



Orgoma

Λεβαντής Χρήστος, Προπτυχιακός φοιτητής

321-2020110

Τμήμα Μηχανικών και Πληροφοριακών Συστημάτων

Σάμος, 2023-2024

Εφαρμογές κινητού υπολογισμού

Περιεχόμενα

1	Πληροφορίες για την εφαρμογή	3
1.1	Τεχνολογίες που χρησιμοποιήθηκαν	4
1.2	Βασικά διαγράμματα χρήσης	5
1.2.1	Use case για τις βασικές λειτουργίες των χρηστών	5
1.2.2	Use case την διαδικασία της αυθεντικοποίησης	7
1.2.3	Sequence diagram για την Mark a field	8
1.2.4	Sequence diagram για την Record	11
1.3	Ικανοποίηση απαιτήσεων εφαρμογής	14
1.3.1	Ικανοποίηση 1 ^{ης} απαίτησης	14
1.3.2	Ικανοποίηση 2 ^{ης} απαίτησης	17
1.3.3	Ικανοποίηση 3 ^{ης} απαίτησης	20
1.3.4	Ικανοποίηση 4 ^{ης} απαίτησης	25
1.3.5	Ικανοποίηση 5 ^{ης} και 6 ^{ης} απαίτησης	26
1.3.6	Ικανοποίηση 7 ^{ης} απαίτησης	28
2	Οδηγίες εγκατάστασης	31
2.1	Εγκατάσταση από το GitHub	31
3	Χαρακτηριστικά βίντεο σχετικά με την εφαρμογή	32
4	Κατάλογος σχημάτων	33

1 Πληροφορίες για την εφαρμογή

Η παρούσα εφαρμογή σκοπό έχει να αποτελέσει ένα ολοκληρωμένο βοηθό για τους ανθρώπους που ασχολούνται με τις γεωργικές εργασίες συγκεκριμένα με τον ψεκασμό χωραφιών.

Η εφαρμογή δίνει την δυνατότητα στους χρήστες να εισάγουν τα χωράφια τους μαζί με κάποιες σημαντικές πληροφορίες για αυτά πάνω στον χάρτη. Επίσης δίνει την δυνατότητα στους χρήστες να γνωρίζουν ποια χωράφια είναι βιολογικής καλλιέργειας και ποια όχι. Δηλαδή ποια χωράφια μπορούν να ψεκαστούν και ποια όχι.

Υπάρχουν δύο βασικοί ρόλοι στην εφαρμογή, ο πρώτος είναι ο **Owner (ιδιοκτήτης)** ο οποίος είναι αυτός που θα εισάγει τα χωράφια του επάνω στον χάρτη. Η εφαρμογή διακρίνει και τον ρόλο του **Spray worker (Εργάτης ψεκασμού)**, ο οποίος θα ψεκάζει τα χωράφια για την καταπολέμηση των παράσιτων.

1.1 Τεχνολογίες που χρησιμοποιήθηκαν

Η εφαρμογή έχει υλοποιηθεί με την χρήση της γλώσσας **Java** με την βοήθεια του IDE **Android Studio** και προορίζεται μόνο για **Android συσκευές**. Στην διαχείριση των δεδομένων χρησιμοποιήθηκε η μη σχεσιακή βάση δεδομένων **Firestore database**. Η δομή που έχει φαίνεται παρακάτω.

+ Start collection	+ Add document	+ Start collection
Fields >	4n33up12UItLdceMmFH >	+ Add field
Users	H5nfJNODI1647b52zaBT IDj09wbh8NGyYVay0quN LXfdQcJVKZ9ebfJLF77i MN6cDu8atLu7qIuuzSsT N6ZUw8myj15gb3vVJUAN bM7CX0jpAjYQWj8Y9KUP bRusJUGBcystVrNwFIq dzjYrNXsYPOZ8cQuFYx6 myomCDfRbpZ9qEpZlzm rpnxC4iiZC206zZ0tQcN tfU2iyQ625Rg29ckFyF4 znEIXpckMQN1DMZ2yeIo zsyLpsCrDWxAPI8oJp7M	cord_1: [38.02483153728906° N, 23.900504037737846° E] cord_2: [38.02459859039119° N, 23.90063278377056° E] cord_3: [38.02471612038122° N, 23.90108473598957° E] cord_4: [38.024957518466096° N, 23.90093319118023° E] creator_email: "cris13levantis13@gmail.com" creator_name: "Chris LEV." olive_variety: "Gaeta" organic_farming: "Yes" sprayed: "No"

Figure 1-1 Εικόνα της βάσης δεδομένων

Επίσης χρησιμοποιήθηκε και η τεχνολογία **Firestore database** για το στάδιο της αυθεντικοποίησης αλλά και πολλές άλλες τεχνολογίες οι οποίες παρέχονται από την Google όπως είναι το **Google Authenticate**, **Locations** και **Maps**.

Για την ωραιοποίηση του UI της εφαρμογής χρησιμοποιήθηκε το **material** το οποίο πρόσφερε ωραία TextFields, InputFields και Spinners.

Τέλος πέρα από όλες τις παραπάνω τεχνολογίες προφανώς πολύ χρήσιμες ήταν και αυτές που παρέχονται από το Android (για την Java γλώσσα). Όλες οι χρησιμοποιούμενες τεχνολογίες μπορούν να εντοπιστούν στο build.gradle.kts(:app) αρχείο στο σημείο που δηλώνονται τα dependencies.

1.2 Βασικά διαγράμματα χρήσης

Θα παρουσιαστούν κάποια διαγράμματα χρήσης με σκοπό την καλύτερη κατανόηση της λειτουργίας της εφαρμογής αλλά και καλύτερη γνώση των αλληλεπιδράσεων που λαμβάνουν χώρα ανάμεσα στις οντότητες της εφαρμογής.

Ξεκινάμε με την παρουσίαση κάποιων Use case διαγραμμάτων και στην συνέχεια κάποιων Sequence diagrams για τις δύο πιο βασικές λειτουργίες της εφαρμογής μας. Μέσα από τα διαγράμματα θα γίνει και διάκριση των ρόλων που υπάρχουν.

1.2.1 Use case για τις βασικές λειτουργίες των χρηστών

Basic user activities

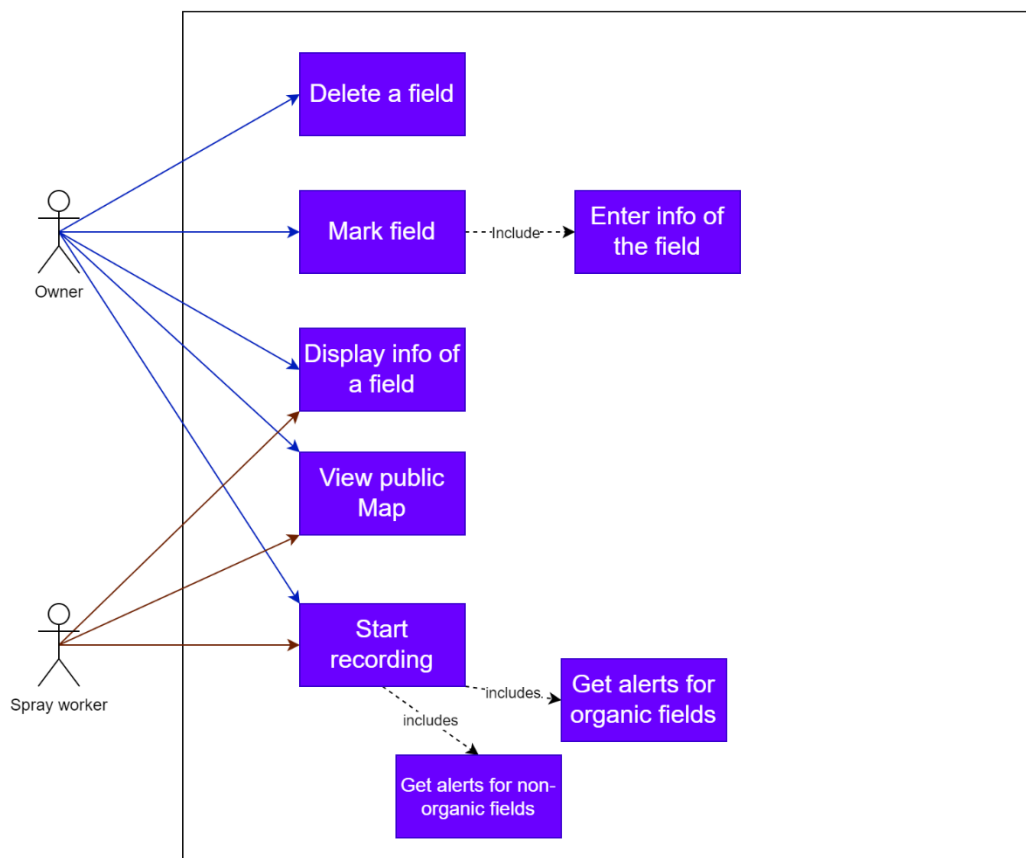


Figure 1-2 Basic user activities, Use case

Όπως βλέπουμε η εφαρμογή μας διακρίνει δύο ρόλους, τον **Owner** ο οποίος είναι ο ιδιοκτήτης ενός χωραφιού και έναν **Spray worker** ο οποίος είναι αυτός που θα ψεκάζει τα

χωράφια. Οι βασικές λειτουργίες που προσφέρονται από την εφαρμογή είναι οι διαγραφή ενός χωραφιού (Delete a field), μαρκάρισμα ενός χωραφιού (Mark a field) η οποία μετέπειτα περιέχει και την λειτουργία εισαγωγής των στοιχείων του χωραφιού¹, Απεικόνιση του δημόσιου χάρτη (Display public map) και έναρξη εγγραφής (Start recording) η οποία δίνει την δυνατότητα στον χρήστη να πάρει ειδοποιήσεις όταν πλησιάζει οργανικά χωράφια αλλά και όταν πλησιάζει μη-οργανικά χωράφια (διότι άλλαξαν κατάσταση από μη ψεκασμένα σε ψεκασμένα).

Με βάση το παραπάνω διάγραμμα δημιουργούμε το παρακάτω πίνακα ώστε να δούμε ποιες λειτουργίες προσφέρονται στον κάθε χρήστη.

	Mark field	Start recording	Delete field	Display info of a field	View public Map
Owner	+	+	+ ²	+	+
Spray worker		+		+	+

Παρατηρούμε με βάση τον πίνακα αλλά και το διάγραμμα ότι ο Owner έχει δυνατότητα να πραγματοποιήσει όλες τις προσφερόμενες δυνατότητες της εφαρμογής³.

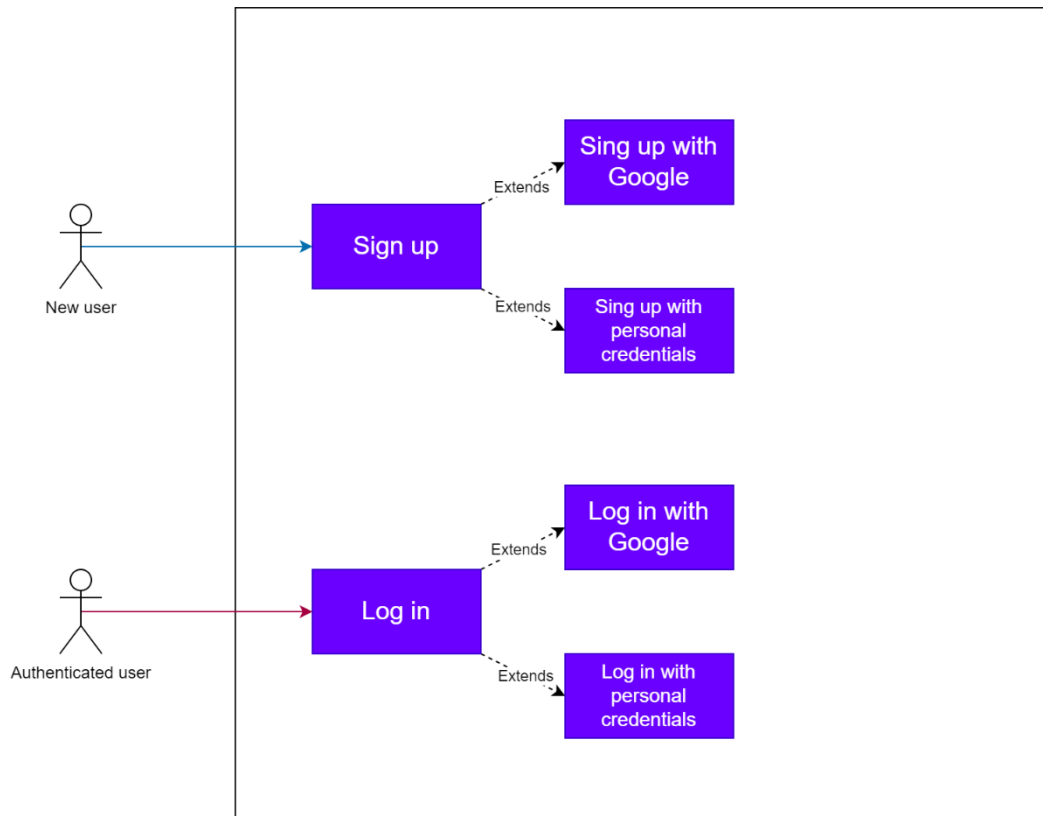
¹ Έγινε include και όχι extend διότι είναι απαραίτητη για την εκπλήρωση της Mark field και δεν μπορεί να παραληφθεί.

² Μόνο αν του ανήκει.

³ Στην πραγματικότητα δεν θα έπρεπε να δίνεται η δυνατότητα Record στον χρήστη Owner γιατί δεν είναι ο βασικός σκοπός του αυτός.

1.2.2 Use case την διαδικασία της αυθεντικοποίησης

Log in and Sing up



Πολύ απλό διάγραμμα το παραπάνω δεν χρειάζεται κάποια ιδιαίτερη περιγραφή.

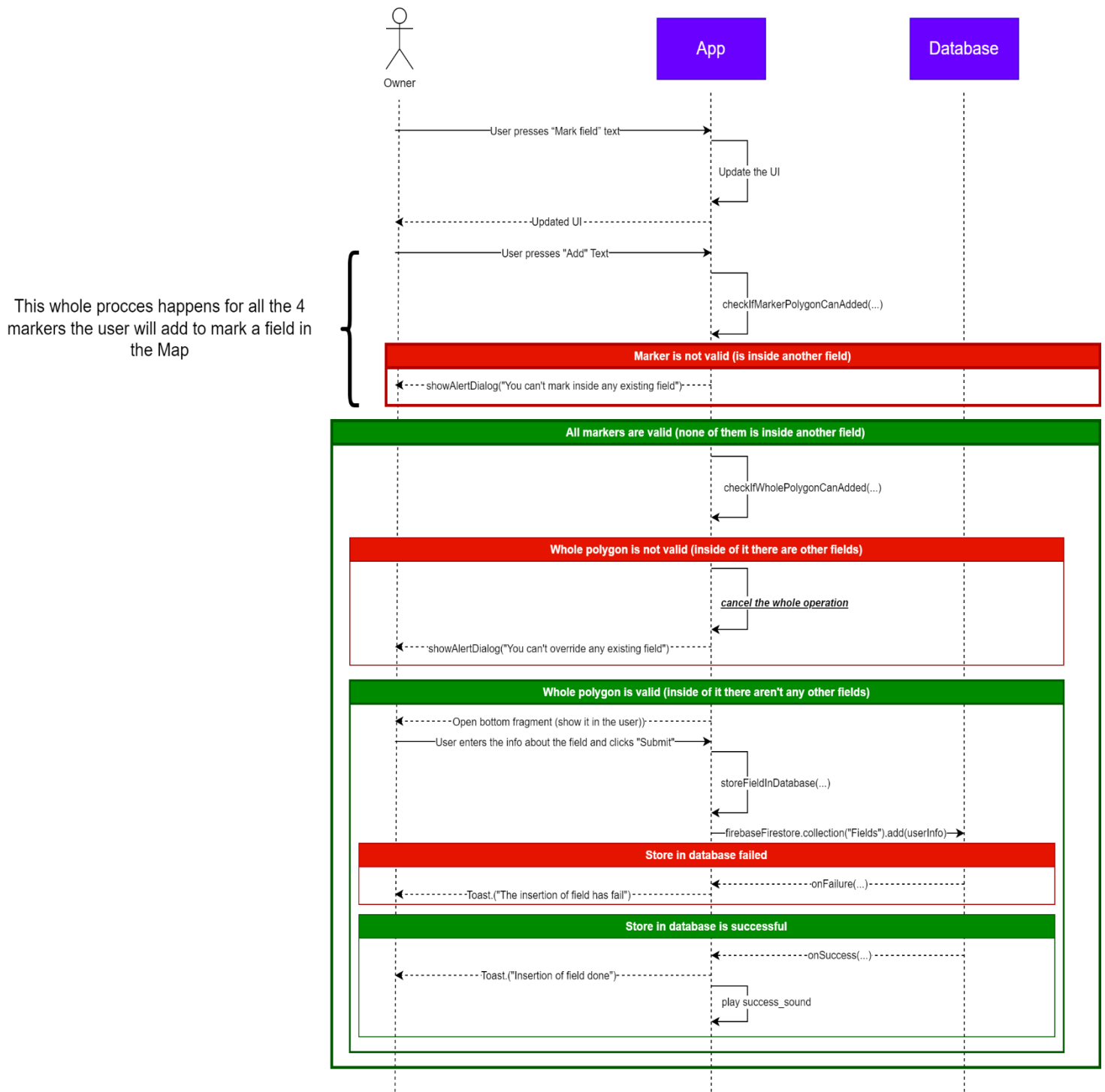
Διακρίνουμε ως actor έναν New user ο οποίος θα έχει κατεβάσει για πρώτη φορά την εφαρμογή στο κινητό του οπότε προφανώς δεν θα κατέχει ακόμα κάποιο account στην εφαρμογή μας. Για αυτό την δραστηριότητα που σχετίζεται άμεσα είναι η Sign up και μέσα από αυτήν του δίνονται δύο δυνατότητες να κάνει Sign up με την Google ή με προσωπικό του email. Τέλος έχουμε και τον actor που ονομάζεται Authenticated user που αναπαριστά κάποιον ήδη εγγεγραμμένο χρήστη στην εφαρμογή μας. Ως Authenticated user αναπαριστούμε έναν Owner ή έναν Spray worker. Σε αυτόν τον actor του δίνεται πάλι η δυνατότητα να κάνει Log in ή με την Google ή με το προσωπικό του mail.

Να σημειώσουμε σε αυτό το σημείο ότι κάθε χρήστης *έχει ένα μόνο ρόλο*. Η *αυθεντικοποίηση και η μοναδικότητα γενικά καθορίζεται από το email* κάποιου προσώπου.

1.2.3 Sequence diagram για την Mark a field

Περνάμε τώρα στα διαγράμματα ακολουθίας, ξεκινώντας από την πρώτη βασικότερη λειτουργία η οποία είναι η διαδικασία μαρκαρίσματος ενός χωραφιού.

Mark a field



Η διαδικασία αυτή ξεκινά με τον χρήστη, ο οποίος είναι ο Owner καθώς μόνο αυτός μπορεί να μαρκάρει χωράφια να έχει πατήσει το κουμπί Mark Field που βλέπει στο αρχικό UI του. Στην συνέχεια η εφαρμογή θα εκτελέσει κάποιες εσωτερικές ενέργειες για να ανανεώσει το UI της εφαρμογής, το ανανεωμένο UI θα φανεί στον χρήστη. Στην συνέχεια ο χρήστης αφού προφανώς αλληλοεπιδράσει με τον χάρτη και πλοηγήθει στο χωράφι που θέλει (δεν το δείχνω στο διάγραμμα για αν μην γίνει πολύ λεπτομερές) θα πατήσει Add για να τοποθετήσει στην οθόνη έναν marker ο οποίος θα δηλώνει μια γωνία του χωραφιού, σημειώνουμε σε αυτό το σημείο ότι ο χρήστης πρέπει να βάλει τέσσερις τέτοιους markers. Η εφαρμογή πρέπει να ελέγξει αν αυτός ο marker είναι valid, θα το κάνει αυτό κάνοντας κάποιες εσωτερικές ενέργειες με πιο βασική την `checkIfMarkerPolygonCanAdded(...)`, καταλαβαίνουμε ότι αποτελεί μια συνάρτηση η οποία σκοπό έχει τον εντοπισμό αν ο marker που ο χρήστης τοποθέτησε είναι μέσα σε κάποιον άλλο ήδη υπάρχον χωράφι. Αν είναι τότε θεωρείται μη αποδεκτός και εμφανίζεται στην οθόνη του χρήστη το κατάλληλο Toast μήνυμα, προφανώς ο marker δεν τοποθετείται στον χάρτη. Αν τώρα δεν βρίσκεται μέσα σε κάποιο μονοπάτι τότε μπορεί να τοποθετηθεί στον χάρτη. Αυτή η διαδικασία πρέπει να γίνει τέσσερις φορές αφού πρέπει να τοποθετηθούν τέσσερις markers, και να ολοκληρωθεί θα πρέπει όλοι να είναι valid. Αν τώρα τοποθετηθούν όλοι οι markers και είναι valid θα πρέπει να εξετάσει η εφαρμογή αν το πολύγωνο που δημιουργήθηκε με βάση τους τέσσερις markers δεν περιέχει μέσα του κάποιο ήδη υπάρχον χωράφι. Αυτό αποτελεί μια ακόμα εσωτερική ενέργεια η οποία καλείται `checkIfWholePolygonCanAdded(...)`. Με βάση αυτήν την εσωτερική ενέργεια θα πρέπει να διακρίνουμε δύο περιπτώσεις. Η πρώτη είναι ότι το πολύγωνο περιέχει μέσα του άλλα χωράφια, σε αυτήν την περίπτωση η διαδικασία Mark Field τερματίζεται από την ίδια την εφαρμογή και εμφανίζεται στον χρήστη το κατάλληλο Alert μήνυμα. Αντίθετος η δεύτερη περίπτωση είναι το πολύγωνο να μην επικαλύπτει άλλα υπάρχοντα χωράφια οπότε η ροή της λειτουργίας συνεχίζεται με την ανάδειξη στον χρήστη

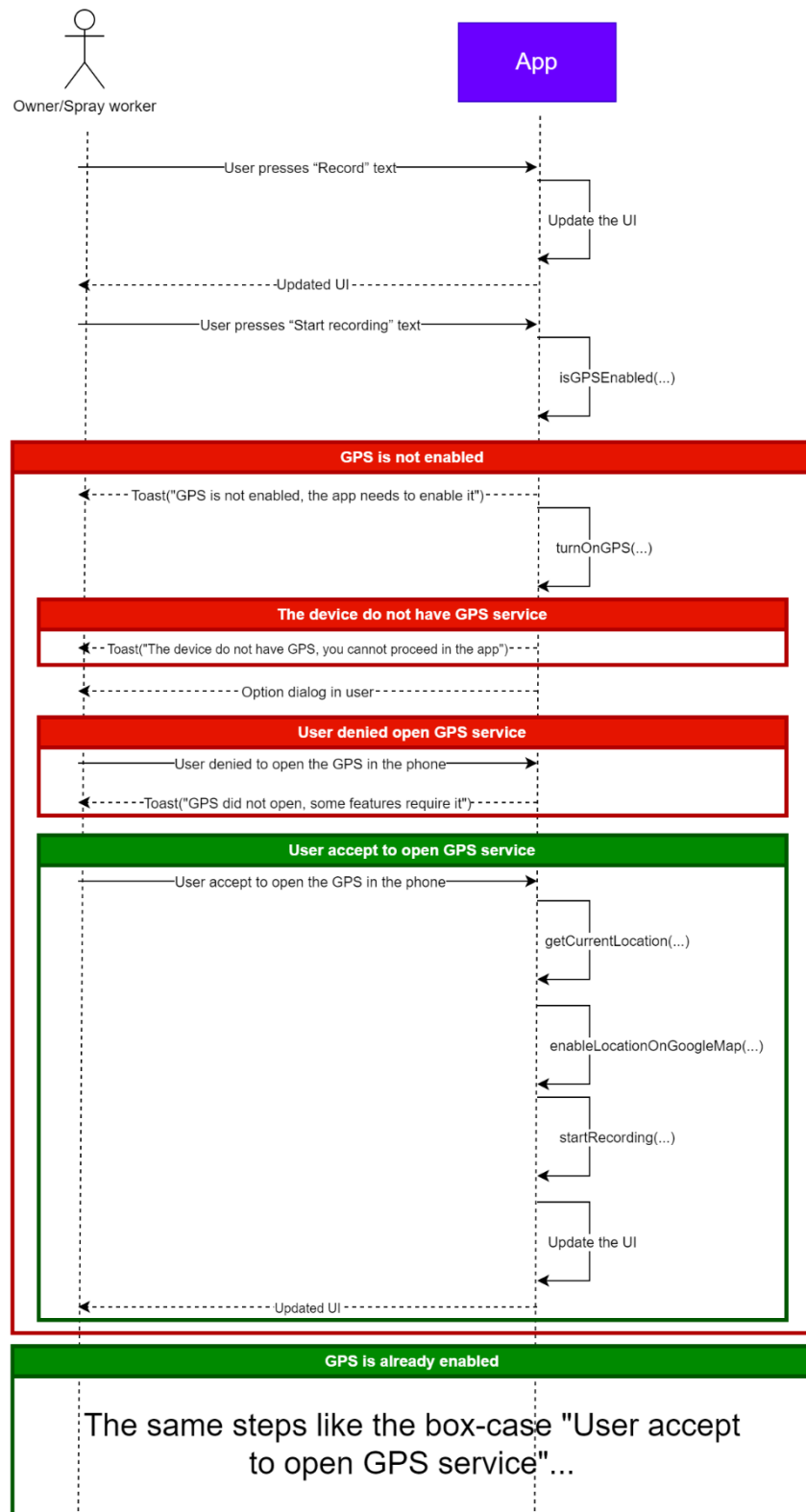
του bottom fragment στο οποίο μπορεί να εισάγει τα στοιχεία του χωραφιού. Εισάγει λοιπόν τα στοιχεία αυτά και πατάει submit. Η εφαρμογή από την πλευρά της θα καλέσει την συνάρτηση `storeFieldInDatabase(...)` που όπως λέει και το όνομα θα αποθηκεύσει στην βάση τα στοιχεία του χωραφιού μαζί με τις συντεταγμένες του χωραφιού (για αναγνώριση του). Ουσιαστικά εδώ είναι και η στιγμή που λαμβάνει δράση και η βάση δεδομένων η οποία θα μας δημιουργήσει άλλες δύο υποπεριπτώσεις καθώς μπορεί η εισαγωγή στην βάση να μην στεφθεί με επιτυχία. Αν όλα πάνε καλά θα λάβουμε έναν μήνυμα επιτυχίας από την βάση (ουσιαστικά θα καλεστεί ο `OnSuccess listener` στην εφαρμογή) και θα εμφανίσει η εφαρμογή ένα μήνυμα Toast αλλά επίσης θα παίζει και έναν ήχο επιτυχίας, φυσικά θα δημιουργηθεί και στον χάρτη το χωράφι με τον κατάλληλο χρωματισμό (δεν το δείχνω στο διάγραμμα για αποφυγή υπερβολικής λεπτομέρειας). Αν δεν πάει καλά η αποθήκευση στην βάση θα επιστραφεί το κατάλληλο μήνυμα Toast στον χρήστη και φυσικά η λειτουργία δεν θα ολοκληρωθεί με επιτυχία.

Να σημειωθεί ότι συμβαίνουν και πολλές άλλες ενέργειες στην εφαρμογή για την συγκεκριμένη λειτουργία που δεν φαίνονται στο διάγραμμα.

1.2.4 Sequence diagram για την Record

Record

Στο



συγκεκριμένο διάγραμμα θα παρουσιαστεί μια άλλη πολύ σημαντική λειτουργία η οποία

είναι η καταγραφή της διαδρομής ενός χρήστη, ουσιαστικά στο διάγραμμα θα δειχτούν μόνο οι πολύ βασικές αλληλεπιδράσεις και λειτουργίες που γίνονται για αυτήν την λειτουργία.

Παρατηρούμε ότι ως User έχουμε και έναν Owner αλλά και έναν Spray worker, επίσης παρατηρούμε ότι δεν έχουμε παρουσία της οντότητας της Βάσης Δεδομένων, αυτό μας υποδηλώνει ότι η συγκεκριμένη ενέργεια δεν απαιτεί την Βάση Δεδομένων για την επιτυχή ολοκλήρωση της, άρα ίσως γλυτώνετε και κάποιος χρόνος αναμονής στην γενική ροή της εφαρμογής μας.

Ξεκινάμε με το user να πατά το κουμπί Record στη συνέχεια στην εφαρμογή ενεργοποιείται ο listener του κουμπιού αυτού και ανανεώνει το UI, το ανανεωμένο UI γίνεται αντιληπτό από τον χρήστη. Ο χρήστης πατάει το κουμπί Start recording και η εφαρμογή θα πρέπει να εξετάσει αν το GPS είναι ενεργοποιημένο στην συσκευή, καθώς η λειτουργία record το χρειάζεται απαραίτητα για την πετυχημένη ολοκλήρωση της. Αν δεν είναι τότε αρχικά θα σταλεί ένα Toast μήνυμα που θα τον ενημερώνει για ο γεγονός ότι το GPS δεν είναι ενεργοποιημένο και η εφαρμογή θα προσπαθήσει να το ενεργοποιήσει. Οπότε στην συνέχεια θα καλεστούν εσωτερικά διάφορες ενέργειες με πιο βασική να είναι η `turnOnGPS(...)` η οποία όπως λέει και το όνομα της θα ενεργοποιήσει το GPS στην συσκευή. Υπάρχει η περίπτωση η συσκευή να μην έχει καν την δυνατότητα για την υπηρεσία GPS, οπότε σε αυτήν την περίπτωση θα πρέπει να σταλεί το κατάλληλο μήνυμα Toast στον χρήστη και φυσικά η λειτουργία δεν θα μπορεί να συνεχιστεί. Αν όμως υπάρχει η δυνατότητα GPS στην συσκευή τότε θα εμφανιστεί στον χρήστη το μήνυμα επιλογής αν θέλει να ενεργοποιήσει το GPS ή όχι (και με βάση τις νέες εκδόσεις Android αν θέλει να γίνεται ακριβής προσδιορισμός της θέσης του ή όχι). Πάλι λοιπόν θα διακρίνουμε δύο περιπτώσεις με την πρώτη να είναι η περίπτωση που ο χρήστης δεν αποδέχεται την ενεργοποίηση της υπηρεσίας GPS στο κινητό του. Τότε θα πρέπει να επιστρέψει η εφαρμογή πάλι το κατάλληλο μήνυμα Toast στον χρήστη και φυσικά πάλι η λειτουργία δεν θα μπορεί να

ολοκληρωθεί με επιτυχία. Αν όμως γίνει αποδεκτή η ενεργοποίηση της GPS υπηρεσίας τότε τα πράγματα είναι θετικά. Θα γίνουν μια σειρά από εσωτερικές ενέργειες στην εφαρμογή με πρώτη να είναι η `getCurrentLocation(...)` η οποία θα πάρει την αρχική τοποθεσία του χρήστη στον χάρτη (αλλά και στην συνέχεια κάθε φορά που αλλάζει θέση), δεύτερη είναι η `enableLocationOnGoogleMap(...)` η οποία το μόνο που θα κάνει είναι να ενεργοποιήσει το εικονίδιο πάνω στον χάρτη το οποίο θα δείχνει την θέση του χρήστη στον χάρτη. Στην συνέχεια τρίτη αλλά και πολύ σημαντική είναι η `startRecording(...)` η οποία θα βάλει την εφαρμογή σε λειτουργία εγγραφής και τέλος θα γίνει ανανέωση όλου του UI της εφαρμογής και αυτό θα φανεί στον χρήστη. Τέλος να πούμε ότι άμα το GPS ήταν ήδη ενεργοποιημένο τότε πολύ απλά θα γινόντουσαν οι ενέργειες που περιεγράφηκαν προηγούμενος.

Μια τελευταία σημείωση που πρέπει να γίνει είναι ότι μια εξίσου σημαντική συνάρτηση για την λειτουργία Record είναι η `onMyLocationChange(...)` η οποία δεν διακρίνεται στο διάγραμμα. Η οποία καλείται κάθε φορά που ο χρήστης αλλάζει θέση με σκοπό την ενημέρωση των συντεταγμένων της θέσης του.

1.3 Ικανοποίηση απαιτήσεων εφαρμογής

Στο παρόν κεφάλαιο θα παρουσιαστούν οι τρόποι ικανοποίησης των απαιτήσεων που τέθηκαν στην περιγραφή της εφαρμογής από τον πελάτη.

1.3.1 Ικανοποίηση 1^{ης} απαίτησης

1^η απαίτηση → Εμφάνιση χάρτη δημόσια προσβάσιμου όπου εμφανίζονται οι ελαιώνες (ως κουκίδες) και θα εμφανίζονται ποιοι είναι ψεκασμένοι με βάση έναν χρωματισμό. Οι βιολογικοί ελαιώνες θα εμφανίζονται με διαφορετικό χρώμα σε πράσινη απόχρωση (αυτές δεν ψεκάζονται άρα δεν θα αλλάζει το χρώμα).

Τρόπος ικανοποίησης → Η εφαρμογή στο LogInView δίνει την δυνατότητα στον χρήστη, χωρίς να συνδεθεί ή να δημιουργήσει κάποιον λογαριασμό να δει έναν χάρτη στον οποίο θα διαφαίνονται όλα τα χωράφια που έχει η εφαρμογή (εικόνα 1.3).

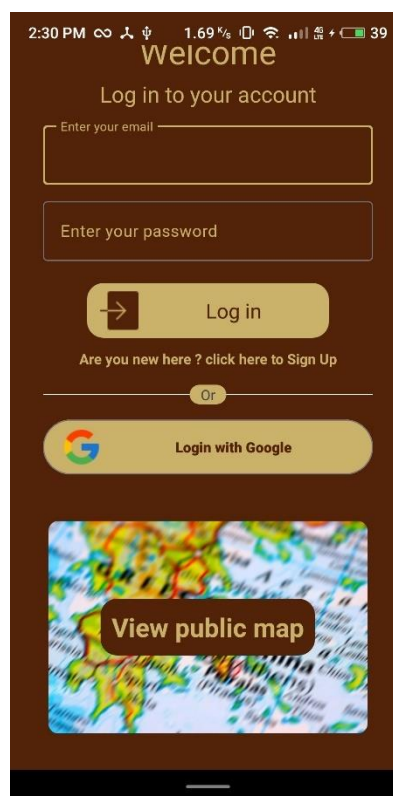


Figure 1-3Επιλογή δημόσιου χάρτη

Στον δημόσιο χάρτη ο χρήστης μπορεί να δει όλα τα χωράφια τα οποία θα έχουν κάποιους συγκεκριμένους χρωματισμούς με βάση το αν είναι βιολογικά ή όχι. Αν δεν είναι

βιολογικά τότε θα είναι ή ψεκασμένα ή όχι. Παρακάτω βλέπουμε κάποιες χαρακτηριστικές εικόνες για καλύτερη κατανόηση.



Figure 1-4 Όψη ενός βιολογικού χωραφιού στον χάρτη της εφαρμογής

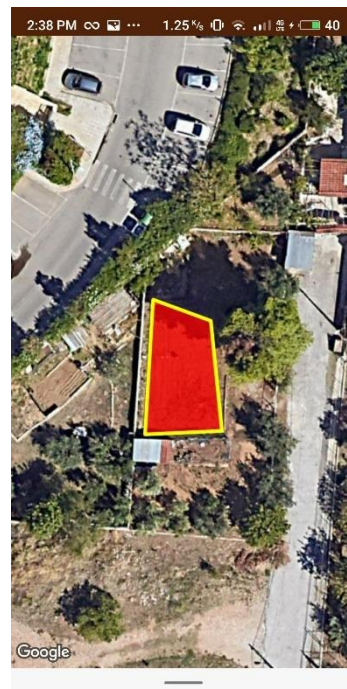


Figure 1-5 Όψη ενός μη βιολογικού χωραφιού στον χάρτη της εφαρμογής μας, το οποίο είναι ψεκασμένο



Figure 1-6 Όψη ενός μη-βιολογικού χωραφίου στο χάρτη μας, το οποίο δεν έχει ψεκαστεί ακόμα

Η εφαρμογή δεν εμφανίζει τα χωράφια ως κουκίδες αλλά ως χρωματισμένα πολύγωνα κάτι το οποίο προσθέτει μια ωραία αίσθηση στην εμπειρία του χρήστη. **Όπως βλέπουμε η εφαρμογή καλύπτει στο έπακρο την συγκεκριμένη απαίτηση.**

1.3.2 Ικανοποίηση 2^{ης} απαίτησης

2^η απαίτηση → Δημιουργία λογαριασμού χρήστη (εργάτης ψεκασμών) με email και με μια μέθοδο εκ των δύο παρακάτω: Εγγραφή μέσω Facebook login SDK ή Google Identity Platform. Η σύνδεση να διατηρείται για ένα διάστημα ημέρων από την τελευταία χρήση.

Τρόπος ικανοποίησης → Η εφαρμογή καλύπτει στο μέγιστο την απαίτηση αυτή. Ο χρήστης μπορεί να εγγραφεί στην εφαρμογή χρησιμοποιώντας κάποιο προσωπικό του Mail ή με την βοήθεια της Google δηλαδή χρησιμοποιώντας κάποιο υπάρχον λογαριασμό που έχει σε αυτήν (...@gmail.com). Μπορεί αντίστοιχα αφού εγγραφεί στην εφαρμογή να συνδεθεί πάλι με τους παραπάνω τρόπους σε υπάρχον λογαριασμό.

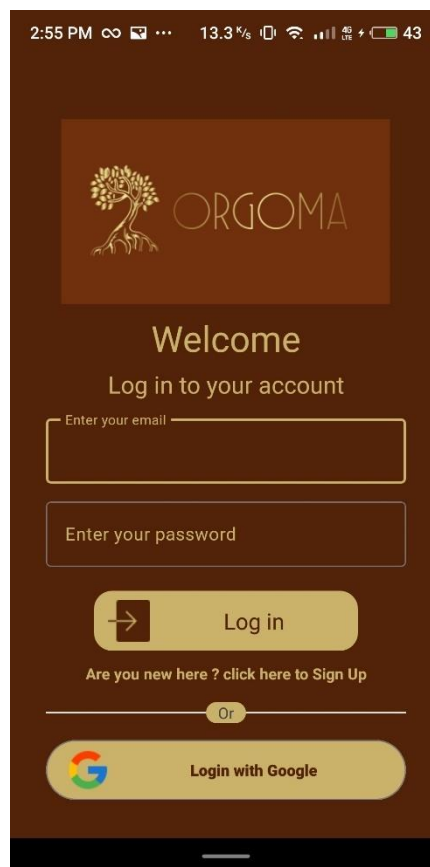


Figure 1-7 Όψη της Log in View της εφαρμογής

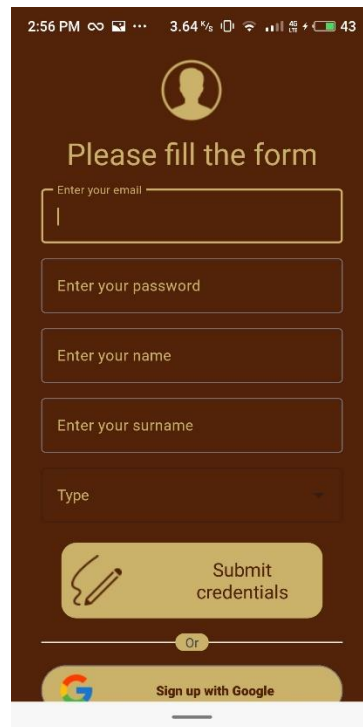


Figure 1-8 Όψη της Sing up View της εφαρμογής

Η παραπάνω απαίτηση επιτυγχάνεται με την βοήθεια του Firebase και συγκεκριμένα αξιοποιώντας της προσφερόμενες υπηρεσίες Authentication και Firestore Database. Φυσικά χρειαζόμαστε και το One tap Google Sign in για το Log in και Sign in με την Google.

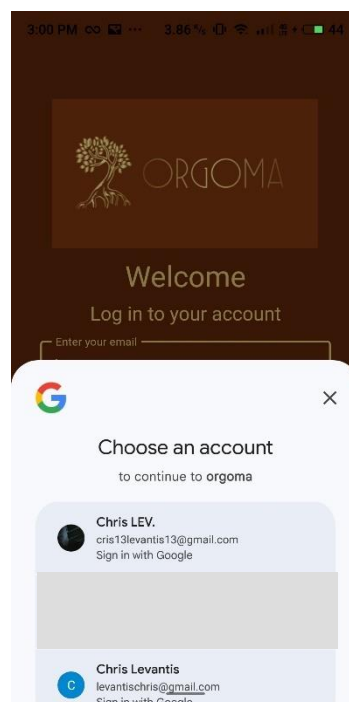


Figure 1-9 Όψη τους One tap log in της Google στην εφαρμογή μας

Να σημειώσουμε σε αυτό το σημείο ότι όπως έγινε αντιληπτό και σε προηγούμενα κεφάλαια η εφαρμογή διακρίνει δύο ρόλους τον Owner και τον Spray worker. Κατά την διαδικασία της εγγραφής στην εφαρμογή ο χρήστης συμπληρώνει το email του, το όνομα του, το επίθετο του και φυσικά το κωδικό του λογαριασμού του επίσης επιλέγει ποιος θα είναι ο ρόλος του, προτού η εφαρμογή τον εγγράψει σε αυτήν. Η επιλογή του ρόλου γίνεται και σε περίπτωση που ο χρήστης θέλει να εγγραφεί με προσωπικό του mail αλλά και σε περίπτωση που θέλει να εγγραφεί με λογαριασμό της Google απλά αν εγγραφεί με λογαριασμό της Google τότε το όνομα, επίθετο και email η εφαρμογή τα παίρνει αυτόματα από την Google.

Όπως ζητάει η απαίτηση η σύνδεση ενός χρήστη διατηρείται για ένα διάστημα ημερών με μόνη διαφορά ότι η εφαρμογή **μετρά την τελευταία φορά που ο χρήστης έκανε Log in στο λογαριασμό** και μόλις περάσουν κάποιες μέρες τότε τον κάνει αυτόματα Sign out, εκτός αν κάνει από μόνος του Sign out και μετά ξανά Log in τότε η εφαρμογή αρχίζει να ξαναμετρά. Στην βάση μας στο collection Users για κάθε document χρήστη κρατάμε ένα πεδίο το οποίο ονομάζεται logInDate που ουσιαστικά αποθηκεύει την ημερομηνία και ώρα που έκανε Log in στο account του ο χρήστης.

1.3.3 Ικανοποίηση 3^{ης} απαίτησης

3^η απαίτηση → Εισαγωγή ελαιώνα μέσω κατάλληλης φόρμας με δεδομένα τα στοιχεία του ιδιοκτήτη, τις ποικιλίες της ελιάς που περιέχει αν είναι βιολογικής καλλιέργειας ή όχι και γεωγραφικές συντεταγμένες του ελαιώνα που μπορεί να είναι μια ή περισσότερες κουκίδες ανάλογα με το μέγεθος του ελαιώνα μπορεί δηλαδή ο ιδιοκτήτης να μαρκάρει τις άκρες του ελαιώνα. Αυτό μπορεί να γίνει είτε με την χρήση geolocation του κινητού ή με marker που επιλέγει ο χρήστης πάνω στον χάρτη.

Τρόποι ικανοποίησης → Την συγκεκριμένη απαίτηση η εφαρμογή την καλύπτει σε καλό βαθμό. Ο χρήστης που μπορεί να εισάγει χωράφια στο χάρτη είναι μόνο ο Owner. Για να το κάνει αυτό ξεκινάει επιλέγοντας το κουμπί Mark Field που εμφανίζεται στην πάνω βοηθητική μπάρα (εικόνα 1-10).

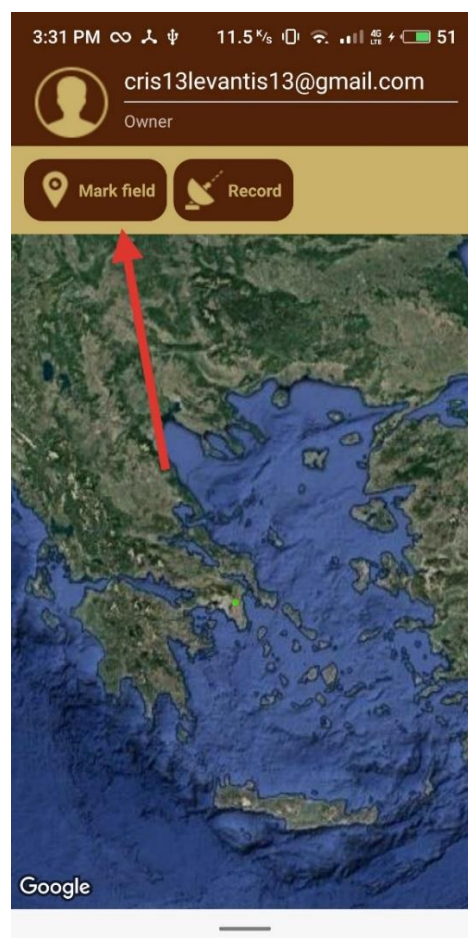


Figure 1-10 Κουμπί Mark field

Στη συνέχεια το UI της εφαρμογής θα ανανεωθεί όπως φαίνεται παρακάτω. Θα εμφανιστούν κάποια νέα κουμπιά στην βοηθητική μπάρα τα οποία θα βοηθήσουν τον χρήστη στην εισαγωγή των τεσσάρων marker ώστε να μαρκάρει το χωράφι που επιθυμεί. Επίσης θα εμφανιστεί στον χάρτη ένα icon που θα ακολουθάει την κάμερα του χρήστη, ο χρήστης μπορεί να το χρησιμοποιήσει για να «στοχεύσει» καλύτερα τα σημεία που επιθυμεί να βάλει τους pointers του χωραφιού.

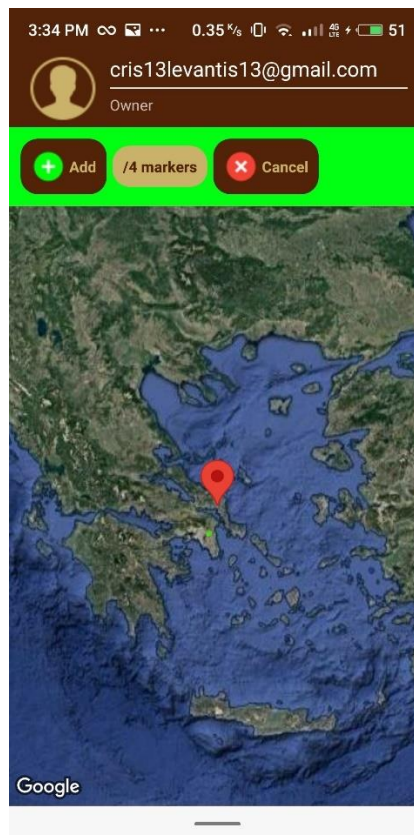


Figure 1-11 Ανανεωμένο UI της εφαρμογής μετά από πάτημα στο κουμπί Mark field

Αφού λοιπόν αλληλοεπιδράσει με τον χάρτη και οδηγηθεί στο σημείο που είναι το χωράφι του μπορεί να πατήσει το Add κουμπί για να τοποθετήσει το πρώτο marker, αυτό θα το κάνει τέσσερις φορές, δηλαδή θα βάλει τέσσερις markers που θα δηλώνουν τις γωνίες του χωραφιού.

Μόλις τοποθετήσει και τους τέσσερις και πατήσει το Add για το τελευταίο marker θα του εμφανιστεί αυτόματα ένα παράθυρο από το κάτω μέρος της οθόνης, στο οποίο θα εισάγει τα στοιχεία του χωραφιού. Φυσικά θα αλλάξει και όψη η πάνω μπάρα της εφαρμογής.

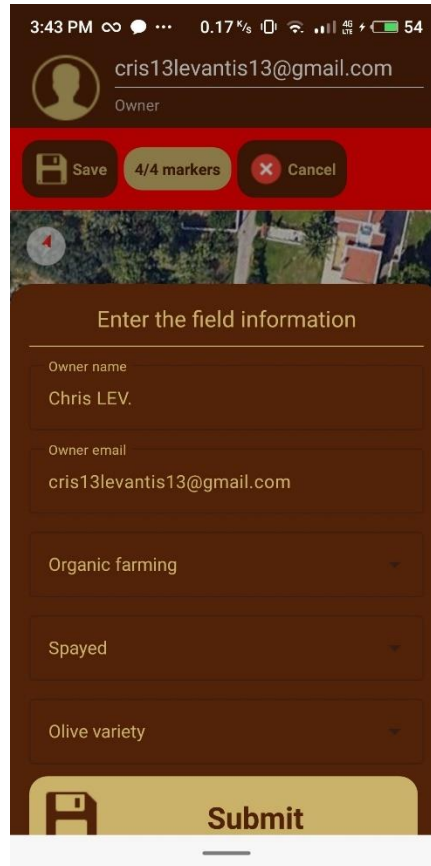


Figure 1-12 Εισαγωγή στοιχείων χωραφιού μετά από έγκυρη τοποθέτηση όλων των marker για αυτό

Βλέπουμε ότι αυτόματα έχουν εισαχθεί τα στοιχεία για το όνομα του ιδιοκτήτη του χωραφιού όπως και το email του. Αφού λοιπόν εισάγει και όλα τα υπόλοιπα πεδία θα μπορεί να πατήσει Submit για να καταχωρήσει το χωράφι στην εφαρμογή.

Αφού πατήσει το submit θα δημιουργηθεί το χωράφι μαζί με έναν χαρακτηριστικό ήχο και θα έχει έναν συγκεκριμένο χρωματισμό ανάλογα με το αν είναι οργανικό ή όχι ή αν είναι ψεκασμένο ή όχι (για του χρωματισμούς χωραφιών δεξ πρώτη απαίτηση).

Το χωράφι που ο χρήστης δημιούργησε θα αποθηκευτεί και στην Βάση στο collection Fields μαζί με κάποια attributes τα οποία είναι, cord_1 (1^η γωνία), cord_2 (2^η γωνία), cord_3

(3^η γωνία), cord_4 (4^η γωνία), creator_email, creator_name, olive_variety, organic_farming και sprayed.

Κλείνοντας να πούμε ότι η εφαρμογή φροντίζει να **μην υπάρχουν χωράφια τα οποία επικαλύπτουν άλλα ήδη υπάρχοντα χωράφια** όπως επίσης φροντίζει να **μην μπορεί ο χρήστης να τοποθετήσει markers του χωραφιού του μέσα σε άλλα ήδη υπάρχοντα χωράφια**. Φυσικά γίνονται και άλλοι πολύ έλεγχοι κατά την διάρκεια της αλληλεπίδρασης του χρήστη με την εφαρμογή.

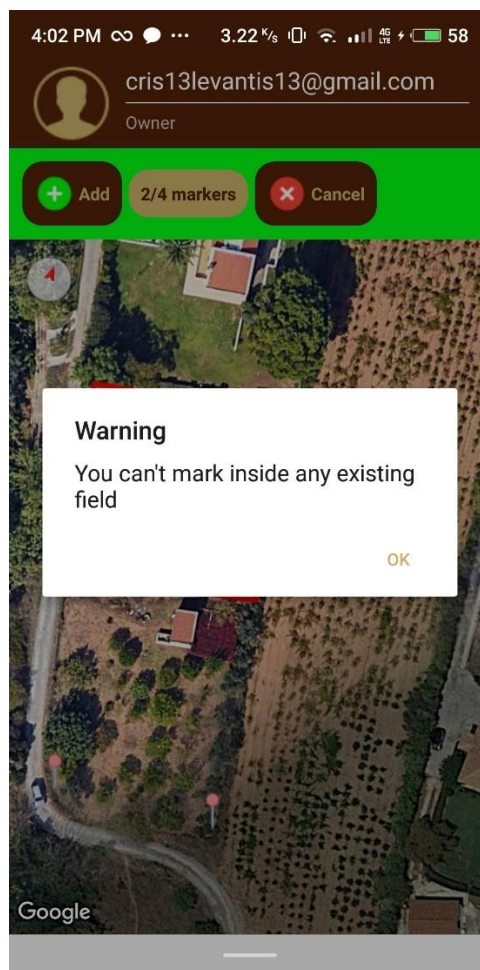


Figure 1-13 Μήνυμα απαγόρευσης τοποθέτησης marker μέσα σε άλλο ήδη υπάρχον χωράφι

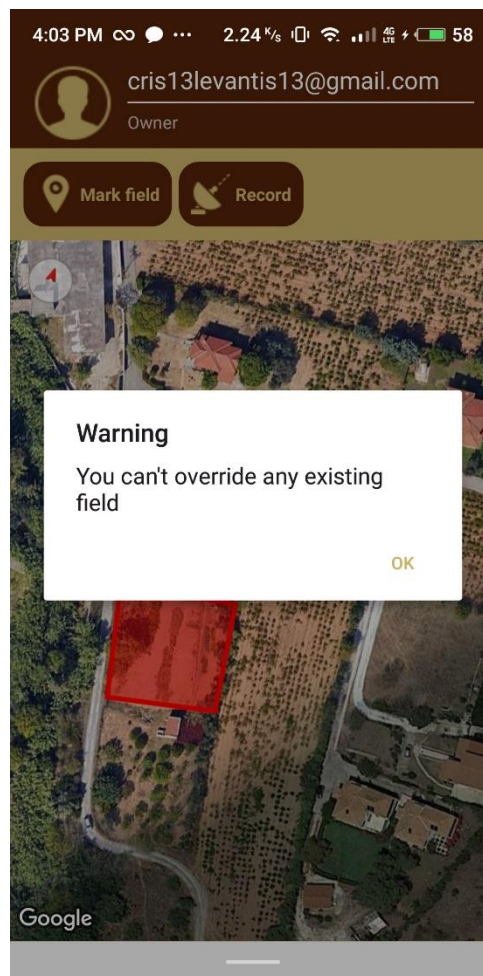


Figure 1-14 Μήνυμα προειδοποίησης ότι το χωράφι που εισήγαγε ο χρήστης επικαλύπτει άλλα ήδη υπάρχοντα χωράφια

1.3.4 Ικανοποίηση 4^{ης} απαίτησης

4^η απαίτηση → Δυνατότητα αποθήκευσης και επεξεργασίας ενός ελαιώνα

Τρόποι ικανοποίησης → Η εφαρμογή όπως επιγράφηκε και στην 3^η απαίτηση δίνει την δυνατότητα στον χρήστη να δημιουργήσει και να αποθηκεύσει στην εφαρμογή ένα χωράφι.

Η εφαρμογή επίσης δίνει την δυνατότητα στον χρήστη να διαγράψει ένα χωράφι, **αλλά μόνο αν του ανήκει**. Για να το κάνει αυτό μπορεί να πραγματοποιήσει ένα παρατεταμένο κλικ επάνω σε ένα χωράφι (δηλαδή μέσα στην μαρκαρισμένη περιοχή του χωραφιού) και στην συνέχεια η εφαρμογή θα εξετάσει αν του ανήκει. Αν του ανήκει θα του δώσει την δυνατότητα να επιλέξει αν θέλει να διαγράψει το χωράφι από τον χάρτη και φυσικά και από την εφαρμογή (από την βάση). Στην συνέχεια η εφαρμογή θα κάνει reload τον χάρτη ώστε να φανούν άμεσα στον χρήστη οι διαφορές.

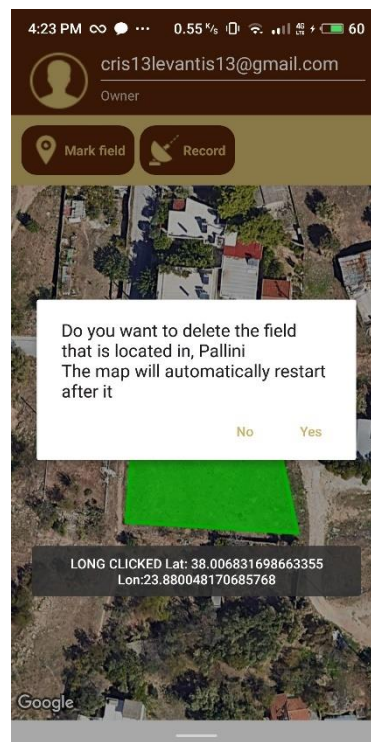


Figure 1-15 Μήνυμα διαγραφής χωραφιού από τον χάρτη

Φυσικά ο μόνος χρήστης που μπορεί να κάνει διαγραφή χωραφιού είναι μόνο ένας Owner.

1.3.5 Ικανοποίηση 5^{ης} και 6^{ης} απαίτησης

5^η απαίτηση → Σε χρήστες συνδεδεμένους ως εργάτες ψεκασμού να καταγράφει την διαδρομή και κάθε φορά που πλησιάζει έναν βιολογικό ελαιώνα στα 20 μέτρα απόσταση, να λαμβάνει στο κινητό μια ειδοποίηση (notification) ότι πλησιάζει βιολογικό ελαιώνα.

6^η απαίτηση → Στους υπόλοιπους συμβατικούς ελαιώνες (δηλαδή μη βιολογικούς), όταν πλησιάζει 10 μέτρα από σημείο (marker) που ανήκει σε κάποιον ελαιώνα να του αλλάζει την κατάσταση σε ψεκασμένο.

Τρόποι ικανοποίησης → Η εφαρμογή δίνει την δυνατότητα σε χρήστες με εξουσιοδότηση ως Spray workers αλλά και σε Owners⁴ να αρχίσουν την καταγραφή της διαδρομής τους. Φυσικά η λειτουργία Record σχετίζεται άμεσα με την πραγματική διαδικασία ψεκασμού ενός χωραφιού ή και χωραφιών, δηλαδή όταν κάποιος εργάτης αρχίζει να ψεκάζει θα πρέπει να χρησιμοποιήσει την λειτουργία Record ώστε να καταγράφεται η διαδρομή του και να λαμβάνει πολύ σημαντικές πληροφορίες που θα τον βοηθήσουν στην δουλεία του.

Πιο συγκεκριμένα η εφαρμογή με την λειτουργία Record μπορεί να ειδοποιεί τον χρήστη όταν βρίσκεται στα 20 μέτρα από έναν βιολογικό ελαιώνα. Η εφαρμογή μόλις εντοπίσει ότι η θέση του χρήστη, δηλαδή οι συντεταγμένες του είναι κοντά σε έναν βιολογικό ελαιώνα θα παράγει έναν χαρακτηριστικό ήχο ειδοποίησης αλλά και θα σταματήσει αυτόματα την καταγραφή της διαδρομής του χρήστη καθώς θα έχει πλησιάσει επικίνδυνα κοντά σε κάποιο χωράφι βιολογικής καλλιέργειας. Επίσης η εφαρμογή θα παράγει μια ειδοποίηση (notification) η οποία θα ενημερώνει με ένα κατάλληλο μήνυμα το χρήστη.

Η εφαρμογή μπορεί επίσης με την λειτουργία Record να ενημερώνει σε πραγματικό χρόνο τις πληροφορίες ενός χωραφιού και συγκεκριμένα το αν είναι ψεκασμένο ή όχι. Αυτό

⁴ κανονικά δεν θα έπρεπε αλλά στα πλαίσια της δοκιμής την έδωσα την δυνατότητα και σε Owners

θα γίνει μόνο αν η θέση του χρήστη είναι 10 μέτρα από ένα μη-βιολογικό και μη ψερασμένο χωράφι. Μόλις ο χρήστης πλησιάσει στα 10 μέτρα το χωράφι αυτό αυτόματα θα αλλάξει χρωματισμό και θα ενημερωθεί στην βάση το αντίστοιχο Field του Collection Fields στο πεδίο sprayed (από No θα γίνει Yes).

Για να γίνει πιο κατανοητό από άποψη υλοποίησης η εφαρμογή στην αρχή της δημιουργίας του MainActivity δηλαδή την στιγμή που δημιουργεί τον χάρτη **παίρνει από την βάση όλα τα χωράφια που υπάρχουν στην εφαρμογή και τα βάζει σε μια λίστα**. Με βάση τώρα αυτή την λίστα **κάθε φορά που αλλάζει το στίγμα του χρήστη**, δηλαδή κάθε φορά που καλείται η συνάρτηση **onMyLocationChange(...)** η εφαρμογή τρέχει την συνάρτηση **calculateDistance(...)** η οποία **υπολογίζει την απόσταση από την θέση του χρήστη σε σχέση με κάθε marker που έχει κάθε χωράφι της εφαρμογής**, και ουσιαστικά μόλις εντοπίσει κάποια απόσταση κάτω από κάποιο όριο (10 ή 20 μέτρα) δείχνει τα κατάλληλα μηνύματα στον χρήστη ή και κάνει τις υπόλοιπες κατάλληλες ενέργειες εσωτερικά στην εφαρμογή.

Για καλύτερη κατανόηση παρακαλώ δείτε τα βίντεο (δες κεφάλαιο 3).

1.3.6 Ικανοποίηση 7^{ης} απαίτησης

7^η απαίτηση → Κεντρική βάση δεδομένων (δηλαδή απομακρυσμένη, εκτός κινητού) η οποία να περιέχει όλες τις πληροφορίες που προκύπτουν από τις παραπάνω απαιτήσεις.

Τρόποι ικανοποίησης → Σαν απομακρυσμένη βάση δεδομένων χρησιμοποιώ το Firestore της Firebase. Η βάση αποτελείται μόνο από **δύο collections το Users και το Fields** τα οποία είναι πολύ απλοϊκά δηλαδή το μόνο που έχουν είναι κάποια attributes τα οποία φαίνονται στις παρακάτω εικόνες.

Δυστυχώς η σχεδίαση της βάσης δεδομένων που έχω κάνει δεν είναι καλή και ούτε είναι σωστή, καθώς πολλά attributes τα επαναλαμβάνω ενώ θα μπορούσα να τα είχα οργανώσει καλύτερα χρησιμοποιώντας αναφορές. Παρόλα αυτά σαν πρώτη γνωριμία με μια No-SQL βάση ήταν πολύ ωραία και σίγουρα στο μέλλον θα την ξανά-επέλεγα για τα project μου.

```
{  
  
  UDI: "feD6ugwmeDMdLBZLR1Tq14P8uVQ2"  
  
  email: "test@mail.com"  
  
  logInDate: "2023-12-23 15:31:00"  
  
  name: "Eleni"  
  
  surname: "Elenaki"  
  
  type: "Owner"  
  
}
```

Figure 1-16 Δομή ενός JSON αρχείου document του collection Users

Όπως βλέπουμε η δομή του ενός document αρχείου του collection Users είναι αρκετά απλή, να σημειώσουμε όμως ότι το πεδίο UID αντιστοιχεί στο ID που παίρνουμε από το Authentication (Δες εικόνα 1-17). Όλα τα πεδία είναι τύπου String.



Figure 1-17 Αντίστοιχο πεδίο στο Authentication (σχετικά με την εικόνα 1-16)

Περνάμε στο collection Fields το οποίο έχει την δομή που φαίνεται παρακάτω.

```
{
  cord_1:
    {
      Latitude: "38.02483153728906"
      Longitude: "23.900504037737846"
    }
  cord_2:
    {
      Latitude: "38.02459859039119"
      Longitude: "23.90063278377056"
    }
  cord_3:
    {
      Latitude: "38.02471612038122"
      Longitude: "23.90108473598957"
    }
  cord_4:
    {
      Latitude: "38.024957518466096"
      Longitude: "23.90093319118023"
    }
  creator_email: cris13levantis13@gmail.com
  creator_name: "Chris Leventis"
  olive_variety: "Gaeta"
  organic_farming: "Yes"
  sprayed: "No"
}
```

Figure 1-18 Δομή ενός JSON αρχείου document του collection Fields

Βλέπουμε ότι πάλι έχουμε απλή δομή του JSON αρχείου. Πάλι όλα τα στοιχεία είναι τύπου String, εκτός από τις συντεταγμένες οι οποίες είναι τύπου GeoPoint για αυτό και έχουμε τα υπό-πεδία Latitude και Longitude.

2 Οδηγίες εγκατάστασης

Αμα η εφαρμογή ήταν ανεβασμένη στο Google play store τότε από εκεί θα μπορούσε κάποιος να κατεβάσει και να χρησιμοποιήσει την εφαρμογή κατευθείαν. Στα πλαίσια όμως της δοκιμής της εφαρμογής τα πράγματα είναι διαφορετικά.

2.1 Εγκατάσταση από το GitHub

Αρχικά το project είναι διαθέσιμο στο GitHub οπότε θα πρέπει να γίνει ένα κατέβασμα του source κώδικα στον τοπικό υπολογιστή. Στην συνέχεια ο χρήστης θα πρέπει να κατευθυνθεί στον φάκελο app και εκεί θα εντοπίσει ένα αρχείο τύπου apk με το όνομα orgoma. Αυτό το αρχείο μπορεί στην συνέχεια να το τρέξει με τον emulator που έχει στο σύστημα του εγκατεστημένο. Για παράδειγμα στην περίπτωση που έχει εγκαταστημένο τον emulator BlueStacks το μόνο που πρέπει να κάνει είναι να διπλοπατήσει πάνω στο αρχείο και αυτό αυτόματα θα ανοίξει στον emulator.

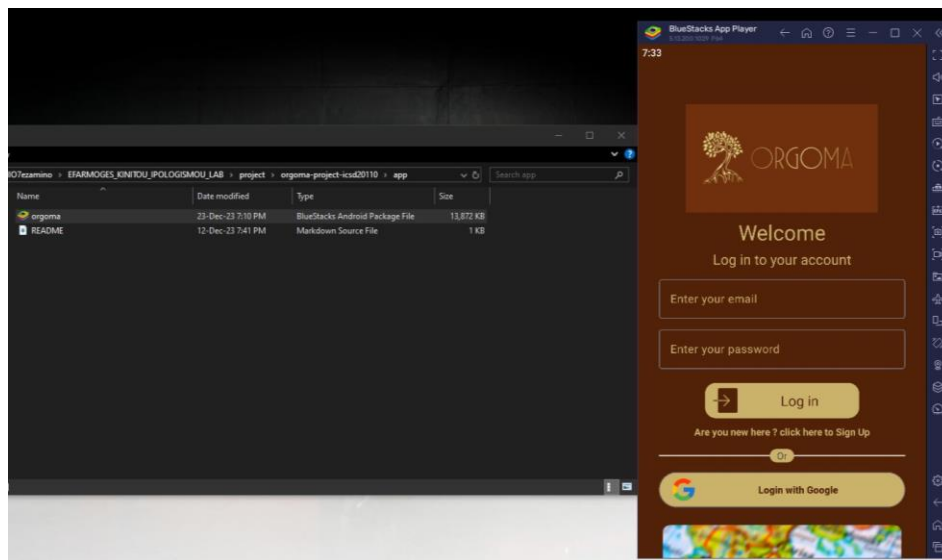


Figure 2-1 BlueStacks emulator

3 Χαρακτηριστικά βίντεο σχετικά με την εφαρμογή

Για την επίδειξη της εφαρμογής έχουν δημιουργηθεί δύο βίντεο. Στον πρώτο βίντεο (https://youtu.be/B_7Aa6_7Rcc) γίνεται παρουσίαση της εφαρμογής σε Tablet χωρίς να φαίνονται οι αλλαγές στην βάση. Ουσιαστικά στο βίντεο αυτό γίνεται μια πραγματική χρήση της εφαρμογής σε εξωτερικό χώρο, ο χρήστης μπορεί να δει ξεκάθαρα την λειτουργία της Record αλλά και το τρόπο που η εφαρμογή μας υποστηρίζει το Google Sing in/Log in, αλλά φυσικά και πολλά άλλα. Στο δεύτερο βίντεο (https://youtu.be/Ij_Zi1Wo5qU) γίνεται μια παρόμοια περίπου παρουσίαση αλλά αυτήν την φορά σε Smartphone με μεγαλύτερη βάση στις αλλαγές που γίνονται στην βάση, καθώς παρέχονται και χαρακτηριστικά screenshots. Σε αυτό το βίντεο φαίνεται και πως η εφαρμογή μας χειρίζεται το Sing in και Log in χωρίς την χρήση Google λογαριασμού, αλλά φυσικά και πολλά άλλα.

4 Κατάλογος σχημάτων

Figure 1-1 Εικόνα της βάσης δεδομένων.....	4
Figure 1-2 Basic user activities, Use case.....	5
Figure 1-3Επιλογή δημόσιου χάρτη.....	14
Figure 1-4 Όψη ενός βιολογικού χωραφιού στον χάρτη της εφαρμογής.....	15
Figure 1-5 Όψη ενός μη βιολογικού χωραφιού στον χάρτη της εφαρμογής μας, το οποίο είναι ψεκασμένο	15
Figure 1-6 Όψη ενός μη-βιολογικού χωραφιού στο χάρτη μας, το οποίο δεν έχει ψεκαστεί ακόμα	16
Figure 1-7 Όψη της Log in View της εφαρμογής	17
Figure 1-8 Όψη της Sing up View της εφαρμογής	18
Figure 1-9 Όψη τους One tab log in της Google στην εφαρμογή μας.....	18
Figure 1-10 Κουμπί Mark field.....	20
Figure 1-11 Ανανεωμένο UI της εφαρμογής μετά από πάτημα στο κουμπί Mark field	21
Figure 1-12 Εισαγωγή στοιχείων χωραφιού μετά από έγκυρη τοποθέτηση όλων των marker για αυτό	22
Figure 1-13 Μήνυμα απαγόρευσης τοποθέτησης marker μέσα σε άλλο ήδη υπάρχον χωράφι	23
Figure 1-14 Μήνυμα προειδοποίησης ότι το χωράφι που εισήγαγε ο χρήστης επικαλύπτει άλλα ήδη υπάρχοντα χωράφια	24
Figure 1-15 Μήνυμα διαγραφής χωραφιού από τον χάρτη	25
Figure 1-16 Δομή ενός JSON αρχείου document του collection Users	28
Figure 1-17Αντίστοιχο πεδίο στο Authentication (σχετικά με την εικόνα 1-16).....	29
Figure 1-18 Δομή ενός JSON αρχείου document του collection Fields.....	29
Figure 2-1 BlueStacks emulator.....	31

