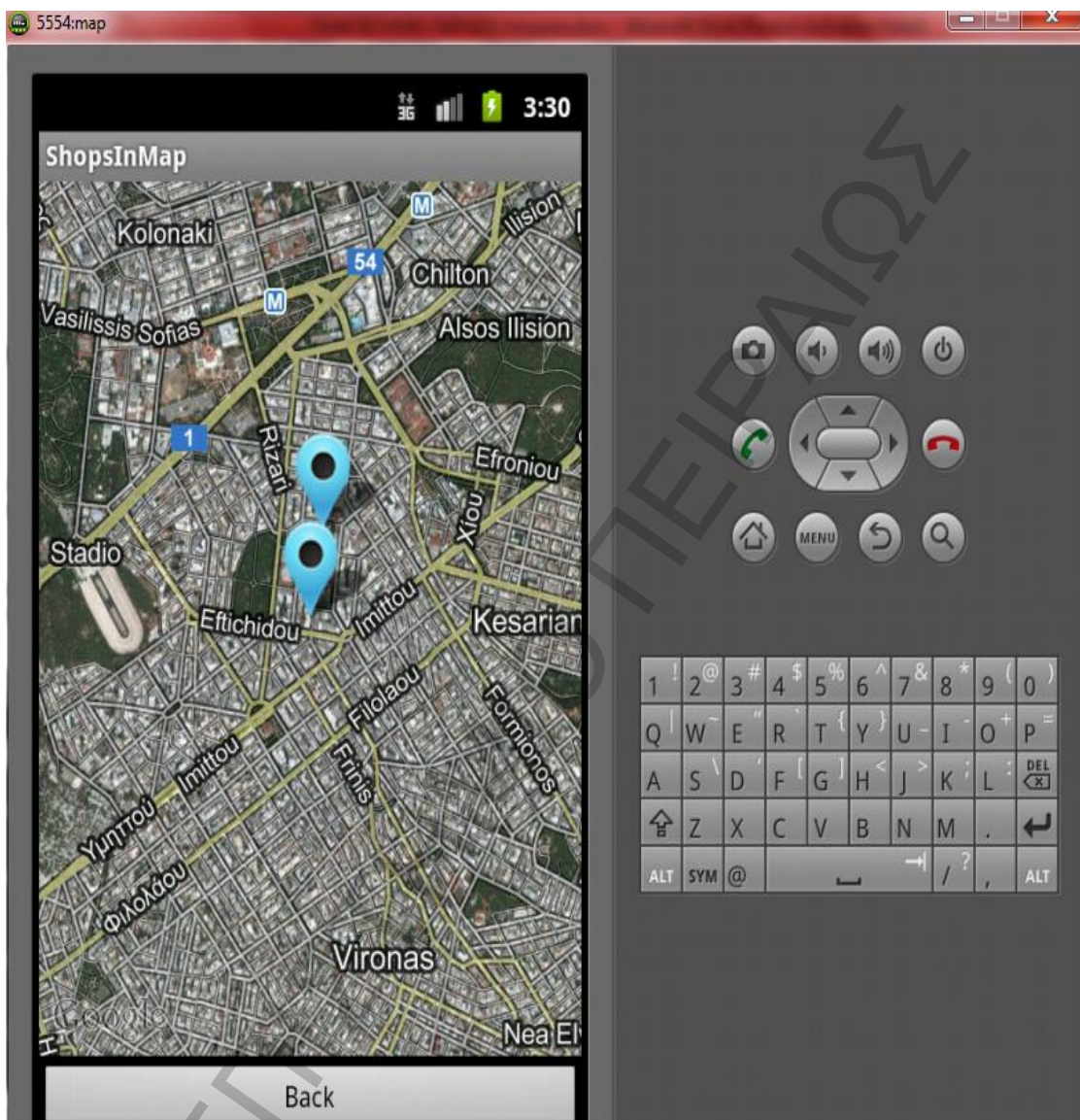
26. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει Clothes Shop και δύο αστέρια ως ranking αυτή την φορά.



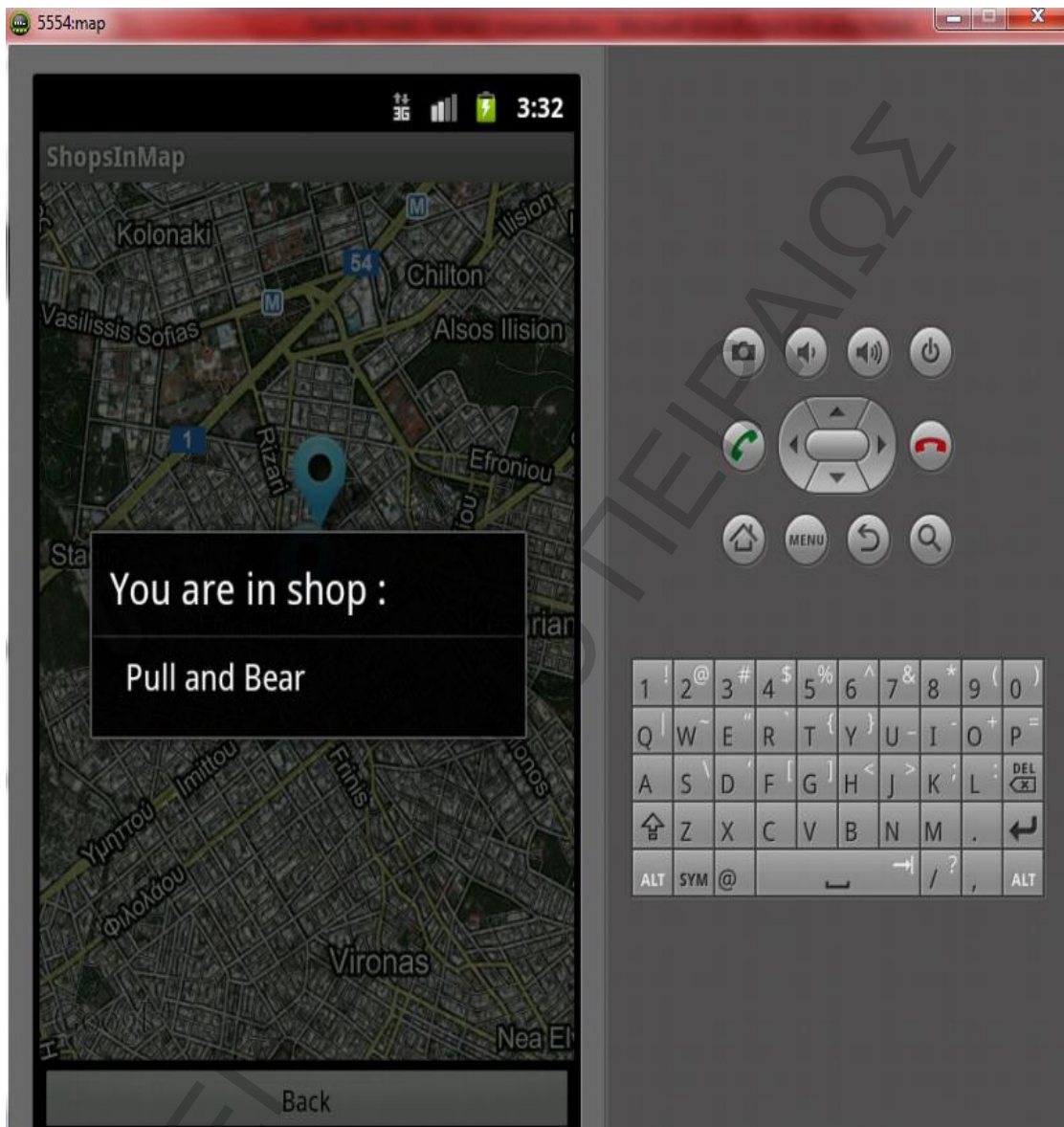
Εικόνα 38 Activity2.java Activity

27. Έπειτα ο χρήστης επιλέγει το κουμπί show



Εικόνα 39 Activity3.java Activity

28. Επιλέγοντας ένα OverlayItem :



Εικόνα 40 OverlayItems.java Activity

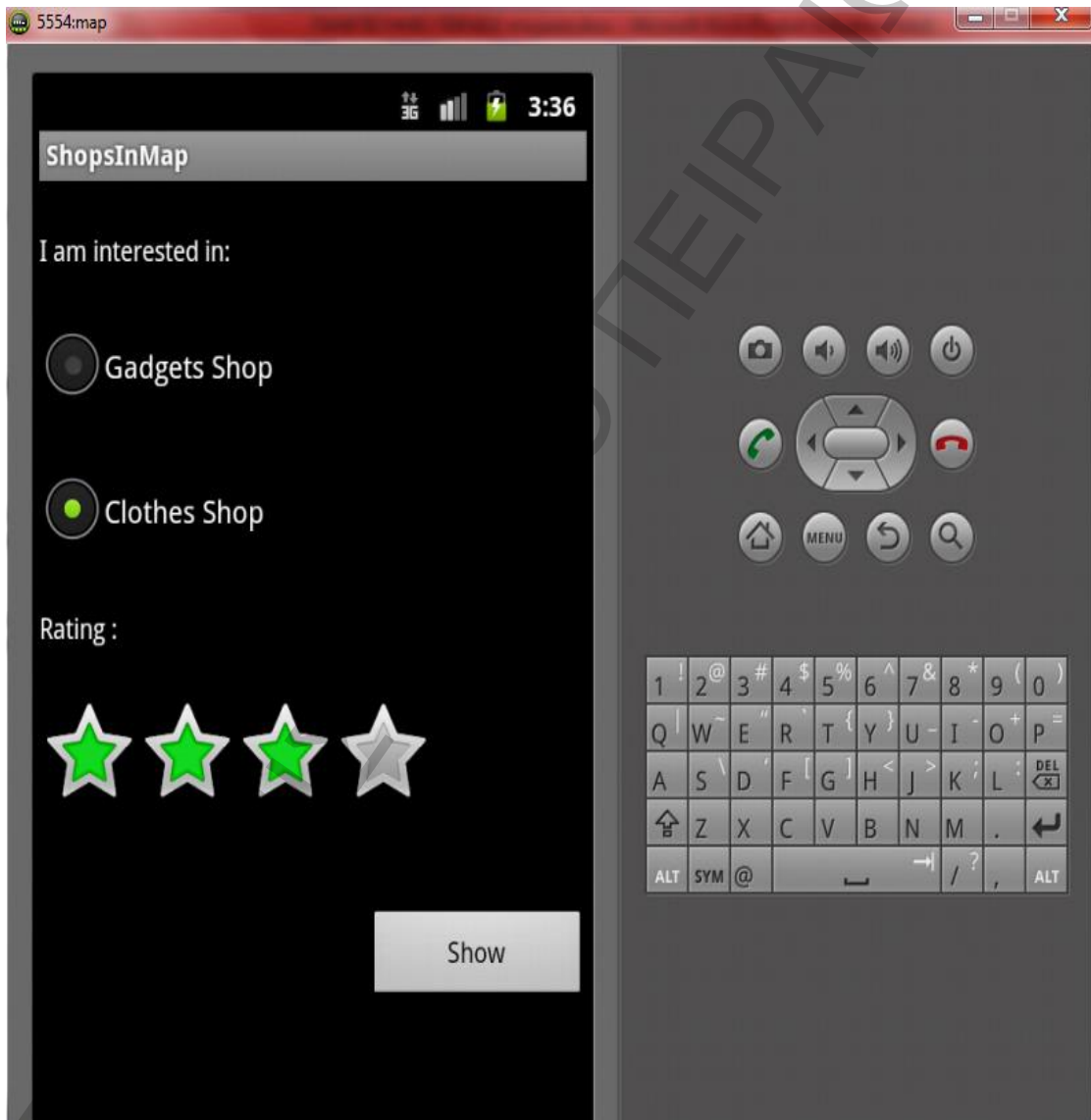
29.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 41 Activity2.java Activity

30. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει Clothes Shop και τρία αστέρια ως ranking αυτή την φορά.



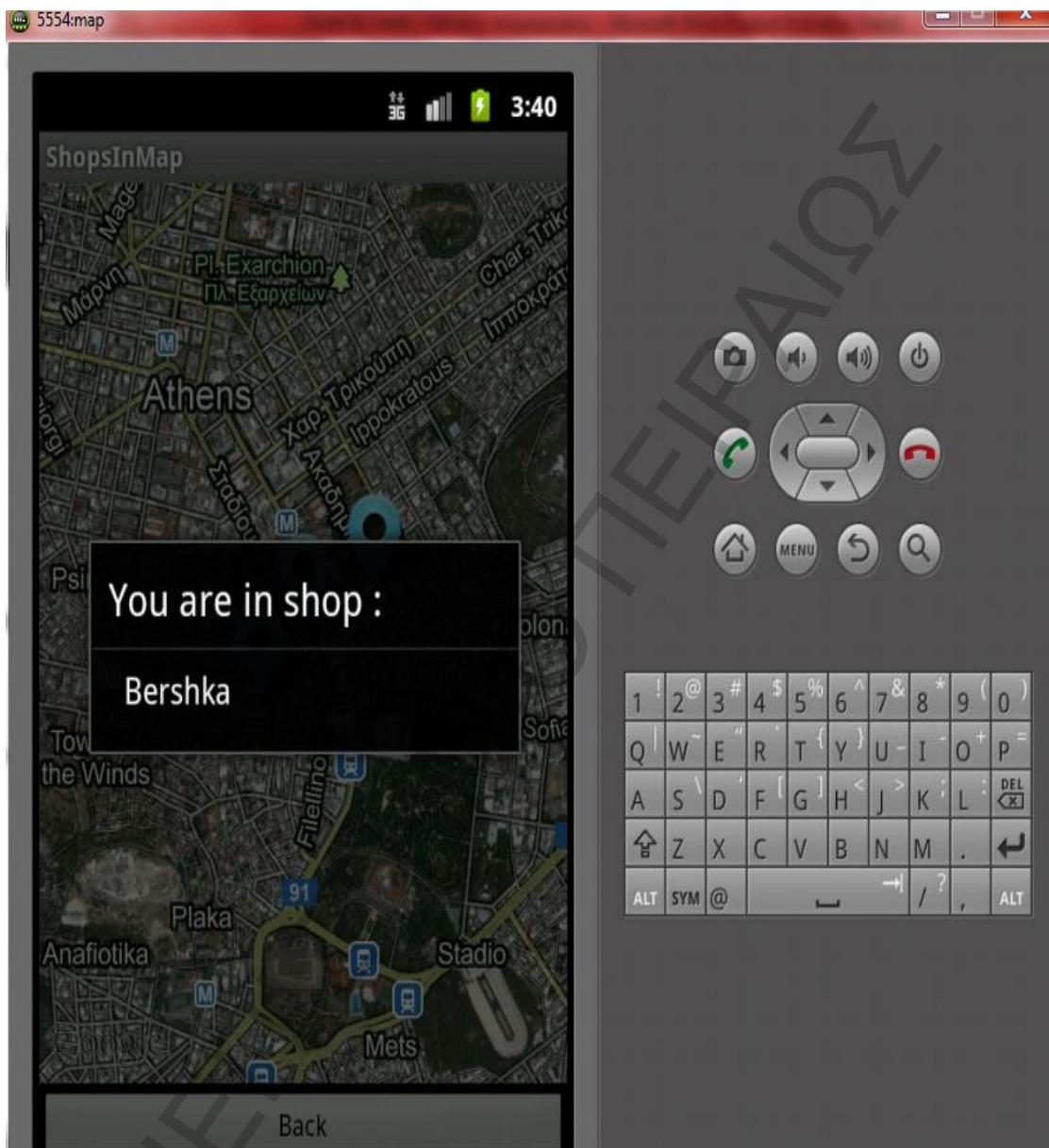
Εικόνα 42 Activity2.java Activity

31. Έπειτα ο χρήστης επιλέγει το κουμπί show



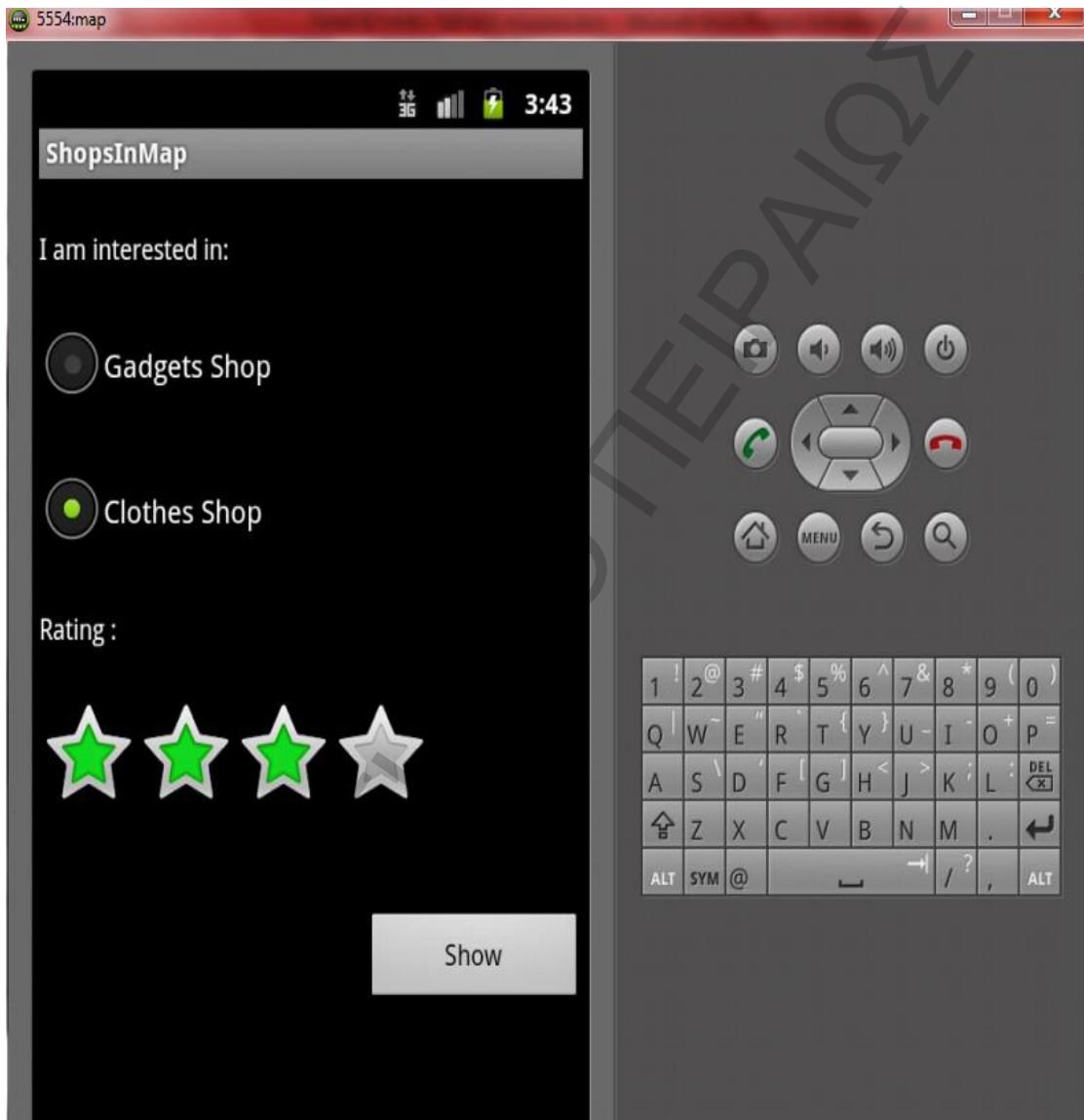
Εικόνα 43 Activity3.java Activity

32. Επιλέγοντας ένα OverlayItem :



Εικόνα 44 OverlayItems.java Activity

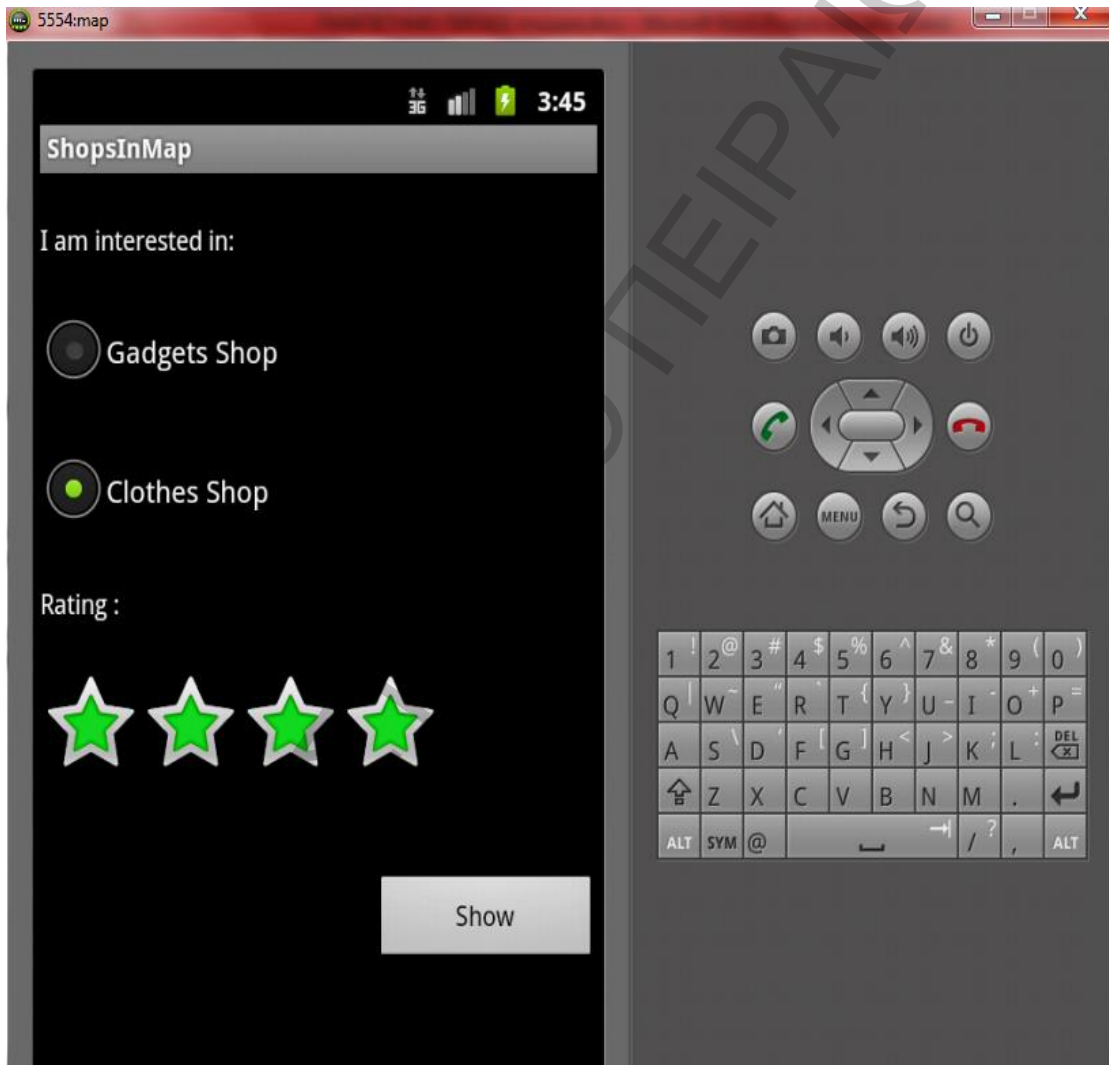
33.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 45 Activity2.java Activity

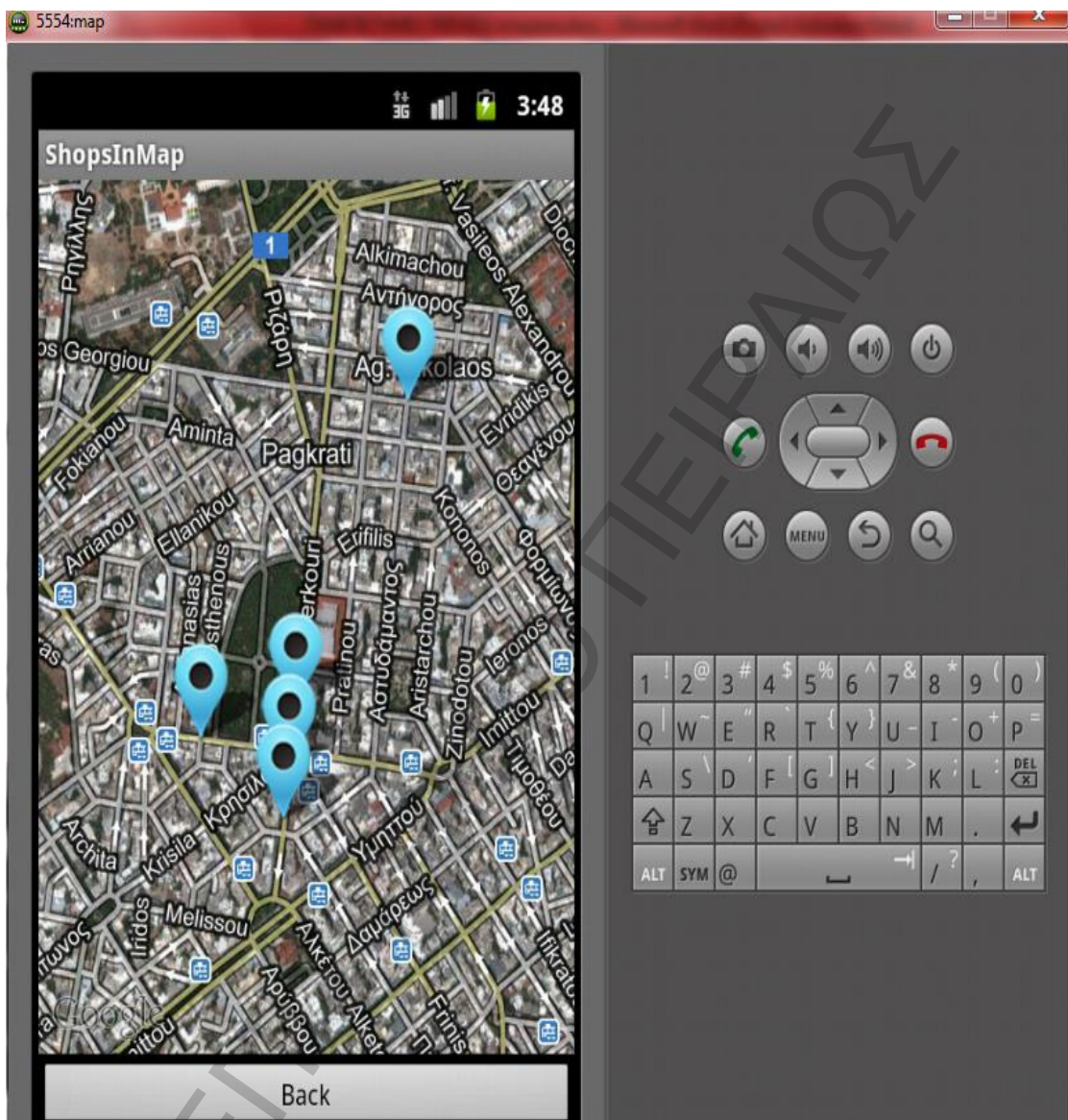
34. Από εδώ ο χρήστης μπορεί να κάνει μια νέα αναζήτηση αλλάζοντας τα προηγούμενα κριτήρια αναζήτησης του.

- Έστω ότι ο χρήστης επιλέγει Clothes Shop και τέσσερα αστέρια ως ranking αυτή την φορά.



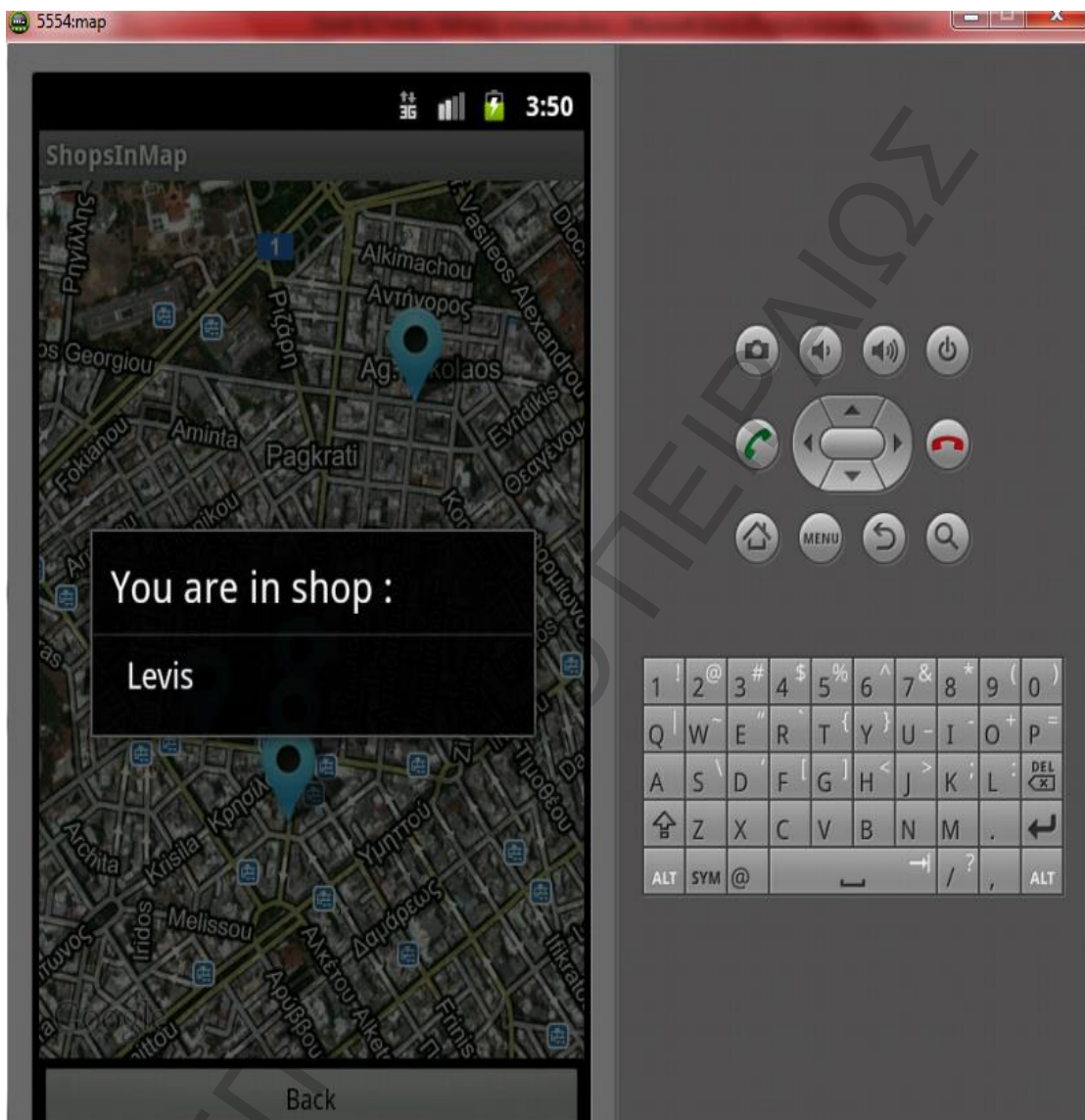
Εικόνα 46 Activity2.java Activity

35. Έπειτα ο χρήστης επιλέγει το κουμπί show



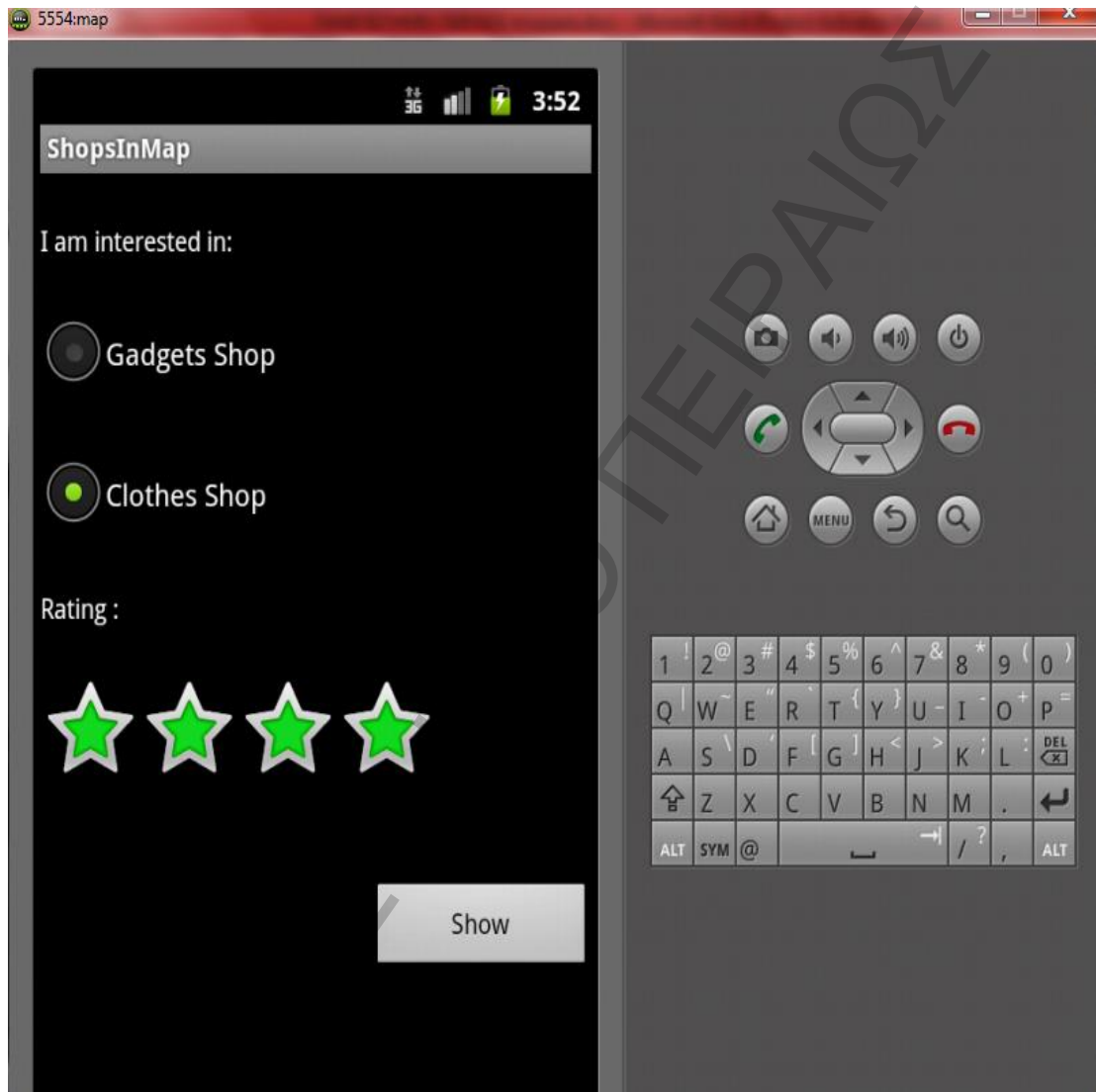
Εικόνα 47 Activity3.java Activity

36. Επιλέγοντας ένα OverlayItem :



Εικόνα 48 OverlayItems.java Activity

37.Επιλέγοντας το κουμπί back ο χρήστης μπορεί να επιστρέψει και πάλι στη φόρμα συμπλήρωσης των κριτηρίων αναζήτησης του.



Εικόνα 49 Activity2.java Activity

7. Συμπεράσματα και προτάσεις περαιτέρω ανάπτυξης

Στο κεφάλαιο αυτό συνοψίζουμε σύντομα κάποια χρήσιμα συμπεράσματα της εφαρμογής που αναπτύχθηκε καθώς και κάποιες κατευθύνσεις για μελλοντικές επεκτάσεις της. Πλέον το Android έχει καταφέρει να εισβάλλει για τα καλά στην παγκόσμια αγορά. Κυριαρχεί σχεδόν σε όλες τις κατασκευάστριες εταιρείες κινητών τηλεφώνων και όχι μόνο, που το επιλέγουν ως κύριο λειτουργικό σύστημα για τις φορητές συσκευές τους. Υπάρχουν χιλιάδες δωρεάν και επί πληρωμή εφαρμογές διαθέσιμες για όλους τους χρήστες του Android. Μετά την ολοκλήρωση της εφαρμογής μπορούμε να βγάλουμε κάποια βασικά συμπεράσματα για όλη την διαδικασία υλοποίησης και να προτείνουμε κάποιες ιδέες για περαιτέρω ανάπτυξη της εφαρμογής.

Όπως είδαμε ο σχεδιασμός μίας Android εφαρμογής είναι μια σχετικά εύκολη και ευχάριστη διαδικασία. Σ' αυτό έχουν συνεισφέρει πολύ τα έτοιμα εργαλεία ανάπτυξης Android λογισμικού που προσφέρει η Google και μπορεί να τα βρει κανείς εντελώς δωρεάν στο Διαδίκτυο. Σε συνεργασία με το προγραμματιστικό περιβάλλον Eclipse, το οποίο είναι ένα εύχρηστο πρόγραμμα ανοιχτού κώδικα, αποτελούν ένα πολύ ισχυρό εργαλείο για όλους τους προγραμματιστές. Για την ανάπτυξη της εφαρμογής μας χρησιμοποιήσαμε τη Java, η οποία είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού, που χρησιμοποιείται κατά κόρον στην ανάπτυξη εφαρμογών.

Γενικά η φιλοσοφία αυτής της πτυχιακής στηρίζεται στην ιδέα του λογισμικού ανοιχτού κώδικα, το οποίο βρίσκει όλο και πιο πολλούς φανατικούς θαυμαστές τελευταία. Το λογισμικό ανοιχτού κώδικα είναι ένα λογισμικό του οποίου ο πηγαίος κώδικας διατίθεται ελεύθερα σε αυτούς που θέλουν να τον εξετάσουν, τροποποιήσουν ή χρησιμοποιήσουν σε άλλες εφαρμογές. Αυτό σημαίνει ότι για προγράμματα ανοιχτού λογισμικού όπως είναι οι Android εφαρμογές υπάρχουν άπειρα παραδείγματα στο Διαδίκτυο. Σ' αυτό ακριβώς στηριχθήκαμε και μας βοήθησε κατά την υλοποίηση της εφαρμογής μας

7.1 Μελλοντική επέκταση

Η εφαρμογή μας παρόλο που είναι μια ολοκληρωμένη εφαρμογή , έχει ακόμα πολλά περιθώρια ανάπτυξης. Μερικές κατευθύνσεις για μελλοντική επέκτασή της είναι:

- Η εφαρμογή θα μπορούσε να χρησιμοποιηθεί για πολλά περισσότερα είδη καταστημάτων όχι μόνο δύο (Gadgets και Clothes) όπως έχει υλοποιηθεί αυτή την στιγμή.
- Επίσης τα καταστήματα αυτά θα μπορούσαν να μην είναι αποθηκευμένα τοπικά σε μια βάση δεδομένων αλλά να αντλούμε τις πληροφορίες αυτές online.

7.2 Βιβλιογραφία

[1] V. Stavroulaki, D. Petromanolakis, P. Demestichas, "Utility-Aware Cognitive Network Selections in Wireless Infrastructures", Wireless Personal Communications Journal, 2010, doi:10.1007/s11277-010-0105-6

[2] Android Developers: What is Android?

<http://developer.android.com/guide/basics/what-is-android.html>

[3] Android versions comparasion

<http://socialcompare.com/en/comparison/android-versions-comparison>

[4] Τι είναι το Android

<http://www.adds.gr/company/technology/what-is-android/>

[5]Millennial Media's May 2011 Mobile Mix Report

<http://www.mobilemarketingwatch.com/millennial-medias-may-2011-mobile-mix-report-showsmajor-gains-across-the-mobile-landscape-16274>

[6] Download the Android SDK

<http://developer.android.com/sdk/index.html>

[7] Android Developers: Platform Versions

<http://developer.android.com/resources/dashboard/platform-versions.html>

[8] Activity Lifecycle

<http://developer.android.com/reference/android/app/Activity.html>

[9] Internet of Things

<http://www.iot-visitthefuture.eu/>

[10] European Future Internet Portal

<http://www.future-internet.eu/home.html>

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Παράρτημα

Κώδικας

MainActivity.Java

```
package com.example.shopsinmap;

import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.widget.Button;

public class MainActivity extends Activity {

    Button create;

    @Override
    public void onCreate(Bundle
savedInstanceState) {

        super.onCreate(savedInstanceState);
```



```

        setContentView(R.layout.activity_main);

        Button
create=(Button)findViewById(R.id.create);

        create.setOnClickListener(new
View.OnClickListener() {

            public void onClick(View view) {

                Intent myIntent = new
Intent(view.getContext(), Activity2.class);

                startActivityForResult(myIntent, 0);

            }

        });

    }

    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {

```

```
getMenuInflater().inflate(R.menu.activity_main,
menu);

    return true;
}
}
```

Activity2.java

```
package com.example.shopsinmap;
import java.io.IOException;
import java.io.InputStream;
import java.util.ArrayList;
import java.util.List;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import org.apache.http.HttpResponse;
import
org.apache.http.client.ClientProtocolException;
import org.apache.http.client.methods.HttpGet;
import
org.apache.http.impl.client.DefaultHttpClient;
import org.xml.sax.InputSource;
import org.xml.sax.XMLReader;
```

```

import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.RadioButton;
import android.widget.RatingBar;

@SuppressLint({ "ShowToast", "ShowToast",
"ShowToast", "ShowToast" })

public class Activity2 extends Activity
implements OnClickListener{

    private String httpURL =
"http://192.168.2.2/PhpProject1/index.php?";
    private String kind;
    private RatingBar rating;

    public void onCreate(Bundle
savedInstanceState){

```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_2);

Button btnShow= (Button)
findViewById(R.id.show);

btnShow.setOnClickListener(this);

RadioButton mobileRadio = (RadioButton)
findViewById(R.id.mobile);

mobileRadio.setOnClickListener(this);

RadioButton clothesRadio =
(RadioButton) findViewById(R.id.clothes);

clothesRadio.setOnClickListener(this);

rating = (RatingBar)
findViewById(R.id.ratingBar1);
}
```

```

private static SAXParserFactory spf =
SAXParserFactory.newInstance();

SAXParser sp = null;

XMLReader xr = null;

public static List<Shop> shopNames = new
ArrayList<Shop>();

public void onClick(View v) {
    switch (v.getId()) {
        case R.id.show:

            try {
                StringBuilder finalUrl = new
StringBuilder(WithURL);
                finalUrl.append("Kind="+kind);

                finalUrl.append("&Star="+rating.getRating());
5000).show();

                DefaultHttpClient httpClient = new
DefaultHttpClient();

                HttpGet getRequest = new
HttpGet(finalUrl.toString());

```

```
    HttpResponse response=  
httpClient.execute(getRequest);
```

```
    InputStream inXML =  
response.getEntity().getContent();
```

```
    httpClient.getConnectionManager().shutdown(  
);
```

```
    if (inXML != null) {  
        ShopParser parser = new  
ShopParser();  
        try {  
            if (sp == null)  
                sp =  
spf.newSAXParser();  
            if (xr == null)  
                xr =  
sp.getXMLReader();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }
```

```

        try {

            xr.setContentHandler(parser);

            xr.parse(new
InputSource(inXML));

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            shopNames =
parser.getShopsNames();

            Log.i("", "FINISH XML
PARSING FOR MAIN");
        }
    }

    Intent myIntent = new
Intent(v.getContext(), Activity3.class);

    startActivityForResult(myIntent, 0);

} catch (ClientProtocolException e1) {

```

```
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
```

```
        break;
    case R.id.mobile:
        kind = "Mobile";//Mobile
        break;
    case R.id.clothes:
        kind = "Clothes";
        break;
    default:
        break;
    }
}
```



```
}
```

Activity3.java

```
package com.example.shopsinmap;  
import java.util.List;  
import android.graphics.drawable.Drawable;  
import android.os.Bundle;  
import android.view.View;  
import com.google.android.maps.GeoPoint;  
import com.google.android.maps.MapActivity;  
import com.google.android.maps.MapController;  
import com.google.android.maps.MapView;  
import com.google.android.maps.OverlayItem;  
  
import com.google.android.maps.Overlay;  
public class Activity3 extends MapActivity {  
  
    MapController mControl;  
    GeoPoint GeoP;  
    MapView MapV;
```

```
@Override

public void onCreate(Bundle
savedInstanceState){

    super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_3);

        MapV=(MapView)
findViewById(R.id.mvMain);
        MapV.displayZoomControls(true);
        MapV.setBuiltInZoomControls(true);
        MapV.setSatellite(true);

        double lat= 37.9837155;
        double longi= 23.7293097;

        GeoP = new GeoPoint ((int)(lat*1E6) ,
(int) (longi*1E6));

        mControl=MapV.getController();
```

```

mControl.setCenter(GeoP);
mControl.setZoom(13);

for(Shop shop : Activity2.shopNames){
    List<Overlay> mapOverlays =
MapV.getOverlays();

    Drawable drawable =
this.getResources().getDrawable(R.drawable.and
roid1);

    OverlayItems itemizedoverlay = new
OverlayItems(drawable, this);

    float lats =
Float.valueOf(shop.getLat());

    float longis =
Float.valueOf(shop.getLongt());

    GeoPoint point = new GeoPoint
((int)(lats *1E6) , (int) (longis *1E6));

    OverlayItem overlayitem =new
OverlayItem(point, "You are in shop :",
shop.getOnoma());

```

```
itemizedoverlay.addOverlay(overlayitem);

        mapOverlays.add(itemizedoverlay);
        mControl.animateTo(point);
        mControl.setZoom(16);
    }

@Override
protected boolean isRouteDisplayed() {

    // TODO Auto-generated method stub
    return false;
}

public void finishActivity(View v){
    finish();
}
}
```

```
}
```

OverlayItems.java

```
package com.example.shopsinmap;

import java.util.ArrayList;

import android.app.AlertDialog;
import android.content.Context;
import android.graphics.drawable.Drawable;

import
com.google.android.maps.ItemizedOverlay;
import
com.google.android.maps.OverlayItem;

public class OverlayItems extends
ItemizedOverlay<OverlayItem> {

    private
    ArrayList<OverlayItem> mOverlays = new
    ArrayList<OverlayItem>();
    private Context mContext;

    public OverlayItems(Drawable
defaultMarker, Context context) {

super(boundCenterBottom(defaultMarker));
    mContext = context;
```

```

    }

    public void
addOverlay(OverlayItem overlay) {
    mOverlays.add(overlay);
    populate();
}

@Override
protected OverlayItem
createItem(int i) {
    return mOverlays.get(i);
}
@Override
public int size() {
    return mOverlays.size();
}

@Override
protected boolean onTap(int
index) {
    OverlayItem item =
mOverlays.get(index);
    AlertDialog.Builder dialog =
new AlertDialog.Builder(mContext);

    dialog.setTitle(item.getTitle());

    dialog.setMessage(item.getSnippet());
}

```

```
        dialog.show();
        return true;
    }
}
```

Shop.java

```
package com.example.shopsinmap;

public class Shop {
    private String onoma;
    private String lat;
    private String longt;
    private String star;

    public Shop(String onoma,
String lat, String longt, String star) {
        super();
        this.onoma = onoma;
        this.lat = lat;
        this.longt = longt;
        this.star=star;
    }
    public String getOnoma() {
        return onoma;
    }
}
```

```

public void setOnoma(String
onoma) {
    this.onoma = onoma;
}
public String getLat() {
    return lat;
}
public void setLat(String lat) {
    this.lat = lat;
}
public String getLongt() {
    return longt;
}
public void setLongt(String
longt) {
    this.longt = longt;
}
public String getstar(){
    return star;
}
public void setstar(String
star){
    this.star=star;
}
}

```


ShopParser.java

```
package com.example.shopsinmap;

import java.util.ArrayList;
import java.util.List;

import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;

public class ShopParser extends
DefaultHandler {
    @Override
    public void startDocument() {

    }

    private List<Shop> shopNames = new
ArrayList<Shop>();

    public List<Shop> getShopsNames() {
        return shopNames;
    }

    @Override
    public void startElement(String
namespaceURI, String localName, String
qName, Attributes attrs) throws SAXException
{
```

```

        if (localName.equals("result")) {
            Shop shop = new
Shop(attrs.getValue("onoma"),
attrs.getValue("Lat"), attrs.getValue("Long"),
attrs.getValue("asteri"));
            shopNames.add(shop);
        }
    }
}

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/
apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/white"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

    android:padding="10dp"
    android:textColor="#000000"
/>

```

```
<Button
    android:id="@+id/create"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="150dp"
    android:text="Create your profil" />
```

```
</LinearLayout>
```

activity_2.xml

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

```
<TextView
    android:id="@+id/interest"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="clip_vertical"
    android:layout_marginTop="25dp"
```

```
android:text="I am interested in:"  
android:textColor="#ffffff" />
```

```
<RadioGroup
```

```
android:id="@+id/radioShops"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```
>
```

```
<RadioButton
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:textColor="#ffffff"  
        android:layout_marginTop="25dp"  
    android:id="@+id/mobile"  
    android:text="Gadgets Shop" />
```

```
<RadioButton
```

```
    android:id="@+id/clothes"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
        android:layout_marginTop="25dp"  
    android:textColor="#ffffff"  
    android:text="Clothes Shop" />
```

```
</RadioGroup>
```

```
<TextView
```

```
    android:id="@+id/rating"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
        android:layout_marginTop="25dp"  
    android:textColor="#ffffff"
```

```
        android:text="Rating :" />
```

```
<LinearLayout
```

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

```
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:orientation="horizontal" >
```

```
        <RatingBar
```

```
            android:id="@+id/ratingBar1"  
            android:layout_width="wrap_content"  
            android:layout_height="wrap_content"  
            android:layout_marginTop="25dp"  
            android:numStars="4" />
```

```
</LinearLayout>
```

```
<Button
```

```
    android:id="@+id/show"  
    android:layout_width="128dp"  
    android:layout_height="wrap_content"  
    android:layout_gravity="right"  
    android:layout_marginTop="50dp"
```

```
    android:text="Show" />
```

```
</LinearLayout>
```

activity_3.xml

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

    <com.google.android.maps.MapView

        android:id="@+id/mvMain"

        android:apiKey="0FpXNyjuwr1Qxlfnr1JzqTjirv
RwhOPn50xc_TQ"
        android:layout_width="fill_parent"

        android:layout_height="665px"
        android:clickable="true"
    />

    <Button
        android:id="@+id/buttonBack"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_marginTop="447dp"
```

```
        android:onClick="finishActivity"  
        android:text="Back" />
```

```
</RelativeLayout>
```

AndroidManifest.xml

```
<manifest  
xmlns:android="http://schemas.android.com/  
apk/res/android"  
    package="com.example.shopsinmap"  
    android:versionCode="1"  
    android:versionName="1.0" >  
  
    <uses-sdk  
        android:minSdkVersion="8"  
        android:targetSdkVersion="15" />  
    <uses-permission  
android:name="android.permission.ACCESS_  
NETWORK_STATE"/>  
        <uses-permission  
android:name="android.permission.INTERNET  
"/>  
        <uses-permission  
android:name="android.permission.ACCESS_F  
INE_LOCATION"/>  
        <uses-permission  
android:name="android.permission.ACCESS_C  
OARSE_LOCATION"/>
```

```

<application
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"

    >
    <uses-library
android:name="com.google.android.maps"/>

    <activity

        android:name=".MainActivity"

android:label="@string/title_activity_main" >
        <intent-filter>
            <action
android:name="android.intent.action.MAIN"
/>

            <category
android:name="android.intent.category.LAUN
CHER" />
            </intent-filter>

        </activity>
        <activity
android:name=".Activity2"> </activity>
        <activity
android:name=".Activity3"> </activity>
    </application>

```


</manifest>

Index.php

```
<?php
// $format = strtolower($_GET['format']) ==
'json' ? 'json' : 'xml';
header("Content-type: text/xml");
if (isset($_GET['Kind']) &&
isset($_GET['Star'])) {
    if (!( $database =
mysql_connect("localhost", "root", "") ))
        die("Could not connect to database");

    if (!mysql_select_db("android", $database))
        die("Could not open android database");

    mysql_query("SET NAMES 'greek'");

    $query = "SELECT * FROM shop WHERE
Kind='".trim($_GET['Kind'])."' AND
Star='".trim($_GET['Star']).'";

    $rs = "<?xml version=\"1.0\"?>";
    if (!( $result = mysql_query($query,
$database) )) {
```

```

        exit("<results
status=\"error\"><message>Error in
SQL</message></results>");
        die(mysql_error());
    }

    $rs.= "<results statu=\"success\">";
    while ($row = mysql_fetch_array($result))
    {
        $onoma = $row['Name'];
        $diethinsi = $row['Address'];
        $eidos = $row['Kind'];
        $asteri = $row['Star'];
        $Lat = $row['lat'];
        $Long = $row['long'];

        $rs.='<result onoma = "'. $onoma.'"
diethinsh="'. $diethinsi.'" eidos="'. $eidos.'"
asteri="'. $asteri.'" Lat="'. $Lat.'"
Long="'. $Long.'" />';
    }
    //wrap the results in a XML parent
    $retstr=$rs;
    $retstr.="</results>";

    //print_r($retstr);
    //return the XML data!
    echo $retstr;
}
?>

```