

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



- Πτυχιακή Εργασία -

Ανάπτυξη μίας mobile crowd sensing εφαρμογής
για έξυπνους πολίτες

Ηλίας Τζάνης

Π2016083

Επιβλέπουσα καθηγήτρια: Ελένη Χριστοπούλου

Κέρκυρα, Ιούνιος 2021

Επιβλέπουσα καθηγήτρια:

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή:

Κωνσταντίνος Οικονόμου, Καθηγητής

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης, Αναπληρωτής Καθηγητής

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Ελένη Χριστοπούλου, Εργαστηριακό Διδακτικό Προσωπικό

Τμήμα Πληροφορικής, Ιόνιο Πανεπιστήμιο

Αυτή η προσπάθεια αφιερώνεται στην οικογένειά μου, την κοπέλα μου και τους φίλους μου οι οποίοι στέκονται πάντα δίπλα μου στις δύσκολες στιγμές και με βοηθάνε να πετυχαίνω τους στόχους μου!

Περίληψη

Η εξάπλωση της τεχνολογίας σε συνδυασμό με την πληθώρα αισθητήρων που υπάρχουν στις κινητές συσκευές μας, μας δίνει τη δυνατότητα να αναπτύξουμε και να δημιουργήσουμε εφαρμογές οι οποίες χρησιμοποιούν τεχνικές κινητής πληθεπισκόπησης (Mobile - Crowdsensing). Με την κινητή πληθεπισκόπηση μπορούμε να δημιουργήσουμε εφαρμογές οι οποίες βασίζονται στη συνεργασία μεταξύ των χρηστών, ή στους αισθητήρες μίας κινητής συσκευής.

Η παρούσα πτυχιακή εργασία ασχολείται με τη δημιουργία μίας εφαρμογής η οποία χρησιμοποιεί την κινητή πληθεπισκόπηση. Πιο συγκεκριμένα, η εφαρμογή που αναπτύχθηκε προορίζεται για χρήση από κινητές συσκευές (Smartphones και Tablets) και χρησιμοποιεί τη συνεργασία μεταξύ χρηστών με σκοπό την ανάδειξη τουριστικών τοποθεσιών και αξιοθεάτων. Μέσω της εφαρμογής μπορούμε να προσθέσουμε πληροφορίες για μία τοποθεσία, αλλά και να αναζητήσουμε τοποθεσίες, ώστε να δούμε τι πληροφορίες έχουν προσθέσει οι υπόλοιποι χρήστες. Μια τέτοια εφαρμογή θα μπορούσε να βοηθήσει αρκετά άτομα να βρουν ή να ανακαλύψουν νέους προορισμούς.

Οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής είναι η React – Native για το front - end κομμάτι και η AWS Amplify για το back - end κομμάτι. Για τις δοκιμές των λειτουργιών της εφαρμογής Mapography έγινε χρήση ενός Android-Emulator. Ορισμένα συμπεράσματα που προέκυψαν κατά την διάρκεια της ανάπτυξης της εφαρμογής είναι πως για την υλοποίησή της απαιτούνται σίγουρα βασικές γνώσεις προγραμματισμού καθώς και μία εξοικείωση με τις τεχνολογίες που χρησιμοποιούνται. Επίσης αρκετά χρήσιμη είναι και μία αξιολόγηση της εφαρμογής από άλλα άτομα ούτως ώστε να δούμε ποιες λειτουργίες πραγματικά χρειάζεται η εφαρμογή μας και ποιες όχι.

Λέξεις κλειδιά: Έξυπνη πόλη, Mobile Crowdsensing, Internet of Things, React-Native, Expo, JavaScript, AWS Amplify.

Περιεχόμενα

Περίληψη.....	- 3 -
Κεφάλαιο 1. Εισαγωγή.....	- 8 -
1.1 Κοινωνικές εφαρμογές.....	- 8 -
1.2 Σκοπός της Πτυχιακής Εργασίας.....	- 8 -
1.3 Στόχος της Εργασίας.....	- 9 -
1.4 Δομή της Εργασίας.....	- 9 -
Κεφάλαιο 2. Σχετική βιβλιογραφία και εφαρμογές.....	- 11 -
2.1 Έξυπνες πόλεις.....	- 11 -
2.2 Διαδίκτυο των πραγμάτων (Internet of Things - IoT).....	- 12 -
2.2.1 Χαρακτηριστικά του Διαδικτύου των πραγμάτων.....	- 12 -
2.2.2 Εφαρμογές IoT.....	- 13 -
2.3 Mobile Crowdsensing.....	- 14 -
2.3.1 Mobile Crowdsensing και κινητές συσκευές.....	- 15 -
2.3.2 Mobile Crowdsensing Framework.....	- 16 -
2.4 Mobile crowdsensing εφαρμογές με προσανατολισμό τη συνεργασία μέσω χαρτών.....	- 17 -
2.4.1 Το Google Street View.....	- 18 -
2.4.2 Η εφαρμογή Mapillary.....	- 19 -
2.4.3 Η εφαρμογή KartaView.....	- 20 -
2.5 Αρχικός σχεδιασμός της εφαρμογής Mapography.....	- 21 -
Κεφάλαιο 3. React, React Native και το λειτουργικό σύστημα Android.....	- 25 -
3.1 Εισαγωγή στις τεχνολογίες που χρησιμοποιήθηκαν.....	- 25 -
3.2 Εισαγωγή στη React.....	- 25 -
3.2.1 Virtual Dom.....	- 26 -
3.2.2 JSX.....	- 27 -
3.3 Ανάπτυξη Cross-Platform εφαρμογών.....	- 27 -
3.3.1 React Native.....	- 28 -
3.3.2 Ο πυρήνας της React Native.....	- 29 -
3.3.3 Πλεονεκτήματα React Native.....	- 29 -
3.3.4 Μειονεκτήματα της React Native.....	- 30 -
3.3.5 Τα στοιχεία (components) της React Native.....	- 31 -
3.3.6 Διαχείριση κατάστασης στη React Native.....	- 31 -
3.4 Οικοσύστημα Android.....	- 34 -
3.4.1 Αρχιτεκτονική Android.....	- 34 -

Κεφάλαιο 4. Η αρχιτεκτονική της εφαρμογής Mapography και τεχνολογίες που χρησιμοποιήθηκαν.....	- 38 -
4.1 Γενική Περιγραφή συστήματος Mapography	- 38 -
4.2 Visual Studio Code	- 39 -
4.3 Node.js.....	- 40 -
4.4 Expo	- 40 -
4.4.1 Εγκατάσταση του Expo στον υπολογιστή μας	- 41 -
4.5 AWS AMPLIFY	- 44 -
4.5.1 Πλεονεκτήματα χρήσης του AWS Amplify σε ένα project	- 44 -
4.5.2 Εγκατάσταση AWS Amplify στον υπολογιστή	- 45 -
4.5.3 Προσθήκη AWS Amplify στην εφαρμογή.....	- 46 -
4.5.4 Amazon Cognito.....	- 47 -
4.5.5 AWS AppSync.....	- 49 -
4.6 Λειτουργίες της εφαρμογής Mapography.....	- 55 -
4.6.1 Το Ευρετήριο της εφαρμογής Mapography	- 57 -
Κεφάλαιο 5. Παρουσίαση της εφαρμογής Mapography	- 60 -
5.1 Είσοδος ή Εγγραφή του χρήστη στο Σύστημα.....	- 60 -
5.2 Αρχική οθόνη της εφαρμογής.....	- 63 -
5.2.1 Λειτουργία αναζήτησης τοποθεσιών	- 65 -
5.2.2 Λειτουργία κοντινών τοποθεσιών	- 67 -
5.3 Οθόνη προσθήκης νέου πόστ.....	- 68 -
5.4 Οθόνη Αποσύνδεσης.....	- 70 -
Κεφάλαιο 6. Αξιολόγηση της εφαρμογής	- 71 -
6.1 Παρουσίαση του ερωτηματολογίου	- 71 -
6.1.2 Δημογραφικά στοιχεία	- 71 -
6.1.2 Mobile Crowdsensing	- 72 -
6.1.3 Εφαρμογές με φωτογραφίες τουριστικών περιοχών	- 75 -
6.2 Συμπεράσματα από το Ερωτηματολόγιο	- 82 -
Κεφάλαιο 7. Συμπεράσματα και μελλοντικές επεκτάσεις.....	- 83 -
7.1 Συμπεράσματα	- 83 -
7.2 Μελλοντικές επεκτάσεις.....	- 83 -
Βιβλιογραφία	- 85 -

Πίνακας εικόνων

Εικόνα 2. 1 Mobile Crowdsensing Framework ^[13]	- 17 -
Εικόνα 2. 2 Google Street View	- 19 -
Εικόνα 2. 3 Google Street View	- 19 -
Εικόνα 2. 4 Google Street View	- 19 -
Εικόνα 2. 5 Mapillary.....	- 20 -
Εικόνα 2. 6 Mapillary.....	- 20 -
Εικόνα 2. 7 Mapillary.....	- 20 -
Εικόνα 2. 8 KartaView	- 21 -
Εικόνα 2. 9 KartaView	- 21 -
Εικόνα 2. 10 KartaView	- 21 -
Εικόνα 2. 11 Αρχικά σχέδια	- 22 -
Εικόνα 2. 12 Αρχικά σχέδια	- 22 -
Εικόνα 2. 13 Αρχικά σχέδια	- 22 -
Εικόνα 2. 14 Αρχικά σχέδια	- 23 -
Εικόνα 2. 15 Αρχικά σχέδια	- 23 -
Εικόνα 2. 16 Αρχικά σχέδια	- 23 -
Εικόνα 2. 18 Αρχικά σχέδια	- 24 -
Εικόνα 2. 17 Αρχικά σχέδια	- 24 -
Εικόνα 3. 1 React Component ^[20]	- 25 -
Εικόνα 3. 2 React Component & React Dom ^[20]	- 26 -
Εικόνα 3. 3 Rendering in React and React Native ^[25]	- 28 -
Εικόνα 3. 4 State & Props ^[21]	- 32 -
Εικόνα 3. 5 Action & Reducers ^[21]	- 33 -
Εικόνα 3. 6 Αρχιτεκτονική του λειτουργικού συστήματος Android ^[25]	- 35 -
Εικόνα 4. 1 Αρχιτεκτονική του συστήματος Mapography	- 38 -
Εικόνα 4. 2 Visual Studio Code.....	- 39 -
Εικόνα 4. 3 Δημιουργία νέου Project με το Expo.....	- 41 -
Εικόνα 4. 4 Ευρετήριο του νέου Project	- 42 -
Εικόνα 4. 5 Metro Bundler	- 43 -
Εικόνα 4. 6 Αρχική εικόνα του App.js.....	- 43 -
Εικόνα 4. 7 Σελίδες του Amazon Cognito.....	- 49 -
Εικόνα 4. 8 Σελίδες του Amazon Cognito.....	- 49 -
Εικόνα 4. 9 Σελίδες του Amazon Cognito.....	- 49 -
Εικόνα 4. 10 Επερώτηση GraphQL	- 50 -
Εικόνα 4. 11 Αποτελέσματα Επερώτησης.....	- 51 -
Εικόνα 4. 12 AWS Appsync console	- 54 -
Εικόνα 4. 13 GraphQL Mutation	- 54 -
Εικόνα 4. 14 Mapography Αναζήτηση του πόστ	- 55 -
Εικόνα 4. 15 Καρτέλα πλοήγησης της εφαρμογής Mapography	- 56 -
Εικόνα 4. 16 Πλοηγός Αποτελεσμάτων της εφαρμογής Mapography	- 56 -
Εικόνα 4. 17 Ευρετήριο της εφαρμογής Mapography	- 57 -
Εικόνα 4. 18 Ο src φάκελος του ευρετηρίου	- 57 -
Εικόνα 4. 19 Εξαρτήσεις της εφαρμογής Mapography	- 59 -
Εικόνα 5. 1 Είσοδος στην εφαρμογή.....	- 60 -

Εικόνα 5. 2 Επαναφορά κωδικού.....	- 61 -
Εικόνα 5. 3 Δημιουργία λογαριασμού.....	- 61 -
Εικόνα 5. 4 Επιβεβαίωση δημιουργίας λογαριασμού	- 62 -
Εικόνα 5. 5 Αρχική οθόνη εφαρμογής	- 63 -
Εικόνα 5. 6 Μπάρα Αναζήτησης	- 63 -
Εικόνα 5. 7 Κοντινές τοποθεσίες	- 64 -
Εικόνα 5. 8 Καρτέλα πλοήγησης	- 64 -
Εικόνα 5. 9 Σελίδα αναζήτησης.....	- 65 -
Εικόνα 5. 10 Auto-Complete	- 65 -
Εικόνα 5. 11 Αποτελέσματα σε λίστα	- 66 -
Εικόνα 5. 12 Αποτελέσματα στον χάρτη.....	- 66 -
Εικόνα 5. 13 Στοιχείο της λίστας του χάρτη.....	- 66 -
Εικόνα 5. 14 Κοντινές τοποθεσίες στον χάρτη.....	- 67 -
Εικόνα 5. 15 Πόστ κοντινών τοποθεσιών.....	- 68 -
Εικόνα 5. 16 Οθόνη προσθήκης νέου πόστ	- 69 -
Εικόνα 5. 17 Οθόνη αποσύνδεσης	- 70 -

Εικόνα 6. 1 Ερωτηματολόγιο Ερώτηση 1	- 71 -
Εικόνα 6. 2 Ερωτηματολόγιο Ερώτηση 2	- 72 -
Εικόνα 6. 3 Ερωτηματολόγιο Ερώτηση 3	- 72 -
Εικόνα 6. 4 Ερωτηματολόγιο Ερώτηση 4	- 73 -
Εικόνα 6. 5 Ερωτηματολόγιο Ερώτηση 5	- 73 -
Εικόνα 6. 6 Ερωτηματολόγιο Ερώτηση 6	- 74 -
Εικόνα 6. 7 Ερωτηματολόγιο Ερώτηση 7	- 75 -
Εικόνα 6. 8 Ερωτηματολόγιο Ερώτηση 8	- 75 -
Εικόνα 6. 9 Ερωτηματολόγιο Ερώτηση 9	- 76 -
Εικόνα 6. 10 Ερωτηματολόγιο Ερώτηση 10.....	- 76 -
Εικόνα 6. 11 Ερωτηματολόγιο Ερώτηση 11.....	- 77 -
Εικόνα 6. 12 Ερωτηματολόγιο Ερώτηση 12.....	- 77 -
Εικόνα 6. 13 Ερωτηματολόγιο Ερώτηση 13.....	- 78 -
Εικόνα 6. 14 Ερωτηματολόγιο Ερώτηση 14.....	- 78 -
Εικόνα 6. 15 Ερωτηματολόγιο Ερώτηση 15.....	- 79 -
Εικόνα 6. 16 Ερωτηματολόγιο Ερώτηση 16.....	- 80 -
Εικόνα 6. 17 Ερωτηματολόγιο Ερώτηση 17.....	- 80 -
Εικόνα 6. 18 Ερωτηματολόγιο Ερώτηση 18.....	- 81 -
Εικόνα 6. 19 Ερωτηματολόγιο Ερώτηση 19.....	- 81 -

Κεφάλαιο 1. Εισαγωγή

1.1 Κοινωνικές εφαρμογές

Σήμερα αρκετές κοινωνικές εφαρμογές, χρησιμοποιούν τεχνολογίες οι οποίες καθιστούν δυνατή την συνεργασία μεταξύ των χρηστών. Το mobile crowdsensing είναι μία τέτοια τεχνολογία και στοχεύει στην παροχή υπηρεσιών οι οποίες μέσω της συμμετοχής από το ευρύ κοινό θα μπορέσουν να συλλέξουν και να χρησιμοποιήσουν δεδομένα ανίχνευσης από ένα περιβάλλον με αποτελεσματικό τρόπο. Αυτό επιτυγχάνεται και από τις κινητές συσκευές που χρησιμοποιούμε αφού με αυτές μπορούμε να συλλέξουμε δεδομένα και να λύσουμε διάφορα προβλήματα συνεργατικά. Η ποικίλη διαθεσιμότητα αισθητήρων ανίχνευσης στις κινητές συσκευές επιτρέπει στα κοινωνικά δίκτυα τη πρόσβαση σε περισσότερα δεδομένα αλλά και την επέκταση των υπηρεσιών που προσφέρουν στους χρήστες τους. Θα μπορούσαμε λοιπόν να πούμε ότι δημιουργείται ένα αυξανόμενο ενδιαφέρον από εταιρίες κοινωνικών εφαρμογών για τη συγχώνευση των υφιστάμενων υπηρεσιών με τις υπηρεσίες που παρέχουν οι αισθητήρες πραγματικού χρόνου, όπως είναι το crowdsensing. Τα κινητά κοινωνικά δίκτυα (Mobile Social Networks) θα μπορούν αφενός να προσφέρουν μια ιδανική και πανταχού παρούσα πλατφόρμα η οποία θα επιτρέπει στους χρήστες να συμβάλλουν στο crowdsourcing και αφετέρου αυτή η πλατφόρμα θα καταφέρνει να ευαισθητοποιήσει πολλούς χρήστες σε διάφορα ζητήματα όπως περιβαλλοντικά, οικονομικά, υγειονομικά κ.α.

1.2 Σκοπός της Πτυχιακής Εργασίας

Ο σκοπός της παρούσας πτυχιακής εργασίας είναι να δούμε τον τρόπο αλλά και τις τεχνολογίες που χρειάζονται προκειμένου να αναπτυχθεί μια Mobile Crowdsensing εφαρμογή για έξυπνους πολίτες. Αρχικά θα πρέπει να γίνει μελέτη των τεχνολογιών που θα χρησιμοποιήσουμε. Οι τεχνολογίες που μελετήσαμε είναι το Mobile Crowdsensing, οι έξυπνες πόλεις αλλά και οι κοινωνικές εφαρμογές οι οποίες χρησιμοποιούν τη συνεργασία των χρηστών για την δημιουργία δεδομένων. Όλη αυτή η μελέτη βοήθησε στην καλύτερη κατανόηση αυτών των τεχνολογιών, καθώς και στο να δούμε κάποιες βασικές λειτουργίες που υπάρχουν σε παρόμοιες εφαρμογές. Άλλη μία μελέτη η οποία έπρεπε να γίνει ήταν αυτή των διαθέσιμων εργαλείων για την ανάπτυξη μίας εφαρμογής. Για να αναπτύξουμε μία εφαρμογή για κινητές συσκευές θα πρέπει αρχικά να επιλέξουμε το λειτουργικό σύστημα που θα θέλαμε να χρησιμοποιείται η εφαρμογή. Όταν αναφερόμαστε σε κινητές συσκευές, τα δύο πιο συνηθισμένα λειτουργικά είναι το iOS και το Android. Υπάρχουν εργαλεία ανάπτυξης εφαρμογών τα οποία χρησιμοποιούνται για την υλοποίηση εγγενών εφαρμογών, καθώς και κάποια τα οποία υλοποιούν cross-platform εφαρμογές. Αφού ολοκληρωθεί και η διαδικασία επιλογής κατάλληλου εργαλείου θα πρέπει να δούμε εάν η ανάπτυξη της εφαρμογής που θα υλοποιήσουμε μπορεί να ικανοποιήσει κάποια σημαντικά κριτήρια. Θα πρέπει αρχικά να μπορεί να εγκατασταθεί σε ένα από τα δύο λειτουργικά συστήματα και να είναι λειτουργική. Εφόσον επιτευχθεί αυτό άλλο ένα κριτήριο το οποίο θα πρέπει να ικανοποιεί η εφαρμογή είναι αυτό της συνεργασίας. Μέσω της εφαρμογής πρέπει οι χρήστες να είναι σε θέση να μοιράζονται συνεργατικά τα δεδομένα μεταξύ. Τέλος η εφαρμογή θα πρέπει να είναι απλή στην χρήση της και οι λειτουργίες που περιέχει να μην είναι περίπλοκες. Με αυτόν τον τρόπο οι χρήστες είναι σε θέση να καταλαβαίνουν πιο εύκολα τις λειτουργίες της εφαρμογής.

1.3 Στόχος της Εργασίας

Σε αυτή την ενότητα θα δούμε τους στόχους της εργασίας καθώς και τι θέλουμε να πετύχουμε με την εφαρμογή που θα υλοποιήσουμε. Ο αρχικός στόχος της εργασίας είναι να γίνει μία παρουσίαση που αφορά τις έξυπνες πόλεις και το Mobile Crowdsensing. Μέσω αυτής της εργασίας ο αναγνώστης θα μπορεί να κατανοήσει πως λειτουργούν οι έξυπνες πόλεις καθώς και τι τεχνολογίες χρησιμοποιούν. Μία από αυτές τις τεχνολογίες είναι και το Mobile Crowdsensing το οποίο μας βοηθάει να κάνουμε διαμοιρασμό των δεδομένων μέσω των κινητών συσκευών. Ο επόμενος στόχος αφορά τις λειτουργίες που θα περιέχει η εφαρμογή. Παρακάτω θα δούμε τι θέλουμε να μπορεί να κάνει η εφαρμογή που θα υλοποιήσουμε:

- **Δημιουργία λογαριασμού και σύνδεση:** Μέσω της εφαρμογής θα πρέπει αρχικά να μπορούμε να δημιουργήσουμε έναν λογαριασμό. Στη συνέχεια μέσω αυτού του λογαριασμού πρέπει να μπορούμε να συνδεθούμε στην εφαρμογή.
- **Αναζήτηση τοποθεσίας:** Η εφαρμογή θα πρέπει να παρέχει λειτουργία αναζήτησης. Μέσω αυτής της λειτουργίας μπορούμε να αναζητήσουμε μία τοποθεσία και να μεταφερθούμε σε αυτή.
- **Εμφάνιση αποτελεσμάτων:** Όταν ψάχνουμε μία τοποθεσία και μεταφερόμαστε σε αυτήν η εφαρμογή θα πρέπει να εμφανίζει κάποια αποτελέσματα (εικόνες, περιγραφή) τα οποία αφορούν αυτή την τοποθεσία.
- **Προσθήκη δεδομένων στην εφαρμογή:** Άλλη μία λειτουργία που αφορά την εφαρμογή είναι αυτή της προσθήκης. Μέσω της προσθήκης ο κάθε χρήστης της εφαρμογής θα είναι σε θέση να ανεβάζει τα δικά του δεδομένα στην εφαρμογή με σκοπό να τα δουν οι υπόλοιποι χρήστες της εφαρμογής.
- **Κοντινές τοποθεσίες:** Μέσω της εφαρμογής, ο χρήστης θα μπορεί να δει δημοσιεύσεις από άλλους χρήστες οι οποίες αφορούν τις κοντινές τοποθεσίες.
- **Αποσύνδεση από την εφαρμογή:** Ο τελευταίος στόχος για της λειτουργίες της εφαρμογής είναι η προσθήκη ενός κουμπιού το οποίο θα αποσυνδέει τον χρήστη από την εφαρμογή.

Άλλος ένας στόχος της εργασίας είναι η δημιουργία ερωτηματολογίου το οποίο θα απαντηθεί από διάφορους πολίτες. Το ερωτηματολόγιο θα αποτελέσει σημαντική αξιολόγηση για την εφαρμογή καθώς αρκετές απαντήσεις θα βοηθήσουν να επιτευχθεί η σωστή ανάπτυξη της εφαρμογής.

1.4 Δομή της Εργασίας

Τα επόμενα κεφάλαια της πτυχιακής εργασίας διαμορφώνονται ως εξής: Στο δεύτερο κεφάλαιο παρουσιάζεται η σχετική βιβλιογραφία και κάποιες από τις δημοφιλέστερες εφαρμογές που υπάρχουν και σχετίζονται με την mobile crowdsensing εφαρμογή Mapography. Επίσης γίνεται και μια σύντομη παρουσίαση που αφορά την εφαρμογή Mapography.

Στο τρίτο κεφάλαιο παρουσιάζεται το framework της React και της React-Native καθώς είναι η τεχνολογία που χρησιμοποιήθηκε για την υλοποίηση του demo της εφαρμογής, μαζί με τις βασικές αρχές του λειτουργικού συστήματος Android αφού είναι το λειτουργικό σύστημα στο οποίο έγιναν όλες οι δοκιμές της εφαρμογής.

Το τέταρτο κεφάλαιο περιέχει την δομή της εφαρμογής, καθώς και τις τεχνολογίες λογισμικού που σχετίζονται και επιλέχθηκαν προκειμένου να υλοποιηθεί.

Στο πέμπτο κεφάλαιο παρουσιάζεται βήμα-βήμα το demo της εφαρμογής Mapography

Το κεφάλαιο έξι αφορά την αξιολόγηση που έγινε για την εφαρμογή Mapography, καθώς δημιουργήθηκε ένα ερωτηματολόγιο με σχετικές ερωτήσεις.

Στο κεφάλαιο επτά έχουμε τα συμπεράσματα μαζί με πιθανές μελλοντικές προκλήσεις και προτάσεις.

Κεφάλαιο 2. Σχετική βιβλιογραφία και εφαρμογές

2.1 Έξυπνες πόλεις

Ο άνθρωπος από τα τέλη του 20 αιώνα μέχρι και σήμερα έχει βιώσει μια ανοδική πορεία όσον αφορά την τεχνολογία που χρησιμοποιεί. Το ίδιο φαίνεται να συμβαίνει και με τις πόλεις αφού παρατηρείται πως όσο περνάνε τα χρόνια εξελίσσονται τεχνολογικά έτσι ώστε να μπορέσουν να συμβαδίσουν με τον άνθρωπο. Τα τελευταία χρόνια μάλιστα δεν ακούμε τόσο συχνά την λέξη «πόλη» όσο ακούμε την έκφραση «έξυπνη πόλη». Όμως για αρχή ας δούμε τι είναι μία πόλη.

Σύμφωνα με το Wikipedia(2021):

«Μία πόλη είναι ένας μεγάλος και μόνιμος οικισμός με πολλά σπίτια, κατοίκους και διάφορες διοικητικές, οικονομικές ή άλλες υπηρεσίες. Στην Ελλάδα ένας οικισμός για να χαρακτηριστεί ως πόλη πρέπει να έχει τουλάχιστον 10.000 κατοίκους, ωστόσο σε άλλες χώρες αυτό δεν ισχύει.»[1]

Χάρη στις υπηρεσίες που μας παρέχουν οι πόλεις θα ήταν λάθος να πούμε ότι δεν αποτελούν σημαντικό κομμάτι για το μέλλον και την εξέλιξη του ανθρώπινου είδους. Ωστόσο η μεγάλη αύξηση του πληθυσμού δημιουργεί νέες ανάγκες και πολλές φορές οι νέες ανάγκες δημιουργούν καινούρια προβλήματα. Κάποια από τα προβλήματα που διακρίνονται σήμερα στις μεγαλουπόλεις είναι η κυκλοφοριακή συμφόρηση, η διαχείριση απορριμμάτων αλλά και ρύπανση της ατμόσφαιρας. Τέτοιου είδους προβλήματα έρχεται να λύσει μία «έξυπνη πόλη». Όμως πως γίνεται να πούμε ότι μία πόλη ξεχωρίζει από τις άλλες και μπορεί να θεωρηθεί «έξυπνη»; Σύμφωνα με τους παρακάτω αγγλικούς ορισμούς μία έξυπνη πόλη είναι:

« Μία έξυπνη πόλη μπορεί να αγγίξει όλες τις όψεις της ζωής στην πόλη: την οικονομία και την ικανότητα μετασχηματισμού, τις μεταφορές και τις υποδομές ΤΠΕ, τη διαχείριση πόρων και την προστασία του περιβάλλοντος, το ανθρώπινο κεφάλαιο και τη συμμετοχή στη δημόσια ζωή, την κοινωνικοπολιτισμική πρόοδο και τη συμμετοχή στη λήψη δημόσιων αποφάσεων.»[2]

«Μια έξυπνη πόλη συνδυάζει τρεις βαθμούς νοημοσύνης. τη νοημοσύνη και τη δημιουργικότητα των ατόμων, τη συλλογική νοημοσύνη του πληθυσμού της πόλης, καθώς και την τεχνητή νοημοσύνη που είναι ενσωματωμένη στην πόλη.»[3]

«Μια έξυπνη πόλη επενδύει σε ανθρώπινο και κοινωνικό κεφάλαιο, σε υποδομές μεταφορών και επικοινωνιών ΤΠΕ, προκειμένου να τροφοδοτήσει τη βιώσιμη οικονομική ανάπτυξη και μια υψηλή ποιότητα ζωής, καθώς και να επιτύχει συνετή διαχείριση των φυσικών πόρων μέσω συμμετοχικής δράσης και δέσμευσης.»[4]

«Μια κοινωνική έξυπνη πόλη είναι μια πλούσια σε υποδομή πόλη, η οποία συνδυάζει και εκμεταλλεύεται τόσο τους ανθρώπους όσο και την τεχνητή νοημοσύνη, μέσα από την ενδυνάμωση και εμπλοκή των ανθρώπων σε δραστηριότητες όπου οι αστικές κοινωνικές αλληλεπιδράσεις ευδοκιμούν με στόχο την προώθηση της ποιότητας ζωής και του πολιτισμού.»[5]

Για να κάνουμε λοιπόν αυτόν το διαχωρισμό θα πρέπει να χρησιμοποιηθούν τεχνολογίες οι οποίες βασίζονται σε συνδυασμό χαρακτηριστικών όπως: έξυπνο περιβάλλον, έξυπνη κινητικότητα, έξυπνη οικονομία. Αυτές οι τεχνολογίες ανήκουν στον «έξυπνο» κλάδο των Τεχνολογιών Πληροφορικής και Επικοινωνιών(ΤΠΕ) και εξαιτίας αυτού του κλάδου ο άνθρωπος κατάφερε να λύσει πολλά προβλήματα τα οποία πριν μερικά χρόνια θα μπορούσαν

να χαρακτηριστούν ως «αδύνατα». Εφαρμόζοντας λοιπόν αυτού του είδους τεχνολογίες επιτυγχάνεται η αύξηση του βιοτικού επιπέδου του ανθρώπου και οι υπηρεσίες που του παρέχει μια πόλη γίνονται πιο απλές και εύκολες.[8][8]

2.2 Διαδίκτυο των πραγμάτων (Internet of Things - IoT)

Σήμερα ζούμε στην εποχή των «έξυπνων τεχνολογιών» και μία από αυτές είναι το Διαδίκτυο των πραγμάτων (Internet of Things), το οποίο αντιπροσωπεύει την «πανταχού παρούσα πληροφορική – ubiquitous computing». Για να εφαρμοστούν οι τεχνολογίες που χρειάζεται μία έξυπνη πόλη είναι απαραίτητη η χρήση του Διαδικτύου των πραγμάτων (IoT). Ο όρος IoT χρησιμοποιήθηκε πρώτη φορά από τον Kevin Ashton το 1999 και αναφέρονταν στην ιδέα του να μπορεί μία τεχνολογία να αποκτά και να εφαρμόζει ανεξάρτητα τη γνώση. [6] Δεν υπάρχει κάποιος ξεκάθαρος ορισμός για το IoT. Μπορούμε όμως πολύ χοντρικά να πούμε ότι απευθύνεται σε όλες τις συσκευές με αισθητήρες οι οποίες είναι μοναδικές, ξεχωρίζουν με βάση τα πρωτόκολλα επικοινωνίας τους, προσφέρουν αυτονομία και αυξημένη ασφάλεια. Με την χρήση αυτών των συσκευών μπορούμε να δημιουργήσουμε διάφορες εφαρμογές οι οποίες θα διευκολύνουν την καθημερινότητα των ανθρώπων στα αστικά κέντρα αλλά θα κάνουν και την πόλη πιο έξυπνη. Τέτοιου είδους συσκευές είναι:

- Συσκευές που μπορούν να συνδεθούν στο Internet
- Δίκτυα αισθητήρων πραγματικού χρόνου
- Δυναμικά και εξελισσόμενα δίκτυα ενσωματωμένων υπολογιστικών συσκευών.

Συμπερασματικά μία συσκευή με αισθητήρες η οποία έχει τη δυνατότητα να συνδεθεί στο διαδίκτυο, μπορούμε να πούμε ότι ανήκει στο IoT.[7]

2.2.1 Χαρακτηριστικά του Διαδικτύου των πραγμάτων

Όπως όλες οι τεχνολογίες, έτσι και το Διαδίκτυο των πραγμάτων αποτελείται από κάποια θεμελιώδη χαρακτηριστικά. Σε αυτή την Ενότητα θα δούμε αυτά τα χαρακτηριστικά τα οποία απαρτίζουν το Διαδίκτυο των πραγμάτων: [9]

- **Δικτύωση (Networking):** Το ζήτημα της δικτύωσης είναι μεγάλης σημασίας καθώς περιλαμβάνει μερικούς από τους πιο σημαντικούς παράγοντες οι οποίοι χρησιμοποιούνται στη διαχείριση των δικτύων.
- **Δρομολόγηση (Routing):** Με τον όρο δρομολόγηση εννοούμε την εύρεση και επιλογή της καλύτερης διαδρομής μεταξύ μιας πηγής και του προορισμού, ώστε να ολοκληρωθεί η διαδικασία επικοινωνίας με επιτυχία.
- **Ετερογένεια (Heterogeneity):** Το IoT είναι ετερογενές επειδή περιέχει πληθώρα διαφορετικών συσκευών. Αυτές μπορούν να αλληλοεπιδράσουν μεταξύ τους καθώς ο κύριος στόχος του IoT είναι η δημιουργία κοινού τρόπου επικοινωνίας.
- **Διαλειτουργικότητα (Interoperability):** Η έννοια της διαλειτουργικότητας μπορεί να οριστεί ως η ικανότητα δημιουργίας συστημάτων ή συσκευών που μπορούν να συνεργάζονται μεταξύ τους με αποτελεσματικό τρόπο.

- **Ποιότητα υπηρεσίας (Quality of Service):** Ιδανικά θα λέγαμε πως η ποιότητα υπηρεσίας ορίζεται από τον χρόνο που απαιτεί η παράδοση ενός μηνύματος από τον αποστολέα και από τον παραλήπτη. Εάν αυτός ο χρόνος είναι μικρότερος ή ίσος από την προκαθορισμένη απαίτηση χρόνου τότε επιτυγχάνεται η ποιότητα υπηρεσίας.
- **Επεκτασιμότητα (Scalability):** Η επεκτασιμότητα είναι μία από τις πιο σημαντικές προκλήσεις ενός IoT συστήματος. Είναι η ικανότητα που έχει ένα σύστημα ή δίκτυο να χειρίζεται την αυξανόμενη κλίμακα οποιουδήποτε περιβάλλοντος χωρίς να επηρεάζεται η απόδοσή του.
- **Εικονική παρουσίαση (Virtualization):** Είναι η ικανότητα κοινής χρήσης πόρων μεταξύ πολλαπλών λειτουργικών συστημάτων. Η τεχνολογία αυτή επιτρέπει σε πολλά λειτουργικά συστήματα, εφαρμογές και υπηρεσίες να εκτελούνται στον ίδιο διακομιστή(server).
- **Μεγάλα Δεδομένα (Big Data):** Ως μεγάλα δεδομένα εννοούμε τα δομημένα και μη δομημένα δεδομένα τα οποία είναι δύσκολο να αντιμετωπιστούν με τη χρήση μίας βάσης δεδομένων ή με τεχνικές λογισμικού.
- **Υπολογιστικό Νέφος (Cloud Computing):** Το υπολογιστικό νέφος και το IoT αντιπροσωπεύουν το πεδίο «πανταχού παρόν υπολογιστές (ubiquitous computing)» αφού και τα δύο χρησιμοποιούν την έννοια της κατανεμημένης πληροφορικής. Με το υπολογιστικό νέφος έχουμε πρόσβαση σε έναν μεγάλο αριθμό υπολογιστικών πόρων και υποστηρίζει μεγάλο αριθμό χρηστών με αποκεντρωμένο τρόπο.
- **Κατανάλωση ενέργειας (Power Consumption):** Η κατανάλωση ενέργειας είναι ένα κρίσιμο πεδίο των ασύρματων δικτύων. Συνήθως η απόδοση της λειτουργίας των αισθητήρων εξαρτάται από την διάρκεια ζωής της μπαταρίας
- **Ασφάλεια και προστασία ιδιωτικότητας (Security and Privacy):** Η ασφάλεια και προστασία ιδιωτικότητας στοχεύει στην προστασία του συστήματος ή των συσκευών από απειλές. Αυτού του είδους απειλές μπορεί να είναι εξωτερικές, όπως μια επίθεση στο σύστημα από εισβολείς ή εσωτερικές απειλές, που σημαίνει κατάχρηση του συστήματος ή των πληροφοριών. [8][9]

2.2.2 Εφαρμογές IoT

Οι εφαρμογές IoT αυξάνονται καθώς ολοένα και περισσότερες συσκευές συνδέονται στο διαδίκτυο. Σχεδόν όλες οι εφαρμογές που βλέπουμε στις μέρες μας χρησιμοποιούν έξυπνες συσκευές. Αυτές οι συσκευές μοιράζονται χρήσιμες πληροφορίες, δεδομένα και μπορούν να επικοινωνούν μεταξύ τους, βελτιώνοντας τη ποιότητα της ζωής μας. Οι εφαρμογές IoT ταξινομούνται σε διάφορους τομείς όπως:

- **Έξυπνο σπίτι:** Το IoT μπορεί να φανεί χρήσιμο για ένα σπίτι αφού με τη βοήθεια του μπορούμε να ελέγξουμε συσκευές που υπάρχουν σε αυτό από απόσταση. Η παρακολούθηση του ρεύματος και του νερού είναι κάποια από τα πολλά παραδείγματα που μπορούμε να παραθέσουμε καθώς με τη χρήση του IoT

επιτυγχάνεται μείωση στους λογαριασμούς αλλά και εξοικονόμηση ενέργειας και νερού.

- **Έξυπνη μετακίνηση:** Η μετακίνηση παίζει πολύ σημαντικό ρόλο σε μία πόλη. Τα προβλήματα μετακινήσεων πρέπει να αντιμετωπίζονται αλλά και να διαχειρίζονται σωστά. Χρησιμοποιώντας IoT συσκευές μπορούμε να βελτιώσουμε την κατάσταση των μετακινήσεων με βάση τα δεδομένα που λαμβάνουμε από αυτές.
- **Έξυπνο περιβάλλον:** Κάποιες IoT τεχνολογίες μπορούν να βοηθήσουν στην πρόγνωση του καιρού και των ανεπιθύμητων φυσικών καταστροφών. Αυτό βοηθά στην έγκαιρη εκκένωση των ανθρώπων από περιοχές που πιθανόν θα πληγούν.
- **Έξυπνη υγεία:** Στον τομέα της υγείας οι IoT συσκευές θα μπορούν να φανούν πολύ χρήσιμες. Για παράδειγμα εξοπλίζοντας ένα νοσοκομείο με τέτοιες συσκευές θα μπορούμε να βοηθήσουμε τους γιατρούς να παρακολουθούν την υγεία των ασθενών τους από απόσταση χωρίς να χρειάζεται να πηγαίνουν στον καθένα ξεχωριστά.
- **Κοινωνικές εφαρμογές:** Οι συσκευές IoT συνδέονται με άλλες συσκευές και αντικείμενα μέσω του διαδικτύου. Παρέχουν διάφορες λειτουργίες οι οποίες προωθούν τις προσωπικές ανάγκες και την κοινωνική αλληλεπίδραση. Οι συσκευές αυτές βοηθούν στη συλλογή πληροφοριών σχετικά με την τοποθεσία ενός χρήστη και την παρακολούθηση των δραστηριοτήτων του.

Οι παραπάνω τομείς αποτελούν κάποια από τα πολλά παραδείγματα εφαρμογής IoT συσκευών στην καθημερινότητα των ανθρώπων. [10]

2.3 Mobile Crowdsensing

Το Mobile Crowdsensing (MCS) έχει αποκτήσει δημοτικότητα τα τελευταία χρόνια και αποτελεί χαρακτηριστικό παράδειγμα παρακολούθησης και συλλογής δεδομένων. Η τεχνολογία MCS βασίζεται σε αισθητήρες και διεπαφές επικοινωνίας που βρίσκονται σε φορητές συσκευές (smartphones, wearables). Η χρήση αυτής της τεχνολογίας αντιπροσωπεύει μία νέα προσέγγιση αξιοποίησης της δύναμης των φορητών συσκευών καθώς χρησιμοποιεί την ανθρώπινη ικανότητα απόκτησης πληροφορίας και γνώσης αλλά και τη δυνατότητα κοινοποίησης αυτών των πληροφοριών/γνώσεων με άλλους χρήστες ή μια ευρεία κοινότητα.^[Error! Reference source not found.] Οι πληροφορίες που κοινοποιούνται από τους χρήστες μπορεί να είναι από φωτογραφίες και video μέχρι δεδομένα θερμοκρασίας ή και κείμενο που περιγράφει μια κατάσταση (π.χ. έντονα καιρικά φαινόμενα). Έτσι, μπορούμε να θεωρήσουμε την MCS τεχνολογία ως ανθρωποκεντρική, αφού πυρήνα της αποτελούν δεδομένα που συλλέγονται από μεμονωμένους χρήστες φορητών συσκευών (smartphones, tablets, wearables). Η ανθρώπινη συμμετοχή αποτελεί το σημαντικότερο χαρακτηριστικό αυτής της τεχνολογίας αφού χωρίς αυτό δεν θα μπορεί να λειτουργήσει. Η βασική διαδικασία μιας MCS τεχνολογίας περιλαμβάνει τρία βήματα: συλλογή δεδομένων, αποθήκευση δεδομένων και μεταφόρτωση δεδομένων. Η συλλογή δεδομένων είναι η πρώτη φάση του MCS. Οι στρατηγικές που χρησιμοποιούνται για τη συλλογή των δεδομένων μπορούν να χωριστούν σε τρεις κατηγορίες:

- Όλα τα δεδομένα συλλέγονται χειροκίνητα από έναν χρήστη που ελέγχει τη συσκευή ανίχνευσης (smartphone, wearable) έχοντας ανοιχτή μια συγκεκριμένη εφαρμογή. Αυτή η προσέγγιση είναι αναποτελεσματική καθώς απαιτεί τη συνεχή προσοχή από τον χρήστη.
- Η συλλογή δεδομένων ελέγχεται εν μέρει από τον χρήστη και εν μέρει με δειγματοληψία η οποία πραγματοποιείται περιοδικά. Μερικές φορές τα δεδομένα μπορεί να συλλεχθούν όταν ο χρήστης ανοίξει ορισμένες εφαρμογές.
- Ανίχνευση δεδομένων με γνώμονα το περιβάλλον, όπως η τοποθεσία ενός χρήστη μία συγκεκριμένη χρονική στιγμή. Με αυτή τη μέθοδο ο χρήστης απαλλάσσεται από το να επικεντρώνεται στη συλλογή δεδομένων.

Αφού εφαρμόσουμε μία από τις τρεις στρατηγικές για την συλλογή των δεδομένων, περνάμε στη φάση της αποθήκευσης. Κατά την αποθήκευση μπορούμε να χρησιμοποιήσουμε μία μέθοδο εξάλειψης διπλότυπων δεδομένων ώστε να μειώσουμε το κόστος πόρων και να αυξήσουμε την ποιότητα της εφαρμογής. Τέλος μπορούμε να αποφασίσουμε εάν θα αποθηκεύσουμε τα δεδομένα που συλλέξαμε τοπικά ή εάν θα τα μεταφορτώσουμε στον server/cloud υπηρεσία. [12]

2.3.1 Mobile Crowdsensing και κινητές συσκευές

Οι κινητές συσκευές έχουν γίνει αναπόσπαστο κομμάτι από την καθημερινότητα μας και δεν μπορούμε να «ζήσουμε» χωρίς αυτές. Επειδή οι συσκευές αυτές είναι μονίμως συνδεδεμένες στο διαδίκτυο αποτελούν ένα πολύ σημαντικό εργαλείο που θα βοηθήσει στην ανάπτυξη και τη χρήση mobile crowdsensing εφαρμογών. Οι κινητές συσκευές προσφέρουν την δυνατότητα επέκτασης των ήδη υπάρχουσων εφαρμογών web crowdsensing καθώς μέσω των κινητών ολοένα και περισσότεροι θα μπορούν να συμβάλλουν και να συνεισφέρουν από όπου και αν βρίσκονται. Επιπλέον η δυνατότητα του να μπορείς να έχεις πολλαπλούς αισθητήρες σε μία μόνο συσκευή όπως: αισθητήρες γεωγραφικής τοποθεσίας, αισθητήρες φωτός, αισθητήρες κίνησης και αισθητήρες ήχου και εικόνας, προσφέρουν μία τεράστια ποικιλία νέων και αποδοτικών τρόπων συλλογής δεδομένων αλλά επιτρέπουν και την ανάπτυξη νέων εφαρμογών crowdsensing. Μπορούμε να ταξινομήσουμε τις crowdsensing εφαρμογές κινητών συσκευών είτε σαν επέκταση ήδη εφαρμογών web που ήδη υπάρχουν είτε ως νέες εφαρμογές. Άλλη μία διάκριση που μπορούμε να κάνουμε για αυτές τις εφαρμογές είναι εάν η συμβολή του πλήθους είναι συμμετοχική ή ευκαιριακή. Συνήθως το συμμετοχικό crowdsensing λειτουργεί με υπολογισμούς ή με τη δημιουργία δεδομένων εισόδου από τους ίδιους τους χρήστες. Από την άλλη το ευκαιριακό crowdsensing λειτουργεί μέσα από δεδομένα που παράγονται από αισθητήρες και αυτόματους υπολογισμούς που εκτελούνται από τις κινητές συσκευές. Οι web crowdsensing εφαρμογές ανήκουν στην συμμετοχική κατηγορία καθώς απαιτούν την ενεργή συμμετοχή των χρηστών τους. Αντίθετα στις περισσότερες mobile crowdsensing εφαρμογές οι εργασίες συλλογής δεδομένων εκτελούνται στο παρασκήνιο χρησιμοποιώντας τους αισθητήρες και έτσι ανήκουν στην ευκαιριακή κατηγορία. [13]

2.3.2 Mobile Crowdsensing Framework

Το crowdsensing χωρίζεται σε διαφορετικούς τομείς. Σε αυτή την ενότητα θα εξετάσουμε αυτούς τους τομείς και θα δούμε ένα σχέδιο της δομής ενός crowdsensing συστήματος:

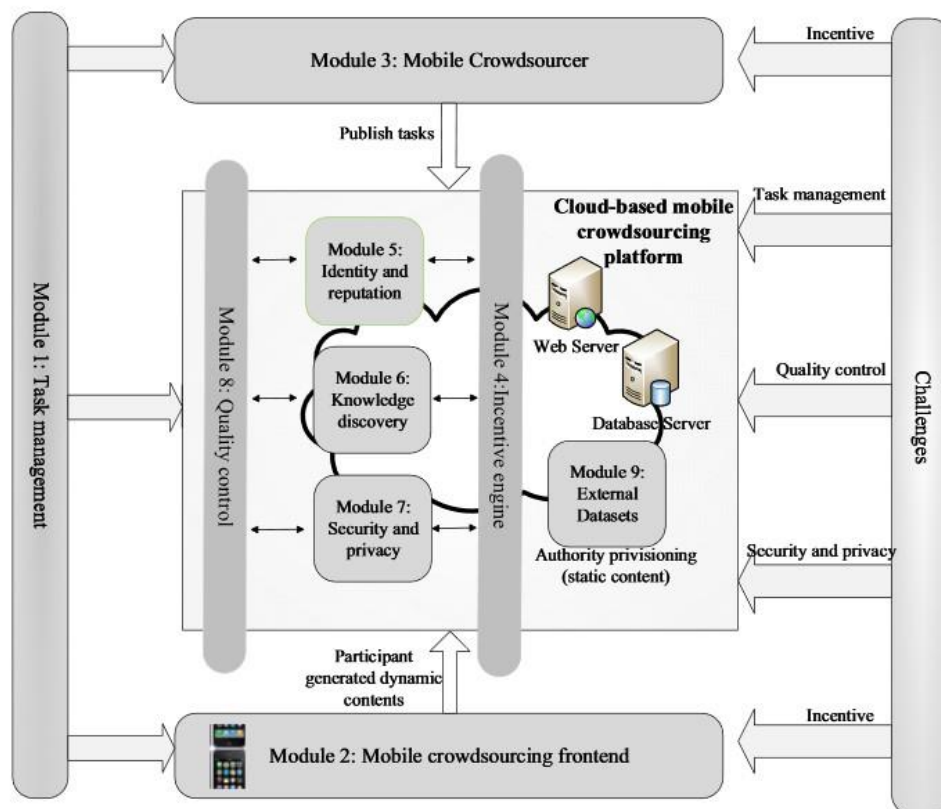
- **The Crowd** (Το πλήθος): Το πλήθος αποτελείται από τους ανθρώπους οι οποίοι συμμετέχουν σε μία δραστηριότητα crowdsensing.
- **The crowdsourcer**: Ο crowdsourcer είναι μία οντότητα (άτομο, κερδοσκοπικός οργανισμός, μη κερδοσκοπικός οργανισμός, κ.λπ.) η οποία αναζητά την δύναμη και τη γνώση του κοινού που μπορεί να έχει πάνω σε ένα θέμα.
- **Η πλατφόρμα crowdsensing**: είναι ένα σύστημα μέσα στο οποίο εκτελείται μια εργασία crowdsensing.

Στο παρακάτω σχήμα παρουσιάζεται η γενική δομή ενός mobile crowdsensing συστήματος που αφορά τις κινητές συσκευές. Όπως φαίνεται και στην **Εικόνα 2.1** η παρακάτω δομή αποτελείται από πολλαπλά modules τα οποία είναι ανεξάρτητα και μπορούν να περιλαμβάνουν πολλαπλές πηγές (sources) από δεδομένα. Θα εξηγήσουμε εκτενώς το κάθε module παρακάτω:

- **Module 1, task management**: αφορά τις προδιαγραφές ανίχνευσης (sensing) και τις περιπτώσεις χρήσης, συμπεριλαμβανομένων τον τύπο συμμετεχόντων, τον απαιτούμενο ρυθμό δειγματοληψίας του κάθε αισθητήρα και τις απαιτήσεις οπτικοποίησης και αναπαράστασης δεδομένων.
- **Module 2, front-end**: παρέχει στους χρήστες τη διεπαφή με την οποία αλληλοεπιδρούν και μπορούν να αναφέρουν crowdsensing δεδομένα με συμμετοχικό ή / και ευκαιριακό τρόπο.
- **Module 3, mobile crowdsourcer**: είναι υπεύθυνο για την ανάθεση της κατάλληλης διεργασίας στην πλατφόρμα. Αλληλοεπιδρά με το Task management module και ανατροφοδοτεί τα δεδομένα που έχουν δημιουργηθεί από τους χρήστες.
- **Module 4, incentive engine**: η ύπαρξη αυτού του module έχει δύο στόχους. Ο πρώτος στόχος είναι να μπορεί να ενθαρρύνει τους χρήστες μίας εφαρμογής να συμμετέχουν ενεργά και να προσφέρουν δεδομένα υψηλής ποιότητας. Ο δεύτερος στόχος είναι να μπορεί ο crowdsourcer να παρέχει ανατροφοδότηση σχετικά με την ποιότητα των δεδομένων που έχουν συλλεχθεί.
- **Module 5, identity, and reputation**: διαχειρίζεται την ταυτότητα των συμμετεχόντων και το πόσο σωστά είναι τα δεδομένα που αντλούνται από αυτούς. Στόχος του συγκεκριμένου module είναι η βελτίωση της ποιότητας των crowdsensing δεδομένων που προσφέρουν οι χρήστες.
- **Module 6, knowledge discovery**: Σε ένα mobile crowdsensing σύστημα τα δεδομένα που υποβάλλονται από τους χρήστες ή / και τους αισθητήρες των κινητών συσκευών μπορεί να είναι μη δομημένα και πολλές φορές λανθασμένα. Εδώ έρχεται το Knowledge discovery module το οποίο αφορά την επεξεργασία, την εξαγωγή και την

ανακατασκευή χρήσιμων πληροφοριών από τα ανεπεξέργαστα δεδομένα που υποβάλλουν οι συμμετέχοντες.

- **Module 7, security, and privacy:** στοχεύει στην ασφάλεια και την προστασία του απορρήτου των χρηστών κατά τη διάρκεια χρήσης μίας mobile crowdsensing εφαρμογής.
- **Module 8, quality control:** είναι υπεύθυνο για τον έλεγχο και την προσαρμογή των module 5,6 και 7 με στόχο την επίτευξη υψηλότερης ποιότητας υπηρεσίας.
- **Module 9, external datasets:** καθώς ορισμένες κυβερνητικές αρχές ή εταιρίες παρέχουν δωρεάν ανοιχτά δεδομένα, αυτό το module αφορά το πως μπορούμε να ενσωματώσουμε στο σύστημά μας αυτά τα δεδομένα ώστε να το κάνουμε πιο λειτουργικό και να το εμπλουτίσουμε με περισσότερες υπηρεσίες.



Εικόνα 2. 1 Mobile Crowdsensing Framework [14]

2.4 Mobile crowdsensing εφαρμογές με προσανατολισμό τη συνεργασία μέσω χαρτών

Η αναπαράσταση δεδομένων σε έναν χάρτη είναι ένα πολύτιμο πλεονέκτημα το οποίο για την εποχή μας θεωρείται κάτι απλό και δεδομένο. Αυτό συμβαίνει διότι οι συσκευές μας πλέον

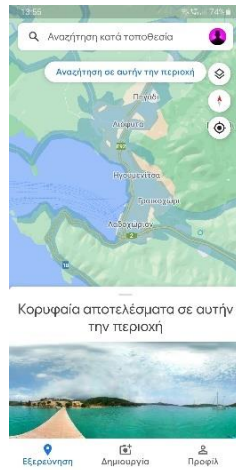
έχουν το απαραίτητο υλικό για την εύρεση τοποθεσίας, οπτικοποίηση και άντληση δεδομένων, αλλά και οι χρήστες τα τελευταία χρόνια με την τεράστια εξάπλωση των social media αρέσκονται στο να είναι «κοινωνικοί» και να συνεισφέρουν. Η εφαρμογή που θα δημιουργήσουμε (Mapography) αποτελεί μία εφαρμογή ή οποία θέλει να «εκμεταλλευτεί» το hardware των κινητών συσκευών αλλά και την πολύτιμη βοήθεια των χρηστών του καθώς ο κύριος στόχος είναι η αναπαράσταση δεδομένων σε έναν χάρτη τα οποία έχουν δημιουργηθεί από τους χρήστες. Τα δεδομένα αυτά είναι φωτογραφίες οι οποίες έχουν δημιουργηθεί αποκλειστικά και μόνο από τους χρήστες. Σκοπός είναι η ανάδειξη και η εξερεύνηση των διάφορων τοπίων-προορισμών από τους υπόλοιπους χρήστες που είτε βρίσκονται κοντά σε αυτά, είτε θέλουν να βρεθούν στο μέλλον. Αδιαμφισβήτητα λοιπόν το Mapography δεν αποτελεί κάποια ιδέα η οποία είναι καινούρια και δεν έχει ήδη υπάρξει. Για τον λόγο αυτό είναι σημαντικό σε αυτή την υπό-ενότητα να δούμε παρόμοιες εφαρμογές που ήδη υπάρχουν καθώς και τον τρόπο με τον οποίο λειτουργούν.

2.4.1 To Google Street View

Όταν θέλουμε να ψάξουμε για ένα μέρος ή μία τοποθεσία το πρώτο πράγμα που μας έρχεται στο μυαλό είναι το Google Maps και όταν θέλουμε να δούμε street-level φωτογραφίες το Google Street View. Το Street View ξεκίνησε το 2007 στις Η.Π.Α. [16] και από τότε εξελίσσεται ραγδαία, με διεθνή κάλυψη συμπεριλαμβανόμενων αρκετών μεγάλων περιοχών όπως η Αμερική, η Ευρώπη και τμήματα της Ασίας. Διαθέτει έναν μεγάλο αριθμό από φωτογραφίες οι οποίες περιέχουν λεπτομερείς γεωγραφικές συντεταγμένες και πληροφορίες το οποίο δίνει τη δυνατότητα να διεξαχθούν διάφορες επιστημονικές μελέτες. Το περιεχόμενο του Street View προέρχεται από δύο πηγές, την Google και τους χρήστες. Ο χρήστης μπορεί να ανεβάσει φωτογραφίες μέσω του κινητού του, αλλά μπορεί επίσης και να δημιουργήσει ένα δικό του Street View χρησιμοποιώντας τον κατάλληλο εξοπλισμό που παρέχεται από την Google. Αυτό μπορεί να το κάνει είτε περπατώντας, είτε κάνοντας ποδήλατο, είτε οδηγώντας. Επίσης δίνεται και η δυνατότητα προσθήκης νέων μονοπατιών ή δρόμων που δεν είναι καταχωρημένοι στην εφαρμογή. Μπορεί λοιπόν κανείς να πει ότι το street-view αποτελεί μία διασκεδαστική υπηρεσία για τους χρήστες του. Η ιδέα του να μπορείς να καταγράψεις το δρόμο και τα κτίρια μπορεί κάποτε να φαινόταν ξεπερασμένη, αλλά η σημερινή εποχή έχει αποδείξει πόσο χρήσιμη και απαραίτητη είναι για εμάς.

Ας δούμε λοιπόν κάποιες βασικές λειτουργίες του Google Street View. Αρχικά για να χρησιμοποιήσουμε την εφαρμογή θα πρέπει να είμαστε συνδεδεμένοι σε έναν λογαριασμό Google. Με το που συνδεθούμε στο λογαριασμό αποκτάμε πρόσβαση στις λειτουργίες της εφαρμογής. Στην **Εικόνα 2.2** βλέπουμε την αρχική σελίδα στην οποία μπορούμε να εξερευνήσουμε μέσω της αναζήτησης τις αμέτρητες τοποθεσίες που παρέχει το Google street view και από κάτω κάποια κορυφαία αποτελέσματα. Τα κορυφαία αποτελέσματα αφορούν φωτογραφίες που έχουν ανεβάσει άλλοι χρήστες της εφαρμογής, οι οποίες έχουν τραβηχτεί γύρω από την εκάστοτε περιοχή που αναζητεί ο χρήστης. Μπορούμε επίσης να επιλέξουμε τον τύπο χάρτη δηλαδή ο κανονικός ή ο δορυφόρος. Στην **Εικόνα 2.3** έχουμε τη δημιουργία και το ανέβασμα μίας φωτογραφίας 360 μοιρών. Εδώ βλέπουμε δύο επιλογές, την εισαγωγή υπάρχουσας φωτογραφίας και την σύνδεση μίας κάμερας 360 μοιρών ώστε να τραβήξουμε μία καινούρια φωτογραφία. Τέλος στην **Εικόνα 2.4** έχουμε το προφίλ του συνδεδεμένου χρήστη. Μπορούμε αρχικά να αναζητήσουμε το περιεχόμενο που έχει ανεβάσει στην εφαρμογή ο συνδεδεμένος χρήστης, να δούμε κάποια στατιστικά που αφορούν τον συγκεκριμένο χρήστη και υπάρχει και η επιλογή που αφορά τη συνεισφορά και πατώντας το κουμπί έναρξης μεταφερόμαστε στη **Εικόνα 2.3**. Συνοψίζοντας η χρήση της συγκεκριμένης εφαρμογής είναι

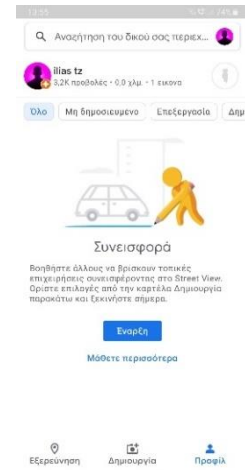
πολύ απλή χωρίς πολλά menu τα οποία θα αυξάνανε την πολυπλοκότητά της και προσφέρει τεράστιο όγκο δεδομένων από τοποθεσίες, ο οποίος δημιουργήθηκε τόσο από την Google αλλά και από την συνεργασία και την συνεισφορά των χρηστών της.



Εικόνα 2. 2 Google Street View



Εικόνα 2. 4 Google Street View



Εικόνα 2. 3 Google Street View

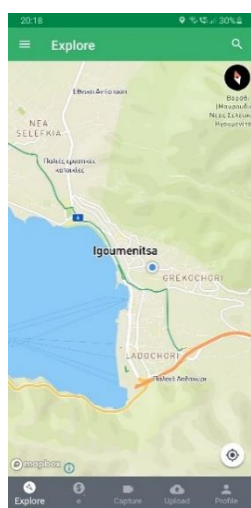
2.4.2 Η εφαρμογή Mapillary

Το Mapillary είναι μία υπηρεσία που κυκλοφόρησε το 2013 από την εταιρία Mapillary AB και αγοράστηκε το 2020 από το Facebook. [17] Η εφαρμογή υπάρχει για συσκευές Android, iOS αλλά και για υπολογιστές. Κύριος στόχος του Mapillary είναι η παροχή street-level φωτογραφιών και δεδομένων χαρτών για όλους τους χρήστες της υπηρεσίας. Σύμφωνα με την ίδια την εταιρία οι άνθρωποι και οι οργανισμοί που συνεργάζονται μεταξύ τους είναι ο καλύτερος τρόπος για τη συλλογή, οπτικοποίηση και κατανόηση των δεδομένων που υπάρχουν γύρω μας. [17] Το Mapillary σαν εφαρμογή είναι αρκετά απλό. Αρχικά όταν ανοίξουμε για πρώτη φορά την εφαρμογή θα πρέπει να δημιουργήσουμε έναν λογαριασμό ώστε να μπορούμε να την χρησιμοποιήσουμε. Οι δυνατότητες δημιουργίας ενός λογαριασμού είναι οι εξής:

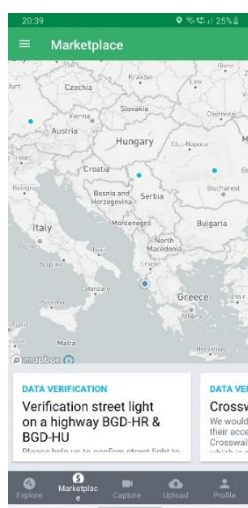
- Μέσω λογαριασμού Facebook
- Μέσω λογαριασμού Google
- Μέσω email και κωδικού
- Μέσω Open street Map λογαριασμού

Αφού διαλέξουμε έναν από τους παραπάνω τρόπους για να δημιουργήσουμε λογαριασμό, θα πρέπει να επιλέξουμε ένα μοναδικό username και να αποδεχτούμε τους όρους χρήσης της εφαρμογής. Τελειώνοντας τα παραπάνω βήματα αποκτάμε πρόσβαση στην κεντρική σελίδα της εφαρμογής (**εικόνα 2.5**). Στην **εικόνα 2.5** βλέπουμε ότι στο κάτω μέρος υπάρχουν 5 tabs και το κύριο tab είναι το πρώτο (explore). Στο explore όπως το λέει και η λέξη ο χρήστης μπορεί να εξερευνήσει τους χάρτες, να δει διάφορα σημεία που επιθυμεί μέσω street level φωτογραφιών και να προσθέσει και κάποιο πιθανό σχόλιο στο σημείο που έχει επιλέξει. Όλα

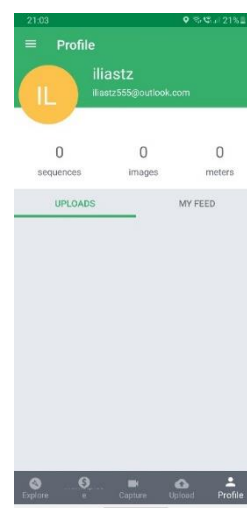
τα σημεία στο Mapillary που επιθυμεί να δει ένας χρήστης είναι φωτογραφίες που έχουν τραβηχτεί από τους υπόλοιπους χρήστες της εφαρμογής ανά τον κόσμο. Στην **εικόνα 2.6** βλέπουμε το δεύτερο tab το οποίο ονομάζεται Marketplace. Στο Marketplace μπορεί ο χρήστης να επιλέξει κάποιο από τα Data Verification που είναι διαθέσιμο μια συγκεκριμένη χρονική περίοδο. Ζητάει δηλαδή η εφαρμογή από τους χρήστες της σε διάφορες περιοχές του κόσμου να επιβεβαιώσουν διάφορα δεδομένα όπως: εάν υπάρχουν πεζοδρόμια, εάν υπάρχουν φώτα ή εάν υπάρχουν πινακίδες για άτομα με ειδικές ανάγκες κ.α. Τα επόμενα δυο tabs το Capture και το Upload δίνουν τη δυνατότητα στον χρήστη να τραβήξει φωτογραφία με την ακριβή τοποθεσία όπου βρίσκεται και ύστερα πηγαίνοντας στο Upload να την «ανεβάσει» στους χάρτες της εφαρμογής. Τέλος στην **εικόνα 2.7** βλέπουμε το προφίλ ενός χρήστη της εφαρμογής. Εδώ παρέχονται στατιστικά του συγκεκριμένου προφίλ, ακολουθίες, φωτογραφίες, μέτρα και από κάτω υπάρχουν τα Uploads του εκάστοτε χρήστη και δίπλα το My Feed το οποίο αφορά διάφορες ειδοποιήσεις ή πιθανά συμβάντα που μπορεί να ενδιαφέρουν τον συγκεκριμένο χρήστη.



Εικόνα 2. 7 Mapillary



Εικόνα 2. 6 Mapillary



Εικόνα 2. 5 Mapillary

2.4.3 Η εφαρμογή KartaView

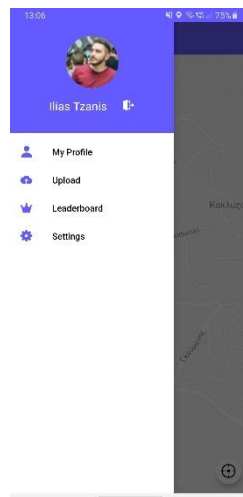
Το KartaView είναι άλλη μία mobile crowdsensing εφαρμογή η οποία παίρνει τα δεδομένα που της δίνουν οι χρήστες της και τα αναπαριστά στους χάρτες. Όπως και το Mapillary έτσι και αυτό είναι μια εφαρμογή η οποία προσφέρει φωτογραφίες street-level επιπέδου πάνω σε έναν χάρτη. Ιδρύθηκε το 2009 με το όνομα OpenStreetView, το 2016 μετονομάστηκε σε OpenStreetMap και το 2020 πήρε το όνομα που έχει μέχρι και σήμερα, το KartaView. Ο σκοπός του KartaView είναι η βελτίωση του OpenStreetMap και η εταιρία που διαχειρίζεται την εφαρμογή είναι η Grab Holdings. [18]

Η κύρια διαφορά του KartaView από το Mapillary είναι ότι και η web αλλά και οι mobile εφαρμογές είναι εν μέρη open source. Ανοίγοντας για πρώτη φορά την εφαρμογή διαπιστώνει κανείς ότι σε σύγκριση με το Mapillary δεν χρειάζεται κάποιος να έχει λογαριασμό προκειμένου να ανεβάσει ή να δει κάποια φωτογραφία στον χάρτη. Αυτό είναι ένα πολύ

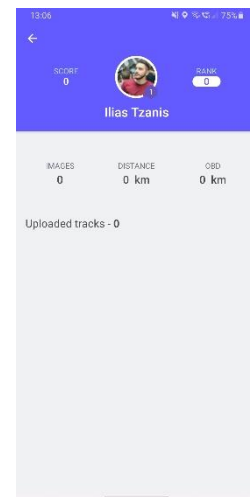
χρήσιμο feature καθώς υπάρχουν πολλοί χρήστες οι οποίοι θέλουν να δοκιμάσουν μία εφαρμογή πριν δημιουργήσουν λογαριασμό ή δεν επιθυμούν να δημιουργήσουν κάποιο λογαριασμό. Στην **εικόνα 2.8** φαίνεται πόσο απλή είναι η εφαρμογή καθώς το μόνο που υπάρχει εκτός από τον χάρτη είναι το κεντρικό κουμπί με το οποίο μπορεί ο χρήστης να τραβήξει μία φωτογραφία και δίπλα το κουμπί εντοπισμού. Επίσης πάνω αριστερά υπάρχει και το toast menu από το οποίο μπορούμε να μεταβούμε στο προφίλ και στις ρυθμίσεις της εφαρμογής (**εικόνα 2.9**). Κάτι που αξίζει να σημειωθεί είναι ότι όταν ο χρήστης θέλει να δει ένα μέρος στον χάρτη παρέχεται η δυνατότητα οπτικοποίησης φωτογραφιών 360 μοιρών. Επίσης μία ακόμη διαφορά του KartaView από το Mapillary είναι και η παροχή επιβράβευσης του χρήστη καθώς όπως φαίνεται στην **εικόνα 2.10** υπάρχει το score που έχει κάνει ένας χρήστης μαζί με το rank, τα οποία όσο περισσότερο συνεισφέρει κάποιος τόσο περισσότερο αυξάνονται. Τέλος παρέχονται πληροφορίες και για τις φωτογραφίες που έχει ανεβάσει ο χρήστης, η δυνατότητα διαγραφής και επεξεργασίας τους μαζί με τις αποστάσεις που έχει διανύσει χρησιμοποιώντας την εφαρμογή.



Εικόνα 2. 9 KartaView



Εικόνα 2. 10 KartaView



Εικόνα 2. 8 KartaView

2.5 Αρχικός σχεδιασμός της εφαρμογής Mapography

Σε αυτή την υπό-ενότητα θα δούμε κάποιες ιδέες καθώς και τον σχεδιασμό της εφαρμογής πριν αναπτυχθεί το demo του υπολογιστικού συστήματος. Ανοίγοντας την εφαρμογή για πρώτη φορά βρίσκομαστε στην σελίδα Εισόδου - Log-in (**εικόνα 2.11**). Εδώ ο χρήστης μπορεί να συνδεθεί στην εφαρμογή μέσω λογαριασμού ή να εισέλθει ως επισκέπτης (guest). Αν επιλέξει να συνδεθεί σαν επισκέπτης θα χάσει σχεδόν όλες τις βασικές δυνατότητες της εφαρμογής καθώς το μόνο που θα μπορεί να κάνει θα είναι να δει απλά φωτογραφίες σε διάφορες περιοχές αλλά δεν θα μπορεί να συνεισφέρει.

Στην **εικόνα 2.12** βλέπουμε το σχέδιο της αρχική σελίδας της εφαρμογής. Εδώ ο χρήστης έχει αρκετές επιλογές οι οποίες θα εξηγηθούν αναλυτικά. Αρχικά είτε έχει συνδεθεί με λογαριασμό είτε ως επισκέπτης (guest) θα μπορεί να δει στον χάρτη το σημείο που βρίσκεται και κάποια κοντινά points of interest. Επίσης έχει τη δυνατότητα να κάνει αναζήτηση πατώντας το αντίστοιχο κουμπί και να ψάξει κάποια τοποθεσία. Εφόσον έχει συνδεθεί ως επισκέπτης (guest) η εφαρμογή δεν του παρέχει παραπάνω δυνατότητες. Εάν όμως έχει συνδεθεί με κάποιο

λογαριασμό τότε ο χρήστης “ξεκλειδώνει” όλες τις δυνατότητες της εφαρμογής. Θα μπορεί πλέον πέρα από να πατήσει σε ένα point of interest και απλά να το δει, να κάνει μία κριτική για το πόσο ακριβής είναι οι φωτογραφίες και να γράψει ένα σχόλιο. Επίσης μία σημαντική δυνατότητα που δεν έχει κάποιος επισκέπτης (guest) είναι η δυνατότητα αποστολής μηνυμάτων (κουμπί κάτω δεξιά εικόνα 2.12). Τελευταία δυνατότητα που παρατηρείται στην εικόνα 2.12 είναι το κουμπί πάνω δεξιά το οποίο μας μεταφέρει στο προφίλ του χρήστη.

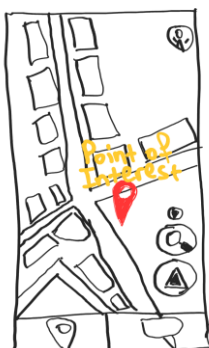
Στην εικόνα 2.13 παρουσιάζεται μία περιγραφή για το τι συμβαίνει εάν ο χρήστης πατήσει σε κάποιο point of interest. Μόλις γίνει αυτό, η εφαρμογή μας πηγαίνει σε μια καινούρια σελίδα η οποία περιέχει:

- Όνομα του σημείου (Statue)
- Φωτογραφία ή φωτογραφίες που έχουν ανέβει από τους άλλους χρήστες
- Όνομα και ημερομηνία του χρήστη που πρόσθεσε την/τις φωτογραφία/ες
- Αξιολόγηση από τους υπόλοιπους χρήστες της/των φωτογραφίας/ων
- Κουμπί για να αξιολογήσει ο χρήστης
- Σχόλια χρηστών
- Κουμπί αναζήτησης άλλου σημείου
- Κουμπί προσθήκης νέας/ων φωτογραφίας/ων ή προσθήκης σχόλιου

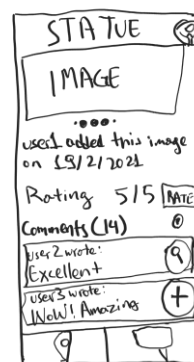
Με αυτόν τον τρόπο ο χρήστης μπορεί αν αλληλοεπιδράσει μαζί με τους υπόλοιπους και να “αποφασίσουν” όλοι μαζί πόσο χρήσιμες ή όχι είναι οι πληροφορίες που παρέχονται.



Εικόνα 2. 11 Αρχικά σχέδια



Εικόνα 2. 12 Αρχικά σχέδια



Εικόνα 2. 13 Αρχικά σχέδια

Πηγαίνοντας στην εικόνα 2.14 βλέπουμε τη λειτουργία της εφαρμογής όταν ο χρήστης θα προσπαθήσει να ανεβάσει ένα καινούριο point of interest. Η εφαρμογή θα παρέχει τη δυνατότητα να προστεθούν μία ή περισσότερες φωτογραφίες και από δίπλα ένα όνομα / περιγραφή. Κάτω από αυτές τις δύο επιλογές θα πρέπει να προστεθεί ή τοποθεσία του point of interest. Ύστερα ο χρήστης καλείται να δώσει μία αναλυτική περιγραφή για αυτό το σημείο. Ακολουθώντας τα παραπάνω βήματα και εφόσον ο χρήστης έχει τελειώσει μπορεί να προσθέσει το νέο σημείο πατώντας το κουμπί “ADD POST” ή εάν άλλαξε γνώμη υπάρχει το κουμπί “CANCEL” για να ακυρώσει την νέα προσθήκη.

Στην **εικόνα 2.15** έχουμε το σχέδιο για το chat. Το να έχει κάποιος τη δυνατότητα συνομιλίας με κάποιον άλλον είναι μία σημαντική δυνατότητα που παρέχεται σχεδόν από όλες τις κοινωνικές εφαρμογές. Ο κάθε εγγεγραμμένος χρήστης της εφαρμογής θα μπορεί να ανταλλάσσει μηνύματα και απόψεις τα οποία μπορεί να αφορούν είτε τοποθεσίες είτε φωτογραφίες. Επίσης θα μπορεί να αναζητήσει συνομιλίες αλλά και να δημιουργήσει κάποια καινούρια.

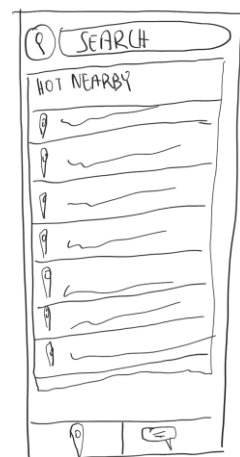
Το σχέδιο της **εικόνας 2.16** αφορά την αναζήτηση που μπορεί να κάνει ο χρήστης μέσα στην εφαρμογή. Πατώντας το κουμπί Search θα μπορεί ο κάθε χρήστης να δει και να αναζητήσει την τοποθεσία που επιθυμεί. Πριν κάνει κάποια αναζήτηση θα υπάρχουν και κάποιες προτάσεις οι οποίες θα είναι μοναδικές καθώς αφορούν την τοποθεσία του εκάστοτε χρήστη.



Εικόνα 2. 16 Αρχικά σχέδια



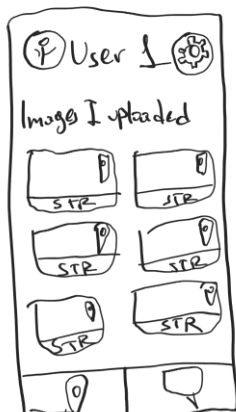
Εικόνα 2. 14 Αρχικά σχέδια



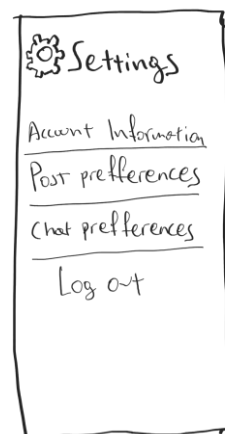
Εικόνα 2. 15 Αρχικά σχέδια

Το επόμενο σχέδιο που θα αναλυθεί είναι η **εικόνα 2.17**. Εδώ βλέπουμε πως θα φαίνεται το προφίλ του εγγεγραμμένου χρήστη. Στο επάνω μέρος της δραστηριότητας έχουμε την εικόνα προφίλ, το όνομα καθώς και ένα κουμπί που θα μεταβαίνει στις ρυθμίσεις της εφαρμογής. Ύστερα θα υπάρχει ένα preview το οποίο αφορά όλες τις φωτογραφίες που έχει ανεβάσει ο χρήστης μαζί με την αντίστοιχη τοποθεσία. Τέλος στο κάτω μέρος της εικόνας έχουμε το κουμπί που πηγαίνει στον χάρτη καθώς και το κουμπί που μεταφέρει τον χρήστη στα μηνύματά του.

Το τελευταίο σχέδιο που περιγράφει την εφαρμογή που θα αναπτυχθεί **Εικόνα 2.18** αφορά τις ρυθμίσεις του συστήματος. Σε αυτή τη δραστηριότητα ο χρήστης έχει τέσσερις διαφορετικές επιλογές. Η πρώτη επιλογή αφορά τις ρυθμίσεις του λογαριασμού. Εδώ θα υπάρχει η δυνατότητα αλλαγής προσωπικών στοιχείων όπως όνομα, κωδικός, email καθώς και κάποιες προτιμήσεις που αφορούν την ιδιωτικότητα του χρήστη. Στη δεύτερη επιλογή θα υπάρχουν ρυθμίσεις οι οποίες αφορούν τις προτιμήσεις που έχει ο χρήστης για το κάθε post που ανεβάζει. Στην τρίτη επιλογή υπάρχουν ρυθμίσεις προτιμήσεων που αφορούν το chat. Τέλος υπάρχει και η επιλογή αποσύνδεσης του χρήστη από την εφαρμογή.



Εικόνα 2. 18 Αρχικά σχέδια



Εικόνα 2. 17 Αρχικά σχέδια

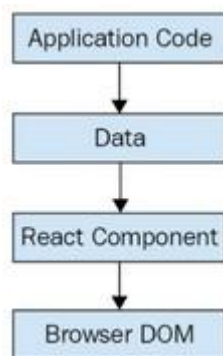
Κεφάλαιο 3. React, React Native και το λειτουργικό σύστημα Android

3.1 Εισαγωγή στις τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο θα δούμε τις κύριες τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη και την εγκατάσταση της εφαρμογής Marography. Οι τεχνολογίες αυτές είναι η React Native με την οποία αναπτύχθηκε η εφαρμογή και το λειτουργικό σύστημα Android στο οποίο εγκαταστάθηκε και έγιναν όλες οι απαραίτητες δοκιμές. Όμως για αρχή θα δούμε κάποιες βασικές λειτουργίες της React, καθώς η React Native αποτελεί επέκτασή της και οι περισσότερες λειτουργίες της προέρχονται από αυτή.

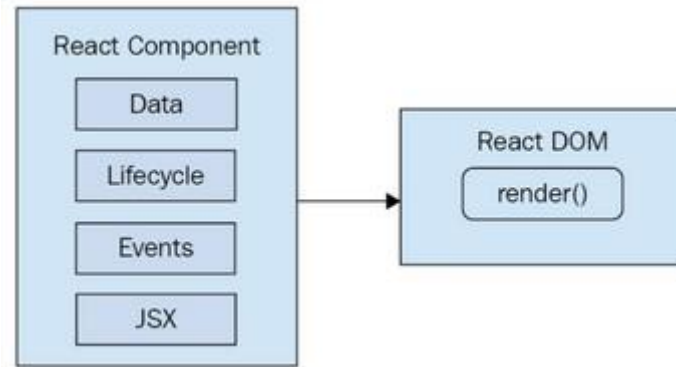
3.2 Εισαγωγή στη React

Η React είναι μία βιβλιοθήκη της γλώσσας προγραμματισμού JavaScript με την οποία μπορεί κάποιος να αναπτύξει διαδικτυακές εφαρμογές, ιστοσελίδες, εφαρμογές για κινητά και διεπαφές χρήστη. Κάνει χρήση βοηθητικών βιβλιοθηκών όπως η βιβλιοθήκη React Router και η βιβλιοθήκη React Redux, διότι επικεντρώνεται κυρίως στο να αποδίδει δεδομένα στο μοντέλο αντικείμενου εγγράφου (DOM). Η React δεν αποτελεί μία full-stack λύση η οποία θα μπορεί διαχειρίζεται τα πάντα από μία βάση δεδομένων μέχρι και την ενημέρωση σε πραγματικό χρόνο μέσω δικτύου. Γενικά η React θεωρείται ότι ανήκει στο επίπεδο της όψης (view) μίας εφαρμογής, και για να λειτουργήσει χρησιμοποιεί τα λεγόμενα στοιχεία-components τα οποία μπορούν να αλλάζουν το τί μπορεί να δει ένας χρήστης. Το παρακάτω διάγραμμα απεικονίζει το επίπεδο που βρίσκεται η React σε έναν front-end κώδικα:



Εικόνα 3. 1 React Component [19]

Έχουμε μία εφαρμογή η οποία δημιουργεί κάποια δεδομένα. Τα δεδομένα αυτά θέλουμε να τα αποδώσουμε σε μία διεπαφή χρήστη, οπότε τα περνάμε μέσα σε ένα React στοιχείο και αυτό με τη σειρά του διαχειρίζεται το τρόπο εμφάνισης τους σε μία σελίδα. Έτσι επιτυγχάνεται μια πιο εύκολη και ανώδυνη δημιουργία και υλοποίηση διαδραστικών διεπαφών χρήστη. Η React χωρίζεται σε δύο APIs. Το πρώτο είναι το React Dom το οποίο αναφέρθηκε και νωρίτερα και το δεύτερο είναι το React Component.



Εικόνα 3. 2 React Component & React Dom [19]

Όπως φαίνεται και στο παραπάνω σχήμα το React Dom αφορά την απόδοση των δεδομένων σε μία ιστοσελίδα ενώ το React Component αποτελείται από 4 υποκατηγορίες:

- **Data:** αφορούν δεδομένα τα οποία μπορούν να προέρχονται από οπουδήποτε.
- **Lifecycle:** Οι μέθοδοι κύκλου ζωής (Lifecycle) της React είναι από τα πιο σημαντικά της στοιχεία και από αυτά που την χαρακτηρίζουν, καθώς αυτές καθορίζουν την εκτέλεση κώδικα στα διάφορα σημεία της εφαρμογής μέσω ειδικών συναρτήσεων, όπως ComponentDidUpdate, Render, ComponentDidMount κ.λπ.
- **Events:** Τα γεγονότα (events) αντιπροσωπεύουν το κομμάτι κώδικα μιας εφαρμογής το οποίο είναι υπεύθυνο για την αλληλοεπίδρασή της με τον χρήστη.
- **JSX:** Το JSX αποτελεί την σύνταξη των React στοιχείων τα οποία περιγράφουν μία διεπαφή χρήστη.

Όπως γίνεται εύκολα αντιληπτό, η React βασίζεται σε μεγάλο βαθμό στα στοιχεία (components) αφού αυτά αποτελούν την βασικότερη δομή της. Με άλλα λόγια, μπορούμε να πούμε ότι κάθε εφαρμογή είτε κινητού είτε υπολογιστή η οποία θα αναπτυχθεί σε React αποτελείται από πολλά κομμάτια τα οποία ονομάζονται στοιχεία. Παρέχουν μεγάλη ευκολία στον προγραμματιστή γιατί του επιτρέπουν να χωρίσει τη διεπαφή που πρόκειται να υλοποιηθεί σε πολλά κομμάτια και ύστερα να τα ενώσει όλα για το τελικό αποτέλεσμα. Γενικά η βιβλιοθήκη React είναι αρκετά απλή και δεν χρειάζεται κάποιος που έχει ήδη καλές προγραμματιστικές γνώσεις να ξοδέψει πολύ χρόνο προκειμένου να τη κατανοήσει. Μόλις μάθει τις βασικές αρχές θα μπορέσει σε μικρό χρονικό διάστημα να τη χρησιμοποιήσει και να δημιουργήσει εφαρμογές. [19]

3.2.1 Virtual Dom

Μόλις μια ιστοσελίδα φορτωθεί σε ένα πρόγραμμα περιήγησης, δημιουργείται ένα μοντέλο αντικειμένου εγγράφου (Document Object Model) οποίο περιέχει αυτήν την ιστοσελίδα. Το έγγραφο DOM είναι μία αναπαράσταση σε μορφή δέντρου και εμφανίζει την κατάσταση και την δομή της τρέχουσας ιστοσελίδας χρησιμοποιώντας στοιχεία HTML. Όταν ο χρήστης εκτελέσει μία ενέργεια στην ιστοσελίδα, για παράδειγμα θέλει να μεταβεί σε κάποια άλλη σελίδα ή εφαρμογή, τότε η νέα σελίδα δημιουργείται εκ νέου ή το περιεχόμενο του DOM αρχείου χειρίζεται από τη JavaScript. Γενικά για να λάβουμε τον συγχρονισμό μεταξύ της κατάστασης της διεπαφής χρήστη (UI state) και της κατάστασης του μοντέλου δεδομένων (data

model) θα πρέπει να χρησιμοποιηθεί η λεγόμενη αμφίδρομη δέσμευση δεδομένων (two-way data binding). Αυτό επιτυγχάνεται με τη χρήση ενός κλειδιού-τιμής. Εδώ λοιπόν έρχεται η React και αντί να χρησιμοποιήσει την αμφίδρομη δέσμευση για τον συγχρονισμό χρησιμοποιεί ένα μοντέλο το οποίο ονομάζεται Virtual DOM. Το Virtual Dom αποδίδει (renders) επιλεκτικά κάποια υπό-δέντρα (sub-trees) των κόμβων που υπάρχουν στο DOM και το κριτήριο αυτής της επιλογής είναι η τρέχουσα κατάσταση. Αυτό έχει ως αποτέλεσμα να απαιτείται ελάχιστος χειρισμός προκειμένου μία σελίδα να παραμένει ενημερωμένη. Το Virtual DOM είναι στην ουσία μία αναπαράσταση του κανονικού DOM με μόνη διαφορά το ότι αντιμετωπίζει τη JavaScript και το DOM σαν να είναι αντιδραστικά (reactive). Η React αποθηκεύει την κατάσταση της εφαρμογής εσωτερικά και θα εκτελέσει το έγγραφο DOM μόνο όταν η τρέχουσα κατάσταση αλλάξει. Όταν η κατάσταση του μοντέλου δεδομένων (data model state) αλλάξει, η React και το Virtual DOM θα ξανά αποδώσουν (re-render) τη διεπαφή χρήστη ως μία αναπαράσταση του Virtual DOM. Μόλις συμβεί αυτό η React θα συγκρίνει τα δύο Virtual DOM με στόχο να βρει τις διαφορές μεταξύ τους και να κάνει τις απαραίτητες αλλαγές. Αυτές οι αλλαγές μετά ενημερώνουν και αποδίδονται στο πραγματικό αρχείο DOM. Αυτή η διαδικασία ονομάζεται συμφιλίωση (reconciliation) και οφείλεται στη δημιουργία στοιχείων (components). Την πρώτη φορά που θα δημιουργήσουμε ένα στοιχείο, καλείται η μέθοδος απόδοσης (render method) και δημιουργείται ένα string μορφής markup το οποίο εισάγεται στο αρχείο DOM. Όταν αλλάξει κάποιο από τα δεδομένα τότε η μέθοδος της απόδοσης (render method) καλείται ξανά και οι διαφορές της τιμής από την προηγούμενη και την νέα κλήση είναι το ελάχιστο σύνολο αλλαγών που πρέπει να γίνουν. [19]

3.2.2 JSX

Για τη δημιουργία ενός στοιχείου σε μία εφαρμογή της React μπορούμε είτε να χρησιμοποιήσουμε απλή JavaScript, είτε χρησιμοποιώντας JSX. Το JSX αποτελεί μία επέκταση της JavaScript και μπορεί να κάνει αναπαράσταση αντικειμένων τα οποία βασίζονται στην XML. Επίσης έχει το πλεονέκτημα της δημιουργίας μεγάλων δέντρων τα οποία είναι ευανάγνωστα. Επιπλέον με τη χρήση της React μπορούμε να δημιουργήσουμε ετικέτες HTML ως συμβολοσειρές ή να δημιουργήσουμε React στοιχεία με τη μορφή κλάσης. [24]

3.3 Ανάπτυξη Cross-Platform εφαρμογών

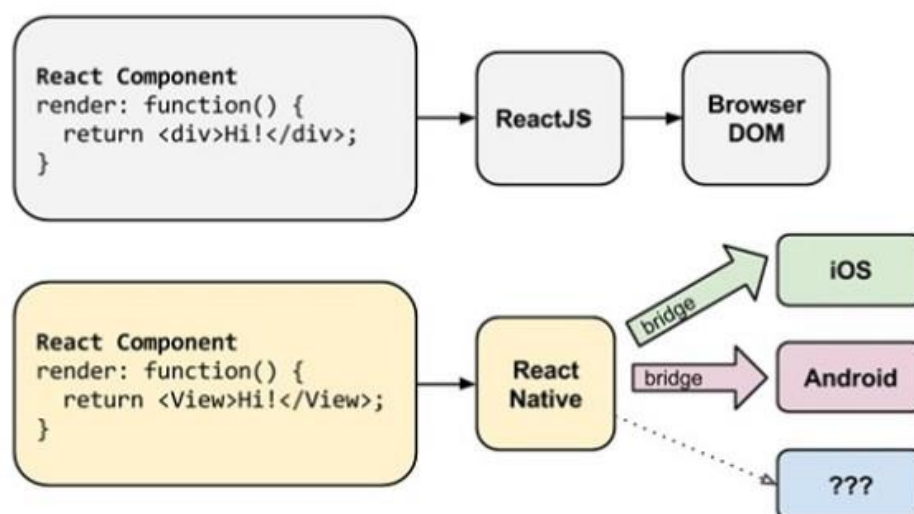
Τα τελευταία χρόνια, υπάρχει μεγάλη αύξηση βιβλιοθηκών αλλά και τεχνικών οι οποίες έρχονται να δώσουν τη λύση στο πρόβλημα της ανάπτυξης μίας Cross-Platform εφαρμογής. Ο στόχος είναι η ανάπτυξη μίας υπηρεσίας, μία μόνο φορά, η οποία θα είναι προσπελάσιμη, για διαφορετικά λειτουργικά συστήματα και θα παρέχει μία ομοιογενή και εγγενή αίσθηση.

Μία δημοφιλής τεχνική για να γίνει αυτό είναι να δημιουργήσουμε μία ιστοσελίδα, η οποία θα είναι προσβάσιμη και μέσω του Browser των κινητών συσκευών. Χρησιμοποιώντας για παράδειγμα HTML, CSS3 σε συνδυασμό με τη βιβλιοθήκη Bootstrap θα μπορούσε κάποιος να δημιουργήσει διεπαφές Χρήστη οι οποίες είναι εύκολα προσβάσιμες και από υπολογιστή αλλά και από κινητές συσκευές. Με αυτόν τον τρόπο η διεπαφή χρήστη θα φαίνεται όπως μία κανονική ιστοσελίδα όταν ο χρήστης χρησιμοποιεί έναν υπολογιστή και όταν χρησιμοποιεί κάποια κινητή συσκευή θα μοιάζει με τις εγγενείς εφαρμογές που υπάρχουν για τις συσκευές αυτές.

Υπάρχει όμως και μία εναλλακτική λύση με την οποία μπορούμε να δημιουργήσουμε εφαρμογές που χρησιμοποιούνται από διαφορετικά λειτουργικά συστήματα και αυτή η εναλλακτική αφορά τις υβριδικές βιβλιοθήκες (Hybrid Frameworks). Αυτή η προσέγγιση συνδυάζει την ανάπτυξη ιστοσελίδων με την ανάπτυξη εγγενών εφαρμογών. Η εφαρμογή που θα δημιουργηθεί με τη χρήση μίας υβριδικής βιβλιοθήκης, κατά τη δημιουργία της χρησιμοποιεί τεχνικές ίδιες με αυτές των ιστοσελίδων, αλλά εκτελείται, αποδίδεται (rendered) και εμφανίζεται ως εγγενής εφαρμογή. Στη συνέχεια για να μπορεί η εφαρμογή αυτή να έχει πρόσβαση στις λειτουργίες και τα χαρακτηριστικά της συσκευής, γίνεται η χρήση διεπαφών προγραμματισμού εφαρμογών (APIs) της γλώσσας JavaScript. Ωστόσο, αν και το κόστος ανάπτυξης υβριδικών εφαρμογών είναι σημαντικά χαμηλότερο σε σχέση με την ανάπτυξη εγγενών εφαρμογών, οι υβριδικές εφαρμογές δεν καταφέρνουν να προσφέρουν την ίδια εμπειρία χρήσης με αυτή των εγγενών. Στη παρακάτω ενότητα παρουσιάζεται ή React Native η οποία είναι μία υβριδική βιβλιοθήκη με την οποία μπορούμε να αναπτύξουμε εφαρμογές για κινητές συσκευές.

3.3.1 React Native

Η React Native παρουσιάστηκε το 2015 από το Facebook. Είναι μία open source επέκταση της βιβλιοθήκης React και είναι υπεύθυνη για την ανάπτυξη πραγματικών εγγενών εφαρμογών (Native apps) για κινητές συσκευές. Όταν πρωτοπαρουσιάστηκε υπήρχε μόνο υποστήριξη για κινητές συσκευές iOS αλλά αργότερα προστέθηκε και η υποστήριξη για Android συσκευές. Χάρη στη React Native οι προγραμματιστές ιστού (web developers) οι οποίοι ήδη ξέρουν να προγραμματίζουν με την χρήση της React, μπορούν πλέον να αναπτύξουν και εφαρμογές για κινητές συσκευές σχεδόν χωρίς περαιτέρω γνώσεις. Οι εφαρμογές που αναπτύσσονται μέσω της React Native δίνουν την αίσθηση μιας πραγματικά εγγενούς εφαρμογής χωρίς να χρειάζεται κάποιος να γράψει κώδικα δύο φορές μία σε java ή kotlin για το λειτουργικό Android και μία σε objective C ή Swift για το iOS. Οι εφαρμογές της React Native γράφονται συνδυάζοντας την γλώσσα προγραμματισμού JavaScript και JSX. Στη συνέχεια λειτουργεί ως μία γέφυρα, αποδίδοντας APIs σε Objective-C (για iOS) ή Java (για Android) (**εικόνα 3.3**).



Εικόνα 3. 3 Rendering in React and React Native [24]

Με αυτό τον τρόπο η εφαρμογή που δημιουργείται απεικονίζεται με πραγματικά στοιχεία διεπαφής (UI Elements) στα κινητά και όχι σαν μία προβολή ιστού. Στο παρελθόν έχουν γίνει αρκετές προσπάθειες για τη δημιουργία ενός υβριδικού πλαισίου με το οποίο θα δημιουργούνται εφαρμογές για διαφορετικές πλατφόρμες. Ωστόσο κανένα δεν κατάφερε να παρέχει το σύνολο δυνατοτήτων που προσφέρει η React Native. Για αυτό και από την κυκλοφορία της React Native μέχρι και σήμερα υπάρχει τεράστια υποστήριξη από εταιρίες και οργανισμούς ανάπτυξης εφαρμογών οι οποίοι την έχουν ως πρώτη επιλογή για την ανάπτυξη των εφαρμογών τους. [20]

3.3.2 Ο πυρήνας της React Native

Όπως αναφέρθηκε και νωρίτερα η React Native λειτουργεί ως γέφυρα και έχει την ικανότητα να αποδίδει (render) τα React Native στοιχεία σε πραγματικές εγγενείς όψεις (Views) για το Android και προβολές διεπαφής χρήστη (UI Views) για το iOS. Επιπλέον δίνει την δυνατότητα απόκτησης πρόσβασης σε ορισμένες platform-specific λειτουργίες όπως είναι οι πληροφορίες για τη μπαταρία ή την τοποθεσία της συσκευής. Η React Native βασίζεται σε νήματα (threads) και τα τρία πιο βασικά είναι:

1. **The shadow queue thread:** Το νήμα αυτό διαχειρίζεται τη διάταξη της εφαρμογής.
2. **The main thread:** Αυτό το νήμα είναι υπεύθυνο για την σωστή απόδοση της διεπαφής χρήστη.
3. **The JavaScript thread:** Σε αυτό το νήμα εκτελούνται τα διάφορα scripts.

3.3.3 Πλεονεκτήματα React Native

- **Εφαρμογές με μεγαλύτερη ταχύτητα:** Όπως αναφέρθηκε και νωρίτερα οι εφαρμογές που αναπτύσσονται με τη χρήση της React Native μοιάζουν σε συμπεριφορά αλλά και στη διεπαφή χρήστη (User-Interface) με τις εγγενείς εφαρμογές (Native Apps) λόγω του ότι χρησιμοποιούνται έτοιμα εξειδικευμένα μπλοκ κώδικα για κάθε πλατφόρμα στην οποία θέλουμε να υποστηρίζεται η εφαρμογή. Όταν λοιπόν δεν μας νοιάζει τόσο η πολυπλοκότητα μίας εφαρμογής, αλλά μας ενδιαφέρει η ταχύτητά της είναι προτιμότερο να την αναπτύξουμε με τη χρήση της React Native.
- **Εύκολη επαναχρησιμοποίηση του κώδικα:** Το να μπορείς να επαναχρησιμοποιήσεις τμήματα κώδικα είναι ένα από τα βασικότερα πλεονεκτήματα που προσφέρει η React Native. Υπάρχουν αρκετές αναφορές οι οποίες κάνουν λόγο για ποσοστό επαναχρησιμοποίησης έως και 95%, κάτι που αδιαμφισβήτητα επιταχύνει τη διαδικασία ανάπτυξης και ταυτόχρονα μειώνει το κόστος υλοποίησης της.
- **Λιγότερος χρόνος για την υλοποίηση της εφαρμογής:** Η παράλληλη παρουσία σε διαφορετικές πλατφόρμες χωρίς διπλότυπο κώδικα μαζί με την ευκολία ανάπτυξης μειώνουν σημαντικά τον χρόνο που χρειάζεται για την ανάπτυξη μίας εφαρμογής, σε σύγκριση με την ανάπτυξη μιας εγγενούς εφαρμογής (Native App). Εξίσου σημαντική όμως είναι και η κοινότητα των προγραμματιστών React Native καθώς υπάρχει τεράστια βοήθεια και υποστήριξη σε φόρουμ από μεριάς τους. Επίσης έχουν αναπτυχθεί πολλά πρόσθετα (plugins) τα οποία είναι πολλές φορές χρήσιμα και συμβάλλουν και αυτά στη μείωση του χρόνου.

- **Συμβατότητα με κώδικα εγγενών εφαρμογών:** Η React Native μεταφράζει τον κώδικα που έχουμε γράψει σε εγγενή στοιχεία διεπαφής χρήστη (Native UI elements) και αυτό επιτρέπει την συμβατότητα με τον κώδικα των Native εφαρμογών. Είναι επίσης πιθανό να χρησιμοποιηθούν React Native στοιχεία (Components) σε εγγενείς εφαρμογές (Native Apps) προκειμένου να προστεθούν συγκεκριμένες λειτουργίες και για τις δύο πλατφόρμες (Android και IOS) ταυτόχρονα. Αυτό συμβαίνει στην περίπτωση του Facebook και του Instagram.
- **Εύκολη εκμάθηση από έμπειρους προγραμματιστές:** Η διαδικασία ανάπτυξης μίας React Native εφαρμογής είναι σχετικά εύκολη και απλή. Για να προχωρήσει στην ανάπτυξη της ένας προγραμματιστής το μόνο που χρειάζεται είναι να έχει εμπειρία στη σύνταξη κώδικα JavaScript, και, σίγουρα, να γνωρίζει αν όχι καλά τουλάχιστον τις βασικές αρχές της React. Σε κάθε περίπτωση πάντως, για να παρουσιαστεί ένα ποιοτικό Android ή iOS προϊόν, απαιτείται η ύπαρξη προγραμματιστικής εμπειρίας. Τόσο η μία όσο και η άλλη πλατφόρμα έχουν αρκετές ομοιότητες, υπάρχουν όμως κάποια ειδικά χαρακτηριστικά τα οποία ανεβάζουν τον κίνδυνο λάθους κατά την ανάπτυξη. Με λίγα λόγια πρέπει ο προγραμματιστής να είναι σε θέση να αντιληφθεί αυτές τις παγίδες ώστε αν χρειαστεί να μπορέσει να τις αντιμετωπίσει. [20][21][22]

3.3.4 Μειονεκτήματα της React Native

- **Περισσότερος κόπος για τις Android εφαρμογές:** Επειδή η React Native στην αρχή δημιουργήθηκε για την πλατφόρμα iOS υπάρχει μεγάλη πιθανότητα ορισμένα στοιχεία (components) που αφορούν την Android πλατφόρμα να χρειάζονται περισσότερη προσπάθεια ώστε να μπορέσουν να λειτουργήσουν σωστά και εν τέλη η εφαρμογή που σχεδιάζουμε να μην έχει προβλήματα.
- **Μειωμένη υποστήριξη πολύπλοκων, εξειδικευμένων ανά πλατφόρμα, λειτουργιών:** Η συγκεκριμένη τεχνολογία μοιάζει να ταιριάζει περισσότερο στην ανάπτυξη απλών εφαρμογών και χάνει το ανταγωνιστικό της πλεονέκτημα όταν απαιτείται υλοποίηση σύνθετων λειτουργιών, οι οποίες πρέπει να είναι ποιο εξειδικευμένες για την κάθε πλατφόρμα.
- **Απαιτείται προγραμματιστική εμπειρία σε εγγενείς εφαρμογές:** Είναι πολύ πιθανό κατά τη διάρκεια ανάπτυξης μίας εφαρμογής να χρειαστούν από τον προγραμματιστή δεξιότητες και εμπειρία πάνω σε στοιχεία εγγενών εφαρμογών (Native Apps Components). Μπορεί δηλαδή να χρειαστεί ο προγραμματιστής να έχει γνώσεις και από τις γλώσσες προγραμματισμού Objective C / Swift ή Java / Kotlin ώστε να είναι σε θέση να συνεχίσει την ανάπτυξη της εφαρμογής.
- **Η εκμάθηση της React ίσως απαιτήσει αρκετό χρόνο εξοικείωσης:** Είναι γεγονός ότι όταν πρόκειται να μάθει κάποιος μία νέα τεχνολογία αναλόγως την εμπειρία και τις γνώσεις που έχει, ίσως χρειαστεί αρκετός χρόνος για την εξοικείωση του με αυτή τη τεχνολογία. Έτσι λοιπόν και με την React μπορεί να χρειαστεί αρκετός χρόνος ώστε να μπορέσει κανείς να είναι σε θέση να αναπτύξει μία εφαρμογή.

- **Χαμηλότερη απόκριση εφαρμογών (συγκριτικά με τις εγγενείς εφαρμογές):** Τις περισσότερες φορές η απόδοση μίας εφαρμογής που αναπτύχθηκε με την χρήση της React Native παραμένει χαμηλότερη από την απόδοση που θα είχε μία εγγενής εφαρμογή (Native App). Αυτό συμβαίνει διότι όπως αναφέρθηκε και πιο πριν απαιτείται η διαδικασία γεφύρωσης των React Native στοιχείων (components) με τα αντίστοιχα εγγενή στοιχεία (Native components) του συστήματος. [20][21][22]

3.3.5 Τα στοιχεία (components) της React Native

Τα στοιχεία (components) αποτελούν τη βασική δομή μίας React Native εφαρμογής και είναι παρόμοια με τις συναρτήσεις (functions) τις οποίες συναντάμε σε άλλες γλώσσες προγραμματισμού. Για να μπορέσει να λειτουργήσει ένα στοιχείο ενδέχεται να απαιτεί τη λειτουργία άλλων στοιχείων ή μπορεί να είναι αυτόνομο και να αντιπροσωπεύει ένα μέρος της εφαρμογής. Τα στοιχεία της React Native χωρίζονται σε δύο κατηγορίες, στα στοιχεία παρουσίασης (presentational components) και στα στοιχεία περιεχομένου (container components ή και stateful components). Αν και αυτές οι κατηγορίες έχουν τα δικά τους χαρακτηριστικά, η κύρια διαφορά τους είναι ότι τα στοιχεία παρουσίασης δεν συνδέονται με κάποιο κύκλο ζωής (lifecycle) και δεν έχουν κάποια δική τους κατάσταση (state). Τα δεδομένα στα στοιχεία παρουσίασης ρέουν μέσω των React properties (props). Επομένως η χρήση ενός στοιχείου παρουσίασης γίνεται αποκλειστικά για την παρουσίαση περιεχομένου στον χρήστη. Παρακάτω βλέπουμε κάποια χαρακτηριστικά των στοιχείων παρουσίασης:

- Η χρήση τους γίνεται για την παρουσίαση δεδομένων και περιεχομένου στον χρήστη
- Μπορούν στο εσωτερικό τους να περιέχουν και άλλα στοιχεία παρουσίασης ή στοιχεία περιεχομένου
- Είναι ανεξάρτητα από την υπόλοιπη εφαρμογή
- Δέχονται δεδομένα μέσα από React props
- Δεν έχουν κάποια δική τους κατάσταση δεδομένων, μπορεί όμως να έχουν κατάσταση διεπαφής χρήστη
- Μπορούν εύκολα να επαναχρησιμοποιηθούν

Από την άλλη τα στοιχεία περιεχομένου συνδέονται με αποθήκες δεδομένων και ενέργειες οι οποίες μπορεί να έχουν δημιουργηθεί από τα ίδια στοιχεία, επηρεάζοντας την κατάσταση των δεδομένων. Τα παρακάτω χαρακτηριστικά αφορούν τα στοιχεία περιεχομένου:

- Είναι υπεύθυνα για τη λειτουργία της εφαρμογής
- Μπορούν και αυτά να αποτελούνται είτε από στοιχεία παρουσίασης είτε από στοιχεία περιεχομένου
- Είναι υπεύθυνα για την συμπεριφορά των υπόλοιπων στοιχείων παρουσίασης ή περιεχομένου.
- Συνήθως βρίσκονται σε κάποια κατάσταση και χρησιμεύουν σαν πηγές δεδομένων.
- Συνδέονται απευθείας με την κατάσταση των δεδομένων μιας εφαρμογής.

3.3.6 Διαχείριση κατάστασης στη React Native

Η κατάσταση (state) σε μία εφαρμογή React Native μπορεί να οριστεί ως ένα σύνολο τιμών το οποίο καταναλώνεται και διαχειρίζεται από ένα στοιχείο (component). Η κατάσταση

χρησιμοποιείται για τη δημιουργία και τη διαχείριση δεδομένων όλων των στοιχείων μίας React Native εφαρμογής. Η κατάσταση ενός στοιχείου είναι στην ουσία ένα JavaScript αντικείμενο το οποίο ορίζεται κατά τη δημιουργία του. Για να ενημερώσουμε την κατάσταση ενός στοιχείου θα πρέπει να χρησιμοποιήσουμε μία συνάρτηση που μας παρέχει η React και ονομάζεται **setState()**. Οι πρόσφατες εκδόσεις της React επιτρέπουν στους προγραμματιστές να καθορίζουν δικές τους παραμέτρους για την κατάσταση ενός στοιχείου με την χρήση του React Hooks. Ωστόσο η βασική ιδέα για το πως γίνεται η ενημέρωση μίας κατάστασης παραμένει ίδια. Παρόλο που η κατάσταση(state) χρησιμοποιείται κυρίως για τον χειρισμό δεδομένων μίας εφαρμογής στα στοιχεία περιεχομένου (container components), είναι επίσης πιθανή και η χρήση της για την διαχείριση δεδομένων που αφορούν τα στοιχεία παρουσίασης (presentation components). Αυτό επιτυγχάνεται με τη χρήση των React props. Τα props μπορούμε να τα περάσουμε κατά τη δημιουργία ενός στοιχείου ως μεταβλητές (parameters). Σε αντίθεση με την κατάσταση (state) τα props είναι μη μεταβλητά και δεν μπορούν να αλλάξουν. Όταν η κατάσταση ενός στοιχείου αλλάζει, τότε το στοιχείο αυτό ξανά αποδίδεται (re-renders) μαζί με όλα τα άλλα στοιχεία που περιέχουν την κατάσταση του στοιχείου ως props. Ο παρακάτω πίνακας εξηγεί τη συμπεριφορά της κατάστασης (state) και των props που έχουν τα στοιχεία (components) στη React Native.

State	Props
Internal data	external data
Mutable	Immutable
Created in the component	Inherited from a parent
Can only be updated in the component	Can be updated by parent component
Can be passed down as props	Can be passed down as props
Can set default values inside Component	Can set default values inside Component

Εικόνα 3. 4 State & Props [23]

Η κατάσταση (state) ενός στοιχείου και τα props που περνάμε σε αυτό, είναι σε θέση να διαχειριστούν δεδομένα για απλές React Native εφαρμογές. Εάν όμως έχουμε να κάνουμε με ένα πιο περίπλοκο σύστημα τότε η διαχείριση των δεδομένων δυσκολεύει. Μία από τις πιο δημοφιλής third-party βιβλιοθήκες που χρησιμοποιεί η React Native είναι το Redux. Σύμφωνα με τους δημιουργούς του, το Redux είναι μία προβλέψιμη κατάσταση περιεχομένου που αφορά εφαρμογές JavaScript. Είναι ένα global αντικείμενο κατάστασης (state object) το οποίο χρησιμοποιείται σαν μοναδική πηγή δεδομένων σε μία εφαρμογή και μπορούμε να το μεταφέρουμε ανάμεσα στα διάφορα React Native στοιχεία (components) μέσω props. Το Redux περιέχει όλες τις καταστάσεις της εφαρμογής μέσα σε ένα μέρος το οποίο ονομάζεται Redux Store. Τη στιγμή που το Redux καταλάβει ότι κάποια δεδομένα έχουν αλλάξει, θα στείλει τα καινούρια δεδομένα στην εφαρμογή με την χρήση props. Οι τρεις κύριοι άξονες της βιβλιοθήκης Redux είναι οι Reducers, το Store και τα Actions.

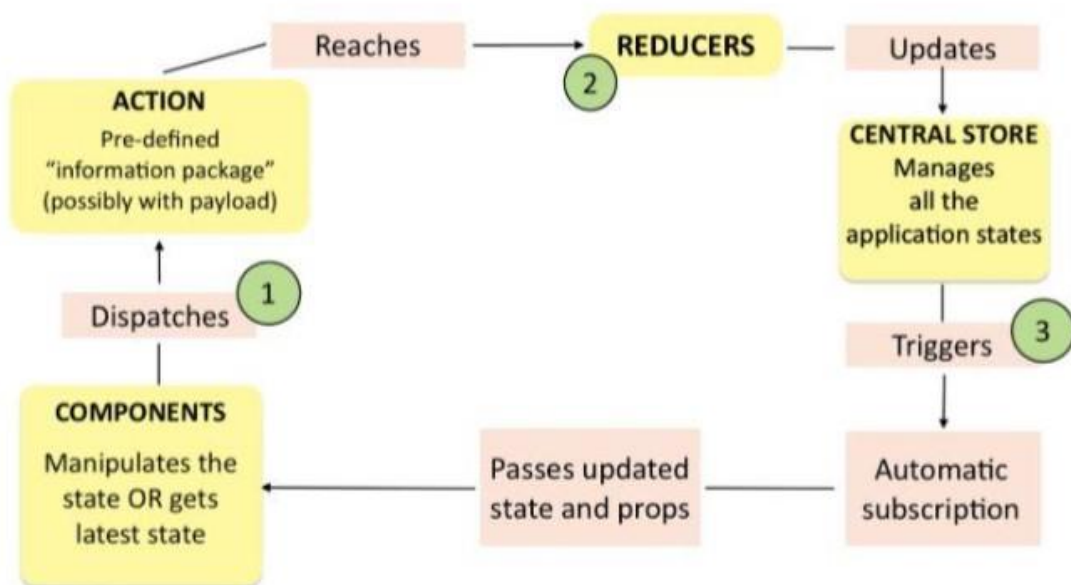
- **Reducers:** Οι Reducers είναι συναρτήσεις οι οποίες επιστρέφουν ένα αντικείμενο. Δημιουργούν μία παγκόσμια κατάσταση (global state) όταν συνδυάζονται μεταξύ τους.

Είναι υπεύθυνοι για τον τρόπο που γίνεται η αλλαγή κατάστασης της εφαρμογής. Μπορούν επίσης να θεωρηθούν και αποθήκες δεδομένων.

- **Store:** Το Redux store είναι ένα αντικείμενο το οποίο ενώνει τους Reducers και τα Actions. Επίσης είναι υπεύθυνο για την διατήρηση της κατάστασης της εφαρμογής και επιτρέπει την πρόσβαση και τις ενημερώσεις σε αυτή. Επιπλέον επιτρέπουν στους ακροατές (listeners) την εγγραφή (subscribe) και την απεγγραφή (unsubscribe).
- **Actions:** Τα Actions είναι και αυτά συναρτήσεις οι οποίες είναι υπεύθυνες για την ενημέρωση των Reducers, επιστρέφοντας τους ένα αντικείμενο το οποίο στέλνει δεδομένα στο Store. Η χρήση των Actions είναι ο μόνος τρόπος για την προσπέλαση των δεδομένων που υπάρχουν στο Redux store.

Στο παρακάτω σχήμα θα δούμε πως η βιβλιοθήκη Redux χειρίζεται και διαχειρίζεται την κατάσταση (state) μίας React Native εφαρμογής. Οι σημαντικότερες διαδικασίες του σχήματος εξηγούνται ως εξής:

- Ο μόνος τρόπος για να χειριστούμε μία Redux κατάσταση είναι στέλνοντας ένα action. Τα actions περνάνε μηνύματα στο store με σκοπό την ευκολότερη ενημέρωση των δεδομένων.
- Μόλις λάβουμε ένα action, ο reducer ενημερώνει το Redux state. Μία εφαρμογή μπορεί να έχει πολλαπλούς reducers.
- Όταν το Redux store λαμβάνει μία ενημέρωση, την περνάει μέσα σε όλα τα React Native στοιχεία τα οποία αφορά αυτή η ενημέρωση. Ένα στοιχείο μπορεί να λαμβάνει ενημερώσεις από το store μόνο εάν έχει εγγραφεί σε αυτό.



Εικόνα 3. 5 Action & Reducers [23]

Παρόλο που η React Native καταφέρνει να δημιουργήσει εφαρμογές για κινητές συσκευές διαφορετικής πλατφόρμας, είναι σημαντικό να καταλάβουμε ότι εξαρτάται πολλές φορές από third-party βιβλιοθήκες προκειμένου να υλοποιηθούν κάποιες λειτουργίες μίας εφαρμογής, όπως: σύνθετη διαχείριση κατάστασης (complex state management) και πλοήγηση

(navigation). Η React Native από μόνη της διαθέτει ένα μικρό σύνολο pre-built στοιχείων, επομένως η ολοκληρωμένη ανάπτυξη εφαρμογής εξαρτάται πολύ από third-party βιβλιοθήκες. Απαιτεί επίσης και κάποιες γνώσεις από τις τεχνολογίες ιστού που χρησιμοποιεί η βιβλιοθήκη React. Πάρα της προκλήσεις αυτές, η React Native συνεχώς βελτιώνεται και υιοθετεί νέες και πιο αποτελεσματικές αλλαγές. Αδιαμφισβήτητα με τη χρήση της React Native μπορούμε να έχουμε cross-platform εφαρμογές οι οποίες μοιάζουν με τις εγγενείς εφαρμογές τόσο σε ταχύτητα όσο και σε απόδοση. [23]

3.4 Οικοσύστημα Android

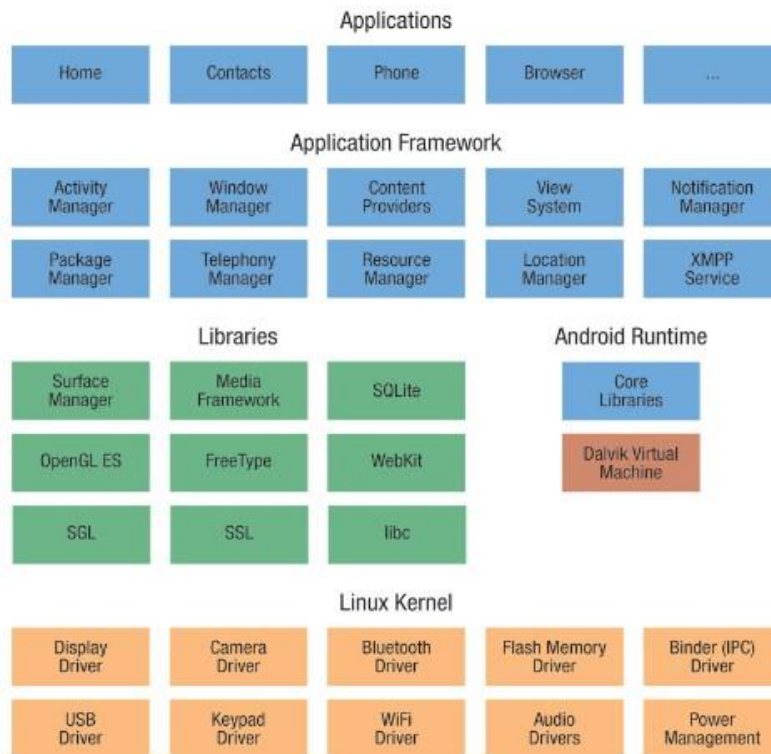
Το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα, βασισμένο σε έναν τροποποιημένο πυρήνα Linux και ανήκει στην Google. Η Google είναι υπεύθυνη για την ανάπτυξη του μαζί με το Open Hand-set Alliance (OHA). Το 2005 όταν πρωτοξεκίνησε η ανάπτυξη του Android, το λογισμικό προοριζόταν να χρησιμοποιηθεί σε ψηφιακές κάμερες. Ωστόσο, η Google αποφάσισε να αγοράσει την εταιρεία και τρία χρόνια αργότερα πωλήθηκε το πρώτο smartphone με Android. Το 2015, σχεδόν το 83% της αγοράς κινητής τηλεφωνίας χρησιμοποιεί Android και περίπου 1,5 δισεκατομμύριο εφαρμογές κατεβαίνουν από χρήστες Android κάθε μήνα. [23] Οι διάφοροι κατασκευαστές κινητών συσκευών μπορούν να προσαρμόζουν το λειτουργικό σύστημα Android στις συσκευές τους ανάλογα με τις απαιτήσεις τους. Το λειτουργικό σύστημα Android αποτελείται από τρεις κύριες κατηγορίες:

1. Ένα δωρεάν και ανοιχτού κώδικα λειτουργικό σύστημα
2. Μια ανοιχτού κώδικα πλατφόρμα ανάπτυξης με την οποία μπορούμε να δημιουργήσουμε λογισμικό
3. Τις συσκευές που χρησιμοποιούν το Android για να λειτουργήσουν, μαζί με τις εφαρμογές που έχουν δημιουργηθεί για αυτό.

Το Android μπορεί να θεωρηθεί ως ένα πρωτοπόρο και μιας νέας γενιάς λειτουργικό σύστημα για κινητές συσκευές που σχεδιάζεται αποκλειστικά για την υποστήριξη ανάπτυξης εφαρμογών σε ένα όλο και ισχυρότερο hardware κινητών συσκευών. Ένα σημαντικό ζήτημα που υπάρχει στο Android είναι η παροχή γρήγορης και αποτελεσματικής αναβάθμισης του λογισμικού. Αυτό συγκριτικά με τον κύριο ανταγωνιστή iOS φαίνεται να αποτελεί σημαντικό μειονέκτημα. Ο βασικός λόγος αυτού του ζητήματος είναι ότι υπάρχουν πάνω από 24 χιλιάδες συσκευές όλων των ειδών (smartphones, smart watches, smart TVs, IoT devices κλπ.) τις οποίες κατασκευάζουν πάνω από 1.300 κατασκευαστές! [7] Επίσης αυτοί οι κατασκευαστές τις περισσότερες φορές τροποποιούν το λογισμικό φτιάχνοντας δικές τους εκδόσεις και αυτό έχει ως αποτέλεσμα οι ενημερώσεις να αργούν ακόμα περισσότερο. [23]

3.4.1 Αρχιτεκτονική Android

Το λειτουργικό σύστημα Android είναι μία στοίβα από στοιχεία λογισμικού (software components). Όπως βλέπουμε και στο παρακάτω σχήμα η αρχιτεκτονική του λειτουργικού συστήματος αποτελείται από 4 επίπεδα: τον πυρήνα (Kernel) Linux, τις βιβλιοθήκες (Libraries) και τον χρόνο εκτέλεσης (Android Runtime), το πλαίσιο εφαρμογών (Application Framework) και τις εφαρμογές (Applications).



Εικόνα 3. 6 Αρχιτεκτονική του λειτουργικού συστήματος Android [24]

3.4.1.1 Linux Kernel

Το πρώτο επίπεδο για το οποίο θα μιλήσουμε είναι ο πυρήνας του Linux, το οποίο είναι και το βασικό επίπεδο. Το Android βασίζεται στον πυρήνα Linux 2.6 (και στην έκδοση 3.x για το Android 4.0 και πάνω) ο οποίος έχει τροποποιηθεί ώστε να λειτουργεί σε μία κινητή συσκευή. Παρόμοια με τα υπόλοιπα λειτουργικά συστήματα εξυπηρετεί λειτουργίες όπως: εσωτερική αποθήκευση, πρωτόκολλο διαδικτύου, διαχείριση διεργασιών, διαχείριση συσκευών κλπ. Ο πυρήνας περιέχει τα απαραίτητα προγράμματα οδήγησης υλικού(drivers) ώστε να μπορεί να αλληλοεπιδρά με το υλικό. Οι drivers αποτελούν προγράμματα τα οποία ελέγχουν και επικοινωνούν με το υλικό (Hardware) μιας συσκευής. Για να κατανοήσουμε καλύτερα τι είναι ένα πρόγραμμα οδήγησης μπορούμε να σκεφτούμε τη λειτουργία του Wi-Fi στα κινητά. Όλες οι συσκευές έχουν το απαραίτητο υλικό ώστε να μπορούν να συνδέονται σε ασύρματα δίκτυα. Για να λειτουργήσει όμως σωστά αυτό το υλικό θα πρέπει ο πυρήνας να περιέχει ένα πρόγραμμα οδήγησης το οποίο επικοινωνεί με το υλικό. Εν κατακλείδι σε αυτό το επίπεδο το Android αλληλοεπιδρά με το υλικό των συσκευών και οι λειτουργικές μονάδες (modules) και τα προγράμματα οδήγησης είναι γραμμένα σε γλώσσα προγραμματισμού C.

3.4.1.2 Βιβλιοθήκες και χρόνος εκτέλεσης Android

Πάνω από το πυρήνα Linux βρίσκονται οι εγγενείς βιβλιοθήκες και η λειτουργική μονάδα που αφορά τον χρόνο εκτέλεσης Android. Οι εγγενείς βιβλιοθήκες χρησιμοποιούνται από τις εφαρμογές και είναι το μέσο που συνδέει τις εφαρμογές με τις δυνατότητες που προσφέρει το

Android. Οι βιβλιοθήκες αυτές είναι γραμμένες σε γλώσσα C ή C++ και καλούνται μέσω μίας Java διεπαφής η οποία επιτρέπει την πρόσβαση σε διαφορετικά είδη δεδομένων ή ήχου τις οποίες παρέχει μία κινητή συσκευή. Παρακάτω θα δούμε κάποιες από τις βασικές βιβλιοθήκες:

- **System C library:** Η πρότυπη βιβλιοθήκη συστήματος της C (libc) τροποποιημένη για κινητές συσκευές που βασίζονται στον πυρήνα Linux.
- **SQLite:** Μια ισχυρή και ελαφριά, ενσωματωμένη σχεσιακή μηχανή βάσης δεδομένων.
- **SGL:** Μια δισδιάστατη (2D) μηχανή γραφικών.
- **OpenGL:** Οι βιβλιοθήκες χρησιμοποιούν είτε τρισδιάστατη επιτάχυνση υλικού, όπου αυτή είναι διαθέσιμη, είτε μια βελτιωμένη τρισδιάστατη επιτάχυνση λογισμικού σε περίπτωση που η πρώτη δεν είναι διαθέσιμη.
- **Webkit:** Η μηχανή περιήγησης που χρησιμοποιείται για να εμφανίσουμε περιεχόμενο HTML.
- **FreeType:** Μια open source βιβλιοθήκη γραμμένη σε C για διαχείριση των γραμματοσειρών των εφαρμογών του συστήματος.
- **Multimedia Framework:** Ένα Framework που παρέχει διάφορες βιβλιοθήκες πολυμέσων που υποστηρίζουν την αναπαραγωγή και την καταγραφή διαφόρων μορφών ήχου και βίντεο, καθώς και των στατικών αρχείων εικόνων, όπως mpeg4, mp3 κλπ.
- **Surface Manager:** Διαχειρίζεται την πρόσβαση στο υποσύστημα προβολής, και συν-θέτει δισδιάστατα και τρισδιάστατα επίπεδα γραφικών από διάφορες εφαρμογές. [7]

Ο χρόνος εκτέλεσης Android είναι ένας συνδυασμός των βιβλιοθηκών πυρήνα και του ART (Android Runtime) ο οποίος έχει αντικαταστήσει την εικονική μηχανή Dalvik. Ο πυρήνας της βιβλιοθήκης Java παρέχει τις πιο σημαντικές λειτουργίες και το ART χρησιμοποιείται για την εκτέλεση των εφαρμογών. Η εικονική μηχανή Dalvik ήταν μία τροποποιημένη Java Virtual Machine η οποία λειτουργούσε σε κινητές συσκευές επειδή είχε απαιτήσεις χαμηλής ισχύος και μνήμης. Επιπλέον η εικονική μηχανή Dalvik επέτρεπε στις εφαρμογές να εκτελούνται πάνω σε αυτή. Η Dalvik έτρεχε αρχεία της μορφής .dex τα οποία είναι ένας μετασχηματισμός των αρχείων τύπου .class .jar και εκτελούνται χρησιμοποιώντας χαμηλότερους πόρους. Η εικονική μηχανή Dalvik είχε γραφτεί έτσι ώστε μια συσκευή να μπορεί να εκτελέσει πολλές εικονικές μηχανές ταυτόχρονα και αποτελεσματικά παρέχοντας ασφάλεια, διαχείριση μνήμης και πολυνηματική υποστήριξη. Ωστόσο από το Android 5.0 η εικονική μηχανή Dalvik αντικαταστάθηκε πλήρως από το ART το οποίο τώρα είναι υπεύθυνο για την διαχείριση του χρόνου εκτέλεσης και έχει δημιουργηθεί αποκλειστικά για το Android λειτουργικό σύστημα. Το ART μετατρέπει τον κώδικα σε Byte, τον μεταγλωττίζει σε εγγενείς οδηγίες και στη συνέχεια εκτελείται στο περιβάλλον χρόνου εκτέλεσης. [23]

3.4.1.3 Πλαίσιο Εφαρμογών

Το πλαίσιο εφαρμογών είναι το επίπεδο στο οποίο η εφαρμογή μας επικοινωνεί με τα δομικά στοιχεία. Σε αυτό το επίπεδο βρίσκονται οι διεπαφές προγραμματισμού εφαρμογών (API's) και άλλες υπηρεσίες υψηλότερου επιπέδου σε μορφή κλάσεων της γλώσσας προγραμματισμού Java. Μία σημαντική υπηρεσία του επιπέδου πλαισίου εφαρμογής είναι ο διαχειριστής δραστηριοτήτων ο οποίος είναι υπεύθυνος για τον κύκλο ζωής μίας εφαρμογής. Άλλες δυνατότητες που έχουμε σε αυτό το επίπεδο είναι και η πρόσβαση σε υπηρεσίες εντοπισμού τοποθεσίας, σε διεργασίες παρασκηνίου, ορισμός ειδοποιήσεων και άλλα.

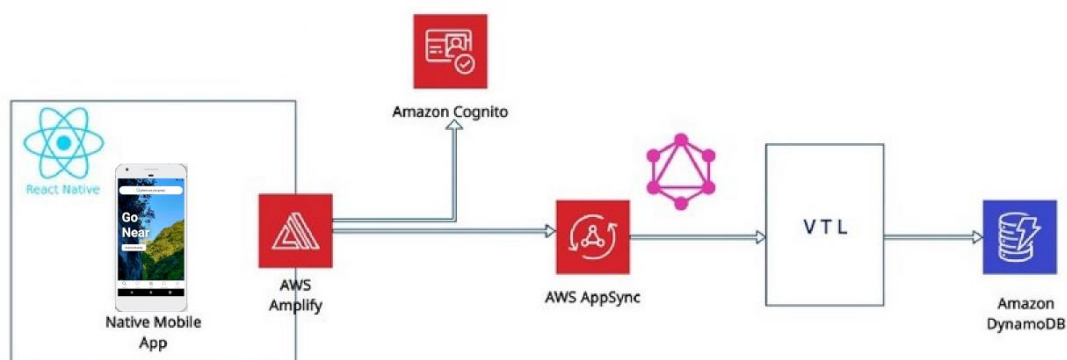
3.4.1.4 Εφαρμογές

Το επίπεδο εφαρμογών είναι το ανώτερο στρώμα στην αρχιτεκτονική και είναι το επίπεδο στο οποίο βρίσκονται οι εφαρμογές συστήματος. Κάποιες από τις εφαρμογές σε αυτό το επίπεδο είναι προ-εγκατεστημένες και παρέχουν τις βασικές λειτουργίες που χρειάζεται να έχει ένα τηλέφωνο. Κάποιες από αυτές τις λειτουργίες είναι η πραγματοποίηση τηλεφωνικών κλήσεων, η αποστολή SMS και η περιήγηση στο Διαδίκτυο. Εκτός από τις προ-εγκατεστημένες εφαρμογές το λειτουργικό σύστημα Android δίνει την δυνατότητα στους προγραμματιστές να δημιουργήσουν τις δικές τους εφαρμογές. Αυτές οι εφαρμογές μπορούν να είναι κάτι καινούριο και διαφορετικό είτε να είναι εφαρμογές οι οποίες θα αντικαταστήσουν τις ήδη υπάρχουσες προ-εγκατεστημένες.

Κεφάλαιο 4. Η αρχιτεκτονική της εφαρμογής Mapography και τεχνολογίες που χρησιμοποιήθηκαν.

4.1 Γενική Περιγραφή συστήματος Mapography

Το Mapography είναι ένα υπολογιστικό σύστημα mobile crowdsensing. Με αυτό το σύστημα ο κάθε χρήστης θα μπορεί να ενημερώνεται για κοντινά σημεία στην περιοχή που βρίσκεται μαζί με διάφορες φωτογραφίες οι οποίες έχουν δημιουργηθεί από τους υπόλοιπους χρήστες. Η κύρια καινοτομία του συστήματος Mapography είναι ότι προσπαθεί να συνδυάσει τις γνώσεις μαζί με τις φωτογραφίες που έχουν οι χρήστες μίας περιοχής με σκοπό κάποιος άλλος χρήστης να μπορέσει να μάθει και να ανακαλύψει νέες τοποθεσίες και νέα πράγματα για αυτή τη τοποθεσία. Ο τρόπος άντλησης δεδομένων από τον χρήστη γίνεται με βάση μόνο την γεωγραφική τοποθεσία του. Εφόσον η εφαρμογή γνωρίζει την τοποθεσία του θα μπορεί να προτείνει στον χρήστη διάφορες κοντινές τοποθεσίες. Ωστόσο η τοποθεσία δεν είναι απαραίτητη προκειμένου να μπορέσει να λειτουργήσει η εφαρμογή. Ο κάθε χρήστης έχει τη δυνατότητα να αναζητήσει μία συγκεκριμένη τοποθεσία και γύρω από αυτή να δει τις διάφορες πληροφορίες που υπάρχουν από τους υπόλοιπους χρήστες. Η αρχιτεκτονική της εφαρμογής Mapography παρουσιάζεται στο εικόνα 4.1. Έχουμε μία συσκευή με λειτουργικό σύστημα Android στην οποία είναι εγκατεστημένη η εφαρμογή Mapography. Το front-end περιβάλλον της εφαρμογής έγινε με τη χρήση του Expo, μία επέκταση της βιβλιοθήκης React Native με την οποία μπορούμε να φτιάξουμε cross-platform εφαρμογές. Το back-end κομμάτι της εφαρμογής το διαχειρίζεται η AWS Amplify, ένα σετ προϊόντων και υπηρεσιών τα οποία μπορούν να χρησιμοποιηθούν μαζί ή ξεχωριστά. Πιο συγκεκριμένα χρησιμοποιείται η υπηρεσία Amazon Cognito η οποία είναι υπεύθυνη για την αυθεντικοποίηση των χρηστών και η υπηρεσία AWS AppSync η οποία είναι υπεύθυνη για τη διαχείριση των δεδομένων. Αρχικά για να λειτουργήσει η εφαρμογή Mapography χρειάζεται να υπάρχει σύνδεση στο διαδίκτυο. Έπειτα αφού γίνει η αυθεντικοποίηση με το Amazon Cognito ο χρήστης έχει πλήρη πρόσβαση στην εφαρμογή.

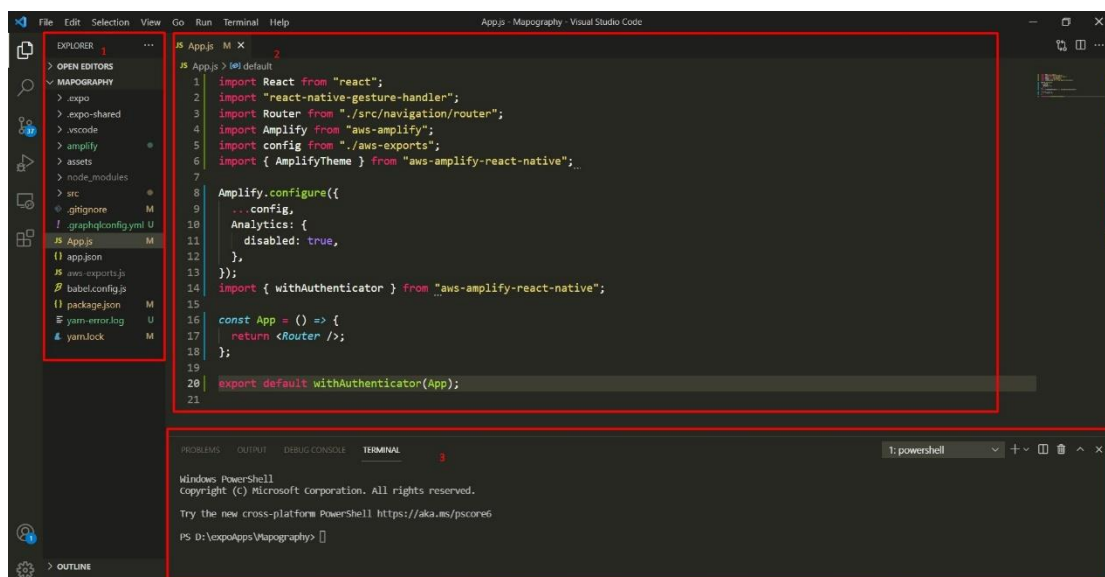


Εικόνα 4. 1 Αρχιτεκτονική του συστήματος Mapography

Μόλις ο χρήστης αποκτήσει πρόσβαση στην εφαρμογή, θα μπορεί να ανακτήσει, αλλά και να εισάγει πληροφορίες, αφού η εφαρμογή Mapography χρησιμοποιεί το AppSync το οποίο είναι υπεύθυνο για αυτή τη διαδικασία. Επίσης το AppSync χρησιμοποιεί ένα GraphQL το οποίο επικοινωνεί με τη βάση δεδομένων προκειμένου να αποθηκεύσουμε ή να ανακτήσουμε πληροφορίες. Οι πληροφορίες αυτές αποθηκεύονται στην Amazon DynamoDB, μία NoSQL βάση δεδομένων που παρέχει η υπηρεσία AWS AppSync.

4.2 Visual Studio Code

Ο Editor που επιλέχθηκε για την υλοποίηση της εφαρμογής Mapography ήταν ο Visual Studio Code. Ο Visual Studio Code είναι ένας freeware source-code editor και έχει δημιουργηθεί από τη Microsoft για τα λειτουργικά συστήματα Linux, Windows και MacOS. [25] Με τον Visual Studio Code έχουμε τη δυνατότητα να υλοποιήσουμε σχεδόν οποιοδήποτε Project, καθώς υποστηρίζει μεγάλη ποικιλία από γλώσσες προγραμματισμού συμπεριλαμβανόμενων των JavaScript, Python, Java, C++. Ο κύριος λόγος που επιλέχθηκε για την υλοποίηση του Mapography είναι ότι παρέχει μεγαλύτερη υποστήριξη στα project που χρησιμοποιούν τεχνολογίες όπως το Node.js, JavaScript και React για την ανάπτυξή τους. Σαν Editor είναι πολύ ελαφρύς, γρήγορος και προσφέρει παράλληλα πολλές δυνατότητες στον χρήστη. Μία από αυτές τις δυνατότητες είναι και οι επεκτάσεις. Με τις επεκτάσεις μπορούμε να προσθέσουμε στον Editor καινούριες δυνατότητες και λειτουργίες με σκοπό να κάνουμε την υλοποίηση του Project μας πιο εύκολη αλλά και πιο γρήγορη. Άλλη μία πολύ χρήσιμη δυνατότητα που παρέχει είναι ο Debugger, με τον οποίο μπορούμε να εντοπίσουμε διάφορα σφάλματα που υπάρχουν στο Project, να μεταβούμε στον πηγαίο κώδικα, να ελέγξουμε μεταβλητές και να εκτελέσουμε εντολές στην κονσόλα. Επίσης ο Visual Studio Code έχει υποστήριξη για το Git, ώστε εάν θέλουμε να εργαστούμε με το source control να μη χρειαστεί να φύγουμε από τον Editor. Στην **Εικόνα 4.2** βλέπουμε πώς φαίνεται η αρχική σελίδα του Visual Studio Code όταν ανοίξουμε ένα project.



Εικόνα 4. 2 Visual Studio Code

Στο κουτάκι με τον αριθμό (1) βρίσκεται ο Explorer. Εδώ βρίσκονται όλοι οι υπό-φάκελοι που περιέχει το project μαζί με τα αρχεία. Από εκεί μπορούμε να επιλέξουμε και να ψάξουμε το αρχείο που θέλουμε να ανοίξουμε και να επεξεργαστούμε. Όταν επιλέξουμε το αρχείο αυτό, ο Editor ανοίγει ένα νέο παράθυρο (κουτάκι 2) και μέσα σε αυτό το παράθυρο μπορούμε να δούμε και να επεξεργαστούμε τον κώδικά. Στο (κουτάκι 3) ο Editor παρέχει 4 επιλογές:

1. **Problems:** Η ενότητα Problems αφορά προβλήματα που μπορεί να έχουν δημιουργηθεί στο Project.
2. **Output:** Αν έχουμε συνδέσει το Project με το Git, τότε σε αυτή την ενότητα θα αποτυπώνονται σφάλματα ή ενημερώσεις που αφορούν το Git.
3. **Debug Console:** Εδώ εμφανίζονται τα αρχεία καταγραφής του Debugger.
4. **Terminal:** ένα τερματικό (στα Windows συνήθως είναι το PowerShell) το οποίο υπάρχει για να μην χρειαστεί να βγούμε από τον Editor.

Αντιλαμβάνεται λοιπόν κάποιος, κοιτάζοντας και το παραπάνω σχήμα, ότι ο Visual Studio Code είναι ένας πολύ απλός εύχρηστος και προσαρμόσιμος Editor μαζί με αρκετές χρήσιμες προσθήκες.

4.3 Node.js

Το Node.js είναι ένα open source περιβάλλον διακομιστή το οποίο τρέχει σε διάφορες πλατφόρμες (Windows, Linux, MacOS) και χρησιμοποιεί ασύγχρονο προγραμματισμό. Το Node.js έχει σχεδιαστεί για τη δημιουργία επεκτάσιμων εφαρμογών δικτύου. Κάποιες από τις λειτουργίες του Node.js είναι οι εξής:

- μπορεί να δημιουργήσει δυναμικό περιεχόμενο σελίδας
- μπορεί να δημιουργήσει, να ανοίξει, να διαβάσει, να γράψει, να διαγράψει και να κλείσει αρχεία στο διακομιστή
- μπορεί να συλλέξει δεδομένα φόρμας
- μπορεί να προσθέσει, να διαγράψει, να τροποποιήσει δεδομένα στη βάση δεδομένων

Στη περίπτωση του Mapography το Node.js χρειάζεται για να εγκαταστήσουμε το Expo CLI καθώς και βιβλιοθήκες οι οποίες θα χρησιμοποιηθούν για την καλύτερη ανάπτυξη της εφαρμογής μέσω του npm (Node Project Manager). Το npm είναι ο προεπιλεγμένος διαχειριστής πακέτων για το Node.js. [26][27]

4.4 Expo

Το Expo αποτελεί επέκταση της React Native, οπότε όταν γράφουμε κώδικα χρησιμοποιώντας το, είναι το ίδιο με το να γράφουμε κώδικα σε React Native, απλά με μερικές παραπάνω δυνατότητες. Το Expo απευθύνεται κυρίως στους αρχάριους προγραμματιστές με μικρή εμπειρία σε React Native, καθώς και σε αυτούς που θέλουν να δημιουργήσουν κάποιο μικρό project. Αποτελείται από δυο βασικά κομμάτια τα οποία είναι:

1. **Expo CLI:** Πρόκειται για μία διεπαφή γραμμής εντολών (Command Line Interface) η οποία μπορεί να δημιουργήσει ένα νέο Project, να προβάλει τα αρχεία καταγραφής, να

ανοίξει το project σε κάποια κινητή συσκευή ή emulator για να μπορέσουμε να το τεστάρουμε και τέλος μας δίνει τη δυνατότητα να το δημοσιεύσουμε.

2. **Expo Client:** Ο expo client είναι μία εφαρμογή που μπορεί να εγκατασταθεί σε οποιαδήποτε κινητή συσκευή που τρέχει λειτουργικό σύστημα Android ή iOS. Με αυτή την εφαρμογή μπορούμε να ανοίξουμε και να δοκιμάσουμε απευθείας στην κινητή συσκευή μας το project που έχουμε δημιουργήσει με το Expo CLI.

Το Expo λοιπόν αποτελεί ένα σετ εργαλείων φτιαγμένα σε React Native τα οποία κάνουν πιο εύκολη τη δημιουργία εφαρμογών. Στις περισσότερες εφαρμογές που θα θελήσει να αναπτύξει κάποιος με το expo, σπάνια θα υπάρξει κάποια δυσκολία ή η ανάγκη να γράψει εγγενή κώδικα. Σε περίπτωση που συμβεί αυτό πάντα υπάρχει η δυνατότητα από το Expo να κάνουμε eject το project και να το συνεχίσουμε από εκεί που το αφήσαμε και όχι από την αρχή. [28][29]

4.4.1 Εγκατάσταση του Expo στον υπολογιστή μας

Για να μπορέσουμε να εγκαταστήσουμε το Expo στον υπολογιστή μας, θα πρέπει να έχουμε και εγκατεστημένο το node.JS και ύστερα να ανοίξουμε ένα τερματικό στον υπολογιστή. Αφού ανοίξουμε το τερματικό θα πρέπει να γράψουμε την παρακάτω εντολή προκειμένου να εγκαταστήσουμε το Expo CLI στον υπολογιστή:

```
C:\Users\Ilias>npm install -g expo-cli
```

Μόλις τελειώσει η εγκατάσταση των απαραίτητων πακέτων και δεδομένου ότι δεν θα υπάρχει κάποιο σφάλμα στο αρχείο καταγραφής του τερματικού μας, τότε σημαίνει πως έχουμε εγκαταστήσει το Expo CLI επιτυχώς στον υπολογιστή μας. Μπορούμε δηλαδή να ξεκινήσουμε ανά πάσα στιγμή κάποιο project χρησιμοποιώντας το Expo. Για να δημιουργήσουμε ένα καινούριο project πηγαίνουμε στο ευρετήριο που θέλουμε να το δημιουργήσουμε, ανοίγουμε το τερματικό και γράφουμε την παρακάτω εντολή:

```
PS D:\expoApps> expo init Mapography
```

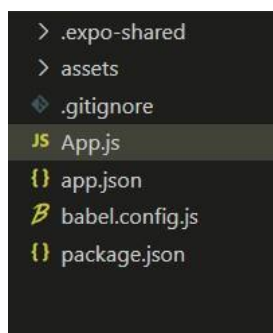
Με την εντολή **expo init** (initialize) λέμε στο Expo να δημιουργήσει το project με το όνομα που θέλουμε. Πατώντας το κουμπί enter του υπολογιστή, το expo θα ρωτήσει τι template θα θέλουμε να χρησιμοποιήσουμε για το project, όπως φαίνεται και στην παρακάτω **εικόνα 4.3**.

```
? Choose a template: » - Use arrow-keys. Return to submit.
----- Managed workflow -----
> blank a minimal app as clean as an empty canvas
  blank (TypeScript) same as blank but with TypeScript configuration
  tabs (TypeScript) several example screens and tabs using react-navigation and TypeScript
----- Bare workflow -----
  minimal bare and minimal, just the essentials to get you started
  minimal (TypeScript) same as minimal but with TypeScript configuration
```

Εικόνα 4. 3 Δημιουργία νέου Project με το Expo

Οι προσεγγίσεις που υπάρχουν για την κατασκευή ενός project με το Expo είναι δύο, οι διαχειριζόμενες ροές εργασίας (managed workflow) και οι γυμνές ροές εργασίας (bare

workflow). Στις διαχειριζόμενες ροές εργασίας ο χρήστης αναπτύσσει το project του χρησιμοποιώντας JavaScript ή TypeScript και για τα υπόλοιπα εργαλεία και υπηρεσίες που είναι αναγκαίες τα φροντίζει το expo. Από την άλλη στη γυμνή ροή εργασίας, έχουμε τον πλήρη έλεγχο σε κάθε πτυχή του project και τα εργαλεία και οι υπηρεσίες που έχουμε στη διάθεσή μας είναι λίγο πιο περιορισμένα σε σχέση με αυτά της διαχειριζόμενης ροής. Στη περίπτωση του Mapography έχει χρησιμοποιηθεί το blank managed workflow. Μόλις επιλέξουμε τη ροή εργασίας, το Expo κάνει τις απαραίτητες ενέργειες που χρειάζονται για να δημιουργηθεί το project και φτιάχνει ένα ευρετήριο με όλα τα αναγκαία εργαλεία και υπηρεσίες που χρειάζονται για να τρέξει η εφαρμογή. Στην **εικόνα 4.4** βλέπουμε το ευρετήριο που δημιουργείται εάν επιλέξουμε το blank managed workflow. Το αρχείο App.js περιέχει τον κώδικα της εφαρμογής που δημιουργήθηκε από το Expo, το gitignore περιέχει τους φακέλους και αρχεία τα οποία πρέπει να αγνοηθούν από το git και οι υπόλοιποι φάκελοι και αρχεία περιέχουν τα απαραίτητα εργαλεία και υπηρεσίες που χρειάζονται για να τρέξει η εφαρμογή.

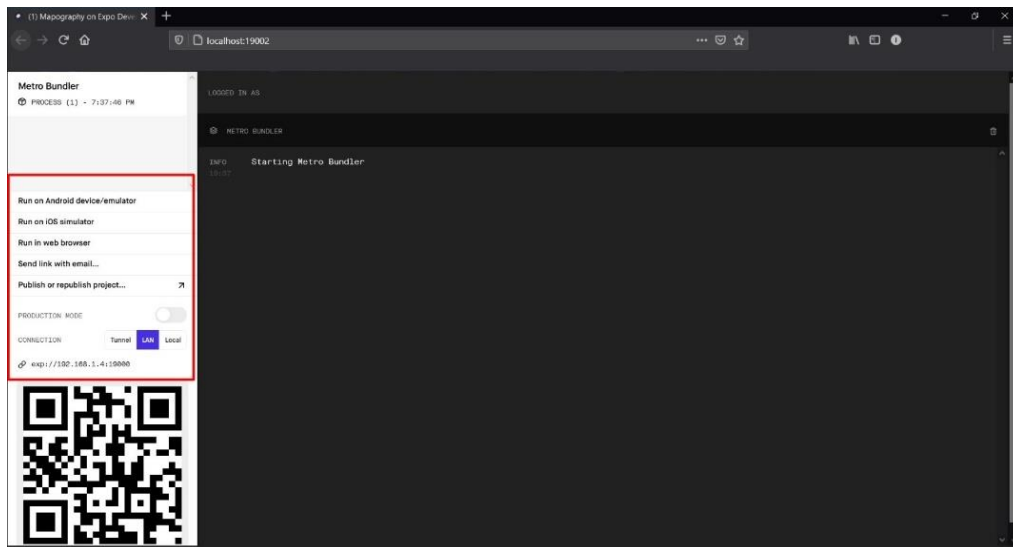


Εικόνα 4. 4 Ευρετήριο του νέου Project

Για να τρέξουμε την εφαρμογή του project μας, πρέπει να γράψουμε στο τερματικό την ακόλουθη εντολή:

```
PS D:\expoApps\Test12> expo start
```

Μόλις τρέξουμε στο τερματικό την παραπάνω εντολή, θα ανοίξει στον browser μας μία σελίδα με τα εργαλεία για προγραμματιστές **εικόνα 4.5**. Εδώ μπορούμε να επιλέξουμε πού θέλουμε να τρέξει η εφαρμογή emulator, simulator, browser ή κινητή συσκευή σκανάροντας το QR code.



Εικόνα 4. 5 Metro Bundler

Εφόσον διαλέξουμε τον τρόπο με τον οποίο θα ανοίξουμε την εφαρμογή, τότε το Expo θα κάνει τις απαραίτητες ενέργειες που χρειάζονται ώστε να τρέξει η εφαρμογή στη συσκευή μας. Την πρώτη φορά που θα τρέξουμε την εφαρμογή, το μόνο που θα εμφανιστεί θα είναι ένα άσπρο background στο οποίο θα γράφει “Open up App.js to start working on your app!” **Εικόνα 4.6**



Εικόνα 4. 6 Αρχική εικόνα του App.js

Μας ενημερώνει δηλαδή πως εάν θέλουμε να επεξεργαστούμε την αρχική σελίδα της εφαρμογής θα πρέπει να βρούμε το App.js στο ευρετήριο του project και να κάνουμε αλλαγές στον κώδικά του. Πηγαίνοντας στο ευρετήριο του project και αφού ανοίξουμε το App.js μπορούμε να το επεξεργαστούμε, και μόλις κάνουμε οποιαδήποτε αλλαγή την οποία θα

σώσουμε (save) θα εμφανιστεί απευθείας στην συσκευή που τρέχει η εφαρμογή. Γενικά το Expo είναι πολύ εύκολο στη χρήση του και δύσκολα θα συναντήσει κανείς προβλήματα στην εγκατάσταση του ή στη χρήση του. Ειδικότερα για τους αρχάριους που δεν έχουν πολλές γνώσεις από React Native, το Expo είναι μονόδρομος καθώς είναι πιο φιλικό και καθοδηγητικό σε σχέση με το React Native CLI. [30][31]

4.5 AWS AMPLIFY

Το AWS Amplify είναι μία πλατφόρμα η οποία αποτελείται από ένα σύνολο εργαλείων και υπηρεσιών που μπορούν να χρησιμοποιηθούν μαζί ή μόνα τους, αναλόγως τις ανάγκες που έχει ένας προγραμματιστής για το εκάστοτε project. Όπως αναφέρθηκε και στην **Ενότητα 4.1** η εφαρμογή Mapography κάνει χρήση μερικών υπηρεσιών που παρέχει η AWS Amplify ώστε να μπορεί να λειτουργεί σωστά το back-end κομμάτι της. Η χρήση αυτών των εργαλείων και υπηρεσιών βοήθησε να χτιστεί μία Full-stack εφαρμογή πιο γρήγορα αλλά και πιο εύκολα. Με το AWS Amplify μπορούμε να διαμορφώσουμε το back-end μίας εφαρμογής μέσα σε λίγα μόνο λεπτά, και έτσι θα έχουμε δημιουργήσει μία αποδοτική, χωρίς διακομιστή και ασφαλή εφαρμογή. Μπορούμε να σκεφτούμε το AWS Amplify ως μία JavaScript βιβλιοθήκη η οποία μας επιτρέπει να αναπτύξουμε εφαρμογές χωρίς διακομιστή στο cloud. Τα τρία κυριότερα στοιχεία του AWS Amplify είναι τα εξής:

- **Βιβλιοθήκες:** Αυτό το στοιχείο μας επιτρέπει να προσθέσουμε, να ενσωματώσουμε και να αλληλοεπιδράσουμε με τις υπηρεσίες cloud που προσφέρει η AWS. Οι βιβλιοθήκες διευκολύνουν επίσης την προσθήκη ασφαλούς ταυτοποίησης σε μία εφαρμογή, την αποθήκευση αρχείων και δεδομένων χωρίς να χρειαζόμαστε κάποιον διακομιστή, την προσθήκη ειδοποιήσεων και πολλές άλλες λειτουργίες τις οποίες μπορεί να χρειαστεί μία εφαρμογή.
- **Διεπαφή χρήστη:** Πρόκειται για κάποιες έτοιμες διεπαφές χρήστη που παρέχονται από τις υπηρεσίες της AWS. Μία τέτοια υπηρεσία χρησιμοποιείται και από την εφαρμογή Mapography και ονομάζεται Amazon Cognito. Συγκεκριμένα οι διεπαφές χρήστη που έχει η εφαρμογή για την σύνδεση των χρηστών με username και κωδικό πρόσβασης είναι έτοιμες και δεν χρειάστηκε κάποια προγραμματιστική εμπειρία για την υλοποίησή τους.
- **Εργαλειοθήκη τερματικού:** Η εργαλειοθήκη τερματικού είναι εξαιρετικά χρήσιμη σε περιπτώσεις που χρειαζόμαστε να προσθέσουμε επιπλέον AWS υπηρεσίες στην εφαρμογή μας. Σε περίπτωση που συμβεί αυτό, μέσα από τη χρήση κάποιων απλών εντολών στο τερματικό μας μπορούμε να κάνουμε αλλαγές αλλά και να διαχειριστούμε το back-end της εφαρμογής μας.

Τα τρία αυτά στοιχεία συνεργάζονται μεταξύ τους με στόχο να μπορούν να διαχειριστούν τον κύκλο ζωής (lifecycle) ανάπτυξης εφαρμογών. [32][33]

4.5.1 Πλεονεκτήματα χρήσης του AWS Amplify σε ένα project

Κατά την υλοποίηση ενός full-stack project ένας προγραμματιστής θα πρέπει να αποφασίσει τι τεχνολογίες θα χρησιμοποιήσει τόσο για το front-end κομμάτι του project όσο και για το back-end κομμάτι. Η απόφαση αυτή λαμβάνεται με βάση την εμπειρία χρήσης που έχει ο κάθε προγραμματιστής σε κάποια από τις διαθέσιμες τεχνολογίες, και με βάση τα πλεονεκτήματα

που μπορεί να έχει μία τεχνολογία σε σχέση με κάποια άλλη. Σε αυτή την ενότητα θα δούμε τα πλεονεκτήματα που έχουν οι υπηρεσίες του AWS Amplify, τα οποία ήταν ο λόγος που επιλέχθηκε για την ανάπτυξη του back-end κομματιού της εφαρμογής Mapography:

- **Εύκολη και καθοδηγούμενη ανάπτυξη διεπαφών χρήστη:** Το AWS Amplify παρέχει μια απλή, γρήγορη και σύγχρονη προσέγγιση όσον αφορά τη δημιουργία διεπαφών χρήστη σε μία εφαρμογή. Όταν θέλουμε για παράδειγμα να φτιάξουμε μία εφαρμογή η οποία επιτρέπει στους χρήστες να συνδεθούν μέσω e-mail, με τη χρήση του AWS Amplify μπορούμε να έχουμε έτοιμες διεπαφές χρήστη χωρίς να χρειάζεται να τις προγραμματίσουμε. Αυτό γλυτώνει αρκετό χρόνο που θα χρειαζόταν κάποιος προγραμματιστής για την ανάπτυξη ενός τέτοιου μοντέλου εφαρμογής.
- **Πληρωμή με βάση τη χρήση:** Το κόστος για την χρήση των εργαλείων και υπηρεσιών του AWS Amplify στην εφαρμογή Mapography είναι σχεδόν μηδενικό, καθώς για να αυξηθεί θα πρέπει να χρησιμοποιείται από μεγάλο αριθμό χρηστών.
- **Μεγάλη Υποστήριξη:** Όταν θέλουμε να χρησιμοποιήσουμε υπηρεσίες ή εργαλεία που παρέχει το AWS Amplify μπορούμε πολύ εύκολα να βρούμε οδηγούς οι οποίοι θα μας βοηθήσουν με τις ανάγκες που έχουμε εκείνη τη στιγμή. Επίσης σε περίπτωση που κάποιος οδηγός δεν μας βοηθήσει, υπάρχουν και αρκετά φόρουμ στα οποία μπορούμε να υποβάλλουμε τις ερωτήσεις μας και να απαντηθούν άμεσα.
- **Ανάλυση Δεδομένων:** Το AWS Amplify διαθέτει έναν πίνακα με δεδομένα που αφορούν το εκάστοτε project. Αυτός ο πίνακας είναι εξαιρετικά χρήσιμος αφού παρακολουθεί τις περιόδους σύνδεσης των χρηστών και προσφέρει μετρήσεις εντός μίας εφαρμογής. Έτσι κάποιος προγραμματιστής, σχεδιαστής ή διαχειριστής έργων θα βοηθηθεί αρκετά και θα μπορεί πιο εύκολα να διαχειρίζεται το project, να πάρει αποφάσεις που το αφορούν αλλά και να αποφασίσει για μελλοντικές προσθήκες χαρακτηριστικών. [32][33]

4.5.2 Εγκατάσταση AWS Amplify στον υπολογιστή

Για να μπορέσουμε να χρησιμοποιήσουμε τις υπηρεσίες και τα εργαλεία που μας παρέχει το AWS Amplify θα πρέπει αρχικά να δημιουργήσουμε έναν AWS λογαριασμό και ύστερα να εγκαταστήσουμε τη διεπαφή τερματικού που παρέχει το AWS Amplify στο υπολογιστή που θα χρησιμοποιήσουμε για την ανάπτυξη του project. Με τη χρήση του npm και αφού ανοίξουμε το τερματικό της αρεσκείας μας πληκτρολογούμε την παρακάτω εντολή:

```
npm install -g @aws-amplify/cli
```

Όταν τελειώσει η εκτέλεση αυτής της εντολής και εφόσον δεν εμφανιστεί κάποιο σφάλμα στο τερματικό μας, θα έχουμε ολοκληρώσει την εγκατάσταση της διεπαφής τερματικού του AWS Amplify. Στη συνέχεια θα πρέπει να διαμορφώσουμε τη διεπαφή τερματικού με τα στοιχεία του AWS λογαριασμού. Για να γίνει αυτό θα πρέπει πρώτα να συνδεθούμε στον λογαριασμό μας και ύστερα να φτιάξουμε ένα Amazon IAM (Identity and Access Management) χρήστη με δικαιώματα διαχειριστή. Αφού ολοκληρώσουμε τα παραπάνω βήματα, πηγαίνουμε πάλι στο τερματικό της επιλογής μας και πληκτρολογούμε την επόμενη εντολή:

```
amplify configure
```

Με την εκτέλεση της παραπάνω εντολής θα εμφανιστεί στο τερματικό μας το ακόλουθο μήνυμα:

```
Enter the access key of the newly created user:
? accessKeyId: # YOUR_ACCESS_KEY_ID
? secretAccessKey: # YOUR_SECRET_ACCESS_KEY
This would update/create the AWS Profile in your local machine
? Profile Name: # (default)

Successfully set up the new user.
```

Εδώ θα πρέπει να πληκτρολογήσουμε το Access Key Id και το Secret Access Key τα οποία δημιουργήθηκαν όταν φτιάξαμε τον IAM χρήστη στον AWS λογαριασμό μας. Μόλις δώσουμε τα δύο αυτά κλειδιά και εφόσον δεν ήρθαμε αντιμέτωποι με κάποιο σφάλμα κατά τη διάρκεια της εγκατάστασης, θα είμαστε σε θέση να χρησιμοποιήσουμε της υπηρεσίες και τα εργαλεία του AWS Amplify σε οποιοδήποτε project τα χρειαστούμε. [36]

4.5.3 Προσθήκη AWS Amplify στην εφαρμογή

Αφού ολοκληρώσαμε την εγκατάσταση του AWS Amplify στον υπολογιστή μας, μπορούμε να χρησιμοποιήσουμε τις υπηρεσίες του στην εφαρμογή μας. Για να συνδέσουμε ένα project με το AWS Amplify πρέπει να πληκτρολογήσουμε στο τερματικό την παρακάτω εντολή:

```
PS D:\expoApps\Mapography> amplify init
```

Μόλις εκτελέσουμε την εντολή θα εμφανιστούν στο τερματικό μας οι ακόλουθες επιλογές:

```
$ amplify init
Note: It is recommended to run this command from the root of your app directory
? Enter a name for the project Mapography
? Enter a name for the environment dev
? Choose your default editor: Visual Studio Code
? Choose the type of app that you're building javascript
Please tell us about your project
? What javascript framework are you using react-native
? Source Directory Path: /
? Distribution Directory Path: /
? Build Command: npm.cmd run-script build
? Start Command: npm.cmd run-script start
Using default provider awscloudformation

For more information on AWS Profiles, see:
https://docs.aws.amazon.com/cli/latest/userguide/cli-configure-profiles.html

? Do you want to use an AWS profile? Yes
? Please choose the profile you want to use default
```

Στην πρώτη επιλογή, πρέπει να καταχωρήσουμε ένα όνομα για το AWS Amplify project, στην περίπτωση μας το όνομα Mapography. Μετά αφήνουμε το προεπιλεγμένο όνομα dev για το όνομα περιβάλλοντος αφού βρισκόμαστε σε περιβάλλον ανάπτυξης. Στην επόμενη επιλογή θα πρέπει να προσθέσουμε τον editor που χρησιμοποιούμε για την ανάπτυξη της εφαρμογής. Ο editor που χρησιμοποιήθηκε για την εφαρμογή μας είναι ο Visual Studio Code οπότε αυτόν πρέπει και να επιλέξουμε. Στην τέταρτη επιλογή προσθέτουμε τον τύπο της εφαρμογής που

υλοποιούμε. Επειδή η React Native είναι μία βιβλιοθήκη - επέκταση της JavaScript, επιλέγουμε JavaScript. Στην επόμενη ενότητα θα πρέπει να βάλουμε την React Native καθώς είναι η βιβλιοθήκη JavaScript που χρησιμοποιήθηκε για το project. Τις υπόλοιπες τέσσερις επιλογές τις αφήνουμε στις προκαθορισμένες απαντήσεις καθώς δεν χρειάζεται να αλλάξουμε κάτι σε αυτές. Τέλος θα πρέπει να επιλέξουμε το AWS προφίλ που δημιουργήσαμε κατά την εγκατάσταση του AWS Amplify στον υπολογιστή μας. Όταν ολοκληρώσουμε τα παραπάνω βήματα, θα δημιουργηθούν στο ευρετήριο του project ένας νέος φάκελος και ένα νέο αρχείο:

- **amplify:** Αυτός ο φάκελος δημιουργείται στη ρίζα του ευρετηρίου του project και περιέχει τη διαμόρφωση του backend περιβάλλοντος της εφαρμογής.
- **aws-exports.js:** Αυτό το αρχείο δημιουργείτε επίσης στη ρίζα του ευρετηρίου και περιέχει διάφορες πληροφορίες οι οποίες αφορούν τις υπηρεσίες και τα εργαλεία που θα δημιουργηθούν με το AWS Amplify.

Το επόμενο βήμα που θα πρέπει να ακολουθήσουμε είναι να προσθέσουμε κάποιες νέες εξαρτήσεις (Dependencies) οι οποίες είναι απαραίτητες για την ομαλή λειτουργία του AWS Amplify με την εφαρμογή μας. Πληκτρολογούμε λοιπόν την παρακάτω εντολή στο τερματικό μας για να τις εγκαταστήσουμε στο project μας:

```
npm install aws-amplify aws-amplify-react-native @react-native-community/netinfo
```

Όταν ολοκληρωθεί και η εγκατάσταση των εξαρτήσεων στο project μας, θα έχουμε ολοκληρώσει όλα τα απαραίτητα βήματα που πρέπει να ακολουθήσουμε για να προσθέσουμε το AWS Amplify στο project μας.

4.5.4 Amazon Cognito

Το Amazon Cognito είναι η μία από τις δύο υπηρεσίες του AWS Amplify η οποία χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής Mapography. Συγκεκριμένα το Amazon Cognito καθιστά δυνατή την ταυτοποίηση των χρηστών σε μία εφαρμογή. Είναι μία απλή υπηρεσία ταυτοποίησης η οποία βοηθά τη διαχείριση και το συγχρονισμό δεδομένων των χρηστών με ασφάλεια σε όλες τις κινητές συσκευές τους. Με τη χρήση αυτής της υπηρεσίας μπορούμε είτε να δημιουργήσουμε μοναδικές ταυτότητες για τους χρήστες της εφαρμογής, για παράδειγμα μέσω username και κωδικού, είτε να επιλέξουμε μία δημόσια παροχή σύνδεσης όπως το Google login, το Amazon login ή το Facebook login. Μπορούμε επίσης να αποθηκεύσουμε τοπικά δεδομένα στη συσκευή ενός χρήστη και με αυτόν τον τρόπο θα επιτρέψουμε στην εφαρμογή μας να λειτουργεί ακόμα και όταν η συσκευή βρίσκεται εκτός σύνδεσης.[34]

4.5.4.1 Προσθήκη του Amazon Cognito στην εφαρμογή Mapography

Η διαδικασία που απαιτείται για την προσθήκη του Amazon Cognito στην εφαρμογή μας είναι αρκετά απλή και δεν χρειάζεται αρκετό χρόνο για να ολοκληρωθεί. Αρχικά πηγαίνουμε πάλι στο τερματικό και πληκτρολογούμε την παρακάτω εντολή:

```
amplify add auth
```

Εκτελώντας την θα εμφανιστεί το παρακάτω παράθυρο στο οποίο θα επιλέξουμε την προκαθορισμένη διαμόρφωση (default configuration), το e-mail ως τρόπο σύνδεσης στην εφαρμογή και στην ερώτηση της διαμόρφωσης προχωρημένων ρυθμίσεων επιλέγουμε «Όχι»

```
$ amplify add auth
Using service: Cognito, provided by: awscloudformation

The current configured provider is Amazon Cognito.

Do you want to use the default authentication and security configuration? Default configuration
Warning: you will not be able to edit these selections.
How do you want users to be able to sign in? Email
Do you want to configure advanced settings? No, I am done.
```

Όταν τελειώσει η εγκατάσταση του Amazon Cognito στο project μας θα πρέπει να πληκτρολογήσουμε ακόμα μία εντολή στο τερματικό μας προκειμένου να αποθηκευτούν οι αλλαγές που κάναμε.

```
amplify push
```

Σε αυτό το σημείο έχουμε ολοκληρώσει επιτυχώς την εγκατάσταση του Amazon Cognito και μπορούμε να το χρησιμοποιήσουμε σε οποιοδήποτε κομμάτι της εφαρμογής επιθυμούμε. Όπως είπαμε και στην **Ενότητα 4.5.4** το Amazon Cognito προσφέρει έτοιμες και εύκολα επεξεργάσιμες διεπαφές χρήστη για την σύνδεση χρήστη, δημιουργία λογαριασμού και επαναφορά κωδικού πρόσβασης. Στην εφαρμογή Mapography η προσθήκη της ταυτοποίησης χρηστών έγινε στο App.js αρχείο. Αρχικά προστέθηκε η παρακάτω γραμμή κώδικα:

```
import { withAuthenticator } from "aws-amplify-react-native";
```

Το withAuthenticator είναι ένα στοιχείο υψηλότερης τάξης, το οποίο μπορεί να εντοπίζει αυτόματα εάν κάποιος χρήστης της εφαρμογής είναι ήδη συνδεδεμένος και ενημερώνει τη διεπαφή χρήστη ανάλογα. Το τελευταίο πράγμα που πρέπει να κάνουμε για να ολοκληρώσουμε τη διαδικασία είναι να αλλάξουμε την τελευταία γραμμή κώδικα στο App.js όπως φαίνεται στην παρακάτω εικόνα:

```
export default withAuthenticator(App);
```

Στην ουσία προσθέτουμε «πάνω» από το στοιχείο App, το στοιχείο withAuthenticator, και την επόμενη φορά που θα τρέξουμε την εφαρμογή θα εμφανιστεί η διαδικασία ταυτοποίησης όπως φαίνεται και στις τρεις παρακάτω εικόνες:

Εικόνα 4. 9 Σελίδες του Amazon Cognito

Εικόνα 4. 8 Σελίδες του Amazon Cognito

Εικόνα 4. 7 Σελίδες του Amazon Cognito

4.5.5 AWS AppSync

Το AWS AppSync είναι μία πλήρως διαχειριζόμενη υπηρεσία η οποία κάνει εύκολη την ανάπτυξη των GraphQL APIs καθώς έχει τη δυνατότητα να τα συνδέει με ασφάλεια σε βάσεις δεδομένων όπως η AWS DynamoDB. Με το AppSync μπορούμε να βελτιώσουμε την απόδοση της εφαρμογής μας αλλά και να παρέχουμε ενημερώσεις στους χρήστες της σε πραγματικό χρόνο. Οι δυνατότητες που προσφέρει το AppSync είναι οι εξής:

- **Ασφαλής πρόσβαση σε δεδομένα:** Το AWS Appsync διευκολύνει την ασφάλεια των δεδομένων της εφαρμογής μας διότι χρησιμοποιεί ταυτόχρονα πολλούς ελέγχους ταυτότητας, μας επιτρέπει να ορίσουμε δικές μας μεθόδους ασφάλειας, να πραγματοποιήσουμε προσωρινή αποθήκευση και να έχουμε απευθείας έλεγχο στα δεδομένα από το GraphQL.
- **Λειτουργίες εκτός σύνδεσης και δυνατότητα ενημέρωσης σε πραγματικό χρόνο:** Όπως είπαμε και στην αρχή της ενότητας με τη χρήση του AWS AppSync μπορούμε να προσφέρουμε στους χρήστες της εφαρμογής μας ενημερώσεις δεδομένων σε πραγματικό χρόνο. Μας δίνει όμως και τη δυνατότητα όταν κάποιος χρήστης αποσυνδεθεί από το διαδίκτυο να αποθηκεύσουμε τα δεδομένα τοπικά στη συσκευή του και όταν επανασυνδεθεί γίνεται αυτόματος συγχρονισμός των δεδομένων.
- **Δεν χρειάζεται να διαχειριστούμε κάποιον διακομιστή:** Με το AWS Appsync μπορούμε εύκολα να δημιουργήσουμε ένα GraphQL API, να το διαχειριστούμε και να το συντηρήσουμε χωρίς να είναι αναγκαίο να διαχειριστούμε κάποιον διακομιστή.

Με λίγα λόγια προσθέτοντας το AWS AppSync στην εφαρμογή μας, έχουμε μία γρήγορη, ασφαλή και εύκολα υλοποιήσιμη λύση για την διαχείριση των δεδομένων της. [35]

4.5.5.1 Τι είναι το GraphQL

Στην **Ενότητα 4.5.5** όταν μιλήσαμε για το AWS Appsync είπαμε ότι μας βοηθάει να δημιουργήσουμε με ευκολία GraphQL APIs και ότι μπορούμε να ελέγχουμε τα δεδομένα που

περιέχουν. Για να καταλάβουμε όμως καλύτερα το AppSync καθώς και πως λειτουργεί καλό θα ήταν να εξηγήσουμε και τι ακριβώς είναι το GraphQL. Το GraphQL λοιπόν είναι μία βιβλιοθήκη ανοιχτού κώδικα η οποία διαχειρίζεται επερωτήσεις δεδομένων (data queries). Δημιουργήθηκε το 2012 από την εταιρία Facebook και το 2015 έγινε διαθέσιμη για χρήση δημοσίως. Με το GraphQL μπορούμε εμείς να καθορίσουμε την δομή των απαιτούμενων δεδομένων και όταν δημιουργήσουμε κάποια επερώτηση, μόνο αυτή θα επιστραφεί. Με αυτόν τον τρόπο αποφεύγουμε την επιστροφή μεγάλου όγκου δεδομένων. Στην παρακάτω εικόνα βλέπουμε μία επερώτηση που δημιουργήθηκε για το GraphQL API της εφαρμογής Mapography:

```
query showPropertiesBasedOnLocation {  
  listPosts(filter: {  
    and: {  
      latitude: {  
        between: [27.99643616103738, 28.617893425551618],  
      }  
      longitude: {  
        between: [-16.945665756405603, -16.134050770346267],  
      }  
    }  
  }) {  
    items {  
      id  
      title  
      latitude  
      longitude  
    }  
  }  
}
```

Εικόνα 4. 10 Επερώτηση GraphQL

Με αυτή την επερώτηση ζητάμε από το API να μας επιστρέψει το **id**, **title**, **latitude** (γεωγραφικό πλάτος) και **longitude** (γεωγραφικό μήκος) από όλα τα πόστ που έχουν δημιουργηθεί και βρίσκονται ανάμεσα από συγκεκριμένες τιμές γεωγραφικού πλάτους και μήκους. Μόλις τρέξουμε την παραπάνω επερώτηση θα λάβουμε ως αποτέλεσμα τα δεδομένα που φαίνονται στην παρακάτω εικόνα:

```
{
  "data": {
    "listPosts": {
      "items": [
        {
          "id": "9386fa86-c5e7-4f8b-b861-2f91b21efb3c",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.3815637,
          "longitude": -16.6291304
        },
        {
          "id": "c0862c31-dbe3-48ec-b8a3-9c27b5ef567f",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.4676648,
          "longitude": -16.3356302
        },
        {
          "id": "34cb1b31-cfdc-4d75-a1ce-d89c3674dbfc",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.5346675,
          "longitude": -16.4442102
        },
        {
          "id": "44ad689d-903d-420e-b179-a969a59279a1",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.4815637,
          "longitude": -16.2291304
        },
        {
          "id": "fd47c90b-38ed-46dc-b748-1ad25e04f029",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.4815637,
          "longitude": -16.2291304
        },
        {
          "id": "909d8e65-7161-4010-b0e3-57c01eac365b",
          "title": "The 40 Most Beautiful Places in the World.",
          "latitude": 28.4817637,
          "longitude": -16.4592403
        }
      ]
    }
  }
}
```

Εικόνα 4. 11 Αποτελέσματα Επερώτησης

Τη στιγμή που έγινε η επερώτηση στο GraphQL, υπήρχαν σε αυτό έξι πόστ τα οποία βρισκόταν ανάμεσα στο γεωγραφικό πλάτος και μήκος που αναζητήσαμε. Όπως βλέπουμε και στην παραπάνω εικόνα καθένα από αυτά επιστράφηκε ακριβώς με τα δεδομένα που ζητήθηκαν στην αρχή: **id**, **title**, **latitude**, **longitude**.

4.5.5.2 Προσθήκη του AWS AppSync στην εφαρμογή Mapography

Η διαδικασία που θα ακολουθήσουμε για την προσθήκη του AWS AppSync είναι παρόμοια με τη διαδικασία εγκατάστασης του Amazon Cognito. Αρχικά πηγαίνουμε στο τερματικό και πληκτρολογούμε την παρακάτω εντολή:

```
D:\expoApps\Mapography> amplify add api
```

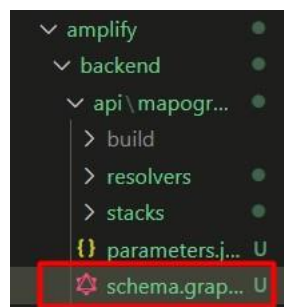
Πατώντας enter θα μας εμφανιστούν στο τερματικό οι ακόλουθες ερωτήσεις τις οποίες πρέπει να απαντήσουμε:

```

? Please select from one of the below mentioned services: GraphQL
? Provide API name: mapography
? Choose the default authorization type for the API API key
? Enter a description for the API key: demo
? After how many days from now the API key should expire (1-365): 367
? Do you want to configure advanced settings for the GraphQL API No, I am done.
? Do you have an annotated GraphQL schema? No
? Choose a schema template: Single object with fields (e.g., "Todo" with ID, name, description)

```

Η πρώτη ερώτηση αφορά το τύπο της υπηρεσίας API που χρειαζόμαστε, και στην περίπτωση του Mapography επιλέχθηκε το GraphQL. Μετά την επιλογή υπηρεσίας θα πρέπει να δώσουμε και ένα όνομα για αυτό το API. Η επόμενη ερώτηση γίνεται για την επιλογή προκαθορισμένης ταυτοποίησης και επιλέγουμε την προεπιλογή, η οποία είναι το API key. Στην τέταρτη ερώτηση πρέπει να δώσουμε μία περιγραφή για το αυτό το API. Αφού τελειώσουμε και με την περιγραφή πρέπει να επιλέξουμε το διάστημα για το οποίο θα ισχύει το API key που χρησιμοποιείται για την ταυτοποίηση. Στις επόμενες δύο ερωτήσεις αφήνουμε ως απάντηση τις προεπιλογές καθώς δεν χρειαζόμαστε προηγμένες ρυθμίσεις για το GraphQL αλλά ούτε έχουμε ήδη κάποιο GraphQL schema. Τέλος στο σχήμα προτύπου επιλέγουμε το μοναδικό αντικείμενο ώστε κάθε πόστ από τους χρήστες της εφαρμογής να είναι μοναδικό και να έχει το δικό του id, όνομα και περιγραφή. Αφού απαντήσουμε τις παραπάνω ερωτήσεις θα δημιουργηθούν οι απαραίτητοι φάκελοι που χρειάζεται το API για τη σωστή λειτουργία του. Μόλις ολοκληρωθεί αυτή η διαδικασία θα πρέπει να βρούμε το GraphQL σχήμα για να του προσθέσουμε κάποια πεδία. Στην εφαρμογή Mapography το GraphQL σχήμα βρίσκεται στον φάκελο: **amplify > backend > api > mapography > schema.graphql**



Ανοίγοντας το σχήμα την πρώτη φορά βλέπουμε τις παρακάτω πληροφορίες:

```

amplify > backend > api > mapography > schema.graphql
1  type Post @model {
2    id: ID!
3    name: String!
4    description: String!
5  }
6
7

```

Το σχήμα σε αυτή τη μορφή είναι ελλιπές καθώς το κάθε πόστ περιέχει πληροφορίες μόνο για το Id, το όνομα και την περιγραφή. Για να μπορέσουμε να εμφανίσουμε κάποια φωτογραφία με όνομα, τίτλο και περιγραφή στον χάρτη της εφαρμογής Mapography θα πρέπει να τροποποιήσουμε το σχήμα ώστε να περιέχει όλες τις απαραίτητες πληροφορίες. Το τελικό σχήμα της εφαρμογής φαίνεται στην παρακάτω εικόνα:

```
amplify > backend > api > mapography > schema.graphql
1  type Post @model {
2    id: ID!
3    image: String!
4    location: String!
5    title: String!
6    description: String!
7    latitude: Float!
8    longitude: Float!
9  }
10
```

Κάθε πόστ της εφαρμογής Mapography αποτελείται από:

- Ένα μοναδικό **Id** το οποίο δημιουργείται αυτόματα
- Μία εικόνα (**image**) σε μορφή String καθώς οι εικόνες προέρχονται από Uri
- Την τοποθεσία (**Location**) η οποία αναγράφεται κάτω από την εικόνα, σε μορφή String
- Τον τίτλο (**title**) της φωτογραφίας σε μορφή String
- Την περιγραφή (**description**) της φωτογραφίας και αυτή σε μορφή String
- Το γεωγραφικό πλάτος (**latitude**) σε μορφή Float καθώς πρόκειται για αριθμό
- Το γεωγραφικό μήκος (**longitude**) σε μορφή Float καθώς πρόκειται για αριθμό

Κοιτώντας κάποιος την παραπάνω φωτογραφία, θα παρατηρήσει πως δίπλα από κάθε **ID**, **String**, και **Float** υπάρχει το θαυμαστικό. Αυτό συμβαίνει γιατί όλα αυτά τα στοιχεία είναι απαραίτητα και για να υπάρξει πόστ θα πρέπει να δοθούν όλα από τον χρήστη, εκτός από το **ID** το οποίο είναι μορφής **id** και δημιουργείται αυτόματα με την δημιουργία κάποιου πόστ. Αφού τροποποιήσαμε και το σχήμα, θα πρέπει να πληκτρολογήσουμε την παρακάτω εντολή στο τερματικό μας προκειμένου να αποθηκευτούν οι αλλαγές:

```
amplify push
```

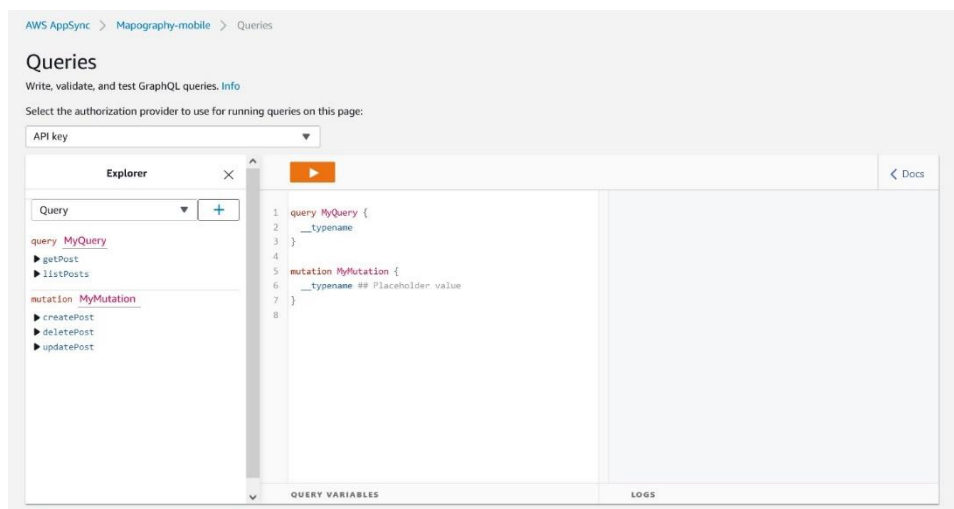
Μόλις ολοκληρωθεί και αυτή η εντολή, θα έχουμε ολοκληρώσει την διαδικασία που απαιτείται για την προσθήκη του AppSync στην εφαρμογή μας.

4.5.5.3 Δημιουργία ενός πόστ στην εφαρμογή Mapography μέσω του GraphQL API

Σε περίπτωση που θέλουμε να βεβαιωθούμε ότι η παραπάνω διαδικασία προσθήκης του AWS Appsync στην εφαρμογή μας έγινε επιτυχώς, μπορούμε να προσθέσουμε κάποιο πόστ στην εφαρμογή απευθείας από το GraphQL API. Για να το κάνουμε αυτό θα πρέπει να πληκτρολογήσουμε την παρακάτω εντολή στο τερματικό:

```
amplify api console
```

Με αυτήν την εντολή θα ανοίξει ένα καινούριο παράθυρο στον περιηγητή μας, όπως φαίνεται και στην παρακάτω **εικόνα 4.12**, στο οποίο είτε μπορούμε να κάνουμε επερωτήσεις (Queries) στο GraphQL σχήμα, είτε κάποια μεταλλαγή (mutation).



Εικόνα 4. 12 AWS Appsync console

Προκειμένου να προσθέσουμε ένα καινούριο πόστ στην εφαρμογή θα πρέπει να κάνουμε μία μεταλλαγή (mutation). Για να γίνει σωστά η διαδικασία, θα πρέπει στη μεταλλαγή που θα πραγματοποιήσουμε να προσθέσουμε όλες τις πληροφορίες που χρειάζεται το GraphQL σχήμα. Οι πληροφορίες αυτές είναι ο τίτλος, η εικόνα, η περιγραφή, το γεωγραφικό πλάτος και το γεωγραφικό μήκος. Η παρακάτω μεταλλαγή αποτελεί παράδειγμα προσθήκης ενός πόστ στην τοποθεσία Ποντικονήσι Κέρκυρας:



Εικόνα 4. 13 GraphQL Mutation

Για να ολοκληρωθεί αυτή η διαδικασία, θα πρέπει να πατήσουμε το πορτοκαλί κουμπί προκειμένου να τρέξουν οι παραπάνω πληροφορίες που υπάρχουν στην μεταλλαγή createNewPost. Μπορούμε τώρα να πάμε στην εφαρμογή μας, να αναζητήσουμε για την τοποθεσία Ποντικονήσι και να δούμε αν θα εμφανιστεί το πόστ που δημιουργήσαμε.

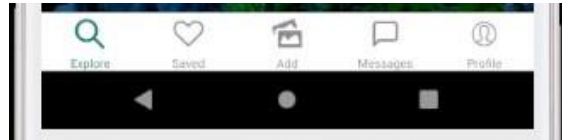


Εικόνα 4. 14 Mapography Αναζήτηση του πόστ

Κάνοντας μία αναζήτηση στην εφαρμογή Mapography βλέπουμε πως το πόστ που δημιουργήσαμε προστέθηκε επιτυχώς στην εφαρμογή μας, όπως φαίνεται και στην παραπάνω εικόνα. Είμαστε λοιπόν σίγουροι ότι το AWS Appsync προστέθηκε επιτυχώς στην εφαρμογή μας και είναι πλήρως λειτουργικό.

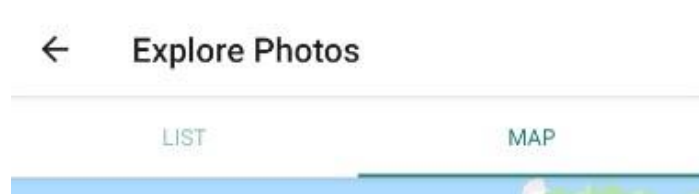
4.6 Λειτουργίες της εφαρμογής Mapography

Η δημιουργία της εφαρμογής Mapography έγινε με τη χρήση του Expo το οποίο αναλύθηκε στην **Ενότητα 4.4**, και σε αυτή την ενότητα θα παρουσιαστεί η δομή της. Κατά τη διάρκεια της ανάπτυξης της εφαρμογής, η δομή της άλλαξε πολλές φορές μέχρι να καταλήξει στην τελευταία της μορφή. Υπήρξαν αρκετές αρχικές ιδέες οι οποίες υλοποιήθηκαν, αλλά υπήρξαν και αρκετές οι οποίες είτε υλοποιήθηκαν χωρίς να είναι 100% λειτουργικές, μαζί με κάποιες οι οποίες δεν υλοποιήθηκαν καθόλου. Καταρχάς για να μπορέσει να λειτουργήσει η εφαρμογή, θα πρέπει να έχουμε σύνδεση στο διαδίκτυο. Εφόσον είμαστε συνδεδεμένοι στο διαδίκτυο και ανοίξουμε την εφαρμογή, εάν είναι η πρώτη φορά που την ανοίγουμε, θα πρέπει να συνδεθούμε με κάποιον λογαριασμό. Στην περίπτωση που δεν έχουμε κάποιον λογαριασμό, έχουμε τη δυνατότητα να δημιουργήσουμε καινούριο λογαριασμό με username, μία διεύθυνση e-mail, έναν κωδικό και ένα κινητό τηλέφωνο. Βάζοντας τα απαραίτητα στοιχεία που χρειάζονται για τη δημιουργία λογαριασμού, γίνεται αποστολή ενός κωδικού στο e-mail προκειμένου να επιβεβαιωθεί η διεύθυνση. Μόλις ολοκληρωθούν τα παραπάνω βήματα, ο χρήστης θα είναι σε θέση να συνδεθεί στην εφαρμογή με το username και τον κωδικό πρόσβασης. Όταν ο χρήστης συνδεθεί στην εφαρμογή, το πρώτο πράγμα που θα παρατηρήσει, είναι ότι η αρχική σελίδα αποτελείται από έναν πλοηγό καρτελών (tab navigator) ο οποίος περιέχει πέντε κύριες καρτέλες (**εικόνα 4.15**).



Εικόνα 4. 15 Καρτέλα πλοήγησης της εφαρμογής Marography

Η κάθε μία από τις πέντε επιλογές του πλοηγού είναι και μία διαφορετική οθόνη(screen), και η κάθε οθόνη αποτελείται από το δικό της κομμάτι κώδικα και τον δικό της σχεδιασμό. Στην πρώτη επιλογή του πλοηγού καρτελών είναι το explore. Σε αυτήν την οθόνη ο χρήστης έχει τη δυνατότητα να αναζητήσει τοποθεσίες μέσω του google places API και του google Auto-complete και στη συνέχεια θα εμφανιστούν τα αποτελέσματα της αναζήτησης είτε σε μία λίστα είτε στον χάρτη. Για την εμφάνιση των αποτελεσμάτων χρησιμοποιείται ακόμα ένας πλοηγός ο οποίος έχει δύο μόνο επιλογές και φαίνεται στην **Εικόνα 4.16**.

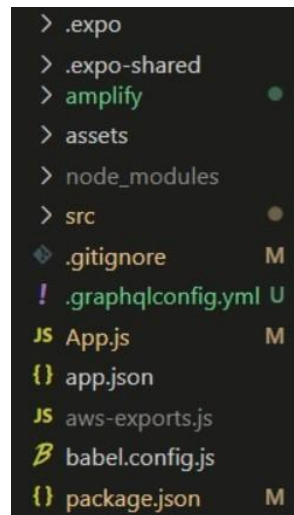


Εικόνα 4. 16 Πλοηγός Αποτελεσμάτων της εφαρμογής Marography

Η επόμενη επιλογή του πλοηγού της **Εικόνας 4.15** ονομάζεται “Saved” και επιλέγοντάς το, ο χρήστης μεταφέρεται σε μία οθόνη στην οποία μπορεί να δει τις αγαπημένες του φωτογραφίες ή σημεία με σκοπό να έχει μία πιο εύκολη πρόσβαση σε αυτά. Στην Τρίτη επιλογή του πλοηγού έχουμε την προσθήκη φωτογραφιών. Σε αυτή την οθόνη ο χρήστης έχει τη δυνατότητα να προσθέσει τις φωτογραφίες που έχει στη συσκευή του με σκοπό να μπορούν άλλοι χρήστες να τις δουν και ενδεχομένως να ανακαλύψουν μία καινούρια τοποθεσία. Μαζί με την τοποθεσία που τραβήχτηκε αυτή η φωτογραφία υπάρχει και η δυνατότητα να προσθέσει και μία περιγραφή σχετική με την φωτογραφία ή την τοποθεσία. Η επόμενη καρτέλα είναι τα μηνύματα μεταξύ χρηστών. Τα μηνύματα και η καρτέλα με τις αποθηκευμένες φωτογραφίες υπάρχουν στην εφαρμογή και έχουν υλοποιηθεί όσον αφορά το front-end κομμάτι, αλλά δεν είναι λειτουργικές. Όπως είπαμε και στην αρχή της ενότητας κάποιες ιδέες υλοποιήθηκαν και υπάρχουν στην εφαρμογή χωρίς όμως να είναι 100% λειτουργικές. Η τελευταία καρτέλα του πλοηγού περιέχει το προφίλ του χρήστη, και μέσω αυτής της καρτέλας, ο χρήστης που είναι συνδεδεμένος στην εφαρμογή μπορεί να αποσυνδεθεί πατώντας το κουμπί “log out” και θα μεταφερθεί στην οθόνη σύνδεσης. Αυτές οι 5 επιλογές που υπάρχουν στον πλοηγό καρτελών μαζί με τις λειτουργίες τους αποτελούν την τελική μορφή της δομής της εφαρμογής Marography.

4.6.1 Το Ευρετήριο της εφαρμογής Mapography

Το ευρετήριο του project αποτελείται από ένα πλήθος φακέλων και αρχείων που περιέχουν είτε κώδικα είτε πόρους όπως εικόνες, και .json αρχεία. Για να είναι πιο ευανάγνωστο το ευρετήριο του project, εκτός από το App.js το οποίο είναι το πρώτο αρχείο που δημιουργείται όπως είδαμε και στο **Κεφάλαιο 4.4**, όλα τα υπόλοιπα αρχεία, έχουν τοποθετηθεί σε κατάλληλους φακέλους. Μπορούμε να δούμε τη δομή του ευρετηρίου στην **εικόνα 4.17** και θα εξηγηθούν όλοι οι φάκελοι που υπάρχουν καθώς και η χρησιμότητά τους. Αρχικά έχουμε τους πρώτους δύο φακέλους .expo και .expo-shared. Αυτοί οι δύο φάκελοι έχουν δημιουργηθεί από το expo, περιέχουν κάποια αρχεία με πληροφορίες για την εφαρμογή και είναι απαραίτητοι προκειμένου να μπορέσει να τρέξει σωστά. Ύστερα έχουμε τον φάκελο amplify, ο οποίος δημιουργήθηκε αυτόματα όταν έγινε η προσθήκη του Backend στο project. Στις προηγούμενες Ενότητες είδαμε πως εγκαταστάθηκε καθώς και πως λειτουργεί το Backend της εφαρμογής και είπαμε ότι χρησιμοποιήθηκε η AWS Amplify.



Εικόνα 4. 17 Ευρετήριο της εφαρμογής Mapography

Κάτω από τον φάκελο amplify, βρίσκεται ο φάκελος assets ο οποίος περιέχει τα logo της εφαρμογής καθώς και κάποιες άλλες φωτογραφίες που χρησιμοποιήθηκαν στο εσωτερικό της. Ο φάκελος node-modules περιέχει όλους τους φακέλους και αρχεία τα οποία έχουν εγκατασταθεί με τη χρήση του npm (node project manager). Ο src φάκελος περιέχει όλα τα αρχεία τα οποία καθορίζουν την εμφάνιση αλλά και τη σωστή λειτουργία της εφαρμογής (**εικόνα 4.18**).



Εικόνα 4. 18 Ο src φάκελος του ευρετηρίου

Ο διαχωρισμός του κώδικα σε υπό φακέλους έγινε με κύριο στόχο την πιο εύκολη εύρεση κάποιου αρχείου αλλά και για να είναι πιο ευανάγνωστο και εύκολα κατανοήσιμο όλο το project. Έχουμε λοιπόν τέσσερις υπό φακέλους:

- **τα στοιχεία (components):** Σε αυτόν τον φάκελο βρίσκονται όλα τα αρχεία κώδικα που αφορούν κάποια εμφανισιακά στοιχεία της εφαρμογής. Ένα από αυτά τα αρχεία που βρίσκονται εδώ είναι υπεύθυνο για το πώς εμφανίζεται κάποιο πόστ στην οθόνη του κινητού.
- **το GraphQL:** αυτός ο φάκελος δημιουργήθηκε αυτόματα με την προσθήκη του AWS Appsync και περιέχει κάποια αρχεία για τη σωστή λειτουργία του GraphQL.
- **την πλοήγηση (navigation):** Στον φάκελο της πλοήγησης έχουμε τα αρχεία κώδικα που καθιστούν δυνατή την πλοήγηση στις οθόνες της εφαρμογής.
- **τις οθόνες (screens):** Εδώ βρίσκονται τα αρχεία κώδικα που αφορούν όλες τις οθόνες της εφαρμογής μας.

Πηγαίνοντας ξανά στην **Εικόνα 4.8**, μετά τον φάκελο src έχουμε το αρχείο gitignore. Σε περίπτωση που χρησιμοποιούμε το git και ανεβάζουμε τα αρχεία μας για παράδειγμα στο GitHub, μπορούμε μέσα στο gitignore να προσθέσουμε φακέλους ή αρχεία τα οποία δεν επιθυμούμε να συμπεριληφθούν στο ανέβασμα. Κάτω από το gitignore βρίσκεται το αρχείο graphqlconfig.yml, το οποίο αφορά τη διαμόρφωση του GraphQL. Προχωρώντας παρακάτω στο ευρετήριο έχουμε το App.js, το οποίο περιέχει τον κώδικα που θα τρέξει την πρώτη φορά που θα ανοίξουμε την εφαρμογή. Στο App.js υπάρχει και ο κώδικας για την διαδικασία ταυτοποίησης των χρηστών. Επόμενο στη σειρά βρίσκεται το app.json που έχει κάποιες πληροφορίες για την εφαρμογή. Τα αρχεία aws-exports.js και babel.js περιέχουν κώδικα που αφορά το Backend της εφαρμογής και τέλος το package.json περιέχει τις πληροφορίες, τις εξαρτήσεις (dependencies) και τις εκδόσεις των πακέτων που έχουν εγκατασταθεί για την επέκταση των δυνατοτήτων της εφαρμογής.

4.6.2 Οι εξαρτήσεις (dependencies) που χρησιμοποιεί η εφαρμογή Mapography

Σε προηγούμενες ενότητες είδαμε ξανά τον όρο εξαρτήσεις (dependencies) χωρίς όμως να εξηγηθεί ποια μπορεί να είναι η χρησιμότητά τους. Ας δούμε λοιπόν πρώτα γιατί είναι απαραίτητο να υπάρχουν οι εξαρτήσεις στο project μας και ύστερα θα παρουσιάσουμε τις εξαρτήσεις που έχουν εγκατασταθεί σε αυτό. Οι εξαρτήσεις μίας React Native εφαρμογής, αποτελούν μέρος third - party κώδικα, από τον οποίο εξαρτάται η εφαρμογή. Όπως στον πραγματικό κόσμο ένα παιδί εξαρτάται από τους γονείς έτσι και η εφαρμογή μας εξαρτάται από αυτόν τον κώδικα που έχουν γράψει τρίτοι. Με τη εγκατάσταση των εξαρτήσεων μειώνουμε τους περιορισμούς και μπορούμε να προσθέσουμε περισσότερες λειτουργίες στην εφαρμογή μας οι οποίες χωρίς αυτές δεν θα ήταν υλοποιήσιμες. Η διαδικασία εγκατάστασης μίας εξάρτησης στο project είναι αρκετά απλή. Η εγκατάσταση γίνεται μέσω του npm (node project manager) και αφού ολοκληρωθεί, μπορούμε να τη χρησιμοποιήσουμε στην εφαρμογή μας. Παρακάτω στην **Εικόνα 4.19** έχουμε όλες τις εξαρτήσεις που έχουν εγκατασταθεί και χρησιμοποιούνται από την εφαρμογή Mapography.

```

"dependencies": {
  "@react-native-community/masked-view": "0.1.10",
  "@react-native-community/netinfo": "5.9.7",
  "@react-native-community/viewpager": "4.2.0",
  "@react-navigation/bottom-tabs": "^5.11.8",
  "@react-navigation/material-top-tabs": "^5.3.14",
  "@react-navigation/native": "^5.9.3",
  "@react-navigation/stack": "^5.14.3",
  "aws-amplify": "^3.3.27",
  "aws-amplify-react-native": "^4.3.2",
  "expo": "~40.0.0",
  "expo-image-picker": "~9.2.0",
  "expo-location": "~10.0.0",
  "expo-permissions": "~10.0.0",
  "expo-status-bar": "~1.0.3",
  "react": "16.13.1",
  "react-dom": "16.13.1",
  "react-native": "https://github.com/expo/react-native/archive/sdk-40.0.1.tar.gz",
  "react-native-elements": "^3.3.2",
  "react-native-gesture-handler": "~1.8.0",
  "react-native-google-places-autocomplete": "^2.2.0",
  "react-native-keyboard-aware-scroll-view": "^0.9.3",
  "react-native-maps": "0.27.1",
  "react-native-pager-view": "^5.1.2",
  "react-native-reanimated": "~1.13.0",
  "react-native-safe-area-context": "3.1.9",
  "react-native-screens": "~2.15.2",
  "react-native-tab-view": "^3.0.0",
  "react-native-vector-icons": "^8.1.0",
  "react-native-web": "~0.13.12"
},

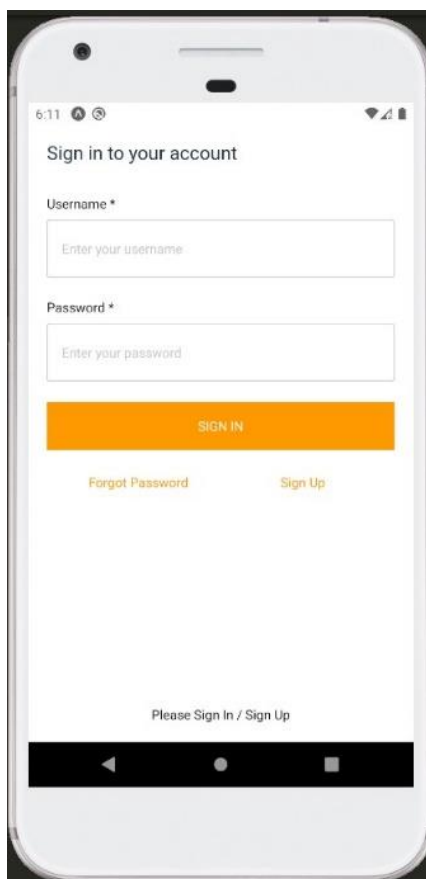
```

Εικόνα 4. 19 Εξαρτήσεις της εφαρμογής Mapography

Κεφάλαιο 5. Παρουσίαση της εφαρμογής Mapography

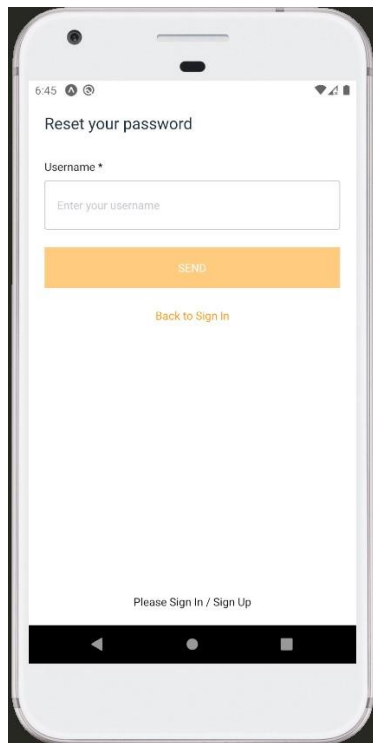
5.1 Είσοδος ή Εγγραφή του χρήστη στο Σύστημα

Ανοίγοντας πρώτη φορά την εφαρμογή Mapography, για να μπορέσουμε να χρησιμοποιήσουμε τις λειτουργίες τις, θα πρέπει να δημιουργήσουμε έναν λογαριασμό ή εφόσον έχουμε δημιουργήσει ήδη, να συνδεθούμε με αυτόν. Η πρώτη οθόνη που θα δούμε ανοίγοντας την εφαρμογή, φαίνεται στην παρακάτω εικόνα 5.1:

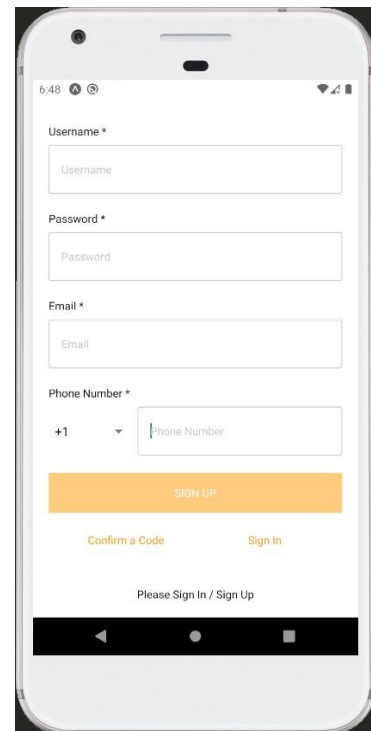


Εικόνα 5. 1 Είσοδος στην εφαρμογή

Εφόσον έχουμε ήδη κάποιον λογαριασμό, σε αυτήν την οθόνη θα πρέπει να βάλουμε το **username** και τον **κωδικό πρόσβασης (password)**. Σε περίπτωση που έχουμε ξεχάσει τον κωδικό πρόσβασης και δεν μπορούμε να συνδεθούμε στην εφαρμογή, θα πρέπει να πατήσουμε το κουμπί **Forgot Password** το οποίο θα μας φέρει στην σελίδα επαναφοράς κωδικού (εικόνα 5.2). Εάν όμως δεν έχουμε κάποιον λογαριασμό, τότε θα πρέπει να δημιουργήσουμε πατώντας το κουμπί **Sign Up**, το οποίο θα μας μεταφέρει στην σελίδα δημιουργίας λογαριασμού (εικόνα 5.3).



Εικόνα 5. 3 Επαναφορά κωδικού

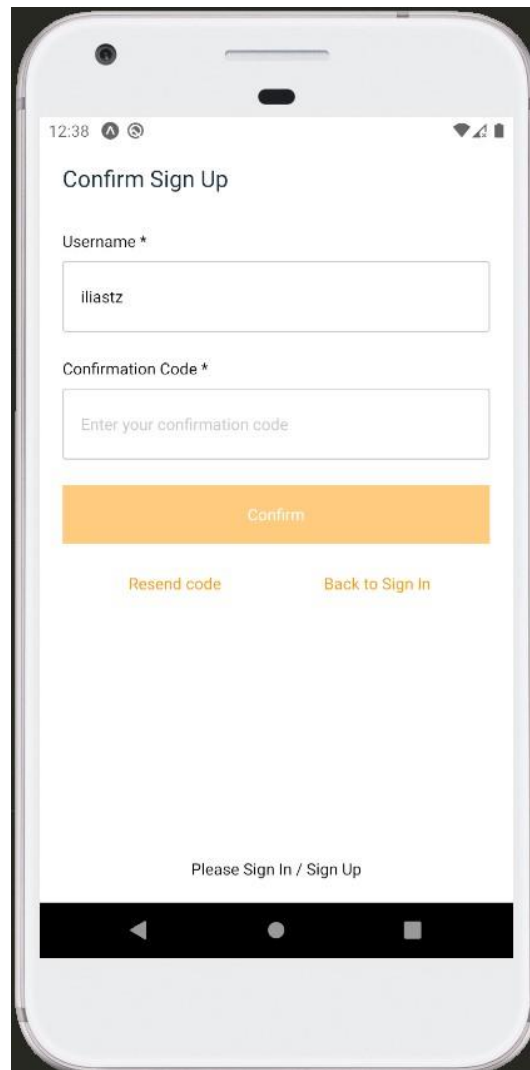


Εικόνα 5. 2 Δημιουργία λογαριασμού

Στην **εικόνα 5.2** φαίνεται η σελίδα επαναφοράς κωδικού. Για να επαναφέρουμε τον κωδικό θα πρέπει να γράψουμε το **username** με το οποίο έχουμε εγγραφεί στην εφαρμογή και ύστερα πατώντας το κουμπί **Send**, θα λάβουμε ένα e-mail το οποίο θα μας καθοδηγήσει για να επαναφέρουμε τον κωδικό. Στην **εικόνα 5.3** φαίνεται η οθόνη εγγραφής νέου χρήστη. Για να εγγραφεί κάποιος στην εφαρμογή θα πρέπει να δώσει τις παρακάτω πληροφορίες:

- **Username:** Το username θα πρέπει να είναι μοναδικό και να μην χρησιμοποιείται από κάποιον άλλον χρήστη. Ως username μπορούμε να βάλουμε είτε αριθμούς είτε χαρακτήρες.
- **Κωδικός πρόσβασης (Password):** Για κωδικό πρόσβασης μπορούμε να βάλουμε ότι θέλουμε αρκεί να είναι πάνω από 8 ψηφία. Γενικά προτείνεται στους κωδικούς μας να βάζουμε εκτός από χαρακτήρες και αριθμούς ή σύμβολα για μεγαλύτερη ασφάλεια.
- **Διεύθυνση Ηλεκτρονικού ταχυδρομείου (E-mail):** Εκτός από το username και τον κωδικό πρόσβασης, θα πρέπει να προστεθεί και μία διεύθυνση ηλεκτρονικού ταχυδρομείου. Μία διεύθυνση ηλεκτρονικού ταχυδρομείου μπορεί να χρησιμοποιηθεί μόνο για την δημιουργία ενός λογαριασμού. Σε περίπτωση που θέλουμε να δημιουργήσουμε διαφορετικό λογαριασμό στην εφαρμογή θα πρέπει να προσθέσουμε άλλη διεύθυνση σε αυτό το πεδίο.
- **Τηλεφωνικός αριθμός:** Σε αυτό το πεδίο πρέπει να προσθέσουμε έναν αριθμό τηλεφώνου. Αυτό το πεδίο δεν έχει κάποια χρησιμότητα και υπάρχει στο προκαθορισμένο μοντέλο που παρέχει το Amazon Cognito. Εάν δεν θέλουμε να βάλουμε πραγματικό αριθμό τηλεφώνου μπορούμε να βάλουμε κάποιον ψεύτικο.

Όταν προσθέσουμε όλα τα απαραίτητα στοιχεία που αναφέραμε πιο πάνω, τότε θα μεταφερθούμε στην οθόνη όπου θα πρέπει να πληκτρολογήσουμε έναν κωδικό ο οποίος θα σταλεί στην ηλεκτρονική διεύθυνση που καταχωρήσαμε. Όπως βλέπουμε και στην εικόνα 5.4, εκτός από τον κωδικό έχουμε ακόμα τη δυνατότητα να επεξεργαστούμε το **username**.

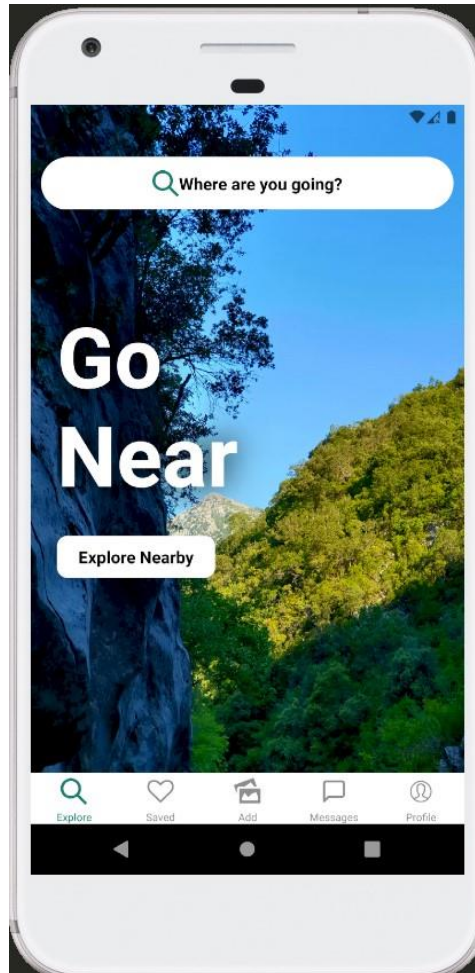


Εικόνα 5. 4 Επιβεβαίωση δημιουργίας λογαριασμού

Αφού ολοκληρώσουμε και την παραπάνω διαδικασία μπορούμε να πατήσουμε το κουμπί **Confirm** και αυτό θα μας μεταφέρει ξανά στην αρχική οθόνη όπως φαίνεται στην εικόνα 5.1. Εκεί τώρα θα μπορούμε να πληκτρολογήσουμε το **username** και τον **κωδικό πρόσβασης** που μόλις δημιουργήσαμε και έτσι θα μπορούμε να έχουμε πρόσβαση στην αρχική οθόνη της εφαρμογής.

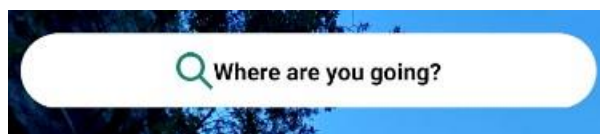
5.2 Αρχική οθόνη της εφαρμογής

Η αμέσως επόμενη οθόνη που θα δει ο χρήστης μόλις εισέλθει στην εφαρμογή είναι η αρχική οθόνη (εικόνα 5.5)



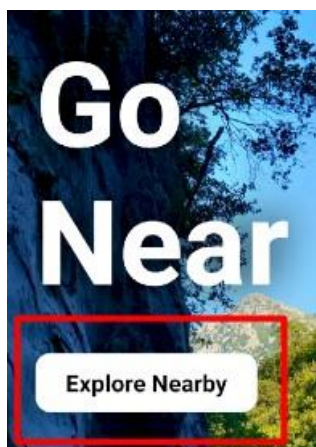
Εικόνα 5. 5 Αρχική οθόνη εφαρμογής

Στην Αρχική Οθόνη της εφαρμογής έχουμε, αρκετές λειτουργίες τις οποίες θα δούμε μία-μία. Στο επάνω μέρος της εφαρμογής έχουμε την αναζήτηση (εικόνα 5.6). Πατώντας το κουμπί της αναζήτησης ο χρήστης έχει τη δυνατότητα ψάξει και να μεταφερθεί σε μία τοποθεσία της επιλογής του.



Εικόνα 5. 6 Μπάρα Αναζήτησης

Αμέσως μετά την αναζήτηση έχουμε το κουμπί για την εξερεύνηση κοντινών σημείων (**Explore Nearby**) το οποίο φαίνεται και στην εικόνα 5.7. Με αυτή την επιλογή, ο χρήστης έχει τη δυνατότητα να δει διάφορα σημεία τα οποία βρίσκονται σε κοντινή απόσταση.



Εικόνα 5. 7 Κοντινές τοποθεσίες

Η τελευταία επιλογή που έχει ο χρήστης είναι η καρτέλα πλοήγησης (εικόνα 5.8). Η καρτέλα πλοήγησης είναι διαθέσιμη και προσβάσιμη από όλες τις οθόνες και επιτρέπει στον χρήστη να επιλέξει αυτή που θέλει να χρησιμοποιήσει τη δεδομένη στιγμή. Οι πέντε επιλογές που προσφέρει ο πλοηγός της εφαρμογής έχουν αναλυθεί πλήρως στην **Ενότητα 4.6** και σε αυτό το κεφάλαιο θα γίνει η παρουσίαση της κάθε μία οθόνης από αυτές τις επιλογές.



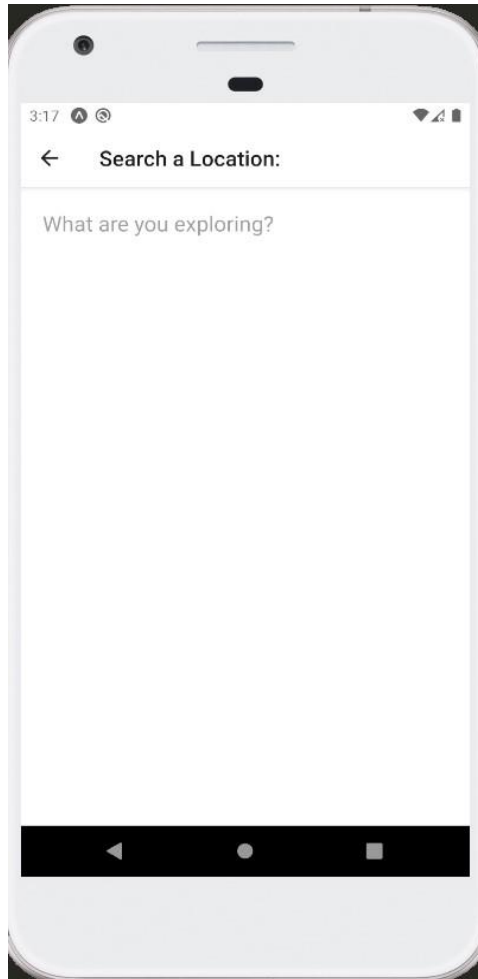
Εικόνα 5. 8 Καρτέλα πλοήγησης

Συνοψίζοντας η αρχική οθόνη της εφαρμογής Mapography είναι μία αρκετά λιτή οθόνη με έναν πλοηγό καρτελών και δύο βασικές επιλογές, την αναζήτηση και την εξερεύνηση. Κάποιοι από τους λόγους που επιλέχθηκε αυτή η προσέγγιση και δεν τοποθετήθηκαν περισσότερες λειτουργίες στην αρχική οθόνη είναι:

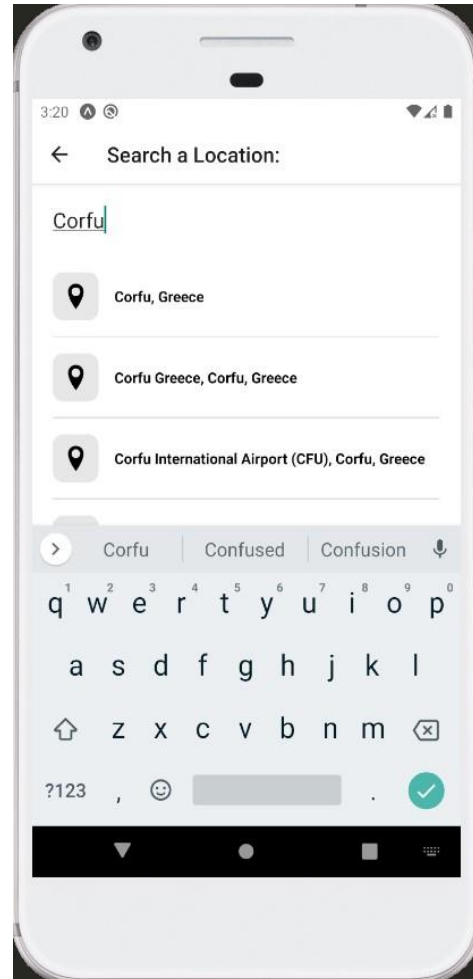
- Δεν υπάρχει ανάγκη προσθήκης επιπρόσθετων λειτουργιών στην αρχική οθόνη, καθώς αυτό μπορεί να αυξήσει την πολυπλοκότητά της εφαρμογής και να μπερδέψει του χρήστες.
- Ήδη αυτές οι δύο επιλογές είναι αρκετές και επιτρέπουν ακόμα και σε έναν πιο απαιτητικό χρήστη να βρει αυτό που ψάχνει.

5.2.1 Λειτουργία αναζήτησης τοποθεσιών

Όταν θέλουμε να πραγματοποιήσουμε κάποια αναζήτηση για να βρούμε μία τοποθεσία, θα πρέπει αρχικά να πατήσουμε στο κουμπί της αρχικής οθόνης το οποίο φαίνεται στην εικόνα 5.6. Έστερα θα εμφανιστεί μία καινούρια οθόνη όπως φαίνεται στην παρακάτω εικόνα 5.9:



Εικόνα 5. 10 Σελίδα αναζήτησης

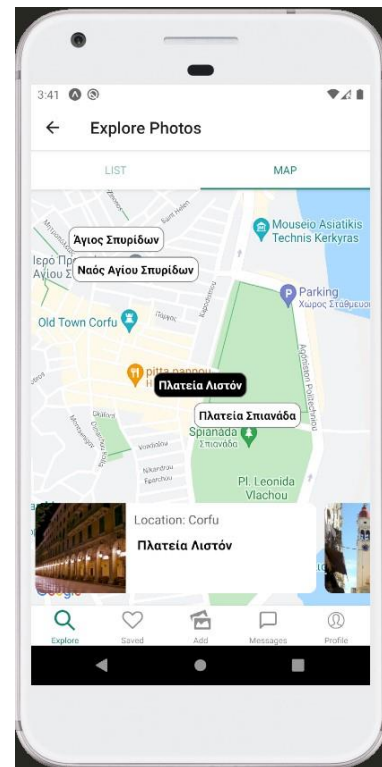


Εικόνα 5. 9 Auto-Complete

Σε αυτή την οθόνη μπορούμε να πληκτρολογήσουμε οποιαδήποτε τοποθεσία θέλουμε να αναζητήσουμε και κατά τη διάρκεια της πληκτρολόγησης θα εμφανίζονται από κάτω προτάσεις σχετικές με αυτό που έχουμε γράψει (εικόνα 5.10). Μόλις επιλέξουμε την τοποθεσία που θέλουμε να αναζητήσουμε, θα μεταφερθούμε στην οθόνη αποτελεσμάτων. Σε αυτή την οθόνη μπορούμε να δούμε τα αποτελέσματα της αναζήτησης είτε με τη μορφή λίστας είτε στον χάρτη. Τα αποτελέσματα της λίστας μπορούμε να δούμε πως φαίνονται στην εικόνα 5.11. Αρχικά έχουμε μία φωτογραφία από την τοποθεσία η οποία έχει προστεθεί από κάποιον χρήστη. Μετά την φωτογραφία ακολουθεί η ονομασία της τοποθεσίας, ένας τίτλος και τέλος μία περιγραφή σχετική με αυτήν την τοποθεσία. Στην εικόνα 5.12 βλέπουμε πως φαίνονται τα αποτελέσματα πάνω στον χάρτη. Έχουμε τα αποτελέσματα τοποθετημένα στο σημείο όπου βρίσκονται οι τοποθεσίες με το όνομά τους και από κάτω υπάρχει μία μικρή λίστα με αυτά τα αποτελέσματα.

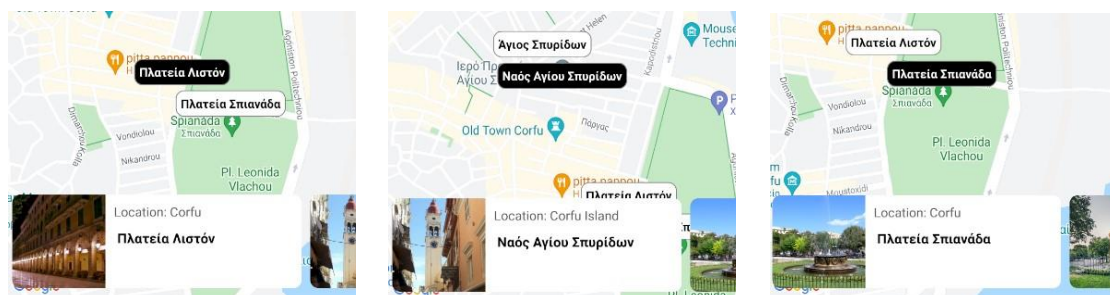


Εικόνα 5. 12 Αποτελέσματα σε λίστα



Εικόνα 5. 11 Αποτελέσματα στον χάρτη

Στην **εικόνα 5.13** φαίνεται πως λειτουργεί η λίστα που υπάρχει στο κάτω μέρος του χάρτη. Το κάθε στοιχείο της λίστας περιέχει την φωτογραφία της τοποθεσίας, την τοποθεσία στην οποία βρίσκεται και το όνομα του σημείου που βλέπουμε εκείνη τη στιγμή. Επίσης όταν αλλάζουμε το σημείο της λίστας και μεταφερόμαστε στο επόμενο, τότε ο χάρτης επικεντρώνεται στο σημείο το οποίο αλλάξαμε. Το τελευταίο πράγμα που θα παρατηρούσε κάποιος στον χάρτη της εφαρμογής είναι πως το σημείο το οποίο είναι επιλεγμένο έχει μαύρο φόντο με άσπρα γράμματα, ενώ τα υπόλοιπα σημεία εκεί κοντά έχουν άσπρο φόντο με μαύρα γράμματα.

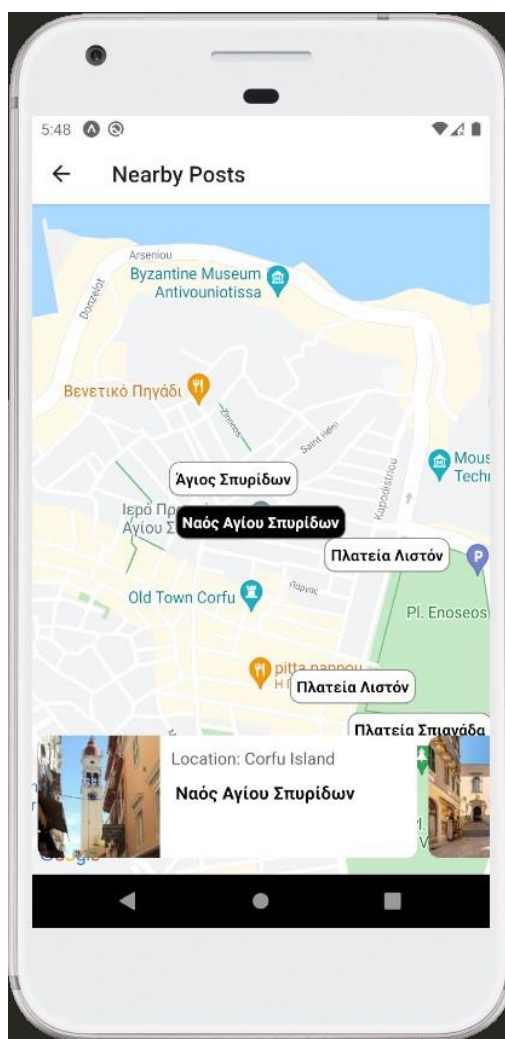


Εικόνα 5. 13 Στοιχείο της λίστας του χάρτη

Είδαμε λοιπόν πως γίνεται η διαδικασία αναζήτησης μίας τοποθεσίας στην εφαρμογή Mapography. Στην επόμενη ενότητα θα δούμε τι συμβαίνει όταν επισκεφθούμε την οθόνη με τις κοντινές τοποθεσίες.

5.2.2 Λειτουργία κοντινών τοποθεσιών

Η δεύτερη λειτουργία που υπάρχει στη Αρχική Οθόνη της εφαρμογής είναι η εξερεύνηση των κοντινών τοποθεσιών (Explore Nearby) το οποίο φαίνεται και στην εικόνα 5.7. Εφόσον επιλέξουμε να δούμε αυτές τις τοποθεσίες, θα ανοίξει μία νέα οθόνη η οποία φαίνεται στην εικόνα 5.14.



Εικόνα 5. 14 Κοντινές τοποθεσίες στον χάρτη

Σε αυτή τη λειτουργία τα αποτελέσματα αποτυπώνονται με λίγο διαφορετικό τρόπο σε σχέση με τα αποτελέσματα της αναζήτησης. Αρχικά δεν υπάρχει ο διαχωρισμός σε μορφή λίστας ή σε μορφή χάρτη. Τα αποτελέσματα υπάρχουν μόνο στον χάρτη και η διάταξη είναι ακριβώς ίδια με την του χάρτη της αναζήτησης. Η μικρή λίστα με τα αποτελέσματα διατηρείται και σε αυτόν τον σχεδιασμό έχοντας όμως μία μικρή διαφορά. Το κάθε αντικείμενο αυτής της λίστας είναι και ένα κουμπί, το οποίο μπορούμε να πατήσουμε και έτσι θα μεταφερθούμε σε μία καινούρια οθόνη. Αυτή η οθόνη περιέχει όλες τις πληροφορίες από το πόστ που θα επιλέξουμε. Αυτές οι πληροφορίες είναι η φωτογραφία, η τοποθεσία, το όνομα της περιοχής και η περιγραφή (εικόνα 5.15).



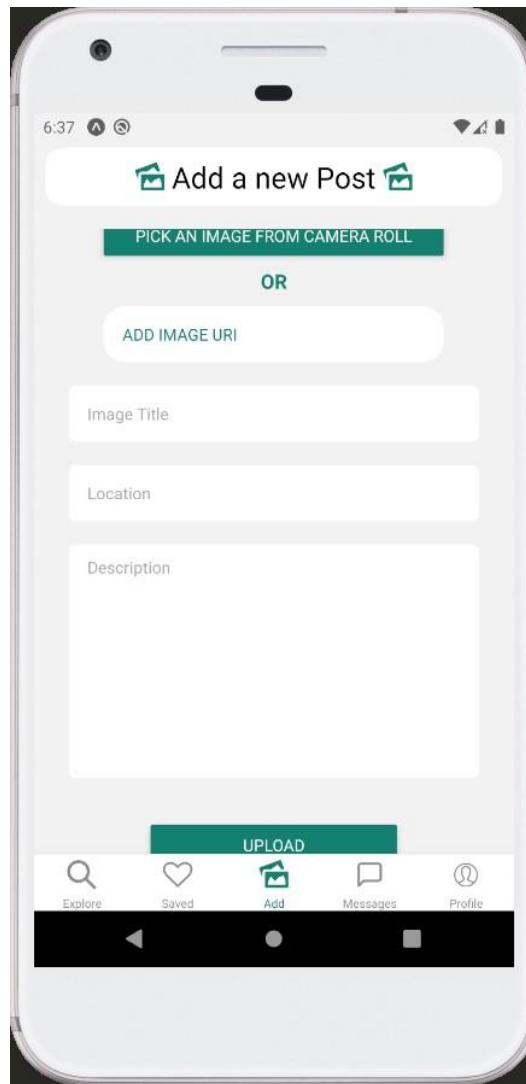
Εικόνα 5. 15 Πόστ κοινωνικών τοποθεσιών

5.3 Οθόνη προσθήκης νέου πόστ

Η επόμενη λειτουργία που υπάρχει στην εφαρμογή, είναι αυτή της προσθήκης ενός πόστ. Αρχικά θα πρέπει να επιλέξουμε από την καρτέλα πλοήγησης το κουμπί **Add**.



Αυτό το κουμπί θα μας μεταφέρει στην οθόνη προσθήκης νέου πόστ, την οποία μπορούμε να δούμε στην **εικόνα 5.16**



Εικόνα 5. 16 Οθόνη προσθήκης νέου πόστ

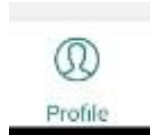
Οι επιλογές που έχουμε σε αυτήν την οθόνη είναι οι εξής:

- **Κουμπί επιλογής εικόνας:** Πατώντας αυτό το κουμπί και εφόσον παραχωρήσουμε το δικαίωμα πρόσβασης στις εικόνες της συσκευής, θα μπορούμε να επιλέξουμε μία από τις φωτογραφίες που υπάρχουν στη συσκευή.
- **Επιλογή για Image URI:** Εάν δεν θέλουμε να προσθέσουμε κάποια εικόνα από τη συσκευή μας, υπάρχει και η δυνατότητα να προσθέσουμε το URI από μία φωτογραφία που υπάρχει στο διαδίκτυο.
- **Image title:** Είναι ένα πλαίσιο κειμένου όπου πρέπει να προστεθεί ο τίτλος της εικόνας.
- **Location:** Σε αυτό το πλαίσιο κειμένου πρέπει να προστεθεί η τοποθεσία που τραβήχτηκε η εικόνα
- **Description:** Το τελευταίο πλαίσιο κειμένου το οποίο αφορά τις πληροφορίες για αυτήν την τοποθεσία που θέλει να προσθέσει ο χρήστης.
- **Upload:** Μόλις πατήσουμε το κουμπί **upload**, θα γίνει η προσθήκη του πόστ στην εφαρμογή.

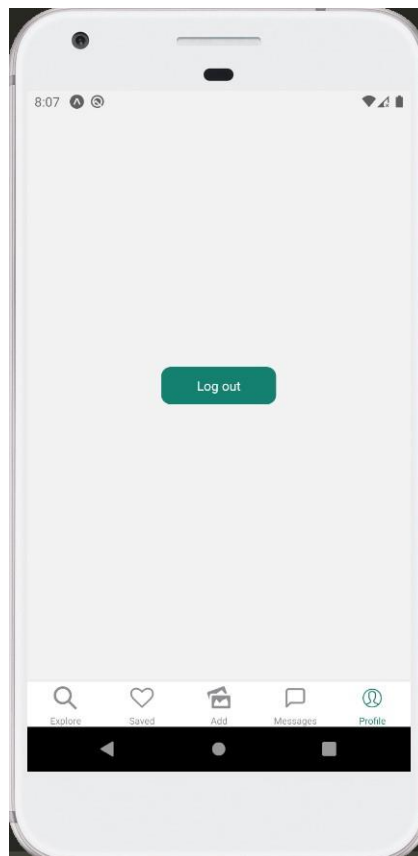
Ο χρήστης θα πρέπει να δώσει όλες τις παραπάνω πληροφορίες προκειμένου να μπορέσει να ανεβάσει ένα πόστ καθώς είναι απαραίτητες. Όταν ολοκληρώσει την προσθήκη, τότε το πόστ θα είναι διαθέσιμο και θα μπορούν να το δουν όλοι οι χρήστες της εφαρμογής.

5.4 Οθόνη Αποσύνδεσης

Η τελευταία λειτουργία της εφαρμογής Mapography είναι αυτή της αποσύνδεσης του χρήστη. Καταρχάς πρέπει να επιλέξουμε από την καρτέλα πλοήγησης το κουμπί **Profile**.



Με αυτό το κουμπί θα μεταφερθούμε στην οθόνη αποσύνδεσης όπως φαίνεται και στην **Εικόνα 5.17**. Σε αυτήν την οθόνη έχουμε μόνο το κουμπί **Logout** και εάν το πατήσουμε θα μας μεταφέρει στην οθόνη σύνδεσης (**εικόνα 5.1**) η οποία αναλύθηκε στην **Ενότητα 5.1**.



Εικόνα 5. 17 Οθόνη αποσύνδεσης

Κεφάλαιο 6. Αξιολόγηση της εφαρμογής

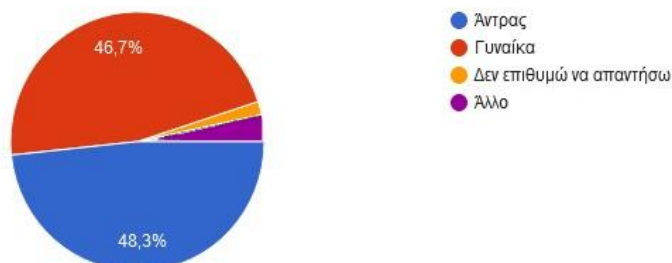
6.1 Παρουσίαση του ερωτηματολογίου

Κατά την διάρκεια της ανάπτυξης της εφαρμογής, δημιουργήθηκε ένα ερωτηματολόγιο με σκοπό να γίνει μία αξιολόγηση τόσο για τις τεχνολογίες που χρησιμοποιήθηκαν, αλλά και για το περιεχόμενο της εφαρμογής που αφορά την ένδειξη τουριστικών αξιοθέατων. Το ερωτηματολόγιο αυτό δημιουργήθηκε μέσω των Google Forms και χωρίστηκε σε τρεις διαφορετικές ενότητες. Οι συνολικές ερωτήσεις που περιέχει είναι 19 και ο χρόνος που χρειάζεται για να απαντηθούν είναι περίπου 5 λεπτά. Το ερωτηματολόγιο απαντήθηκε από 60 άτομα και το feedback που λάβαμε για τις λειτουργίες της εφαρμογής ήταν αρκετά θετικό. Σε αυτό το κεφάλαιο θα γίνει παρουσίαση των απαντήσεων που δόθηκαν στο ερωτηματολόγιο με τη μορφή πίτας και διαγραμμάτων τα οποία δημιουργήθηκαν αυτόματα από το Google Forms.

6.1.2 Δημογραφικά στοιχεία

Η πρώτη ενότητα του ερωτηματολογίου αφορούσε τα δημογραφικά στοιχεία των ατόμων που το απαντήσαν. Για την πρώτη ερώτηση (εικόνα 6.1) που αφορά το φύλλο έχουμε την παρακάτω πίτα με τα εξής στοιχεία:

1) Φύλο
60 απαντήσεις



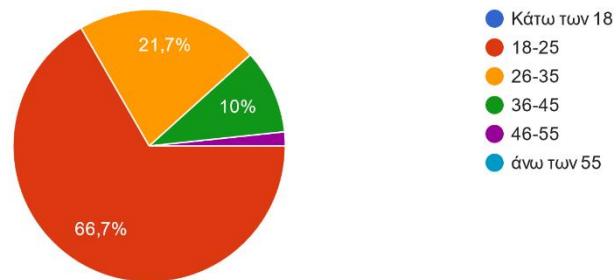
Εικόνα 6. 1 Ερωτηματολόγιο Ερώτηση 1

Το 48,3% των συμμετεχόντων ήταν Άντρες και το 46,7% ήταν Γυναίκες. Έχουμε όμως και ένα 2,5% το οποίο δεν επιθυμούσε να απαντήσει καθώς και ένα ποσοστό 2,5% το οποίο έδωσε την απάντηση «Άλλο». Η δεύτερη και τελευταία ερώτηση που υπάρχει στα Δημογραφικά στοιχεία αφορά την Ηλικία των συμμετεχόντων. Σε αυτήν την ερώτηση οι απαντήσεις ήταν σε μορφή ηλικιακού εύρους και διαμορφώθηκαν ως εξής:

- Κάτω των 18
- 18-25
- 26-35
- 36-45
- 46-55
- Άνω των 55

Όπως φαίνεται και στην παρακάτω πίτα το μεγαλύτερο ποσοστό από τους συμμετέχοντες ανήκει στην ηλικιακή ομάδα των 18-25 με ποσοστό 66,7%. Αμέσως μετά έχουμε τις ηλικίες **(εικόνα 6.2)** 26-35 και το ποσοστό αυτών ανέρχεται στο 21,7%. Στο 10% ήταν οι ηλικίες από 36-45 και στο 1,7% ήταν οι ηλικίες 46-55. Στο ερωτηματολόγιο αυτό δεν λάβαμε καμία απάντηση από κάποιο άτομο κάτω των 18 ή άνω των 55 ετών.

2) Ηλικία
60 απαντήσεις

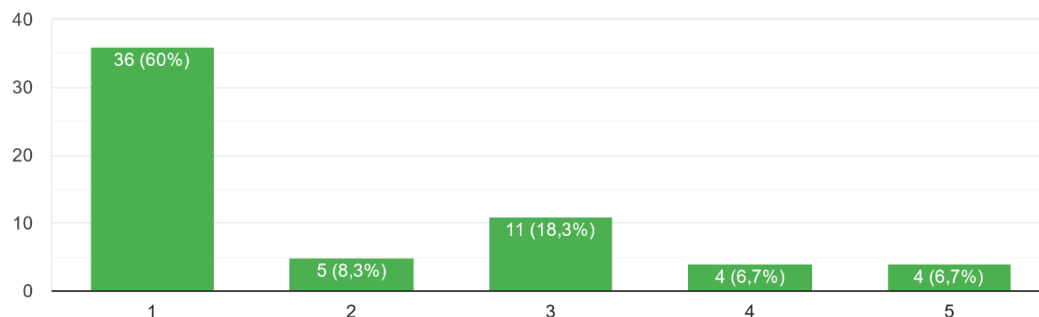


Εικόνα 6. 2 Ερωτηματολόγιο Ερώτηση 2

6.1.2 Mobile Crowdsensing

Η επόμενη ενότητα του ερωτηματολογίου αφορά την τεχνική Mobile Crowdsensing. Σε αυτή την ενότητα θέλουμε να ερευνήσουμε πόσοι από τα άτομα που απάντησαν το ερωτηματολόγιο είναι εξοικειωμένοι με αυτήν την τεχνολογία καθώς και πόσοι την θεωρούν χρήσιμη. Για όσους δεν γνώριζαν την τεχνική Mobile Crowdsensing δόθηκε και ένας ορισμός με σκοπό να τους βοηθήσει να απαντήσουν τις ερωτήσεις. Η πρώτη ερώτηση **(εικόνα 6.3)** είναι η εξής:

3) Πόσο εξοικειωμένοι είστε με τον όρο Mobile Crowdsensing;
60 απαντήσεις



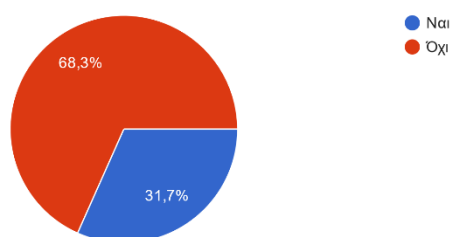
Εικόνα 6. 3 Ερωτηματολόγιο Ερώτηση 3

Οι απαντήσεις είναι σε μορφή κλίμακας, όπου το 1 σημαίνει καθόλου εξοικείωση με τον όρο και το 5 σημαίνει μεγάλη εξοικείωση με τον όρο. Όπως φαίνεται και στο διάγραμμα οι απαντήσεις που δόθηκαν ήταν οι εξής:

1. **Καθόλου εξοικείωση:** Το 60% των ατόμων δεν είναι καθόλου εξοικειωμένο με τον όρο Mobile rowdsensing και ενδεχομένως τον συναντά για πρώτη φορά.
2. **Σχεδόν καθόλου εξοικείωση:** Το 6,3% απάντησε πως τείνει να μην είναι εξοικειωμένο με τον όρο.
3. **Μέτρια εξοικείωση:** Το 18,3% από τα άτομα απάντησε πως είναι μέτρια εξοικειωμένο.
4. **Αρκετή εξοικείωση:** Το 6,7% έχει αρκετή εξοικείωση.
5. **Μεγάλη εξοικείωση:** Το υπόλοιπο 6,7% απάντησε πως έχει μεγάλη εξοικείωση.

Η επόμενη ερώτηση (**εικόνα 6.4**) είναι:

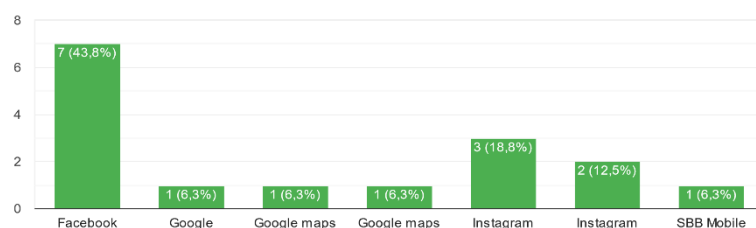
4) Γνωρίζετε ότι αρκετές εφαρμογές κοινωνικών δικτύων χρησιμοποιούν το Mobile Crowdsensing προκειμένου να αντλήσουν δεδομένα από τους χρήστες τους;
60 απαντήσεις



Εικόνα 6. 4 Ερωτηματολόγιο Ερώτηση 4

Σε αυτή την πίτα παρατηρούμε πως οι περισσότεροι απάντησαν «Όχι» με ποσοστό 68,3%. Μόνο το 31,7% γνωρίζει πως αρκετές εφαρμογές κοινωνικών δικτύων χρησιμοποιούν το Mobile Crowdsensing. Στα άτομα που γνώριζαν για την χρήση του στα κοινωνικά δίκτυα έγινε και η παρακάτω ερώτηση (**εικόνα 6.5**):

5) Εάν ναι γράψτε μία εφαρμογή που γνωρίζετε ή μπορεί να έχετε χρησιμοποιήσει.
16 απαντήσεις



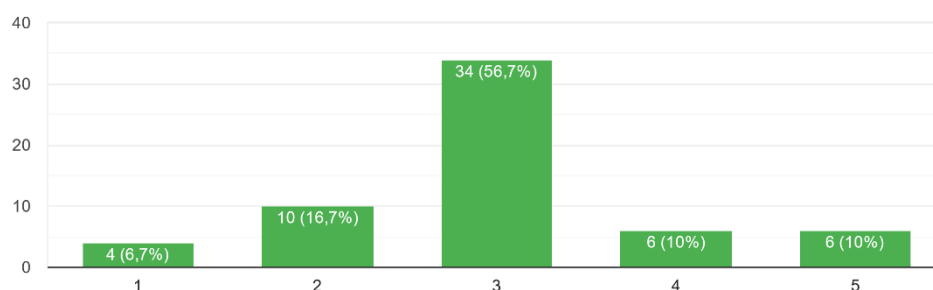
Εικόνα 6. 5 Ερωτηματολόγιο Ερώτηση 5

Οι εφαρμογές που έδωσαν ως απάντηση ήταν:

- Το Facebook: 43,8% , 7 άτομα.
- Το Instagram: 31,3% , 5 άτομα.
- Το Google Maps: 12,6% , 2 άτομα
- Το Google: 6,3% , 1 άτομο
- Το SBB Mobile: 6,3% , 1 άτομο

Στην επόμενη ερώτηση (**εικόνα 6.6**) οι χρήστες έπρεπε να απαντήσουν εάν πιστεύουν πως είναι καλό οι εφαρμογές να χρησιμοποιούν τεχνικές Mobile Crowdsensing.

6)Πιστεύετε πως είναι καλό οι εφαρμογές κινητών συσκευών να χρησιμοποιούν τεχνικές Mobile Crowdsensing;
60 απαντήσεις



Εικόνα 6. 6 Ερωτηματολόγιο Ερώτηση 6

Για απάντηση έχουμε πάλι την κλίμακα από το 1 έως το 5 , με το 1 να σημαίνει καθόλου καλό και το 5 πολύ καλό.

1. **Καθόλου καλό:** Οι 4 από του 60 (6,7%) πιστεύουν πως δεν είναι καθόλου καλό να χρησιμοποιείται το Mobile Crowdsensing στις εφαρμογές.
2. **Σχεδόν καθόλου καλό:** Οι 10 από τους 60 (16,7%) πιστεύουν πως μάλλον δεν είναι καλό να χρησιμοποιούνται.
3. **Ουδέτερη στάση:** Οι 34 από τους 60 (56,7%) κρατήσαν μία ουδέτερη στάση απαντώντας πως δεν είναι ούτε καλό ούτε κακό.
4. **Αρκετά καλό:** Οι 6 στους 60 (10%) είπαν πως είναι αρκετά καλό.
5. **Πολύ καλό:** Οι 6 στους 60 (10%) απάντησαν πως είναι πολύ καλό.

Η τελευταία ερώτηση (**εικόνα 6.7**) αυτής της ενότητας απευθύνεται στα άτομα τα οποία απάντησαν πως η χρήση τεχνικών Mobile Crowdsensing σε μία εφαρμογή δεν είναι καλή. Όπως φαίνεται και παρακάτω λάβαμε 7 γραπτές απαντήσεις. Τα περισσότερα άτομα τα οποία πιστεύουν πως η χρήση αυτής της τεχνολογίας είναι κακή, θεωρούν ότι υπάρχει κίνδυνος με τα προσωπικά μας δεδομένα. Ότι τα χρησιμοποιούνε δηλαδή προς όφελος τρίτων και ότι τα

πουλάνε. Άλλη μία σημαντική απάντηση είναι και η πρώτη, η οποία λέει ότι αντλούν τις προσωπικές πληροφορίες μας με σκοπό την καλύτερη διαφήμιση και προώθηση των προϊόντων.

7) Εάν όχι, γιατί;

7 απαντήσεις

Τις περισσότερες φορές αντλούν προσωπικές πληροφορίες σχετικές με προτιμήσεις προϊόντων μέσω της αναζήτησης και προωθώντας όμοια προϊόντα τακτικά επεμβαίνουν στις καταναλωτικές ανάγκες μας.

Άντληση προσωπικών δεδομένων/χρήση προς όφελος τρίτων

Θα ήθελα να γνωρίζω ότι οι προτιμήσεις μου αποτελούν στοιχεία έρευνας

Δεν γνωρίζω τι είναι

Η κατασκοπεία δεν είναι καλή. Θέλουν να μας ελέγχουν

Πουλάνε τα δεδομένα μας

Δεν είναι καλό για τα προσωπικά δεδομένα μας

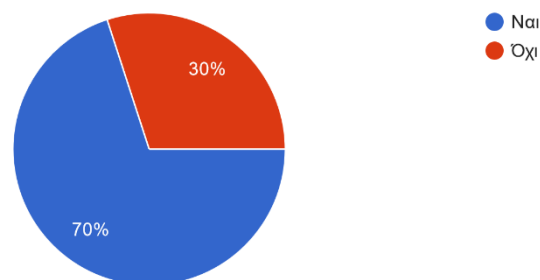
Εικόνα 6. 7 Ερωτηματολόγιο Ερώτηση 7

6.1.3 Εφαρμογές με φωτογραφίες τουριστικών περιοχών

Η τελευταία ενότητα του Ερωτηματολογίου περιέχει ερωτήσεις που είναι σχετικές με την εφαρμογή Mapography. Η πρώτη ερώτηση (**εικόνα 6.8**) φαίνεται παρακάτω:

8) Έχετε χρησιμοποιήσει στο παρελθόν κάποια εφαρμογή η οποία περιέχει φωτογραφίες και πληροφορίες από τουριστικές περιοχές;

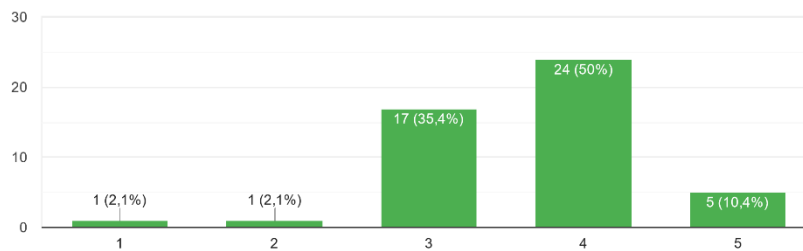
60 απαντήσεις



Εικόνα 6. 8 Ερωτηματολόγιο Ερώτηση 8

Ένα μεγάλο ποσοστό (70%) από τα άτομα που απαντήσανε είπε πως έχει χρησιμοποιήσει στο παρελθόν κάποια εφαρμογή με φωτογραφίες και πληροφορίες από τουριστικές περιοχές. Το υπόλοιπο 30% απάντησε πως δεν έχει χρησιμοποιήσει μία τέτοια εφαρμογή. Στη συνέχεια τα άτομα ερωτήθηκαν πόσο ευχαριστημένοι μέινανε από αυτή την εφαρμογή (**εικόνα 6.9**):

9) Εάν ναι, πόσο ευχαριστημένοι μέναιτε από αυτή την εφαρμογή;
48 απαντήσεις



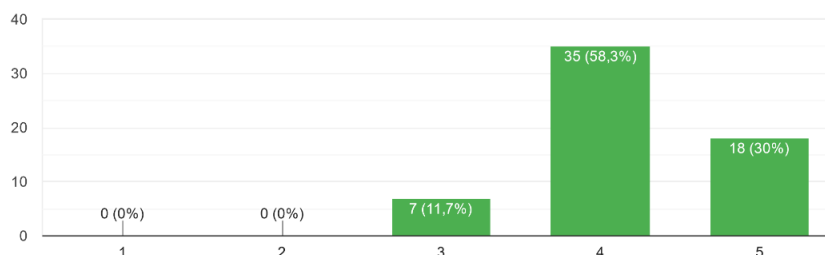
Εικόνα 6. 9 Ερωτηματολόγιο Ερώτηση 9

Σύμφωνα με το παραπάνω διάγραμμα τα περισσότερα άτομα μείνανε αρκετά ευχαριστημένα καθώς οι απαντήσεις διαμορφώθηκαν ως εξής:

1. **Καθόλου ευχαριστημένος:** Το 2,1% των απαντήσεων (1 άτομο) απάντησε πως δεν έμεινε καθόλου ευχαριστημένο με την εφαρμογή.
2. **Σχεδόν καθόλου ευχαριστημένος:** Το 2,1% των απαντήσεων (1 άτομο) απάντησε πως δεν έμεινε σχεδόν καθόλου ευχαριστημένο με την εφαρμογή.
3. **Ουδέτερη στάση:** Το 35,4% (17 άτομα) κράτησαν μία ουδέτερη στάση
4. **Ευχαριστημένος:** Το 50% (24 άτομα) απάντησε πως μείνανε αρκετά ευχαριστημένα.
5. **Πολύ ευχαριστημένος:** Το 10% (5 άτομα) απάντησε πως έμεινε πολύ ευχαριστημένο από την εφαρμογή που χρησιμοποίησε.

Η επόμενη ερώτηση (**εικόνα 6.10**) φαίνεται παρακάτω:

10) Πόσο χρήσιμη θα σας φαινόταν μία εφαρμογή με την οποία οι χρήστες μπορούν να αναδείξουν τουριστικές περιοχές;
60 απαντήσεις



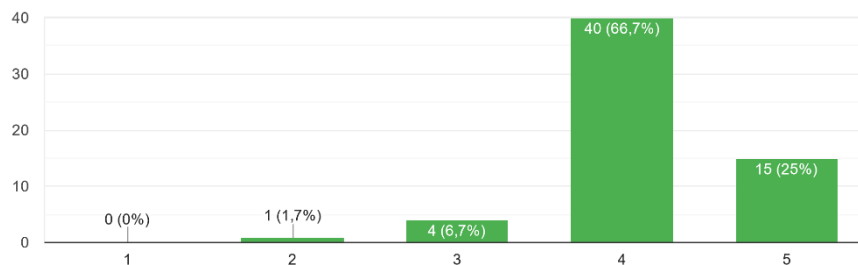
Εικόνα 6. 10 Ερωτηματολόγιο Ερώτηση 10

Μόνο τα 7 άτομα από τα 60 κράτησαν ουδέτερη στάση σε αυτή την ερώτηση, ενώ οι υπόλοιποι 53 απάντησαν θετικά. Συγκεκριμένα:

- το 11,7% απάντησε ουδέτερα
- το 58,3% απάντησε «Χρήσιμη»
- 30% απάντησε «Πολύ χρήσιμη»

Στην επόμενη ερώτηση (**εικόνα 6.11**) τα άτομα έπρεπε να απαντήσουν κατά πόσο θα χρησιμοποιούσαν μία εφαρμογή όπως το Marography εάν ήταν επισκέπτες σε μία περιοχή

11) Ως επισκέπτης σε μία περιοχή θα χρησιμοποιούσατε μία τέτοια εφαρμογή προκειμένου να ανακαλύψετε μία νέα τοποθεσία/ες;
60 απαντήσεις



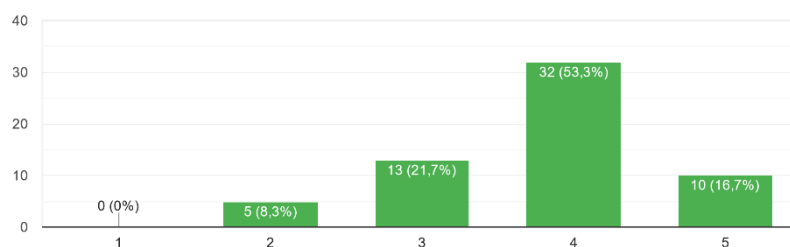
Εικόνα 6. 11 Ερωτηματολόγιο Ερώτηση 11

Αρκετά θετικές ήταν και οι απαντήσεις σε αυτήν την ερώτηση καθώς:

- 1 άτομο στα 60 (1,7%) απάντησε πως μάλλον δεν θα χρησιμοποιούσε μία τέτοια εφαρμογή.
- 4 άτομα (6,7 %) κράτησαν ουδέτερη στάση.
- 40 άτομα (66,7%) είπαν πως θα την χρησιμοποιούσαν αρκετά.
- 15 άτομα (25%) είπαν πως θα την χρησιμοποιούσαν συνέχεια.

Ύστερα έχουμε την ερώτηση (**εικόνα 6.12**) η οποία ήταν ακριβώς ίδια, με την μόνη αλλαγή στο ότι δεν θα χρησιμοποιούμε την εφαρμογή ως επισκέπτες αλλά ως κάτοικοι μίας περιοχής.

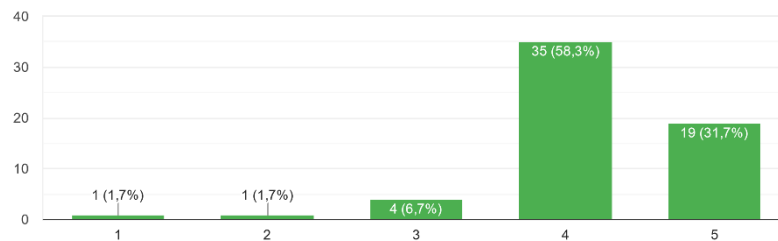
12) Ως κάτοικος μίας περιοχής θα χρησιμοποιούσατε μία τέτοια εφαρμογή προκειμένου να αναδείξετε κάποια τοποθεσία/ες;
60 απαντήσεις



Εικόνα 6. 12 Ερωτηματολόγιο Ερώτηση 12

Όπως βλέπουμε από το διάγραμμα οι απαντήσεις σε αυτή την ερώτηση είναι λίγο διαφορετικές σε σχέση με τις απαντήσεις της προηγούμενης ερώτησης. Πάλι η πλειοψηφία είπε πως θα την χρησιμοποιούσε αρκετά, αλλά αυξήθηκαν οι απαντήσεις στην ουδέτερη στάση και στην απάντηση «Δεν θα τη χρησιμοποιούσα αρκετά». Το συμπέρασμα που βγήκε από αυτές τις δύο ερωτήσεις είναι πως οι χρήστες θα προτιμήσουν να κάνουν χρήση μίας εφαρμογής όπως το Mapography, όταν πρόκειται να επισκεφτούν μία νέα τοποθεσία, αλλά όταν πρόκειται για το μέρος που κατοικούν ίσως να μην χρησιμοποιήσουν στον ίδιο βαθμό την εφαρμογή. Η επόμενη ερώτηση που θα δούμε είναι η εξής (**εικόνα 6.13**):

13) Πιστεύετε ότι θα μπορούσε μία τέτοια εφαρμογή να αναδείξει περιοχές οι οποίες δεν είναι πολύ διαδεδομένες;
60 απαντήσεις



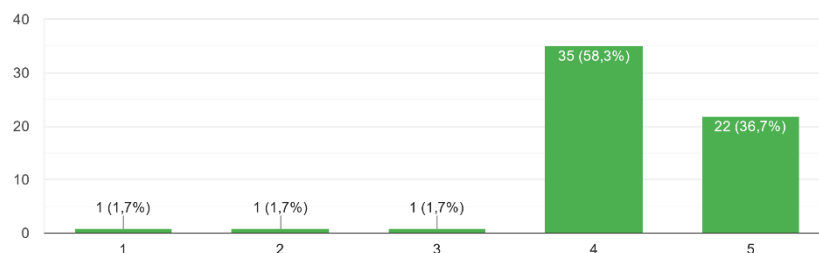
Εικόνα 6. 13 Ερωτηματολόγιο Ερώτηση 13

Το μεγαλύτερο ποσοστό των ατόμων πιστεύει πως μία εφαρμογή όπως το Mapography θα μπορέσει να αναδείξει περιοχές που δεν είναι πολύ διαδεδομένες. Συγκεκριμένα:

- Μόνο 2 άτομα (3,4%) πιστεύουν πως δεν θα μπορέσει να αναδείξει κάποια τοποθεσία
- 4 άτομα (6,7%) απάντησαν ουδέτερα
- 54 άτομα (90%) είπαν ότι μία τέτοια εφαρμογή θα μπορέσει να αναδείξει μη διαδεδομένες περιοχές.

Για την επόμενη ερώτηση (**εικόνα 6.14**), πήραμε τις παρακάτω απαντήσεις:

14) Πιστεύετε πως θα σας βοηθούσε να ανακαλύψετε νέα αξιοθέατα που δεν γνωρίζατε ότι υπήρχαν;
60 απαντήσεις

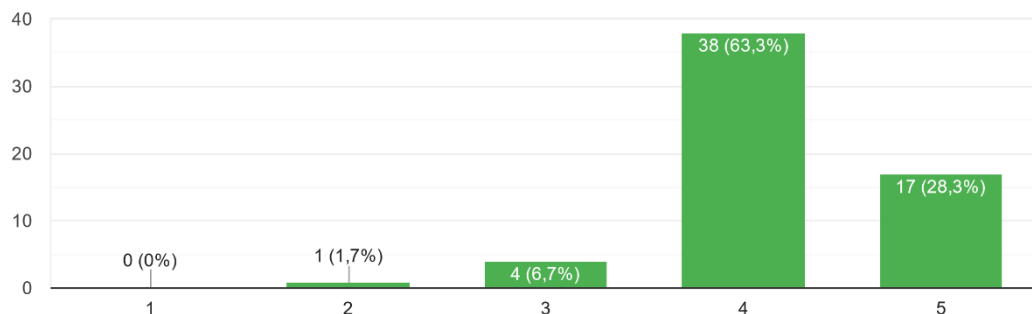


Εικόνα 6. 14 Ερωτηματολόγιο Ερώτηση 14

- **Δεν θα βοηθούσε καθόλου:** 1 άτομο (1,7%) πιστεύει πως δεν θα βοηθούσε καθόλου.
- **Δεν θα βοηθούσε αρκετά:** 1 άτομο (1,7%) πιστεύει πως δεν θα βοηθούσε αρκετά.
- **Ουδέτερη απάντηση:** 1 άτομο (1,7%) κράτησε ουδέτερη στάση.
- **Θα βοηθούσε:** 35 άτομα (58,3%) απάντησαν πως θα βοηθούσε.
- **Θα βοηθούσε αρκετά:** 22 άτομα απάντησαν ότι θα βοηθούσε αρκετά

Η επόμενη ερώτηση (**εικόνα 6.15**) αφορά μία λειτουργία της εφαρμογής καθώς ρωτάει τα άτομα, πως θα τους φαινόταν μέσω της αναζήτησης μίας τοποθεσίας να μπορούν να δουν σχετικά αξιοθέατα:

15) Θα σας φαινόταν χρήσιμο να μπορείτε να αναζητήσετε μία περιοχή και να δείτε τα διάφορα τουριστικά αξιοθέατα που έχουν δημοσιευθεί σε αυτήν από άλλους χρήστες;
60 απαντήσεις



Εικόνα 6. 15 Ερωτηματολόγιο Ερώτηση 15

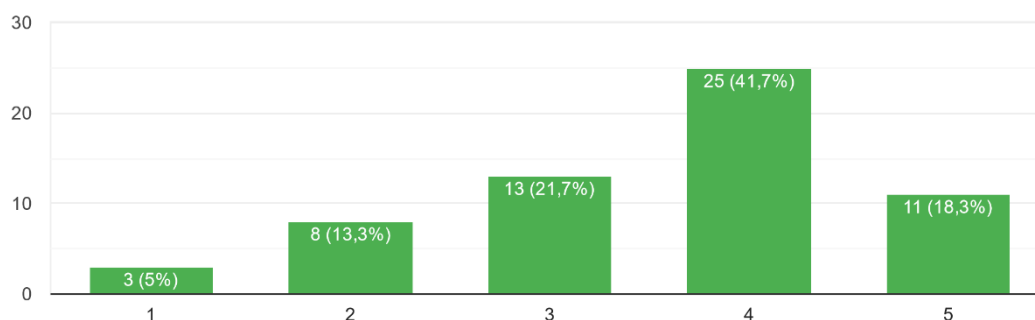
Και εδώ οι απαντήσεις που δόθηκαν είναι αρκετά θετικές καθώς:

- 1 άτομο (1,7%) πιστεύει πως η εμφάνιση αξιοθέατων μέσω της αναζήτησης μάλλον δεν θα ήταν χρήσιμη.
- 4 άτομα (6,7%) έδωσαν μία ουδέτερη απάντηση.
- 38 άτομα (63,3%) πιστεύουν πως η εμφάνιση αξιοθέατων μέσω της αναζήτησης θα ήταν χρήσιμη.
- 17 άτομα (28,3%) απάντησαν πως αυτή η λειτουργία θα ήταν πολύ χρήσιμη.

Οι απαντήσεις που πήραμε από την παρακάτω ερώτηση (**εικόνα 6.16**) ήταν πάλι θετικές. Αυτή τη φορά τα άτομα έπρεπε να απαντήσουν εάν θα δημοσίευαν φωτογραφίες τουριστικών περιοχών σε μία εφαρμογή με σκοπό να τις ανακαλύψουν άλλοι χρήστες.

16) Θα δημοσιεύατε φωτογραφίες τουριστικών περιοχών σε μία εφαρμογή με σκοπό να τις ανακαλύψουν άλλοι χρήστες;

60 απαντήσεις



Εικόνα 6. 16 Ερωτηματολόγιο Ερώτηση 16

- 3 άτομα (5%) απάντησαν πως δεν θα δημοσίευαν καμία φωτογραφία στην εφαρμογή.
- 8 άτομα (13,3%) απάντησαν πως μάλλον δεν θα δημοσίευαν φωτογραφίες.
- 13 άτομα (21,7%) έδωσαν μία ουδέτερη απάντηση.
- 25 άτομα (41,7%) είπαν πως θα δημοσίευαν αρκετές φωτογραφίες στην εφαρμογή.
- 11 άτομα (18,3%) απάντησαν πως θα δημοσίευαν συνέχεια

Από τα άτομα που απάντησαν πως μάλλον δεν θα δημοσίευαν φωτογραφία ή δεν θα δημοσίευαν καθόλου, ζητήθηκε να γράψουν μερικούς λόγους που δεν θα το κάνανε **(εικόνα 6.17)**.

17) Εάν όχι, γιατί;

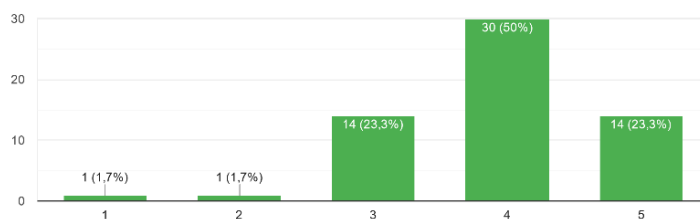
6 απαντήσεις

Den mu aresi i tecnologia
Δεν θέλω να δημοσιεύω φωτογραφίες
Δεν μου αρέσει να δημοσιεύω φωτογραφίες σε εφαρμογές
Isos thn krataga gia ton eafto mou kai thn pareia mou san mustikh hremh topothesia gia xalarosh
Αδράνεια
Δεν έχω όρεξη. Βαριέμαι

Εικόνα 6. 17 Ερωτηματολόγιο Ερώτηση 17

Οι συνολικές απαντήσεις αυτής της ερώτησης ήταν 6 και έγινε κατανοητό πως τα περισσότερα άτομα δεν θα ήθελαν να δημοσιεύσουν κάποια φωτογραφία στην εφαρμογή επειδή το θεωρούν βαρετό, δεν τους αρέσει η δημοσίευση φωτογραφιών και δεν θα επιθυμούσαν να διαδοθεί μία τοποθεσία που τους αρέσει. Η ερώτηση **(εικόνα 6.18)** που ακολουθεί αφορά την άποψη που έχουν άλλοι χρήστες για μία τοποθεσία και μπορούμε παρακάτω να δούμε τις απαντήσεις:

18) Θα θέλατε να ξέρετε τι άποψη έχουν άλλοι χρήστες της εφαρμογής για μία τοποθεσία ή περιοχή που σας ενδιαφέρει να επισκεφθείτε;
60 απαντήσεις

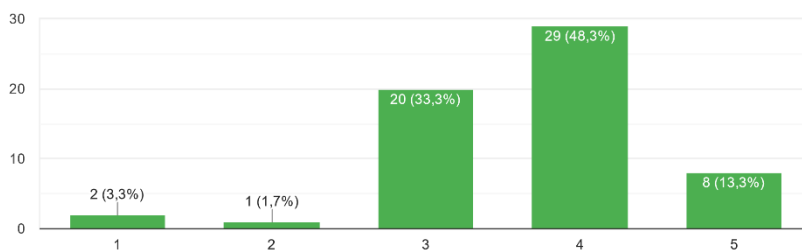


Εικόνα 6. 18 Ερωτηματολόγιο Ερώτηση 18

- 1 άτομο (1,7%) απάντησε ότι δεν τον ενδιαφέρει καθόλου η άποψη των άλλων χρηστών.
- 1 άτομο (1,7%) απάντησε ότι μάλλον δεν τον ενδιαφέρει η άποψη των άλλων χρηστών.
- 14 άτομα (23,3%) έδωσαν ουδέτερη απάντηση.
- 30 άτομα (50%) απάντησαν πως θα ήθελαν να ξέρουν τι άποψη έχουν οι άλλοι χρήστες.
- 14 άτομα (23,3%) απάντησαν πως σίγουρα θα ήθελαν να ξέρουν τι άποψη έχουν οι άλλοι χρήστες.

Η τελευταία ερώτηση **(εικόνα 6.19)** που είχε το ερωτηματολόγιο έχει να κάνει με το πόσο θα επηρεαζόταν ένα άτομο από την άποψη που έχουν οι άλλοι χρήστες της εφαρμογής.

19) Πιστεύετε ότι η άποψη άλλων χρηστών για μία τοποθεσία ή περιοχή θα σας επηρέαζε είτε θετικά είτε αρνητικά;
60 απαντήσεις



Εικόνα 6. 19 Ερωτηματολόγιο Ερώτηση 19

Σε αυτήν την ερώτηση πήραμε τις παρακάτω απαντήσεις:

1. **Δεν θα με επηρέαζε καθόλου:** 2 άτομα (3,3%) είπαν πως δεν θα τους επηρέαζε καθόλου η άποψη άλλων χρηστών για μία τοποθεσία.
2. **Μάλλον δεν θα με επηρέαζε:** 1 άτομο (1,7%) απάντησε πως μάλλον δεν θα τον επηρέαζε η άποψη άλλων χρηστών για μία τοποθεσία.
3. **Ουδέτερη απάντηση:** 20 άτομα (33,3%) έδωσαν ουδέτερη απάντηση
4. **Μάλλον θα με επηρέαζε:** 29 άτομα (48,3%) απάντησαν πως μάλλον θα τους επηρέαζε η άποψη άλλων χρηστών για μία τοποθεσία.
5. **Θα με επηρέαζε πολύ:** 8 άτομα (13,3%) απάντησαν πως θα τους επηρέαζε πολύ η άποψη άλλων χρηστών για μία τοποθεσία.

6.2 Συμπεράσματα από το Ερωτηματολόγιο

Τα συμπεράσματα που βγάλαμε από τις απαντήσεις του ερωτηματολογίου ήταν αρκετά. Αρχικά, θα μπορούσε να πει κάποιος ότι τα περισσότερα άτομα που το απάντησαν δεν είναι καθόλου εξοικειωμένα με το Mobile Crowdsensing. Σίγουρα όμως κάποια στιγμή στη ζωή τους έχουν χρησιμοποιήσει μια εφαρμογή η οποία λειτουργεί με τεχνικές Mobile Crowdsensing. Αρκετοί ήταν και αυτοί οι οποίοι γνωρίζανε τον όρο και έδωσαν και κάποια σωστά παραδείγματα εφαρμογών, όπως το Facebook και το Instagram. Άλλο ένα σημαντικό συμπέρασμα που βγάλαμε από το ερωτηματολόγιο, είναι πως τα περισσότερα άτομα θα ήθελαν να υλοποιηθεί και να χρησιμοποιήσουν μία εφαρμογή με λειτουργίες σαν αυτές του Mapography. Επίσης αρκετά άτομα δήλωσαν πως θα ήθελαν να αναδείξουν ή και να μάθουν για διάφορα αξιοθέατα και τοποθεσίες που μπορεί να μην είναι τόσο διαδεδомένα. Ωστόσο τα περισσότερα άτομα μάλλον θα έκαναν μεγαλύτερη χρήση της εφαρμογής όταν θα βρίσκονταν σε μία τοποθεσία ως επισκέπτες παρά στον κύριο τόπο διαμονής τους. Μία εξίσου σημαντική διαπίστωση ήταν πως μία εφαρμογή όπως το Mapography και θα βοηθήσει να αναδειχτούν περισσότερο κάποια αξιοθέατα, αλλά και θα είναι πιο εύκολα στο να τα ανακαλύψει κάποιο άτομο. Τέλος η πλειοψηφία των ατόμων που απάντησαν το ερωτηματολόγιο υποστήριξαν πως και τους ενδιαφέρει η άποψη των άλλων, αλλά και θα τους επηρέαζε αυτή η άποψη είτε θετικά είτε αρνητικά.

Κεφάλαιο 7. Συμπεράσματα και μελλοντικές επεκτάσεις

7.1 Συμπεράσματα

Η παρούσα πτυχιακή εργασία αποτελεί μία προσπάθεια ανάπτυξης μιας Mobile Crowdsensing εφαρμογής για έξυπνους πολίτες. Η εφαρμογή αυτή προορίζεται, όπως γίνεται κατανοητό και από τον τίτλο για χρήση σε έξυπνες συσκευές, δηλαδή τα Smartphones και τα Tablets. Η διαδικασία που απαιτείται για να αναπτυχθεί μία τέτοια εφαρμογή απαιτεί σίγουρα βασικές γνώσεις προγραμματισμού καθώς και μία εξοικείωση με τις τεχνολογίες που χρησιμοποιούνται. Το demo που δημιουργήθηκε περιέχει κάποιες βασικές λειτουργίες και προσφέρει τη δυνατότητα στους χρήστες να αλληλοεπιδρούν στέλνοντας δεδομένα από τοποθεσίες στο σύστημα ή να ενημερώνονται για αυτές μέσω της αναζήτησης. Σύμφωνα και με την αξιολόγηση που έγινε μία εφαρμογή όπως το Mapography θα βοηθούσε τους χρήστες της παρέχοντας λεπτομερείς για τις τοποθεσίες και θα ήταν σε θέση να τις αναδείξει. Σίγουρα πολλοί επηρεαζόμαστε από τις απόψεις των υπολοίπων οπότε θα μας ενδιέφερε να μπορούμε να τις δούμε μέσω αυτής της εφαρμογής. Συμπεραίνουμε λοιπόν ότι αξίζει να αναπτυχθεί πλήρως μία τέτοια εφαρμογή, για αυτό και στην παρακάτω ενότητα θα δούμε κάποιες μελλοντικές επεκτάσεις που μπορούν να γίνουν.

7.2 Μελλοντικές επεκτάσεις

Η εφαρμογή που υλοποιήθηκε μπορεί να βελτιωθεί μελλοντικά εφόσον γίνει προσθήκη των ακόλουθων χαρακτηριστικών:

- Να γίνει υλοποίηση των οθονών Saved και Messages. Όπως είδαμε στο Κεφάλαιο 5 η υλοποίηση για την οθόνη των αποθηκευμένων φωτογραφιών και των μηνυμάτων δεν έχει πραγματοποιηθεί. Αυτή η υλοποίηση θα κάνει την εφαρμογή λειτουργικότερη και θα της προσφέρει νέες και χρήσιμες δυνατότητες. Επίσης θα αυξηθεί η αλληλοεπίδραση μεταξύ χρηστών και εφαρμογής.
- Άλλη μία οθόνη που θα μπορούσε να έχει περισσότερες λειτουργίες, είναι και η οθόνη του Προφίλ. Μπορεί να γίνει προσθήκη των πόστ που έχει ανεβάσει ο χρήστης στην οθόνη του προφίλ του. Επίσης θα μπορούσαν να υλοποιηθούν και κάποιες ρυθμίσεις οι οποίες αφορούν το προφίλ, όπως αλλαγή κωδικού πρόσβασης, προσθήκη εικόνας προφίλ και διαγραφή λογαριασμού
- Μπορεί να γίνει υλοποίηση διαγραφής πόστ εφόσον ο χρήστης που το ανέβασε επιθυμεί να το σβήσει για διάφορους του λόγους
- Θα μπορούσε να γίνει υλοποίηση αξιολόγησης για τα πόστ που έχει ανεβάσει ο κάθε χρήστης και έτσι θα μπορεί να καταλάβει κάποιος εάν η φωτογραφία ή η περιγραφή ανταποκρίνεται στην πραγματικότητα.
- Εφόσον προστεθεί και η αξιολόγηση, άλλη μία μελλοντική επέκταση θα ήταν αυτή της ταξινόμησης των πόστ όταν κάνουμε αναζήτηση. Με την ταξινόμηση θα μπορεί

ο χρήστης να επιλέξει τη σειρά που επιθυμεί να δει τα πόστ, για παράδειγμα με βάση την αξιολόγηση ή με βάση τις περισσότερες προβολές.

- Μπορεί να γίνει η υλοποίηση προσθήκης ειδοποιήσεων και όταν ο χρήστης βρίσκεται σε μία τοποθεσία με πολλά αξιοθέατα να ενημερώνεται για αυτά.

Βιβλιογραφία

1. Πόλη. (2020, April 24). Wikipedia. Retrieved from <https://el.wikipedia.org/wiki/%CE%A0%CF%8C%CE%BB%CE%B7>
2. Giffinger R. Fertner C. Kramar H. Kalasek R. Pichler-Milanović N. Meijers E. (2007). *Smart Cities: Ranking of European Medium-Sized Cities*. Vienna, Austria: Centre of Regional Science (SRF), Vienna University of Technology. Retrieved from http://www.smart-cities.eu/download/smart_cities_final_report.pdf.
3. Komninos N. (2008). *Intelligent cities and globalisation of innovation networks*. New York: Routledge. Retrieved from <https://books.google.com/books?hl=en&lr=&id=GjZ8AgAAQBAJ&oi=fnd&pg=PP1&dq=komninos+intelligent+cities+and+globalization&ots=RgNb4XaOBQ&sig=fbuJpMp7AV-lvLqvyR-dOaVewN8>
4. Caragliu A. Del Bo C. Nijkamp P. (2013). *Smart Cities in Europe*. Routledge. Retrieved from <https://www.taylorfrancis.com/chapters/edit/10.4324/9780203076224-20/smart-cities-europe-andrea-caragliu-chiara-del-bo-peter-nijkamp>
5. Christopoulou E., Ringas D., Garofalakis J. (2014) The Vision of the Sociable Smart City. In: Streitz N., Markopoulos P. (eds) *Distributed, Ambient, and Pervasive Interactions*. DAPI 2014. Lecture Notes in Computer Science, vol 8530. Springer, Cham. https://doi.org/10.1007/978-3-319-07788-8_50
6. Shafique K. Khawaja B. A. Sabir F. Qazi S. Mustaqim M. "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," in *IEEE Access*, vol. 8, pp. 23022-23040, 2020, doi: 10.1109/ACCESS.2020.2970118.
7. Γιακούπ Μ. (2019). Ανάπτυξη εφαρμογής Android για Mobile Crowd Sensing σύστημα που σχετίζεται με την παρακολούθηση της υγείας μιας πάσχουσας κοινωνικής ομάδας. Διπλωματική Εργασία. Retrieved from <http://hdl.handle.net/10889/13150>
8. Παλαιοδήμος Σ. (2018-05-17). Έξυπνες πόλεις παρόν και μέλλον. Πτυχιακή Εργασία. Retrieved from <http://oceanis.lib2.uniwa.gr/xmlui/handle/123456789/4337>
9. Zozo H. Hesham Arafat A. (October 2015). *Internet of Things(IoT): Definitions, Challenges and Recent Research Directions*. Project: Internet of Things. Retrieved from https://www.researchgate.net/profile/Zozo_Hassan/publication/320532203_Internet_of_Things_IoT_Definitions_Challenges_and_Recent_Research_Directions/links/59ea1d4ba6fdcccf8b08cc3e/Internet-of-Things-IoT-Definitions-Challenges-and-Recent-Research-Directions.pdf
10. Parul Goyal. Ashok Kumar Sahoo. Tarun Kumar Sharma. Pramod K. Singh. (2021). *Internet of Things: Applications, security, and privacy: A survey* Pages 752-759, ISSN 2214-7853, <https://doi.org/10.1016/j.matpr.2020.04.737>.
11. Capponi A. Fiandrino C. Kantarci B. Foschini L. Kliazovich D. Bouvry P. (2019) "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions, and Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2419-2465, third quarter 2019, doi: 10.1109/COMST.2019.2914030.
12. Jinwei Liu, Haiying Shen, Husnu S. Narman, Wingyan Chung, and Zongfang Lin. (2018). A Survey of Mobile Crowdsensing Techniques: A Critical Component for The Internet of Things. *ACM Trans. Cyber-Phys. Syst.* 2, 3, Article 18 (July 2018), 26 pages. DOI: <https://doi.org/10.1145/3185504>
13. Chatzimilioudis G. Konstantinidis A. Laoudias C. Zeinalipour-Yazti D. (2012) "Crowdsourcing with Smartphones," in *IEEE Internet Computing*, vol. 16, no. 5, pp. 36-44, Sept.-Oct. 2012, doi: 10.1109/MIC.2012.70.

14. Y. Wang Y. Jia X. Jin Q. Ma J. (2015) "Mobile Crowdsourcing: Architecture, Applications, and Challenges," *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, 2015, pp. 1127-1132, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.206.
15. Angelov D. (2010) "Google Street View: Capturing the World at Street Level," in *Computer*, vol. 43, no. 6, pp. 32-38, June 2010, doi: 10.1109/MC.2010.170.
16. Yulin Y. Youngryel R. (2021). Exploring Google Street View with deep learning for crop type mapping. *ISPRS Journal of Photogrammetry and Remote Sensing*. Volume 171. Pages 278-296. ISSN 0924-2716. Retrieved from <https://doi.org/10.1016/j.isprsjprs.2020.11.022>.
17. Mapillary - OpenStreetMap (2021, March 22). Wikipedia. Retrieved from <https://wiki.openstreetmap.org/wiki/Mapillary>
18. KartaView. (2021, May 17). Wikipedia. Retrieved from <https://en.wikipedia.org/wiki/KartaView>
19. Boduch A. (2017). React and React Native. Birmingham – Mumbai: Packt Publishing. Retrieved from https://books.google.gr/books?hl=el&lr=&id=jLkrDwAAQBAJ&oi=fnd&pg=PP1&dq=react+native+&ots=m2XuLv5G6&sig=OXTGvONetlectH1ctsx3KROS6nI&redir_esc=y#v=onepage&q&f=true
20. Eisenman B. (2016, January 19). Learning React Native. O'Reilly. Retrieved from <https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html>
21. Borodinets A. (2019, March 22). Choosing Mobile Hybrid Frameworks: React Native vs Ionic. Medium. Retrieved from <https://medium.com/react-native-or-how-to-simplify-e-banking/choosing-mobile-hybrid-frameworks-react-native-vs-ionic-2d1ef69e1666>
22. Kremer M. Comparing Cross-Platform Frameworks. Ionic (2021). Retrieved from <https://ionicframework.com/resources/articles/ionic-vs-react-native-a-comparison-guide>
23. Fentaw. Eshetu A. (2020) Cross platform mobile application development: a comparison study of React Native Vs Flutter. Retrieved from <https://jyx.jyu.fi/handle/123456789/70969>
24. Danielsson, W. (2016). React Native application development : A comparison between native Android and React Native (Dissertation). Retrieved from <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-131645>
25. Visual Studio Code. (2021, June 6). Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Visual_Studio_Code
26. Node.js Introduction. (2021). w3schools. Retrieved from https://www.w3schools.com/nodejs/nodejs_intro.asp
27. NPM (2021, June 8). Wikipedia. Retrieved from [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))
28. Newton M. (2018, March 5). Understanding Expo for React Native. Hackernoon. Retrieved from <https://hackernoon.com/understanding-expo-for-react-native-7bf23054bbcd>
29. Brentvatne. (2016, August 26). What is the difference between Expo and React Native?. Stack-Overflow. Retrieved from <https://stackoverflow.com/questions/39170622/what-is-the-difference-between-expo-and-react-native>
30. Expo CLI. (2021). Expo. Retrieved from <https://docs.expo.io/workflow/expo-cli/>

31. Expo Workflows. (2021). Expo. Retrieved from <https://docs.expo.io/introduction/managed-vs-bare/>
32. AWS Amplify. (2021). Amazon. Retrieved from <https://aws.amazon.com/amplify/?nc=s&loc=1>
33. Clark J. (2021). What is AWS Amplify?. Back4app. Retrieved from <https://blog.back4app.com/what-is-aws-amplify/>
34. Introducing Amazon Cognito. (2014, July 10). Amazon. Retrieved from <https://aws.amazon.com/about-aws/whats-new/2014/07/10/introducing-amazon-cognito/>
35. AWS AppSync. (2021). Amazon. Retrieved from <https://aws.amazon.com/appsync/>
36. Dabit N. (2018, August 27). Installing & Configuring the AWS Amplify CLI. Amplify. Retrieved from <https://docs.amplify.aws/start/getting-started/installation/q/integration/react#option-2-follow-the-instructions>