



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

ΚΑΤΕΥΘΥΝΣΗ: ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**E-Governance Platform For Local Communities Management and
Citizens Engagement**

**Πλατφόρμα Ηλεκτρονικής Διακυβέρνησης για τη Διαχείριση
Τοπικών Κοινοτήτων και την Ενεργή Συμμετοχή Πολιτών**

ΓΕΩΡΓΙΟΣ ΧΑΤΖΑΚΗΣ

A.M.: E16160

Επιβλέπων Καθηγητής:

ΑΝΔΡΕΑΣ ΜΕΝΥΧΤΑΣ

ΠΕΙΡΑΙΑΣ, ΣΕΠΤΕΜΒΡΙΟΣ 2023

ΠΕΡΙΛΗΨΗ

Στα πλαίσια της πτυχιακής εργασίας δημιουργείται μία εφαρμογή κοινωνικής δικτύωσης για έξυπνα κινητά η οποία έχει στόχο την δημιουργία τοπικών κοινοτήτων και την προώθηση δράσεων μέσα από τη συμμετοχή του τοπικού πληθυσμού σε αυτές. Με τον προσδιορισμό των κοινοτήτων να πραγματοποιείται με γεωγραφικά κριτήρια (ταχυδρομικός κώδικας) μιας περιοχής, η πλατφόρμα δίνει τη δυνατότητα σε όποιους βρίσκονται εντός του γεωγραφικού χώρου να εντάσσονται στη συγκεκριμένη ομάδα και να έχουν μια σειρά από δυνατότητες. Οι δυνατότητες αυτές αφορούν την ανάδειξη εκπροσώπων της ομάδας μέσα από ψηφοφορίες, τη συμμετοχή σε κάθε λογής τοπικές ψηφοφορίες, την ανάδειξη προβλημάτων και δράσεων, την προβολή πολυμεσικού περιεχομένου και σημείων ενδιαφέροντος, κ.α. Η ψηφιακή πλατφόρμα περιλαμβάνει εφαρμογή για έξυπνα κινητά για την αλληλεπίδραση με τους χρήστες καθώς και τη σχετική υποδομή υπολογιστικού νέφους για την υποστήριξή της.

Για την ανάπτυξη της εφαρμογής, χρησιμοποιήθηκε η πλατφόρμα υπηρεσιών νέφους Google Firebase. Συνδέεται με την βασική γλώσσα και μοναδική που χρησιμοποιήθηκε για την ανάπτυξη των λειτουργιών της εφαρμογής React Native. Βασική λειτουργία αποτελεί η λειτουργία location η οποία τραβάει τα απαραίτητα γεωγραφικά δεδομένα, και τα χρησιμοποιεί για την είσοδο του χρήστη σε μία κοινότητα. Η εφαρμογή χρησιμοποιεί λειτουργίες εγγραφής (Sign up) και εισόδου (Login) για την είσοδο του χρήστη και την εγγραφή του στην κοινότητα. Ο χρήστης εισέρχεται στην αρχική σελίδα της κοινότητας από την οποία μπορεί να πλοηγηθεί στις υπόλοιπες οθόνες και λειτουργίες, και να δηλώσει υποψηφιότητα για εκπρόσωπος κοινότητας. Ως απλός χρήστης, μπορεί να ψηφίσει θέματα που δημιούργησε ο εκλεγμένος εκπρόσωπος, να δει τα προγραμματισμένα events καθώς και να ανακαλέσει τον εκλεγμένο αλλάζοντας την ψήφο του ανά πάσα ώρα και στιγμή. Επίσης έχει την δυνατότητα να συζητήσει με όλα τα ενεργά μέλη της κοινότητας μέσω της συζήτησης (Discussion). Ως εκπρόσωπος κοινότητας έχει την δυνατότητα να δημιουργήσει και να ακυρώσει events, καθώς και να δημιουργήσει ψηφοφορίες προς όλα τα μέλη της κοινότητας. Όλες οι ενέργειες αφορούν την κοινότητα που αντιστοιχεί στον ταχυδρομικό κώδικα στον οποίο βρίσκεται εκείνη την στιγμή ο κάθε χρήστης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ψηφιακές Κοινότητες

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ:

1. Δημιουργία τοπικών κοινοτήτων
2. Κοινωνική εφαρμογή βάση Γεωγραφικών δεδομένων
3. Συμμετοχή πολιτών
4. Λήψη αποφάσεων με τη συμμετοχή του κοινού
5. Τοπική κοινωνική οργάνωση

ABSTRACT

In the context of the bachelor's thesis, a social application is developed with the aim of creating local communities and promoting actions through the participation of the local population. The identification of communities is based on geographic criteria (postal code) of an area. The platform allows those within the geographic area to join the specific group and have a range of capabilities. These capabilities include the selection of community representatives through voting, participation in various local polls, highlighting issues and actions, showcasing multimedia content and points of interest, and more. The digital platform includes a mobile application for interaction with users, along with the necessary cloud computing infrastructure for support.

The application was developed using the Google Firebase cloud services platform and is connected to the core language and framework used for the development of its features, React Native. The main functionality of the application is the "location" function, which retrieves necessary geographic data and uses it for user entry into a community. The app includes "Sign up" and "Login" features for user registration and login to the community. Upon logging in, the user is directed to the community's main page, from which they can navigate to other screens and functionalities. Users can also declare their candidacy to become a community representative. As a regular user, they can vote on topics created by the elected representative, view scheduled events, and change their vote at any time. They also have the ability to engage in discussions with all active community members through the "Discussion" feature. As a community representative, they can create and cancel events, as well as initiate polls for all community members. All actions and features are related to the community corresponding to the postal code in which the user is currently located.

SUBJECT AREA: Digital Communities

KEYWORDS:

1. Local Community Building
2. Geographically-Based Social Application
3. Civil Engagement
4. Crowd-Sourced Decision Making
5. Local Social organization

*Την παρούσα πτυχιακή εργασία την αφιερώνω στις μελλοντικές γενιές των ανθρώπων
που θα ζουν αμεσοδημοκρατικά με τη βοήθεια της τεχνολογίας.*

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Πτυχιακής Εργασίας, θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον καθηγητή του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά, Ανδρέα Μενύχτα, για τη συνεργασία, την καθοδήγησή του και την πολύτιμη συμβολή του στην ολοκλήρωση της.

ΠΕΡΙΕΧΟΜΕΧΟΜΕΝΑ

| | |
|---|-----------|
| ΠΡΟΛΟΓΟΣ..... | 10 |
| 1. ΕΙΣΑΓΩΓΗ..... | 11 |
| 1.1 Σκοπός και Στόχοι της Διπλωματικής Εργασίας | 12 |
| 2. ΘΕΩΡΗΤΙΚΟ ΚΑΙ ΕΠΙΣΤΗΜΟΝΙΚΟ ΥΠΟΒΑΘΡΟ..... | 13 |
| 2.1. Μέσα Κοινωνικής Δικτύωσης..... | 13 |
| 2.2 Η Εξέλιξη των Μέσων Κοινωνικής Δικτύωσης | 13 |
| 2.3 Κοινωνικά Μέσα και η Δημοκρατία του Διαδικτύου..... | 16 |
| 2.3.1 Ο Ρόλος των Μέσων Κοινωνικής Δικτύωσης στη Δημοκρατία..... | 16 |
| 2.4 Δημοκρατία του διαδικτύου..... | 17 |
| 2.4.1 Πλεονεκτήματα..... | 17 |
| 2.4.2 Μειονεκτήματα..... | 18 |
| 3. ΨΗΦΙΑΚΕΣ ΠΛΑΤΦΟΡΜΕΣ ΚΟΙΝΩΝΙΚΗΣ ΑΥΤΟΟΡΓΑΝΩΣΗΣ | 20 |
| 3.1 Πώς οι Χρήστες των Κοινωνικών Μέσων Μπορούν να Αυτοοργανωθούν Γύρω από Κοινωνικά, Πολιτικά Ζητήματα;..... | 20 |
| 3.2 Κοινωνική Αυτοοργάνωση με Γεωγραφικά Κριτήρια..... | 22 |
| 3.2.1 Συμμετοχική Δημοκρατία | 23 |
| 3.2.2 Ηλεκτρονικές Ψηφοφορίες..... | 25 |
| 3.2.3 Ανάδειξη Εκπροσώπου Κοινότητας | 27 |
| 4. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΚΟΙΝΩΝΙΚΩΝ ΕΦΑΡΜΟΓΩΝ | 29 |
| 4.1.Περιβάλλον Ανάπτυξης εφαρμογών για κινητά (IDE) | 29 |
| 4.2 Framework/Γλώσσα | 31 |
| 4.3 Επιλογή Πλατφόρμας για τη Διπλωματική Εργασία | 39 |
| 4.4 Επιπρόσθετες Τεχνολογίες και Εργαλεία συστήματος | 41 |
| 5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ | 43 |
| 5.1 Σχεδιασμός Συστήματος Αυθεντικοποίησης | 43 |
| 5.2. Σχεδιασμός Βάσης Δεδομένων | 44 |
| 5.3 Σχεδιασμός Συστήματος Κοινότητας | 45 |
| 6. ΥΛΟΠΟΙΗΣΗ ΣΥΤΗΜΑΤΟΣ | 47 |
| 6.1 Δημιουργία Expo project/app | 47 |
| 6.2 Σύνδεση του Expo Project με την Firebase | 48 |
| 6.3 Δημιουργία των Απαραίτητων Components..... | 50 |
| 6.4 Υλοποίηση Βασικών Λειτουργιών της Εφαρμογής | 51 |
| 6.4.1 Η Αρχική σελίδα..... | 54 |
| 6.4.2 Ο Χάρτης | 55 |
| 6.4.3 Οι Ψηφοφορίες..... | 55 |
| 6.4.4 Η Συζήτηση | 58 |

| | |
|---|-----------|
| 6.4.5 Το Προφίλ του Χρήστη | 59 |
| 6.4.6 Δημιουργία Ψηφοφορίας ή Εκδήλωσης | 60 |
| 6.4.7 Αποφάσεις Κοινότητας | 62 |
| 6.4.8 Λειτουργίες Διαγραφής Περιττών Δεδομένων..... | 63 |
| 6.5 Τα Αποτελέσματα στη Βάση Δεδομένων..... | 67 |
| 6.6 Η Ασφάλεια Δεδομένων και Ρόλοι Χρηστών | 71 |
| 7. ΤΟ ΣΥΣΤΗΜΑ ΣΕ ΛΕΙΤΟΥΡΓΙΑ | 72 |
| 8. ΣΥΜΠΕΡΑΣΜΑΤΑ..... | 83 |
| ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ | 84 |
| ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ | 85 |

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

| | |
|---|----|
| Εικόνα 1 - Δημοφιλή Μέσα Κοινωνικής Δικτύωσης | 15 |
| Εικόνα 2- Επανάσταση των μέσων κοινωνικής δικτύωσης | 21 |
| Εικόνα 3 - JAVA | 33 |
| Εικόνα 4 - kotlin | 34 |
| Εικόνα 5 - Dart | 35 |
| Εικόνα 6 - Swift | 37 |
| Εικόνα 7 - React Native Logo | 39 |
| Εικόνα 8- React Native | 40 |
| Εικόνα 9 - Visual Studio Code | 41 |
| Εικόνα 10- UML διάγραμμα ελέγχου ταυτότητας | 43 |
| Εικόνα 11- UML διάγραμμα δραστηριότητας | 44 |
| Εικόνα 12- UML διάγραμμα βάσης δεδομένων | 45 |
| Εικόνα 13- UML συστήματος κοινότητας | 46 |
| Εικόνα 14- Εντολή για την δημιουργία expo project | 47 |
| Εικόνα 15- Εγκατάσταση των απαραίτητων αρχείων | 47 |
| Εικόνα 16 - firebase-ath overview | 48 |
| Εικόνα 17- Δημιουργία προηποθέσεων σύνδεσης για διαδικτυακές εφαρμογές | 49 |
| Εικόνα 18- Το αρχείο firebase.js | 49 |
| Εικόνα 19- Imports και μεταβλητές | 50 |
| Εικόνα 20- Καλεί το API και λαμβάνει την τοποθεσία | 50 |
| Εικόνα 21- Το location.js αρχείο επιστρέφει τις ανάλογες μεταβλητές | 51 |
| Εικόνα 22- Screen components | 51 |
| Εικόνα 23- Όλα τα imports του App.js αρχείου | 52 |
| Εικόνα 24- Το component App() επιστρέφει το component «RootNavigator» | 52 |
| Εικόνα 25- Το «App» component ελέγχει την μεταβλητή location | 52 |
| Εικόνα 26- Το «RootNavigator» component | 53 |
| Εικόνα 27- Το AuthStack() component | 54 |
| Εικόνα 28- Home1 component | 54 |
| Εικόνα 29- Η Back() επιστρέφει τις οθόνες | 54 |
| Εικόνα 30- Το «Map» component | 55 |
| Εικόνα 31- Η συνάρτηση «getPartSis» | 56 |
| Εικόνα 32- Η συνάρτηση «handleParticipationToggle» | 57 |
| Εικόνα 33- Η «handleRemoveToggle» συνάρτηση | 57 |
| Εικόνα 34- Η «votes» συνάρτηση | 58 |
| Εικόνα 35- Η «onSend» συνάρτηση | 59 |
| Εικόνα 36- Η «handleSave» συνάρτηση | 60 |
| Εικόνα 37- Η «handleDeleteAccount» συνάρτηση | 60 |
| Εικόνα 38 - Το «createVoting» component μέρος 1 | 61 |
| Εικόνα 39– Το «createVoting» component μέρος 2 | 61 |
| Εικόνα 40– Το «createVoting» component μέρος 3 | 62 |
| Εικόνα 41– Το «createVoting» component μέρος 4 | 62 |
| Εικόνα 42- Real time vote μέρος 1 | 63 |
| Εικόνα 43- Real time vote μέρος 2 | 63 |
| Εικόνα 44- Υποβολή ψήφου μέρος 1 | 64 |
| Εικόνα 45- Υποβολή ψήφου μέρος 2 | 64 |
| Εικόνα 46- Διαγραφή των ανενεργών χρηστών | 65 |
| Εικόνα 47- Διαγραφή των περιττών εκδηλώσεων | 66 |
| Εικόνα 48- Το ανέβασμα των functions στο cloud firestore | 67 |
| Εικόνα 49- To collection «Community» | 68 |
| Εικόνα 50- To subcollection «users» με τα πεδία για κάθε χρήστη | 68 |
| Εικόνα 51- Το «event» subcollection με τα documents και τα fields των εγγραφών των events | 69 |
| Εικόνα 52- Το «candidates» subcollection. τα documents και τα fields των εγγραφών των υποψηφίων | 69 |
| Εικόνα 53- Το «voting» subcollection, τα documents και τα fields των εγγραφών των ψηφοφοριών | 70 |

| | |
|--|----|
| <u>Εικόνα 54- Το «results» subcollection με τα documents να παίρνουν τιμές τις υποψήφιες απαντήσεις</u> | 70 |
| <u>Εικόνα 55- Δικαιοδοσίες ανάγνωσης και εγγραφής στους χρήστες</u> | 71 |
| <u>Εικόνα 56- Αναμονή της οθόνης μέχρι το location να πάρει την σωστή τιμή</u> | 72 |
| <u>Εικόνα 57- Η οθόνη εγγραφής χρήστη στο σύστημα</u> | 72 |
| <u>Εικόνα 58- Η οθόνη εισόδου χρηστών με το ηλεκτρονικό ταχυδρομείο και τον κωδικό ως πεδία</u> | 73 |
| <u>Εικόνα 59- Η οθόνη της αρχικής σελίδας της κοινότητας</u> | 73 |
| <u>Εικόνα 60- Ο χάρτης τοποθεσίας του χρήστη και των εκδηλώσεων</u> | 73 |
| <u>Εικόνα 61- Η οθόνη του «vote» component</u> | 73 |
| <u>Εικόνα 62- Επιλογή του χρήστη να βάλει υποψηφίότητα για admin και το αντίστοιχο Alert</u> | 74 |
| <u>Εικόνα 63- Ο χρήστης επέλεξε Apply και πλέον είναι στους υποψήφιους</u> | 74 |
| <u>Εικόνα 64- Το «Chat» component</u> | 74 |
| <u>Εικόνα 65- Ο χρήστης στέλνει ένα μήνυμα στην ομαδική συζήτηση</u> | 74 |
| <u>Εικόνα 66- Ο χρήστης βρίσκεται στο component «Profile»</u> | 75 |
| <u>Εικόνα 67- Ο χρήστης αλλάζει την περιγραφή για τον εαυτό του που προβάλει στους υπόλοιπους χρήστες μέσω της υποψηφίότητας</u> | 75 |
| <u>Εικόνα 68- Ο χρήστης βλέπει την περιγραφή ενός υποψηφίου</u> | 75 |
| <u>Εικόνα 69- Ο χρήστης επέλεξε να ακυρώσει την υποψηφίότητά του</u> | 75 |
| <u>Εικόνα 70- Ο χρήστης ψήφισε έναν υποψήφιο πατώντας το εικονίδιο δεξιά του υποψηφίου</u> | 76 |
| <u>Εικόνα 71- Ο χρήστης βλέπει στην οθόνη «Home» ότι είναι ο admin της κοινότητας</u> | 76 |
| <u>Εικόνα 72- Οι ειδοποίησης όταν ο αλλάζει ο διαχειριστής της κοινότητας</u> | 76 |
| <u>Εικόνα 73- Πατώντας το «clear notifications» το εικονίδιο καθάρισε</u> | 76 |
| <u>Εικόνα 74- Ο διαχειριστής στη συζήτηση της κοινότητας είναι διακριτός</u> | 77 |
| <u>Εικόνα 75- Οθόνη δημιουργίας «event»</u> | 77 |
| <u>Εικόνα 76- Επεξεργασία ώρας και μέρας απ' την by default</u> | 77 |
| <u>Εικόνα 77- Δημιουργία event και το αντίστοιχο alert</u> | 77 |
| <u>Εικόνα 78- Εμφάνιση του event στην αρχική σελίδα κατόπιν reload</u> | 78 |
| <u>Εικόνα 79- Η αρχική σελίδα με την επιλογή για διαγραφή του event από τον διαχειριστή</u> | 78 |
| <u>Εικόνα 80- Η ειδοποίηση από την δημιουργία event</u> | 78 |
| <u>Εικόνα 81- Επιλογή και ειδοποίηση διαγραφής event</u> | 78 |
| <u>Εικόνα 82- Το στύγμα της τοποθεσίας του event στον χάρτη</u> | 79 |
| <u>Εικόνα 83- Η επιλογή δημιουργίας voting</u> | 79 |
| <u>Εικόνα 84- Η κάρτα του voting1 εμφανίζεται στην οθόνη «vote»</u> | 79 |
| <u>Εικόνα 85- Άνοιγμα της κάρτας του voting1 με τις επιλογές ως προς ψήφιση</u> | 79 |
| <u>Εικόνα 86- Επιλογή μιας συγκεκριμένης απάντησης</u> | 80 |
| <u>Εικόνα 87- Υποβολή της απάντησης και μεταφορά στο «vote» component</u> | 80 |
| <u>Εικόνα 88- Ο χρήστης έχει ανοίξει μια κάρτα στην οποία έχει ήδη ψηφίσει</u> | 80 |
| <u>Εικόνα 89- Αφού λήξει η προθεσμία υποβολής, η κάρτα μεταφέρεται στην οθόνη History</u> | 80 |
| <u>Εικόνα 90- Ο χρήστης έχει ανοίξει την κάρτα voting1 που υπάρχει πλέον στο ιστορικό</u> | 81 |
| <u>Εικόνα 91- Στην αρχική σελίδα φαίνεται η απόφαση που πήρε η κοινότητα ως προς το voting1</u> | 81 |

ΠΡΟΛΟΓΟΣ

Η παρούσα εργασία είναι εμπνευσμένη από την κοινωνική οργάνωση των μαζών και των λαών γενικότερα, απόν την προϊστορία μέχρι και τις μέρες μας. Από την αρχαιότητα, οι κοινότητες των ανθρώπων προσπαθούσαν να λειτουργήσουν με κάποιον συλλογικό και οργανωμένο τρόπο για την επιβίωση και την εξέλιξη της κοινωνίας. Η ανθρωπότητα μπόρεσε να εξελιχθεί και να περάσει από διάφορες μορφές δημοκρατίας, οι οποίες κατά κύριο λόγο αφορούσαν συγκεκριμένες κοινωνικές και ταξικές ομάδες. Συνεπώς οι ανθρώπινες κοινωνίες περάσανε από τη πιο απλή μορφή δημοκρατίας των κοινοτήτων, στην δουλοκτητική, στην φεουδαρχική και μετέπειτα στην σημερινή σύγχρονη αστική δημοκρατία[16].

Απομονώνοντας λοιπόν τη λέξη “δημοκρατία” από τις κοινωνικές και ταξικές ομάδες που αφορούσε την εκάστοτε περίοδο, τα χαρακτηριστικά τα οποία την αποτελεί, όπως για παράδειγμα η αρχή της πλειοψηφίας ή η ελευθερία της έκφρασης διαφορετικών απόψεων, υπήρξε ως ένας σημαντικός και φιλολαϊκός παράγοντας για την εξέλιξη αυτή. Παρ ‘όλα αυτά, ο σύγχρονος άνθρωπος δεν έχει γνωρίσει την άμεση δημοκρατία. Φτάνοντας στη σημερινή εποχή, με την εξέλιξη της τεχνολογίας και την ενσωμάτωση του διαδικτύου ως κοινωνικό και ενημερωτικό μέσο, η προσαρμογή των δημοκρατικών διαδικασιών και μορφών οργάνωσης για τους πολίτες είναι κάτι που χρήζει αναβάθμισης και εμπλουτισμού.

Στον δυτικό κόσμο του 21ου αιώνα ,ο άνθρωπος χρησιμοποιεί καθημερινά υπολογιστές ενσύρματους και ασύρματους, καθώς και έξυπνα κινητά (smartphones) για να εργαστεί, να κοινωνικοποιηθεί, να ενημερωθεί και να ψυχαγωγηθεί. Η ανάπτυξη των δημοκρατικών διαδικασιών μέσα στο διαδίκτυο μπορούν να βοηθήσουν τους πολίτες μιας γεωγραφικής περιοχής να αυτοοργανώνονται σε κοινωνικό αλλά και σε πολιτικό επίπεδο. Η δημοκρατία συνεπώς μπορεί πλέον να συνδεθεί και να γίνει απλούστερη συνδυάζοντας την με την τεχνολογία. Παρατηρούμε πως σε πολλά μέσα κοινωνικής δικτύωσης κάποια χαρακτηριστικά της δημοκρατίας είναι κάπως κεκτημένα. Η αναβάθμιση όμως του φαινομένου αυτού δε μπορεί να είναι άλλη από το να μπορέσει να μπει στη ζωή μας και με κοινωνικοπολιτικά χαρακτηριστικά. Εξάλλου το διαδίκτυο φέρνει όλο τον κόσμο πιο κοντά.

1. ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια, παρατηρείται μια ραγδαία αύξηση της χρήσης των μέσων κοινωνικής δικτύωσης (social media). Αυτή η αύξηση είναι εντυπωσιακή, ιδίως όταν συγκρίνουμε την τρέχουσα κατάσταση με την προηγούμενη δεκαετία. Πολλοί παράγοντες συνέβαλαν σε αυτήν την εξέλιξη, και η σημασία των κοινωνικών δικτύων στην καθημερινή μας ζωή μας δεν πρέπει να υποτιμάται.

Ένας από τους κύριους λόγους για την αυξανόμενη χρήση των μέσων κοινωνικής δικτύωσης είναι η ευκολία πρόσβασης σε αυτά. Οι άνθρωποι μπορούν να συνδεθούν σε πλατφόρμες κοινωνικών μέσων από οπουδήποτε με πρόσβαση στο διαδίκτυο, είτε μέσω κινητών συσκευών είτε μέσω υπολογιστών. Αυτή η ευκολία προσβασιμότητας έχει δημιουργήσει μια σταθερή σύνδεση με τα κοινωνικά μέσα, καθιστώντας τα έναν αναπόσπαστο μέρος της καθημερινής μας ρουτίνας.

Η συστηματική της χρήσης των κοινωνικών μέσων μπορεί να αποδοθεί στην αυξανόμενη ανάγκη για κοινωνική σύνδεση και επικοινωνία. Οι άνθρωποι χρησιμοποιούν τα κοινωνικά μέσα για να διατηρούν επαφές με φίλους και οικογένεια, να ενημερώνονται για τα τελευταία γεγονότα και να μοιράζονται τις σκέψεις και τις εμπειρίες τους. Τα κοινωνικά μέσα δίνουν τη δυνατότητα να διατηρούμε σχέσεις ακόμα και όταν βρισκόμαστε μακριά από τους αγαπημένους μας.

Επιπλέον, η όξυνση της χρήσης των μέσων κοινωνικής δικτύωσης έχει επηρεάσει πολλούς τομείς, όπως την πολιτική, την επιχειρηματικότητα και την ενημέρωση. Οι πολιτικοί χρησιμοποιούν τα κοινωνικά μέσα για να επικοινωνούν με τους ψηφοφόρους τους, ενώ οι επιχειρηματίες τα χρησιμοποιούν για την προώθηση των προϊόντων και των υπηρεσιών τους. Τα μέσα κοινωνικής δικτύωσης έχουν επίσης αλλάξει τον τρόπο που λαμβάνουμε και αντιλαμβανόμαστε ειδήσεις, καθώς πολλά μέσα ενημέρωσης χρησιμοποιούν αυτές τις πλατφόρμες για να διαδώσουν πληροφορίες. Συνολικά, η αύξηση της χρήσης των κοινωνικών δικτύων αποτελεί σημαντική εξέλιξη στην τρόπο που επικοινωνούμε, ενημερωνόμαστε και συνδεόμαστε με τον υπόλοιπο κόσμο, επηρεάζοντας πολλούς τομείς της ζωής μας.

1.1 Σκοπός και Στόχοι της Διπλωματικής Εργασίας

Βασικός σκοπός της παρούσας εργασίας είναι η επιλογή και η μελέτη μιας κοινωνικής εφαρμογής για έξυπνα κινητά η οποία αφορά την κοινωνική και κοινωνικοπολιτική αυτοοργάνωση των πολιτών μιας γεωγραφικής περιοχής. Η πλατφόρμα που θα δημιουργηθεί θα δίνει την δυνατότητα στον χρήστη να μπαίνει στην ηλεκτρονική/διαδικτυακή κοινότητα βάση της τοποθεσίας που βρίσκεται την τρέχουσα στιγμή, την συμμετοχή του στα κοινά της περιοχής την επικοινωνία με τα υπόλοιπα μέλη της κοινότητας μέσω συνελεύσεων και ομαδικής συζήτησης, αλλά και την δυνατότητα ανάδειξης του ως διαχειριστή της κοινότητας. Αποτελεί ένα σύστημα που προσπαθεί να φέρει τον κόσμο πιο κοντά με τις δημοκρατικές διαδικασίες και τους θεσμούς της άμεσης δημοκρατίας.

Στα επόμενα κεφάλαια θα αναλυθεί η επιλογή της πλατφόρμας που θα χρησιμοποιηθεί για το σκοπό αυτό, όπως επίσης και θα αιτιολογηθούν και οι λόγοι για τους οποίους επιλέχθηκαν οι συγκεκριμένες λειτουργικές μέθοδοι που αντανακλούν την συγκεκριμένη κοινωνική αλληλεπίδραση μεταξύ των πολιτών.

Οι γενικότεροι στόχοι της εργασίας είναι οι εξής:

- Δημιουργία τοπικής αυτοοργάνωσης πολιτών μιας γεωγραφικής περιοχής
- Μελέτη της αλληλεπίδρασης των πολιτών μέσω δημοκρατικών διαδικασιών
- χρήση μιας εφαρμογής που θα ενημερώνει τον χρήστη για την περιοχή στην οποία βρίσκεται αλλά και τη δυνατότητα να γίνει κομμάτι αυτής.

2. ΘΕΩΡΗΤΙΚΟ ΚΑΙ ΕΠΙΣΤΗΜΟΝΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτή την ενότητα θα παρουσιαστεί όλο το βασικό θεωρητικό και επιστημονικό υπόβαθρο για την κατανόηση των μέσω κοινωνικής δικτύωσης και επικοινωνίας καθώς και την κοινωνική αυτοοργάνωση μέσω της δημοκρατίας.

2.1. Μέσα Κοινωνικής Δικτύωσης

Τα μέσα κοινωνικής δικτύωσης (social media) αναφέρονται σε διαδικτυακές πλατφόρμες και εφαρμογές που επιτρέπουν στους χρήστες να δημιουργούν προσωπικά προφίλ, να επικοινωνούν, να συνδέονται και να αλληλεπιδρούν με άλλους χρήστες μέσω του Διαδικτύου. Παρέχουν έναν διαδικτυακό χώρο όπου οι άνθρωποι μπορούν να μοιράζονται πληροφορίες, φωτογραφίες, βίντεο, σκέψεις, και να συμμετέχουν σε διάφορες δραστηριότητες.^[12]

Έχουν επηρεάσει σημαντικά τον τρόπο που οι άνθρωποι επικοινωνούν, ενημερώνονται, διασκεδάζουν και αλληλεπιδρούν με τον κόσμο γύρω τους, και έχουν εξελιχθεί σε ένα σημαντικό κομμάτι της καθημερινής ζωής πολλών ανθρώπων.

2.2 Η Εξέλιξη των Μέσων Κοινωνικής Δικτύωσης

Η ιστορική εξέλιξη των μέσω κοινωνικής δικτύωσης ξεκινά από τις πρώτες προσπάθειες διαδικτυακής κοινωνικής δικτύωσης στις αρχές της δεκαετίας του '90 και φτάνει στο σημερινό δυναμικό τοπίο των κοινωνικών δικτύων. Ας ρίξουμε μια ματιά στα σημαντικά βήματα στην εξέλιξή τους^[14]:

Τέλη της Δεκαετίας του '90 - Πρώτες Προσπάθειες:

Το 1997, δημιουργήθηκε το "Six Degrees," ένα από τα πρώτα κοινωνικά δίκτυα. Παρόλο που δεν πέτυχε, υπήρξε η αρχή της ιδέας της επικοινωνίας και της σύνδεσης μέσω του διαδικτύου.

Αρχές του 21ου Αιώνα - Η Εμφάνιση του Friendster και του MySpace:

Το 2002, δημιουργήθηκε το Friendster, ένα κοινωνικό δίκτυο που ήταν δημοφιλές στην Ασία. Το 2003, ξεκίνησε το MySpace, που ήταν εξαιρετικά δημοφιλές στις Ηνωμένες Πολιτείες. Και τα δύο δίκτυα επέτρεπαν στους χρήστες να δημιουργούν προφίλ και να συνδέονται με φίλους.

To 2004 - Γέννηση του Facebook:

Το Facebook ιδρύθηκε από τον Mark Zuckerberg και τους συνιδρυτές του το Φεβρουάριο του 2004. Αρχικά, ήταν διαθέσιμο μόνο για φοιτητές του Harvard, αλλά στη συνέχεια διευρύνθηκε σε άλλα πανεπιστήμια και τελικά για τον κόσμο γενικότερα. Το Facebook έπαιξε καθοριστικό ρόλο στο να καθιερώσει την ιδέα της κοινωνικής δικτύωσης και ηλεκτρονικής αλληλεπίδρασης ως κύριας διαδικτυακής δραστηριότητας. Σήμερα, To Facebook είναι ίσως το πιο δημοφιλές κοινωνικό μέσο με δισεκατομμύρια χρήστες παγκοσμίως. Επιτρέπει στους χρήστες να δημιουργούν προσωπικά προφίλ, να μοιράζονται φωτογραφίες και βίντεο, να συνδέονται με φίλους και να επικοινωνούν μέσω μηνυμάτων, να δημιουργούν groups και ομάδες συγκεκριμένου θεματικού περιεχομένου.

Τα Περαιτέρω Βήματα - Twitter, YouTube, LinkedIn κ.α.:

Μετά την επιτυχία του Facebook, πολλές άλλες πλατφόρμες κοινωνικής δικτύωσης εμφανίστηκαν. Το Twitter (2006), το YouTube (2005), το LinkedIn (2002) και άλλα κοινωνικά δίκτυα έγιναν αναγνωρίσιμα ονόματα.

Το Twitter είναι ένα μέσο κοινωνικής δικτύωσης που επικεντρώνεται στη σύντομη και γρήγορη επικοινωνία μέσω "tweets," που είναι μηνύματα με μέγιστο τους 280 χαρακτήρες. Είναι αρκετά δημοφιλές αφού δίνει την δυνατότητα οι χρήστες να ανταλλάζουν σκέψεις και πληροφορίες, και ενημερώνονται αρκετά γρήγορα.

Το LinkedIn είναι μία διαδικτυακή κοινωνική εφαρμογή που αφορά κατά κύριο λόγο επαγγελματίες. Επιτρέπει στους χρήστες να δημιουργήσουν επαγγελματικά προφίλ, να συνδέονται με συναδέλφους και να αναζητούν εργασία και απασχόληση.



Εικόνα 1 - Δημοφιλή Μέσα Κοινωνικής Δικτύωσης

Κοινωνικά Μέσα και Κοινωνική Επανάσταση:

Κατά τη διάρκεια της αραβικής άνοιξης το 2010, τα κοινωνικά μέσα έπαιξαν καθοριστικό ρόλο στην οργάνωση διαμαρτυριών και ανταλλαγή πληροφοριών. Αποτελεί ένα παράδειγμα του πώς τα μέσα κοινωνικής δικτύωσης επηρεάζουν την πολιτική και την κοινωνία.

Συνολικά, η ιστορία των κοινωνικών μέσων είναι μια ιστορία της επικοινωνίας και της σύνδεσης, που άλλαξε τον τρόπο που ζούμε, εργαζόμαστε, ενημερωνόμαστε και επικοινωνούμε με τον κόσμο γύρω μας.

2.3 Κοινωνικά Δίκτυα και η Δημοκρατία του Διαδικτύου

Πολλά από τα κοινωνικά δίκτυα έχουν αναπτύξει την δυνατότητα οι χρήστες να μπορούν να δημιουργούν κοινωνικά, πολιτικά, επαγγελματικά και επιστημονικά events και δρώμενα. Με την πάροδο του χρόνου, η συγκεκριμένη λειτουργία έχει εδραιωθεί στην κοινωνική και ενημερωτική συνείδηση του κόσμου.

Τα μέσα κοινωνικής δικτύωσης δίνουν στους πολίτες τη δυνατότητα να έχουν πρόσβαση σε πληροφορίες και νέα από πολλές πηγές. Αυτό βοηθάει στην ενημέρωση των πολιτών και στη διαμόρφωση της δημόσιας γνώμης. Δημιουργούν πλατφόρμες για τη διεξαγωγή δημόσιων συζητήσεων και διαλόγων. Οι χρήστες μπορούν να σχολιάζουν ζητήματα που τους ενδιαφέρουν, να ανταλλάσσουν απόψεις και να προωθούν τον δημόσιο διάλογο. Η αλληλεπίδραση περιορίζεται περισσότερο σε ομάδες που αφορούν κάποιο κοινό χαρακτηριστικό, όπως η μουσική, κάποιο άθλημα ή χόμπι, οι πολιτικές ή επιστημονικές θέσεις κτλ.[18]

2.3.1 Ο Ρόλος των Μέσων Κοινωνικής Δικτύωσης στη Δημοκρατία

Τα μέσα κοινωνικής δικτύωσης έχουν μεταμορφώσει τη δημόσια συζήτηση και τον πολιτικό διάλογο, προσφέροντας πολλαπλά οφέλη. Καταρχάς, διασφαλίζουν διαφάνεια στη δημοκρατία, επιτρέποντας στους πολίτες να παρακολουθούν τις δραστηριότητες των πολιτικών και των κυβερνήσεων. Επιπλέον, προσφέρουν γρήγορη και εκτενή διακίνηση των νέων, ενημερώνοντας το κοινό για σημαντικά γεγονότα. Επιτρέπουν τη συζήτηση, την ανταλλαγή απόψεων και την έκφραση ανησυχιών για πολιτικά ζητήματα. Οι δημόσιες συζητήσεις μπορούν να διαμορφώσουν τη δημόσια γνώμη και να ενθαρρύνουν την ενεργή πολιτική συμμετοχή. Επίσης, οι πολιτικοί χρησιμοποιούν τα μέσα κοινωνικής δικτύωσης για την προώθηση των καμπανιών τους, απευθυνόμενοι σε ευρύτερα κοινά και επικοινωνώντας απευθείας με τους ψηφοφόρους. Συνολικά, τα κοινωνικά μέσα αποτελούν ισχυρό εργαλείο για την ενίσχυση της δημοκρατίας και τη διαμόρφωση του πολιτικού διαλόγου.[19]

2.4 Δημοκρατία του Διαδικτύου

2.4.1 Πλεονεκτήματα

Τα κοινωνικά δίκτυα έχουν φέρει πολλά πλεονεκτήματα στο πλαίσιο της "δημοκρατίας του διαδικτύου," η οποία αναφέρεται στη δημιουργία και εξάπλωση δημοκρατικών αξιών και διαδικασιών μέσω του διαδικτύου.

Η **ελευθερία έκφρασης** στα κοινωνικά δίκτυα ανοίγει την πόρτα για πολυποίκιλες απόψεις και προσεγγίσεις σε δημόσια θέματα. Ανεξάρτητα από το πολιτικό ή κοινωνικό υπόβαθρο των ατόμων, μπορούν να συμμετέχουν σε συζητήσεις και να προβάλλουν τις απόψεις τους. Αυτό δεν είναι μόνο πλεονέκτημα για τους πολίτες αλλά και για τη δημοκρατία ως σύστημα, καθώς διευρύνει τη συμμετοχή των ανθρώπων στη διαμόρφωση των αποφάσεων. Επίσης, τα μέσα κοινωνικής δικτύωσης ενθαρρύνουν τη **συνεργασία και τη διασύνδεση ανθρώπων με παρόμοια ενδιαφέροντα, ανεξαρτήτως απόστασης**. Αυτό επιπρέπει τη δημιουργία ευρύτερων και πιο ποικίλων δικτύων πολιτών που μπορούν να συνεργάζονται για το κοινό καλό και να πιέζουν για την εφαρμογή πολιτικών αλλαγών. Συνολικά, η ελευθερία έκφρασης στα μέσα κοινωνικής δικτύωσης δημιουργεί ένα περιβάλλον που ενθαρρύνει τον δημόσιο διάλογο, τη συμμετοχή των πολιτών και τη διαφάνεια στην πολιτική διαδικασία, ενισχύοντας έτσι την ιδέα της δημοκρατίας του διαδικτύου[13].

Επίσης, προσφέρουν **άμεση πρόσβαση σε πληροφορίες και νέα**. Οι πολίτες μπορούν να ενημερώνονται γρήγορα για τις τρέχουσες εξελίξεις και να προσεγγίζουν διάφορες πηγές πληροφοριών, βελτιώνοντας την κατανόησή τους για πολλά ζητήματα.

Ένα ακόμα σημαντικό πλεονέκτημα της χρήσης τους στο πλαίσιο της δημοκρατίας του διαδικτύου είναι η **ενίσχυση της δημοκρατικής συμμετοχής**. Οι πλατφόρμες μέσων κοινωνικής δικτύωσης διευκολύνουν τους πολίτες να αναλαμβάνουν ενεργό ρόλο στη δημόσια σφαίρα και στη διαμόρφωση της πολιτικής πραγματικότητας.[18]

Συγκεκριμένα, μέσω των μέσων κοινωνικής δικτύωσης, οι πολίτες έχουν τη **δυνατότητα να συμμετέχουν σε πολιτικές συζητήσεις και να εκφράζουν τις απόψεις τους σχετικά με σημαντικά κοινά θέματα**. Μπορούν να ανταλλάσσουν

απόψεις, να συνεργάζονται με άλλους πολίτες και να διαμορφώνουν απόψεις και πρωτοβουλίες. Αυτή η αυξημένη συμμετοχή μετατρέπει τους πολίτες από απλούς θεατές σε ενεργούς συμμετέχοντες στην πολιτική διαδικασία.

Επιπλέον, επιτρέπουν στους πολίτες να οργανώνονται γύρω από πολιτικές απόψεις και αξίες και να δρουν συλλογικά. Μπορούν να δημιουργούν κινήματα, να στηρίζουν υποψηφίους που συμμερίζονται τις απόψεις τους και να κινητοποιούνται για την υπεράσπιση δικαιωμάτων και αξιών που θεωρούν σημαντικά. Η δημοκρατική συμμετοχή μέσω των κοινωνικών δικτύων ενισχύει τη φωνή των πολιτών και την ικανότητά τους να επηρεάζουν τις αποφάσεις και τις πολιτικές πρακτικές.

Συνολικά, τα κοινωνικά δίκτυα ανοίγουν νέους δρόμους για τη συμμετοχή και τη συνεισφορά των πολιτών στη δημοκρατική διαδικασία, ενισχύοντας έτσι τη δημοκρατία του διαδικτύου.

2.4.2 Μειονεκτήματα

Τα μέσα κοινωνικής δικτύωσης, παρά τα πλεονεκτήματά τους, έχουν και ορισμένα μειονεκτήματα που πρέπει να ληφθούν υπόψη.

Ένα από τα σημαντικότερα είναι η διάδοση ψευδών πληροφοριών και ψευδών ειδήσεων, που μπορεί να έχει επικίνδυνες επιπτώσεις στη δημόσια συνείδηση και τον πολιτικό διάλογο. Επιπλέον, στα κοινωνικά δίκτυα, εκδηλώνεται αντικοινωνική συμπεριφορά, συμπεριλαμβανομένου του cyber bullying, της προπαγάνδας, του μίσους και του εκφοβισμού, που μπορεί να έχει αρνητικές επιπτώσεις στη ψυχολογία και την ευημερία των ατόμων.^[14]

Επίσης, η χρήση μέσων κοινωνικής δικτύωσης συχνά οδηγεί σε προβλήματα ιδιωτικότητας και ασφάλειας των δεδομένων, καθώς οι προσωπικές πληροφορίες γίνονται πιο προσβάσιμες. Οι υποκλοπές δεδομένων είναι συχνό έως και καθημερινό φαινόμενο

Είναι σημαντικό να συνειδητοποιήσουμε ότι τα μέσα κοινωνικής δικτύωσης δεν αποτυπώνουν στο 100% την πραγματικότητα. Το περιεχόμενο που βλέπουμε σε αυτά είναι συχνά επιλεκτικό και μπορεί να παρουσιάζει ένα στιγμιότυπο, μια αποκομμένη από τον ευρύτερο πλαίσιο εικόνα της πραγματικότητας. Οι χρήστες επιλέγουν να μοιράζονται στιγμιότυπα της ζωής τους, ενδεχομένως να επιλέγουν

να παρουσιάσουν μια εικόνα που ανταποκρίνεται στις προσδοκίες του κοινού, και να παραλείπουν τα πιο προσωπικά ή δύσκολα στιγμιότυπα. Επιπλέον, οι αλγόριθμοι που διαχειρίζονται το ροή των πληροφοριών μπορεί να δημιουργούν φίλτρα και "φυλλοκροτήματα" (echo chambers), περιορίζοντας τον πλουραλισμό των απόψεων που αντικρίζουμε. Αυτό σημαίνει ότι οι πραγματικές απόψεις και αντιλήψεις των ανθρώπων είναι πολύ πιο πολύπλοκες από ό,τι μπορεί να φαίνεται στον κόσμο των κοινωνικών δικτύων. Επομένως, πρέπει να είμαστε προσεκτικοί και ενημερωμένοι, αξιολογώντας τις πληροφορίες και τις απόψεις που συναντάμε με κριτικό πνεύμα και ανοικτό μυαλό.

Τέλος, η εθιστική χρήση των μέσων κοινωνικής δικτύωσης μπορεί να οδηγήσει σε προβλήματα ψυχικής υγείας και κοινωνικής απομόνωσης, καθώς οι άνθρωποι είναι περισσότερο αποσυνδεδεμένοι από την πραγματική κοινωνική επαφή. Παρόλα αυτά, είναι σημαντικό να αναγνωρίσουμε ότι η χρήση τους μπορεί να είναι επωφελής όταν χρησιμοποιείται με σύνεση και ευαισθησία, ενώ οι χρήστες πρέπει να αναπτύξουν κριτική σκέψη και ευαισθησία στους κινδύνους που ενδέχεται να εμφανιστούν.^[14]

3. ΨΗΦΙΑΚΕΣ ΠΛΑΤΦΟΡΜΕΣ ΚΟΙΝΩΝΙΚΗΣ ΑΥΤΟΟΡΓΑΝΩΣΗΣ

Η κοινωνική αυτοοργάνωση μέσω των κοινωνικών δικτύων αποτελεί μια σημαντική εξέλιξη στη σύγχρονη κοινωνία και έχει επηρεάσει δραστικά τον τρόπο με τον οποίο οι άνθρωποι ενώνονται, επικοινωνούν, και αναπτύσσουν πρωτοβουλίες. Η αυτοοργάνωση μέσω του διαδικτύου αναφέρεται στην ικανότητα των ανθρώπων να οργανώνονται και να δρουν ανεξάρτητα, χρησιμοποιώντας διαδικτυακές πλατφόρμες για να επιτύχουν κοινωνικούς, πολιτικούς, και πολιτιστικούς στόχους.

Ένα από τα πρωταρχικά χαρακτηριστικά της διαδικτυακής κοινωνικής αυτοοργάνωσης είναι η δημιουργία ψηφιακών κοινοτήτων και δικτύων που μοιράζονται κοινά ενδιαφέροντα και αξίες. Οι πλατφόρμες όπως το Facebook, το Twitter και το Reddit επιτρέπουν στους ανθρώπους να συγκεντρώνονται γύρω από θέματα όπως ο περιβαλλοντικός ακτιβισμός, η ανθρώπινη δικαιοσύνη, η πολιτική συμμετοχή, η τέχνη, η εκπαίδευση, το ποδόσφαιρο και πολλά άλλα. Οι χρήστες μπορούν να συνδεθούν με άλλους που έχουν παρόμοια ενδιαφέροντα και να συνεργαστούν για την προώθηση των στόχων τους.

Ένα παράδειγμα πρακτικής αυτοοργάνωσης μέσω των κοινωνικών δικτύων είναι ο ακτιβισμός πάνω στο ζήτημα της κοινωνικής δικαιοσύνης και των δικαιωμάτων. Πολλές οργανώσεις και άτομα χρησιμοποιούν τα κοινωνικά δίκτυα για να προωθήσουν τα δικαιώματα του ανθρώπου και να διαδώσουν ενημέρωση σχετικά με περιστατικά παραβιάσεων ανθρωπίνων δικαιωμάτων.

3.1 Πώς οι Χρήστες των Μέσων Κοινωνικής Δικτύωσης Μπορούν να Αυτοοργανωθούν Γύρω από Κοινωνικά, Πολιτικά Ζητήματα;

Οι χρήστες των μέσων κοινωνικής δικτύωσης μπορούν να αυτοοργανωθούν γύρω από κοινωνικά, πολιτικά ή περιβαλλοντικά ζητήματα με διάφορους τρόπους. Αυτό επιτρέπει στους ανθρώπους να συνεργάζονται και να ασκούν

πίεση για αλλαγές σε αυτά τα πεδία. Παρακάτω παραθέτουμε μερικούς τρόπους που αυτό μπορεί να συμβεί:

Δημιουργία Ομάδων και Κοινοτήτων: Οι χρήστες μπορούν να δημιουργήσουν ομάδες και κοινότητες που να τους αφορούν συγκεκριμένα ζητήματα. Οι ομάδες αυτές συγκεντρώνουν ανθρώπους με κοινά ενδιαφέροντα και στόχους.

Εκδηλώσεις, Καμπάνιες Κινητοποίησης: Οι χρήστες μπορούν να χρησιμοποιήσουν τα μέσα κοινωνικής δικτύωσης για τη δημιουργία εκδηλώσεων, καμπανιών και κινητοποιήσεων. Μπορούν να δημιουργήσουν δημοφιλή περιεχόμενα σε post κλπ., να χρησιμοποιήσουν τα λεγόμενα hashtags και να καλέσουν τα άτομα που τους ακολουθούν να συμμετάσχουν σε δράσεις.



Εικόνα 2- Επανάσταση των μέσων κοινωνικής δικτύωσης

Διαμαρτυρίες και Πορείες: Τα διαδικτυακά κοινωνικά μέσα μπορούν να χρησιμοποιηθούν για τον συντονισμό διαμαρτυριών, διαδηλώσεων και εκδηλώσεων. Υπάρχει πλήρης και σαφής ενημέρωση για την ημερομηνία και την τοποθεσία των εκδηλώσεων.

Ενημέρωση και Ευαισθητοποίηση: Οι χρήστες μπορούν να χρησιμοποιήσουν τα μέσα κοινωνικής δικτύωσης για να ενημερώσουν/ενημερωθούν πάνω σε κοινωνικά, πολιτικά ή περιβαλλοντικά ζητήματα.

Διαμοιρασμός Εμπειριών: Οι χρήστες μπορούν να διαμοιράσουν τις προσωπικές τους εμπειρίες και ιστορίες σχετικά με αυτά τα ζητήματα. Αυτό μπορεί να δημιουργήσει συναισθηματική σύνδεση και συνειδητοποίηση.

3.2 Κοινωνική Αυτοοργάνωση με Γεωγραφικά Κριτήρια

Η "κοινωνική αυτοοργάνωση με γεωγραφικά κριτήρια" αναφέρεται στην ικανότητα των ανθρώπων να οργανώνονται και να δρουν σε κοινωνικά ή πολιτικά θέματα, λαμβάνοντας υπόψη τη γεωγραφική τους θέση και τοπικές ανάγκες. Αυτό το φαινόμενο έχει γίνει εξαιρετικά σημαντικό με την ανάπτυξη των ψηφιακών μέσων ενημέρωσης και των κοινωνικών δικτύων. Θα εξετάσουμε τον τρόπο με τον οποίο η κοινωνική αυτοοργάνωση με γεωγραφικά κριτήρια επηρεάζει την κοινότητα, τον ακτιβισμό και τη δημοκρατία.[21]

Ένα από τα κύρια πλεονεκτήματα της κοινωνικής αυτοοργάνωσης βασισμένη στην τοποθεσία των χρηστών είναι η δυνατότητα της τοπικής κοινότητας να ενωθεί γύρω από κοινά ζητήματα και προβλήματα. Οι πολίτες μπορούν να χρησιμοποιήσουν τα ψηφιακά μέσα για να διοργανώσουν τοπικές εκδηλώσεις, να αναζητήσουν λύσεις σε τοπικά προβλήματα και να ενημερώσουν την κοινότητα για τις εξελίξεις στον τοπικό τομέα. Αυτή η ενσωμάτωση της τοπικής διάστασης μπορεί να ενισχύσει την αλληλεγγύη και τη συνεργασία στην κοινότητα.[21]

Επιπλέον, επιτρέπει στους ανθρώπους να ασκήσουν ακτιβισμό σε τοπικό επίπεδο. Τα κοινωνικά δίκτυα και οι ψηφιακές πλατφόρμες επιτρέπουν στους ακτιβιστές να οργανώνονται γρήγορα και αποτελεσματικά για να αγωνιστούν για τα δικαιώματά τους ή να προωθήσουν αλλαγές στην τοπική πολιτική και κοινωνική πραγματικότητα. Αυτό έχει οδηγήσει σε επιτυχημένες εκστρατείες, όπως αυτές για τον περιβαλλοντικό ακτιβισμό, την προστασία των δασών, την καταπολέμηση της φτώχειας και πολλές άλλες.

Η κοινωνική αυτοοργάνωση βάση γεωγραφίας συμβάλλει στην ενίσχυση της δημοκρατίας. Η δυνατότητα των πολιτών να επικοινωνούν, να ανταλλάσσουν απόψεις, να ψηφίζουν και να ενημερώνονται.

3.2.1 Συμμετοχική Δημοκρατία

Η συμμετοχική δημοκρατία είναι μια προσέγγιση στη δημοκρατία όπου η συμμετοχή των πολιτών στη λήψη αποφάσεων και τη διαμόρφωση της πολιτικής ατζέντας είναι κεντρικό στοιχείο. Στη συμμετοχική δημοκρατία, οι πολίτες δεν είναι απλά παρατηρητές ή ψηφοφόροι, αλλά ενεργοί συμμετέχοντες στη διαμόρφωση και εφαρμογή των πολιτικών αποφάσεων[7].

Κάποια στοιχεία της συμμετοχικής δημοκρατίας περιλαμβάνουν:

1. Συμμετοχή των πολιτών

Στη συμμετοχική δημοκρατία, οι πολίτες έχουν την ευκαιρία να συμμετέχουν σε διαδικασίες λήψης αποφάσεων και να εκφράζουν τις απόψεις τους σε δημόσια θέματα.[7][18]

2. Ψηφοφορίες και διαβουλεύσεις

Οι κυβερνήσεις και άλλοι θεσμικοί φορείς μπορούν να διεξάγουν ψηφοφορίες και διαβουλεύσεις με τους πολίτες προκειμένου να συγκεντρώσουν απόψεις και να λάβουν αποφάσεις.[7]

3. Διαφάνεια

Η διαφάνεια είναι σημαντική στη συμμετοχική δημοκρατία, καθώς οι πληροφορίες και οι αποφάσεις πρέπει να είναι προσβάσιμες και κατανοητές από τους πολίτες

4. Ψηφιακή τεχνολογία

Η τεχνολογία, και ειδικά οι ψηφιακές πλατφόρμες και τα κοινωνικά μέσα, επιτρέπουν τη συμμετοχή των πολιτών σε διαδικασίες λήψης αποφάσεων και τη διευκόλυνση της διαβούλευσης.

5. Ανοικτή διακυβέρνηση

Οι πολιτικοί θεσμοί εργάζονται πιο στενά με τους πολίτες και την κοινωνία των πολιτών για τη λήψη αποφάσεων και την επίλυση προβλημάτων. Συνολικά, η συμμετοχική δημοκρατία ενισχύει την ενεργή συμμετοχή των πολιτών στην πολιτική διαδικασία και επιδιώκει να ενσωματώσει περισσότερες απόψεις και γνώμες στη δημοκρατική διακυβέρνηση.[7][18]

Η Συμμετοχική δημοκρατία όπως αναφέρεται και στο άρθρο «E-democracy: A new challenge for measuring democracy» μπορεί να συνδυαστεί με τη χρήση της τεχνολογίας, κυρίως του διαδικτύου και των μέσων κοινωνικής δικτύωσης, για την ενθάρρυνση της συμμετοχής των πολιτών στις πολιτικές διαδικασίες. Αυτό μπορεί να περιλαμβάνει τη διαβούλευση τη συμμετοχή σε αποφάσεις, και τη δημιουργία πολιτικού περιεχομένου.

Προκλήσεις στη «μέτρηση» της Δημοκρατίας*: Επισημαίνεται πως η ανάπτυξη της συμμετοχικής δημοκρατίας προκαλεί νέες προκλήσεις όσον αφορά τον τρόπο μέτρησης της δημοκρατίας. Η παραδοσιακή μέτρηση μπορεί να μην λαμβάνει υπόψη τις νέες μορφές συμμετοχής που προσφέρει η τεχνολογία.

Δημοκρατικές Αξίες: Η συμμετοχική δημοκρατία είναι συχνά συνυφασμένη με δημοκρατικές αξίες όπως η διαφάνεια, η ανοικτή διακυβέρνηση και η διαλειτουργικότητα.

Εργαλεία Μέτρησης: Το άρθρο επισημαίνει την ανάγκη ανάπτυξης νέων εργαλείων και μεθοδολογιών για τη μέτρηση της συμμετοχικής δημοκρατίας και την αξιολόγηση των αποτελεσμάτων της.

* Η "μέτρηση της δημοκρατίας" αναφέρεται στην προσπάθεια να κατανοήσουμε και να αξιολογήσουμε την υγεία και την αποτελεσματικότητα των δημοκρατικών διαδικασιών, δομών και θεσμών σε ένα κράτος ή σε μια κοινωνία. Στόχος της μέτρησης της δημοκρατίας είναι να παρέχει μια αντικειμενική και ποσοτική αξιολόγηση του βαθμού δημοκρατικής λειτουργίας και συμμετοχής σε μια κοινωνία.

3.2.2 Ηλεκτρονικές Ψηφοφορίες

Η ηλεκτρονική ψηφοφορία (e-voting) αναφέρεται στη χρήση της ηλεκτρονικής τεχνολογίας και του διαδικτύου για την ψηφοφορία σε εκλογικές διαδικασίες και αναφοράς σε δημοκρατικές αποφάσεις. Αν και η ιδέα της e-voting έχει το δυναμικό να βελτιώσει τη διαδικασία των εκλογών και να αυξήσει τη συμμετοχή των πολιτών, έχουν αναδυθεί και πολλές ανησυχίες και προκλήσεις που συνδέονται με αυτήν την τεχνολογία.

Το επιστημονικό άρθρο με τίτλο "Electronic Voting" παρουσιάζει τις βασικές αρχές και λειτουργίες της ηλεκτρονικής ψηφοφορίας όπως αναλύονται στο πλαίσιο του ερευνητικού έργου. Η ηλεκτρονική ψηφοφορία αποτελεί ένα σημαντικό εργαλείο στον τομέα της εκλογικής διαδικασίας και παρέχει τη δυνατότητα στους πολίτες να ψηφίσουν ηλεκτρονικά μέσω υπολογιστικών συστημάτων. Ακολουθούν οι κύριες αρχές και λειτουργίες της ηλεκτρονικής ψηφοφορίας βάσει του περιεχομένου του άρθρου:

- ❖ **Αυθεντικοποίηση του ψηφοφόρου:** Η ηλεκτρονική ψηφοφορία πρέπει να διασφαλίζει ότι ο ψηφοφόρος είναι πράγματι εκείνος που ισχυρίζεται ότι είναι, χρησιμοποιώντας μοναδικές αναγνωριστικές πληροφορίες, όπως ηλεκτρονικές υπογραφές ή κωδικούς πρόσβασης.
- ❖ **Ασφάλεια:** Πρέπει να ληφθούν μέτρα για την αποτροπή ανεπιθύμητων παρεμβάσεων στη διαδικασία ψηφοφορίας.
- ❖ **Εμπιστευτικότητα:** Η ψήφος του πολίτη πρέπει να παραμένει εμπιστευτική και να μην αποκαλύπτεται σε τρίτους. Η κρυπτογραφία είναι συχνά απαραίτητη για να διασφαλιστεί αυτή η εμπιστευτικότητα.

- ❖ Έλεγχος: Οι ψηφοφόροι πρέπει να έχουν τη δυνατότητα να επαληθεύσουν ότι η ψήφος τους καταγράφηκε σωστά και ότι δεν υπήρξε καμία παραβίαση του συστήματος.
- ❖ Συναίνεση: Οι ψηφοφόροι πρέπει να δίνουν τη συναίνεσή τους για να ψηφίσουν με αυτόν τον τρόπο. Δηλαδή, πρέπει να είναι ενήμεροι για τη χρήση της ηλεκτρονικής ψηφοφορίας και να συμφωνούν να τη χρησιμοποιήσουν.

Αυτές οι αρχές και λειτουργίες σχεδιάζονται και εφαρμόζονται στα ηλεκτρονικά συστήματα ψηφοφορίας για να διασφαλίσουν την ορθή λειτουργία τους και την αξιοπιστία των αποτελεσμάτων. Αυτό επιτρέπει στους πολίτες να ψηφίζουν με ασφάλεια και ευκολία, ενισχύοντας έτσι την εφαρμογή της ηλεκτρονικής ψηφοφορίας ως σημαντικού εργαλείου στην εκλογική διαδικασία.

Οφέλη της Ηλεκτρονικής Ψηφοφορίας:

- A. Εξοικονόμηση Χρόνου και Πόρων: Η ηλεκτρονική ψηφοφορία μειώνει τον ανθρώπινο και χρονικό πόρο που απαιτείται στις παραδοσιακές εκλογές. Οι ψήφοι μετρούνται αυτόματα και γρήγορα.
- B. Αυξημένη Συμμετοχή: Η διαθεσιμότητα της ηλεκτρονικής ψηφοφορίας μπορεί να ενθαρρύνει περισσότερους πολίτες να ψηφίσουν, ειδικά εκείνους που αντιμετωπίζουν δυσκολίες στην παραδοσιακή ψηφοφορία.
- Γ. Εξομάλυνση Εκλογικών Διαδικασιών: Η τεχνολογία μπορεί να βοηθήσει στην αποφυγή σφαλμάτων κατά την ψηφοφορία και τη μέτρηση, προσφέροντας πιο ακριβή αποτελέσματα.

Κινδύνοι της Ηλεκτρονικής Ψηφοφορίας:

- A. Κυβερνοασφάλεια: Η ασφάλεια των ψηφοφοριών αποτελεί σημαντική πρόκληση. Κυβερνοαπειλές μπορούν να απειλήσουν την ακεραιότητα της διαδικασίας και τα αποτελέσματα.[11]
- B. Κοινωνικός Αποκλεισμός: Η ηλεκτρονική ψηφοφορία απαιτεί πρόσβαση σε τεχνολογία και σύνδεση στο Διαδίκτυο. Αυτό μπορεί να αποκλείσει ορισμένες κοινωνικές ομάδες που δεν έχουν εύκολη πρόσβαση σε αυτούς τους πόρους.[11]
- Γ. Εμπιστοσύνη: Η αξιοπιστία της ηλεκτρονικής ψηφοφορίας πρέπει να διατηρηθεί για να εξασφαλιστεί η εμπιστοσύνη του κοινού. Οποιαδήποτε αμφιβολία για την ακεραιότητα του συστήματος μπορεί να αποθαρρύνει τους ψηφοφόρους.[11]

3.2.3 Ανάδειξη Εκπροσώπου Κοινότητας

Το άρθρο που δημοσιεύεται στην ιστοσελίδα της Zeus Ηλεκτρονικής Ψηφοφορίας αναφέρεται στον ρόλο του εκπροσώπου, στο πλαίσιο των ηλεκτρονικών ψηφοφοριών. Ο εκπρόσωπος αναλαμβάνει τον ρόλο του του διαμεσολαβητή μεταξύ των ψηφοφόρων και του διαμεσολαβητή μεταξύ των ψηφοφόρων και του συστήματος ψηφοφορίας, διασφαλίζοντας την ασφάλεια και την εγκυρότητα των ψήφων.

Το άρθρο περιλαμβάνει ερωτήσεις που συχνά καθίστανται σημαντικές κατά τη διάρκεια μιας ηλεκτρονικής ψηφοφορίας. Συγκεκριμένα, αναφέρεται σε θέματα που αφορούν την προετοιμασία και την εκτέλεση της ψηφοφορίας, καθώς και τις απαιτήσεις που πρέπει να πληροί ένας εκπρόσωπος. Ο εκπρόσωπος, πρέπει να διασφαλίζει την ασφάλεια του ψηφοφόρου, εξασφαλίζοντας ότι η ταυτότητά του επιβεβαιώνεται πριν από την ψηφοφορία, καθώς και την ακεραιότητα της ψήφου του κατά τη διαμεταγγή της προς το σύστημα ψηφοφορίας. Επιπλέον, αναλαμβάνει την ευθύνη της διανομής των εκλογικών ψήφων στο σύστημα ψηφοφορίας.

Αναδεικνύεται η σημασία του εκπροσώπου ως ενός ενδιάμεσου μεταξύ των ψηφοφόρων και του συστήματος ψηφοφορίας για την ασφάλεια, την αξιοπιστία και την εγκυρότητα της διαδικασίας ψηφοφορίας. Επίσης, αναφέρεται σε τεχνικές λεπτομέρειες και διαδικαστικά θέματα που αφορούν τον εκπρόσωπο κατά την διεξαγωγή της ψηφοφορίας.

Συνοψίζοντας τα δύο προηγούμενα κεφάλαια, για την επιλογής της λειτουργίας της ψηφοφορίας καθώς και την ανάδειξη του εκπροσώπου της κοινότητας, επιλέξαμε να χρησιμοποιήσουμε την μέθοδο της ανοιχτής ψηφοφορίας όπως αναδεικνύεται και στη λειτουργία μιας συνέλευσης ενός κοινωνικού συνόλου.

4. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΚΟΙΝΩΝΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

Στην ενότητα 4 θα παρουσιάσουμε τα εργαλεία και τις τεχνολογίες ανάπτυξης για εφαρμογές. Θα γίνει σύγκριση αυτών και η επιλογή των καταλληλότερων για την παρούσα πτυχιακή εργασία.

4.1. Περιβάλλον Ανάπτυξης Εφαρμογών (IDE)

Σύγκριση Περιβαλλόντων Ανάπτυξης Κώδικα (IDE: Integrated Development Environments) για εφαρμογές κινητών τηλεφώνων:

Οι επιλογές περιβάλλοντος ανάπτυξης κώδικα για κινητές εφαρμογές έχουν διάφορα πλεονεκτήματα και μειονεκτήματα, και η επιλογή εξαρτάται από τις ανάγκες και τον στόχο της εφαρμογής. Εδώ είναι μια σύντομη σύγκριση μεταξύ των πιο δημοφιλών περιβαλλόντων ανάπτυξης για κινητές εφαρμογές:

Τα Android Studio, Xcode και Visual Studio Code είναι τρία δημοφιλή περιβάλλοντα ανάπτυξης (IDE) που χρησιμοποιούνται για τη δημιουργία εφαρμογών για κινητές συσκευές. Είναι υποψήφια μιας και είναι δωρεάν για χρήση. Ας συγκρίνουμε και αναλύσουμε αυτά τα τρία IDE:

Android Studio:

Πλεονεκτήματα:

1. Επίσημο περιβάλλον ανάπτυξης από τη Google, σχεδιασμένο ειδικά για το Android.
2. Πλούσια τεκμηρίωση και υποστήριξη από την Google και την κοινότητα.
3. Ισχυρά εργαλεία για τον σχεδιασμό του χρήστη (UI), όπως το Android Layout Editor.
4. Δυνατότητα ανάπτυξης εφαρμογών για πολλές συσκευές Android.

Μειονεκτήματα:

1. Περιορισμένη δυνατότητα ανάπτυξης εφαρμογών για την πλατφόρμα iOS.

2. Απαιτεί γνώση της γλώσσας προγραμματισμού Java (ή Kotlin) για την ανάπτυξη εφαρμογών Android.
3. Είναι αρκετά πιο βαρύ πρόγραμμα συγκριτικά με τα άλλα 2.

Xcode:

Πλεονεκτήματα:

1. Επίσημο περιβάλλον ανάπτυξης από την Apple, σχεδιασμένο ειδικά για το iOS, το macOS, το watchOS και το tvOS.
2. Βαθιά ενσωματωμένη με το οικοσύστημα της Apple, προσφέροντας την καλύτερη συμβατότητα και απόδοση στις συσκευές της Apple.
3. Πολύ καλή υποστήριξη για την ανάπτυξη εφαρμογών για τις πλατφόρμες της Apple.

Μειονεκτήματα:

1. Περιορισμένη δυνατότητα ανάπτυξης εφαρμογών για τα Android.
2. Απαιτεί γνώση της γλώσσας προγραμματισμού Swift (ή Objective-C) για την ανάπτυξη εφαρμογών iOS.

Visual Studio Code (VS Code):

Πλεονεκτήματα:

1. Πολυλειτουργικό: To VS Code είναι πολυλειτουργικό και μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών για πολλές πλατφόρμες, συμπεριλαμβανομένων των Android και iOS με τη χρήση επεκτάσεων.
2. Δωρεάν και ανοικτού κώδικα: To VS Code είναι δωρεάν και ανοικτού κώδικα, προσφέροντας πολλές επεκτάσεις και πρόσθετα εργαλεία.

3. Κοινότητα: Διαθέτει μια ενεργή και μεγάλη κοινότητα χρηστών, που σημαίνει υποστήριξη και επεξηγήσεις για προβλήματα.

Μειονεκτήματα:

Δεν έχει την ίδια ενσωμάτωση με τα επίσημα περιβάλλοντα ανάπτυξης Android Studio και Xcode, οπότε μπορεί να χρειάζεται περισσότερη ρύθμιση για κάθε πλατφόρμα.

4.2 Framework/Γλώσσα

Flutter:

Πλεονεκτήματα:

1. Συγκέντρωση σε πολλές πλατφόρμες: To Flutter σας επιτρέπει να δημιουργήσετε εφαρμογές για πολλές πλατφόρμες (Android, iOS, Web και Desktop) με έναν κοινό κώδικα, μειώνοντας τον χρόνο και το κόστος ανάπτυξης.
2. Σχεδιασμός UI με πλούσια εργαλεία: To Flutter προσφέρει εξαιρετικά εργαλεία για τον σχεδιασμό του χρήστη (UI) με χρήση του Material Design (για Android) και του Cupertino (για iOS), δίνοντάς σας έναν εύκολο τρόπο να δημιουργήσετε όμορφες εφαρμογές.
3. Ενσωματωμένος εξομοιωτής: To Flutter διαθέτει ενσωματωμένους εξομοιωτές για όλες τις υποστηριζόμενες πλατφόρμες, που επιτρέπουν να δοκιμαστεί η εφαρμογή χωρίς την ανάγκη φυσικού συστήματος.
4. Ταχεία ανάπτυξη: Η hot-reload λειτουργία του Flutter σας επιτρέπει να δει ο χρήστης αμέσως τις αλλαγές που έκανε στον κώδικα, γεγονός που γλυτώνει χρόνο στον developer.

Μειονεκτήματα:

1. ****Επιπλέον μέγεθος αρχείου**:** Οι εφαρμογές που αναπτύσσονται με Flutter μπορεί να έχουν μεγαλύτερο μέγεθος αρχείου σε σύγκριση με αυτές που αναπτύσσονται με τα επίσημα περιβάλλοντα ανάπτυξης των πλατφορμών (Android Studio, Xcode).
2. ****Εκμάθηση της γλώσσας προγραμματισμού Dart**:** To Flutter χρησιμοποιεί τη γλώσσα προγραμματισμού Dart, η οποία δεν είναι τόσο διαδεδομένη όσο άλλες γλώσσες, όπως η Java ή η Swift, και ενδέχεται να απαιτήσει μια επιπλέον περίοδο εκμάθησης για τους προγραμματιστές.

JAVA

Η Java είναι μια από τις κύριες γλώσσες προγραμματισμού που χρησιμοποιούνται για την ανάπτυξη εφαρμογών για κινητές συσκευές, όπως έξυπνα κινητά και tablets:

Πλεονεκτήματα της Java:

Συμβατότητα: Η Java είναι γνωστή για την υψηλή της συμβατότητα με διάφορες πλατφόρμες. Ο κώδικας που γράφετε σε Java μπορεί να τρέξει σε πολλές διαφορετικές συσκευές Android χωρίς να απαιτείται μεγάλος κώδικας για προσαρμογή.

Μεγάλη κοινότητα: Η Java έχει μια τεράστια κοινότητα προγραμματιστών, που σημαίνει ότι υπάρχουν πολλοί πόροι, βιβλιοθήκες, εργαλεία και εκπαιδευτικό υλικό διαθέσιμα για τους αναπτυσσόμενους.

Ασφάλεια: Η Java έχει σχεδιαστεί με έμφαση στην ασφάλεια. Αυτό την καθιστά κατάλληλη για την ανάπτυξη εφαρμογών που απαιτούν υψηλά επίπεδα ασφάλειας, όπως τραπεζικές εφαρμογές και εφαρμογές υγείας.

Διαχείριση μνήμης: Η Java χρησιμοποιεί αυτόματη διαχείριση μνήμης, γεγονός που μειώνει τον κίνδυνο σφαλμάτων που σχετίζονται με τη διαχείριση της μνήμης.



Eikόνα 3 - JAVA

Μειονεκτήματα της Java:

Αποδοτικότητα: Ενώ η Java είναι ασφαλής και ευκολότερη στην ανάπτυξη, μπορεί να μην είναι τόσο αποδοτική όσο κάποιες άλλες γλώσσες όταν πρόκειται για απαιτητικές εφαρμογές που απαιτούν υψηλή απόδοση.

Περιορισμένος έλεγχος συσκευής: Λόγω της αφαιρετικής φύσης της Java, ενδέχεται να υπάρχουν περιορισμοί στον έλεγχο των λειτουργιών της συσκευής, όπως των κινητήρων της κάμερας ή των αισθητήρων.

Καθυστέρηση ενημερώσεων: Σε ορισμένες περιπτώσεις, οι ενημερώσεις της Java μπορεί να καθυστερήσουν να εφαρμοστούν σε όλες τις συσκευές, προκαλώντας προβλήματα συμβατότητας.

KOTLIN

Η Kotlin είναι μια γλώσσα προγραμματισμού που δημιουργήθηκε από την JetBrains και κυκλοφόρησε για πρώτη φορά το 2011. Έχει γρήγορα αυξανόμενη

δημοτικότητα και έχει γίνει η επίσημη γλώσσα ανάπτυξης εφαρμογών Android από το 2017.

Πλεονεκτήματα της Kotlin:

Απλότητα και ευανάγνωστος κώδικας: Η Kotlin είναι γνωστή για την απλότητα και την ευανάγνωστη σύνταξή της. Ο κώδικας σε Kotlin είναι συχνά πιο σύντομος και ευανάγνωστος από την Java, χωρίς την ανάγκη για πολλούς λογικούς φράγματα (boilerplate code).

100% συμβατότητα με Java: Μια από τις σημαντικότερες πτυχές της Kotlin είναι η πλήρης συμβατότητα με την Java. Αυτό σημαίνει ότι μπορείτε να χρησιμοποιήσετε την Kotlin σταδιακά σε υπάρχοντα Java projects και να αλληλεπιδράσετε με υπάρχοντα Java βιβλιοθήκες.

Ασφάλεια: Η Kotlin ενισχύει την ασφάλεια του κώδικα, προσφέροντας εργαλεία όπως τον τύπο δεδομένων "Nullable" που βοηθά στην αντιμετώπιση των σφαλμάτων που συνήθως σχετίζονται με τις αναφορές null.

Σύγχρονη γλώσσα: Η Kotlin συμπεριλαμβάνει πολλές σύγχρονες χαρακτηριστικά της γλώσσας προγραμματισμού, όπως lambda expressions, parametric types, και ενσωματωμένες συναρτήσεις (extension functions).

Διαφορετικές πλατφόρμες: Εκτός από την ανάπτυξη Android, η Kotlin μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε διάφορες πλατφόρμες, όπως χρήση για διαδικτυακές εφαρμογές και desktop εφαρμογών.



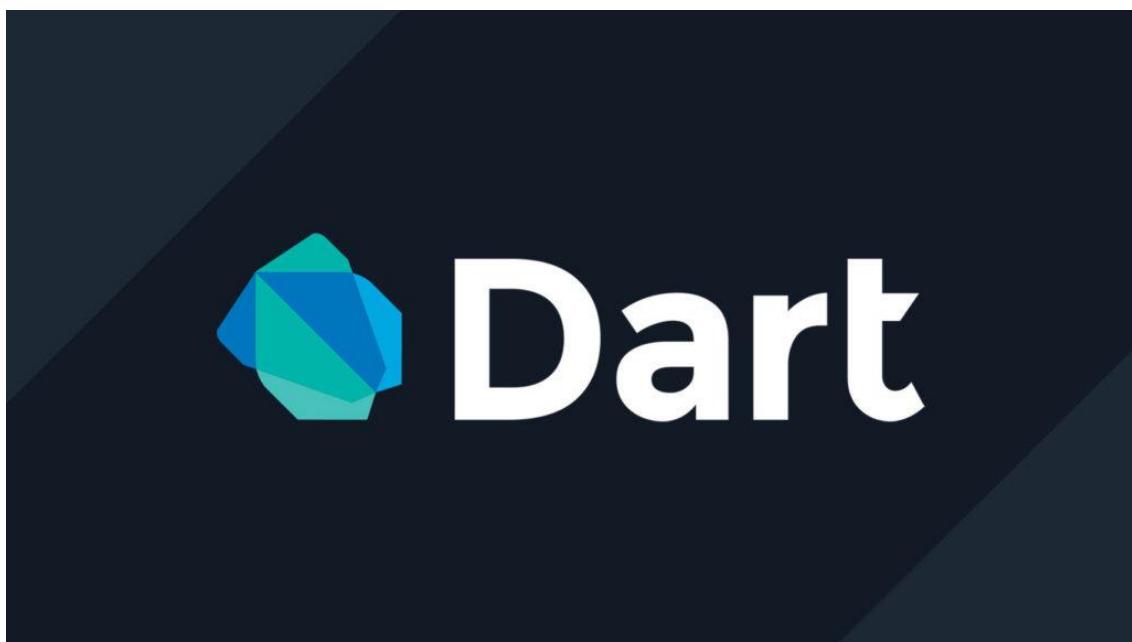
Eikόνα 4 - kotlin

Μειονεκτήματα της Kotlin:

Παρόλο που η Kotlin έχει πολλά πλεονεκτήματα, δεν είναι η κατάλληλη γλώσσα για κάθε εργασία. Σημαντικό ότι Kotlin έχει μικρότερη κοινότητα σε σχέση με την Java, αλλά αυτό αλλάζει γρήγορα καθώς αυξάνεται η δημοτικότητά της.

DART

Η Dart είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε από την Google και χρησιμοποιείται ευρέως για την ανάπτυξη κινητών εφαρμογών, κυρίως με το framework της Flutter. Ακολουθούν πλεονεκτήματα και χαρακτηριστικά της Dart για την ανάπτυξη εφαρμογών για κινητές συσκευές:



Eikόνα 5 - Dart

Πλεονεκτήματα της Dart:

Σύγχρονη Γλώσσα: Η Dart είναι μια σύγχρονη γλώσσα προγραμματισμού που συνδυάζει στοιχεία από άλλες γλώσσες, όπως η Java και η JavaScript. Αυτό την

καθιστά ευανάγνωστη και ευκολομάθητη από προγραμματιστές με διάφορα υπόβαθρα.

Επίσημη Γλώσσα του Flutter: Η Dart είναι η επίσημη γλώσσα που χρησιμοποιείται με το IDE Flutter για την ανάπτυξη κινητών εφαρμογών. Το Flutter είναι γνωστό για την αποτελεσματική ανάπτυξη πολλαπλών εφαρμογών καθώς και την απόδοση τους.

Γρήγορη Εκτέλεση: Η Dart είναι μια γλώσσα που μεταγλωτίζεται σε native κώδικα, οδηγώντας σε γρήγορη εκτέλεση και απόδοση εφαρμογών.

Εφαρμογές για Πολλαπλές Πλατφόρμες: Χάρη στο Flutter, ένας χρήστης μπορεί να δημιουργήσει εφαρμογές που λειτουργούν σε πολλαπλές πλατφόρμες, όπως Android, iOS, web και desktop, χρησιμοποιώντας τον ίδια κώδικα.

Κοινότητα: Παρά το γεγονός ότι η Dart δεν έχει τη δημοτικότητα της Java ή της Swift, έχει μια ακτινοβόλο και αυξανόμενη κοινότητα προγραμματιστών που συνεισφέρουν στην ανάπτυξη της.

Μειονεκτήματα της Dart:

Περιορισμένο Οικοσύστημα Βιβλιοθηκών: Ενώ ο αριθμός των βιβλιοθηκών και πακέτων για την Dart αυξάνεται, εξακολουθεί να είναι περιορισμένος σε σύγκριση με άλλες γλώσσες όπως η Java.

Μικρότερη Κοινότητα: Παρόλο που η κοινότητα της Dart αυξάνεται, είναι πιο μικρή σε σύγκριση με τις πιο δημοφιλείς γλώσσες προγραμματισμού, όπως η Java ή η JavaScript.

Περιορισμένη Συμβατότητα: Ενώ ο προσδιορισμός της Dart στο Flutter την καθιστά καλή για την ανάπτυξη κινητών εφαρμογών, είναι λιγότερο κατάλληλη για άλλους τύπους ανάπτυξης κώδικα, όπως για διαδικτυακές εφαρμογές.

SWIFT

Η Swift είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε από την Apple για την ανάπτυξη κινητών εφαρμογών και εφαρμογών για το macOS, το watchOS και το tvOS:



Eikόνα 6 - Swift

Πλεονεκτήματα της Swift:

Εύχρηστη και Ευανάγνωστη: Η Swift σχεδιάστηκε με έμφαση στην ευανάγνωστη σύνταξη του κώδικα, κάτι που την καθιστά ευκολότερη για τους προγραμματιστές να την αναπτύξουν και να την κατανοήσουν.

Ασφαλής και Αξιόπιστη: Η Swift προσφέρει ασφαλές προγραμματισμό, με ενσωματωμένα χαρακτηριστικά για τον περιορισμό των σφαλμάτων κατά τον χρόνο εκτέλεσης. Αυτό μειώνει τις πιθανότητες ατυχημάτων και προβλημάτων ασφαλείας.

Γρήγορη Απόδοση: Η Swift έχει σχεδιαστεί για να είναι γρήγορη και αποδοτική στην εκτέλεση, καθιστώντας την ιδανική για απαιτητικές εφαρμογές και παιχνίδια.

Ενεργή Κοινότητα: Έχει ενεργή και αυξανόμενη κοινότητα προγραμματιστών που συνεισφέρουν στην ανάπτυξη και βελτίωση της Swift και της δημιουργούν καινούργιες βιβλιοθήκες και εργαλεία.

Μειονεκτήματα της Swift:

Περιορισμένη Εμβέλεια: Η Swift είναι κυρίως σχεδιασμένη για το λειτουργικό σύστημα της Apple, πράγμα που σημαίνει ότι είναι περιορισμένη στην ανάπτυξη εφαρμογών που τρέχουν αποκλειστικά σε πλατφόρμες της Apple.

Μάθηση: Η μάθηση της Swift απαιτεί χρόνο και αφοσίωση, ειδικά για ένα νέο προγραμματιστή.

Επεκτασιμότητα: Η Swift δεν έχει ακόμη την ίδια επεκτασιμότητα και τον αριθμό των βιβλιοθηκών που υπάρχουν για άλλες γλώσσες, όπως η Java.

REACT NATIVE

Η React Native είναι ένα ανοικτού κώδικα πλαίσιο ανάπτυξης (framework) που χρησιμοποιείται για τη δημιουργία κινητών εφαρμογών για τις πλατφόρμες iOS και Android:

Πλεονεκτήματα της React Native:

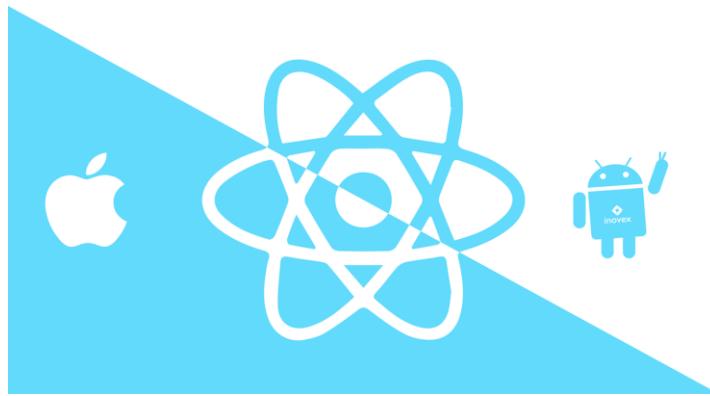
Διακοινοποίηση Κώδικα: Ένα από τα μεγάλα πλεονεκτήματα του React Native είναι ότι μπορείτε να χρησιμοποιήσετε την ίδια βάση κώδικα για τις εφαρμογές iOS και Android. Αυτό σημαίνει ότι μπορείτε να εξοικονομήσετε χρόνο και πόρους στην ανάπτυξη.

Γρήγορη Ανάπτυξη: Η React Native προσφέρει γρήγορη ανάπτυξη εφαρμογών με τη δυνατότητα ζωντανής ανανέωσης (live reload). Αυτό επιτρέπει στον χρήστη να δεί τις αλλαγές στον κώδικα αμέσως, χωρίς την ανάγκη για επαναφόρτωση της εφαρμογής.

Εξοικονόμηση Χρόνου: επαναχρησιμοποίηση των components για την επιτάχυνση της ανάπτυξης της εφαρμογής. Επίσης, διαθέτει μεγάλο πλήθος από διάφορες βιβλιοθήκες.

Κοινότητα και Εργαλεία: Υπάρχει μια ενεργή κοινότητα προγραμματιστών που υποστηρίζει την React..

Επίσημη Υποστήριξη από τη Facebook: Η React Native υποστηρίζεται από το Facebook, το οποίο είναι υπεύθυνο για τη συνεχή ανάπτυξη και βελτίωση της.



Εικόνα 7 - React Native Logo

Μειονεκτήματα της React Native:

Περιορισμένες Επιδόσεις: Ενώ η React Native προσφέρει καλές επιδόσεις, ορισμένες προηγμένες λειτουργίες μπορεί να απαιτούν τη χρήση κατασκευαστικών επεκτάσεων (native modules) για τη βελτιστοποίηση της απόδοσης.

Εξαρτημένο από Third-Party Libraries: Κάποιες λειτουργίες μπορεί να απαιτούν τη χρήση τρίτων βιβλιοθηκών, οι οποίες ενδέχεται να μην είναι πάντα συμβατές με τις νεότερες εκδόσεις της React Native.

Μικρότερη Κοινότητα από τις Επίσημες Γλώσσες: Ενώ οι γλώσσες όπως η Java έχουν μεγάλη κοινότητα προγραμματιστών, η React Native διαθέτει μια μικρότερη κοινότητα σε σχέση με αυτές.

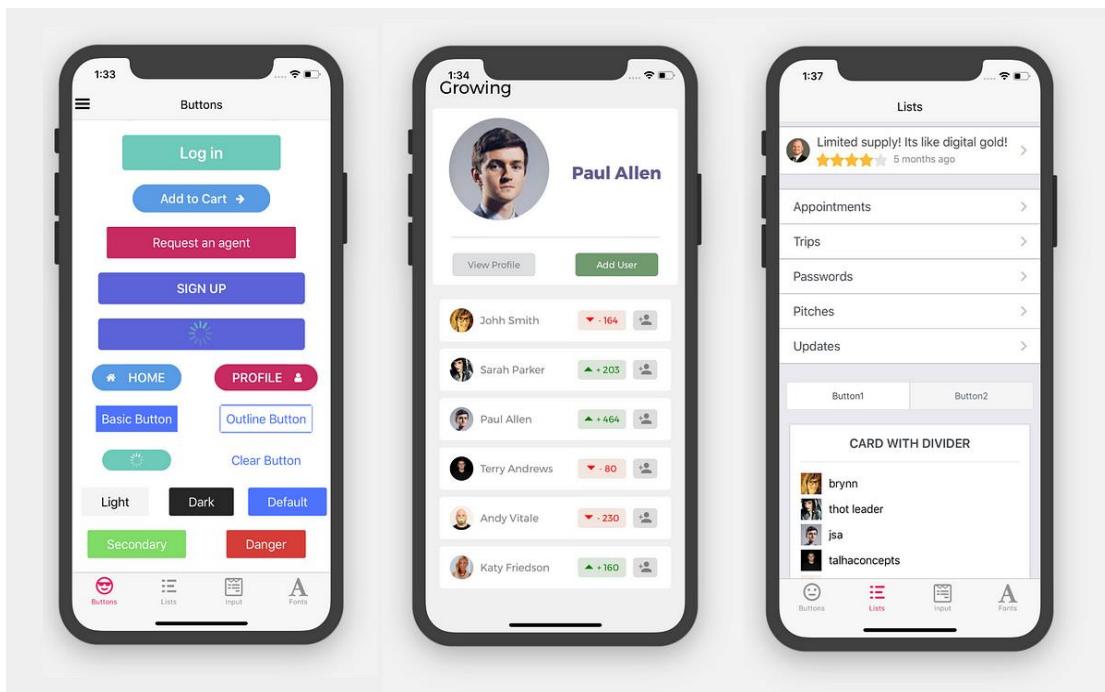
4.3 Επιλογή Πλατφόρμας για τη Διπλωματική Εργασία

Για την παρούσα διπλωματική εργασία και την ανάπτυξη της εφαρμογής για έξυπνα κινητά επιλέχθηκε η React Native στο περιβάλλον Visual Studio Code. Η επιλογή της React Native σε συνδυασμό με το περιβάλλον ανάπτυξης Visual Studio Code (VS Code) για τη δημιουργία ενός κινητού εφαρμογής βασίστηκε σε διάφορους σημαντικούς παράγοντες που συμβάλλουν στην αποτελεσματική και

αποδοτική διαδικασία ανάπτυξης. Βασικό χαρακτηριστικό και των 2 είναι πως διατείθονται δωρεάν και είναι αρκετά εύχρηστα.

Πρώτον, η React Native επιτρέπει τη δημιουργία εφαρμογών για τις πλατφόρμες iOS και Android με κοινό κώδικα. Συνεπάγεται σημαντική εξοικονόμηση χρόνου και πόρων καθώς αποφεύγεται η ανάγκη να γράφεται κώδικας για κάθε περίπτωση ξεχωριστά..

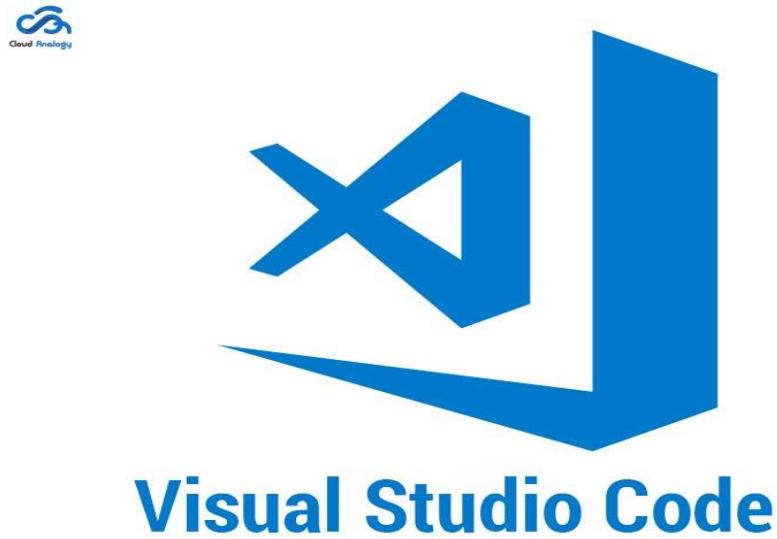
Δεύτερον, το περιβάλλον ανάπτυξης Visual Studio Code είναι ένα από τα δημοφιλέστερα και πανεύκολα προσαρμόσιμα IDE (Integrated Development Environments) που υποστηρίζει την React Native. Ο λόγος πίσω από αυτή την επιλογή είναι η ευχρηστία, η εκτεταμένη κοινότητα χρηστών και οι δυνατότητες προσαρμογής. Το VS Code παρέχει αναδιαρθρωμένο και ελαφρύ περιβάλλον ανάπτυξης, συνδεόμενο με πολλά επιπλέον εργαλεία και πρόσθετες επεκτάσεις που ενισχύουν τη δυνατότητα ανάπτυξης εφαρμογών.



Εικόνα 8- React Native

Τρίτον, η επιλογή της React σε συνδυασμό με το VS Code συμβάλλει στην ευελιξία και στην ταχύτητα προώθησης των εφαρμογών. Οι λειτουργίες από live reload επιτρέπουν βοηθάει τον χρήστη να δει αμέσως τις αλλαγές που κάνει στον κώδικα.

Συνολικά, η επιλογή της React Native σε συνδυασμό με το Visual Studio Code έγινε με βάση την επιδίωξη της αποτελεσματικότητας, της ταχύτητας και της ευχρηστίας στη διαδικασία ανάπτυξης και το γεγονός ότι παρέχουν ένα ισχυρό περιβάλλον για την δημιουργία μίας εφαρμογής για έξυπνα κινητά.



Εικόνα 9 - Visual Studio Code

4.4 Επιπρόσθετες Τεχνολογίες και Εργαλεία Συστήματος

Expo

Το Expo είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητές συσκευές που βασίζεται στο React Native. Ένα Expo Project προσφέρει μια ενδιαφέρουσα εμπειρία ανάπτυξης εφαρμογών, ιδίως για αναπτυσσόμενους εφαρμογές μικρού ή μεσαίου μεγέθους. Ένα από τα σημαντικότερα πλεονεκτήματα του Expo είναι το Expo Go, μια εφαρμογή που σας επιτρέπει να δοκιμάσετε το έργο σας σε πραγματικές συσκευές, χωρίς την ανάγκη για επίπονη εγκατάσταση σε προσωμοιωτές.

Στην παρούσα πτυχιακή εργασία χρησιμοποιήθηκε για την δημιουργία του expo project

Cloud Firestore

To Cloud Firestore είναι μία βάση δεδομένων εγγράφων NoSQL που επιτρέπει την αποθήκευση, τον συγχρονισμό και την εύκολη αναζήτηση δεδομένων σε εφαρμογές για κινητά και σε παγκόσμιο ιστό. Οι τύποι βάσεων δεδομένων NoSQL είναι ειδικά ιδανικοί για εφαρμογές που απαιτούν μεγάλο όγκο δεδομένων. Η Cloud Firestore παρέχεται από την Firebase της Google και παρέχει δυνατότητες όπως τον συγχρονισμό για την ενημέρωση των δεδομένων μεταξύ συσκευών, εντός ή εκτός σύνδεσης, την εύκολη προσαρμογή χωρίς διακομιστές, την ευκαμψία ως προς την δομή των δεδομένων και την εύκολη ανάκτηση των δεδομένων με φίλτραρισμα και ταξινόμηση.

Στη παρούσα διπλωματική εργασία η Firestore χρησιμοποιήθηκε ως βάση για την αποθήκευση δεδομένων αλληλεπίδρασης μεταξύ χρηστών της περιοχής

Authentication

Το authentication ή αλλιώς ο έλεγχος ταυτότητας είναι η διαδικασία αναγνώρισης της ταυτότητας ενός χρήστη. Είναι μία συσχέτιση μεταξύ μίας εισερχόμενης αίτησης με ένα σύνολο διαπιστευτηρίων αναγνώρισης. Τα διαπιστευτήρια που παρέχονται συγκρίνονται με αυτά που είναι αποθηκευμένα σε μία βάση δεδομένων. Ο συνδυασμός ονόματος χρήστη ή το προσωπικό του email και του κωδικού πρόσβασης είναι ο πιο δημοφιλής έλεγχος ταυτότητας και είναι γνωστός ως έλεγχος ταυτότητας κωδικού πρόσβασης.

Στη παρούσα πτυχιακή εργασία για τον έλεγχο ταυτότητας του χρήστη χρησιμοποιήσαμε το Firebase Authentication που μας παρέχει η Firebase και από τους διαθέσιμους τρόπους σύνδεσης και ελέγχου καταλήξαμε στην σύνδεση με email και κωδικούς πρόσβασης, στοιχεία τα οποία συμπληρώνει ο χρήστης στη φόρμα εγγραφής

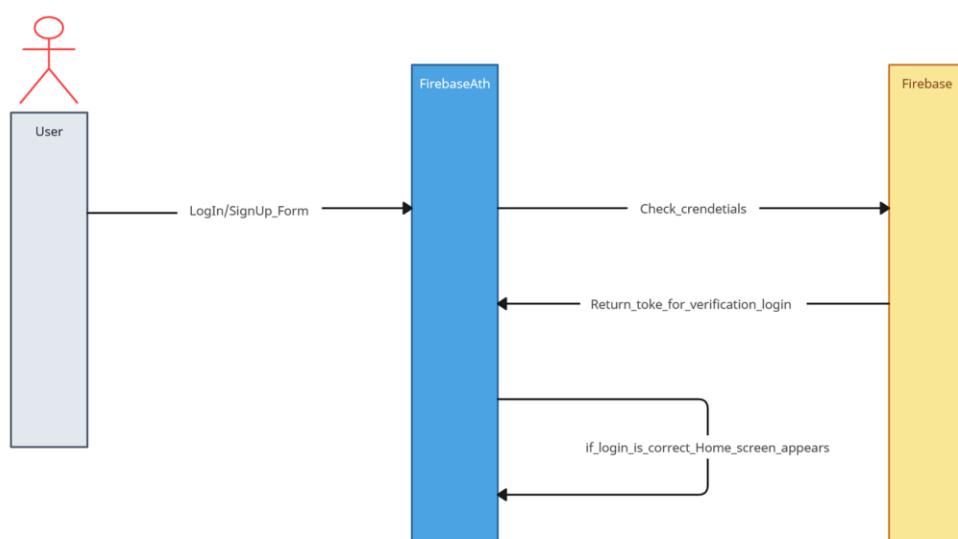
5. ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ

Στο κεφάλαιο αυτό θα αναφερθούμε στην αρχιτεκτονική του συστήματος. Παρουσιάζονται όλα τα απαραίτητα διαγράμματα για ολόκληρο το σύστημα που αφορά τη παρούσα διπλωματική εργασία, προκειμένου να γίνει πιο εύκολα κατανοητή η υλοποίηση και οι μέθοδοι που ακολουθήσαμε. Στη συνέχεια, ακολουθεί μία σύντομη περιγραφή των τεχνολογιών και των εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής και τέλος, αναφέρονται οι παράμετροι λειτουργίας της πλατφόρμας.

5.1 Σχεδιασμός Συστήματος Αυθεντικοποίησης

Διάγραμμα ελέγχου ταυτότητας.

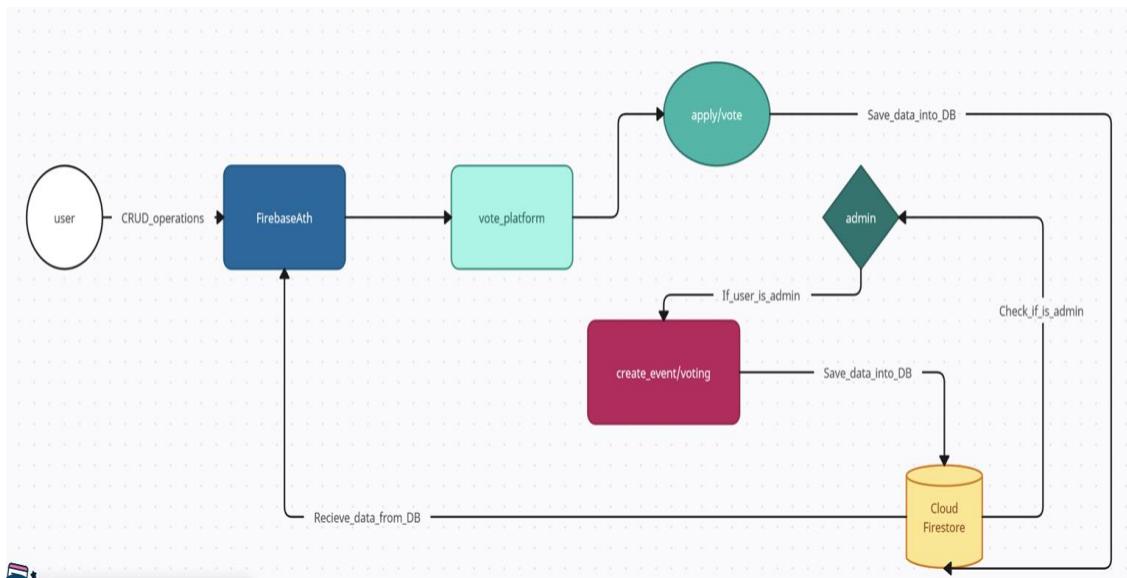
Στόχος του διαγράμματος ελέγχου ταυτότητας είναι να περιγράψει τη διαδικασία που ακολουθείται για να συνδεθεί ο χρήστης στην εφαρμογή για να μπεί στο room που αντιστοιχεί στον ταχυδρομικό κώδικα στον οποίο βρίσκεται η συσκευή του. Ουσιαστικά ο χρήστης ανοίγει την εφαρμογή, εμφανίζεται μία φόρμα όπου θα πρέπει να βάλει τα διαπιστευτήρια του, δηλαδή το email και τους κωδικούς που έχει δηλώσει στην φόρμα εγγραφής. Με τη βοήθεια του Firebase Authentication Service ελέγχονται τα διαπιστευτήρια του χρήστη που έδωσε και στη συνέχεια επιστρέφεται το διακριτικό για την επαλήθευση της σύνδεσης. Μόλις ο χρήστης συνδεθεί, μπορεί να περιηγηθεί στο room της περιοχής.



Εικόνα 10- UML διάγραμμα ελέγχου ταυτότητας

Διάγραμμα δραστηριότητας

Στόχος του διαγράμματος δραστηριότητας είναι να δείξει τις κύριες λειτουργίες της εφαρμογής. Μόλις ο χρήστης συνδεθεί στην εφαρμογή έχει τη δυνατότητα να δει τις υπενθυμίσεις, τα events και τις αποφάσεις της κοινότητας. Στη συνέχεια μπορεί να ψηφίσει, να συνομιλήσει και να επεξεργαστεί το προφίλ του.



Εικόνα 11- UML διάγραμμα δραστηριότητας

5.2 Σχεδιασμός Βάσης Δεδομένων

Στόχος του Database Diagram είναι να δείξει τη δομή της βάσης δεδομένων και τις σχέσεις μεταξύ των αντικειμένων της. Η Firestore της Google Firebase είναι μη-σχεσιακή βάση δεδομένων και δομεί εύκολα τα δεδομένα με συλλογές γνωστές και ως collections και έγγραφα γνωστά ως documents. Η δομή της μη-σχεσιακής βάσης δεδομένων Firestore έχει τη δομή που φαίνεται παρακάτω. Ο βασικός πίνακας από τον οποίο ξεκινάει όλη η διασύνδεση με τους επιμέρους πίνακες είναι ο «Community», στον οποίο βρίσκονται όλοι οι Ταχυδρομικοί κώδικες που έχουν δημιουργηθεί από την αλληλεπίδραση των χρηστών με την εφαρμογή. Ο πίνακας αυτός είναι ο κορμός όλης της βάσης και ονομάζεται collection και περιέχει τα documents. Μέσα σε κάθε Document “TK” βρίσκονται όλοι οι χρήστες που έχουν συνδεθεί στην εκάστοτε γεωγραφική περιοχή, καθώς και όλα τα δεδομένα κάθε περιοχής.. Κάθε document περιέχει τα subcollections:

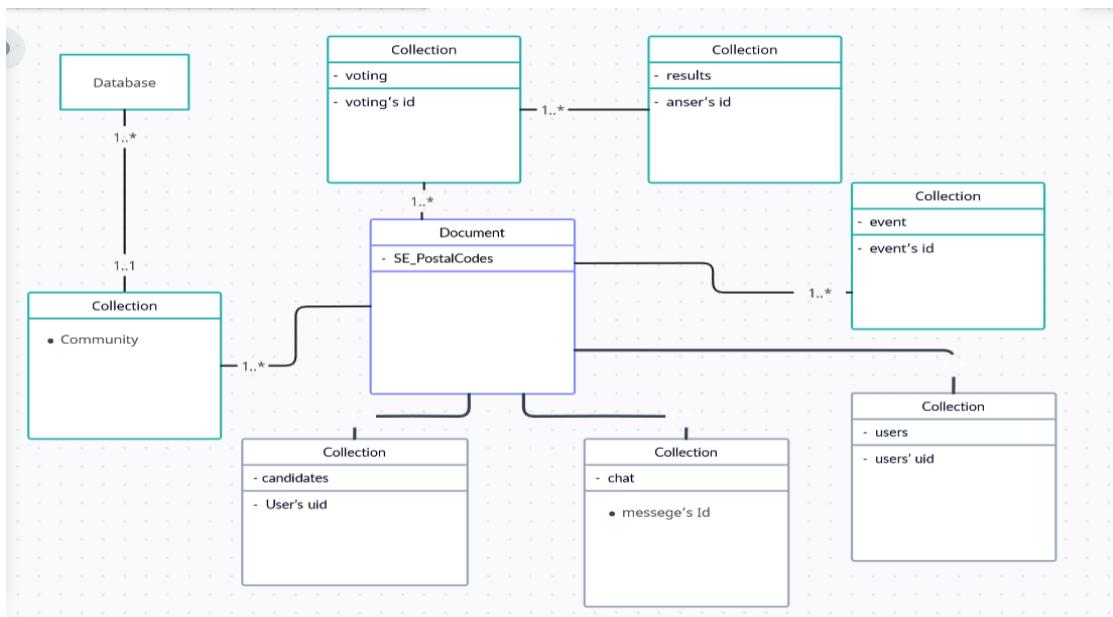
candidates, chat, events, users, votings. Κάθε ένα από αυτά δημιουργείται ή ανανεώνεται από κάποια ενέργεια του χρήστη.

Subcollection “users”: Μόλις ο χρήστης συμπληρώσει τη φόρμα της εγγραφής ή εισόδου με τα προσωπικά του στοιχεία, αποθηκεύονται στο subcollection “users” του αντίστοιχου TK document στο οποίο βρίσκεται.

Subcollection “candidates”: Όταν ο χρήστης δηλώνει υποψηφιότητα, αποθηκεύεται στο candidates και δημιουργεί document που αντιστοιχεί στο id του.

Subcollection “chat”: Όταν κάποιος χρήστης στείλει ένα μήνυμα στο discussion του app, δημιουργείται το chat στο οποίο δημιουργούνται τα πεδία που αναφέρουν πληροφορίες για το μήνυμα που έστειλε ο χρήστης.

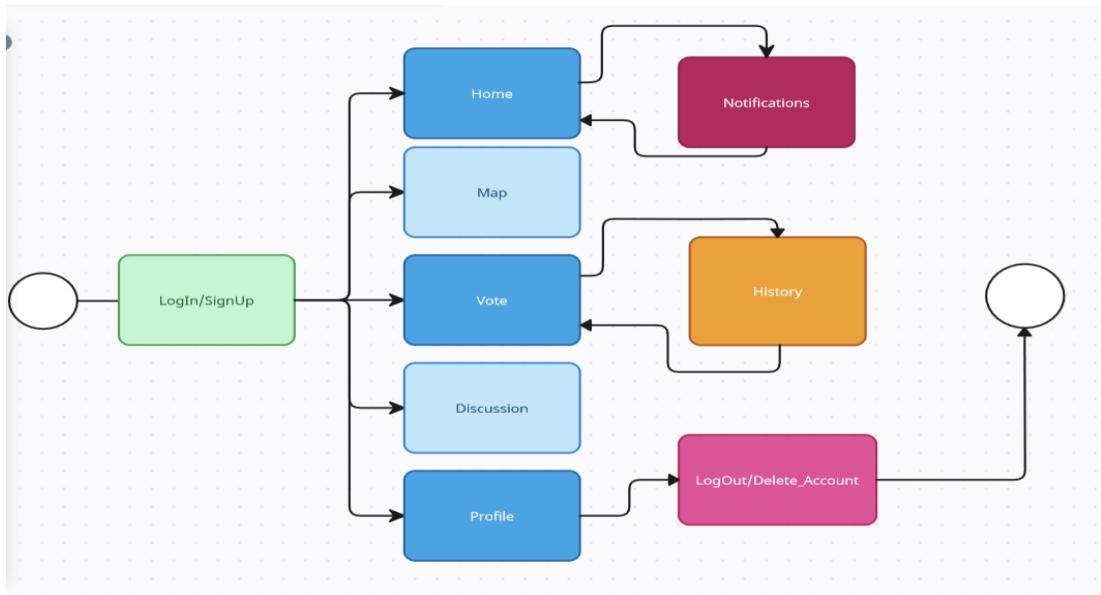
Subcollection “events” και “votings”: δημιουργούνται όταν ο admin δημιουργήσει κάτι απ’ τα δύο και το votings επεξεργάζεται από τις επιλογές των χρηστών.



Εικόνα 12- UML διάγραμμα βάσης δεδομένων

Subcollection results. Ανήκει στο «votings» και αποθηκεύει τις ψήφους σε κάθε document το οποίο αποτελεί την υποψήφια άποψη.

5.3 Σχεδιασμός Συστήματος Κοινότητας



Εικόνα 13- UML συστήματος κοινότητας

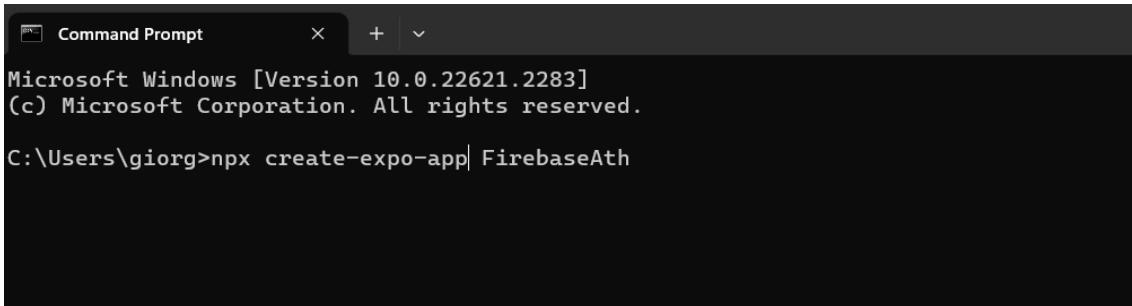
Σε αυτό το διάγραμμα παρουσιάζουμε τον σχεδιασμό του συστήματος. Αποδίδονται οπτικά με την αλληλουχία με την οποία ο χρήστης μπορεί να έχει πρόσβαση σε αυτές και παράλληλα το διάγραμμα αναδεικνύει την πλοήγηση του χρήστη. Το βασικό αρχείο App.js είναι εκείνο το οποίο αποτελεί τον κορμό όλων αυτών και τα συνδέει μεταξύ τους. Όταν ο χρήστης ανοίξει στην εφαρμογή, θα έρθει σε επαφή με την σελίδα εισόδου (πρώτο κουτί). Από εκεί και ύστερα αφού ολοκληρώσει τις απαιτήσεις θα εισέλθει στην κοινότητα η οποία του δίνει παράλληλες λειτουργίες κατά τη χρήση των βασικών 5 λειτουργιών Home, Map, Vote, Discussion, Profile. Στις υπόλοιπες λειτουργίες/οθόνες μπορεί να εισέλθει μέσω κάποιας από τις 5 βασικές λειτουργίες που προαναφέραμε. Ο χρήστης μόνο από την λειτουργία Home μπορεί να έχει πρόσβαση στην λειτουργία Notifications και από εκεί μπορεί μόνο να επιστρέψει πίσω στο Home. Στην Home λειτουργία παρακολουθεί τα events και τις τελευταίες αποφάσεις της κοινότητας. Ως διαχειριστής μπορεί μέσω του Home να δημιουργήσει events ή votings. Από την λειτουργία votes μπορεί να έχει πρόσβαση στη λειτουργία History όπου βλέπει το ιστορικό των αποφάσεων της κοινότητας και από εκεί μπορεί να πλοηγηθεί μόνο από εκεί όπου εισήλθε. Έχει την δυνατότητα να βάλει υποψηφιότητα και να ψηφίσει εκπρόσωπο. Επίσης, μπορεί επιλέγοντας την αντίστοιχη ψηφοφορία, να μεταφερθεί στις ψηφοφορίες και να ψηφίσει για την απόφαση της κοινότητας. Τις λειτουργίες Log Out και Delete Account μπορεί να τις χρησιμοποιήσει μόνο από την λειτουργία/οθόνη Profile.

6. ΥΛΟΠΟΙΗΣΗ ΣΥΤΗΜΑΤΟΣ

Σε αυτό το κεφάλαιο θα αναλύσουμε το πως υλοποιήσαμε την εφαρμογή και τις λειτουργίες που παρέχει στους χρήστες Στη συνέχεια θα δούμε πως συνδέεται η βάση δεδομένων με όλα τα υπόλοιπα συστήματα.

6.1 Δημιουργία Expo project/app

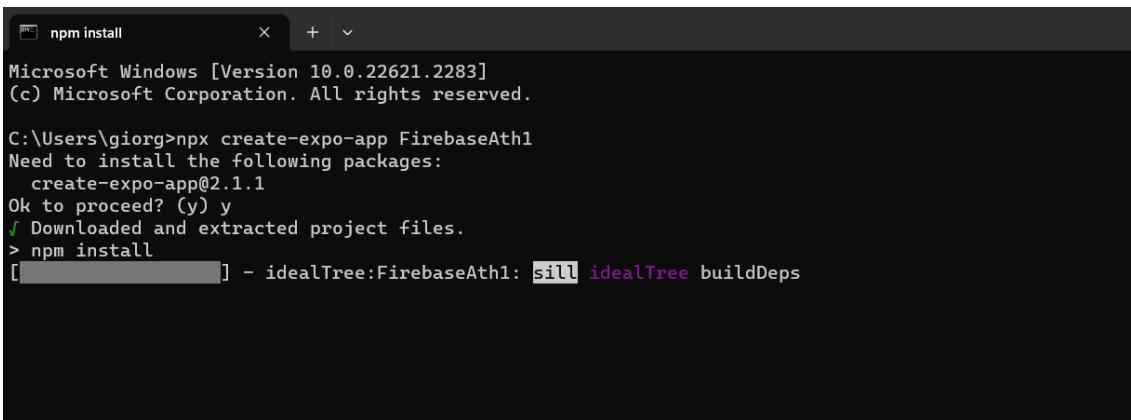
Ξεκινώντας την υλοποίηση της εφαρμογής με τα εργαλεία που αναφέραμε στα προηγούμενα κεφάλαια, θα χρειαστεί να δημιουργήσουμε ένα expo app μέσω του Command Prompt.



```
Command Prompt      x + ▾
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\giorg>npx create-expo-app| FirebaseAth
```

Εικόνα 14 - Εντολή για την δημιουργία expo project



```
npm install      x + ▾
Microsoft Windows [Version 10.0.22621.2283]
(c) Microsoft Corporation. All rights reserved.

C:\Users\giorg>npx create-expo-app FirebaseAth1
Need to install the following packages:
  create-expo-app@2.1.1
Ok to proceed? (y) y
  Downloaded and extracted project files.
> npm install
[██████████] - idealTree:FirebaseAth1: sill idealTree buildDeps
```

Εικόνα 15 - Εγκατάσταση των απαραίτητων αρχείων

Στη συνέχεια κάνουμε εγκατάσταση μέσω του Command Prompt στον φάκελο που δημιουργήσαμε όλες τις απαραίτητες βιβλιοθήκες που σκοπεύουμε να χρησιμοποιήσουμε στην εφαρμογή. Χρησιμοποιούμε από το node.js την εντολή

npm install και δίπλα την αντίστοιχη βιβλιοθήκη που θέλουμε να εγκαταστήσουμε

:

@react-native-community/datetimepicker

@react-native-firebase/app

@react-navigation/bottom-tabs

@react-navigation/native

@react-navigation/native-stack

@react-navigation/stack

@react-native-community

@react-native-picker/picker

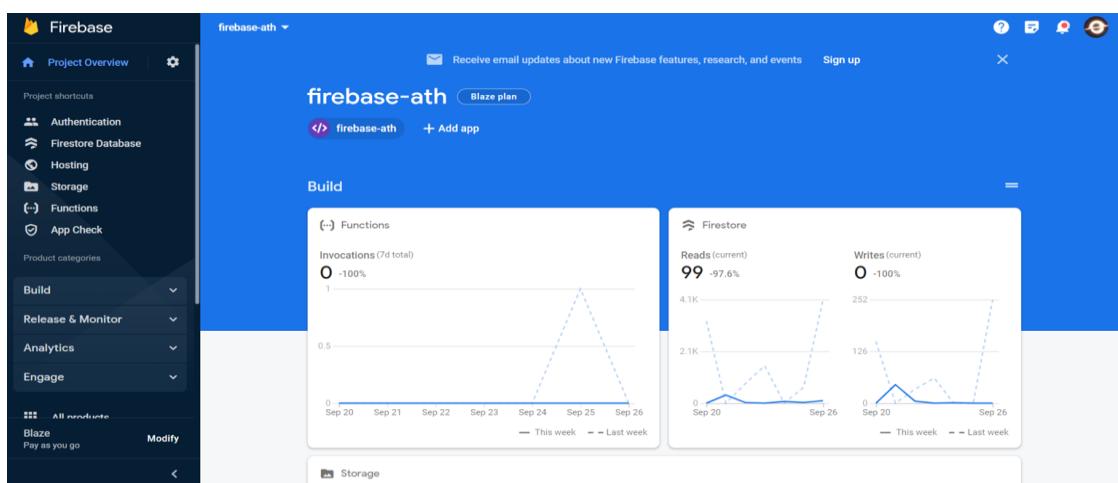
@react-native-gesture-handler

@react-native-screens

Expo-location

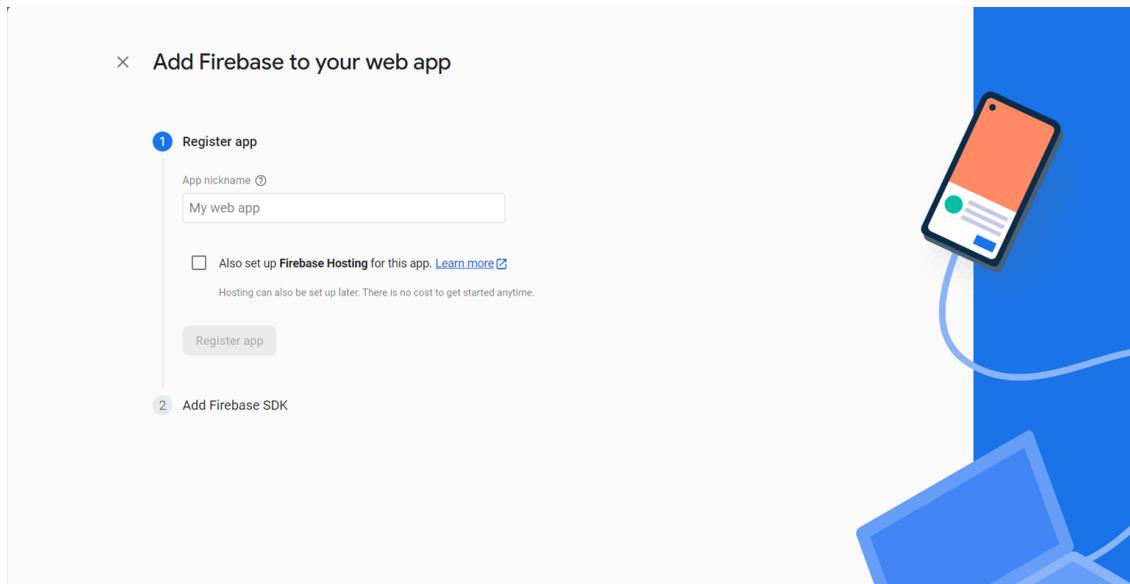
Συνεπώς έχουμε δημιουργήσει το project με όλα τα απαραίτητα εργαλεία και μένει να κάνουμε την σύνδεση με την βάση.

6.2 Σύνδεση του Expo Project με την Firebase



Εικόνα 16 - firebase-ath overview

Για το στήσιμο των τελευταίων εργαλείων ώστε να ξεκινήσουμε με άνεση να γράφουμε τον κώδικα της εφαρμογής, πρέπει να κάνουμε μία επιτυχημένη σύνδεση με την βάση δεδομένων, στην προκείμενη η Firebase. Όπως βλέπουμε στην παρακάτω εικόνα, δημιουργούμε ένα firebase-project. Δημιουργούμε web application μιας και θέλουμε να κάνουμε χρήση και για android και για ios λειτουργικά.



Εικόνα 17 - δημιουργία προηποθέσεων σύνδεσης για διαδικτυακές εφαρμογές

Για να κάνουμε την σύνδεση του expo-project με το firebase-project πρέπει να δημιουργήσουμε ένα αρχείο “firebase.js” μέσα στο expo-project μας και να περάσουμε κάποιες τιμές και κάποιες μεταβλητές. Βλέπουμε στην παρακάτω

```
JS firebase.js > ...
1 // Import the functions you need from the SDKs you need
2 import { initializeApp } from "firebase/app";
3 import { getAuth } from "firebase/auth";
4 import { getFirestore } from "firebase/firestore";
5 // TODO: Add SDKs for Firebase products that you want to use
6 // https://firebase.google.com/docs/web/setup#available-libraries
7
8 // Your web app's Firebase configuration
9 // For Firebase JS SDK v7.20.0 and later, measurementId is optional
10 const firebaseConfig = {
11   apiKey: "AIzaSyD-jrw1E9HJ3RFQKEwHhw04qLvaI-cafOM",
12   authDomain: "fir-ath-60feb.firebaseioapp.com",
13   projectId: "fir-ath-60feb",
14   storageBucket: "fir-ath-60feb.appspot.com",
15   messagingSenderId: "843833529594",
16   appId: "1:843833529594:web:b4d4476749d47a2627e543",
17   measurementId: "G-R9GK9XC4R3"
18 };
19
20 initializeApp(firebaseConfig);
21 export const auth = getAuth();
22 export const database = getFirestore();
```

Εικόνα 18- το αρχείο firebase.js

εικόνα τον κώδικα που συμπεριλαμβάνει το αρχείο firebase.js ώστε να κάνει σωστή και επιτυχημένη επικοινωνία με την βάση

6.3 Δημιουργία των Απαραίτητων Components

Ξεκινώντας την ανάπτυξη κώδικα για τις λειτουργίες τις εφαρμογής, πρέπει να φτιάξουμε όλα τα απαραίτητα components που θα χρειαστούμε. Το αρχείο που είδαμε πιο πάνω αποτελεί κι αυτό ένα component.

Τοποθεσία

To component «Location» βρίσκει την τοποθεσία του χρήστη μέσω latitude και longitude και μας επιστρέφει το iso country code και τον ταχυδρομικό κώδικα σε μία μεταβλητή (displayCurrentAddress), το όνομα (name), την πόλη (city), και την χώρα (country). Σε αυτό το αρχείο είναι δηλωμένη η βιβλιοθήκη exro-location. Τσεκάρει αν ο χρήστης έχει ανοιχτή την τοποθεσία και δίνει στον χρήστη την επιλογή να την ενεργοποίηση μέσω της συνάρτησης CheckIfLocationEnabled() η οποία καλεί την hasServicesEnabledAsync(). Έπειτα, δημιουργήσαμε την GetCurrentLocation() η οποία χρησιμοποιεί το API reverseGeocodeAsync. Παρακάτω βλέπουμε στις εικόνες 20-22 τον κώδικα που εμπεριέχει το αρχείο Location.js.

```
js location.js > ⚡ LocationComponent > ⓘ GetCurrentLocation
1 import * as location from 'expo-location';
2 import React, { useState, createContext, useContext, useEffect } from 'react';
3
4 export default function LocationComponent() {
5   const [locationServiceEnabled, setLocationServiceEnabled] = useState(false);
6
7   const [city, setCity] = useState('Wait, we are fetching your location...');
8   const [country, setCountry] = useState('Wait, we are fetching your location...');
9   const [name, setName] = useState('Wait, we are fetching your location...');
10  const [street, setStreet] = useState('Wait, we are fetching your location...');
11  const [latitude, setLatitude] = useState('');
12  const [longitude, setLongitude] = useState('');
13
14  let city1;
15  let country1;
16  let name1;
17  let street1;
18
19  const [displayCurrentAddress, setDisplayCurrentAddress] = useState(
20    'Wait, we are fetching your location...'
21  );
22  const [locationData, setLocationData] = useState('');
23  let data;
24  let address;
25  useEffect(() => {
26    CheckIfLocationEnabled();
27    GetCurrentLocation();
28  }, []);
29
30  const CheckIfLocationEnabled = async () => {
31    let enabled = await location.hasServicesEnabledAsync();
32  }
```

```
32
33  if (!enabled) {
34    Alert.alert(
35      'Location Service not enabled',
36      'Please enable your location services to continue',
37      [{ text: 'OK' }],
38      { cancelable: false }
39    );
40  } else {
41    setLocationServiceEnabled(enabled);
42  }
43
44  const GetCurrentLocation = async () => {
45    let { status } = await location.requestForegroundPermissionsAsync();
46
47    if (status === 'granted') {
48      Alert.alert(
49        'Permission not granted',
50        'Allow the app to use location service.',
51        [{ text: 'OK' }],
52        { cancelable: false }
53      );
54    }
55
56    let { coords } = await location.getCurrentPositionAsync();
57
58    if (coords) {
59      const { latitude, longitude } = coords;
60      let response = await location.reverseGeocodeAsync([
61        latitude,
62        longitude
63      ]);
64  }
```

Εικόνα 19- imports και μεταβλητές

Εικόνα 20- καλεί το API και λαμβάνει την τοποθεσία

```

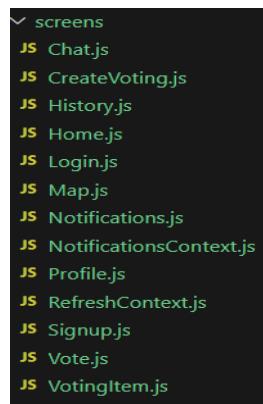
64      setLatitude(latitude);
65      setLongitude(longitude)
66
67      for (let item of response) {
68          address = `${item.isoCountryCode}_${item.postalCode}`;
69          data = `${item.city}, ${item.country}, ${item.name}, ${item.postalCode}`;
70          setLocationData(data);
71          setDisplayCurrentAddress(address);
72          city1 = `${item.city}`;
73          setCity(city1);
74          country1 = `${item.country}`;
75          setCountry(country1);
76          name1 = `${item.district}`;
77          setName(name1);
78          street1 = `${item.street}`;
79          setStreet(street1);
80
81      }
82  }
83 ];
84
85 return [ displayCurrentAddress, street, name, city, country, Latitude, Longitude ];
86
87 };
88

```

Εικόνα 21- το location.js αρχείο επιστρέφει τις ανάλογες μεταβλητές

Το επόμενο βήμα είναι η δημιουργία των screens αρχείων. Μέσα στον αντίστοιχο φάκελο που θα φτιάξουμε θα συμπεριλάβουμε όλες τις οθόνες που θα βλέπει ο χρήστης. Θα αναφέρουμε όλα τα υποψήφια screens components αρχεία:

Login.js, SignUp.js, Home.js, Map.js, Vote.js, Chat.js, Profile.js, Notifications.js, NotificationContext.js, RefresContext.js, CreatingVoting.js, VotingItem.js, History.js



Εικόνα 22- screen components

6.4 Υλοποίηση των Βασικών Λειτουργιών της Εφαρμογής

Προτού αναπτύξουμε το καθένα component ξεχωριστά, πρέπει να αναλύσουμε πως όλα αυτά τα αρχεία επικοινωνούν μεταξύ τους. Γι' αυτό το λόγο η δημιουργία του project μας δίνει το αρχείο App.js χωρίς να χρειαστεί να το δημιουργήσουμε. Μέσα στο αρχείο καλούνται όλα τα components που υπάρχουν μέσα και έξω από τον φάκελο «screens» και βάση της σχέσεις που έχουν με το index αρχείο καλούνται children. Αν κάποιο component δε καλείται από το index αρχείο ή από κάποιο child του App.js τότε δεν συμπεριλαμβάνεται στο run/execute της εφαρμογής. Δηλώνουμε στην αρχή της σελίδας μέσω της εντολής import τις βιβλιοθήκες που θα χρησιμοποιήσουμε μόνο για το συγκεκριμένο αρχείο.

Όπως φαίνεται στην εικόνα 23, καλούμε όλα τα βασικά screens αρχεία που σκοπεύουμε να παρέχουμε στον χρήστη για εγγραφή/είσοδος, και για πλοήγηση. Το αρχείο καλεί το by default component το οποίο έχει το χαρακτηριστικό του root.

```
App.js > App
1 import React, { useState, createContext, useContext, useEffect, useMemo } from 'react';
2 import { NavigationContainer } from '@react-navigation/native';
3 import { createStackNavigator } from '@react-navigation/stack';
4 import { View, ActivityIndicator, StyleSheet } from 'react-native';
5 import { onAuthStateChanged } from 'firebase/auth';
6 import { auth, database } from './firebase';
7 import Login from './screens/Login';
8 import Signup from './screens/Sigup';
9 import Chat from './screens/Chat';
10 import Home from './screens/Home';
11 import Map from './screens/Map';
12 import Vote from './screens/Vote';
13 import Profile from './screens/Profile';
14 import Notifications from './screens/Notifications';
15 import CreateVoting from './screens/CreateVoting';
16 import VotingItem from './screens/VotingItem';
17 import History from './screens/History';
18 import { RefreshProvider } from './screens/RefreshContext';
19 import { NotificationsProvider, useNotifications } from './screens/NotificationsContext';
20 import { createBottomTabNavigator } from '@react-navigation/bottom-tabs';
21 import LocationComponent from './location';
22 import Ionic from 'react-native-vector-icons/Ionicons';
23 import { MaterialCommunityIcons } from '@expo/vector-icons';
24 import {
25   doc,
26   collection,
27   deleteDoc,
28   getDocs,
29   query,
30   where,
31 } from 'firebase/firestore';
```

Εικόνα 23- όλα τα imports του App.js αρχείου

Η app() ξεκινάει παίρνει ως παραμέτρους και εισάγει σε μεταβλητές, τα δεδομένα που επιστρέφει η

```
return (
  <AuthenticatedUserProvider>
    <RootNavigator location={location} name={name} street={street} city={city} country={country} Latitude={Latitude} Longitude={Longitude}>
  </AuthenticatedUserProvider>
```

Εικόνα 24- To component App() επιστρέφει το component «RootNavigator»

LocationComponent (δλδ ταχυδρομικός κωδικας, χωρα, πολη κλπ).

Κάνει τον έλεγχο για την τιμή που έχει η μεταβλητή “location” διότι η συγκεκριμένη

```
export default function App() {
  const [ location, street, name, city, country, Latitude, Longitude ] = LocationComponent();
  console.log("Your current PostalCode is : ", location);

  while(location === 'Wait, we are fetching your location...'){

    return [
      <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
        <Text>Wait, we are fetching your location...</Text>
        <ActivityIndicator size='large' />
      </View>
    ];
  }

  if(location !== 'Wait, we are fetching your location...'){

    if (location == null){
      location = 'unknown place';
      street = 'unknown place';
      name = 'unknown place';
      city = 'unknown place';
      country = 'unknown place';
    }
  }
}
```

Εικόνα 25- To «App» component ελέγχει την μεταβλητή location

Θα χρησιμοποιηθεί στο πλαίσιο προσδιορισμού του κάθε room μέσα στην βάση δεδομένων. Αν η τιμή της είναι η πρωταρχική (by default) τιμή που έχουμε θέσει στο αρχείο location.js τότε θέλουμε να εμφανίζει μια επανάληψη μέχρις ότου λάβουμε μία τιμή πέρα από την προεπιλεχθείσα.

Η συνάρτηση επιστρέφει το RootNavigator και του περνάει σαν ορίσματα τις τιμές που πήρε από το location.js.

```
function RootNavigator({ location, name, street, city, country, Latitude, Longitude }) {
  const { user, setUser } = useContext(AuthenticatedUserContext);
  const [isloading, setIsLoading] = useState(true);
  useEffect(() => {
    const unsubscribeAuth = onAuthStateChanged(
      auth,
      async authenticatedUser => {
        authenticatedUser ? setUser(authenticatedUser) : setUser(null);
        setIsLoading(false);
      }
    );
    return unsubscribeAuth;
  }, [user]);
  if (isloading) {
    return (
      <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
        <ActivityIndicator size='large' />
      </View>
    );
  }
  return (
    <NavigationContainer>
      {user ? <Home1 location={location} name={name} street={street} city={city} country={country} Latitude={Latitude} Longitude={Longitude}></Home1> : <AuthStack>}
    </NavigationContainer>
  );
}
```

Eikόνα 26- To «RootNavigator» component

Όπως φαίνεται στην εικόνα 26, η RootNavigator κάνει σε real time τον έλεγχο για το authentication και επιστρέφει υπό μία συνθήκη τα παρακάτω: Αν έχει γίνει το authentication το καλείται το Home1 component, αλλιώς καλεί την AuthStack. Η τελευταία καλεί τα δύο πρωταρχικά components Login.js και signUp.js που αλληλεπιδρά ο χρήστης που δεν έχει πραγματοποιήσει εγγραφή ή είσοδο στο σύστημα. Παρακάτω θα αναλύσουμε τις βασικές λειτουργίες αλληλεπίδρασης χρήστη με τη βάση δεδομένων για κάθε component χωρίς να συμπεριλάβουμε λεπτομέρειες που αφορούν την εμφάνιση του.

```

function AuthStack({ location, name, street, city, country, Latitude, Longitude }) {
  return (
    <Stack.Navigator screenOptions={{ headerShown: false }}>
      <Stack.Screen name='Login' component={Login} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
      <Stack.Screen name='Signup' component={Signup} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
    </Stack.Navigator>
  );
}

```

Εικόνα 27 - To AuthStack() component

Η Home1 επιστρέφει τα stack.screen component με την βασική να είναι η Back η οποία επιστρέφει όλο το κομμάτι της πλοιόγησης. Καλεί τα components Home, Map, Vote, Chat, Profile με τη μορφή του tabBar.screen (Tab.navigator, Tab.screen).

```

const Stack = createStackNavigator();
const AuthenticatedUserContext = createContext();
const Tab = createBottomTabNavigator();

const AuthenticatedUserProvider = ({ children }) => {
  const [user, setUser] = useState(null);
  return (
    <AuthenticatedUserContext.Provider value={{ user, setUser }}>
      {children}
    </AuthenticatedUserContext.Provider>
  );
};

const Home1 = ({ location, name, street, city, country, Latitude, Longitude }) => {
  return (
    <RefreshProvider>
      <NotificationsProvider>
        <Stack.Navigator defaultScreenOptions={{ Back }}>
          <Stack.Screen name='Back' component={Back} options={{ headerShown: false }} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
          <Stack.Screen name='Notifications' component={Notifications} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
          <Stack.Screen name='CreateVoting' component={CreateVoting} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
          <Stack.Screen name='VotingItem' component={VotingItem} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
          <Stack.Screen name='History' component={History} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
        </Stack.Navigator>
      </NotificationsProvider>
      </RefreshProvider>
    );
};

```

Εικόνα 28- Home1 component

```

<defaultScreenOptions={{ Home }}>
<Tab.Screen name='Home' component={Home} initialParams={{ location, name, street, city, country, Latitude, Longitude }}/>
<Tab.Screen name='Map' component={Map} initialParams={{ location, name, street, Latitude, Longitude }}/>
<Tab.Screen name='Vote' component={Vote} initialParams={{ location }}/>
<Tab.Screen name='Discussion' component={Chat} initialParams={{ location }}/>
<Tab.Screen name='Profile' component={Profile} initialParams={{ location }}/>

```

Εικόνα 29- Η Back() επιστρέφει τις οθόνες

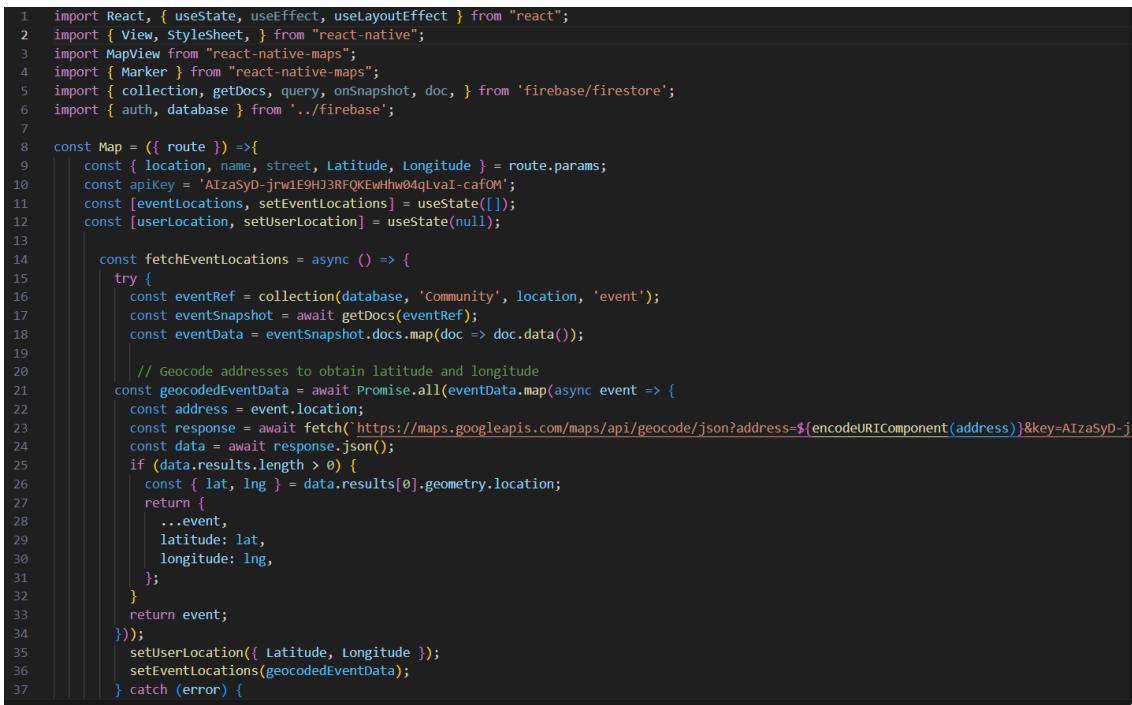
6.4.1 Η Αρχική Σελίδα

Το Component Home αποτελεί την αρχική σελίδα της εφαρμογής, και την πρώτη αλληλεπίδραση του χρήστη με την κοινότητα. Η λειτουργία της εμπεριέχει

κάπτοιες πληροφορίες (Ποιος είναι ο διαχειριστής, το παρόν email και η παρούσα κοινότητα κτλ.) και εμφανίζει τα event που έχουν προγραμματιστεί, και τις αποφάσεις της κοινότητας.

6.4.2 Ο Χάρτης

Το Component Map παρουσιάζει γεωγραφικές πληροφορίες (χάρτης) με την τοποθεσία του χρήστη, τις τοποθεσίες των εκδηλώσεων/δράσεων και τις πληροφορίες τους.



```
1 import React, { useState, useEffect, useLayoutEffect } from "react";
2 import { View, StyleSheet, } from "react-native";
3 import MapView from "react-native-maps";
4 import { Marker } from "react-native-maps";
5 import { collection, getDocs, query, onSnapshot, doc, } from 'firebase/firestore';
6 import { auth, database } from '../firebase';
7
8 const Map = ({ route }) =>{
9   const { location, name, street, Latitude, Longitude } = route.params;
10  const apiKey = 'AIzaSyD-jrwLE9H3RFQKEwHw04qLvaI-cafOM';
11  const [eventLocations, setEventLocations] = useState([]);
12  const [userLocation, setUserLocation] = useState(null);
13
14  const fetchEventLocations = async () => {
15    try {
16      const eventRef = collection(database, 'Community', location, 'event');
17      const eventSnapshot = await getDocs(eventRef);
18      const eventData = eventSnapshot.docs.map(doc => doc.data());
19
20      // Geocode addresses to obtain latitude and longitude
21      const geocodedEventData = await Promise.all(eventData.map(async event => {
22        const address = event.location;
23        const response = await fetch(`https://maps.googleapis.com/maps/api/geocode/json?address=${encodeURIComponent(address)}&key=AIzaSyD-jrwLE9H3RFQKEwHw04qLvaI-cafOM`);
24        const data = await response.json();
25        if (data.results.length > 0) {
26          const { lat, lng } = data.results[0].geometry.location;
27          return {
28            ...event,
29            latitude: lat,
30            longitude: lng,
31          };
32        }
33      }));
34      setEventLocations({ Latitude, Longitude });
35      setEventLocations(geocodedEventData);
36    } catch (error) {
37    }
38  }
39}
```

Εικόνα 30- To «Map» component

Όπως βλέπουμε στην εικόνα 27 καλείται η συνάρτηση «fetchEventLocations» η οποία τραβάει τα δεδομένα που υπάρχουν στο collection «event» απ' το firestore. Χρησιμοποιεί το geolocation Api της google όπως φαίνεται στην εικόνα το οποίο μας επιστρέφει δεδομένα βάση της διεύθυνσης που του έχουμε δώσει «address». Η συνάρτηση με βάση τα δεδομένα που της δίνουμε μας επιστρέφει τα αντίστοιχα longitude και latitude τα οποία τα μαρκάρουμε στον χάρτη.

6.4.3 Οι Ψηφοφορίες

Το Component «Vote» αναφέρεται στη σελίδα όπου ο χρήστης μπορεί να ψηφίσει τον αντιπρόσωπο που επιθυμεί, να υποβάλλει την υποψηφιότητα του και να ψηφίσει για θέματα που αφορούν την κοινότητα. Η συνάρτηση

«getPartSit» ελέγχει αν ο χρήστης έχει υποβάλλει η όχι υποψηφιότητα και μετατρέπει την κατάσταση της υποψηφιότητας της μεταβλητής στο αντίθετό της. Οι συναρτήσεις «handleParticipationToggle» και «handleRomoveToggle» παίρνουν την τιμή που δίνει η «getPartSis» και αλλάζουν τα δεδομένα στη βάση για τον χρήστη. Καλούνται από τον χρήστη όταν επιλέξει να υποβάλλει υποψηφιότητα ή όταν αφαιρεί την υποψηφιότητά του πατώντας το αντίστοιχο κουμπί (button).

```

94  const getPartsit = async () => {
95    const usersPart = query(collection(database, 'Community', location, 'users'), where('user', '==', auth.currentUser?.email));
96    await getDocs(usersPart).then((querySnapshot) => {
97      // Loop through the query results
98      querySnapshot.forEach((doc) => {
99
100        let data = doc.data();
101        partsit = data.participateInElection;
102        exeiPsifisei = data.hasVote;
103        poionpsifises = data.votedCandidateKey
104      });
105    }).catch(error) => {
106      // Log the error
107      console.error(error);
108    };
109    setParticipating(partsit);
110
111    if (partsit == false){
112      setParticipating(false);
113      setHasVote(exeiPsifisei);
114      console.log(exeiPsifisei);
115
116    } else {
117      setParticipating(true);
118      setHasVote(exeiPsifisei);
119      console.log(exeiPsifisei);
120    }
121
122
123    if (exeiPsifisei == false){
124      setHasVote(false);
125    } else {
126      setHasVote(true);
127      setVotedCandidateKey(poionpsifises);
128    }
129  };
130

```

Εικόνα 31- Η συνάρτηση «getPartSis»

```
154 const handleParticipationToggle = async () => {
155
156   Alert.alert(
157     'Confirm declaration candidate',
158     'Are you sure you want to join admin's election?',
159     [
160       {
161         text: 'Not at moment',
162         style: 'destructive',
163       },
164       {
165         text: 'Apply',
166         style: 'cancel',
167         onPress: async () => {
168           try {
169             if (participating === false) {
170               let userId;
171               let description;
172               const updateDocument = query(collection(database, "community", location, "users"), where("user", "==", auth.currentUser?.email));
173               const querySnapshot = await getDocs(updateUserSnapshot);
174               querySnapshot.forEach(doc) => {
175                 userId = doc.id;
176                 description = doc.data().description;
177               };
178               // Update the user's participation status in Firestore
179               await updateDoc(doc(database, "community", location, "users", userId), {
180                 "participateInElection": true,
181               });
182               await setDoc(doc(database, "community", location, "candidates", userId), {
183                 choice: auth.currentUser?.email,
184                 votes: 0,
185               });
186             }
187           }
188         
```

Εικόνα 32- η συνάρτηση «handleParticipationToggle»

```
220 const handleRemoveCandidate = async () => {
221
222   Alert.alert(
223     'Remove your candidacy',
224     'Are you sure you want to remove your candidacy?',
225     [
226       {
227         text: 'Don't remove',
228         style: 'cancel',
229       },
230       {
231         text: 'Remove',
232         style: 'destructive',
233         onPress: async () => {
234           try {
235             if (participating === true) {
236
237               let userRef;
238               const updatedUserRef = query(collection(database, 'Community', location, 'users'), where('user', '==', auth.currentUser?.email));
239               const querySnapshot2 = await getDocs(updatedUserRef);
240               querySnapshot2.forEach(doc) => {
241                 userRef = doc.id;
242               }
243             }
244             // Update the user's participation status in firenews
245             updateDoc(doc(database, 'Community', location, 'users', userRef), {
246               'participateInElection': false,
247               'isAdmin': false,
248             });
249
250             deleteDoc(doc(database, 'Community', location, 'candidates', userRef), {
251               creator: auth.currentUser?.email,
252               votes: 0,
253             });
254           }
255         }
256       ]
257     )
258   );
259 }
260
261 export default App;
```

Εικόνα 33- Η «handleRemoveToggle» συνάρτηση

Ο χρήστης συνεπώς για να ψηφίσει έναν άλλο χρήστη ή τον εαυτό του για εκπρόσωπο της κοινότητας θα καλέσει πατώντας το αντίστοιχο κουμπί την συνάρτηση «votes». Η «votes» λαμβάνοντας ως παράμετρο τη αντίστοιχο id που παίρνει από το view που αντιστοιχεί στον χρήστη που πάτησε να ψηφίσει, ελέγχει αν ο χρήστης έχει ξαναψηφίσει και σε περίπτωσή που δεν έχει τότε τρέχει το κομμάτι του κώδικα το οποίο:

A. Θέτει την μεταβλητή «hasVote» στην βάση δεδομένων για τον αντίστοιχο χρήστη ως true και ορίζει την μεταβλητή «votedCandidateKey» ίση με το πρωτέυον κλειδί (id) του χρήστη που ψήφισε ο εν λειτουργία χρήστης.

Β. Προσθέτει στον πίνακα του υποψηφίου στην βάση δεδομένων, το id του ψηφοφόρου, καθώς και την ημερομηνία του γεγονότος.

```
339
340     const votes = async (candidateId) => {
341
342         try {
343             if (hasVote === false){
344
345                 setHasVote(true);
346
347                 let userId;
348                 const updateUserVote = query(collection(database, "Community", location, 'users'), where('user', '==', auth.currentUser?.email));
349                 const querySnapshot2 = await getDocs(updateUserVote);
350                 querySnapshot2.forEach((doc) => {
351
352                     userId = doc.id;
353                 });
354                 // Update the user's participation status in Firestore
355                 await updateDoc(doc(database, 'Community', location, 'users', userId), {
356                     "hasVote": true,
357                     "votedCandidateKey": candidateId,
358                 });
359
360                 const candidateRef = doc(database, 'Community', location, 'candidates', candidateId);
361                 const candidateDoc = await getDoc(candidateRef);
362                 const currentVoteCount = candidateDoc.data().votes || 0;
363                 const allVotes = candidateDoc.data().usersVotesHim || [];
364
365                 await updateDoc(candidateRef, {
366                     "votes": currentVoteCount + 1,
367                     "usersVotesHim": {
368                         ...allVotes,
369                         [auth.currentUser?.uid]: {
370                             time: serverTimestamp(),
371                         },
372                     },
373                     // Increment the candidate's vote count by 1
374                 });
375             }
376         }
377     }
378 }
```

Εικόνα 34- Η «votes» συνάρτηση

Ο χρήστης όταν επιλέξει να πάρει πίσω την ψήφο του, καλεί την συνάρτηση «unVote», η οποία κάνει τα αντίστροφα πράγματα από την «votes». Ελέγχει αν ο χρήστης έχει ψηφίσει και αν είναι αληθές τότε:

- A. Θέτει την μεταβλητή «hasVoted» = false και το «votedCandidateKey» ώς null.

B. Διαγράφει απ' τον πίνακα του υποψηφίου το id του εν λειτουργία χρήστη.

6.4.4 Η Συζήτηση

Το Component «Chat» αναφέρεται στην συζήτηση(discussion) της κοινότητας. Καλεί δύο βασικές συναρτήσεις. Η μία συνάρτηση έχει να κάνει με την real time αποθήκευση των μηνυμάτων που έχουν σταλεί στην κοινότητα σε μια μεταβλητή-πίνακα, και η άλλη καλείται όταν ο χρήστης στέλνει ένα μήνυμα. Η λειτουργία της δεύτερης, χρησιμοποιεί την «previousMessage» ώστε στο πλαίσιο της

επεξεργασίας του πίνακα που αποθηκεύει όλα τα μηνύματα της κοινότητας, να προσθέτει το μήνυμα και να μην αλλάζει όλη την υπόλοιπη διάταξη των μηνυμάτων.

```

41
42
43
44
45
46
47
48
49  useLayoutEffect(() => {
50
51    const collectionRef = collection(database, 'Community', location, 'chat');
52    const q = query(collectionRef, orderBy('createdAt', 'desc'));
53
54    const unsubscribe = onSnapshot(q, querySnapshot => {
55      console.log('querySnapshot unsubscribe');
56      setMessages(
57        querySnapshot.docs.map(doc => ({
58          _id: doc.data()._id,
59          createdAt: doc.data().createdAt.toDate(),
60          text: doc.data().text,
61          user: doc.data().user
62        }))
63      );
64    });
65    return unsubscribe;
66  }, []);
67
68  const onSend = useCallback((messages = []) => {
69    setMessages(previousMessages =>
70      GiftedChat.append(previousMessages, messages)
71    );
72    // setMessages([...messages, ...messages]);
73    const { _id, createdAt, text, user } = messages[0];
74    addDoc(collection(database, 'Community', location, 'chat'), {
75      _id,
76      createdAt,
77      text,
78      user
79    });
80  }, []);
81

```

Εικόνα 35- Η «onSend» συνάρτηση

6.4.5 Το Προφίλ του Χρήστη

Το Component «Profile» δίνει την δυνατότητα στον χρήστη να κάνει έξοδο (Log out), να δημιουργήσει/επεξεργαστεί την περιγραφή για τον εαυτό του που βλέπουν οι άλλοι χρήστες μέσω της υποψηφιότητάς του, και να διαγράψει τον λογαριασμό του. Όταν ο χρήστης συμπληρώσει κάτι στο field text και πατήσει το κουμπί «Save» καλεί την συνάρτηση «handleSave» η οποία κάνει update το πεδίο που αντιστοιχεί στην περιγραφή. Για την διαδικασία της εξόδου απ την κοινότητα καλεί την «signOut» η οποία χρησιμοποιεί την βιβλιοθήκη που μας δίνει έτοιμη η λειτουργία του firebase Authentication και πραγματοποιεί αυτόματα την διαδικασία του Log Out. Τέλος, όταν επιθυμεί να διαγράψει τον λογαριασμό του, πατώντας το αντίστοιχο κουμπί, καλεί την συνάρτηση «handleDeleteAccount» η οποία αφαιρεί τις εγγραφές που μπορεί να χε κάνει ο χρήστης στο κομμάτι της υποψηφιότητας και τον διαγράφει πλήρως απ' το firebase Authentication.

```
57
58     const handleSave = async () => [
59         // save the input text to Firestore
60         let userId;
61         const docRef = query(collection(database, "community", location, 'users'), where('user', '==', auth.currentUser?.email));
62         const docSnap = await getDocs(docRef);
63
64         docSnap.forEach((doc) => {
65             userId = doc.id;
66         });
67     try{
68         updateDoc(doc(database, 'Community', location, 'users', userId), {
69             "description": profileText }).then(() =>{
70             console.log("on prof update profile ok!!");
71         }).catch((err) => Alert.alert("update description error", err.message));
72
73         updateDoc(doc(database, 'Community', location, 'candidates', userId), {
74             "description": profileText }).then(() =>{
75             console.log("on prof, update desc candidate ok!!");
76         }).catch((err) => console.log("doesn't participate", err.message));
77
78         setSavedText(profileText);
79         setIsEditing(false);
80     } catch (error) {
81         console.error('Error saving profile text:', error);
82     }
83 }
```

Εικόνα 36- Η «handleSave» συνάρτηση

```
85
86 const handleDeleteAccount = async () => {
87   // show a confirmation alert to the user
88   let userId;
89   const docRef = query(collection(database, "Community", location, "users"), where("user", "==", auth.currentUser?.email));
90   const docSnap = await getDoc(docRef);
91   docSnap.forEach((doc) => {
92     userId = doc.id;
93   });
94
95   Alert.alert(
96     "Confirm Account Deletion",
97     "Are you sure you want to delete your account? This action cannot be undone.",
98     [
99       {
100         text: "Cancel",
101         style: "cancel",
102       },
103       {
104         text: "Delete",
105         style: "destructive",
106         onPress: () => {
107           const user = auth.currentUser;
108           if (user) {
109             user
110               .delete()
111               .then(() => {
112                 deletionRef(doc(database, "Community", location, "users", userId)).then(() => {
113                   console.log("User prof deleted profile null!");
114                 }).catch((err) => Alert.alert("Delete profile error!", err.message));
115               }
116               deletionRef(doc(database, "Community", location, "candidates", userId)).then(() => {
117                 console.log("User prof deleted profile null!");
118               }).catch((err) => Alert.alert("Delete candidate", err.message));
119             console.log("User account deleted successfully!");
120           }
121         }
122       ]
123     )
124   );
125 }
```

Εικόνα 37- Η «handleDeleteAccount» συνάρτηση

6.4.6 Δημιουργία Ψηφοφορίας ή Εκδήλωσης

To Component «CreateVotings» καλείται όταν ο διαχειριστής της κοινότητας πατήσει το κουμπί στο component Home για να δημιουργήσει μία ψηφοφορία ή ένα event. Ο χρήστης έχει την επιλογή να διαλέξει τι απ' τα 2 θέλει να δημιουργήσει και αναλόγως τι επέλεξε απ' τα 2, θα του εμφανιστούν και τα αντίστοιχα πεδία προς επεξεργασία ή προς δημιουργία.

```

1 import React, { useState, useEffect } from 'react';
2 import { Alert, ScrollView, View, Text, TextInput, Button, StyleSheet, SafeAreaView, TouchableWithoutFeedback, Keyboard } from 'react-native';
3 import { Picker } from '@react-native-picker/picker';
4 import DatePicker from '@react-native-community/datetimepicker';
5 import TimePicker from '@react-native-community/datetimepicker';
6 import MapView, { Marker } from 'react-native-maps';
7 import { useNavigation } from "@react-navigation/native";
8 import { useNotifications, addNotification } from './NotificationsContext';
9 import { useRefreshContext } from './RefreshContext';
10 import { auth, database } from '../firebase';
11 import {
12   doc,
13   collection,
14   addDoc,
15 } from 'firebase/firestore';
16
17
18 const CreatingVoting = ({ route }) => {
19   const { location, name, street, city, country, Latitude, Longitude } = route.params;
20   const navigation = useNavigation();
21   const [selectedType, setSelectedType] = useState('event'); // Default value is 'event'
22   const [title, setTitle] = useState('');
23   const [description, setDescription] = useState('');
24   const [choices, setChoices] = useState(['', '']);
25   const [showDatePicker, setShowDatePicker] = useState(false);
26   const [showTimePicker, setShowTimePicker] = useState(false);
27   const [selectedDate, setSelectedDate] = useState(new Date());
28   const [selectedTime, setSelectedTime] = useState(new Date());
29   const [selectedLocation, setSelectedLocation] = useState(null);
30   const [address, setAddress] = useState('');
31   const { addNotification } = useNotifications();
32   const { triggerRefresh } = useRefreshContext();
33
34

```

Εικόνα 38 - To «createVoting» component μέρος 1

```

36
37   useEffect(() => { //δοκυμη
38     if (selectedLocation) {
39       fetchAddress(selectedLocation);
40     }
41   }, [selectedLocation]);
42
43   const fetchAddress = async (coordinate) => {
44     const apiKey = 'AIzaSyD-jrw1E9HJ3RFQKEwHlw04qLvaI-caFOM';
45     const url = `https://maps.googleapis.com/maps/api/geocode/json?latlng=${coordinate.latitude},${coordinate.longitude}&key=${apiKey}`;
46
47     try {
48       const response = await fetch(url);
49       const data = await response.json();
50       if (data.results && data.results.length > 0) {
51         setAddress(data.results[0].formatted_address);
52       }
53     } catch (error) {
54       console.error('Error fetching address:', error);
55     }
56   };
57
58   const handleShowDatePicker = () => {
59     setShowDatePicker(true);
60   };
61
62   const handleShowTimePicker = () => {
63     setShowTimePicker(true);
64   };
65
66   const handleChoiceChange = (index, choice) => {
67     const updatedChoices = [...choices];
68     updatedChoices[index] = choice;
69     setChoices(updatedChoices);
70   };
71

```

Εικόνα 39– To «createVoting» component μέρος 2

```

72 const addChoice = () => {
73   setChoices([...choices, '']);
74 };
75
76 const removeChoice = (index) => {
77   const updatedChoices = choices.filter((_, i) => i !== index);
78   setChoices(updatedChoices);
79 };
80
81
82 const handleDateChange = (event, selected) => {
83   const currentDate = selected || selectedDate;
84   setShowDatePicker(false);
85   setSelectedDate(currentDate);
86 };
87
88 const handleTimeChange = (event, selected) => {
89   const currentTime = selected || selectedTime;
90   setShowTimePicker(false);
91   setSelectedTime(currentTime);
92 };
93
94 const handleCreate = async () => {
95   try {
96     if (selectedType === 'event') {
97       // Save data to the 'events' collection
98       const eventDocRef = await addDoc(collection(database, 'Community', location, 'event'), {
99
100       type: selectedType,
101       title,
102       description,
103       location: address,
104       date: selectedDate.toISOString().split('T')[0],
105       time: selectedTime.toTimeString().split(' ')[0],
106
107     });

```

Εικόνα 40– To «createVoting» component μέρος 3

```

108   console.log('Event created with ID: ', eventDocRef.id);
109   Alert.alert("You plan " + title);
110   addNotification('A new Event with title "' + title + '" has been created');
111   triggerRefresh();
112 } else if (selectedType === 'voting') {
113   // Save data to the 'votings' collection
114   const votingDocRef = await addDoc(collection(database, 'Community', location, 'voting'), {
115     type: selectedType,
116     title,
117     description,
118     choices,
119     deadlineDate: selectedDate.toISOString().split('T')[0],
120     deadlineTime: selectedTime.toTimeString().split(' ')[0],
121
122   });
123   console.log('Voting created with ID: ', votingDocRef.id);
124   Alert.alert("You plan " + title);
125 }
126
127 // Add any success notification or navigation logic here
128
129 } catch (error) {
130   console.error('Error creating:', error);
131 }
132 navigation.navigate('Back')
133 };

```

Εικόνα 41– To «createVoting» component μέρος 4

6.4.7 Αποφάσεις Κοινότητας

To Component «votingItmes» αναφέρεται στις ψηφοφορίες που έχει δημιουργήσει ο εκπρόσωπος της κοινότητας. Ο χρήστης μπαίνει στην σελίδα της εκάστοτε ψηφοφορίας επιλέγοντας τη από το component «Vote». Ο Χρήστης μπορεί να επιλέξει μία από τις προτεινόμενες απαντήσεις και πατώντας το κουμπί για την υποβολή, καλεί την συνάρτηση «handleSubmitVote» η οποία

δημιουργεί ένα καινούργιο subcollection εν ονόματι results μέσα στο subcollection της αντίστοιχης ψηφοφορίας. Όταν ο χρήστης έχει ψηφίσει ήδη την συγκεκριμένη ψηφοφορία ή όταν η προθεσμία υποβολής (deadline) έχει περάσει τότε η επιλογή για υποβολή μπλοκάρεται.

```
125
126     useEffect(() => {
127
128         const votingRef = doc(database, 'Community', location, 'voting', votingIndex);
129
130         // Use onSnapshot to listen for real-time updates to the deadline data
131         const unsubscribe = onSnapshot(votingRef, (snapshot) => {
132             if (snapshot.exists()) {
133                 const votingData2 = snapshot.data();
134
135
136                 const currentTimestamp = new Date();
137                 currentTimestamp.setDate(currentTimestamp.getDate());
138                 console.log(currentTimestamp);
139                 const currentDate = currentTimestamp.toISOString().split('T')[0];
140                 const currentTime = currentTimestamp.toTimeString().split(' ')[0];
141                 const deadlineDate = votingData2.deadlineDate;
142                 const deadlineTime = votingData2.deadlineTime;
143                 console.log("Date current : ", currentDate);
144                 console.log("Time current : ", currentTime);
145
146                 if (currentDate === deadlineDate && currentTime >= deadlineTime) {
147
148                     setDisability(true);
149                     setDeadExceed(true);
150                 } else if (currentDate > deadlineDate) {
151
152                     setDisability(true);
153                     setDeadExceed(true);
154                 } else {
155
156                     // Deadline has not passed
157
158                     setDeadExceed(false);
159                 }
160             }
161         });
162     });
163
164     return (
165         <div>
166             <h3>{votingData2.votingTitle}</h3>
167             <p>Deadline : {votingData2.deadlineDate} {votingData2.deadlineTime}</p>
168             <p>Status : {votingData2.status}</p>
169             <p>Disability : {votingData2.disability}</p>
170             <p>DeadExceed : {votingData2.deadExceed}</p>
171         </div>
172     );
173 }
```

Εικόνα 42- Real time vote μέρος 1

```
158         if (hasVotedForEvent === true) {
159             setDisability(true);
160         } else {
161             setDisability(false);
162         }
163     );
164 });
165 return unsubscribe;
166
167 },
168 ],
169 );
170
171 useEffect(() => {
172
173     fetchVotingData();
174     fetchVoteVoting();
175     fetchResultData();
176
177     if (hasVotedForEvent === true ) {
178         setDisability(true);
179     } else {
180         setDisability(false);
181     }
182 },
183 [
184
185     const handleChoiceSelect = (choiceIndex) => (
186         setSelectedChoiceId(choiceIndex);
187     );
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1197
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1397
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1497
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1597
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1697
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2096
2097
2098
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2196
2197
2198
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2396
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495

```

Εικόνα 43- Real time vote μέρος 2

```

188
189 const handleSubmitVote = async () => {
190   if (selectedchoiceid !== null) {
191     try {
192       let userId;
193       const votingRef = doc(database, 'Community', location, 'voting', votingIndex);
194
195       // Update the choice count in the "results" subcollection
196       const resultDocRef = doc(votingRef, 'results', voteChoices[selectedChoiceId].toString());
197       await setDoc(resultDocRef, [
198         { count: increment(1) },
199       ], { merge: true });
200
201       const subcollectionName = voteChoices[selectedChoiceId].toString();
202       const subcollectionRef = doc(database, 'Community', location, 'voting', votingIndex, 'results', subcollectionName);
203
204       const subcollectionQuerySnapshot = await getDoc(subcollectionRef);
205       const numberOfsubcollections = subcollectionQuerySnapshot.size;
206       const allVotes = subcollectionQuerySnapshot.data().votesids || [];
207       await updateDoc(subcollectionRef, {
208         'votesids': [
209           ...allVotes,
210           [auth.currentUser?.uid]: {
211             | time: serverTimestamp(),
212           },
213         ],
214       });
215       console.log(`Number of subcollections under '${subcollectionName}': ${numberOfsubcollections}`);
216
217       setSelectedChoiceId(null);
218
219       const usersPart = query(collection(database, "Community", location, 'users'), where('user', '==', auth.currentUser?.email));
220       const querySnapshot = await getDocs(usersPart);

```

Εικόνα 44 - Υποβολή ψήφου μέρος 1

```

229
230   const userRef = doc(database, 'Community', location, 'users', userId);
231   const userDoc = await getDoc(userRef);
232   const userData = userDoc.data();
233   await updateDoc(userRef, {
234     votedEvents: {
235       ...userData.votedEvents,
236       [voteTitle]: true,
237     },
238   });
239
240   fetchVoteVoting();
241   // Show a success message to the user
242   Alert.alert('Vote submitted successfully');
243   navigation.navigate('Vote')
244 } catch (error) {
245   console.error('Error submitting vote:', error);
246 }
247 } else {
248   Alert.alert('Please select a choice before submitting');
249 }
250 };

```

Εικόνα 45 - Υποβολή ψήφου μέρος 2

Ο χρήστης, για την κάθε ψήφο που υποβάλλει, δημιουργεί το πρωτεύον κλειδί (id) του και την ημερομηνία πραγματοποίησης του γεγονότος στα πεδία (fields) του «results» κάθε ψηφοφορίας. Πλάι στα id των χρηστών που έχουν ψηφίσει, υπάρχει και ένα πεδίο που ονομάζεται 'votes'. Μας δείχνει το άθροισμα των ψήφων. Προσοχή! Η προσμέτρηση των ψήφων γίνεται αθροίζοντας κάθε φορά όλα τα ids που υπάρχουν σαν πεδία.

6.4.8 Λειτουργίες Διαγραφής Περιπτών Δεδομένων

A) Χρήστες:

Όταν κάποιοι χρήστες υπάρχουν προσωρινοί σε μία περιοχή, και παράλληλα με την είσοδο τους συμμετάσχουν σε μία διαδικασία της ομάδας, τα στοιχεία τους θα παραμείνουν καταγεγραμμένα. Όμως, η παρουσία τους στους ουσιαστικά ενεργούς χρήστες της κοινότητας δεν θα αποτελεί πραγματικότητα. Συνεπώς αυτοί οι χρήστες πρέπει να διαγράφονται από τους ενεργούς χρήστες. Πιως όμως μπορούμε να καταλάβουμε αν ένας χρήστης δεν είναι ενεργός στην κοινότητα;

Χρησιμοποιούμε ουσιαστικά την λογική της «ημερομηνίας λήξης». Στην προκειμένη περίπτωση, αυτή η ημερομηνία είναι οι 7 ημέρες (1 εβδομάδα). Όταν ο χρήστης είναι ανενεργός για πάνω από μία βδομάδα, τότε γίνεται αντιληπτό από το σύστημα ότι δεν συμμετέχει πλέον στις διαδικασίες ενεργά, συνεπώς δεν αποτελεί ενεργό χρήστη. Το σύστημα πρέπει με ένα τρόπο να ξεκαθαρίζει τους ενεργούς από τους ανενεργούς χρήστες. Σε αυτό το σημείο, όπως αναφέρθηκε και στο κεφάλαιο 4.4 θα χρησιμοποιήσουμε την τεχνολογία που μας δίνει το cloud functions. Πρόκειται για την μεταφόρτωση κώδικα στο βάση δεδομένων το οποίο θα λειτουργεί ανεξάρτητα από την εφαρμογή και θα κάνει την διαγραφή των χρηστών κάθε ένα χρονικό διάστημα.

```
43 exports.deleteInactiveUsers = functions.https.onRequest(async (req, res) => {
44   try {
45     const { location } = req.body;
46     const weekAgo = new Date();
47     weekAgo.setDate(weekAgo.getDate() - 7);
48     const q = database.firebaseio().collection("Community")
49       .doc(location).collection("users")
50       .where("createdAt", "<", weekAgo);
51
52     const inactiveUsersSnapshot = await q.get();
53
54     inactiveUsersSnapshot.forEach(async (userDoc) => {
55       const userId = userDoc.id;
56
57       await database.collection("Community").doc(location).collection("users")
58         .doc(userId)
59         .delete();
60     });
61     return res.status(200).send("Inactive users deleted successfully.");
62   } catch (error) {
63     console.error(error);
64     return res.status(500).send("Error deleting inactive users.");
65   }
66 });
67 }
```

Εικόνα 46 - Διαγραφή των ανενεργών χρηστών

Β) Εκδηλώσεις:

Την ίδια διαδικασία χρειάζεται να εφαρμόσουμε για περιπτά πράγματα που αφορούν τις πληροφορίες ή λειτουργίες στην κοινότητα. Οι εκδηλώσεις (events) που έχουν προγραμματιστεί απ' τον διαχειριστή για μία συγκεκριμένη ημερομηνία, εμφανίζονται στην αρχική σελίδα τις κοινότητας. Όταν όμως αυτές οι εκδηλώσεις αποτελούν παρελθόν για την κοινότητα, θα πρέπει να απομακρύνονται από την αρχική σελίδα για να αποφεύγεται η υπερφόρτωση. Συνεπώς ακολουθούμε την ίδια διαδικασία όπως και στους χρήστες, μόνο που στην προκειμένη περίπτωση, η διαγραφή ενός event θα γίνεται μία μέρα μετά της λήξης της ημερομηνίας του.

Είναι σημαντικό να μην μένουν περιπτά δεδομένα στην εφαρμογή για την ξεκάθαρη αλληλεπίδραση του χρήστη με την κοινότητα, καθώς και την εξοικονόμηση χρόνου και δεδομένων.

```
68 exports.deleteInactiveEvents = functions.https.onRequest(async (req, res) => {
69   try {
70     const { location } = req.body;
71     const date = new Date();
72     const dateString = date.toISOString().split("T")[0];
73     const q = database.firebaseio().collection("Community").doc(location)
74       .collection("event")
75       .where("date", "<", dateString);
76
77     const inactiveEventsSnapshot = await q.get();
78
79     inactiveEventsSnapshot.forEach(async (eventDoc) => {
80       const eventId = eventDoc.id;
81
82       await database.collection("Community").doc(location)
83         .collection("event").doc(eventId)
84         .delete();
85     });
86     return res.status(200).send("Inactive events deleted successfully.");
87   } catch (error) {
88     console.error(error);
89     return res.status(500).send("Error deleting inactive events.");
90   }
91 });
```

Εικόνα 47- Διαγραφή των περιπτών εκδηλώσεων

Functions

Dashboard Usage

 Protect your Functions resources from abuse, such as billing fraud or phishing [Configure App Check](#) [X](#)

 Looking for logs and health reporting? Visit the Google Cloud Console for a highly customizable [logs view](#), [per-function usage details](#), and [error reporting](#).

[Dismiss](#)

| Function | Trigger | Version | Requests (24 hrs) | Min / Max Instances | Timeout |
|--|---|---------|-------------------|---------------------|---------|
| <code>deleteInactiveEvents</code> us-central1 | HTTP Request https://us-central1-fir-eth-60feb.cloudfunctions.net/deleteInactiveEvents | v1 | 0 | 0 / - | 1m |
| <code>deleteInactiveUsers</code> us-central1 | HTTP Request https://us-central1-fir-eth-60feb.cloudfunctions.net/deleteInactiveUsers | v1 | 0 | 0 / - | 1m |

Items per page: 25 ▾ 1 – 2 of 2 < >

Εικόνα 48- το ανέβασμα των functions στο cloud firestore

Στο επόμενο κεφάλαιο 6.5 αναλύεται η λειτουργία της βάσης δεδομένων η ο οποία προκύπτει απ' το κάλεσμα όλων των συναρτήσεων που καλέσαμε στο κεφάλαιο 6.4.

6.5 Τα Αποτελέσματα στη Βάση Δεδομένων

Όταν ο χρήστης πραγματοποιήσει είσοδο ή εγγραφή δημιουργείται στο collection

Εικόνα 49- το collection «Community»

Community το document με ID τον Ταχυδρομικό Κώδικα (ΤΚ) που βρίσκεται. Μέσα σε αυτό δημιουργεί κάποια subcollections. Με το που μπει στο room, το πρώτο subcollection που δημιουργείται είναι το users. Το id που αντιστοιχεί στον χρήστη ισούται με το uid του authentication του. Το «users» περιέχει τα πεδία (fields) που βλέπουμε στην παρακάτω εικόνα:

Εικόνα 50- το subcollection «users» με τα πεδία για κάθε χρήστη

Στη συνέχεια ο χρήστης έχει τη δυνατότητα να δηλώσει υποψηφιότητα ως διαχειριστής της κοινότητας. Με την ενέργεια αυτή δημιουργείται το subcollection «candidates» το οποίο εμπεριέχει documents τα uid των αντίστοιχων χρηστών

που συμμετέχουν στην εκλογική διαδικασία. Αποθηκεύεται στο document του κάθε χρήστη στο πρωτεύον κλειδί (id) των χρηστών που τον ψήφισαν καθώς και ο συνολικός αριθμός των ψήφων.

| GR_115_25 | candidates | aun1MfaAEFVg5jcuc9QuYopfL313 |
|---------------------------------|--------------------------------|---|
| + Start collection | + Add document | + Start collection |
| candidates > | aun1MfaAEFVg5jcuc9QuYopfL313 > | + Add field |
| chat | | choice: "test@mail.com" |
| + Add field | | usersVotesHim |
| city: "Athens" | | aun1MfaAEFVg5jcuc9QuYopfL313 |
| country: "Greece" | | time: September 28, 2023 at 12:29:13 PM UTC+3 |
| name: "7th Municipal Community" | | votes: 1 |
| postalCode: "GR_115_25" | | |

Εικόνα 52- το «candidates» subcollection. τα documents και τα fields των εγγραφών των υποψηφίων

| GR_115_25 | event | X00R4BtSMHFhxNkher4c |
|---------------------------------|------------------------|--|
| + Start collection | + Add document | + Start collection |
| chat | X00R4BtSMHFhxNkher4c > | + Add field |
| event > | | date: "2023-09-30" |
| + Add field | | description: "event1" |
| city: "Athens" | | location: "Plateia Pinelopis Delta, Pinelopis Delta 16, Athina 115 25, Greece" |
| country: "Greece" | | time: "13:47:00" |
| name: "7th Municipal Community" | | title: "Event1" |
| postalCode: "GR_115_25" | | type: "event" |

Εικόνα 51- Το «event» subcollection με τα documents και τα fields των εγγραφών των events

Εικόνα 53- Το «voting» subcollection, τα documents και τα fields των εγγραφών των ψηφοφοριών

Όταν ο χρήστης ως διαχειριστής δημιουργήσει ένα event ή ένα voting, δημιουργείται το subcollection «event» ή το «voting» αντίστοιχα με τα ανάλογα πεδία που του έχει δώσει ο διαχειριστής.

Στην εικόνα 34 παρατηρούμε τα πεδία (fields) που έχει το κάθε voting. Παράλληλα παρατηρούμε και ένα subcollection μέσα στο voting που ονομάζεται

Εικόνα 54- Το «results» subcollection με τα documents να παίρνουν τιμές τις υποψήφιες απαντήσεις

«results». Το «results» δημιουργεί documents με ids τα choices που επιλέγουν οι χρήστες. Κάθε id περιέχει τα ids των χρηστών που ψήφισαν την συγκεκριμένη απάντηση. Παρακάτω μπορούμε να δούμε τι εμπεριέχει το «results»:

Με την ίδια λογική όπως και στο candidates, κάθε ψήφος προσμετράται από το κάθε πρωτεύον κλειδί (id) που ψήφισε την εκάστοτε άποψη.

6.6 Ασφάλεια Δεδομένων και Ρόλοι Χρηστών

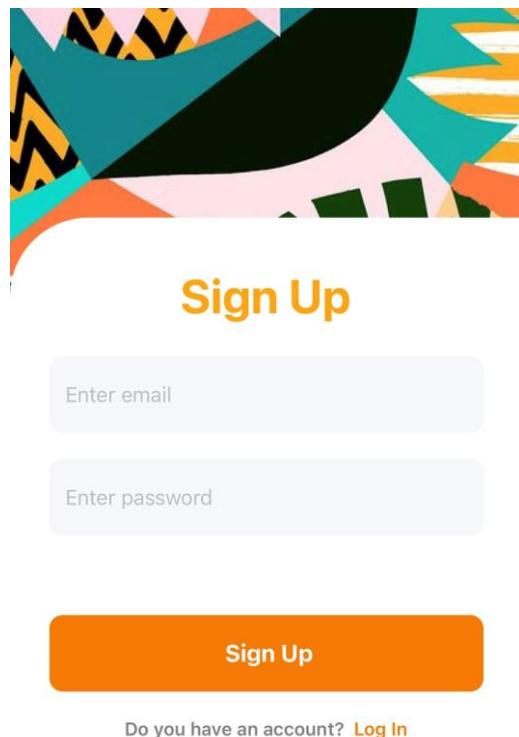
Η αλληλεπίδρασή των χρηστών με τα δεδομένα πρέπει να εμπεριέχει τη δυνατότητα της δικαιοδοσίας για τον κάθε χρήστη που έχει πρόσβαση στο σύστημα και στα δεδομένα του cloud. Όταν ένας χρήστης δεν έχει συνδεθεί στην κοινότητα πρέπει οι δυνατότητες του να είναι περιορισμένες, παρά την μη δυνατότητα αλληλεπίδρασης με όλες τις οθόνες της εφαρμογής. Αυτό γίνεται για λόγους ασφάλειας των δεδομένων από κακόβουλους χρήστες (hackers) αλλά και για να υπάρξει και μία τάξη στις λειτουργίες της εφαρμογής. Γι' αυτό το λόγο, όπως θα δούμε παρακάτω στην εικόνα 55, δίνουμε την δικαιοδοσία μόνο σε όσους έχουν συνδεθεί στην κοινότητα να μπορούν να κάνουν ανάγνωση τα δεδομένα της. Δίνουμε την δυνατότητα μόνο στον διαχειριστή της κάθε κοινότητας να μπορεί να κάνει εγγραφή τα δεδομένα των events. Όπως επίσης μόνο ο χρήστης που έχει ψηφίσει με το αντίστοιχο πρωτεύον κλειδί (id) μπορεί να πάρει πίσω την ψήφο, δηλαδή να κάνει εγγραφή στο subcollection «candidates» και να αφαιρέσει το δικό του πρωτεύον κλειδί (id) και μόνο από τις ψήφους. Το ίδιο κάνουμε και για την διαδικασία των αποτελεσμάτων μιας ψηφοφορίας.

```
3   service cloud.firestore {
4     match /databases/{database}/documents {
5
6       match /Community/{document=**} {
7
8         allow read: if true;
9
10        match /candidates/{userId} {
11          allow read: if request.auth != null;
12          allow write: if request.auth != null && request.auth.uid == userId;
13        }
14        match /event/{eventId} {
15          allow read: if request.auth != null;
16          allow write: if request.auth != null && isAdmin();
17        }
18        function isAdmin() {
19          return get(/databases/${database}/documents/Community/{document}/users/${request.auth});
20        }
21        match /votings/{document} {
22          allow read, write: if request.auth != null;
23
24          match /results/{document} {
25            allow read: if request.auth != null;
26            allow create, update: if request.auth != null &&
27              !(resource.data.votesIds.keys() == request.auth.uid);
28            allow delete: if request.auth != null &&
29              (resource.data.votesIds.keys() == request.auth.uid);
30          }
31        }
32        match /chat/{document} {
33          allow read, write: if request.auth != null;
34        }
35        match /users/{document} {
36          allow read: if request.auth != null;
37          allow write: if request.auth != null;
38        }
      }
```

Εικόνα 55- Δικαιοδοσίες ανάγνωσης και εγγραφής στους χρήστες

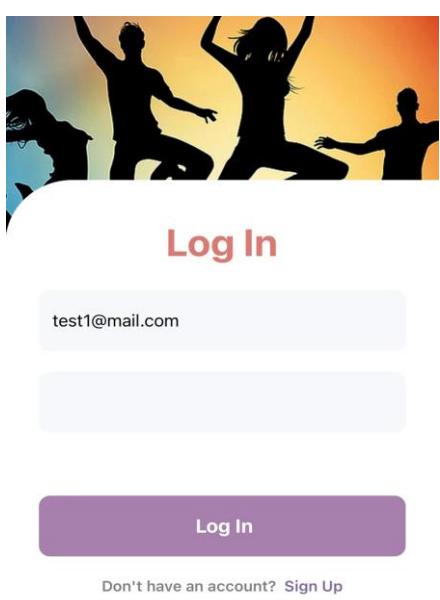
7. ΤΟ ΣΥΣΤΗΜΑ ΣΕ ΛΕΙΤΟΥΡΓΙΑ

Στο παρακάτω κεφάλαιο θα δείξουμε όλες τις λειτουργίες με τις οποίες ένας απλός χρήστης του συστήματος βλέπει και αλληλεπιδρά με αυτές.

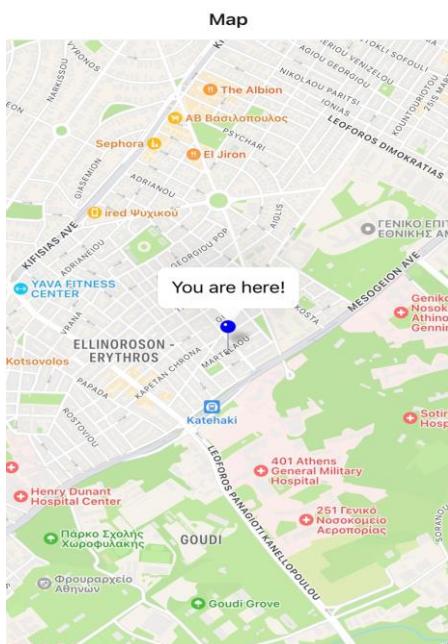


Εικόνα 57- Αναμονή της οθόνης μέχρι το location να πάρει την σωστή τιμή

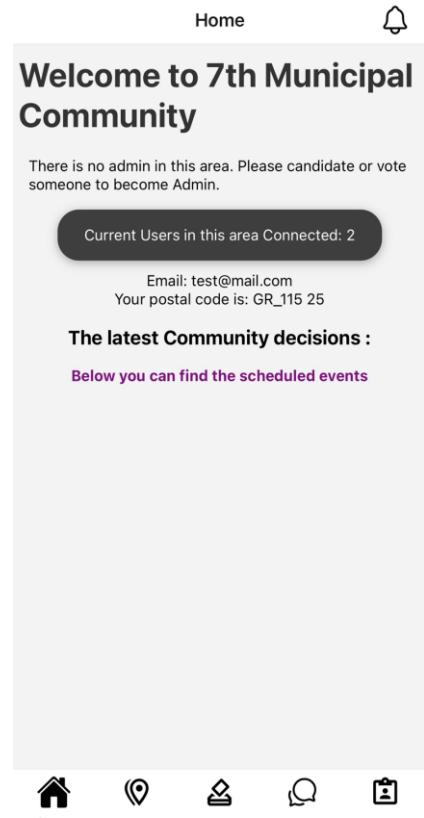
Εικόνα 56- Η οθόνη εγγραφής χρήστη στο σύστημα



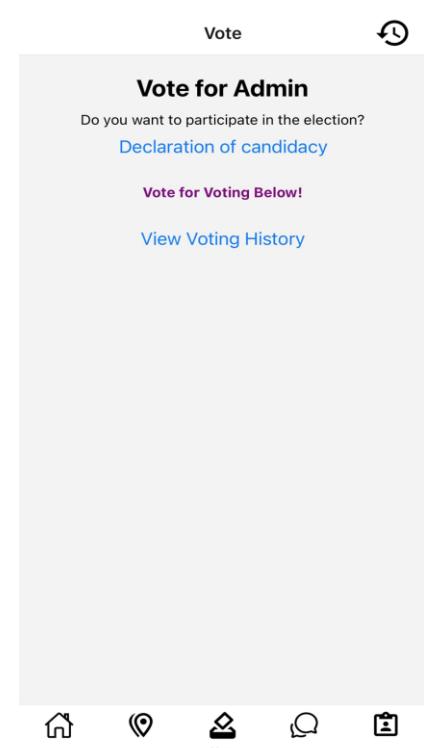
Εικόνα 59- Η οθόνη εισόδου χρηστών με το ηλεκτρονικό ταχυδρομείο και τον κωδικό ως πεδία



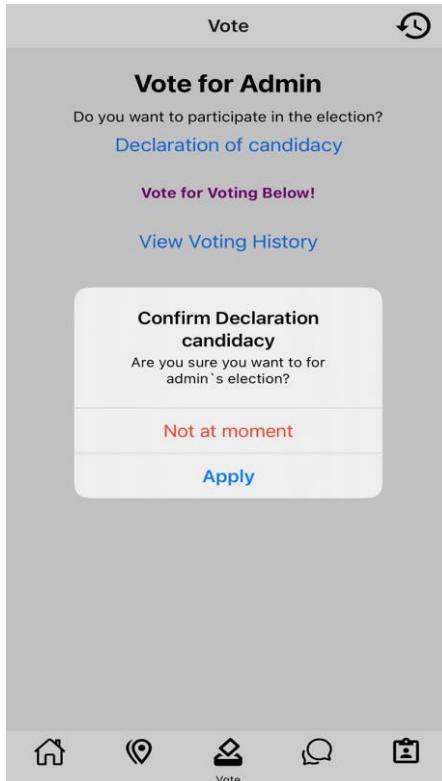
Εικόνα 60- Ο χάρτης τοποθεσίας του χρήστη και των εκδηλώσεων



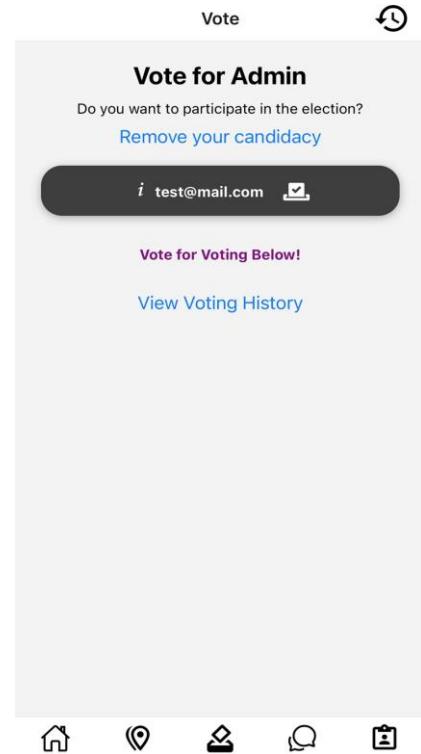
Εικόνα 58- Η οθόνη της αρχικής σελίδας της κοινότητας



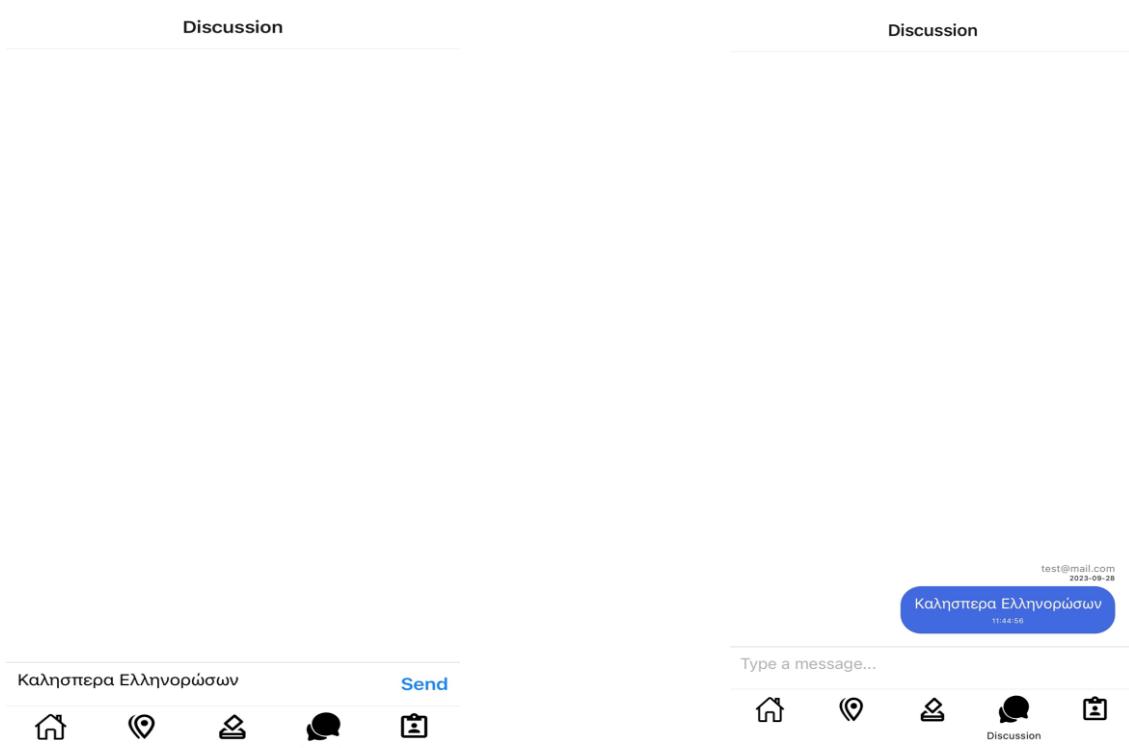
Εικόνα 61- Η οθόνη του «vote» component



Εικόνα 62- Επιλογή του χρήστη να βάλει υποψηφιότητα για admin και το αντίστοιχο Alert



Εικόνα 63- Ο χρήστης επέλεξε Apply και πλέον είναι στους υποψήφιους



Εικόνα 64- To «Chat» component

Εικόνα 65- Ο χρήστης στέλνει ένα μήνυμα στην ομαδική συζήτηση

Profile

Describe yourself about participating in elections.
Why should your fellow citizens vote for you as a community leader?

Description:

Log out

Delete Your Account

Home
Location
Announcement
Comment
Profile

Profile

Describe yourself about participating in elections.
Why should your fellow citizens vote for you as a community leader?

Description:

Log out

Delete Your Account

Home
Location
Announcement
Comment
Profile

Εικόνα 66- Ο χρήστης βρίσκεται στο component «Profile»

Εικόνα 67- Ο χρήστης αλλάζει την περιγραφή για τον εαυτό του που προβάλει στους υπόλοιπους χρήστες μέσω της υποψηφιότητας

Vote

Vote for Admin

Do you want to participate in the election?

[Remove your candidacy](#)

i test@mail.com

Description: Είμαι ο κατάλληλος για admin στο Ελληνορώσων

[Vote for Voting Below!](#)

[View Voting History](#)

Home
Location
Announcement
Comment
Vote

Vote

Vote for Admin

Do you want to participate in the election?

[Remove your candidacy](#)

i test@mail.com

Description: Είμαι ο κατάλληλος για admin στο Ελληνορώσων

[Vote for Voting Below!](#)

Remove your candidacy

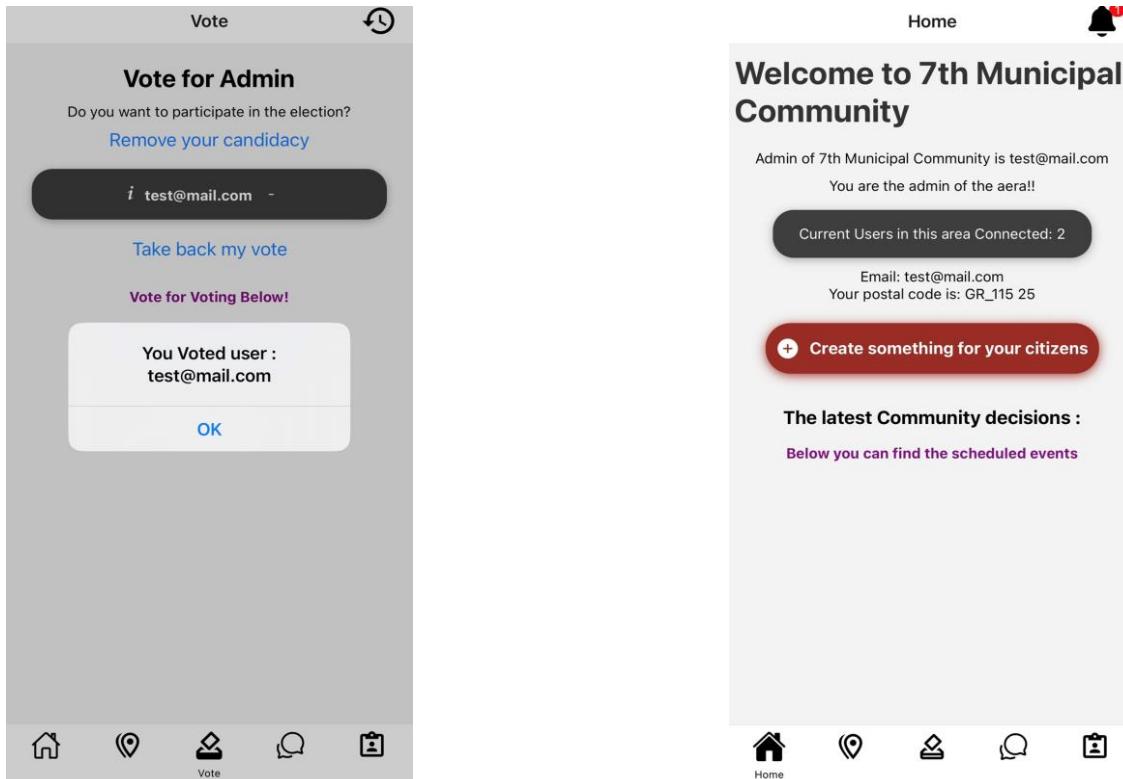
Are you sure you want to remove your candidacy?

[Don't remove](#)
[Remove](#)

Home
Location
Announcement
Comment
Vote

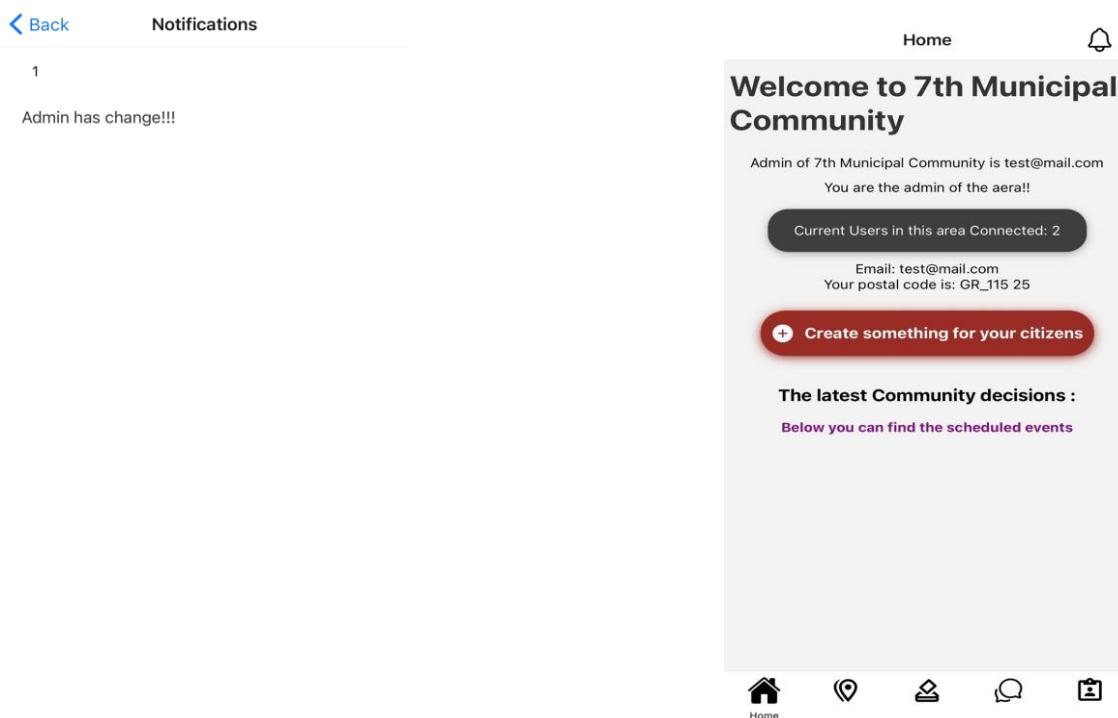
Εικόνα 68- Ο χρήστης βλέπει την περιγραφή ενός υποψηφίου

Εικόνα 69- Ο χρήστης επέλεξε να ακυρώσει την υποψηφιότητά του



Εικόνα 70- Ο χρήστης ψήφισε
έναν υποψήφιο πατώντας το
εικονίδιο δεξιά του υποψηφίου

Εικόνα 71- Ο χρήστης βλέπει στην
οθόνη «Home» ότι είναι ο admin
της κοινότητας



Εικόνα 73- Οι ειδοποίησης όταν ο
αλλάζει ο διαχειριστής της
κοινότητας

Εικόνα 72- Πατώντας το «clear
notifications» το εικονίδιο
καθάρισε

Discussion

Event

Voting for

Back **CreateVoting**

Event

Voting for

Title: Event1

Description: event1

Location:

Date: 28 Sep 2023

Time: 11:47

Eikόνα 74- Ο διαχειριστής στη συζήτηση της κοινότητας είναι διακριτός

Back **CreateVoting**

Event

Voting for

Title: Event1

Description: event1

Location:

Date: 30 Sep 2023

Time: 13:47

Eikόνα 76- Επεξεργασία ώρας και μέρας απ' την by default

Home **Bell**

Welcome to 7th Municipal Community

Admin of 7th Municipal Community is test@mail.com
You are the admin of the area!!

Current Users in this area Connected: 2
Email: test@mail.com
Your postal code is: GR_115 25

You plan Event1

OK

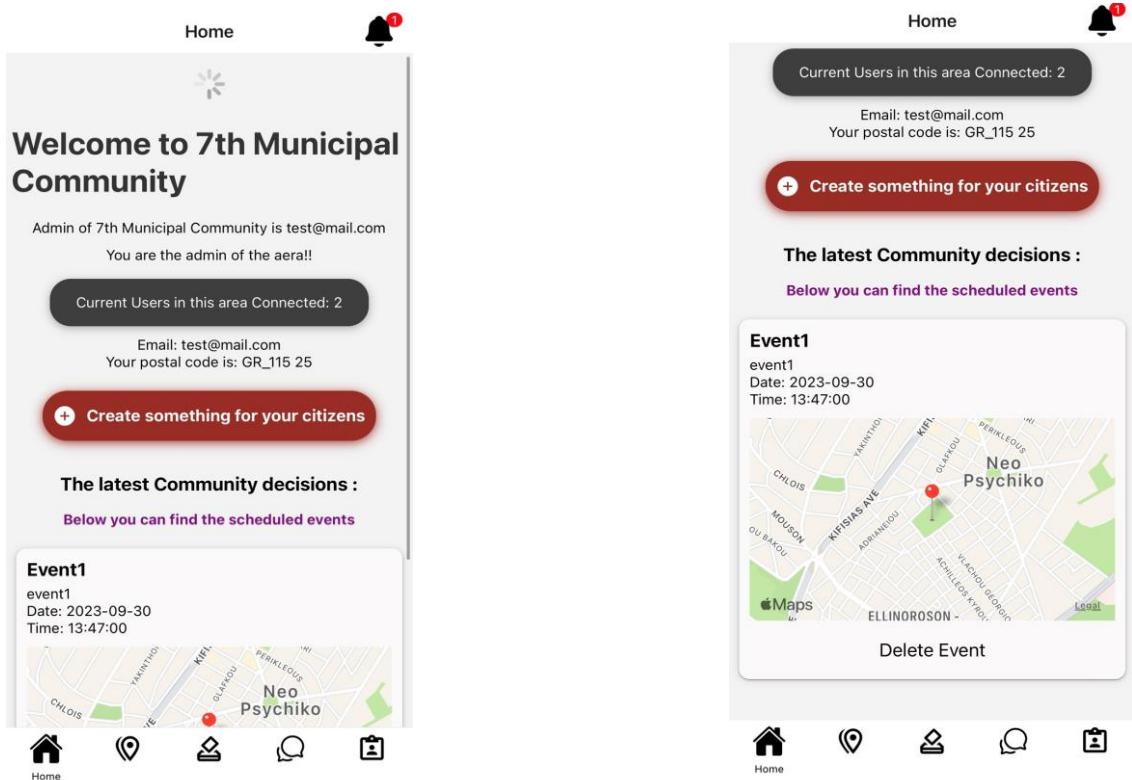
Below you can find the scheduled events

Back **CreateVoting**

Event

Voting for

Eikόνα 77- Δημιουργία event και το αντίστοιχο alert

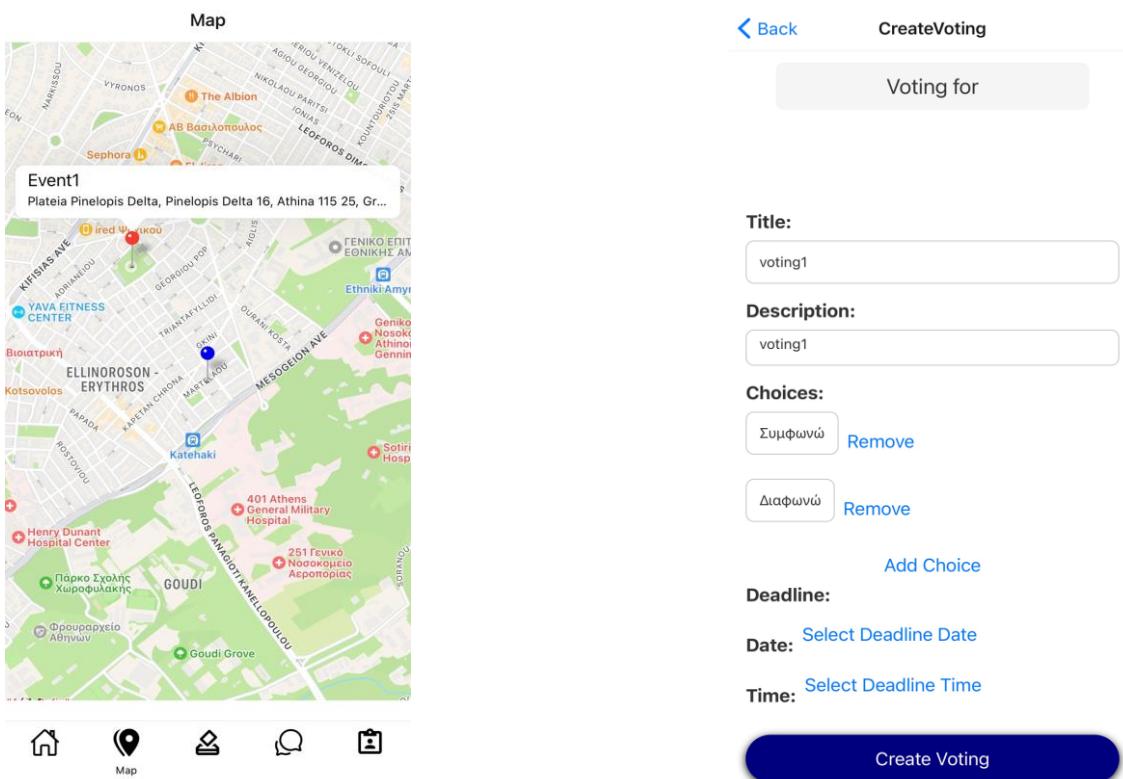


Εικόνα 78- Εμφάνιση του event στην αρχική σελίδα κατόπιν reload

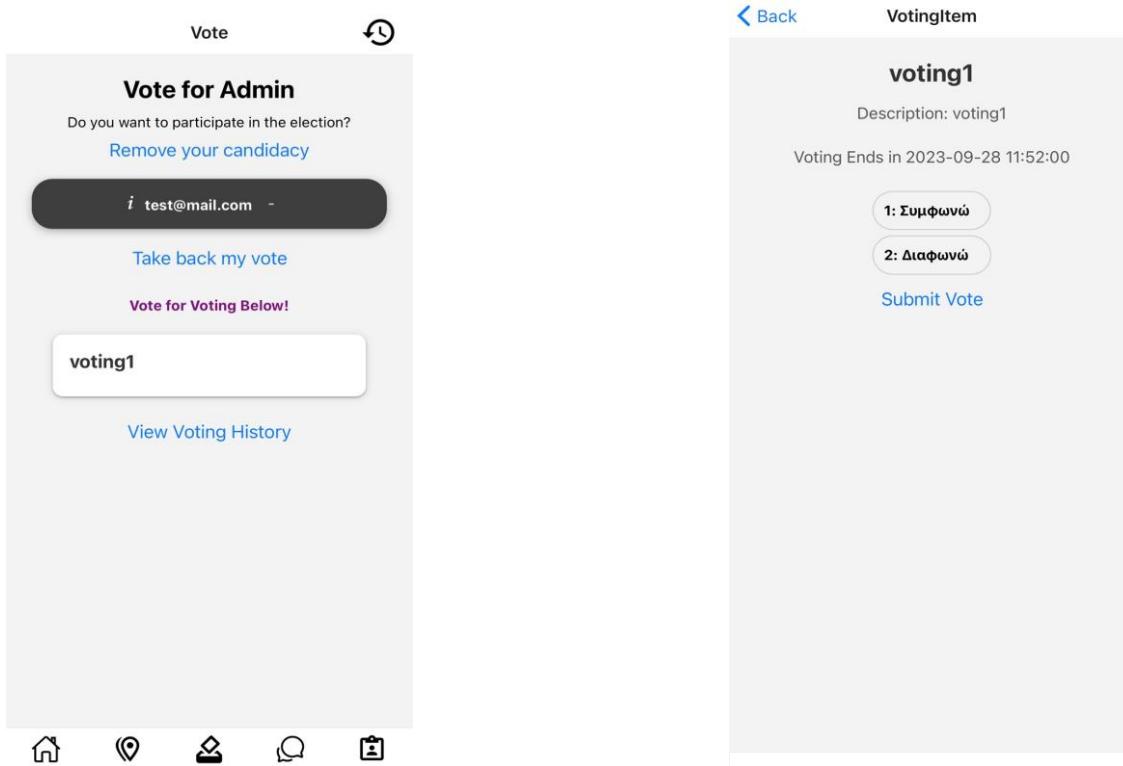
Εικόνα 79- Η αρχική σελίδα με την επιλογή για διαγραφή του event από τον διαχειριστή

Εικόνα 80- Η ειδοποίηση από την δημιουργία event

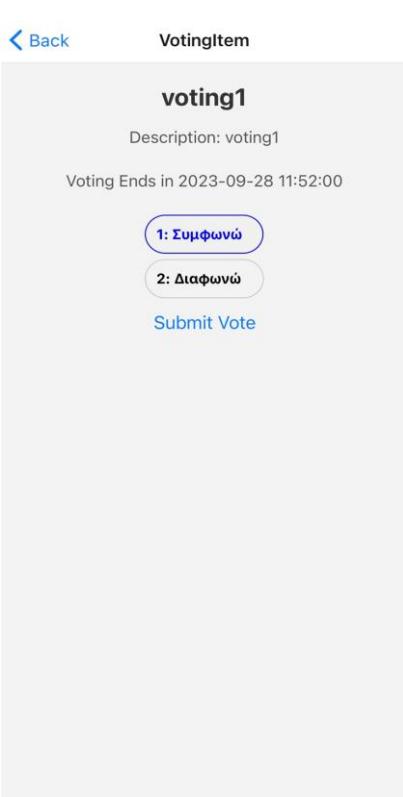
Εικόνα 81- Επιλογή και ειδοποίηση διαγραφής event



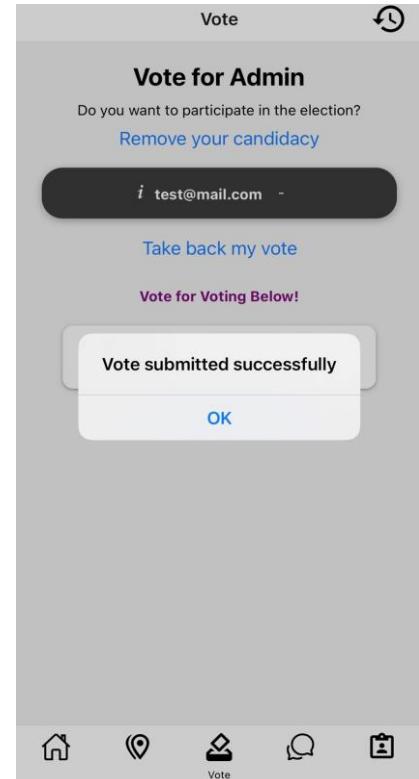
Εικόνα 82- Το στίγμα της τοποθεσίας του event στον χάρτη



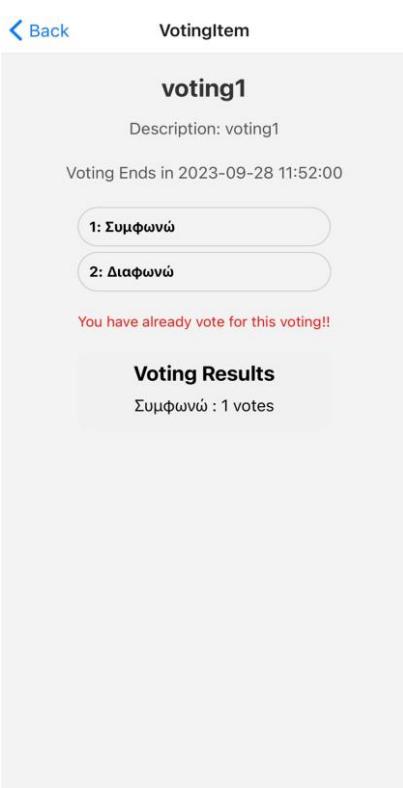
Εικόνα 84- Η κάρτα του voting1 με τις επιλογές ως προς ψήφιση



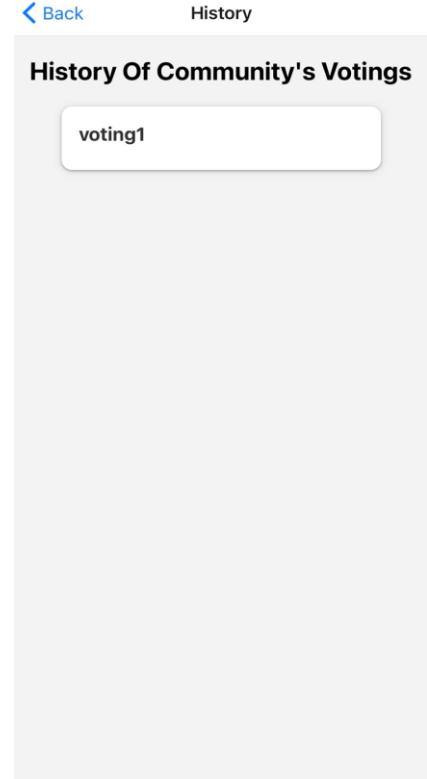
Εικόνα 86 - Επιλογή μιας συγκεκριμένης απάντησης



Εικόνα 87- Υποβολή της απάντησης και μεταφορά στο «vote» component



Εικόνα 88- Ο χρήστης έχει ανοίξει μια κάρτα στην οποία έχει ήδη ψηφίσει



Εικόνα 86- Αφού λήξει η προθεσμία υποβολής, η κάρτα μεταφέρεται στην οθόνη History

[History](#)

VotingItem

voting1

Description: voting1

Voting Ends in 2023-09-27 11:52:00

1: Συμφωνώ

2: Διαφωνώ

You have already vote for this voting!!

The DeadLine of this voting has exceed

Voting Results

Συμφωνώ : 1 votes

The screenshot shows the main screen of the app. At the top, there's a navigation bar with a bell icon and the word "Home". Below it is a large, bold title "Welcome to 7th Municipal Community". Underneath the title, a message says "Admin of 7th Municipal Community is test@mail.com" and "You are the admin of the area!!". A central box contains the text "Current Users in this area Connected: 2" above a list of users with their email addresses. Below this is another box with the text "Email: test@mail.com" and "Your postal code is: GR_115 25". A red button at the bottom left encourages users to "Create something for your citizens". The next section, titled "The latest Community decisions : ", lists a voting decision: "Voting Title: voting1" with "Community decided : Συμφωνώ: 1 votes". Finally, a purple text box at the bottom says "Below you can find the scheduled events".

Εικόνα 87- Ο χρήστης έχει ανοίξει την κάρτα voting1 που υπάρχει πλέον στο ιστορικό

Εικόνα 88- Στην αρχική σελίδα φαίνεται η απόφαση που πήρε η κοινότητα ως προς το voting1

8. ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΒΕΛΤΙΩΣΕΙΣ

Στο τελευταίο κεφάλαιο της ππυχιακής εργασίας μας θα αναφερθούμε στα συμπεράσματα που προέκυψαν από την εκπόνησης και την υλοποίησης της εργασίας αλλά και προοπτικές βελτίωσης που πιστεύουμε ότι θα βοηθήσουν την πλατφόρμα μας να γίνει ακόμα καλύτερη.

Συμπεράσματα

Ως προς την υλοποίηση της εφαρμογής, η χρήση της React Native ως κύριο framework υλοποίησης της εφαρμογής αποδείχθηκε αρκετά χρήσιμη και κατανοητή. Μας παρείχε πολλαπλές δυνατότητες όσον αφορά την τοποθεσία του χρήστη, την επικοινωνία με την βάση μέσω αλληλεπίδρασης πραγματικού χρόνου (real time) και την προσφορά της στη διεπαφή (interface) της εφαρμογής. Η δυναμικές αλλαγές που διαθέτει, αλλά και το ασυγχρόνιστο μοντέλο που παρέχει μας βοήθησαν να δουλέψουμε με την κατάλληλη λεπτομέρεια και αντλώντας τεράστιο πλήθος επιλογών, εξειδικευμένων για κάθε περίπτωση.

Η Firebase αποτέλεσε μια χρήσιμη βάση δεδομένων μιας και η λειτουργία με τα collection και τα documents γλυτώνει χρόνο και κόπο από τον χρήστη. Επίσης συμβάλλουν στην πολύ καλή οργάνωση της βάσης δεδομένων. Το cloud μας δίνει την απαραίτητη ασφάλεια για να μην έχουμε απώλεια δεδομένων.

Για την εφαρμογή επιλέξαμε να είναι απλή και σύντομη ώστε να μπορεί ένας χρήστης να την λειτουργήσει με ταχύτητα και άνεση μιας και ζούμε στην εποχή της ταχύτητας. Διαλέξαμε την δυναμική αλληλεπίδραση με όλα τα μέλη μιας κοινότητας με την λεπτομέρεια ότι ένας χρήστης έχει τη δυνατότητα να έρθει πιο κοντά με τους χρήστες της περιοχής του μέσω της δημοκρατίας και της αυτοοργάνωσης, αλλά και να μπορέσει να γνωρίσει και να ανακαλύψει ξένες τοποθεσίες που επισκέπτεται πρώτη φορά. Να αναζητήσει εκδηλώσεις και να δει τι αποφάσεις παίρνει η εκάστοτε κοινότητα. Επίσης βοηθάει τον χρήστη να μάθει να λειτουργεί σε ένα πλαίσιο ευθύνης ως εκπρόσωπος μιας κοινότητας με ανάλογα αποτελέσματα και συνέπειες της λειτουργίας του ως προς το δυναμικό το οποίο τον επέλεξε για αυτήν την θέση.

Μελλοντικές Βελτιώσεις

Όσον αφορά τις βελτιώσεις που θα μπορούσαν να γίνουν ως προς την περαιτέρω εξέλιξη της λειτουργίας μιας ηλεκτρονικής κοινότητας έχουμε τις εξής:

Σε ότι έχει να κάνει με την ασφάλεια της βάσεις δεδομένων, είναι σημαντικό να δίνεται η σωστή εξουσιοδότηση σε οποιονδήποτε χρησιμοποιεί την εφαρμογή.

Η αλληλεπίδραση με το room να έχει και τις κατάλληλες προτεινόμενες εικόνες που είναι αποτυπώνουν τα πιο δημοφιλή μέρη κάθε περιοχής.

Η επιλογή των χρηστών να μπορούν να μοιράζονται το προφίλ τους και τις εικόνες τους.

Η αλλαγή σε πιο σύγχρονα API και την εμφάνιση/χρήση βασικών σημείων στον χάρτη κάθε περιοχής.

Το σύστημα των ειδοποιήσεων να εξελιχθεί με περισσότερες ειδοποιήσεις και με πλήρη λειτουργικότητα με την βάση δεδομένων.

Μπορεί να χρησιμοποιηθεί η λειτουργία των rules για την ασφάλεια των δεδομένων με έλεγχο και σωστά δικαιώματα ανάγνωσης και εγγραφής στους χρήστες.

Χρήση των cloud function για επιπρόσθετες λειτουργίες κατά τη μη χρήση της εφαρμογής από κάποιον χρήστη.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

| Ξενόγλωσσος όρος | Ελληνικός Όρος |
|------------------|-----------------------|
| Social media | Κοινωνική δικτύωση |
| Chat | Κουβέντα |
| Event | Εκδήλωση |
| Admin | Διαχειριστής |
| Application | Εφαρμογή |
| e-Voting | Ηλεκτρονική ψηφοφορία |
| Mobile | Κινητό τηλέφωνο |
| Developer | Προγραμματιστής |
| Live reloads | Ζωντανές ανανεώσεις |
| Project | Εργασία |
| Screen | Οθόνη |
| By default | Από προεπιλογή |
| Root | Ρίζα |
| Authentication | Αυθεντικοποίηση |
| Reload | Ανανέωση |
| Submit | Υποβολή |

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

1. **Mobile geoblogging** – Ανάπτυξη μιας android social geoblogging εφαρμογής (n.d.).
https://dspace.lib.ntua.gr/xmlui/bitstream/handle/123456789/4653/papagian_nopoulos_geoblogging.pdf?sequence=3. Μία πτυχιακή εργασία με επίκεντρο μια mobile εφαρμογή που λειτουργεί με γεωγραφικά δεδομένα με σκοπό την κοινωνική δικτύωση. Του Κωνσταντίνου Παπαγιαννόπουλου
2. Google. (n.d.). **Build documentation | firebase documentation**. Google.
<https://firebase.google.com/docs/build> . Η επίσημη ιστοσελίδα της firebase από την οποία αντλήσαμε αρκετό υλικό και βοηθηθήκαμε πάνω στο κομμάτι των τεχνικών και των τεχνολογιών που αφορούν την σύνδεση react-firebase.
3. *LogRocket blog. LogRocket Blog.* (n.d.). <https://blog.logrocket.com/>. Αποτελεί blog που αφορά τεχνολογίες που αφορούν την react native.
4. Hall, T. (2012). Electronic voting. In N. Kersting, M. Stein, & J. Trent (Eds.), *Electronic Democracy* (1st ed., pp. 153–176). Verlag Barbara Budrich.
<https://doi.org/10.2307/j.ctvddzwcq.10>. Το βιβλίο αναφέρεται σε συστήματα ηλεκτρονικής δημοκρατίας αναλύωντας περιπτώσεις και τρόπους εφαρμογής της δημοκρατίας μέσω της τεχνολογίας
5. Taş, R., & Tanrıöver, Ö. Ö. (2020, August 9). **A systematic review of challenges and opportunities of blockchain for E-voting**. MDPI.
<https://www.mdpi.com/2073-8994/12/8/1328>. Το άρθρο εξετάζει τις προκλήσεις και τις ευκαιρίες της τεχνολογίας blockchain όσον αφορά την ηλεκτρονική ψηφοφορία.
6. Rana, N. P., Williams, M. D., Dwivedi, Y. K., & Williams, J. (2012). **Theories and Theoretical Models for Examining the Adoption of E-Government Services**. *E-Service Journal*, 8(2), 26–56.
<https://doi.org/10.2979/eservicej.8.2.26>. Το άρθρο αναλύει θεωρίες και θεωρητικά μοντέλα που χρησιμοποιούνται για να μελετήσουν την υιοθέτηση των υπηρεσιών ηλεκτρονικής διακυβέρνησης (e-government services).
7. Kneuer, M. (2016). **E-democracy**: A new challenge for measuring democracy. *International Political Science Review / Revue Internationale de Science Politique*, 37(5), 666–678. <https://www.jstor.org/stable/26556880>. Το άρθρο εξετάζει τη σχέση μεταξύ της ηλεκτρονικής δημοκρατίας και της άμεσης δημοκρατίας
8. *React native · learn once, write anywhere*. **React Native RSS**. (n.d.).
<https://reactnative.dev/>. Η επίσημη ιστοσελίδα της react-native από την οποία πήραμε τις βασικές τεχνολογίες για την δημιουργία του project και την υλοποίηση των τεχνολογιών
9. *Create amazing apps that run everywhere*. **Expo Documentation**. (n.d.).
<https://docs.expo.dev/>. Η επίσημη ιστοσελίδα της Expo.

10. **How to use react native geolocation to get postal address.** Jscrambler. (n.d.).<https://jscrambler.com/blog/how-to-use-react-native-geolocation-to-get-postal-address>. Η ιστοσελίδα που μας βοήθησε για την χρήση του expo location.
11. Rosenzweig, P. (2018). **THINKING ANALYTICALLY ABOUT ELECTORAL SECURITY.** R Street Institute. <http://www.jstor.org/stable/resrep19127>. Το άρθρο βασίζεται στο cyber security.
12. **Wikimedia Foundation.** (2023, April 25). *Μέσα κοινωνικής δικτύωσης*. Wikipedia. [Μέσα κοινωνικής δικτύωσης - Βικιπαίδεια \(wikipedia.org\)](#). Πληροφορίες για τα social media.
13. Abbott, J. (2012). **Social media.** In N. Kersting, M. Stein, & J. Trent (Eds.), **Electronic Democracy** (1st ed., pp. 77–102). Verlag Barbara Budrich. <https://doi.org/10.2307/j.ctvddzwcq.7>. Το κεφάλαιο αυτό του βιβλίου εξετάζει το ρόλο των κοινωνικών μέσων ενημέρωσης (social media) στο πλαίσιο της ηλεκτρονικής δημοκρατίας.
14. Boyd, D. M., & Ellison, N. B. (2010). **Social network sites: Definition, history, and scholarship.** *IEEE Engineering Management Review*, 38(3), 16–31. <https://doi.org/10.1109/emr.2010.5559139> Το παραπάνω βιβλίο αναφέρεται στην ιστορία και εξέλιξη της επικοινωνίας και της κοινωνικοποίησης των ανθρώπων μέσω των μέσων κοινωνικής δικτύωσης και του διαδικτύου.
15. **Zeus electronic voting.** (n.d.). <https://zeus.grnet.gr/zeus/faqs/trustee/> Το zeus αφορά ένα σύστημα ηλεκτρονικών ψηφοφοριών (e-voting) από το οποίο πήραμε ιδέες για το σύστημα ψηφοφορίας που υλοποιήθηκε καθώς και για τον ρόλο του Διαχειριστή.
16. Marx, K., Engels, F., & Lenin, V. I. (1988). Marx, Engels, Lenin on **democracy-bourgeois and Socialist**. Progress Publishers. Το συγκεκριμένο έργο αφορά κείμενα των Μαρξ, Ενγκελς και Λένιν που αφορούν τις μορφές δημοκρατίας που έχει γνωρίσει η ανθρωπότητα.
17. Hsiao, C.-H., Chang, J.-J., & Tang, K.-Y. (2016). **Exploring the influential factors in continuance usage of mobile social apps: Satisfaction, habit, and Customer Value Perspectives.** *Telematics and Informatics*, 33(2), 342–355. <https://doi.org/10.1016/j.tele.2015.08.014>. εξετάζει τους παράγοντες που επηρεάζουν τη συνεχή χρήση των κινητών κοινωνικών εφαρμογών (mobile social apps) από την προοπτική της ικανοποίησης του χρήστη.
18. **Models of E-democracy** - association for information systems. (n.d.-b). <https://aisel.aisnet.org/cgi/viewcontent.cgi?article=3002&context=cais>. Το παρόν επιστημονικό άρθρο αναφέρεται σε μοντέλα δημοκρατίας και ηλεκτρονικής δημοκρατίας.
19. Doytsher, Y., Galon, B., & Kanza, Y. (2010). **Querying geo-social data by bridging spatial networks and Social Networks.** Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks. <https://doi.org/10.1145/1867699.1867707>. Αυτή η εργασία

επικεντρώνεται στη δυνατότητα ερωτήσεων και ανάκτησης δεδομένων από τον χώρο και τα κοινωνικά δίκτυα, συνδυάζοντας πληροφορίες από χωρικά δίκτυα (σχετικές με την τοποθεσία) και κοινωνικά δίκτυα (σχετικά με τις κοινωνικές σχέσεις).

20. H. Yin et al., "**Discovering interpretable geo-social communities for user behavior prediction**," 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland. <https://doi.org/10.1109/icde.2016.7498303>. Αποτελεί έρευνα που ασχολείται με την ανακάλυψη και την ανάπτυξη ερμηνεύσιμων γεω-κοινωνικών κοινοτήτων (*geo-social communities*) για την πρόβλεψη της συμπεριφοράς των χρηστών.
- 21 G. Cardone et al., "**Fostering participation in smart cities: a geo-social crowdsensing platform**," in IEEE Communications Magazine, vol. 51, no. 6, pp. 112-119, June 2013, doi: 10.1109/MCOM.2013.6525603. Το παραπάνω άρθρο ασχολείται με την ανάπτυξη τεχνικών και μοντέλων που θα προωθούν την ενεργό συμμετοχή των πολιτών στην παρακολούθηση και τη βελτίωση της πόλης τους. Βασίζεται σε γεωγραφικά και κοινωνικά δεδομένων.
22. K. P. N. Puttaswamy et al., "**Preserving Location Privacy in Geosocial Applications**," in IEEE Transactions on Mobile Computing, vol. 13, no. 1, pp. 159-173, Jan. 2014, doi: 10.1109/TMC.2012.247.