



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

| | |
|-----------------------|---|
| Τίτλος Διατριβής | Σημσιολογικές συστάσεις μέσω εφαρμογής android κινητού A semantic recommendations mobile application |
| Όνοματεπώνυμο Φοιτητή | Γεώργιος Πυλιάρος |
| Πατρώνυμο | Παναγιώτης |
| Μητρώνυμο | Γεωργία |
| Αριθμός Μητρώου | ΜΠΠΛ/ 16032 |
| Επιβλέπων | Ευθύμιος Αλέπης, Επίκουρος Καθηγητής |

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Κωνσταντίνος Πατσάκης
Επίκουρος Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευχαριστίες:

Για την περάτωση της παρούσας μεταπτυχιακής διατριβής θα ήθελα να ευχαριστήσω θερμά τον κ. Ευθύμιο Αλέπη, για την επίβλεψη της εργασίας και τις άμεσες και στοχευμένες συμβουλές-λύσεις που μου πρόσφερε καθ' όλη τη διάρκεια της μεταπτυχιακής μου διατριβής για το μεταπτυχιακό πρόγραμμα με τίτλο «ΠΜΣ Πληροφορικής».

Επίσης θα ήθελα να ευχαριστήσω την οικογένειά μου για τη δυνατότητα που μου παρείχαν να συμμετάσχω στο μεταπτυχιακό πρόγραμμα του Πανεπιστημίου Πειραιώς αλλά και για την στήριξή τους κατά τη διάρκειά του.

Περίληψη

Τα συστήματα συστάσεων είναι απαραίτητα για το εμπόριο κινητών τηλεφώνων. Μέσω των εξαιρετικά εξατομικευμένων προϊόντων και υπηρεσιών που προσφέρουν, τα οφέλη είναι τεράστια τόσο για τις εταιρείες όσο και για τους ιδιώτες – χρήστες των κινητών. Μια βασική προϋπόθεση για την εφαρμογή τέτοιων συστημάτων είναι να αποκτήσουν αρκετές πληροφορίες για κάθε μεμονωμένο καταναλωτή μέσω του προφίλ του. Ωστόσο, οι περισσότερες υπάρχουσες προσεγγίσεις δημιουργίας προφίλ σε κινητά υποφέρουν από διαφορά προβλήματα (π.χ. θεωρούνται παρεμβατικές (intrusive) ή δεν είναι ικανοποιητικά επεκτάσιμες (non-scalable)) γεγονός που τις εμποδίζει να υιοθετηθούν στην πραγματικότητα. Μέσα από την προσέγγιση αυτής της μεταπτυχιακής διατριβής, επιδιώκουμε να αντιμετωπίσουμε μερικά από αυτά τα προβλήματα χρησιμοποιώντας ανοικτά προσβάσιμα δεδομένα των χρηστών (π.χ. αρχεία εγκατάστασης εφαρμογών) και προσφέροντας συστάσεις με τη χρήση τεχνολογιών σημασιολογικού ιστού.

Abstract

Recommender systems are essential in mobile commerce to benefit both companies and individuals by offering highly personalized products and services. One key pre-requirement of applying such systems is to gain decent knowledge about each individual consumer through user profiling. However, most existing profiling approaches on mobile suffer problems such as non-real-time, intrusive, cold-start, and non-scalable, which prevents them from being adopted in reality. Through the approach of this postgraduate dissertation, we aim to tackle some of these problems by using openly accessible data of the users (e.g. app installation logs) and making semantic recommendations through the use of semantic web technologies.

Περιεχόμενα

| | |
|---|-----------|
| 1. Εισαγωγή..... | 7 |
| 1.1 Παρουσίαση εφαρμογής..... | 7 |
| 1.2 Σκοπός και στόχοι εργασίας..... | 7 |
| 1.3 Φορητές συσκευές – Ιστορική ανασκόπηση..... | 8 |
| 1.3.1 Έξυπνα τηλέφωνα(Smartphones) | 8 |
| 1.3.2 Υπολογιστής ταμπλέτα(Tablet)..... | 9 |
| 1.3.3 Λειτουργικά Συστήματα για Κινητά | 9 |
| 1.3.4 Android και iOS | 12 |
| 1.4 Εφαρμογές Έξυπνων τηλεφώνων | 15 |
| 1.4.1 Ο κατακερματισμός και τα προβλήματα των προγραμματιστών | 15 |
| 1.4.2 Τύποι εφαρμογών κινητών συσκευών | 16 |
| 1.4.3 Ανάπτυξη εφαρμογών σε διαφορετικά λειτουργικά συστήματα | 19 |
| 1.5 Ανασκόπηση πεδίου | 20 |
| 1.5.1 WiseApp – Recommended Apps..... | 20 |
| 1.5.2 itcher - Movies & TV Shows Recommendations | 21 |
| 1.5.3 Taste - Movie Recommendations | 22 |
| 2. Εγχειρίδιο χρήσης | 23 |
| 1.6 Δημιουργία Λογαριασμού και είσοδος | 23 |
| 1.7 Ανάκτηση κωδικού..... | 25 |
| 1.8 Εξερεύνηση των κατηγοριών προτάσεων | 26 |
| 1.8.1 Δραστηριότητα για προτεινόμενα μέρη..... | 28 |
| Δραστηριότητα για προτεινόμενες ιστοσελίδες..... | 30 |
| Δραστηριότητα για προτεινόμενες εφαρμογές..... | 31 |
| 3. Αρχιτεκτονική Συστήματος..... | 32 |
| 1.9 Γενικά | 32 |
| 1.10 Βάση δεδομένων | 33 |
| 1.11 Web Service | 37 |
| 1.12 Android Εφαρμογή..... | 39 |
| 4. Μελλοντικές επεκτάσεις | 44 |
| 5. Βιβλιογραφία..... | 45 |
| 1.13 Βιβλιογραφικές Πηγές..... | 45 |
| 1.14 Ιστογραφία – Δικτυότοποι..... | 45 |

Πίνακας Εικόνων

| | |
|--|----|
| Figure 1: WiseApp – Recommended Apps | 20 |
| Figure 2: Itcher – Movie & TV show Recommendations | 21 |
| Figure 3: Taste – Movie recommendations | 22 |
| Figure 4: Εικόνα εισόδου..... | 23 |
| Figure 5: Δημιουργία νέου λογαριασμού..... | 24 |
| Figure 6: Ανάκτηση χαμένου κωδικού..... | 25 |
| Figure 7: Welcoming Screen | 26 |
| Figure 8: : Κατηγορίες προτάσεων | 27 |
| Figure 9: Προετοιμασία δραστ. Places | 28 |
| Figure 10: Δραστηριότητα Places | 29 |
| Figure 11: Βαθμολόγηση προτάσεων..... | 29 |
| Figure 12: Δραστηριότητα προτεινόμενων ιστοσελίδων | 30 |
| Figure 13: Δραστηριότητα προτεινόμενων εφαρμογών | 31 |
| Figure 14: Individual GeorgePyliaros..... | 34 |
| Figure 15: Protégé: Individual Excel | 35 |
| Figure 16: Protégé κλάση classTwoPhone..... | 36 |
| Figure 17: http://.../recommendations-web-service/webapi/..... | 37 |
| Figure 18: http://.../recommendations-web-service/webapi/auth | 37 |
| Figure 19: http://.../recommendations-web-service/webapi/apps | 37 |
| Figure 20: Εισαγωγή του email και του μοντέλου του κινητού του χρήστη στην βάση | 38 |
| Figure 21: Εξαγωγή κατηγοριών των εγκατεστημένων εφαρμογών του χρήστη..... | 38 |
| Figure 22: Το λογότυπο της εφαρμογής..... | 39 |
| Figure 23: Η πλατφόρμα της firebase | 40 |
| Figure 24: HTTP POST κλήση για την αποθήκευση πληροφοριών που αφορούν τον χρήστη..... | 41 |
| Figure 25: Χρήση της βιβλιοθήκης AndroidDeviceNames..... | 42 |
| Figure 26: Χρήση της βιβλιοθήκης Jsoup για την εξαγωγή των κατηγοριών των εγκατεστημένων εφαρμογών του χρήστη | 42 |
| Figure 27: Τμήμα κώδικα υπεύθυνο για την δημιουργία πινεζών πάνω στον χάρτη. | 43 |

1. Εισαγωγή

1.1 Παρουσίαση εφαρμογής

Η εφαρμογή «Recommendations Client» είναι μια εγγενής android εφαρμογή που προσπαθεί να προτείνει μέρη, ιστοσελίδες και εφαρμογές ξεχωριστά για τον κάθε χρήστη. Με αυτό εννοούμε ότι οι προτάσεις που γίνονται σε ένα χρήστη στο σύνολο τους είναι διαφορετικές από αυτές που γίνονται σε έναν άλλο.

Για την περάτωση της μεταπτυχιακής διατριβής όλες οι προτάσεις βασίζονται σε τρεις πληροφορίες που αντλούμε από τους χρήστες και σε δυο υποθέσεις τις οποίες ορίζουμε ως δεδομένες.

Η πρώτη υπόθεση ορίζει ότι όσο πιο ακριβό είναι το μοντέλο του χρήστη τόσο πιο ακριβές προτάσεις μπορούν να γίνουν σε αυτόν και η δεύτερη πως αν για παράδειγμα ο χρήστης έχει εγκατεστημένη μια εφαρμογή μουσικής στο κινητό του θα τον ενδιέφερε να μπορεί να βρίσκει όλα τα νυχτερινά μαγαζιά που βρίσκονται κοντά σε αυτόν.

Οι τρεις πληροφορίες που χρησιμοποιούμε είναι:

1. Το μοντέλο του κινητού του χρήστη. Ανάλογα με το πόσο ακριβά είναι τα κινητά των χρηστών χωρίζονται σε τρεις κατηγορίες.
2. Όλες οι εφαρμογές που έχουν εγκατασταθεί από τον ίδιο τον χρήστη και οι κατηγορίες στις οποίες ανήκουν.
3. Το σύνολο των εφαρμογών που είναι εγκατεστημένες από την κοινότητα των χρηστών.

Ο χρήστης μέσα από την εφαρμογή μπορεί να βρει σε μια συγκεκριμένη χιλιομετρική απόσταση από το σημείο που βρίσκεται μαγαζιά, οργανισμούς, εταιρίες αξιοθέατα και άλλα μέρη που είναι πολύ πιθανό να τον ενδιαφέρουν.

Επίσης μπορεί να βρει ιστοσελίδες μουσικής, μαγειρικής – υγιεινής διατροφής, γυμναστικής και άλλες.

Τέλος ανάλογα με τις εφαρμογές που έχει εγκατεστημένες στο κινητό του η εφαρμογή του προτείνει εφαρμογές οι οποίες είναι εγκατεστημένες από χρήστες της κοινότητας με ανάλογα ενδιαφέροντα.

1.2 Σκοπός και στόχοι εργασίας

Τα προσωποποιημένα recommendations προς τον πελάτη είναι μια ραγδαία αναπτυσσόμενη λογική. Για αυτό τον λόγο αυτό είναι και το θέμα της εφαρμογής. Ο απώτερος σκοπός όμως της μεταπτυχιακής διατριβής είναι η ολοκλήρωση ενός έργου που περικλείει ένα μεγάλο κομμάτι των αναγκαίων γνώσεων για την κατασκευή μιας εφαρμογής αλλά και η ενσωμάτωση τεχνολογιών του semantic web σε αυτό.

Μια εφαρμογή από την σκοπιά του χρήστη μπορεί να φαίνεται κάτι απλό. Από την σκοπιά όμως του προγραμματιστή απαιτούνται ποικίλες γνώσεις. Εκτός της δημιουργίας του γραφικού περιβάλλοντος επικοινωνίας με τον χρήστη η εφαρμογή που κατασκευάστηκε επικοινωνεί με μια βάση δεδομένων προκειμένου να γίνει το authentication του χρήστη. Χρησιμοποιεί αρκετά APIs (Application Programming Interface) για την επίλυση συγκεκριμένων προβλημάτων. Χρειάστηκε να στηθεί και ένα web service που αποτέλεσε την γέφυρα επικοινωνίας του android application με την βάση δεδομένων που επίσης στήθηκε. Η βάση δεδομένων μας είναι μια owl οντολογία (Web Ontology Language – έχει καθιερωθεί ως πρότυπο γλώσσας για τον σημασιολογικό ιστό από το World Wide Web Consortium (W3C)).

Επομένως μπορούμε να πούμε ότι με τον σχεδιασμό όλων των παραπάνω αλλά και με την επίτευξη της επικοινωνίας τους μεταξύ τους μπορεί κάποιος να αποκομίσει αρκετές γνώσεις.

1.3 Φορητές συσκευές – Ιστορική ανασκόπηση

Οι φορητές συσκευές αποτελούν ένα ευρύ σύνολο. Στην ενότητα αυτή παρουσιάζονται οι δύο επικρατέστερες φορητές συσκευές (Smartphones, tablet) με τα χαρακτηριστικά τους καθώς και οι ανάγκες που επιχειρούν να ικανοποιήσουν, οι οποίες ταυτίζονται αλλά και ποικίλουν σε μεγάλο βαθμό.

1.3.1 Έξυπνα τηλέφωνα(Smartphones)

Τα έξυπνα τηλέφωνα έκαναν την εμφάνισή τους στα τέλη της δεκαετίας του 2000. Πρόκειται για την εξέλιξη των κλασικών κινητών τηλεφώνων και παρέχουν μεγαλύτερη υπολογιστική ισχύ και συνδεσιμότητα. Στη σημερινή εποχή τα έξυπνα τηλέφωνα προσφέρουν δυνατότητες αναπαραγωγής πολυμέσων, λήψης φωτογραφιών και βίντεο σε υψηλή ανάλυση, πλοήγησης μέσω GPS και γρήγορης πρόσβασης στο διαδίκτυο μέσω τεχνολογιών, Wi-Fi ή 4G.

Τα συμβατικά κινητά έχουν τη δυνατότητα να τρέχουν εφαρμογές βασισμένες σε πλατφόρμες Java Micro Edition και BREW, ενώ τα έξυπνα τηλέφωνα επιτρέπουν στους χρήστες να εγκαταστήσουν εφαρμογές ανάλογα με το λειτουργικό σύστημα της συσκευής. Στη διάρκεια των τελευταίων χρόνων, αναπτύχθηκαν αρκετά λειτουργικά συστήματα. Τα δημοφιλέστερα είναι το BlackBerry OS, το iOS, το Symbian OS, το Windows Phone και το Android. Από τα προαναφερθέντα λειτουργικά συστήματα, δύο από αυτά κατάφεραν να επικρατήσουν, δηλαδή το Android και το iOS, για τα οποία θα γίνει περαιτέρω ανάλυση στο επόμενο κεφάλαιο

Κύριο χαρακτηριστικό των έξυπνων τηλεφώνων είναι ότι δίνουν τη δυνατότητα στο χρήστη να έχει πρόσβαση σε λειτουργίες και προγράμματα που μέχρι να εμφανιστούν οι έξυπνες συσκευές είχαν μόνο οι υπολογιστές. Πλέον διαθέτουν οθόνες αφής για ευκολότερη χρήση, υποστήριξη 3G/4G τεχνολογίας σύνδεσης, ενσωματωμένο Wi-Fi για ασύρματη συνδεσιμότητα με το διαδίκτυο μέσω των κατάλληλων σημείων πρόσβασης, αισθητήρες θέσης, κίνησης και περιβάλλοντος. Διαθέτουν μεγάλο αποθηκευτικό χώρο ώστε ο χρήστης να έχει τη δυνατότητα να κάνει λήψη των εφαρμογών που του χρειάζονται, αλλά και για να διατηρεί τα απαραίτητα έγγραφα, φωτογραφίες και μουσική στη συσκευή του. Ο χρήστης μπορεί να επιλέξει και να εγκαταστήσει ανάμεσα σε ένα μεγάλο εύρος εφαρμογών, το οποίο έχει ως αποτέλεσμα οι χρήστες να μπορούν να εξατομικεύσουν το εύρος των εφαρμογών στις συσκευές τους αυτές, με τέτοιο τρόπο ώστε να ταιριάζουν στον τρόπο ζωής τους και στη δουλειά τους

Οι εφαρμογές διατίθενται από τα ηλεκτρονικά καταστήματα των λειτουργικών συστημάτων αλλά και από τρίτους. Το μεγαλύτερο ποσοστό των εφαρμογών που διατίθενται είναι δωρεάν για λήψη. Στο Google Play Store ένα ποσοστό 94% των συνολικών εφαρμογών είναι προς δωρεάν λήψη σύμφωνα με [έρευνα](#) στο 4ο τρίμηνο του 2017 και υπάρχουν εφαρμογές όπου ο χρήστης θα πρέπει να καταβάλει το αντίστοιχο αντίτιμο σε ποσοστό 6 %. Θα πρέπει να ληφθεί υπόψη ότι για ολοένα και περισσότερες εφαρμογές πλέον, οι προγραμματιστές διαθέτουν δωρεάν την εφαρμογή αλλά για να μπορέσει ο χρήστης να έχει πρόσβαση σε όλες τις λειτουργίες της εφαρμογής θα πρέπει να προβεί σε αγορά αυτών

Σύμφωνα με τις επίσημες ανακοινώσεις των εταιριών, η Apple μέχρι και το Μάιο του 2017 είχε συνολικά περισσότερες από 2.2 εκατομμύρια εφαρμογές στο κατάστημά της, ενώ η Google το Ιανουάριο του 2018 ανακοίνωσε ότι το Δεκέμβρη του 2017 οι εφαρμογές στο Google Play Store υπερβαίνουν το 3.5 εκατομμύρια. Ο παρακάτω πίνακας είναι μέρος έρευνας μέχρι το Μάιο του 2017 όπου παρατηρούμε των αριθμών των εφαρμογών στα εκάστοτε ηλεκτρονικά καταστήματα.

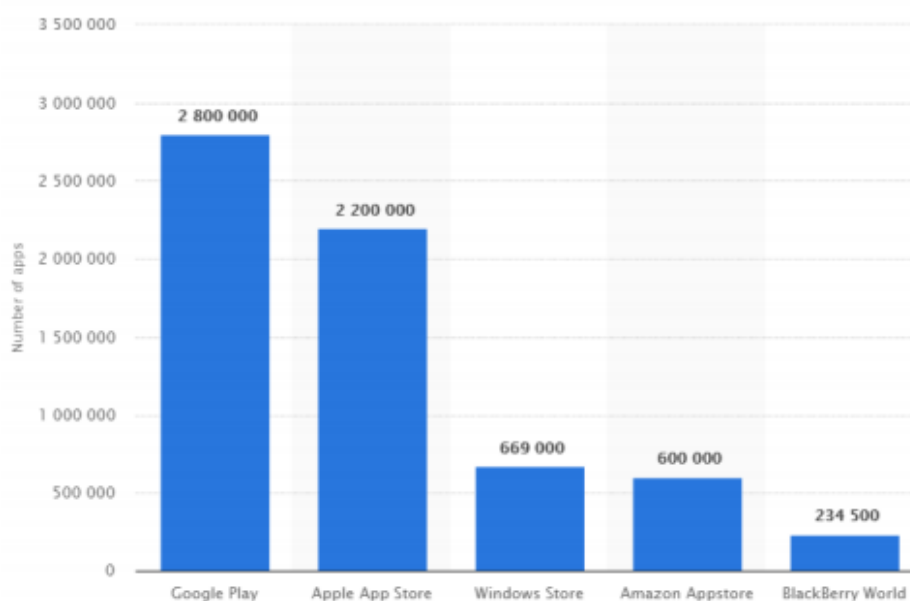


Figure 1: Συνολικός αριθμών εφαρμογών ανά κατάστημα μέχρι το Μάιο του 2017 (<https://www.statista.com>)

Οι χρήστες έξυπνων τηλεφώνων αυξήθηκαν σημαντικά διότι η χρήση εκείνων είναι γρήγορη και εύκολη. Παράλληλα, μέσω αυτών έχουν την δυνατότητα να περιηγηθούν εύκολα στο διαδίκτυο, όταν είναι μακριά από ηλεκτρονικό υπολογιστή. Τα έξυπνα κινητά, ακόμη, διαθέτουν άμεση πρόσβαση στο ηλεκτρονικό ταχυδρομείο και στα κοινωνικά δίκτυα καθώς και μεγάλη ποικιλία στις εφαρμογές. Μπορεί, τέλος να προσαρμοστεί ανάλογα με τις ανάγκες των χρηστών

1.3.2 Υπολογιστής ταμπλέτα (Tablet)

Ο υπολογιστής ταμπλέτα είναι ένα έξυπνο τηλέφωνο με μεγαλύτερη οθόνη αλλά μικρότερη από τους υπολογιστή. Οι διαστάσεις τους συνήθως κυμαίνονται μεταξύ των 7 και 12 ιντσών. Οι δυνατότητες είναι ίδιες με αυτές των έξυπνων κινητών. Κύρια διαφορά είναι ότι υπάρχουν ταμπλέτες που δεν διαθέτουν τεχνολογία 3G/4G. Στα πλεονεκτήματά τους είναι ότι λόγω των μεγαλύτερων διαστάσεων διαθέτουν μεγαλύτερη μπαταρία και έχουν καλύτερη αυτονομία, διαθέτουν περισσότερες θύρες επέκτασης και είναι πιο βολικά σε παράλληλες εργασίες λόγω της μεγαλύτερης οθόνης. Τα λειτουργικά συστήματα στις ταμπλέτες επικεντρώνονται κυρίως σε Android και iOS, αλλά τα τελευταία χρόνια η Microsoft έχει λανσάρει τη σειρά Surface κυρίως για επαγγελματίες. Τα Surface είναι οι πρώτες συσκευές ταμπλέτες που χρησιμοποιούν πλήρες λειτουργικό σύστημα Windows 10 χωρίς ελλείψεις. Επίσης, με τη προσθήκη πληκτρολογίου γίνεται ένας «φορητός υπολογιστής» μικρότερων διαστάσεων

1.3.3 Λειτουργικά Συστήματα για Κινητά

Κατά τη διαδικασία αγοράς ενός έξυπνου τηλεφώνου, σημαντικό ρόλο για την επιλογή του είναι το λειτουργικό σύστημα. Την παρούσα χρονική περίοδο οι χρήστες έχουν περιορισμένες επιλογές λειτουργικών συστημάτων. Τα προηγούμενα χρόνια, κατά την άνθιση των έξυπνων τηλεφώνων ολοένα και περισσότεροι κατασκευαστές διέθεταν το δικό τους λειτουργικό σύστημα, όμως δεν κατάφεραν να επικρατήσουν λόγω των λειτουργικών συστημάτων iOS, Android και Windows Phone. Παρακάτω θα γίνει μια μικρή αναφορά στα λειτουργικά συστήματα που υπάρχουν στην αγορά αυτή την περίοδο.

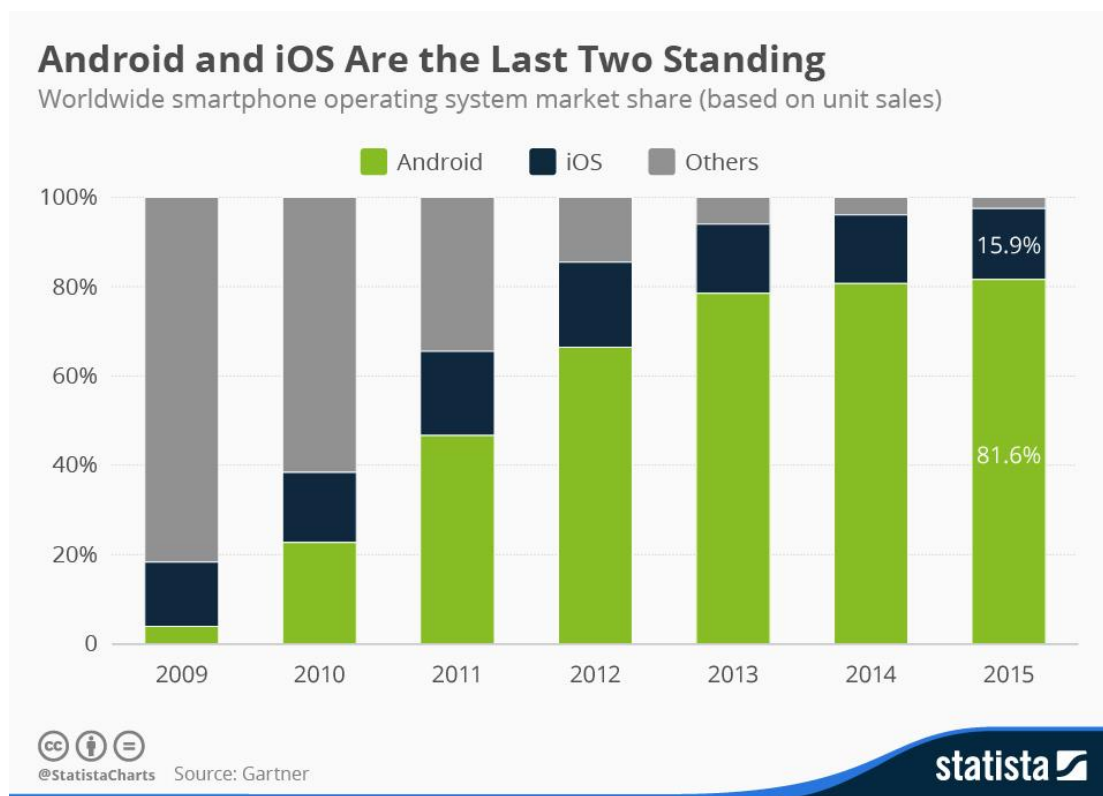


Figure 2: Παγκόσμιο μερίδιο αγοράς Smartphones μέχρι το 2015(<https://www.statista.com>)

- **Android**

Android είναι το λειτουργικό σύστημα της Google και το οποίο θα μας απασχολήσει στη παρούσα μεταπτυχιακή εργασία. Όπως παρατηρούμε στο παραπάνω διάγραμμα, είναι η πιο γρήγορη αναπτυσσόμενη πλατφόρμα και χρησιμοποιείται από πολλούς κατασκευαστές, στην αγορά διατίθενται πολλές διαφορετικές συσκευές σε όλες τις κατηγορίες τιμών, ανάλογα με τις απαιτήσεις του χρήστη.

- **iOs**

Το 2007 η Apple με την παρουσίαση του πρώτου iPhone σηματοδότησε τη νέα γενιά έξυπνων τηλεφώνων. Η μεγαλύτερη διαφορά που έδωσε και την ώθηση στην εταιρεία ήταν η κατάργηση του φυσικού πληκτρολογίου, η ενσωμάτωση οθόνης αφής και πολλών λειτουργιών που έκανε ευχάριστη και φιλική την ενασχόληση με τα κινητά τηλέφωνα στο τελικό χρήστη. Οι συσκευές της Apple τρέχουν στο δικό της λειτουργικό σύστημα, το iOS. Σήμερα, το iOS βρίσκεται στην ενδέκατη έκδοσή του, και έχει τροποποιηθεί κατάλληλα για να τρέχει και στην ταμπλέτα της Apple το iPad. Το λειτουργικό βασίζεται στο macOS, ο προγραμματισμός γίνεται σε Objective-C και Swift. Το λειτουργικό αυτό σύστημα είναι κλειστού κώδικα και οι συσκευές που διατίθενται στην αγορά, κατασκευάζονται αποκλειστικά από την ίδια την Apple.

- **Windows Phone/Mobile**

Στη προσπάθεια της Microsoft να ανταγωνιστεί την Google και την Apple, το 2010 παρουσίασε το λειτουργικό σύστημα Windows Phone. Πρόκειται για την εξέλιξη των Windows Mobile που μέχρι τότε χρησιμοποιούταν από τους υπολογιστές παλάμης. Η γλώσσα προγραμματισμού που υποστηρίζουν είναι οι C#, Visual Basic, και C++. Η Microsoft μέσω των Windows Phone, προσπάθησε να φέρει όσο πιο κοντά γίνεται το γραφικό περιβάλλον και τις λειτουργίες των σταθερών υπολογιστών στα έξυπνα κινητά της, ώστε οι χρήστες της να έχουν μεγαλύτερη οικειότητα με το λειτουργικό της σύστημα. Το 2015 η Microsoft ανακοίνωσε τη τελευταία έκδοση του λειτουργικού και τα μετονόμασε στα σημερινά Windows Mobile 10. Στη διάρκεια αυτών των χρόνων δεν είχε μεγάλη ανταπόκριση από τους χρήστες, λόγω του μικρού αριθμού εφαρμογών και των μεγάλων ελλείψεων σε σχέση με τον ανταγωνισμό. Αυτό δημιουργήθηκε κυρίως λόγω της έλλειψης κινήτρου και των κατάλληλων συνθηκών στους προγραμματιστές στις ώστε να επενδύσουν στο λειτουργικό της Microsoft. Τέλος, τον Οκτώβριο του 2017 ανακοίνωσε επίσημα ότι θα σταματήσει την περαιτέρω εξέλιξη των Windows Mobile 10 και θα συνεχίζουν να παρέχουν απλά ενημερώσεις ασφαλείας στους χρήστες.

1.3.4 Android και iOS

Η Google με τη συνεχή εξέλιξη του Android τα τελευταία χρόνια κατάφερε να φτάσει αλλά και να ξεπεράσει σε πωλήσεις την Apple. Η ραγδαία εξάπλωση του Android προέκυψε επειδή η Google παρέχει στους κατασκευαστές το λειτουργικό της σύστημα δωρεάν. Αυτό είχε ως αποτέλεσμα στην αγορά να βγαίνουν ολοένα και περισσότερες συσκευές, με τις τιμές να κυμαίνονται από τα 50€ έως τα 1000€, με τον τελικό καταναλωτή να έχει πληθώρα επιλογών για την αγορά νέου κινητού. Επίσης χάρις την τιμολογιακή πολιτική της Apple αυτό έδωσε μεγαλύτερη ώθηση στην αγορά των κινητών Android, διότι με λιγότερα χρήματα είχαν εφάμιλλες επιδόσεις και χαρακτηριστικά. Στο παρόν υποκεφάλαιο θα αναφερθούμε στα θετικά και αρνητικά της κάθε πλατφόρμας.



Figure 3: Λειτουργικά Συστήματα Android και iOS

Από την πλευρά των προγραμματιστών, η ανάπτυξη εφαρμογών σε περιβάλλον Android είναι ευκολότερη, διότι ο υποψήφιος προγραμματιστής Android θα πρέπει να διαθέτει έναν προσωπικό υπολογιστή με οποιοδήποτε λειτουργικό σύστημα, συσκευές Android για δοκιμές που είναι αισθητά οικονομικότερες από τις iOS συσκευές και να διαθέτει ένα λογαριασμό Google στο Play Store, για τον οποίο απαιτείται η εφάπαξ καταβολή ενός τιμήματος της τάξης των 25\$ για να μπορέσει να προσθέσει μια εφαρμογή στο ηλεκτρονικό κατάστημα της Google. Τέλος, η ανάπτυξη γίνεται σε Java, που παραμένει η δημοφιλέστερη γλώσσα προγραμματισμού και διαθέτει μεγάλη κοινότητα υποστήριξης.

Από τη πλευρά της Apple, η ανάπτυξη γίνεται σε Objective C και Swift, το πρόγραμμα ανάπτυξης της Apple το XCode τρέχει αποκλειστικά σε λειτουργικό σύστημα MacOS, για το οποίο απαιτείται αγορά MAC υπολογιστή που κοστίζει τουλάχιστον 1000€, συσκευές iOS για τις δοκιμές και ετήσια συνδρομή 100\$ για να μπορεί να ανεβάσει ο προγραμματιστής τις εφαρμογές του στο App Store.

Αισθητή διαφορά μεταξύ των δύο λειτουργικών συστημάτων αφορά το αγοραστικό κοινό. Οι χρήστες iOS είναι περισσότερο διατεθειμένοι να πληρώσουν για να αγοράσουν μια εφαρμογή από ότι οι χρήστες Android. Αυτό οφείλεται στο γεγονός πως το iPhone καλώς ή κακώς θεωρείται στις μέρες μας ως status symbol, καθώς πολλοί από τους κατόχους του είναι σταθεροί οπαδοί του. Θεωρείται στις μέρες μας ότι κάποιος γίνεται περισσότερο αποδεκτός

έχοντας στη κατοχή του κάποιος συσκευή iPhone σε σχέση με Android συσκευή. Τέτοιοι χρήστες δεν είναι επιφυλακτικοί στο να πληρώσουν για να κατεβάσουν μια εφαρμογή και αυτή η συμπεριφορά λειτουργεί προς όφελος των προγραμματιστών

Αντίθετα, το Android βασίζεται στο Linux και έχει ένα καθαρά open source χαρακτήρα. Πολλοί από τους χρήστες του Android ασπάζονται αυτή τη φιλοσοφία και δεν είναι διατεθειμένοι να πληρώσουν για μία εφαρμογή. Θα αγοράζαν μια εφαρμογή μόνο αν είναι εγγυημένα χρήσιμη και προσεγμένη.

Αρνητικά του Android

Στο Android υπάρχει πληθώρα συσκευών με διαφορετικά χαρακτηριστικά όπως οθόνες διαφορετικών διαστάσεων και αναλύσεων, και διαφορετικά σείτ αισθητήρων, τα οποία οι προγραμματιστές καλούνται να λαμβάνουν υπ' όψιν κατά την ανάπτυξη των εφαρμογών τους ώστε αυτές να τρέχουν απροβλημάτιστα και με ομοιόμορφο τρόπο σε όλες τις συσκευές. Αυτό προϋποθέτει επιπλέον δουλειά και πρόνοια από την πλευρά του προγραμματιστή.

Το σημαντικότερο αρνητικό του Android σε σχέση με το iOS είναι το πρόβλημα του κατακερματισμού (Fragmentation). Ο κατακερματισμός στα mobile λειτουργικά συστήματα αφορά τις περιπτώσεις όπου οι χρήστες τρέχουν στις συσκευές τους προηγούμενες εκδόσεις του λειτουργικού συστήματος, ενώ άλλοι τρέχουν νεότερες εκδόσεις. Το χάσμα αυτό δημιουργείται διότι οι κατασκευαστές των συσκευών παύουν να ενημερώνουν τις συσκευές με νεότερες εκδόσεις λόγω περιορισμών στα χαρακτηριστικά των συσκευών ή λόγω της παλαιότητάς τους. Επιπλέον, με την εξέλιξη του λειτουργικού, ορισμένες δυνατότητες δεν μπορεί να εφαρμοστούν σε παλιότερες εκδόσεις με αποτέλεσμα οι προγραμματιστές να μην υποστηρίζουν συσκευές με παλιότερες εκδόσεις. Ενώ στο iOS, η Apple υποστηρίζει για αρκετά χρόνια τις συσκευές της και προβαίνει σε αναβαθμίσεις. Σε αυτό παίζει σημαντικό παράγοντα ο μικρός αριθμός συσκευών που διαθέτει η Apple στην αγορά, συνήθως 2-3 συσκευές το χρόνο.



Figure 4: Εκδόσεις Λογισμικού iOS και Android

Όπως βλέπουμε στο παραπάνω γράφημα το iOS 11 που κυκλοφόρησε το Σεπτέμβριο του 2017 είναι εγκατεστημένο ήδη στο 65 % των συσκευών που κυκλοφορούν. Από την άλλη πλευρά, η όγδοη και τελευταία έκδοση του Android κυκλοφόρησε τον Αύγουστο το 2017 και μόλις το 0,7 % των συσκευών «τρέχουν» τη νεότερη έκδοση του λειτουργικού συστήματος

Η Google μέσω του Android 8.0 έχει προσθέσει το “Project Treble” μέσω του οποίου βοηθάει του κατασκευαστές και τους παρόχους να διαθέτουν τις νεότερες εκδόσεις γρήγορα. Αυτό όμως προϋποθέτει η συσκευή να διαθέτει ήδη Android 8.0 ή να αναβαθμιστεί μελλοντικά.

Αρνητικά του iOS

Το iOS από τη πλευρά του λόγω της κλειστού λογισμικού έχει αρκετούς περιορισμούς που δεν υπάρχουν στο Android. Αρχικά ο κυριότερος περιορισμός βρίσκεται στη συνδεσιμότητα των συσκευών. Για παράδειγμα, ένας χρήστης Android μπορεί να αποστείλει ένα αρχείο μέσω τεχνολογίας Bluetooth σε ένα άλλο χρήστη, στο iOS είναι κλειδωμένη αυτή η δυνατότητα και αναγκαστικά η αποστολή θα γίνει μέσω cloud εφαρμογών. Επιπλέον, το Android υποστηρίζει τη σύνδεση σκληρών δίσκων για τη μεταφορά δεδομένων χωρίς να είναι απαραίτητη η χρήση υπολογιστή. Ο χρήστης (iOS) για τη μεταφορά αρχείων, φωτογραφιών και οποιοδήποτε άλλου αρχείου απαιτείται να χρησιμοποιεί το Apple iTunes.

Ταυτόχρονα, απουσιάζει η παραμετροποίηση του γραφικού περιβάλλοντος από τις iOS συσκευές. Ο χρήστης μπορεί απλά να αλλάξει τη σειρά των εφαρμογών και την ταπετσαρία, ενώ στο Android έχει τη δυνατότητα να τροποποιήσει τη συσκευή του όπως ακριβώς επιθυμεί είτε στο υπάρχον γραφικό περιβάλλον είτε μέσω θεμάτων που θα βρει στο Play Store.

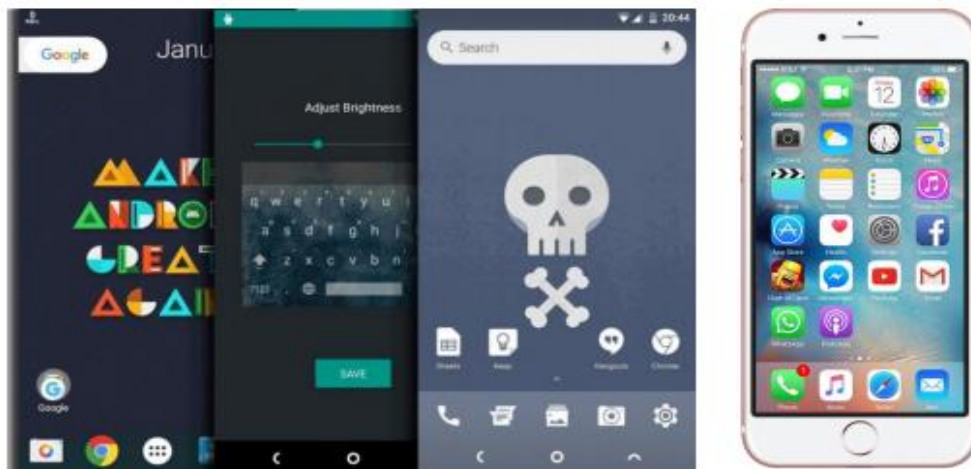


Figure 5: Γραφικό Περιβάλλον Android και iOS

1.4 Εφαρμογές Έξυπνων τηλεφώνων

Τα έξυπνα τηλέφωνα έχουν εισέλθει δυναμικά στην αγορά παρουσιάζοντας μια σταθερά ανοδική πορεία πωλήσεων με εξαιρετικές προοπτικές περαιτέρω ανάπτυξης. Αναπόσπαστο κομμάτι των τηλεφώνων αποτελούν οι εφαρμογές κινητών (mobile applications) οι οποίες μπορούν να καλύψουν μία πληθώρα αναγκών.

Πολλές εταιρίες και πολλοί προγραμματιστές έχουν ήδη αναπτύξει τις δικές τους εφαρμογές για κινητές συσκευές, προσφέροντας υπηρεσίες προστιθέμενης αξίας στους καταναλωτές, καθώς μια κινητή εφαρμογή μπορεί να εξυπηρετήσει προσωπικές, επαγγελματικές και ενδο-επιχειρηματικές ανάγκες.

1.4.1 Ο κατακερματισμός και τα προβλήματα των προγραμματιστών

Στο προηγούμενο κεφάλαιο αναφερθήκαμε στις διαφορές μεταξύ του Android και του iOS, εντοπίσαμε κάποια σοβαρά προβλήματα που αντιμετωπίζουν οι προγραμματιστές και οι χρήστες. Οι κατηγορίες των προβλημάτων είναι η αμοιβή, το κόστος υλοποίησης και το πιο βασικό είναι η δυσκολία των προγραμματιστών να προσαρμόσουν και να υλοποιήσουν μια εφαρμογή σε πολλές διαφορετικές συσκευές και πλατφόρμες.

Είτε πρόκειται για φορητούς υπολογιστές, ταμπλέτες ή έξυπνα τηλέφωνα, ο αριθμός των νέων φορητών συσκευών που εισέρχονται στην αγορά αυξάνεται με ταχείς ρυθμούς. Οι τομείς των αναγκών της καθημερινότητας των καταναλωτών που αυξάνονται συνεχώς δυσκολεύει περαιτέρω το έργο των προγραμματιστών που θα πρέπει να ξοδέψουν ακόμα περισσότερο χρόνο τελειοποιώντας τις εφαρμογές τους για όσο το δυνατόν μεγαλύτερο μέρος των συσκευών.

Έτσι, κατά την ανάπτυξη μίας εφαρμογής για κινητές συσκευές, προκύπτουν εμπόδια και δυσκολίες λόγω του κατακερματισμού στο υλικό (hardware) και στο λογισμικό (software) των κινητών συσκευών. Οι κινητές συσκευές υποστηρίζονται από λειτουργικά συστήματα τα οποία διακρίνονται από συγκεκριμένα γνωρίσματα. Για να αναπτυχθεί μια εφαρμογή για ένα λειτουργικό σύστημα χρησιμοποιείται συγκεκριμένη γλώσσα προγραμματισμού, στην οποία επένδυσε ο κατασκευαστής, και συγκεκριμένα εργαλεία ανάπτυξης που παρέχονται από τον κατασκευαστή. Το πρόβλημα του κατακερματισμού όμως δεν σταματάει εκεί αφού ακόμη και οι διαφορετικές εκδόσεις του ίδιου λειτουργικού συστήματος ορισμένες φορές παρουσιάζουν μεγάλες διαφορές μεταξύ τους. Ακόμα και συσκευές που μπορεί να βασίζονται στο ίδιο λειτουργικό σύστημα είναι πιθανό να παρουσιάσουν διαφορές μεταξύ τους λόγω των διαφορετικών προδιαγραφών του υλικού (hardware) της συσκευής. Για παράδειγμα το πώς υποστηρίζεται η ανίχνευση αφής, διαφέρει στην εκάστοτε συσκευή λόγω του διαφορετικού αισθητήρα αφής.

Τέλος, τα τελευταία χρόνια έχει δημιουργηθεί η ανάγκη για δημιουργία πλατφόρμας όπου ο προγραμματιστής θα γράφει σε μια συγκεκριμένη γλώσσα προγραμματισμού και θα έχει τη δυνατότητα να εξάγει κώδικα όπου θα τρέχει εφάμιλλα σε πολλά διαφορετικά λειτουργικά συστήματα. Ο παρακάτω πίνακας περιέχει τις διαφορές που παρουσιάζονται στα επικρατέστερα λειτουργικά συστήματα σε χαρακτηριστικά που αφορά την ανάπτυξη μιας εφαρμογής, όπως στις γλώσσες προγραμματισμού, τον τύπο των εκτελέσιμων αρχείων, τα περιβάλλοντα ανάπτυξης, και τα διαδικτυακά καταστήματα για την αγορά εφαρμογών.

| | Android | iOS | Windows Phone |
|--------------------------------|--------------------------------------|---------------------|-----------------------|
| Έκδοση λειτουργικού συστήματος | Android 2.3- 8.1 | iOS 8 -11 | Windows Mobile 10 |
| Περιβάλλον ανάπτυξης | Android Studio | XCode | Visual Studio |
| Γλώσσες προγραμματισμού | Java & Kotlin | Swift & Objective C | C# & Visual Basic |
| Ηλεκτρονικά Καταστήματα | Google Play Store & 3rd Party Stores | Apple App Store | Microsoft Marketplace |

Figure 6: Χαρακτηριστικά μιας εφαρμογής στο κάθε λειτουργικό σύστημα (<https://www.wikipedia.org/>)

1.4.2 Τύποι εφαρμογών κινητών συσκευών

Υπάρχουν τρία είδη εφαρμογών για κινητές συσκευές, οι εγγενείς εφαρμογές (Native applications), οι διαδικτυακές εφαρμογές (Web applications) και οι υβριδικές εφαρμογές (Hybrid applications). Η επιλογή της κατάλληλης μεθόδου εξαρτάται από το επιχειρηματικούς στόχους που έχει θεσπίσει ο πελάτης και θεωρείται σημαντική επιλογή διότι θα κρίνει σε ένα μέρος την επιτυχία της εφαρμογής. Κάθε ένας από τους τρόπους έχει τα δικά του μειονεκτήματα και πλεονεκτήματα στα οποία θα αναφερθούμε.

Εγγενείς Εφαρμογές (Native App):

Οι εγγενείς εφαρμογές είναι ο πιο συνηθισμένος τύπος εφαρμογής. Αναπτύσσονται για συγκεκριμένες πλατφόρμες και είναι γραμμένες σε γλώσσες προγραμματισμού που υποστηρίζει η πλατφόρμα. Τόσο η Apple όσο και η Google παρέχουν στους προγραμματιστές τα δικά τους εργαλεία ανάπτυξης, τα στοιχεία διεπαφής και το SDK. Οι περισσότερες εταιρείες επενδύουν σε ανάπτυξη εγγενών εφαρμογών για κινητά λόγω των πλεονεκτημάτων που προσφέρονται σε σύγκριση με άλλους τύπους εφαρμογών.

Πλεονεκτήματα:

- Οι εγγενείς εφαρμογές είναι πολύ γρήγορες και έχουν πολύ καλή απόκριση επειδή είναι κατασκευασμένες για τη συγκεκριμένη πλατφόρμα.
- Έχουν την καλύτερη απόδοση.
- Διανέμονται σε ηλεκτρονικά καταστήματα εφαρμογών.
- Είναι πιο διαδραστικές και τρέχουν πολύ πιο ομαλά .
- Ο εγγενής προγραμματισμός επιτρέπει στους προγραμματιστές να έχουν πρόσβαση στο πλήρες σύνολο των λειτουργιών της πλατφόρμας. Για παράδειγμα πρόσβαση στους αισθητήρες της συσκευής, στην κάμερα και σε πολλά άλλα.
- Δημιουργία περίπλοκων διεπαφών με τα διαθέσιμα γραφικά συστατικά που εμπεριέχονται στο λειτουργικό σύστημα.
- Αξιοποίηση διεπαφών προγραμματισμού εφαρμογών (API) και προσθήκη βιβλιοθηκών από τη κοινότητα της εκάστοτε πλατφόρμας
- Δεν απαιτείται συνεχής σύνδεση στο διαδίκτυο, αν και αυτό εξαρτάται από τη λειτουργικότητα.
- Συνολικά καλύτερη εμπειρία χρήσης για τον χρήστη, η ροή είναι πιο φυσική καθώς έχουν συγκεκριμένα πρότυπα διεπαφών για κάθε πλατφόρμα

Μειονεκτήματα:

- Δυσκολότερες γλώσσες προγραμματισμού για εκμάθηση που σημαίνει ότι χρειάζονται πιο έμπειροι προγραμματιστές.
- Πιο ακριβείς υλοποιήσεις.
- Ο πηγαίος κώδικας δεν μπορεί να χρησιμοποιηθεί σε άλλη πλατφόρμα.
- Για την ανάπτυξη μιας εφαρμογής σε διαφορετικά λειτουργικά συστήματα χρειάζονται ξεχωριστές ομάδες υλοποίησης το οποίο αυξάνει το κόστος υλοποίησης.
- Σε περίπτωση όπου γίνει ενημέρωση στην εφαρμογή, πρέπει να ανέβει στο κατάστημα και αφού εγκριθεί, ο χρήστης μπορεί να κάνει λήψη, ενώ σε εφαρμογές ιστού οι ενημερώσεις γίνονται άμεσα.

Εφαρμογές Ιστού (Web App):

Οι εφαρμογές ιστού για έξυπνα τηλέφωνα είναι διαδικτυακές εφαρμογές ειδικά σχεδιασμένες για χρήση σε κινητά τηλέφωνα ή ταμπλέτες και ανοίγουν μέσω του περιηγητή της συσκευής με την επίσκεψή στην αντίστοιχη ηλεκτρονική διεύθυνση (URL) της εφαρμογής. Ωστόσο, οι εφαρμογές ιστού δεν χρειάζονται λήψη από καταστήματα εφαρμογών, όπως οι εγγενείς εφαρμογές. Οι εφαρμογές ιστού φορτώνονται σε προγράμματα περιήγησης όπως ο Chrome, ο Safari ή ο Firefox. Επίσης, μια εφαρμογή ιστού δεν αποθηκεύεται στην συσκευή του χρήστη και δεν δεσμεύει χώρο αποθήκευσης. Οι εφαρμογές ιστού δίνουν λύση στον κατακερματισμό, διότι δημιουργούνται για να προσαρμόζονται σε όλα τα είδη οθονών χάρις σε κάποια πλαίσια (framework) όπως είναι το Bootstrap και τρέχουν σε διαφορετικά λειτουργικά συστήματα, ανεξάρτητα από τα τεχνικά χαρακτηριστικά της εκάστοτε συσκευής.

Με τη συνεχόμενη ανάπτυξη νέων τεχνολογιών στον ιστό, δημιουργούνται κενά μεταξύ των εγγενών εφαρμογών, του ιστού και των υβριδικών. Σε πολλές περιπτώσεις οι χρήστες μπορούν να κατεβάσουν μια εφαρμογή από το ηλεκτρονικό κατάστημα της πλατφόρμας που χρησιμοποιούν και τελικά η εφαρμογή λειτουργεί μέσω ενός WebView, το οποίο περιέχει το σύνδεσμο της εφαρμογής του ιστού. Οι εφαρμογές ιστού δημιουργούνται μέσω JavaScript\CSS και HTML5. Το WebView είναι ένα γραφικό στοιχείο μέσω του οποίου εμφανίζουμε ιστοσελίδες.

Επίσης, έχουν δημιουργηθεί εφαρμογές ιστού και εφαρμογές κινητών που ονομάζονται Progressive Web Apps. Αυτές είναι μια υβριδική λύση διότι οι εφαρμογές ιστού δεν έχουν πρόσβαση σε λειτουργίες του λειτουργικού όπως αποστολή ειδοποιήσεων ή λειτουργία χωρίς διαδίκτυο. Μέσω των Progressive Web εφαρμογών μπορούν να αξιοποιούν λειτουργίες παρόμοιες με τις εγγενείς εφαρμογές όπως :

- Αποστολή μηνυμάτων.
- Χρήση gestures και του επιταχυνσιόμετρου της συσκευής.
- Αξιοποίηση κάποιων υλικών της συσκευής όπως της δόνησης.

Παρά τις συνεχείς βελτιώσεις, ένα σημαντικό μειονέκτημα είναι ότι μπορούν να τρέχουν μόνο σε περιηγητή Google Chrome. Επιπλέον, οι εφαρμογές ιστού υστερούν στο κομμάτι της ασφάλειας, διότι οι εγγενείς εφαρμογές πριν διατεθούν στο κοινό, εξετάζεται ο κώδικας από τα ηλεκτρονικά καταστήματα για τυχόν κακόβουλο λογισμικό και για λειτουργίες που μπορεί να εκθέσουν τα προσωπικά δεδομένα των χρηστών.

Οι εφαρμογές ιστών έχουν ορισμένα πλεονεκτήματα κυρίως της εύκολης κατασκευής, μικρότερο κόστος κατασκευής και συντήρησης σε σχέση με τις εγγενείς εφαρμογές. Επίσης δημιουργείται μια εφαρμογή για πολλαπλά λειτουργικά συστήματα, αρκεί η συσκευή να διαθέτει πρόγραμμα περιήγησης.

Από την άλλη, οι εφαρμογές ιστού διαθέτουν μια σειρά μειονεκτημάτων:

1. Απαιτείται η συσκευή να διαθέτει πρόγραμμα περιήγησης εγκατεστημένο για να εκτελεστεί.
2. Οι χρήστες θα πρέπει να πληκτρολογήσουν τη διεύθυνση(URL) της εφαρμογής, πράγμα που ισοδυναμεί με κακή εμπειρία χρήσης.
3. Οι εφαρμογές ιστού είναι πολύ πιο αργές, λιγότερο διαδραστικές και διαισθητικές συγκριτικά με τις εγγενείς εφαρμογές.

4. Δεν υπάρχει εικονίδιο της εφαρμογής στην επιφάνεια εργασίας του κινητού όπως θα γινόταν αν είχε γίνει λήψη από τα καταστήματα εφαρμογών.
5. Παρατηρείται η μη αξιοποίηση βοηθητικού υλικού (hardware) των συσκευών και μικρότερη ασφάλεια.

Υβριδικές Εφαρμογές (Native App):

Οι υβριδικές εφαρμογές λειτουργούν σε διάφορες πλατφόρμες και συμπεριφέρονται σαν τις εγγενείς εφαρμογές. Πρόκειται για ένα συνδυασμό των εγγενών και των εφαρμογών ιστού. Οι χρήστες μπορούν να τις εγκαταστήσουν στη συσκευή τους όπως σε μια εγγενή εφαρμογή, αλλά στη πραγματικότητα πρόκειται για μια εφαρμογή ιστού.

Οι συγκεκριμένου τύπου εφαρμογών βασίζονται σε JavaScript, HTML και CSS και τρέχουν σε ένα WebView. Μια υβριδική εφαρμογή αποτελείται από δύο μέρη. Το πρώτο κομμάτι είναι το “back-end” που δημιουργείται χρησιμοποιώντας γλώσσες όπως οι HTML, CSS και JavaScript. Το δεύτερο κομμάτι είναι ένα εγγενές κέλυφος το οποίο γίνεται λήψη και φορτώνει τον κώδικα χρησιμοποιώντας το WebView. Στις υβριδικές εφαρμογές θα προκύψουν προβλήματα όπως και στις εγγενείς, όμως τα σφάλματα είναι δυσκολότερο να αντιμετωπιστούν.

Πλεονεκτήματα:

- Βασίζεται σε τεχνολογίες ιστού (HTML / CSS / JavaScript), που κάνει τη ανάπτυξη ευκολότερη.
- Φθηνότερη υλοποίηση σε σχέση με μια εγγενή εφαρμογή.
- Δημιουργείται μια εφαρμογή για όλες τις πλατφόρμες χρησιμοποιώντας τεχνολογίες όπως η Cordova.
- Δεν απαιτείται πρόγραμμα περιήγησης σε αντίθεση με μια εφαρμογή ιστού.
- Έχουν πρόσβαση στα εσωτερικά API της συσκευής, για παράδειγμα στο χώρο αποθήκευσης, στην κάμερα κ.λπ.
- Ταχύτερη ανάπτυξη από τις εγγενείς εφαρμογές, επειδή υπάρχει μια ενιαία βάση κώδικα.

Μειονεκτήματα:

- Πιο αργές από τις εγγενείς εφαρμογές.
- Πιο ακριβές από τις τυπικές εφαρμογές ιστού, επειδή η παραμετροποίηση κοστίζει περισσότερο σε σχέση με τις εγγενείς εφαρμογές.
- Λιγότερο διαδραστικές από τις εγγενείς εφαρμογές
- Μειωμένη δυνατότητα παραμετροποίησης σε σχέση με τις εγγενείς εφαρμογές.

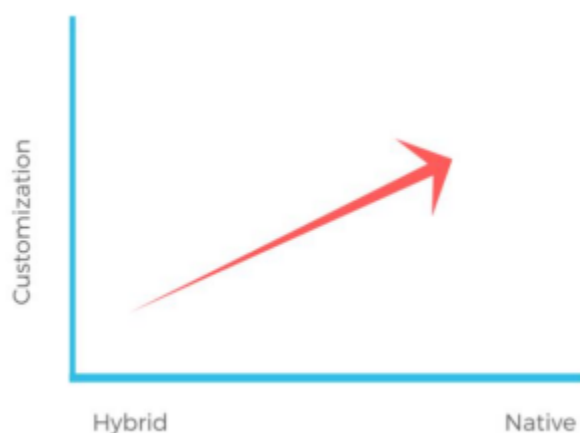


Figure 7: Το κόστος παραμετροποίησης ανάμεσα σε εγγενείς και υβριδικές (<https://www.statista.com>)

1.4.3 Ανάπτυξη εφαρμογών σε διαφορετικά λειτουργικά συστήματα

Προηγουμένως αναφερθήκαμε στα τρία είδη ανάπτυξης εφαρμογών. Στο παρόν υποκεφάλαιο θα αναφερθούμε στις εγγενείς εφαρμογές που μπορούν να εκτελεστούν σε διαφορετικά λειτουργικά συστήματα. Αυτό έρχεται να καλύψει ένα από τα σημαντικότερα μειονεκτήματα των εγγενών εφαρμογών που χρειάζονται ξεχωριστή ανάπτυξη εφαρμογών ανάλογα με τη πλατφόρμα.

Τα τελευταία χρόνια έχουν δημιουργηθεί αρκετές εναλλακτικές λύσεις ώστε οι χρήστες να αποκοιμίσουν οφέλη από την παράλληλη ανάπτυξη χωρίς να θυσιάζεται ή εμπειρία τους ή η πρόσβαση στα εγγενή API. Δύο τέτοιες πλατφόρμες είναι η Xamarin και η React Native.

Η Xamarin δημιουργήθηκε από τη Microsoft και η ανάπτυξη γίνεται σε C#. Η Microsoft παρέχει τα κατάλληλα εργαλεία για την ανάπτυξη, αποσφαλμάτωση και τη διανομή των εφαρμογών σε παράλληλα λειτουργικά συστήματα. Ορισμένες εταιρείες έχουν χτίσει τις εφαρμογές με Xamarin, για παράδειγμα είναι το Slack, Pinterest, Insightly και πολλές άλλες.

Ταυτόχρονα η React Native αναπτύχθηκε από το Facebook. Πρόκειται για μια πλατφόρμα ανοιχτού κώδικα που επιτρέπει τη δημιουργία εγγενών εφαρμογών για iOS και Android με κοινό πηγαίο κώδικα. Η React συνδυάζεται με JavaScript για να επιτευχθεί το τελικό αποτέλεσμα. Σε περίπτωση όπου η React δεν έχει πρόσβαση σε λειτουργίες της συσκευής, μπορεί επίσης να γραφτεί παράλληλα και εγγενής κώδικας εάν χρειαστεί. Αυτή τη στιγμή υπάρχουν μεγάλοι οργανισμοί-εταιρείες που αξιοποιούν αυτή τη πλατφόρμα όπως το Facebook, το Walmart και η Airbnb.

1.5 Ανασκόπηση πεδίου

1.5.1 WiseApp – Recommended Apps



Figure 8: WiseApp – Recommended Apps

Η εφαρμογή WiseApp αποθηκεύει μια συλλογή 250+ προσεκτικά επιλεγμένων εφαρμογών από 2 εκατομμύρια εφαρμογές στην αγορά. Οι κατασκευαστές της παρακολουθούν τις εφαρμογές που κυκλοφορούν στην αγορά και ενημερώνουν συνεχώς την εφαρμογή για να συμβαδίζουν με τις μεταβαλλόμενες ανάγκες των χρηστών.

Η εφαρμογή στηρίζεται στην κατηγοριοποίηση των εφαρμογών, στο rating της κάθε εφαρμογής (στοιχεία που αντλούνται από το play store) και στην προσωπική εκτίμηση των ιδιοκτητών της για να προτείνει πιο στοχευμένα εφαρμογές. Ο χρήστης με ένα απλό ακούμπημα πάνω στην εφαρμογή που τον ενδιαφέρει προωθείται στο play store απ' όπου μπορεί να την κατεβάσει και να την εγκαταστήσει.

1.5.2 itcher - Movies & TV Shows Recommendations

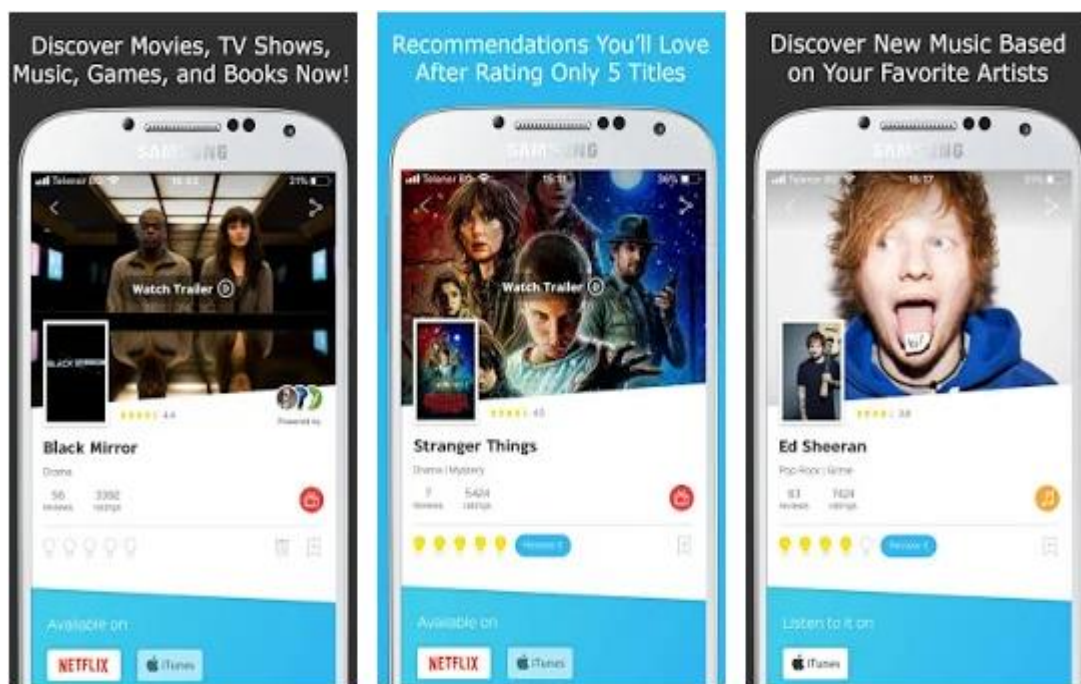


Figure 9: Itcher – Movie & TV show Recommendations

Το itcher είναι μια εφαρμογή συστάσεων ψυχαγωγίας που παράγει τα αποτελέσματά της με βάση τις ατομικές προτιμήσεις των χρηστών, χρησιμοποιώντας τη δύναμη της κοινότητάς της. Προτείνουν ταινίες, τηλεοπτικές εκπομπές, βιβλία, μουσική και παιχνίδια.

Διαθέτει μια βάση δεδομένων με πάνω από 4 εκατομμύρια τίτλους και παρέχει τις πρώτες συστάσεις τις που υποστηρίζονται από την κοινότητα σε λιγότερο από ένα λεπτό από τη στιγμή που την κατεβάζει ο χρήστης.

Στηρίζεται και αυτή στην κατηγοριοποίηση των προτάσεων της την οποία την αντλεί από πηγές όπως το [imdb](#) και το [play store](#). Διαφοροποιείται από τη λογική της «wiseApp» με το να βασίζεται κυρίως στο rating που παίρνει από την δική της την κοινότητα για να προτείνει τίτλους στους χρήστες της. Η μεγαλύτερη όμως διαφορά της, είναι ότι ανάλογα με τα ratings που δίνει ο χρήστης σε κάθε τίτλο συγκεκριμένης κατηγορίας αλλάζουν και οι προτάσεις τις, πιο συγκεκριμένα προτείνει τίτλους που άρεσαν σε άλλους χρήστες της κοινότητας της οι οποίοι μέσα από τα ratings που έχουν δώσει φαίνεται να έχουν παρόμοια ενδιαφέροντα με τον χρήστη.

1.5.3 Taste - Movie Recommendations

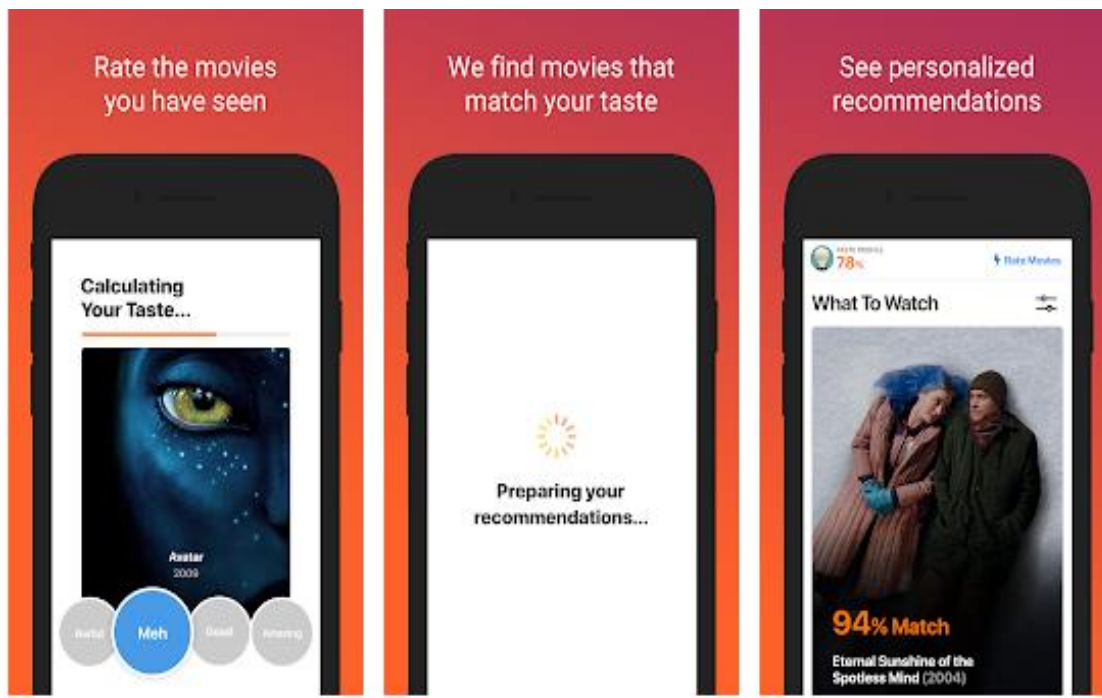


Figure 10: Taste – Movie recommendations

Ακριβώς στην ίδια λογική με την εφαρμογή «itcher» κινείται και η «taste» αλλά προτείνει μόνο ταινίες και τηλεοπτικές σειρές. Σε αντίθεση με το «[imdb](#)», το «[Rotten Tomatoes](#)» και το «[Fandango](#)», η εφαρμογή «taste» εξατομικεύει αξιολογήσεις και κριτικές με βάση τις προτιμήσεις του χρήστη.

Μια μικρή πινελιά διαφοροποίησης της είναι ότι η κατηγοριοποίηση των αποτελεσμάτων της μπορεί να γίνει βάση του ονόματος ενός ηθοποιού αλλά και από το κανάλι (ή ιστοσελίδα) στο οποίο παίζεται (πχ [Netflix](#), [Hulu](#), [HBO](#)).

2. Εγχειρίδιο χρήσης

1.6 Δημιουργία Λογαριασμού και είσοδος

Για να ξεκινήσει ο χρήστης την περιήγηση του στην εφαρμογή αρχικά πρέπει να έχει δημιουργήσει ένα λογαριασμό. Για να δημιουργήσει το λογαριασμό πρέπει, όπως φαίνεται στην εικόνα 4, να πατήσει στο κουμπί «Not a member? Get registered now! »

Στην εικόνα 4 βλέπουμε την δραστηριότητα δημιουργίας νέου λογαριασμού. Ο χρήστης πρέπει να εισάγει το ονοματεπώνυμο του, το email του και τον κωδικό με τον οποίο θέλει να εισέρχεται και μετά να πατήσει το κουμπί «register» Μετά την επιτυχή δημιουργία του λογαριασμού του αυτόματα επιστρέφει στην δραστηριότητα εισόδου (εικόνα 4). (Προσοχή, η διαδικασία εγγραφής μπορεί να διαρκέσει και ένα ολόκληρο λεπτό).

Το κουμπί «Forgot your password? » μεταφέρει τον χρήστη στην δραστηριότητα ανάκτησης κωδικού (εικόνα 6) και το «already registered. Login! » επιστρέφει τον χρήστη στην δραστηριότητα εισόδου (εικόνα 4).

Αφού ο χρήστης εγγραφεί στην βάση δεδομένων μας μπορεί να εισάγει στα ανάλογα πεδία της εικόνας 4 το email του και τον κωδικό του και να πατήσει το κουμπί «LOGIN» για να ξεκινήσει να χρησιμοποιεί την εφαρμογή.

Το κουμπί «Forgot your password» μεταφέρει τον χρήστη στην δραστηριότητα ανάκτησης κωδικού.

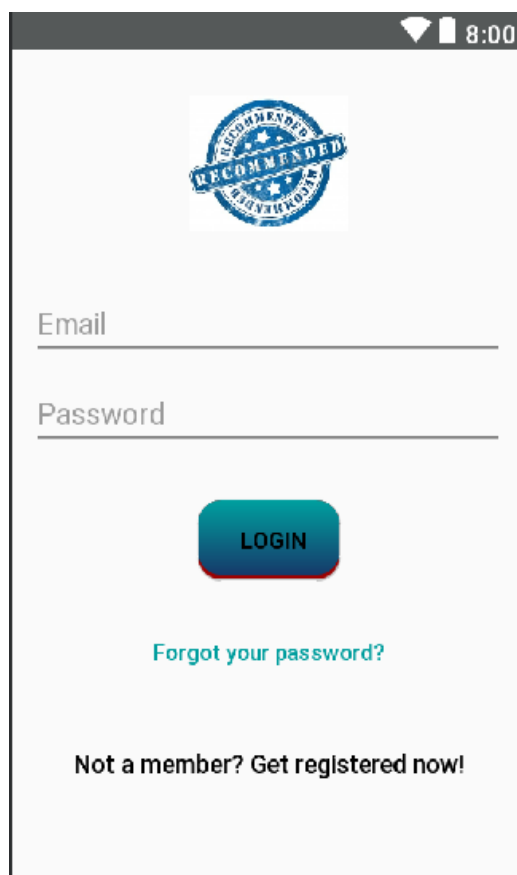
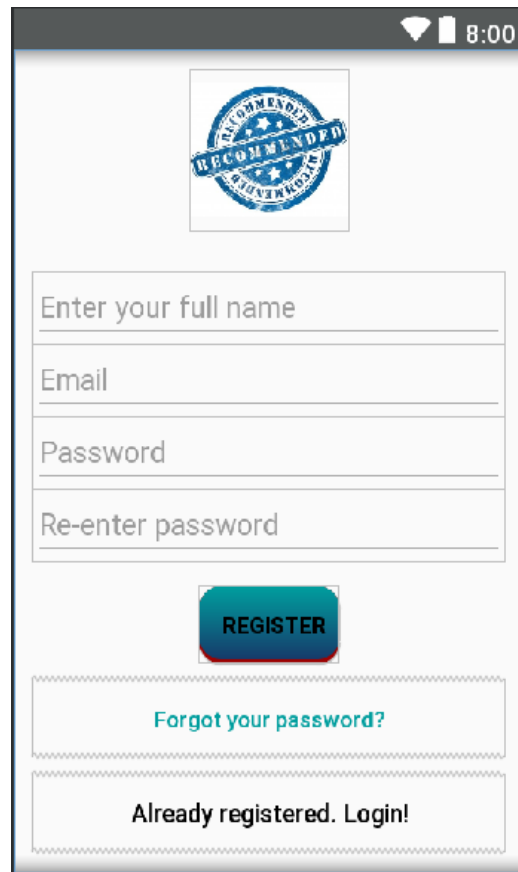


Figure 11: Εικόνα εισόδου



The image shows a mobile application registration screen. At the top, there is a status bar with a Wi-Fi icon, a battery icon, and the time 8:00. Below the status bar is a blue circular logo with the text 'RECOMMENDED' and 'ΣΙΝΕΜΑΤΑ' (Cinema) around it. The main content area contains four text input fields: 'Enter your full name', 'Email', 'Password', and 'Re-enter password'. Below these fields is a blue 'REGISTER' button. Underneath the button are two dashed-line boxes: the first contains the text 'Forgot your password?' in blue, and the second contains the text 'Already registered. Login!' in black.

Figure 12: Δημιουργία νέου λογαριασμού

1.7 Ανάκτηση κωδικού

Στην εικόνα 6 φαίνεται η δραστηριότητα ανάκτησης κωδικού. Για να έχουμε πρόσβαση σε αυτή τη δραστηριότητα αρκεί να πατήσουμε το κουμπί «Forgot your password?» είτε στην δραστηριότητα εισόδου είτε στην δραστηριότητα δημιουργίας νέου λογαριασμού (εικόνες 4&5) Στην δραστηριότητα αυτή ο χρήστης έχει την δυνατότητα να ανακτήσει τον κωδικό του σε περίπτωση που τον ξεχάσει εισάγοντας το email του στο ανάλογο πεδίο και πατώντας το κουμπί «Reset password». Άμεσα θα του σταλεί email το οποίο θα τον καθοδηγήσει στο πως να ανακτήσει τον κωδικό του.

Το κουμπί «Back» επιστρέφει τον χρήστη στην δραστηριότητα εισόδου.

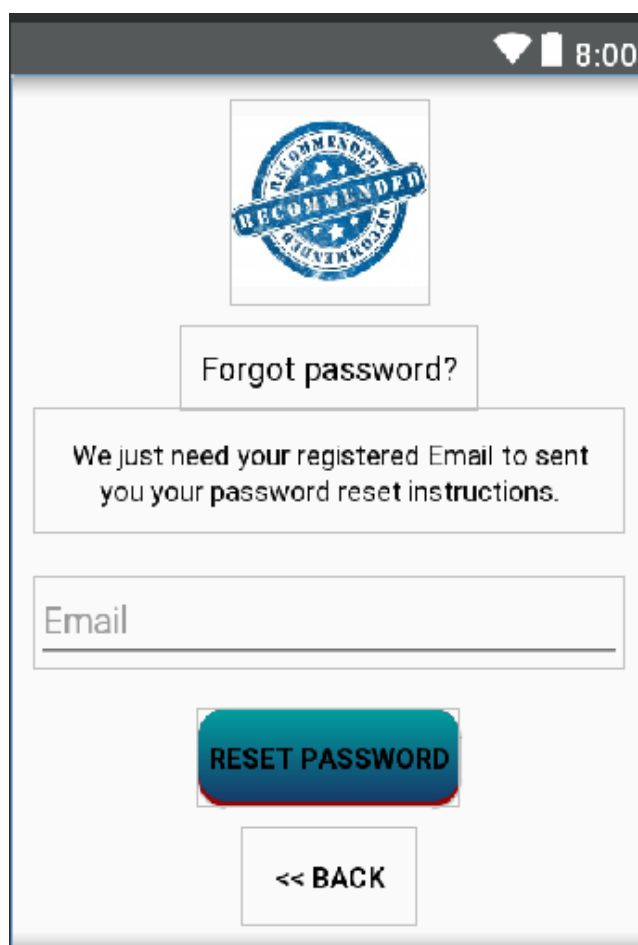


Figure 13: Ανάκτηση χαμένου κωδικού

1.8 Εξερεύνηση των κατηγοριών προτάσεων

Μετά την εγγραφή και την είσοδο του χρήστη στην εφαρμογή εμφανίζεται μια εικόνα χαιρετισμού (εικόνα 7). Αρκεί ο χρήστης να πατήσει το κουμπί «Get recommendations» και θα μεταβεί στην δραστηριότητα των κατηγοριών προτάσεων (εικόνα 8).



Figure 14: Welcoming Screen

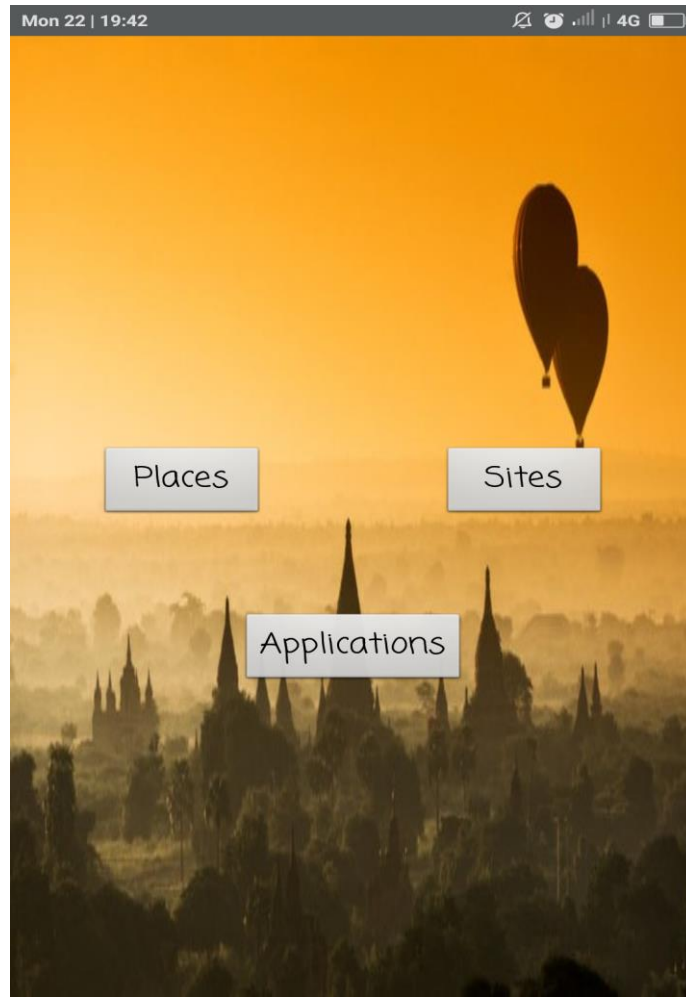


Figure 15: : Κατηγορίες προτάσεων

1.8.1 Δραστηριότητα για προτεινόμενα μέρη

Στην δραστηριότητα προτάσεων (εικόνα 7) ο χρήστης έχει την δυνατότητα να μεταβεί σε τρεις δραστηριότητες που η κάθε μια κάνει προτάσεις για μια διαφορετική θεματολογία.

Πατώντας το κουμπί «Places» η εφαρμογή είναι σε θέση να προτείνει στον χρήστη, ανάλογα με τα ενδιαφέροντα του, μαγαζιά, τουριστικά αξιοθέατα και πολλά άλλα (εικόνα 10).

Η εφαρμογή παρουσιάζει τις προτάσεις τις στηριζόμενη στην διεπαφή προγραμματισμού εφαρμογών (αγγλ.: api) «Places» της Google. Συγκεκριμένα αφού αντλήσει τις αναγκαίες πληροφορίες, όσον αφορά τον χρήστη (εικόνα 9), εμφανίζεται ένας χάρτης ο οποίος εμπλουτίζεται με τοποθεσίες οι οποίες βρίσκονται μέχρι μια συγκεκριμένη χιλιομετρική απόσταση από αυτόν και τις εμφανίζει υπό την μορφή πινακίδων πάνω στον χάρτη (εικόνα 10). Πατώντας πάνω σε μια από τις πινακίδες εμφανίζονται λεπτομέρειες για το συγκεκριμένο μέρος.

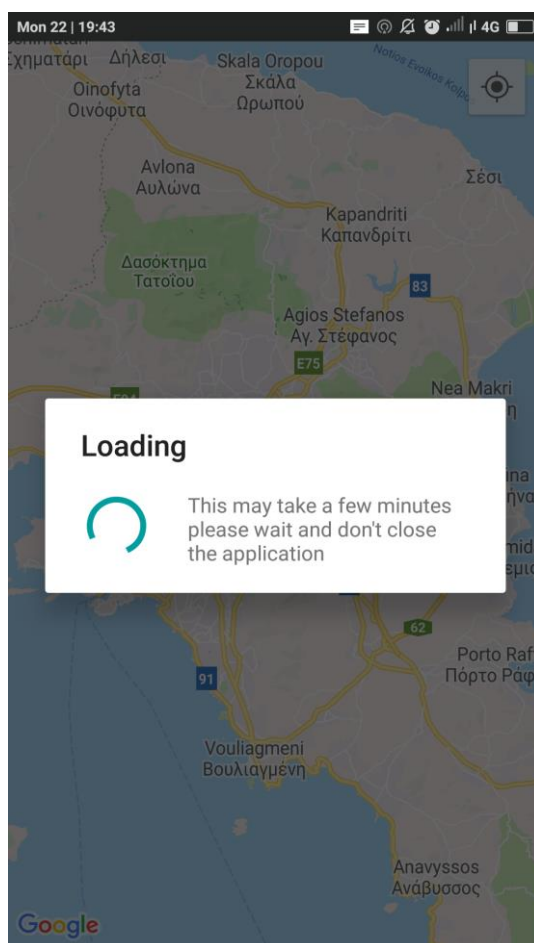


Figure 16: Προετοιμασία δραστ. Places

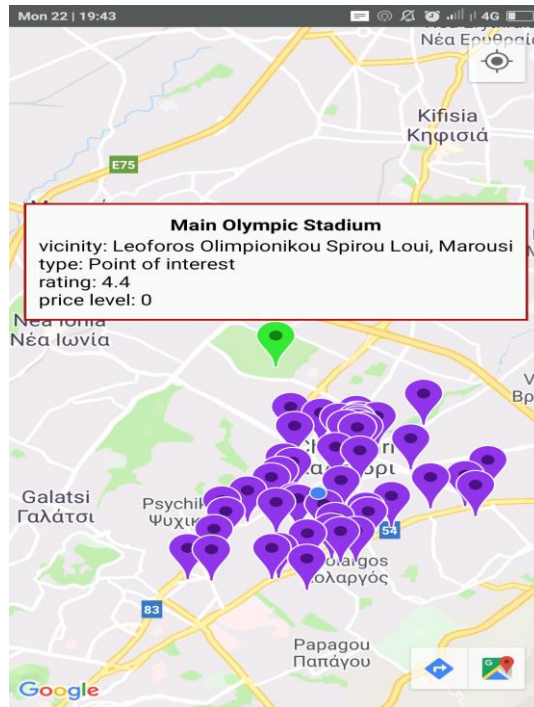


Figure 17: Δραστηριότητα Places

Σε όλες τις δραστηριότητες που μπορεί να επισκεφθεί ο χρήστης (εικόνα 11 εικόνα 12 και εικόνα 13) όταν ο χρήστης πατήσει να επιστρέψει στην δραστηριότητα κατηγοριών καλείται εάν επιθυμεί να βαθμολογήσει την εμπειρία του με 1 έως 5 αστέρια (5 αστέρια ο ανώτερος βαθμός)

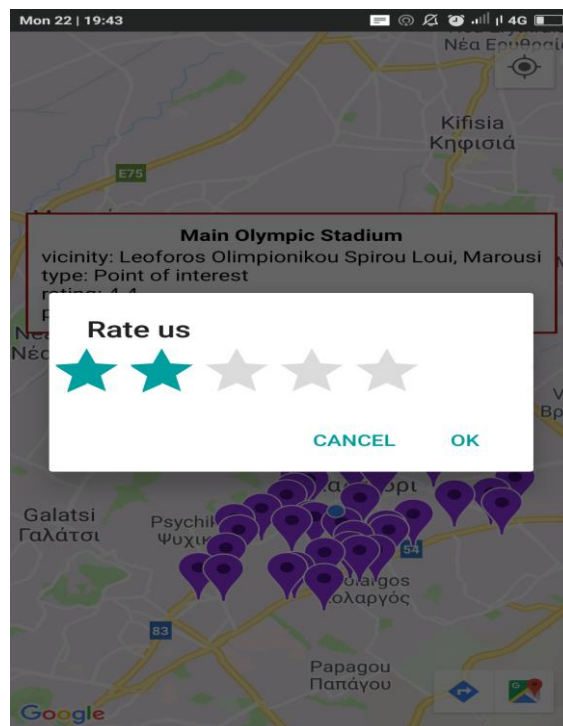


Figure 18: Βαθμολόγηση προτάσεων

Δραστηριότητα για προτεινόμενες ιστοσελίδες

Από την δραστηριότητα προτάσεων εάν ο χρήστης πατήσει το κουμπί «Sites» (εικόνα 8) η εφαρμογή είναι σε θέση να του προτείνει ενδιαφέρουσες ιστοσελίδες (εικόνα 10).

Για παράδειγμα ο χρήστης μπορεί να έχει κατεβάσει στο κινητό του μια εφαρμογή κατηγορίας «Φαγητό & ποτό» σε αυτήν την περίπτωση η συγκεκριμένη δραστηριότητα θα του προτείνει σελίδες μαγειρικής ή blogs υγιεινής διατροφής.

Επίσης μπορεί να υπάρχει μέσα στο κινητό του μια εφαρμογή τύπου «[shazam](#)» κατηγορίας «Μουσική & ήχος» σε αυτή την περίπτωση θα εμφανιστούν στον χρήστη και μερικές προτάσεις με ιστοσελίδες μουσικής.

Οι δυνατότητες είναι αμέτρητες αλλά για την περάτωση της μεταπτυχιακής διατριβής οι κατηγορίες εφαρμογών τις οποίες αναγνωρίζει η εφαρμογή μας είναι μετρημένες στα δάχτυλα.

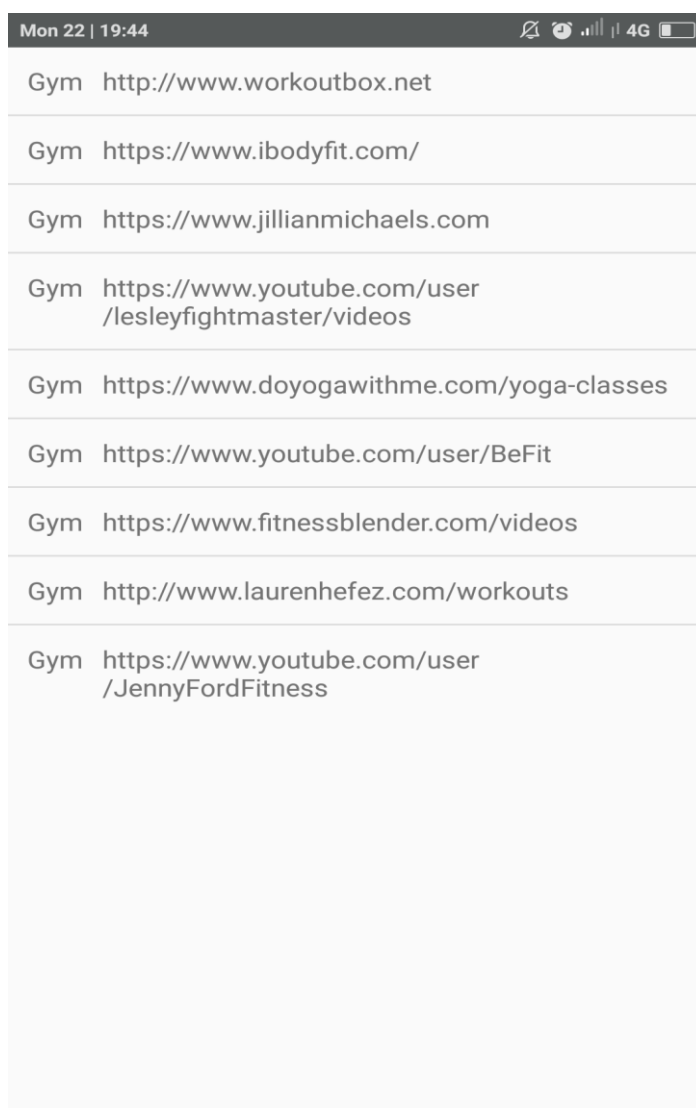


Figure 19: Δραστηριότητα προτεινόμενων ιστοσελίδων

Δραστηριότητα για προτεινόμενες εφαρμογές

Από την δραστηριότητα προτάσεων εάν ο χρήστης πατήσει το κουμπί «Applications» (εικόνα 8) η εφαρμογή είναι σε θέση να του προτείνει ενδιαφέρουσες ιστοσελίδες (εικόνα 13).

Η λειτουργία της δραστηριότητας αυτής στηρίζεται σε μεγάλο βαθμό στις πληροφορίες που έχει αντλήσει η βάση δεδομένων μας από την κοινότητα των χρηστών μας. Συγκεκριμένα οι εξατομικευμένα προτεινόμενες εφαρμογές προκύπτουν από εγκατεστημένες εφαρμογές της πλειοψηφίας των χρηστών, της κοινότητας μας, που έχουν παρόμοιες προτιμήσεις και ενδιαφέροντα με τον χρήστη.

Όπως και στις προηγούμενες δραστηριότητες οι δυνατότητες είναι αμέτρητες αλλά για την περάτωση της μεταπτυχιακής διατριβής η κατηγοριοποίηση των προτεινόμενων εφαρμογών είναι μικρής κλίμακας.

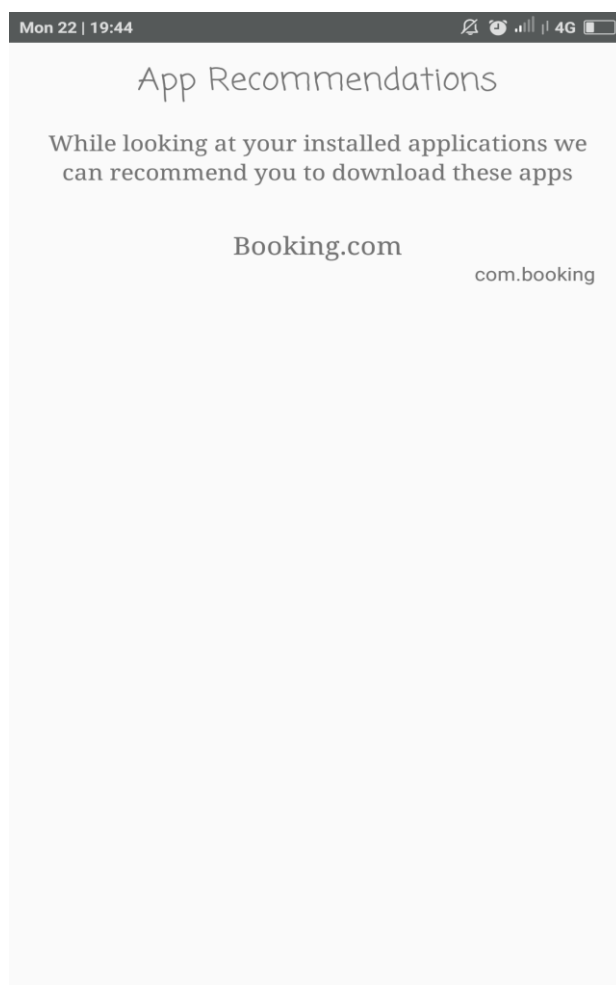


Figure 20: Δραστηριότητα προτεινόμενων εφαρμογών

3. Αρχιτεκτονική Συστήματος

1.9 Γενικά

Η εφαρμογή μας και όλες οι λειτουργίες που δεν είναι εμφανής στον χρήστη μπορούν να χωριστούν σε τρία επιμέρους τμήματα.

Στο front-end υπάρχει η android εφαρμογή μας. Η εφαρμογή όπως αναφέραμε μπορεί να εγκατασταθεί μόνο σε android κινητά ή tablets. Σε αυτήν γίνεται κυρίως η παρουσίαση των προτάσεων στον χρήστη και η επεξεργασία των πληροφοριών που μας επιστρέφει το back-end. Χρησιμοποιήθηκε το IDE «Android studio» για την κατασκευή της εφαρμογής, η γλώσσα προγραμματισμού είναι Java και η εφαρμογή μας κάνει εκτενή χρήση του «Places» api της Google.

Στο back-end υπάρχει το web service και η βάση δεδομένων μας. Το web service λειτουργεί σαν ένα Middleware, δηλαδή σαν μια γέφυρα μεταξύ της βάσης δεδομένων και της εφαρμογής. Είναι κατασκευασμένο μέσω του IDE «Eclipse» και είναι γραμμένο σε java. Σε μεγάλο βαθμό μπορεί να θεωρηθεί ένα «Restful Web Service» μιας και χρησιμοποιεί αρκετά στοιχεία της αρχιτεκτονικής αυτής.

Η βάση δεδομένων μας είναι μια owl οντολογία. Η owl είναι μια γλώσσα αναπαράστασης γνώσης και αποτελεί μια από τις βασικές συστάσεις τη κοινοπραξίας παγκόσμιου ιστού (World Wide Web Consortium – W3C). Η οντολογία μας κατασκευάστηκε με την βοήθεια του IDE «Protege».

1.10 Βάση δεδομένων

Ο Σημασιολογικός Ιστός (Semantic Web) αποτελεί μια επέκταση του σημερινού Ιστού. Η επέκταση αυτή έχει ως σκοπό την αυτοματοποίηση των λειτουργιών και των εφαρμογών του διαδικτύου, όπως είναι οι μηχανές αναζήτησης και οι πράκτορες (agents). Η αυτοματοποίηση αυτή μπορεί να επιτευχθεί μόνον εφόσον η γνώση και η πληροφορία που υπάρχει αποθηκευμένη και δημοσιευμένη αυτή τη στιγμή στο σημερινό Παγκόσμιο Ιστό αποκτήσει τυπικό νόημα (formal meaning) και σημασιολογία (semantics) και δομηθεί με ένα τέτοιο τρόπο ώστε να γίνεται κατανοητή από τις μηχανές που την επεξεργάζονται (machine understandable). Από τη στιγμή που η πληροφορία θα είναι δομημένη με έναν σημασιολογικά πλούσιο τρόπο θα ενισχύεται ο διαμοιρασμός (sharing) και η επαναχρησιμοποίησή της (reusability) επιτυγχάνοντας τη διαλειτουργικότητα (interoperability) και συνδεσιμότητα (interconnectivity) ετερογενών (heterogeneous) συστημάτων και των εφαρμογών.

Προκειμένου η γνώση και η πληροφορία να περιγραφεί με έναν τυπικό (formal) τρόπο ο οποίος θα δηλώνει τη σημασία της πρέπει να χρησιμοποιήσουμε γλώσσες αναπαράστασης γνώσης. Προκειμένου όμως να χρησιμοποιήσουμε τέτοιες τεχνολογίες στο διαδίκτυο θα πρέπει να αναθεωρήσουμε και να τροποποιήσουμε κάποια από τα συστατικά τους. Πιο συγκεκριμένα, όπως είναι γνωστό, στο σημερινό Ιστό κατά ένα πολύ μεγάλο ποσοστό η πληροφορία δομείται με τη χρήση της γλώσσας XML. Έτσι λοιπόν αφενός στο Σημασιολογικό Ιστό πρέπει να περιγράψουμε γνώση με τη χρήση κάποιων γλώσσων αναπαράστασης γνώσης αλλά αφετέρου η σύνταξη της γλώσσας που θα χρησιμοποιήσουμε θα πρέπει να βασίζεται στη γλώσσα XML. Για το λόγο αυτό η W3C, η οποία είναι ο οργανισμός που ασχολείται με την ανάπτυξη και προτυποποίηση τεχνολογιών για τον Παγκόσμιο Ιστό, έχει αναπτύξει δύο γλώσσες αναπαράστασης γνώσης. Οι γλώσσες αυτές είναι η RDF(S) και η OWL.

Με τη βοήθεια της owl δημιουργούμε μια οντολογία η οποία αποτελεί την βάση της βάσης δεδομένων μας. Με την χρήση της οντολογίας μας και των Reasoners (εμείς χρησιμοποιήσαμε τον «Pellet» reasoner) γίνεται πολύ εύκολο να εισαχθούν δεδομένα που να είναι κατανοητά από τους υπολογιστές με αποτέλεσμα να προσφέρουν παραπάνω πληροφορίες απ' ό,τι είναι αρχικά εμφανές. Ένα καλό παράδειγμα αποτελούν η εικόνα 14 και 15.

Κάθε εισαγωγή στην βάση μας γίνεται υπό την μορφή τριπλέτας από URIs π.χ.

Uri: GeorgePyliaros – URI:hasApp – URI:Excel.

Είναι εύκολο με την χρήση των δυνατοτήτων της γλώσσας owl και των reasoners η βάση μας να μπορεί να «καταλάβει» ότι από την στιγμή που ισχύει η παραπάνω τριπλέτα ισχύει και η εξής:

URI:Excel – URI :downloadedBy – URI:GeorgePyliaros

Το μόνο που χρειάζεται να ορίσουμε είναι ότι η object property hasApp είναι inverseOf της downloadedBy και τα υπόλοιπα τα αναλαμβάνουν οι reasoners.

Ο υπολογιστής μπορεί πια και «καταλαβαίνει» την πληροφορία που τον τροφοδοτούμε και μέσω απλών κανόνων μπορεί και εξάγει από μόνος του συμπεράσματα.

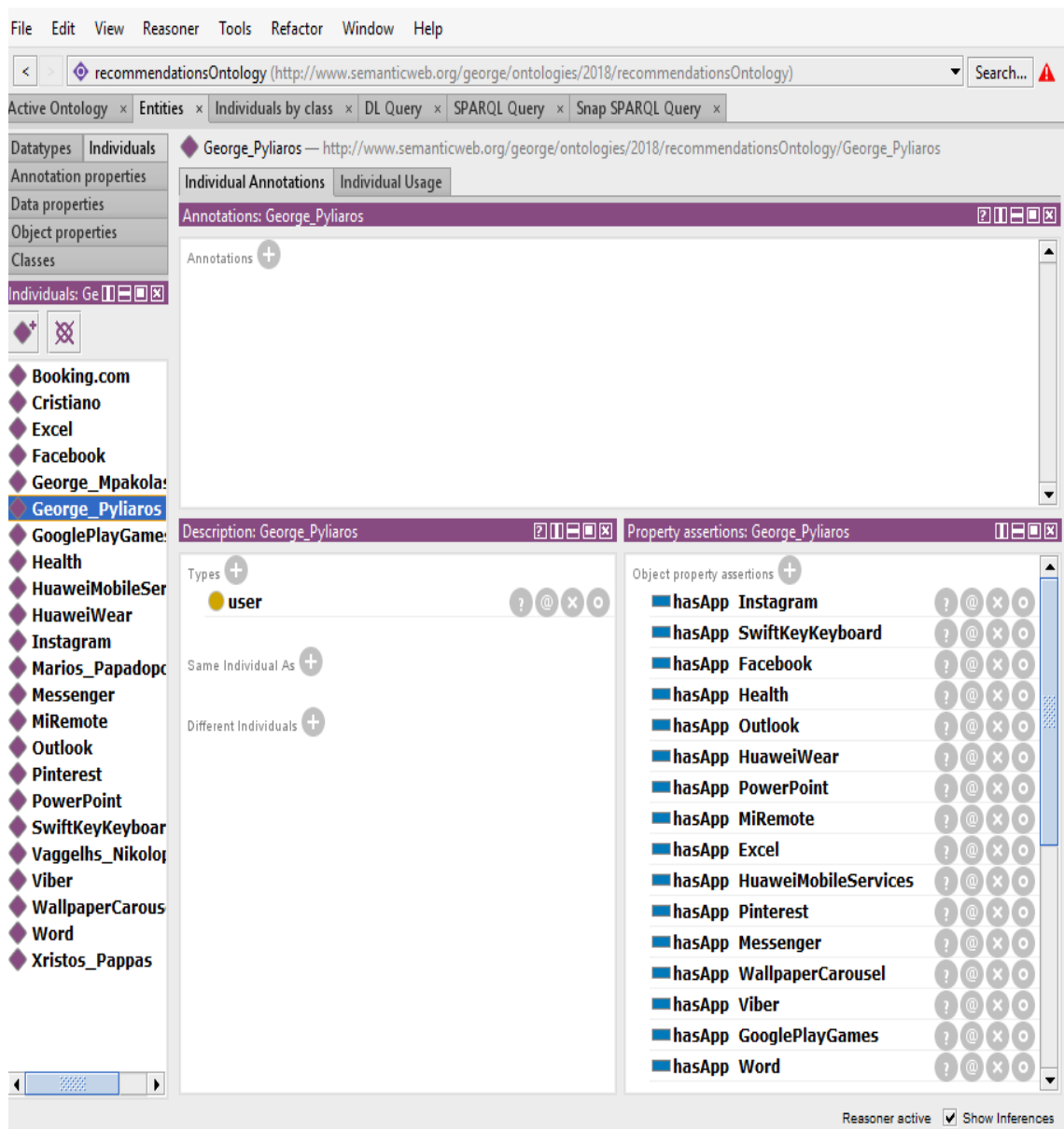


Figure 21: Individual GeorgePylarios

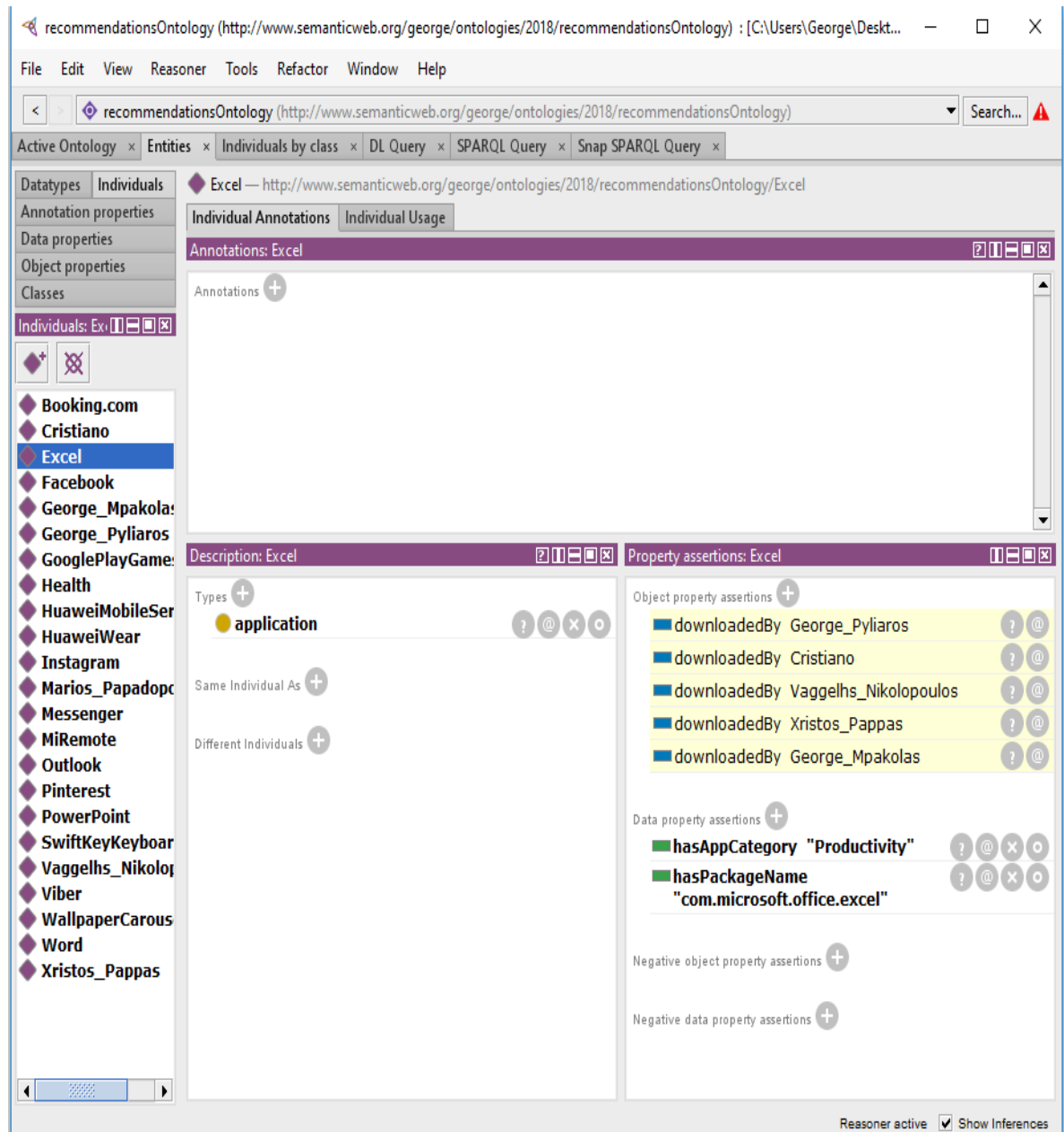


Figure 22: Protégé: Individual Excel

Άλλο ένα καλό παράδειγμα αποτελεί η εικόνα 15. Με την βοήθεια κανόνων έχουμε ορίσει (δυστυχώς hardcoded) ότι εάν το κινητό τηλέφωνο του χρήστη είναι κάποιο από τα παρακάτω μοντέλα τότε ο χρήστης ανήκει στην κατηγορία `classTwoPhone` με αποτέλεσμα να μπορούμε να κατηγοριοποιούνται εσωτερικά τα δεδομένα μας χωρίς να χρειάζεται να κάνουμε κάτι παραπάνω εμείς.

The screenshot shows the Protégé interface for the class `classTwoPhone`. The left sidebar displays the class hierarchy under `owl:Thing`, including `application`, `classOnePhone`, `classTwoPhone`, and `user`. The main area is divided into several sections:

- Annotations:** Currently empty.
- Description:** Shows the class definition:


```
(user and (hasPhone value "Redmi 6") or (hasPhone value "CUBOT H3") or (hasPhone value "CUBOT MAGIC") or (hasPhone value "CUBOT_NOVA") or (hasPhone value "CUBOT_POWER") or (hasPhone value "HUAWEI Y6 2017") or (hasPhone value "HUAWEI Y6 2018") or (hasPhone value "HUAWEI Y6 Prime 2018") or (hasPhone value "M5CR") or (hasPhone value "M5CRD") or (hasPhone value "Meizu 6T") or (hasPhone value "Meizu S6") or (hasPhone value "Moto C") or (hasPhone value "Moto C Plus") or (hasPhone value "R9") or (hasPhone value "R9 Plus") or (hasPhone value "R9Plus") or (hasPhone value "R9s") or (hasPhone value "Redmi 6A") or (hasPhone value "Redmi 6Pro") or (hasPhone value "S7A") or (hasPhone value "S7C") or (hasPhone value "U22") or (hasPhone value "Y221-U22") or (hasPhone value "Y520-U22") or (hasPhone value "Y6") or (hasPhone value "Y6 C") or (hasPhone value "Y6 Pro"))
```
- SubClass Of:** Lists `user` as a subclass.
- General class axioms:** Empty.
- SubClass Of (Anonymous Ancestor):** Empty.

At the bottom right, the status bar indicates "Reasoner active" and "Show Inferences" is checked.

Figure 23: Protégé κλάση `classTwoPhone`

1.11 Web Service

Οι υπηρεσίες Web που βασίζονται στην αρχιτεκτονική REST είναι γνωστές ως RESTful web services. Αυτές οι webservices χρησιμοποιούν μεθόδους HTTP για την εφαρμογή της αρχιτεκτονικής REST. Μια υπηρεσία RESTful web συνήθως ορίζει ένα URI, Uniform Resource Identifier και η υπηρεσία, παρέχει αναπαράσταση πόρων σε διάφορα formats όπως JSON και ένα σύνολο μεθόδων HTTP (GET, PUT, DELETE, POST)

Το πρότυπο REST υπάρχει εδώ και αρκετά χρόνια και εξακολουθεί να τραβάει μεγάλη προσοχή. Ένα RESTful API μπορεί να υλοποιηθεί στην Java με πολλούς τρόπους: μπορούμε να χρησιμοποιήσουμε το Spring ή το JAX-RS ή απλά να γράψουμε τα δικά μας γυμνά servlets (Τα Java Servlets είναι τμήματα προγράμματος Java που ανήκουν σε διακομιστές και επεξεργάζονται και απαντούν σε αιτήματα πελατών και υλοποιούν τη διεπαφή servlet.). Το μόνο που χρειαζόμαστε είναι η δυνατότητα έκθεσης σε μεθόδους HTTP, τα υπόλοιπα αφορούν το πώς θα οργανώσουμε και τον τρόπο που θα καθοδήγησουμε τον πελάτη κατά την πραγματοποίηση κλήσεων στο API μας.

Το JAX-RS δεν είναι παρά μια προδιαγραφή, ένα σύνολο διεπαφών και σχολιασμών που προσφέρονται από την Java EE. Μερικά από τα πιο γνωστά είναι τα RESTEasy και Jersey.

Για την υλοποίηση του web service χρησιμοποιήθηκε το Jersey. Το web service αποτελείται από δυο κύρια URIs :

1. `http://.../recommendations-web-service/webapi/auth`
2. `http://.../recommendations-web-service/webapi/apps`

Στην πρώτη περίπτωση όλες οι κλήσεις αφορούν το authentication του χρήστη και στην δεύτερη γίνονται όλες οι κλήσεις που αφορούν πληροφορίες αναγκαίες για τις προτάσεις της εφαρμογής.

```

~
3+ import javax.ws.rs.ApplicationPath;
5
6 @ApplicationPath("webapi")
7 public class RecommendationsApiApp extends Application{
8
9
10 }

```

Figure 24: `http://.../recommendations-web-service/webapi/`

```

30
31 @Path("auth")
32 @Produces(MediaType.TEXT_PLAIN)
33 public class AuthenticationResource {
~

```

Figure 25: `http://.../recommendations-web-service/webapi/auth`

```

35
36 @Path("apps")
37 @Produces(MediaType.TEXT_PLAIN)
38 public class ApplicationsResource {
39

```

Figure 26: `http://.../recommendations-web-service/webapi/apps`

Ο σκοπός του web service είναι να είναι η γέφυρα επικοινωνίας της βάσης δεδομένων μας με την android εφαρμογή. Η βάση δεδομένων είναι μια οντολογία, για αυτό χρειαζόμαστε ένα Api που να μπορεί να εξάγει αλλά και να εισάγει δεδομένα στην βάση. Το «Apache Jena» είναι ακριβώς αυτό. Παρέχει μια διεπαφή προγραμματισμού για την εξαγωγή και εισαγωγή δεδομένων από και σε γραφήματα RDF (Resource Description Framework) και υποστηρίζει και OWL (Web Ontology Language) οντολογίες.

Όπως προαναφέρθηκε η δομή των δεδομένων μιας οντολογίας είναι σε μορφή τριπλέτας από URIs. Για την εισαγωγή πληροφορίας στην βάση το Apache Jena παρέχει αρκετά εύκολους τρόπους (εικόνα 20)

```

104     ontModel.read(in, "");
105     OntClass user = ontModel.getOntClass(NAMESPACE + "user");
106     DatatypeProperty hasPhone = ontModel.getDatatypeProperty(NAMESPACE + "hasPhone");
107     DatatypeProperty hasEmail = ontModel.getDatatypeProperty(NAMESPACE + "hasEmail");
108     Individual individual = ontModel.createIndividual(NAMESPACE + userName, user);
109     ontModel.add(individual, hasPhone, phone);
110     ontModel.add(individual, hasEmail, userEmail);

```

Figure 27: Εισαγωγή του email και του μοντέλου του κινητού του χρήστη στην βάση

Όσον αφορά την εξαγωγή πληροφορίας είναι πιο εύκολο να βασιστούμε σε έναν άλλο τρόπο. Αυτός είναι η SPARQL (αναδρομικό ακρωνύμιο για SPARQL Protocol And RDF Query Language), μια γλώσσα που μπορεί να κάνει querying σε RDFs - δηλαδή μια σημασιολογική γλώσσα ερωτημάτων για βάσεις δεδομένων - που είναι σε θέση να ανακτήσει και να χειριστεί δεδομένα αποθηκευμένα στη μορφή RDF. Η ομάδα εργασίας για την πρόσβαση δεδομένων RDF (DAWG RDF Data Access Working Group) της Κοινοπραξίας World Wide Web αναγνώρισε την γλώσσα αυτή σαν ένα πρότυπο και αποτελεί μία από τις βασικές τεχνολογίες του σημασιολογικού ιστού.

Η SPARQL επιτρέπει ένα ερώτημα να αποτελείται από μοτίβα τριπλέτων, συζεύξεις, διαζεύξεις και προαιρετικά μοτίβα.

```

ParameterizedSparqlString pss = new ParameterizedSparqlString();
pss.setCommandText("PREFIX owl: <http://www.w3.org/2002/07/owl#> " +
    "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> " +
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#> " +
    "PREFIX rec: <http://www.semanticweb.org/george/ontologies/2018/recommendationsOntology/> " +
    "SELECT ?appCategory " +
    "WHERE { " +
        "?user rec:hasEmail ?z. " +
        "?user rec:hasApp ?app. " +
        "?app rec:hasAppCategory ?appCategory." +
    "}");
pss.setLiteral("z", userEmail);
String q = pss.toString();
Query query = QueryFactory.create(q);
QueryExecution qexec = QueryExecutionFactory.create(query, infModel);
ResultSet results = qexec.execSelect();

```

Figure 28: Εξαγωγή κατηγοριών των εγκατεστημένων εφαρμογών του χρήστη

Στην παραπάνω εικόνα η βάση θα μας επιστρέψει την κατηγορία στην οποία ανήκει κάθε εγκατεστημένη εφαρμογή του χρήστη. Το αποτέλεσμα θα επεξεργασθεί ανάλογα και θα επιστραφεί στην android εφαρμογή σε μορφή string.

1.12 Android Εφαρμογή

Το περιβάλλον ανάπτυξης της εφαρμογής είναι το android Studio και η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Java.

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον (IDE) για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Ανακοινώθηκε στις 16 Μαΐου 2013 στο συνέδριο Google I/O από την Google Product Manager, Katherine Chou. Είναι διαθέσιμο ελεύθερα με την άδεια Apache License 2.0, είναι βασισμένο στο λογισμικό της JetBrains IntelliJ IDEA και σχεδιάστηκε αποκλειστικά για προγραμματισμό Android.

Η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh (και σε PlayStation καθώς και άλλες κονσόλες παιχνιδιών) χωρίς να χρειαστεί να ξαναγίνει μεταγλώττιση (compiling) ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή (Intel x86, IBM, Sun SPARC, Motorola) αλλά και λειτουργικού συστήματος (Windows, Unix, Linux, BSD, MacOS). Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής (Virtual Machine ή VM ή EM στα ελληνικά).



Figure 29: Το λογότυπο της εφαρμογής

Για την πιστοποίηση του χρήστη (authentication) χρησιμοποιήθηκε η Firebase. Η Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητά και web που αναπτύχθηκε από την Firebase, Inc. το 2011, και στη συνέχεια εξαγοράστηκε από την Google το 2014.

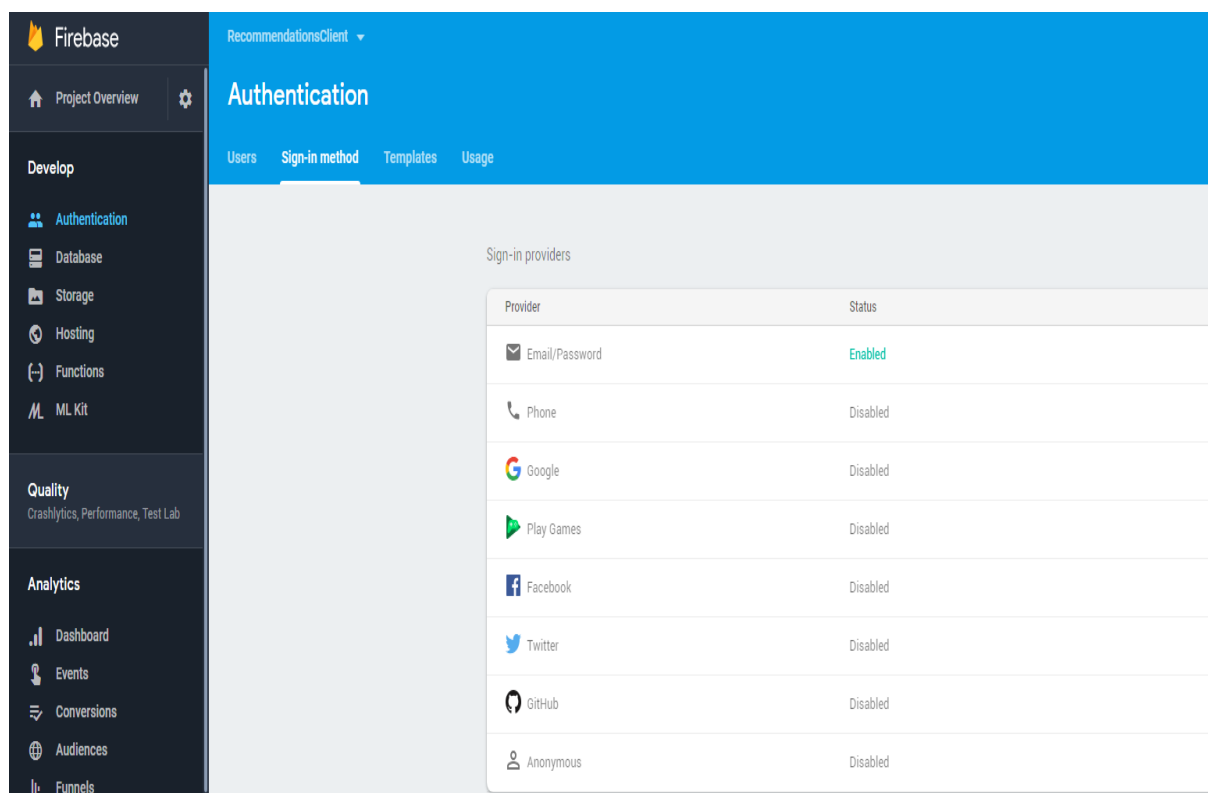


Figure 30: Η πλατφόρμα της firebase

Με την firebase απλοποιείται πάρα πολύ η διαδικασία ταυτοποίησης του χρήστη και προσφέρεται μια μεγάλη ποικιλία έτοιμων μεθόδων ταυτοποίησης (π.χ. με το τηλέφωνο του χρήστη, με τον λογαριασμό του facebook, twitter, github, google). Το μόνο που απαιτείται από τον προγραμματιστή είναι μερικές γραμμές κώδικα από τις οποίες το μεγαλύτερο κομμάτι είναι ο έλεγχος του input του χρήστη.

Η firebase εκτός από την αυτοματοποίηση και την διευκόλυνση της διαδικασίας της ταυτοποίησης του χρήστη προσφέρει και πολλές άλλες υπηρεσίες όπως την μεταφόρτωση και λήψη αρχείων για εφαρμογές, ανεξάρτητα από την ποιότητα του δικτύου, το Firebase Analytics που παρέχει στατιστικές πληροφορίες για τη χρήση της εφαρμογής και την αφοσίωση των χρηστών και πολλά άλλα.

Για την επικοινωνία της εφαρμογής με το web service (HTTP calls) χρησιμοποιήθηκε η βιβλιοθήκη «[Volley](#)» μια βιβλιοθήκη αποκλειστικά φτιαγμένη για την υποστήριξη android εφαρμογών.

```
String url = "http://"+ IP_ADDRESS+ "/recommendations-web-service/webapi/auth/" +email+ "?hasPhone=" +phone+ "userName=" +name;
StringRequest request;
request = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Log.d(TAG, msg: "ontologyRegister: StringRequest: onResponse: got response: " +response);
        callback.onSuccess(response);
    }
}, new Response.ErrorListener(){
    @Override
    public void onErrorResponse(VolleyError volleyError) {
        Log.d(TAG, msg: "ontologyRegister: StringRequest: onErrorResponse: got error response" +volleyError.toString());
        Toast.makeText( context SignupActivity.this, text "Our servers seem to be busy. Please try again in a few minutes!", Toast.LENGTH_SHORT).show();
        user = FirebaseAuth.getInstance().getCurrentUser();
        if (user != null) {
            //we use delete() method of firebase.
            user.delete().addOnCompleteListener((task) - {
                if (task.isSuccessful()) {
                    Log.d(TAG, msg: "getAppCategories: onPostExecute : user.delete(): onComplete: Successful removal of user!");
                } else
                    Log.d(TAG, msg: "getAppCategories: onPostExecute : user.delete(): onComplete: Failed to remove user!");
            });
        }
        pd.dismiss();
    }
});
RequestQueue rQueue = Volley.newRequestQueue( context SignupActivity.this);
rQueue.add(request);
```

Figure 31: HTTP POST κλήση για την αποθήκευση πληροφοριών που αφορούν τον χρήστη

Μερικά από τα πλεονεκτήματα αυτής της βιβλιοθήκης είναι :

1. Αυτόματος προγραμματισμός αιτημάτων δικτύου.
2. Πολλαπλές ταυτόχρονες συνδέσεις δικτύου.
3. Υποστήριξη για την ιεράρχηση των αιτήσεων.
4. API για ακύρωση αιτήματος. Μπορούμε να ακυρώσουμε ένα μόνο αίτημα ή μπορούμε να ορίσουμε μπλοκ αιτημάτων για ακύρωση.
5. Ευκολία προσαρμογής, για παράδειγμα, για επανάληψη κλήσεων (retry) ή αναστροφή κλήσης (backoff).
6. Ισχυρή ταξινόμηση που διευκολύνει το σωστό population του UI του χρήστη με δεδομένα που λαμβάνονται ασύγχρονα από το δίκτυο.
7. Δυνατά εργαλεία εντοπισμού σφαλμάτων και ανίχνευσης.

Για να μπορέσουμε να αναγνωρίσουμε σωστά το μοντέλο του κινητού του χρήστη . χρησιμοποιήσαμε την βιβλιοθήκη «[AndroidDeviceNames](#)» (βιβλιοθήκη του χρήστη jaredrummle στο [github](#)). Με τον τρόπο που προσφέρει το android studio για την εξαγωγή του μοντέλου του κινητού επιστρέφονται πολλαπλά ονόματα για το ίδιο μοντέλο αρκετές φορές ή πολλές φορές το όνομα είναι σε μορφή κωδικού, και στις δυο περιπτώσεις είναι δύσκολο να αναγνωριστεί ξεκάθαρα και σωστά το κινητό του χρήστη για να μπορέσει να κατηγοριοποιηθεί σύμφωνα με το πόσο ακριβό είναι. Η βιβλιοθήκη αυτή προσφέρει ένα api που απαλείφει το πρόβλημα αυτό και επιστρέφει την ονομασία του μοντέλου με την οποία πωλείται.

```
final String deviceName = DeviceName.getDeviceName().replaceAll( regex " ", replacement "_");
```

Figure 32: Χρήση της βιβλιοθήκης AndroidDeviceNames

Για την κατηγοριοποίηση των εφαρμογών του χρήστη βασιστήκαμε σε αυτήν που προσφέρει το «[play store](#)». Προκειμένου όμως να αντλήσουμε την πληροφορία αυτή από το play store χρειαζόμασταν ένα εργαλείο που μπορούσε να αναλύσει (parse) σελίδες HTML και να εντοπίσει την ανάλογη ετικέτα (tag) που κατηγοριοποιεί την εκάστοτε εφαρμογή. Για τον σκοπό αυτό χρησιμοποιήσαμε την βιβλιοθήκη «[jsoup](#)». Η jsoup είναι μια βιβλιοθήκη Java που παρέχει ένα api για την εξαγωγή και τον χειρισμό δεδομένων σε κώδικα HTML, χρησιμοποιώντας τις καλύτερες μεθόδους DOM, CSS και jquery.

```
Log.d(TAG, msg: "getCategories: Entered");
Document doc;
List<String> installedAppsCategories = new ArrayList<>();

for (String installedApp : installedApps) {
    String url = "https://play.google.com/store/apps/details?id=" + installedApp + "&hl=en";
    Log.d(TAG, msg: "getCategories: connecting with :"+url);
    Connection.Response response;
    try {
        response = Jsoup.connect(url).timeout(12000).execute();
        int statusCode = response.statusCode();
        if (statusCode == 200) {
            Log.d(TAG, msg: "getCategories: Response 200");
            doc = Jsoup.connect(url).get();
            Elements categories = doc.select( cssQuery: "a[itemprop]");

            if (!categories.isEmpty()) {
                Log.d(TAG, msg: "getCategories: got category");
                for (Element category : categories) {
                    String temp = category.text();
                    installedAppsCategories.add(temp);
                }
            } else {
                Log.d(TAG, msg: "getCategories: Parsing the above url didnt give a category");
                installedAppsCategories.add("NotFound");
            }
        }
    }
}
```

Figure 33: Χρήση της βιβλιοθήκης Jsoup για την εξαγωγή των κατηγοριών των εγκατεστημένων εφαρμογών του χρήστη

Τέλος για την υλοποίηση της δραστηριότητας που προτείνει μέρη στον χρήστη χρησιμοποιήθηκε το API «[Places](#)» της Google.

Το API Places είναι μια υπηρεσία που επιστρέφει πληροφορίες σχετικά με μέρη χρησιμοποιώντας αιτήσεις HTTP. Τα μέρη ορίζονται ως εγκαταστάσεις, γεωγραφικές τοποθεσίες ή εξέχοντα σημεία ενδιαφέροντος.

Μερικές από τις δυνατότητες που προσφέρει το API είναι:

1. Place Search: επιστρέφει μια λίστα με θέσεις βάσει της τοποθεσίας του χρήστη ή της συμβολοσειράς αναζήτησης.
2. Place Details: επιστρέφει λεπτομερέστερες πληροφορίες σχετικά με ένα συγκεκριμένο μέρος, συμπεριλαμβανομένων των κριτικών από χρήστες.
3. Place Photos: παρέχει πρόσβαση στις εκατομμύρια φωτογραφίες (που είναι σχετικές με το κάθε μέρος) που είναι αποθηκευμένες στη βάση δεδομένων του Google Place.

Κάθε υπηρεσία προσπελάζεται ως αίτημα HTTP και επιστρέφει απάντηση σε μορφή JSON ή XML.

```
if(Float.parseFloat(googlePlace.get("rating"))>3.9) {
    double lat = Double.parseDouble(googlePlace.get("lat"));
    Log.d(TAG, "msg: ShowNearbyPlaces: lat: " + lat);
    double lng = Double.parseDouble(googlePlace.get("lng"));
    Log.d(TAG, "msg: ShowNearbyPlaces: lng: " + lng);
    String rating = googlePlace.get("rating");
    if (!"-NA-".equals(googlePlace.get("price_level"))) {
        price_level = googlePlace.get("price_level");
    }
    String placeName = googlePlace.get("place_name");
    String vicinity = googlePlace.get("vicinity");
    LatLng latLng = new LatLng(lat, lng);
    String type = googlePlace.get("type");
    if ("Point of interest".equals(type)) {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_GREEN));
    } else if ("Gym".equals(type)) {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_VIOLET));
    } else if ("Night club".equals(type)) {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_YELLOW));
    } else if ("Restaurant".equals(type)) {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_AZURE));
    } else {
        markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_RED));
    }
    markerOptions.position(latLng);
    markerOptions.title(placeName);
    markerOptions.snippet("vicinity: " + vicinity + "\ntype: " + type + "\nrating: " + rating + "\nprice level: " + price_level);
    mMap.addMarker(markerOptions);
}
```

Figure 34: Τμήμα κώδικα υπεύθυνο για την δημιουργία πινακίδων πάνω στον χάρτη.

4. ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

1. Η δυνατότητα ανανέωσης του προφίλ του χρήστη. Η εφαρμογή δημιουργεί το προφίλ του χρήστη κατά την διάρκεια δημιουργίας του λογαριασμού του, δηλαδή καταχωρεί στην βάση της το μοντέλο κινητού του χρήστη αλλά και τις εγκατεστημένες εφαρμογές του. Αυτό σημαίνει πως αν ο χρήστης αλλάξει κινητό ή απεγκαταστήσει κάποια εφαρμογή ή εγκαταστήσει μια καινούρια δεν υπάρχει ή δυνατότητα ενημέρωσης της βάσης δεδομένων. Επομένως μια από τις μελλοντικές επεκτάσεις (και προφανώς άμεσα αναγκαία) είναι η δημιουργία μιας δραστηριότητας (activity) που θα είναι σε θέση να ενημερώνει την βάση με τις νέες πληροφορίες που αφορούν το προφίλ του κάθε χρήστη.
2. Περισσότερες αναγνωρίσιμες κατηγορίες εφαρμογών. Όπως προαναφέρθηκε σε ένα από τα υποκεφάλαια της διατριβής, οι δυνατότητες σημασιολογικών προτάσεων ανάλογα με τις κατηγορίες εγκατεστημένων εφαρμογών του χρήστη είναι πάρα πολλές. Για την περάτωση της μεταπτυχιακής διατριβής η εφαρμογή αναγνωρίζει μόνο μερικές κατηγορίες.
3. Επιτάχυνση της διαδικασίας κατηγοριοποίησης των εφαρμογών. Δυστυχώς ο μόνος τρόπος για να κατηγοριοποιήσουμε (δωρεάν) τις εγκατεστημένες εφαρμογές του χρήστη είναι όπως προαναφέρθηκε η χρήση της βιβλιοθήκης Jsoup. Αυτό έχει ως αποτέλεσμα η διαδικασία αυτή να είναι αρκετά αργή (για κάθε εφαρμογή εξετάζεται όλη η ιστοσελίδα της στο Play store και αντλείται η κατηγορία της). Οπότε, άλλη μια μελλοντική επέκταση θα μπορούσε να είναι η χρήση κάποιου έτοιμου API (π.χ [42matters](#)) έναντι κάποιου μηνιαίου αντίτιμου ή η δημιουργία μιας βιβλιοθήκης που θα ανανεώνεται ανά τακτά διαστήματα.
4. Places Api→ Nearby Search→price level
Αυτή την στιγμή το price level που ορίζει το πόσο ακριβό είναι ένα μέρος δυστυχώς λειτουργεί μόνο στην Αμερική. Είναι προφανές ότι όταν αυτή η δυνατότητα nearby search θα επιστρέφει αποτελέσματα και για την Ελλάδα θα προσφέρει μεγάλα οφέλη στην εφαρμογή μας (π.χ. πιο εξατομικευμένα αποτελέσματα ανάλογα με το μοντέλο κινητού του χρήστη).

5. Βιβλιογραφία

1.13 Βιβλιογραφικές Πηγές

- The Busy Coder`s Guide to Android Development, Mark L. Murphy
- Android Application Development Cookbook Second Edition, Rick Boyer Kyle Mew
- Android 6 for Programmers An App-Driven Approach 3rd Edition, Alexander Wald
- Android Programming The Big Nerd Ranch Guide 2nd Edition, Bill Phillips,Chris Stewart, Brian Hardy & Kristin Marsicano

1.14 Ιστογραφία – Δικτυότοποι

1. <https://developers.google.com/places/web-service/search>
2. <https://jsoup.org/>
3. <https://github.com/jaredrummler/AndroidDeviceNames>
4. <https://jena.apache.org/>
5. <https://jersey.github.io/>
6. <https://developer.android.com/training/volley/>
7. http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf
8. <https://en.wikipedia.org/>
9. <https://www.statista.com/>
- 10.