



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής - Ανάπτυξη Λογισμικού και
Τεχνητής Νοημοσύνης»**

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή Android για την εύρεση χαμένων κατοικίδιων και την καταγραφή τοποθεσιών ζώων Android application for finding lost pets and documenting the location of animals
Όνοματεπώνυμο Φοιτητή	Τεμουρτζίδης Κωνσταντίνος
Πατρώνυμο	Νικόλαος
Αριθμός Μητρώου	ΜΠΣΠ2228
Επιβλέπων	Ευθύμιος Αλέπης, Καθηγητής

Ημερομηνία Παράδοσης **Σεπτέμβριος 2024**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευάγγελος Σακκόπουλος
Αναπληρωτής Καθηγητής

Περιεχόμενα

Περίληψη	4
Abstract	4
1. Εισαγωγή	5
2. Ανασκόπηση Πεδίου.....	6
2.1. PawBoost	6
2.2. Missing Pet Finder.....	6
2.3. PetFBI	6
3. Τεχνολογίες	6
3.1. Jetpack Compose	6
3.2. Firebase.....	7
3.3. Spring Boot	8
3.4. Tensorflow.....	9
4. Παρουσίαση Εφαρμογής	12
5. Συμπεράσματα Και Μελλοντικές Επεκτάσεις	43
6. Βιβλιογραφία	44

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή παρουσιάζει μια εφαρμογή Android γραμμένη στην γλώσσα προγραμματισμού Kotlin, η οποία επιτρέπει στους χρήστες της να δημιουργούν δημοσιεύσεις σχετικά με την απώλεια του κατοικίδιου τους ή την εύρεση κάποιου ζώου στον δρόμο με σκοπό την ενημέρωση άλλων χρηστών της εφαρμογής που βρίσκονται κοντά στην περιοχή που όρισε ο χρήστης που δημιούργησε την δημοσίευση. Στην εφαρμογή έχουν επίσης πρόσβαση μέλη των τοπικών αρχών τα οποία μπορούν να ενημερώσουν τις δημοσιεύσεις των χρηστών σχετικά με το πως έχουν ενεργήσει για την εύρεση ή την απομάκρυνση κάποιου ζώου από την περιοχή στην οποία βρίσκεται. Για τις διάφορες λειτουργίες της, η εφαρμογή επικοινωνεί με ένα Spring Boot API γραμμένο σε Java, ένα Python API φτιαγμένο με την βοήθεια της βιβλιοθήκης FastAPI και με την πλατφόρμα Firebase της Google.

Abstract

This master's thesis presents an Android application, written using the Kotlin programming language, which allows its users to create posts about the loss of their pets or about the encounter of an animal in the street, with the goal of notifying other users around the area the indicated by the poster. Members of local government also have access to the application, allowing them to update the status of a post regarding the steps that have been taken to find or relocate an animal. The functionality of the application is supported by a Spring Boot API written using Java, a Python API written using the FastAPI library and the Firebase platform made by Google.

1. Εισαγωγή

Τα ζώα είναι ένα μεγάλο μέρος της καθημερινής ζωής πολλών ανθρώπων καθώς σχεδόν όλοι μας αλληλοεπιδρούμε με αυτά με διάφορους τρόπους, είτε με τα κατοικίδια μας είτε με αδέσποτα ή οποιοδήποτε άλλο ζώο μπορεί να συναντήσουμε στον δρόμο και στην εξοχή, φιλικό ή άγριο.

Το αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη μιας εφαρμογής Android η οποία επιτρέπει στους χρήστες της να δημιουργούν δυο είδους δημοσιεύσεις. Το πρώτο είδος αφορά αγνοούμενα κατοικίδια για τα οποία η εφαρμογή επιτρέπει στους χρήστες της να δημιουργούν δημοσιεύσεις για να ενημερώσουν άλλους χρήστες της εφαρμογής με στόχο αυτοί, εάν θελήσουν ή εάν έχουν κάποια πληροφορία, να βοηθήσουν στην εύρεση του κατοικίδιου αυτού. Το δεύτερο είδος είναι σχετικά με ζώα που οι χρήστες συναντάνε στον δρόμο και πιστεύουν ότι, είτε επειδή μπορεί να είναι το κατοικίδιο κάποιου ή επειδή μπορεί να είναι επιθετικό και να τραυματίσει κάποιον, πρέπει να ενημερώσουν τους συνανθρώπους τους για αυτά. Και στις δυο περιπτώσεις οι χρήστες που βρίσκονται κοντά στην περιοχή που όρισε ο χρήστης που δημιούργησε την δημοσίευση, και έχουν ενεργοποιήσει την λήψη ειδοποιήσεων, θα λάβουν μια ειδοποίηση η οποία τους ενημερώνει για αυτήν και τους προτρέπει να ανοίξουν την εφαρμογή για να λάβουν περισσότερες πληροφορίες εάν το θελήσουν. Όλοι οι χρήστες της εφαρμογής που βρίσκονται στην ίδια χώρα με τον χρήστη που δημιούργησε την δημοσίευση θα μπορέσουν να την δούνε στην ροή δημοσιεύσεων μέσα στην εφαρμογή, η οποία τους επιτρέπει να περιορίσουν τις δημοσιεύσεις μέσω της χρήσης φίλτρων όπως η περιοχή ή το είδος του ζώου.

Σημαντικό ρόλο στην διαχείριση των ζώων μπορούν να έχουν οι τοπικές αρχές. Για τον λόγο αυτό η εφαρμογή δίνει την δυνατότητα σε μέλη των αρχών αυτών να έχουν πρόσβαση στην εφαρμογή η οποία τους επιτρέπει να ενημερώνουν τις δημοσιεύσεις για τα διάφορα βήματα που έχουν ακολουθήσει είτε για να βοηθήσουν στο να βρεθεί κάποιο κατοικίδιο που έχει χαθεί ή για να απομακρύνουν κάποιο πιθανώς επικίνδυνο ζώο που έχει βρεθεί σε κατοικημένη περιοχή. Η λειτουργικότητα αυτή υπάρχει έτσι ώστε οι χρήστες να μπορούν να ενημερωθούν για την κατάσταση του ζώου στο οποίο αναφέρετε η δημοσίευση, δηλαδή εάν ένα πιθανώς επικίνδυνο ζώο έχει απομακρυνθεί και η περιοχή είναι ασφαλής ή εάν ένα κατοικίδιο έχει βρεθεί από τις τοπικές αρχές και δεν υπάρχει λόγος να το αναζητούν και εκείνοι.

2. Ανασκόπηση Πεδίου

Στον τομέα της εύρεσης ζώων και κατοικίδιων έχουν αναπτυχθεί διάφορες εφαρμογές για να βοηθήσουν τους ανθρώπους. Σε αυτό το σημείο θα παρουσιάσουμε μερικές από αυτές που παρουσιάζουν την μεγαλύτερη ομοιότητα με την εφαρμογή που έχουμε υλοποιήσει στα πλαίσια αυτής της μεταπτυχιακής διατριβής.

2.1. PawBoost

Το PawBoost επιτρέπει στο χρήστες του, μέσω της ιστοσελίδας ή της εφαρμογής για συσκευές iOS και Android που έχει, να αναζητούνε τα χαμένα κατοικίδια τους και να βοηθάνε άλλους χρήστες να κάνουν το ίδιο. Σύμφωνα με την ιστοσελίδα του PawBoost, υπάρχουν πάνω από 7 εκατομμύρια εγγεγραμμένοι χρήστες παγκοσμίως, ενώ η εφαρμογή έχει συνεισφέρει στην εύρεση παραπάνω από 1,6 εκατομμυρίων κατοικίδιων. Όταν ένας χρήστης δημιουργεί μία δημοσίευση στο PawBoost γίνονται αυτόματα τα εξής:

- Δημιουργείται ένα δημοσίευμα στην ομάδα Facebook του PawBoost για την περιοχή του χρήστη που περιέχει τις πληροφορίες που παρείχε ο χρήστης.
- Στέλνονται ειδοποιήσεις μέσω email και μέσω τις εφαρμογής σε όσους χρήστες του PawBoost τις έχουν ενεργοποιήσει.
- Δημιουργείται ένα φυλλάδιο για το κατοικίδιο το οποίο ο χρήστης μπορεί να κατεβάσει και να εκτυπώσει, με στόχο να το μοιράσει στην περιοχή του.

Ιστοσελίδα: <https://www.pawboost.com/>

2.2. Missing Pet Finder

Το Missing Pet Finder είναι μια εφαρμογή διαθέσιμη για συσκευές iOS η οποία επιτρέπει στους χρήστες της να δημιουργούνε προφίλ για τα κατοικίδια τους, έτσι ώστε στην περίπτωση που κάποιο από αυτά χαθεί να μπορούνε αμέσως να το μοιραστούνε με άλλους χρήστες της εφαρμογής. Οι χρήστες στην ίδια περιοχή με τον χρήστη που έχασε το κατοικίδιο του θα λάβουν άμεσα μια ειδοποίηση για το γεγονός αυτό και μπορούνε να επικοινωνήσουν κατευθείαν με τον χρήστη μέσω της εφαρμογής εάν έχουν κάποια σχετική πληροφορία. Η εφαρμογή επιτρέπει επίσης στους χρήστες να εκτυπώνουν φυλλάδια τα οποία έχουν δημιουργηθεί αυτόματα για το κατοικίδιο που χάσανε.

Ιστοσελίδα: <https://missingpet.app/>

2.3. PetFBI

Το PetFBI είναι μια βάση δεδομένων που περιέχει δημοσιεύσεις από χρήστες από τις Ηνωμένες Πολιτείες σχετικά με τα χαμένα τους κατοικίδια ή για κατοικίδια που έχουν δει στον δρόμο. Ένας χρήστης μπορεί να δημιουργήσει μια δημοσίευση για ένα ζώο εισάγοντας το είδος του, το όνομα του, μια διεύθυνση κοντά στο σημείο που χάθηκε ή που βρέθηκε και κάποιες πληροφορίες σχετικά με το ζώο και στην συνέχεια οποιαδήποτε χρήστης μπορεί να αναζητήσει στις δημοσιεύσεις της βάσης αυτής είτε για δημοσιεύσεις κοντά στην τοποθεσία του ή σε οποιαδήποτε περιοχή της χώρας.

Ιστοσελίδα: <https://petfbi.org/>

3. Τεχνολογίες

3.1. Jetpack Compose

Το Jetpack Compose είναι ένα μοντέρνο UI framework με βάση την γλώσσα προγραμματισμού Kotlin, το οποίο χρησιμοποιείται για την δημιουργία εφαρμογών για το λειτουργικό σύστημα Android. Σε αντίθεση με την παλαιότερη μέθοδο ανάπτυξης εφαρμογών Android, δηλαδή την

Εφαρμογή Android για την εύρεση χαμένων κατοικίδιων και την καταγραφή τοποθεσιών ζώων 6

χρήση Activity ή Fragment για την λειτουργικότητα και XML για το UI, το Jetpack Compose λειτουργεί με βάση τον δηλωτικό προγραμματισμό, ένα προγραμματιστικό πρότυπο στο οποίο το πρόγραμμα ορίζεται από τις ιδιότητες του αντί να περιγράφουμε τον στόχο ως μια αλληλουχία βημάτων. Για την δημιουργία της εφαρμογής χρησιμοποιούνται μεθόδους που ονομάζονται Composables οι οποίες συγκεντρώνουν την λειτουργικότητα και τον σχεδιασμό της εφαρμογής σε ένα σημείο.

Μια πολύ σημαντική λειτουργία που προσφέρει η Kotlin είναι τα Coroutines. Μέσω αυτών έχουμε την δυνατότητα να εκτελούμε κώδικα παράλληλα με τον κώδικα που είναι υπεύθυνος για το UI, εύκολα από οποιοδήποτε σημείο χωρίς να χρειάζεται να χρησιμοποιήσουμε κάτι αντίστοιχο των Threads που χρησιμοποιούνται συνήθως στην Java, τα οποία χρειάζονται αρκετό κώδικα για να υλοποιηθούν και είναι δύσκολο να επαναχρησιμοποιηθούν. Μια πολύ σημαντική λειτουργικότητα που προσφέρουν τα Coroutines είναι η αλληλεπίδραση τους με ένα νέο είδος μεθόδων που ονομάζονται suspend functions. Οι μέθοδοι αυτές μπορούν να εκτελεστούν μόνο μέσα σε Coroutines και όταν εκτελούνται σταματάνε την εκτέλεση του Coroutine, ελευθερώνοντας έτσι το thread στο οποίο εκτελούνταν μέχρι στιγμής ο κώδικας του, μέχρις ότου να ολοκληρωθεί η εκτέλεση τους. Η λειτουργικότητα αυτή είναι ιδιαίτερα χρήσιμη όταν εκτελούμε αιτήματα δικτύου και θέλουμε να σταματήσουμε την εκτέλεση του κώδικα μέχρι να μας έρθει η απάντηση από το διαδίκτυο.

Η κύρια δυνατότητα που μας προσφέρει το Jetpack Compose είναι να υλοποιήσουμε την λειτουργικότητα και την εμφάνιση της εφαρμογής μας συνδυάζοντας διάφορα Composables, είτε φτιαγμένα από εμάς ή έτοιμα από διάφορες βιβλιοθήκες, με στόχο να δομήσουμε την εφαρμογή μας. Πολλά βασικά Composables προσφέρονται απευθείας από το framework, όπως για παράδειγμα το Text, το Button, το TextField, το Icon κλπ., και έχουμε την δυνατότητα να τα παραμετροποιήσουμε είτε χρησιμοποιώντας τις παραμέτρους των μεθόδων τους ή, εάν θέλουμε να κάνουμε κάποια μεγαλύτερη αλλαγή, χρησιμοποιώντας τα Modifiers. Modifier έχουν όλα τα Composables τα οποία προσφέρονται από το framework και μας επιτρέπουν να κάνουμε αλλαγές που επηρεάζουν την εμφάνιση και την λειτουργικότητα τους, όπως να αλλάζουμε το μέγεθος ή το χρώμα τους ή ακόμα και να δίνουμε την δυνατότητα στον χρήστη να πατήσει πάνω σε ένα Composable το οποίο δεν έχει δημιουργηθεί με αυτήν την λειτουργικότητα. Όλα αυτά μας βοηθάνε στην υλοποίηση της εφαρμογής μας αφού χρειάζεται να γράψουμε λιγότερο κώδικα, καθώς μπορούμε να επαναχρησιμοποιήσουμε πολλά Composables με διαφορετικές παραμέτρους σε διάφορα σημεία της εφαρμογής, κάτι που μειώνει σημαντικά τον χρόνο που θα χρειαστούμε για να ολοκληρώσουμε την εφαρμογή.

Όπως έχουμε αναφέρει, το Jetpack Compose λειτουργεί με functional programming. Για τον λόγο αυτό χρειαζόμαστε έναν τρόπο για να μπορούμε δυναμικά να αλλάζουμε τα δεδομένα που προβάλλουν τα Composables όσο αλληλοεπιδρά ο χρήστης με την εφαρμογή μας. Αυτό επιτυγχάνεται μέσω δυο λειτουργιών. Η πρώτη ονομάζεται Recomposition και είναι η δυνατότητα των Composable Functions να επανεκτελούνται αυτόματα όταν μια μεταβλητή που χρησιμοποιούν αλλάζει, ενημερώνοντας έτσι το UI για αυτήν την αλλαγή. Η επανεκτέλεση αυτή είναι τοπική, κάτι που σημαίνει ότι εκτελείται ξανά μόνο η μέθοδος που περιέχει την μεταβλητή αυτή και όχι όλες οι μέθοδοι που συγκροτούν την οθόνη που βλέπει ο χρήστης εκείνη την στιγμή. Η δεύτερη λειτουργία που χρησιμοποιούμε για να ενημερώνουμε δυναμικά το UI είναι τα View Models. Στα View Models περιέχονται όλες οι μεταβλητές που ενημερώνονται σε ένα Composable καθώς και ο κώδικας που είναι υπεύθυνος για την ενημέρωση αυτή. Τα View Models προσφέρουν την δυνατότητα στα Composables να ενημερώνουν τις μεταβλητές που περιέχουν και τα ενημερώνονται όταν κάποια από αυτές τις μεταβλητές αλλάζει έτσι ώστε να ενημερωθεί το UI.

3.2. Firebase

Μια από τις backend πλατφόρμες με τις οποίες επικοινωνεί η εφαρμογή είναι η Firebase της Google. Η Firebase είναι μια πλατφόρμα cloud computing η οποία μας επιτρέπει να έχουμε πρόσβαση σε ένα μεγάλο εύρος διαφορετικών υπηρεσιών που μπορεί να χρειαστούμε για την εφαρμογή μας. Μερικές από τις σημαντικότερες εξ αυτών είναι οι:

- **Authentication:** Μας προσφέρει ένα σύστημα αυθεντικοποίησης χρηστών με διάφορες μεθόδους όπως η χρήση Email/Password ή η σύνδεση μέσω λογαριασμού κάποιας πλατφόρμας όπως Google, Facebook κλπ.
- **Realtime Database:** Μια μη σχεσιακή βάση δεδομένων στην οποία μπορούμε να αποθηκεύσουμε και να ανακτήσουμε δεδομένα κατευθείαν από την εφαρμογή μας. Η βάση αυτή δεν χρειάζεται καμία παραμετροποίηση πινάκων ή μεταβλητών εκ των προτέρων κάτι που κάνει την χρήση της πολύ πιο εύκολη σε νέους χρήστες
- **Cloud Storage:** Μια υπηρεσία αποθήκευσης αρχείων η οποία διευκολύνει την μεταφόρτωση και πρόσβαση σε αρχεία τα οποία χρειάζονται για την εφαρμογή, είτε από τους διαχειριστές της ή από τους χρήστες.
- **Machine Learning:** Μας δίνει την δυνατότητα να τρέξουμε είτε δικά μας ή έτοιμα μοντέλα μηχανικής μάθησης, είτε στο cloud στέλνοντας μετά το αποτέλεσμα στην συσκευή του χρήστη ή στην συσκευή του χρήστη όταν υπάρχει αυτή η δυνατότητα
- **Cloud Functions:** Μας επιτρέπει να τρέξουμε backend κώδικα όταν συμβαίνουν κάποια events χωρίς να χρειάζεται να έχουμε κάποιο ξεχωριστό backend σε δικό μας server.
- **Cloud Messaging:** Μια υπηρεσία μέσω της οποίας μπορούμε να στέλνουμε ειδοποιήσεις στις συσκευές των χρηστών.
- **Analytics:** Παρέχει στους προγραμματιστές της εφαρμογής πληροφορίες σχετικά με την χρήση της, όπως συμβάντα ή προβλήματα, με στόχο την βελτίωση της εφαρμογής με την χρήση των δεδομένων αυτών.

Όλες αυτές οι υπηρεσίες μπορούνε, αφού τις ενεργοποιήσουμε στην κονσόλα της Firebase, να χρησιμοποιηθούνε κατευθείαν μέσω της εφαρμογής μας, μέσω των βιβλιοθηκών που αντιστοιχούν σε κάθε μια. Οι βιβλιοθήκες αυτές είναι διαθέσιμες για διάφορες γλώσσες προγραμματισμού κάτι που μας επιτρέπει να χρησιμοποιούμε την Firebase ως backend σε διάφορα είδη εφαρμογών.

Για την εφαρμογή της παρούσας διατριβής έχουν χρησιμοποιηθεί οι υπηρεσίες Storage, Authentication και Cloud Messaging. Το Storage χρησιμοποιείται για την αποθήκευση των εικόνων που ανεβάζουν οι χρήστες, καθώς η βιβλιοθήκη της υπηρεσίας αυτής μας επιτρέπει την εύκολη μεταφόρτωση των εικόνων αυτών και στο ίδιο βήμα μας επιστρέφει και τον σύνδεσμο που οδηγεί σε αυτές για μελλοντική χρήση. Το Authentication χρησιμοποιείται όχι για την αυθεντικοποίηση των χρηστών, κάτι που γίνεται μέσω του Spring Security που είναι μέρος του Spring Boot, αλλά για έλεγχο πρόσβασης στις εικόνες που ανεβάζουνε. Με αυτόν τον τρόπο επιτρέπουμε μόνο σε έναν αυθεντικοποιημένο χρήστη να μεταφορτώσει μια εικόνα, και στην συνέχεια αυτός έχει πρόσβαση διαγραφής ή αλλαγής των εικόνων αυτών. Η υπηρεσία Cloud Messaging μας δίνει την δυνατότητα να στέλνουμε ειδοποιήσεις στους χρήστες μέσω του Spring Boot API μας με την χρήση της βιβλιοθήκης Firebase Admin η οποία μας επιτρέπει να ελέγχουμε την εφαρμογή μας μέσω της Firebase. Όταν συμβεί κάποιο event στο backend το οποίο σηματοδοτεί την αποστολή ειδοποίησης, θα σταλεί μια αντίστοιχη ειδοποίηση στους χρήστες που έχουν ενεργοποιήσει την λήψη ειδοποιήσεων και πληρούν κάποιες άλλες προϋποθέσεις ανάλογα με το είδος της ειδοποίησης.

3.3. Spring Boot

Για το κύριο μέρος του backend της εφαρμογής έχουμε ένα API το οποίο έχει φτιαχτεί με την βοήθεια του Spring Boot το οποίο είναι ένα framework της Java το οποίο εξορθολογεί την διαδικασία υλοποίησης APIs και βασίζεται στο Spring Framework. Το Spring Boot υπάρχει για να κάνει την χρήση του Spring Framework πιο απλή, καθώς ένα Spring Boot project έρχεται πακεταρισμένο με τον server στον οποίο θα εκτελεστεί, επιτρέπει την εύκολη προσθήκη και αφαίρεση πακέτων με την χρήση ενός εκ των εργαλείων αυτοματοποίησης Maven ή Gradle και δεν χρησιμοποιεί αρχεία τύπου XML για την διαμόρφωση, τα οποία είναι πολύ δυσανάγνωστα. Τα κυριότερα χαρακτηριστικά του Spring Framework και του Spring Boot που έχουμε χρησιμοποιήσει για το backend μας είναι τα εξής:

- **Inversion Of Control:** Το Inversion Of Control είναι μια σχεδιαστική μέθοδος που χρησιμοποιείται από το Spring Framework η οποία επιτρέπει σε αυτό να διαχειρίζεται τα αντικείμενα που χρησιμοποιούνται, είτε αυτά που υπάρχουν εξ αρχής για την βασική λειτουργία της εφαρμογής ή αυτά που ορίζουμε εμείς για τις διάφορες λειτουργίες της εφαρμογής μας, δημιουργώντας, διαμορφώνοντας και διαγράφοντας τα ανάλογα με τις ανάγκες της εφαρμογής.
- **Dependency Injection:** Το Dependency Injection είναι η μέθοδος με την οποία να αντικείμενα που δημιουργούνται αυτόματα από το Spring Framework διατίθεται στις διάφορες κλάσεις και μεθόδους που δημιουργούμε. Μέσω της μεθόδου αυτής μπορούμε να έχουμε πρόσβαση στα βασικά αντικείμενα που χρειαζόμαστε για τις διάφορες λειτουργίες της εφαρμογής από οποιοδήποτε σημείο του κώδικα, και το Spring εξασφαλίζει ότι ανά πάσα στιγμή υπάρχει μόνο ένα αντικείμενο της εκάστοτε κλάσης που σημαίνει ότι πάντα μας δίνεται πρόσβαση στο ίδιο αντικείμενο σε όλη την εφαρμογή μας.
- **Data Access Framework:** Το Data Access Framework αποτελείται από διάφορες βιβλιοθήκες οι οποίες είναι υπεύθυνες για την σύνδεση και την διαχείριση κάποιου Database Management System (DBMS), όπως για παράδειγμα είναι η MySQL ή η PostgreSQL. Το framework αυτό μας επιτρέπει να διαχειριζόμαστε την βάση δεδομένων μας μέσω του κώδικα, φτιάχνοντας πίνακες από τις κλάσεις μας και εισάγοντας δεδομένα με την δημιουργία αντικειμένων των κλάσεων αυτών, χωρίς να χρειάζεται να γράψουμε καθόλου κώδικα SQL. Οι δημοφιλέστερες βιβλιοθήκες που επιτρέπουν την λειτουργικότητα αυτή είναι η Hibernate, η οποία είναι υπεύθυνη για την αντιστοίχιση των αντικειμένων της Java στις σχεσιακές βάσεις δεδομένων, και το Java Persistence API (JPA) το οποίο χρησιμοποιούμε για να περιγράψουμε στην Hibernate την δομή των δεδομένων μας.

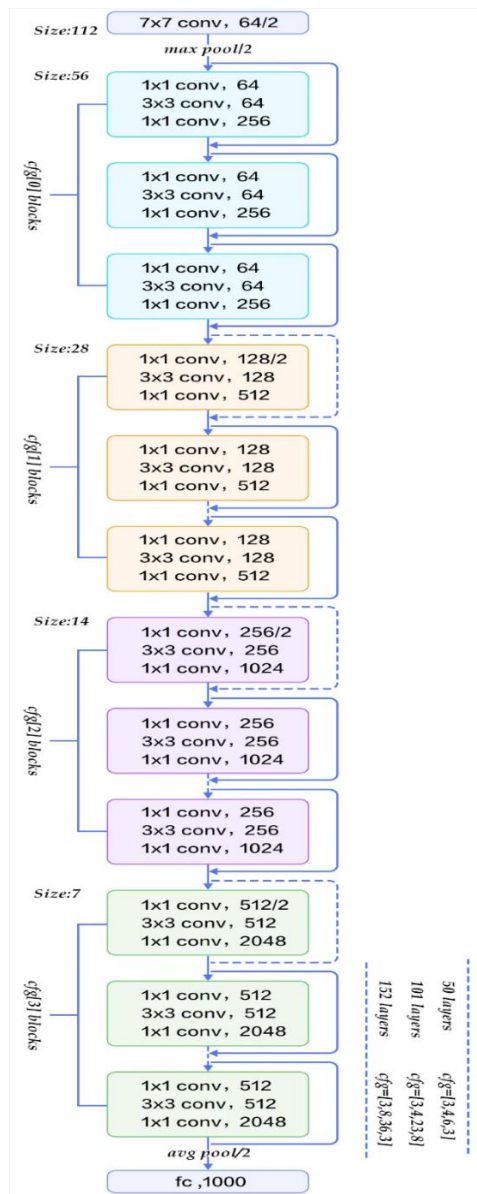
Ένα μεγάλο μέρος της λειτουργικότητας του Spring Boot ενεργοποιείται από τα Annotations της Java, τα οποία είναι λέξεις κλειδιά που ξεκινάνε από το σύμβολο "@" και προστίθενται πριν από τον ορισμό μιας κλάσης, μιας μεθόδου, μιας μεταβλητής ή του τύπου μιας μεταβλητής. Τα Annotations δεν αλλάζουν τον κώδικα που έχουμε γράψει αλλά χρησιμοποιούνται για να ενημερώσουν τον compiler της Java για κάποια επιπρόσθετη λειτουργία που θέλουμε να δώσουμε στον κώδικα που επισημαίνουμε. Μερικές αξιοσημείωτες χρήσεις Annotations στην Spring Boot είναι ο ορισμός των μεθόδων οι οποίες διαχειρίζονται τα αιτήματα των χρηστών ανάλογα με τον σύνδεσμο, ο ορισμός μεθόδων που επιστρέφουν αντικείμενα τα οποία χρησιμοποιούνται για την ρύθμιση παραμέτρων οι οποίες είναι σημαντικές για την λειτουργία της εφαρμογής, ο ορισμός μεταβλητών σε κλάσεις για τις οποίες θέλουμε το Spring είτε να δημιουργήσει ένα αντικείμενο αυτόματα ή να μας δώσει πρόσβαση στο αντικείμενο εάν υπάρχει ήδη ένα και ο ορισμός κλάσεων που περιέχουν βασική λειτουργικότητα της εφαρμογής μας και πρέπει να αρχικοποιηθεί ένα αντικείμενο τους από το Spring Boot κατά την εκκίνηση της εφαρμογής μας. Το JPA είναι μια βιβλιοθήκη που λειτουργεί εξ ολοκλήρου μέσω Annotations για τον ορισμό των κλάσεων που πρέπει να αντιστοιχιστούν σε πίνακες της βάσης δεδομένων και για την προσθήκη πληροφοριών σχετικά με τις μεταβλητές που περιέχει η κλάση, όπως ποια είναι το πρωτεύον κλειδί, ποια ή ποιες δεν θέλουμε να αποθηκευτούν στην βάση, εάν επιτρέπεται σε πάνω από μια καταχώριση να έχει την ίδια τιμή στην εκάστοτε μεταβλητή και για πολλές άλλες λειτουργίες.

3.4. Tensorflow

Μία από τις δυνατότητες που προσφέρει η εφαρμογή μας στους χρήστες είναι ικανότητα της να αναγνωρίζει τα ζώα που φωτογραφίζουν με την χρήση της βιβλιοθήκης Tensorflow. Η βιβλιοθήκη αυτή μας επιτρέπει να σχεδιάσουμε τεχνητά νευρωνικά δίκτυα τα οποία μπορούν να εκτελεστούν είτε στον επεξεργαστή ή, εάν χρειαζόμαστε καλύτερες επιδόσεις κατά την εκπαίδευση και την χρήση των μοντέλων, στην κάρτα γραφικών. Το Tensorflow μπορεί να χρησιμοποιηθεί για διάφορα είδη μηχανικής μαθήσεως όπως είναι η αναγνώριση αντικειμένων σε εικόνες και βίντεο, η γρήγορη αναγνώριση παθήσεων από ιατρικές εξετάσεις και η αναγνώριση ομιλίας και είναι διαθέσιμο στις πιο δημοφιλείς γλώσσες προγραμματισμού όπως Python, C++, Javascript και Java.

Το σημαντικότερο κομμάτι του Tensorflow, και αυτό με το οποίο αλληλοεπιδρούν οι περισσότεροι προγραμματιστές όταν σχεδιάζουν ένα μοντέλο μηχανικής μάθησης, είναι το Keras, το οποίο είναι ένα API υψηλού επιπέδου. Μέσω του Keras οι προγραμματιστές έχουν πρόσβαση στις λειτουργίες της Tensorflow γράφοντας κώδικα ο οποίος είναι απλός και δεν χρειάζεται ιδιαίτερη διαμόρφωση για να εκτελεστεί.

Για την διαδικασία της αναγνώρισης των ζώων από τις φωτογραφίες των χρηστών έχουμε χρησιμοποιήσει ένα ήδη εκπαιδευμένο συνελκτικό νευρωνικό δίκτυο (Convolutional Neural Network ή CNN) το οποίο έχει εκπαιδευτεί σε ένα μεγάλο εύρος διαφορετικών εικόνων, οι οποίες δεν έχουν κάποια συσχέτιση. Τα CNN έχουν σχεδιαστεί για να εκπαιδεύονται με την χρήση εικόνων και βίντεο, έχοντας την δυνατότητα να εξάγουν χαρακτηριστικά από οπτικό υλικό με την χρήση φίλτρων. Το μοντέλο που έχουμε επιλέξει είναι το ResNet50, το οποίο δημοσιεύτηκε από την Microsoft το 2015.



Αρχιτεκτονική ResNet

Το μοντέλο που χρησιμοποιούμε στην εφαρμογή μας περιλαμβάνει μερικά layers προεπεξεργασίας των εικόνων, όπως η αλλαγή μεγέθους και η προσαρμογή των τιμών των pixel της εικόνας στο εύρος [-1,1], το μοντέλο ResNet50 και τα layers εξόδου τα οποία μας παρουσιάζουν το ζώο που έχει προβλεφθεί. Κατά την εκπαίδευση του μοντέλου με τα δικά μας

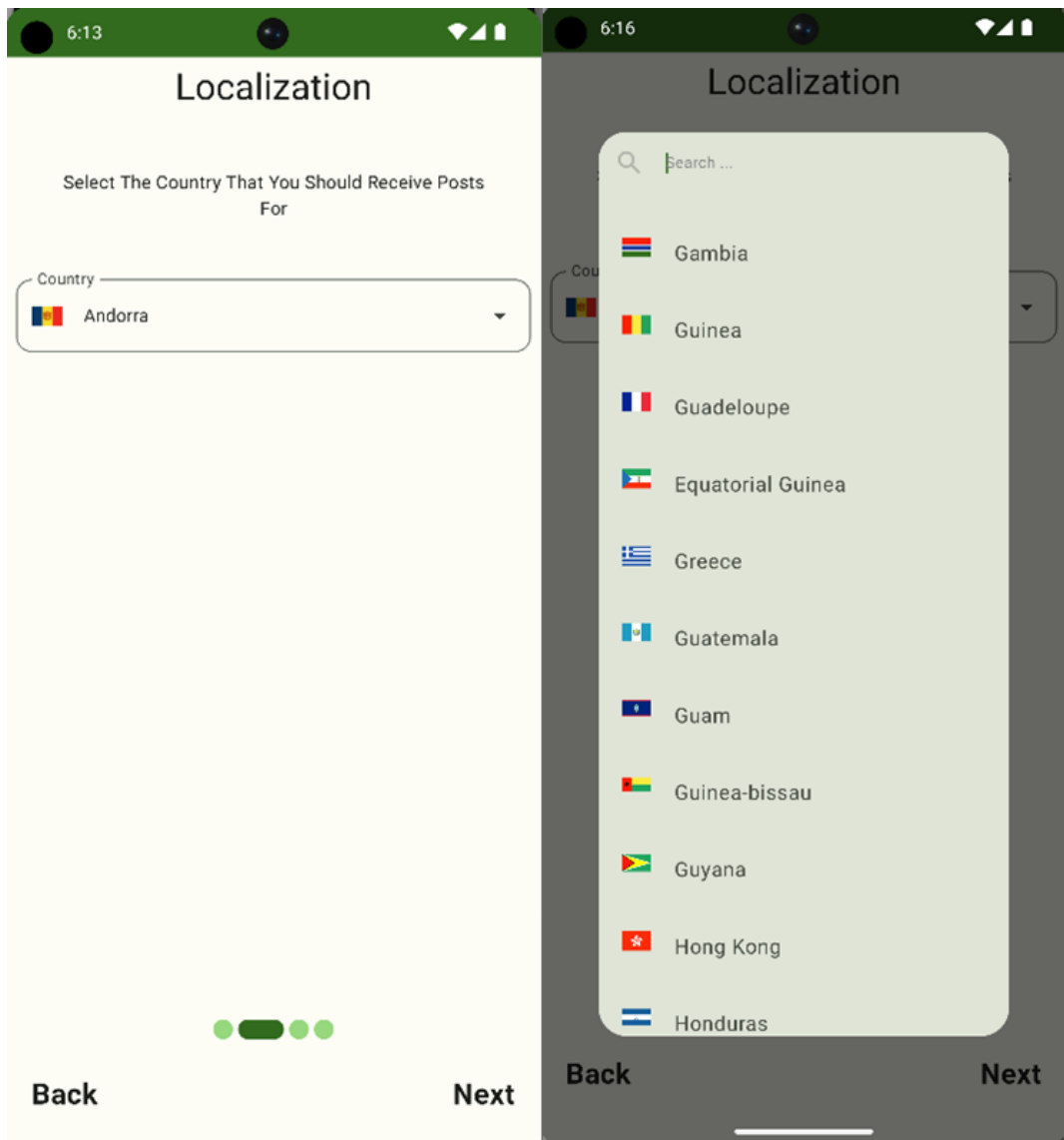
δεδομένα τα 50 layers του ResNet50 δεν επανεκπαιδεύονται καθώς χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών από τις εικόνες και έχουν ήδη εκπαιδευτεί για αυτήν την λειτουργία.

Η αναγνώριση ενός ζώου από το μοντέλο γίνεται σε δυο βήματα. Αρχικά έχουμε μια έκδοση του μοντέλου η οποία έχει εκπαιδευτεί έτσι ώστε να αναγνωρίζει την κατηγορία του ζώου. Οι κατηγορίες των ζώων περιλαμβάνουν ζώα που ανήκουν στην ίδια φυλή και συνεπώς μοιάζουν παρόμοια. Αφού αναγνωρισθεί η κατηγορία, χρησιμοποιούμε ένα μοντέλο που έχει εκπαιδευτεί για να αναγνωρίζει μόνο τα ζώα που περιέχονται στην κατηγορία αυτή για να βρούμε ποιο ζώο είναι. Αυτή η προσέγγιση μας επιτρέπει αρχικά να περιορίσουμε το συνολικό πλήθος των ζώων σε αυτά μιας κατηγορίας και στην συνέχεια να κάνουμε μια τελική πρόβλεψη μόνο από τα ζώα της κατηγορίας αυτής, κάτι που είναι προτιμότερο από το να εκπαιδεύσουμε ένα μοντέλο με όλα τα ζώα καθώς δεν έχουμε αρκετές διαφορετικές φωτογραφίες ζώων ώστε το μοντέλο να μπορεί να αναγνωρίζει με ικανοποιητική ακρίβεια από αυτό το πλήθος των διαφορετικών ζώων.

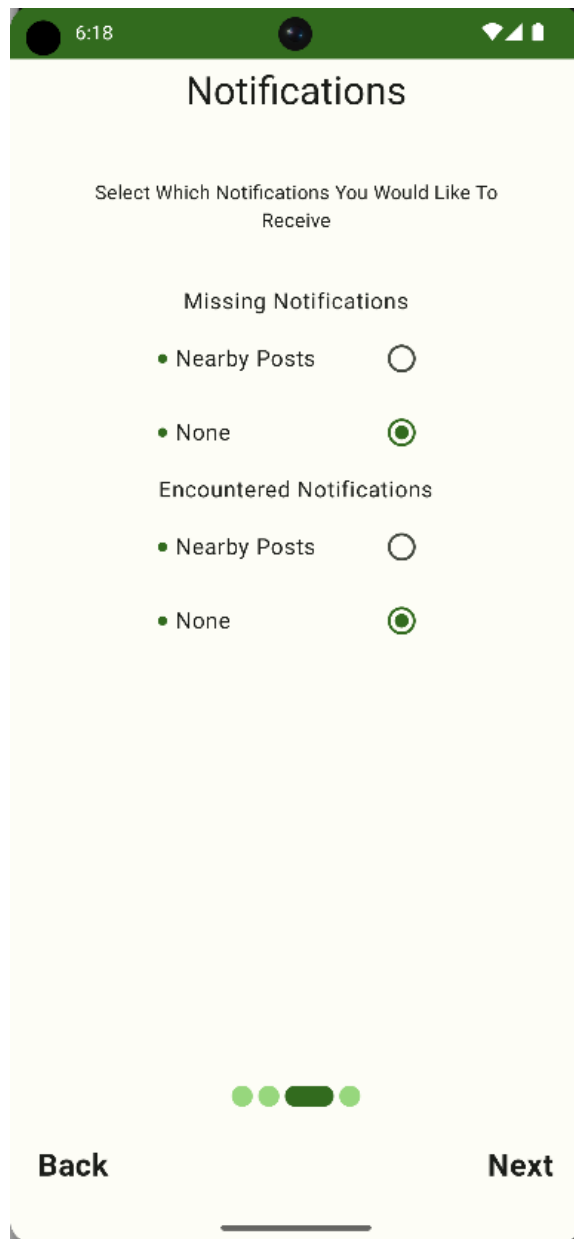
4. Παρουσίαση Εφαρμογής



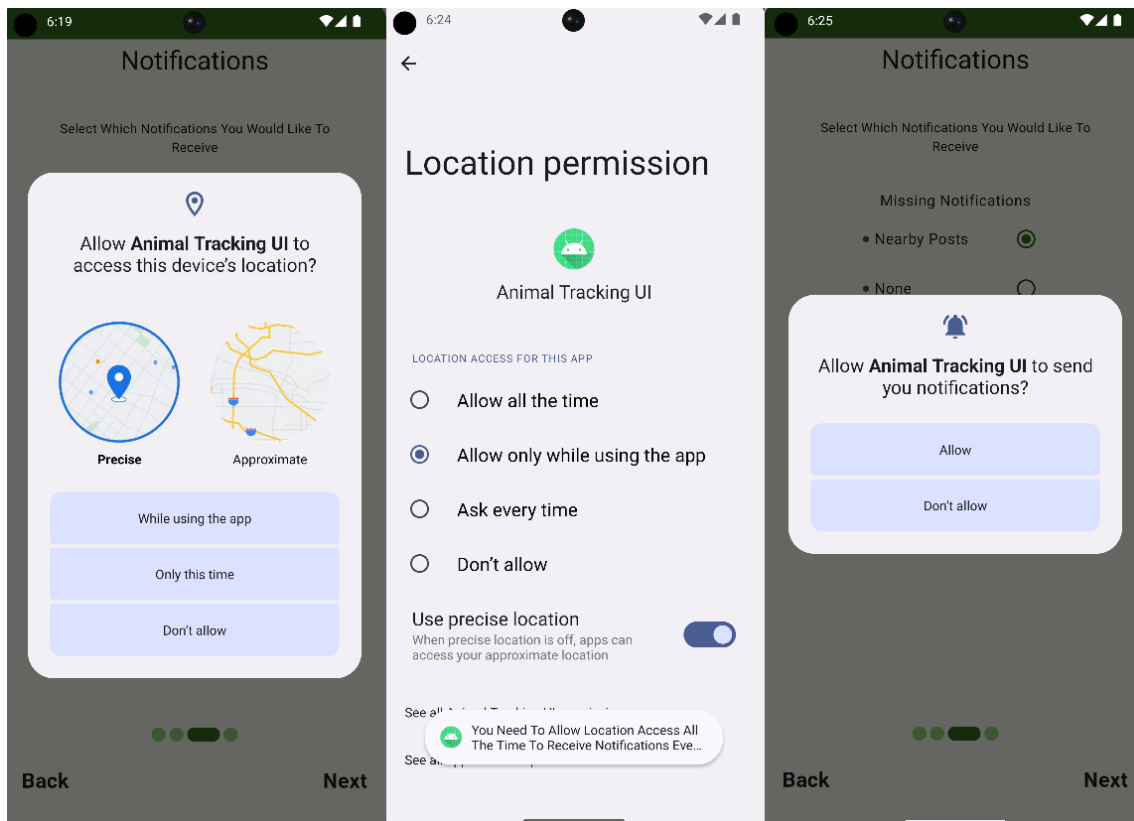
Όταν ο χρήστης ανοίγει την εφαρμογή για πρώτη φορά βλέπει ένα μήνυμα καλωσορίσματος το οποίο ακολουθείται από την ρύθμιση μερικών βασικών για την εφαρμογή παραμέτρων.



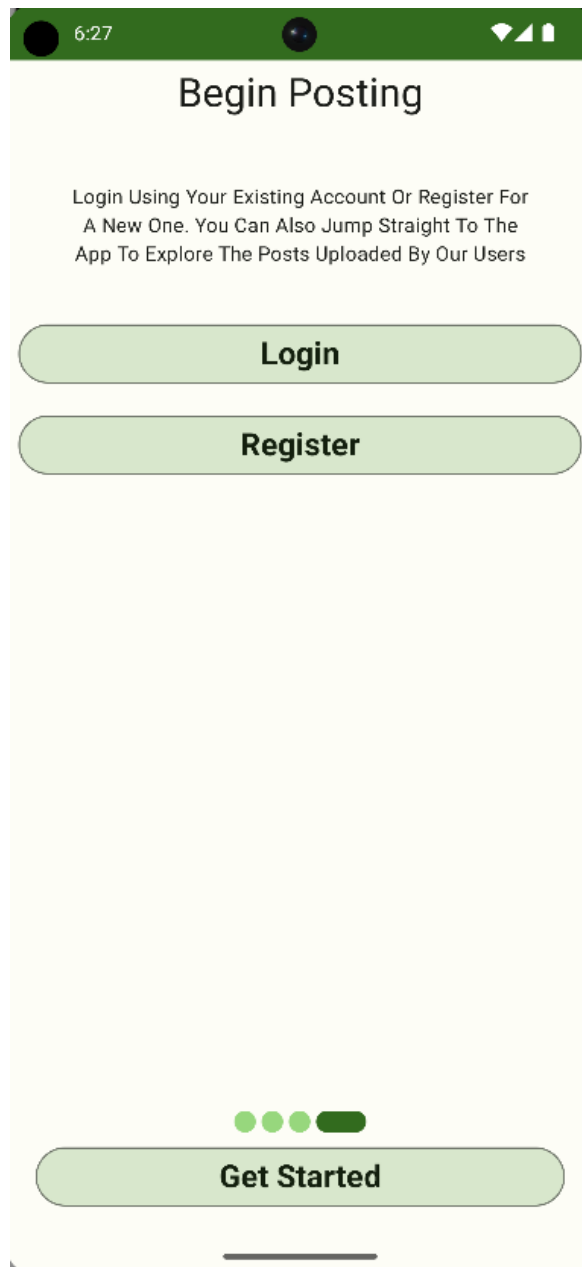
Το πρώτο πράγμα που επιλέγει ο χρήστης είναι η χώρα στην οποία βρίσκεται και για την οποία θέλει να βλέπει δημοσιεύσεις.



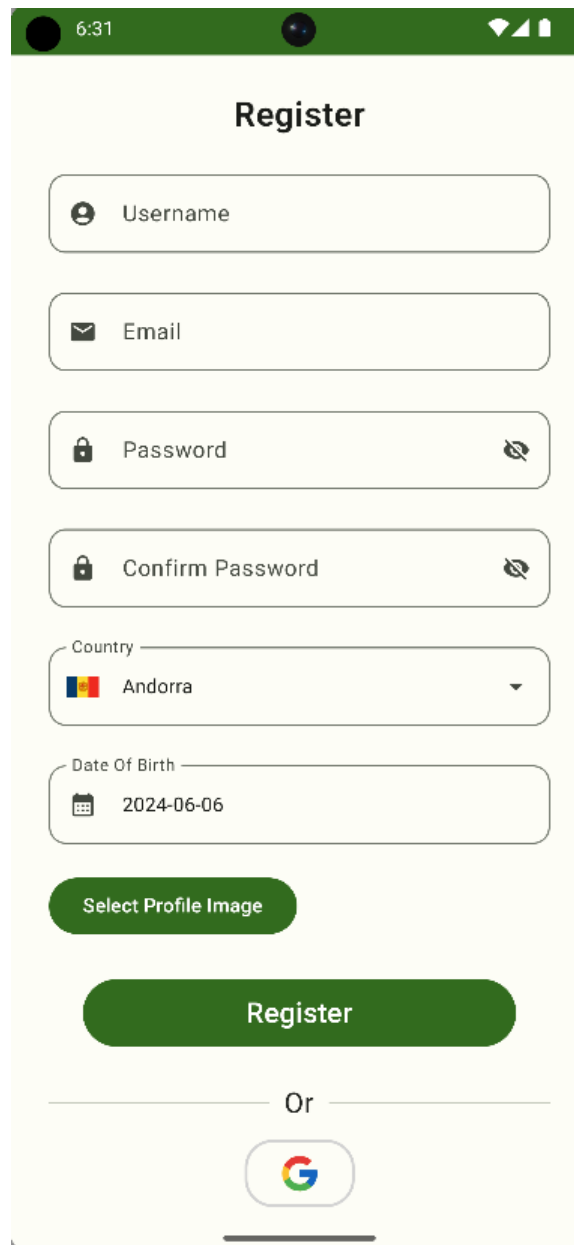
Στην συνέχεια ο χρήστης επιλέγει τι είδους ειδοποιήσεις θέλει να λαμβάνει. Για το κάθε είδος δημοσίευσης ο χρήστης μπορεί να επιλέξει εάν θέλει να λαμβάνει ειδοποιήσεις για δημοσιεύσεις που γίνονται κοντά του ή όχι.



Εάν ο χρήστης επιλέξει να λαμβάνει ειδοποιήσεις για κοντινές δημοσιεύσεις η εφαρμογή θα του ζητήσει να παρέχει δικαιώματα για την τοποθεσία του, πρόσβαση στην τοποθεσία του ακόμα και όταν η εφαρμογή δεν είναι ανοιχτή και για να του στέλνει ειδοποιήσεις.

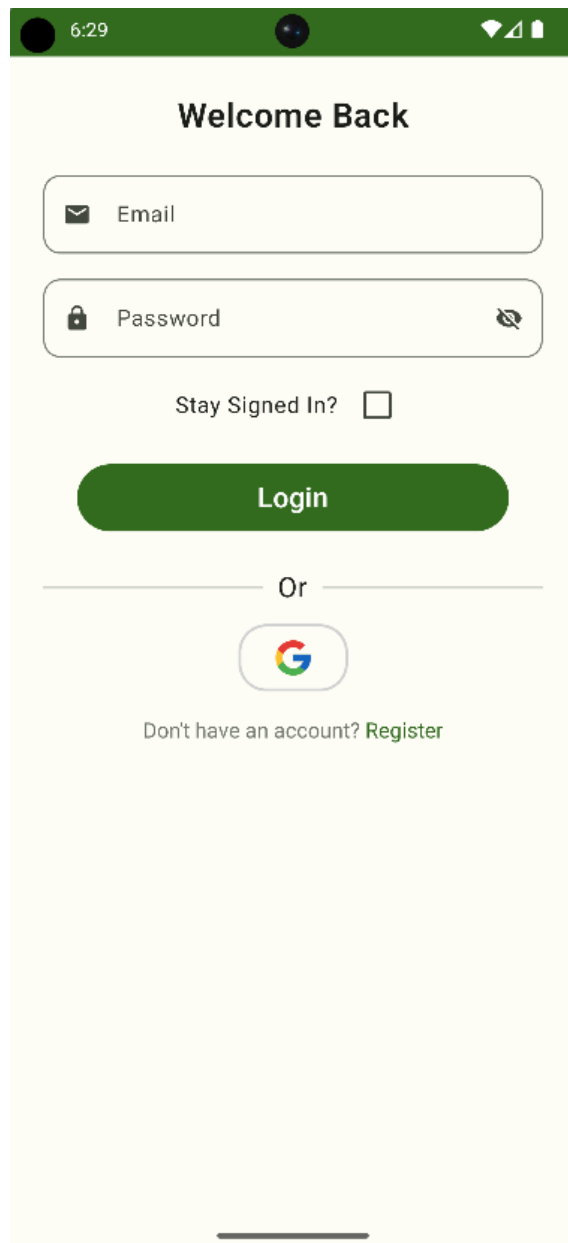


Αφού ο χρήστης ορίσει τις βασικές παραμέτρους της εφαρμογής έχει την επιλογή να κάνει σύνδεση με τον λογαριασμό του σε περίπτωση που έχει έναν, να δημιουργήσει έναν εάν δεν έχει ή να ξεκινήσει κατευθείαν να χρησιμοποιεί την εφαρμογή.

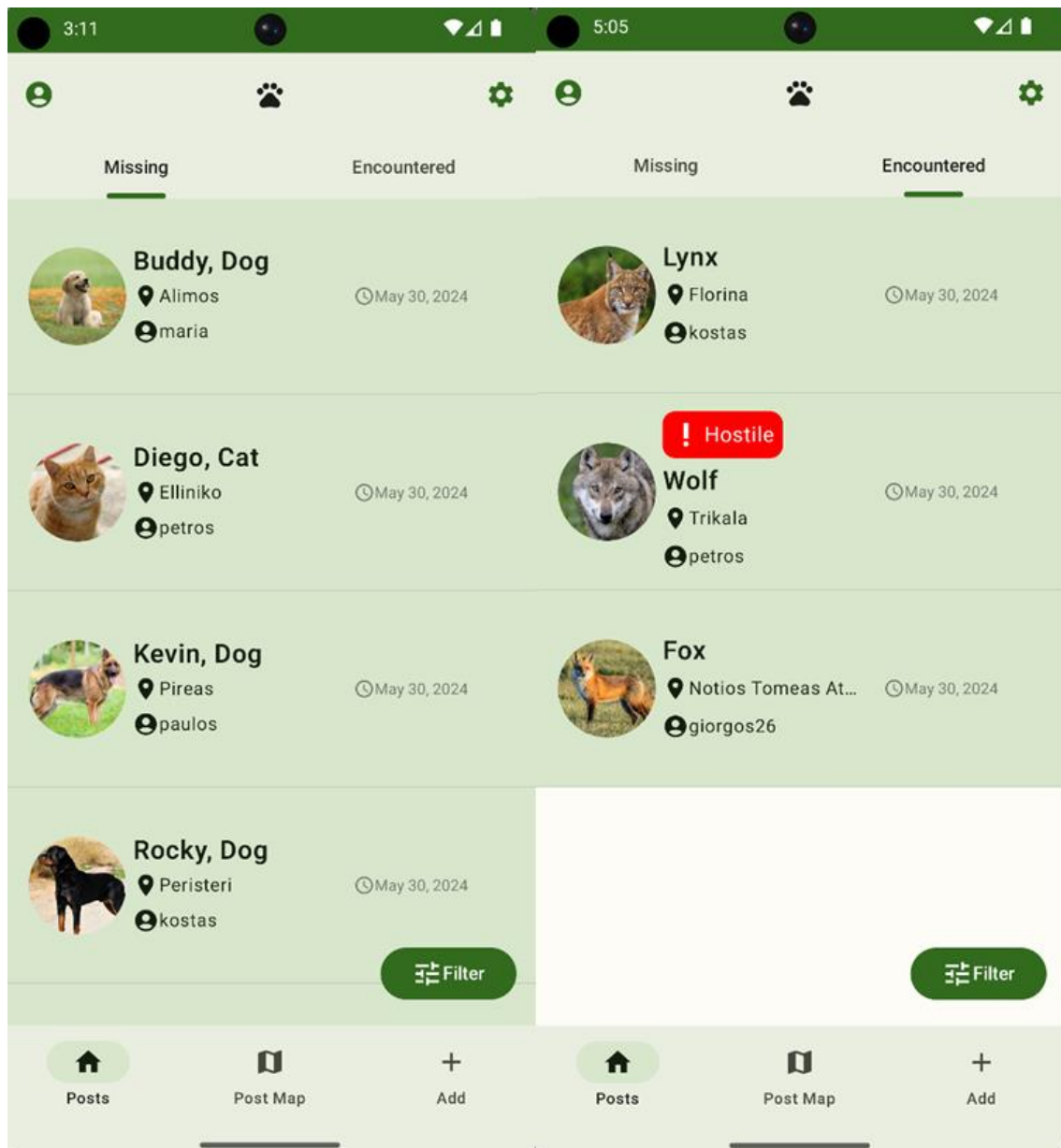


The screenshot shows a mobile application interface for registration. At the top, there is a green status bar with the time 6:31 and signal strength indicators. The main title is "Register". Below the title are several input fields: "Username" with a person icon, "Email" with an envelope icon, "Password" with a lock icon and a toggle for visibility, "Confirm Password" with a lock icon and a toggle for visibility, "Country" with a dropdown menu showing "Andorra" and its flag, and "Date Of Birth" with a calendar icon and the date "2024-06-06". Below these fields is a green button labeled "Select Profile Image". At the bottom, there is a large green button labeled "Register". Below the "Register" button is the word "Or" and a circular icon with the Google logo, indicating a sign-in option.

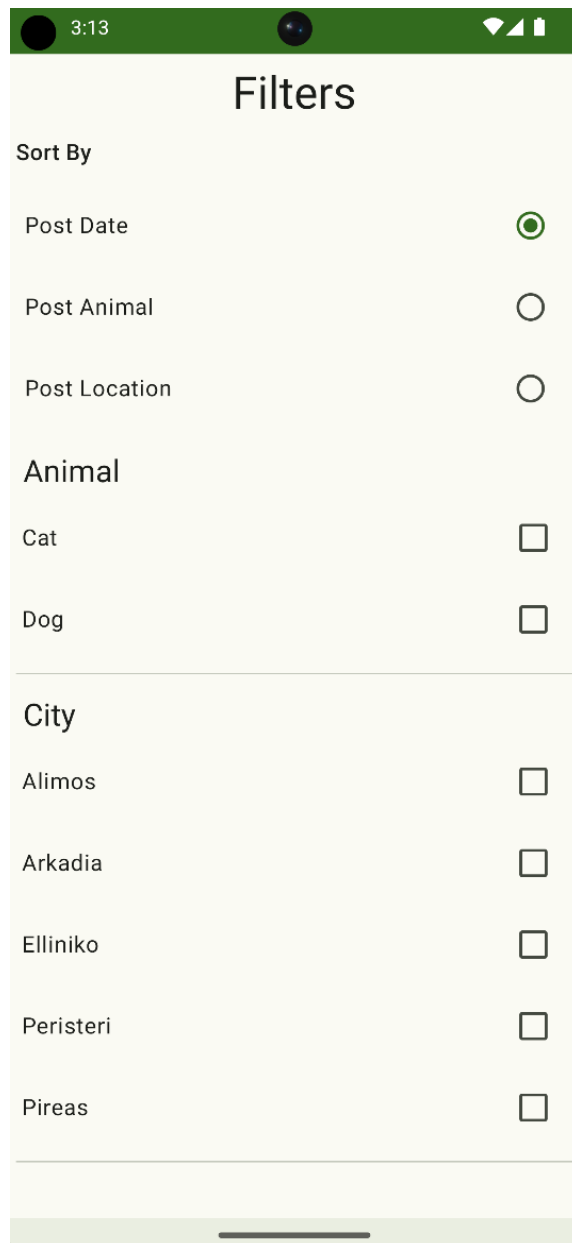
Στην οθόνη εγγραφής ο χρήστης πρέπει να παρέχει ένα όνομα χρήστη το οποίο θα εμφανίζεται στις δημοσιεύσεις που κάνει, ένα email, τον κωδικό πρόσβασης του λογαριασμού του, την χώρα διαμονής του, την ημερομηνία γέννησης του και, προαιρετικά, μια εικόνα προφίλ.



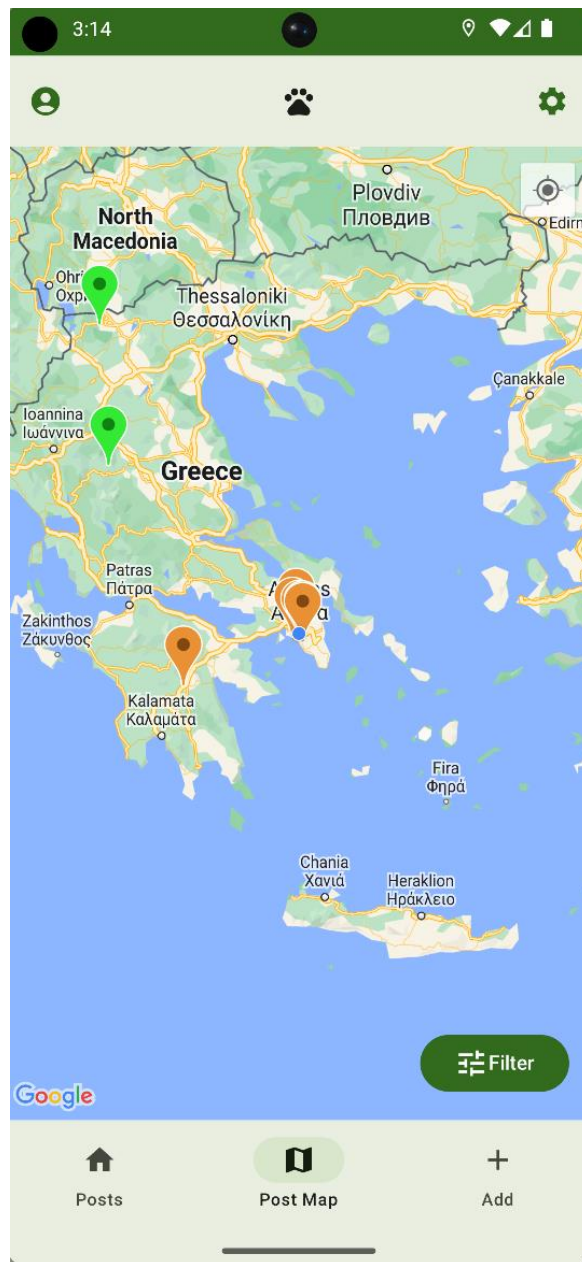
Για την σύνδεση του ένας χρήστης μπορεί να χρησιμοποιήσει το email και τον κωδικό που έφτιαξε κατά την εγγραφή του ή να συνδεθεί με την χρήση ενός λογαριασμού Google στον οποίο είναι συνδεδεμένος από την συσκευή που χρησιμοποιεί.



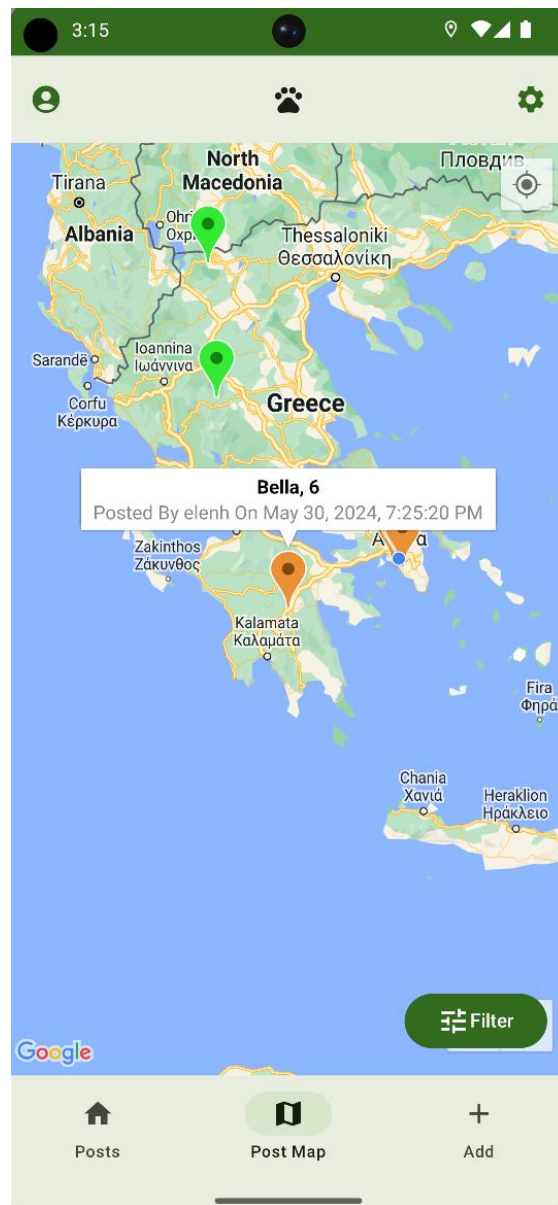
Αφού ο χρήστης περάσει την αρχική ρύθμιση της εφαρμογής έρχεται στην κεντρική οθόνη της εφαρμογής. Σε αυτήν την οθόνη φαίνονται αρχικά οι δημοσιεύσεις για χαμένα κατοικίδια και διάφορες άλλες επιλογές. Ο χρήστης από εδώ μπορεί να αλλάξει τις δημοσιεύσεις που εμφανίζονται σε αυτές με ζώα που έχουν βρεθεί από άλλους χρήστες. Επίσης από εδώ ο χρήστης μπορεί να μεταφερθεί στην οθόνη σύνδεσης εάν δεν είναι συνδεδεμένος πατώντας το εικονίδιο στο πάνω αριστερά μέρος της οθόνης, στην οθόνη ρυθμίσεων πατώντας το γρανάζι στο πάνω δεξιά μέρος της οθόνης, στην οθόνη χάρτη που περιέχει τις δημοσιεύσεις πάνω σε έναν χάρτη της χώρας του χρήστη και στην οθόνη προσθήκης νέας δημοσίευσης.



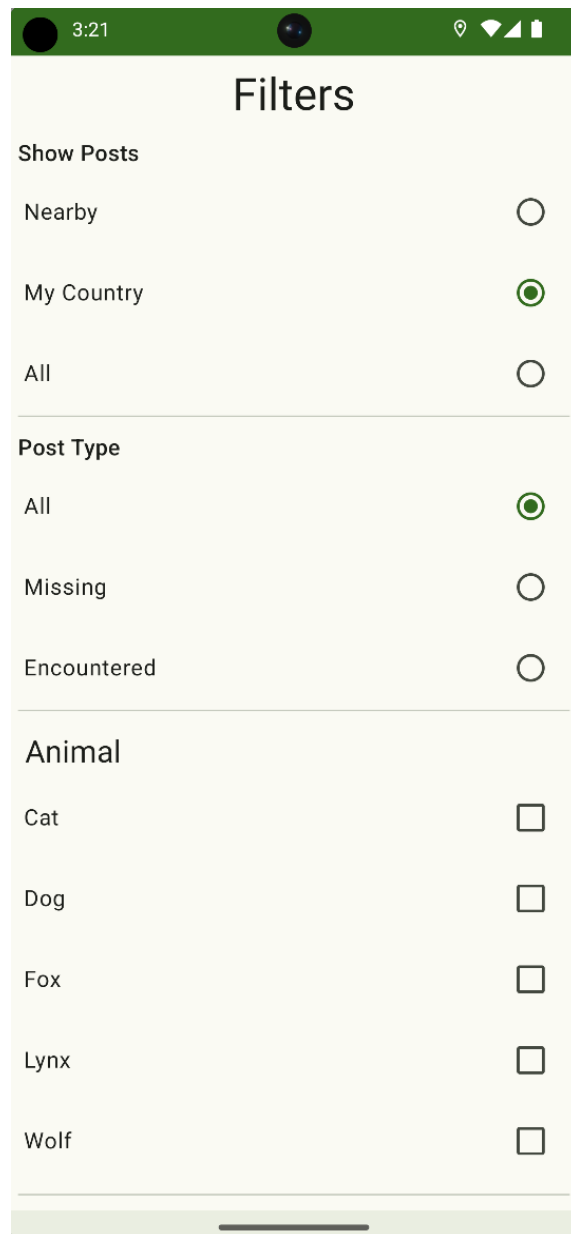
Πατώντας το κουμπί Filter ο χρήστης έχει την δυνατότητα να προσδιορίσει τις δημοσιεύσεις που θέλει να δει με βάση το είδος του ζώου και με βάση την περιοχή στην οποία είχε βρεθεί τελευταία. Επίσης μπορεί να αλλάξει την σειρά των δημοσιεύσεων έτσι ώστε να είναι μαζεμένες αλφαβητικά είτε με βάση το είδος του ζώου ή με βάση την τοποθεσία.



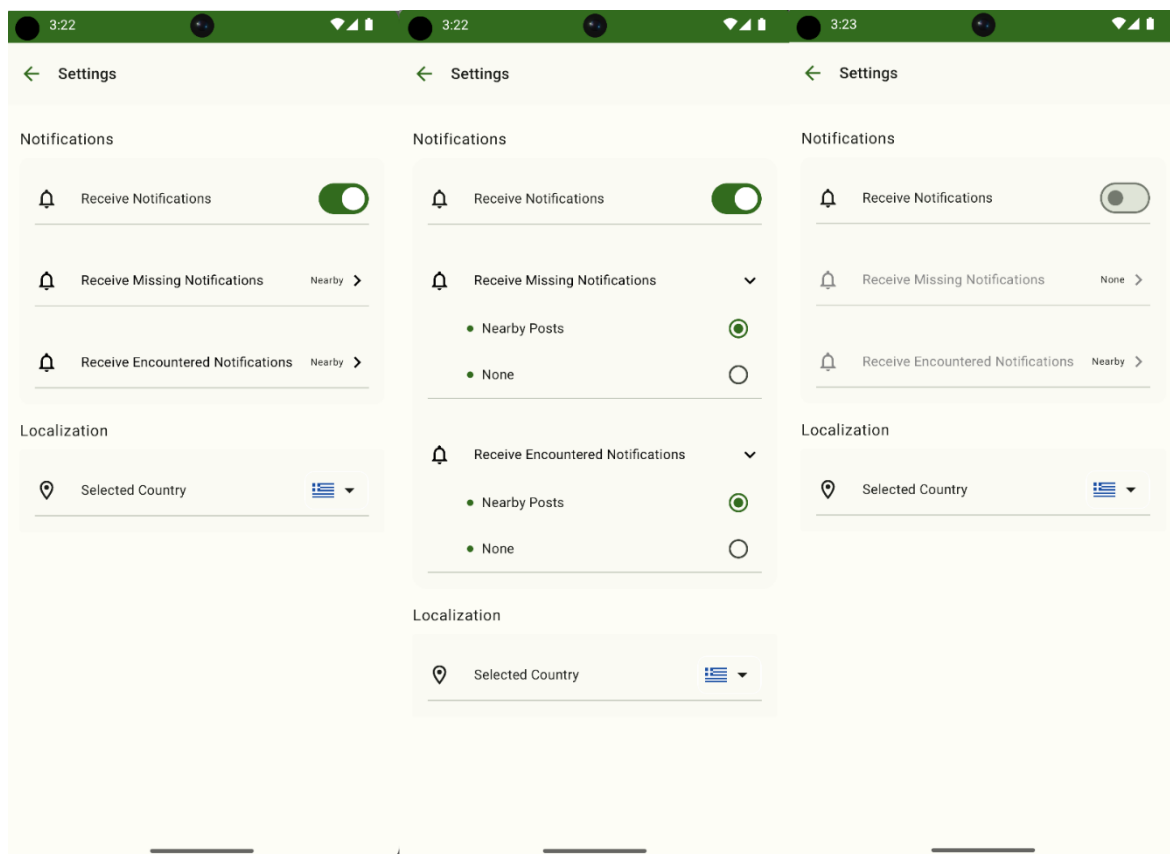
Στην οθόνη του χάρτη ο χρήστης μπορεί να δει τις τοποθεσίες όλων των δημοσιεύσεων πάνω σε έναν χάρτη, με τις δημοσιεύσεις χαμένων κατοικίδιων να έχουν πορτοκαλί σημάδι και οι δημοσιεύσεις για ζώα που έχουν βρεθεί πράσινο.



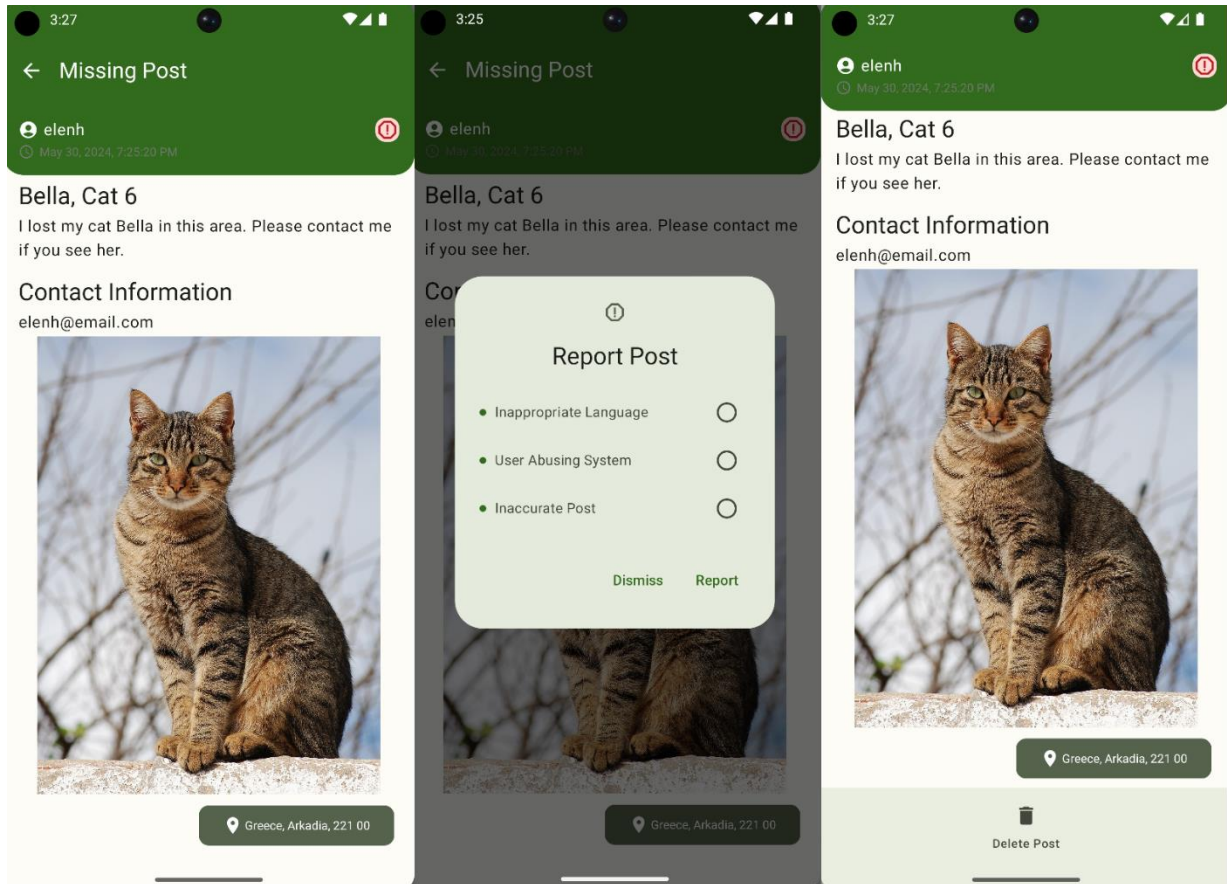
Πατώντας πάνω σε ένα σημάδι στον χάρτη ο χρήστης μπορεί να δει περισσότερες πληροφορίες σχετικά με την δημοσίευση όπως τον χρήστη που έκανε την δημοσίευση και την ημερομηνία και ώρα που έγινε. Πατώντας πάνω στο κουτί που περιέχει τις πληροφορίες αυτές ο χρήστης μεταφέρεται στην οθόνη προβολής δημοσίευσης.



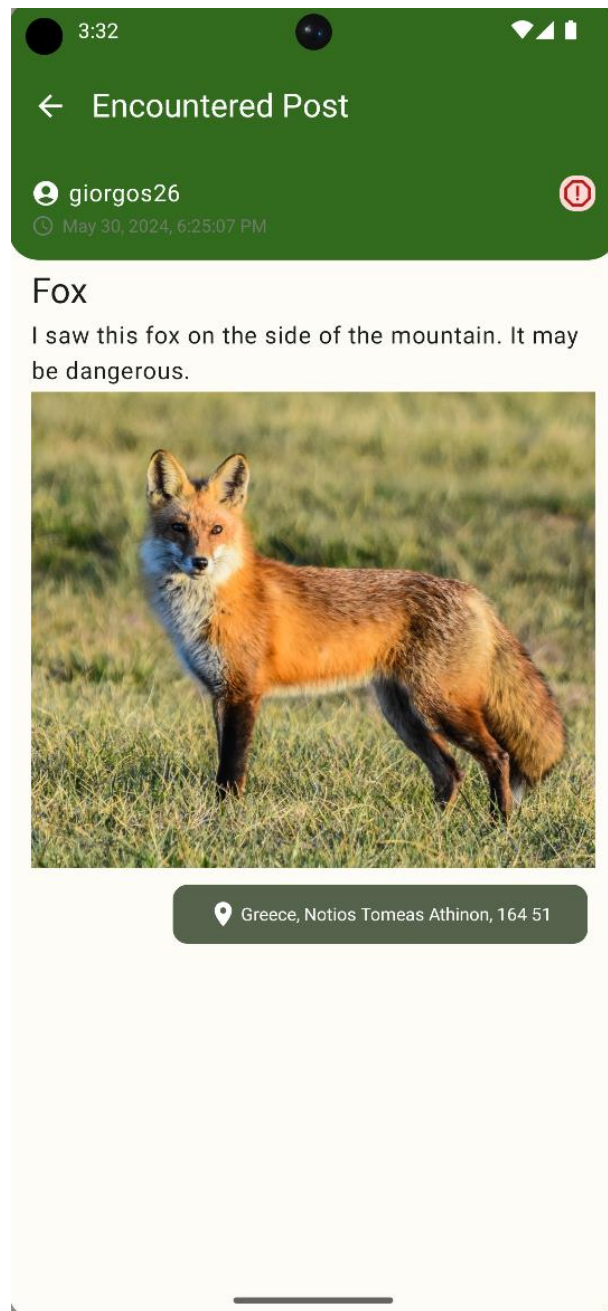
Τα φίλτρα για τις δημοσιεύσεις πάνω στον χάρτη είναι παρόμοια με αυτά στην λίστα δημοσιεύσεων, με την μόνη διαφορά να είναι ότι αντί για την επιλογή για μέθοδο ταξινόμησης ο χρήστης μπορεί να επιλέξει αν θα βλέπει μόνο δημοσιεύσεις που βρίσκονται κοντά του, δηλαδή σε απόσταση έως 5 χιλιομέτρων από την θέση του, δημοσιεύσεις που έγιναν από την χώρα στην οποία βρίσκεται ή όλες τις δημοσιεύσεις που έχουν γίνει στην εφαρμογή. Επίσης μπορεί να επιλέξει αν θα βλέπει όλα τα είδη δημοσιεύσεων, δημοσιεύσεις για χαμένα ζώα, ή δημοσιεύσεις για ζώα που έχουν βρεθεί από χρήστες.



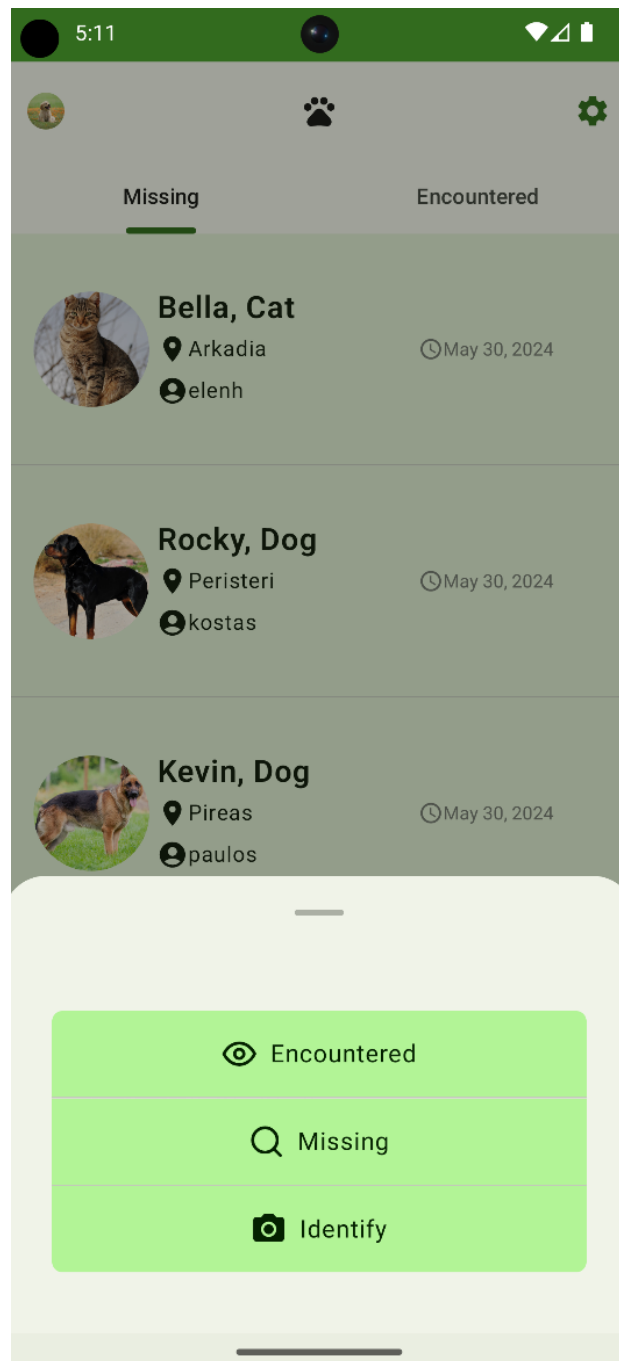
Στην σελίδα ρυθμίσεων ένας χρήστης μπορεί να διαχειριστεί τις παραμέτρους που όρισε κατά την πρώτη εκκίνηση της εφαρμογής. Ο χρήστης μπορεί να ενεργοποιήσει ή να απενεργοποιήσει τις ειδοποιήσεις, να αλλάξει το εύρος των ειδοποιήσεων που λαμβάνει και να αλλάξει την χώρα για τις οποίες τις δημοσιεύσεις βλέπει.



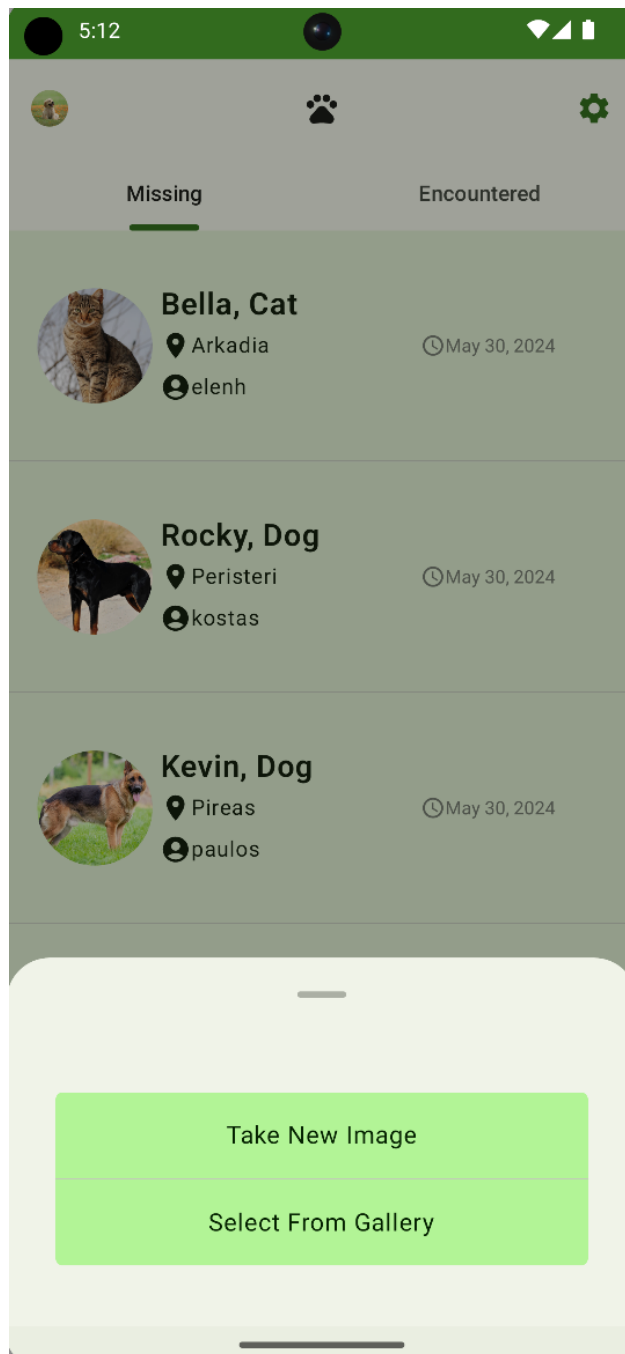
Η οθόνη προβολής δημοσίευσης περιέχει τα στοιχεία που παρέχει ο χρήστης που την δημιούργησε. Η δημοσίευση χαμένου κατοικίδιου προβάλλει το όνομα χρήστη του χρήστη που την δημιούργησε, την ημερομηνία και ώρα που δημιουργήθηκε, τα στοιχεία σχετικά με το κατοικίδιο που παρέχει ο χρήστης, την εικόνα του κατοικίδιου και την περιοχή στην οποία χάθηκε. Πατώντας πάνω στο κουμπί που αναγράφει την περιοχή της δημοσίευσης ο χρήστης θα μεταφερθεί στους χάρτες για να δει την ακριβή θέση που επέλεξε ο χρήστης που την δημιούργησε. Ένας χρήστης μπορεί από αυτήν την οθόνη να κάνει αναφορά της δημοσίευσης αν πιστεύει ότι υπάρχει κάποιο θέμα με αυτήν. Όταν ο χρήστης που δημιούργησε μια δημοσίευση μεταβεί στην οθόνη προβολής της έχει την επιπλέον επιλογή να την διαγράψει εάν το θελήσει.



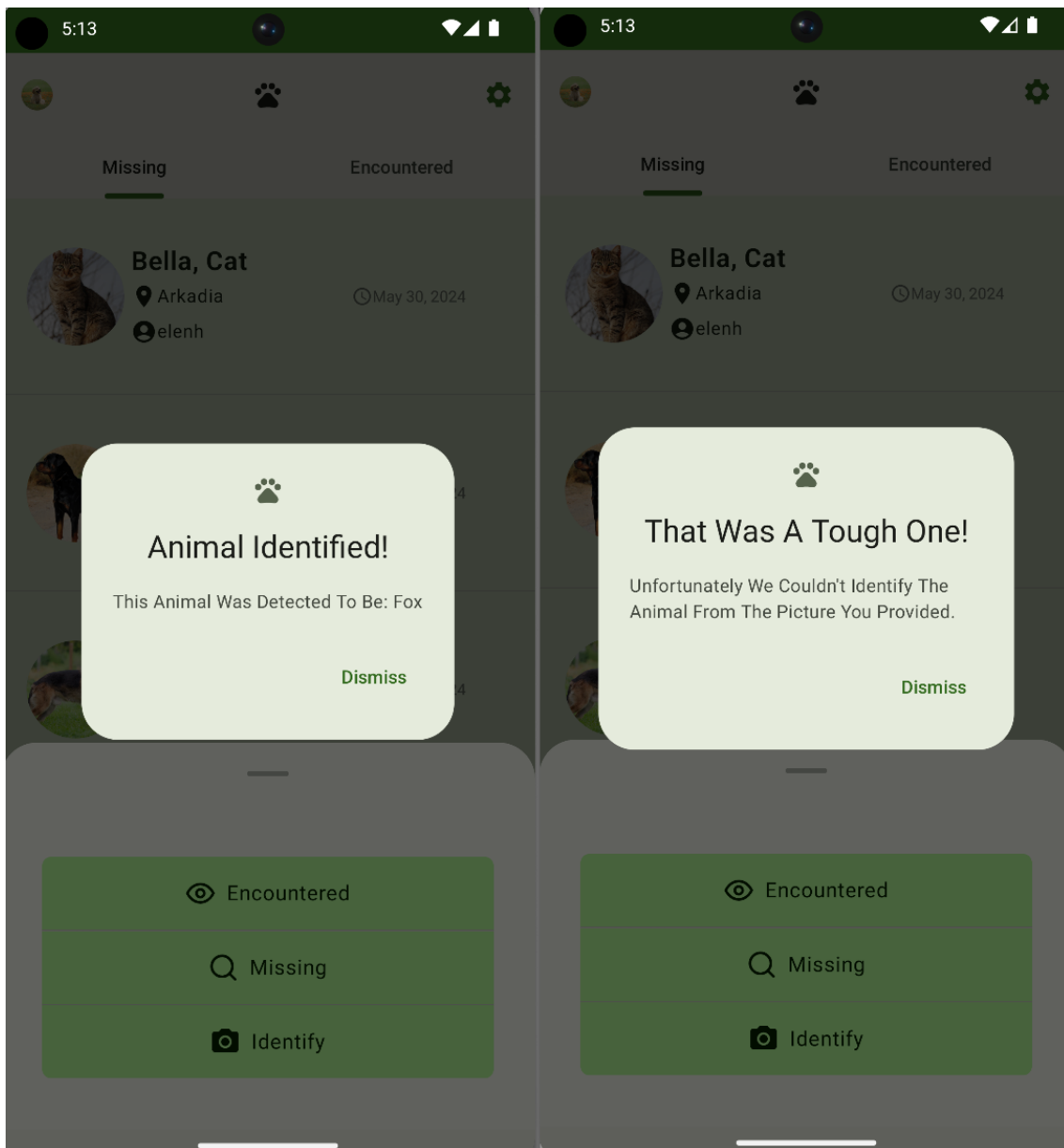
Η οθόνη προβολής δημοσίευσης ζώου που έχει βρεθεί από χρήστη είναι παρόμοια με αυτήν για τα χαμένα ζώα, με μόνη διαφορά την έλλειψη στοιχείων επικοινωνίας με τον χρήστη που την δημιούργησε.



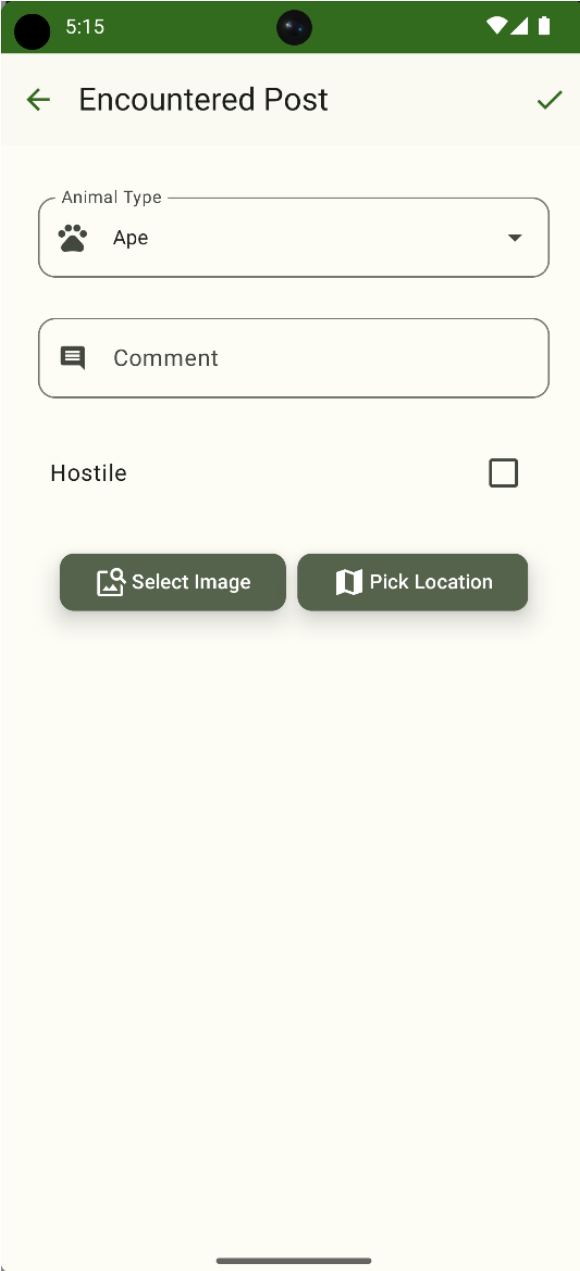
Ένας χρήστης μπορεί να δημιουργήσει μια δημοσίευση πατώντας το κουμπί “Add” στην κεντρική οθόνη της εφαρμογής. Από εκεί μπορεί να επιλέξει εάν θέλει να προσθέσει δημοσίευση για κάποιο ζώο που συνάντησε, ένα κατοικίδιο που έχασε ή αν θέλει απλά να αναγνωρίσει το ένα ζώο που έχει φωτογραφήσει.



Εάν ο χρήστης επιλέξει να αναγνωρίσει ένα ζώο από μια φωτογραφία θα του ζητηθεί να επιλέξει αν θέλει να βγάλει μια νέα φωτογραφία ή να κάνει αναγνώριση χρησιμοποιώντας μια φωτογραφία που έχει βγάλει ήδη.

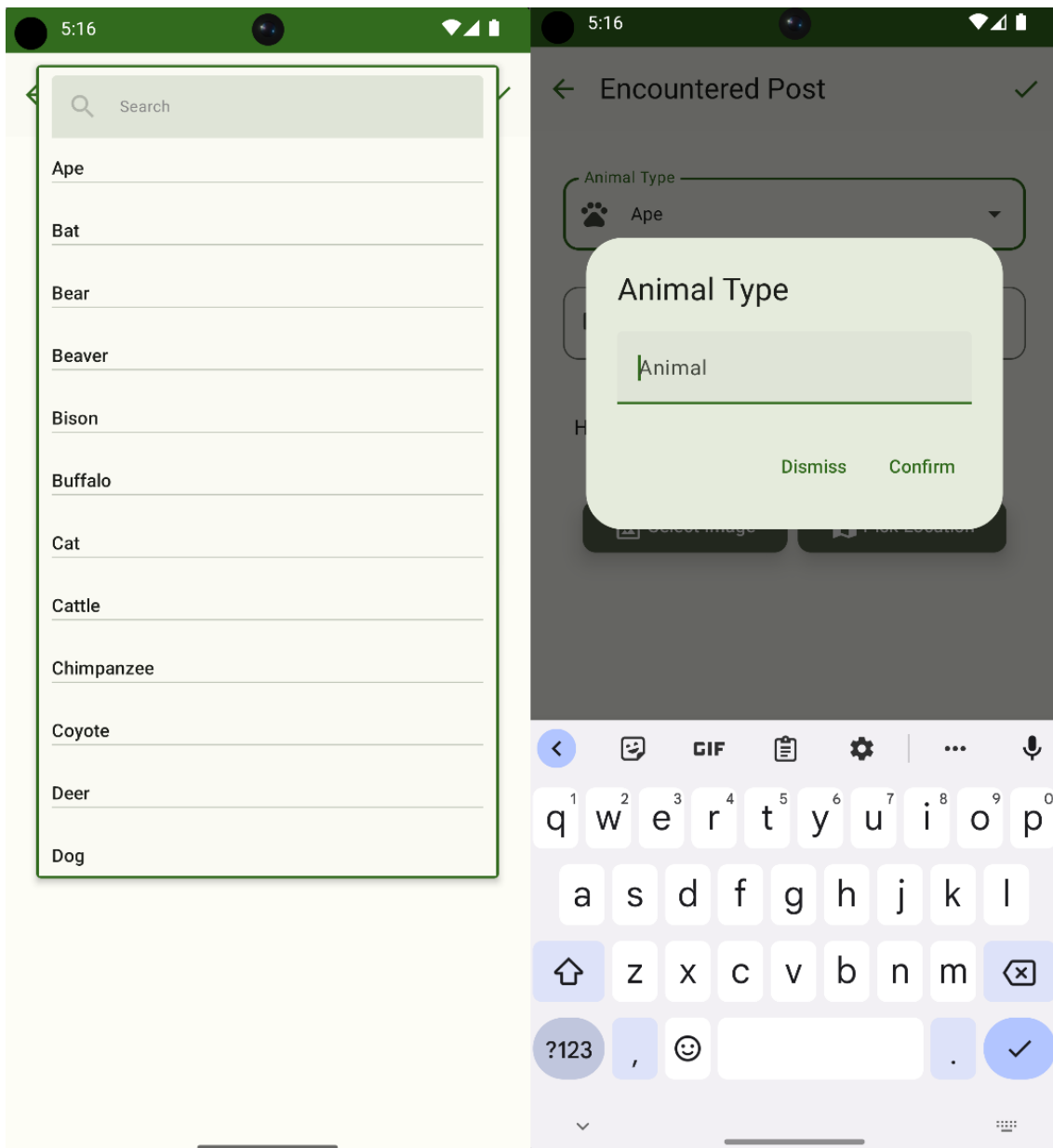


Όταν ο χρήστης στείλει μια εικόνα για αναγνώριση ζώου θα λάβει ως απάντηση είτε το είδος του ζώου ή εάν δεν μπόρεσε να αναγνωρισθεί το ζώο θα λάβει ένα αντίστοιχο μήνυμα.

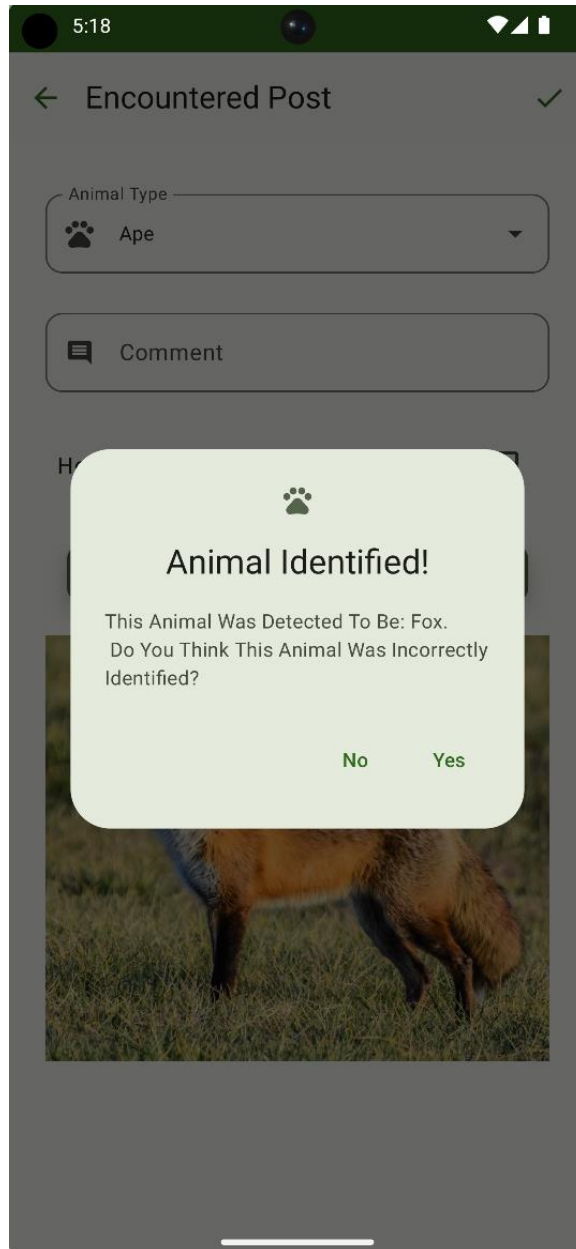


The screenshot shows a mobile application interface for creating a post. At the top, there is a green status bar with the time 5:15 and icons for signal strength, Wi-Fi, and battery. Below this is a white header with a back arrow, the text "Encountered Post", and a checkmark icon. The main content area is white and contains several elements: a dropdown menu for "Animal Type" with a paw print icon and the text "Ape"; a text input field with a speech bubble icon and the text "Comment"; a checkbox labeled "Hostile" which is currently unchecked; and two dark green buttons with white text: "Select Image" (with a camera icon) and "Pick Location" (with a location pin icon).

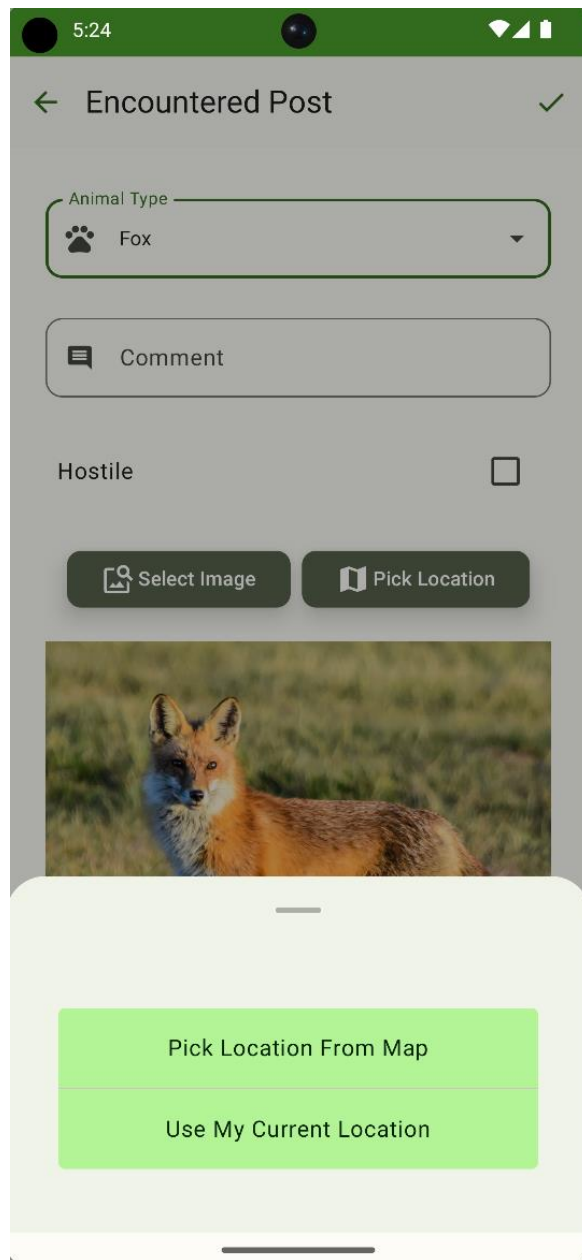
Στην οθόνη δημιουργίας δημοσίευσης για ζώο που έχει συναντήσει ένας χρήστης πρέπει να εισάγει το είδος του ζώου που είδε, ένα σχόλιο σχετικά με αυτό, εάν φάνηκε να είναι επιθετικό ή όχι, μια φωτογραφία του ζώου και την τοποθεσία στην οποία το είδε.



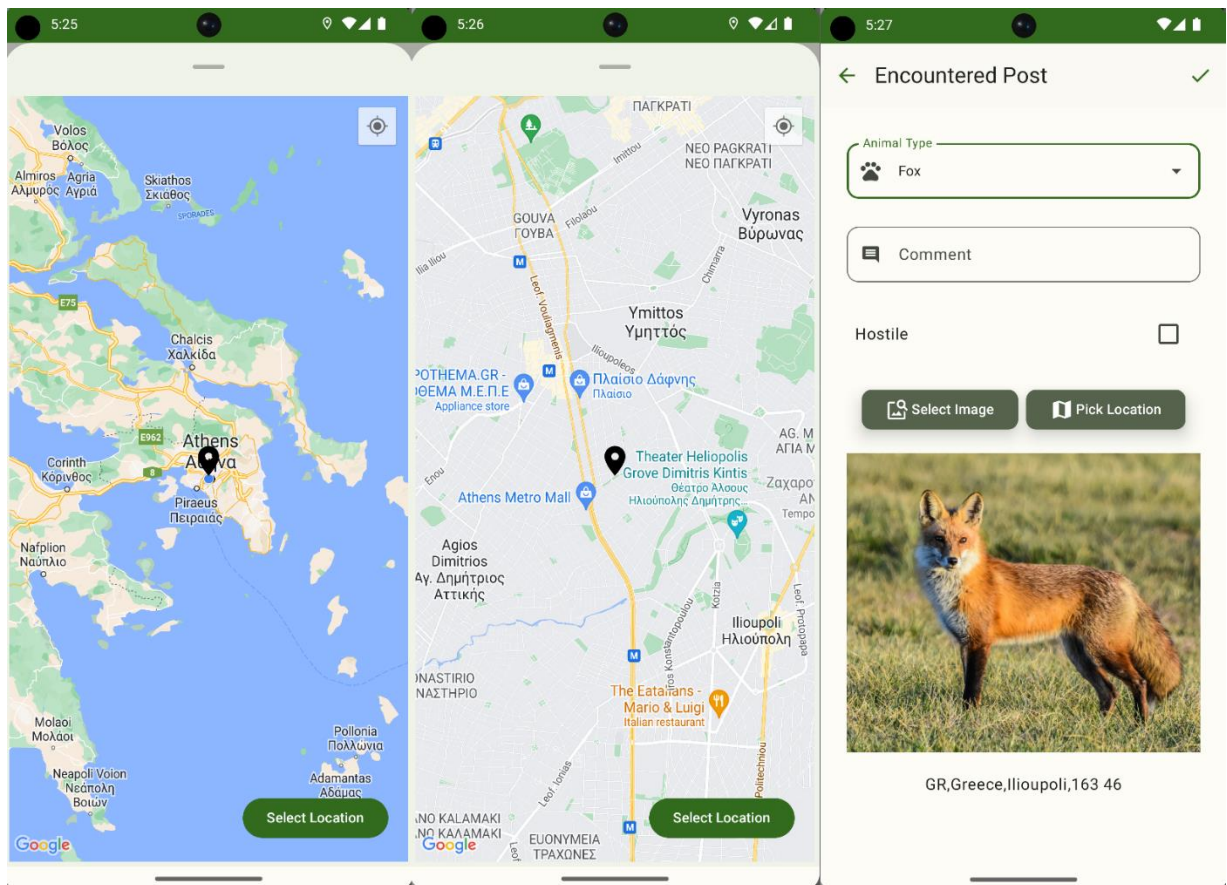
Για τη επιλογή του είδους του ζώου ο χρήστης πρέπει είτε να το επιλέξει από μια λίστα ζώων ή πηγαίνοντας στο τέλος της λίστας να επιλέξει "Other" και να γράψει το είδος του ζώου που είδε εάν δεν υπάρχει στην λίστα.



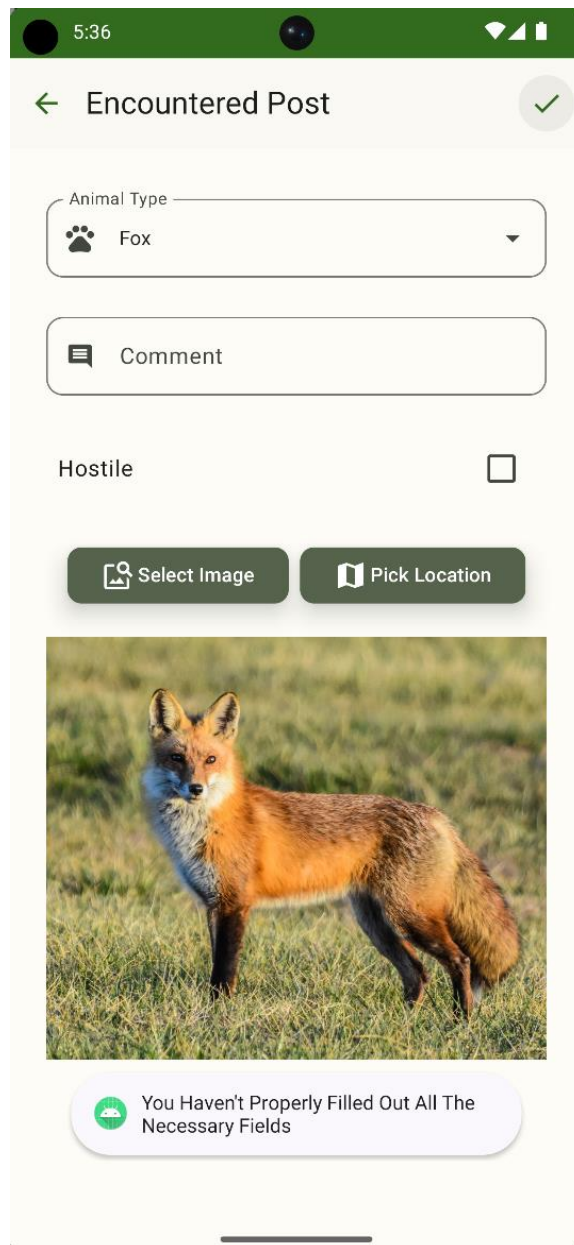
Η διαδικασία επιλογής εικόνας για το ζώο είναι ίδια με αυτήν για την επιλογή εικόνας για αναγνώριση ζώου που περιγράφηκε προηγουμένως. Μετά την επιλογή εικόνας η εφαρμογή θα προσπαθήσει πάλι να αναγνωρίσει το ζώο και θα δώσει την επιλογή στον χρήστη να κρίνει εάν πιστεύει ότι αυτή η αναγνώριση είναι λάθος. Σε περίπτωση που επιλέξει ότι δεν είναι λάθος η εφαρμογή αυτόματα θα ενημερώσει το είδος του ζώου με αυτό που προβλέφθηκε.



Ο χρήστης έχει την δυνατότητα να επιλέξει την τοποθεσία στην οποία είδε το ζώο είτε επιλέγοντας την πάνω σε έναν χάρτη ή χρησιμοποιώντας την θέση του μέσω του GPS εάν έχει δώσει στην εφαρμογή δικαίωμα πρόσβασης στην τοποθεσία του.



Εάν ο χρήστης θελήσει να διαλέξει την θέση της δημοσίευσης από τον χάρτη θα πρέπει να διαλέξει την αντίστοιχη επιλογή η οποία θα ανοίξει τον χάρτη έχοντας στο κέντρο ένα σημάδι για την θέση. Ο χρήστης θα πρέπει να μετακινήσει τον χάρτη και να τον μεγεθύνει για να φέρει το σημάδι στην τοποθεσία που θέλει και όταν το έχει κάνει θα πρέπει να πατήσει “Select Location” για να τελειώσει η επιλογή τοποθεσίας. Ανεξάρτητα από τον τρόπο με τον οποίο ο χρήστης επέλεξε την τοποθεσία, μετά το τέλος της επιλογής θα εμφανιστεί η περιοχή που επέλεξε στο κάτω μέρος της οθόνης όπως φαίνεται στην δεξιά εικόνα παραπάνω.



5:36


← Encountered Post ✓

Animal Type
🐾 Fox

Comment

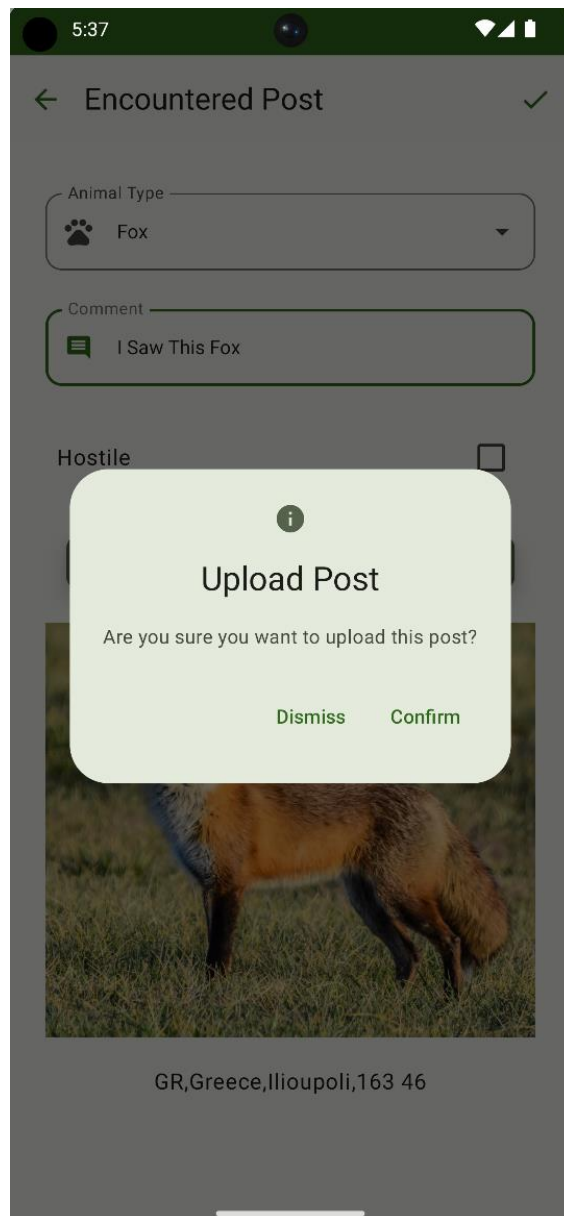
Hostile

Select Image Pick Location

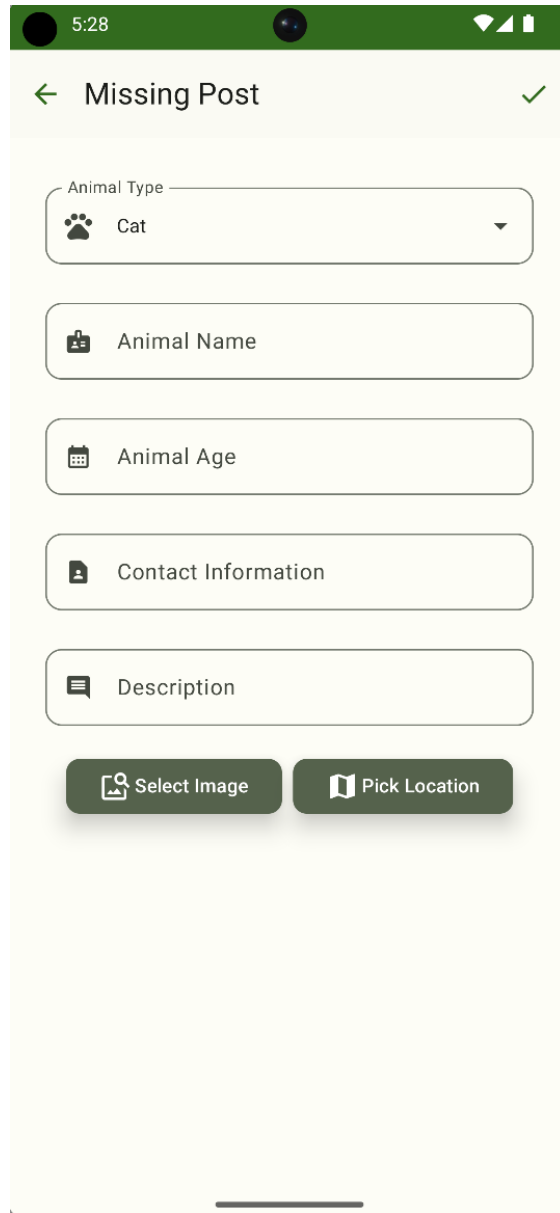


You Haven't Properly Filled Out All The Necessary Fields

Εάν ένας χρήστης προσπαθήσει να δημιουργήσει μια δημοσίευση χωρίς να έχει συμπληρώσει όλα τα απαιτούμενα πεδία θα λάβει ένα μήνυμα που τον προτρέπει να το κάνει προτού ξανά προσπαθήσει.

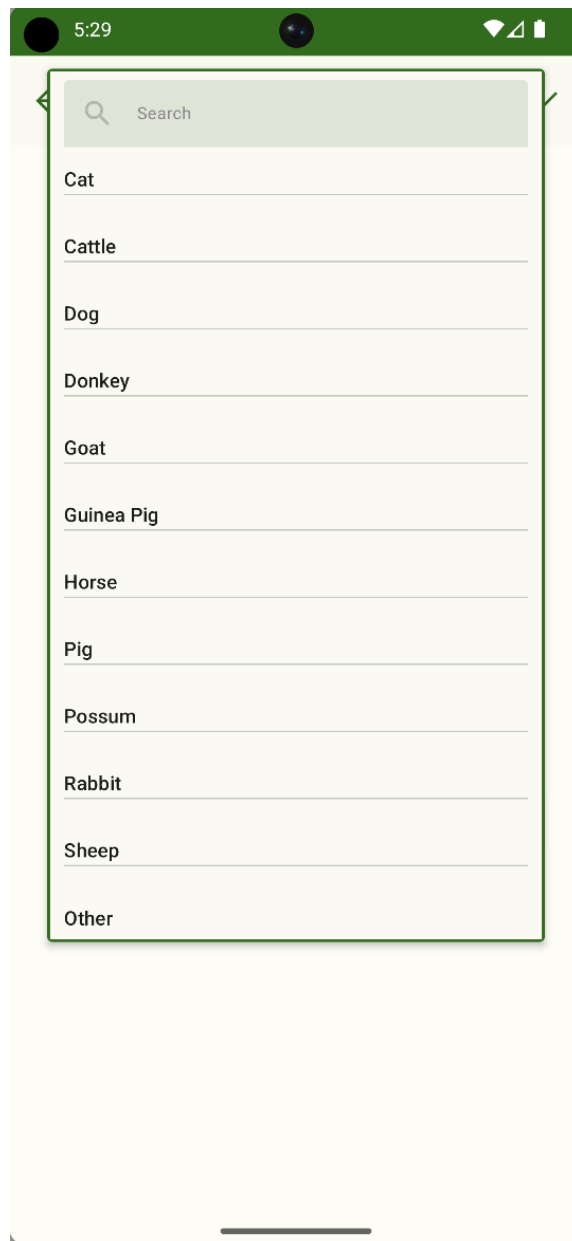


Όταν ο χρήστης συμπληρώσει όλα τα απαιτούμενα πεδία μπορεί να ανεβάσει την δημοσίευση του στην εφαρμογή πατώντας το κουμπί στο πάνω δεξιά μέρος της οθόνης. Αφού το πατήσει θα λάβει ένα τελευταίο μήνυμα που τον ρωτάει αν είναι σίγουρος ότι θέλει να την δημιουργήσει και έχει την επιλογή να το κάνει ή να ακυρώσει την δημιουργία.

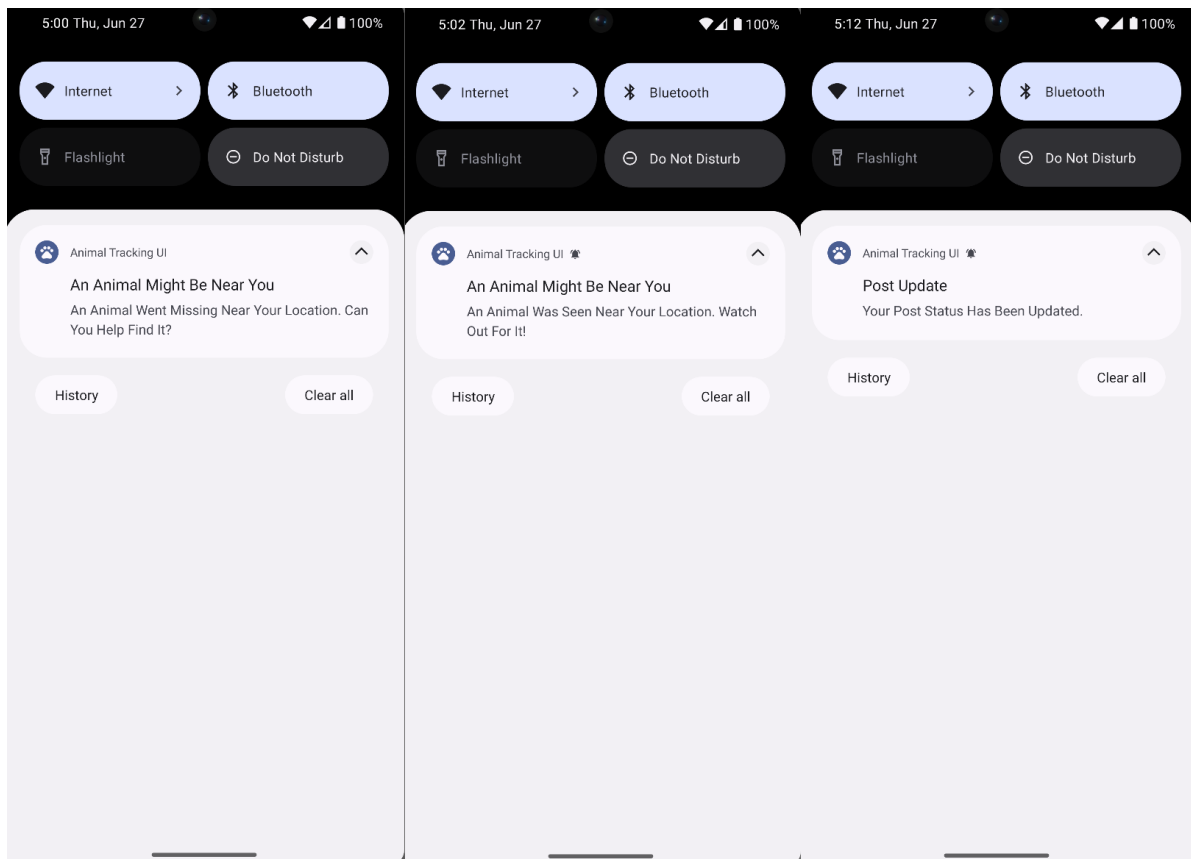


The screenshot displays a mobile application interface for creating a 'Missing Post'. At the top, there is a green status bar with the time 5:28 and standard Android navigation icons. Below the status bar is a white header with a back arrow, the text 'Missing Post', and a green checkmark. The main content area is a light yellow background with several white rounded rectangular input fields. The first field is labeled 'Animal Type' and contains a paw print icon and the text 'Cat'. The subsequent fields are labeled 'Animal Name', 'Animal Age', 'Contact Information', and 'Description'. At the bottom of the form, there are two dark green buttons: 'Select Image' with a camera icon and 'Pick Location' with a location pin icon.

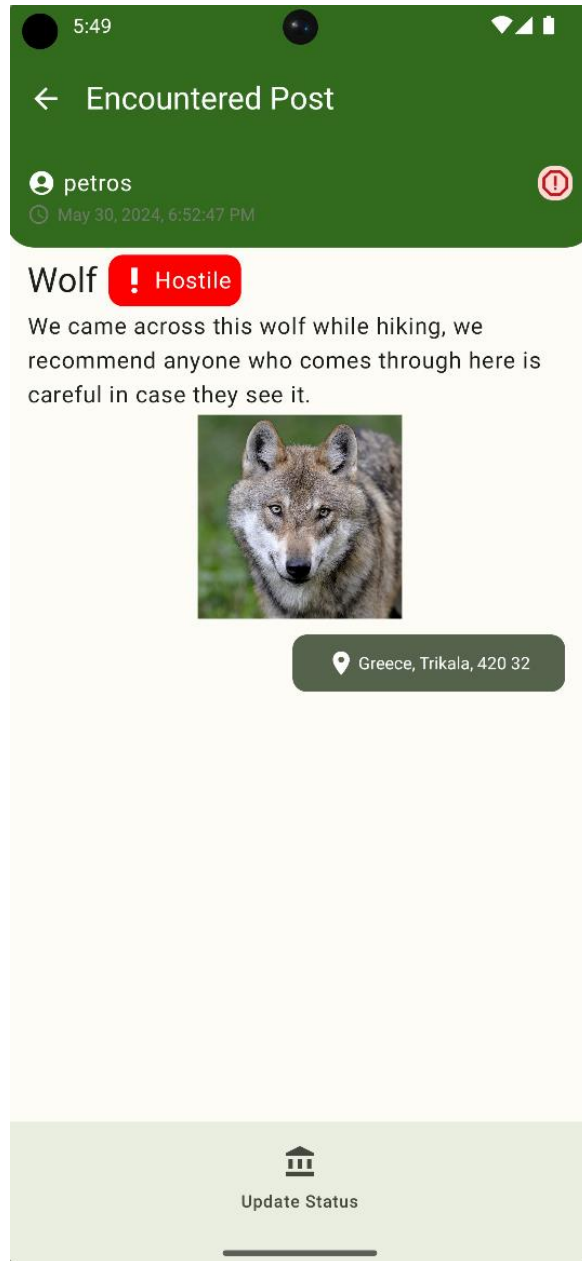
Μια δημοσίευση για χαμένο κατοικίδιο είναι παρόμοια με αυτήν για ζώο που έχει συναντήσει ένας χρήστης, με τις διαφορές να είναι μερικά παραπάνω πεδία τα οποία του ζητούνται να συμπληρώσει όπως είναι το όνομα του κατοικίδιου, η ηλικία του, στοιχεία επικοινωνίας με τον χρήστη και μια περιγραφή για την δημοσίευση που κάνει ενώ δεν χρειάζεται να εισάγει εάν το ζώο είναι επιθετικό ή όχι.



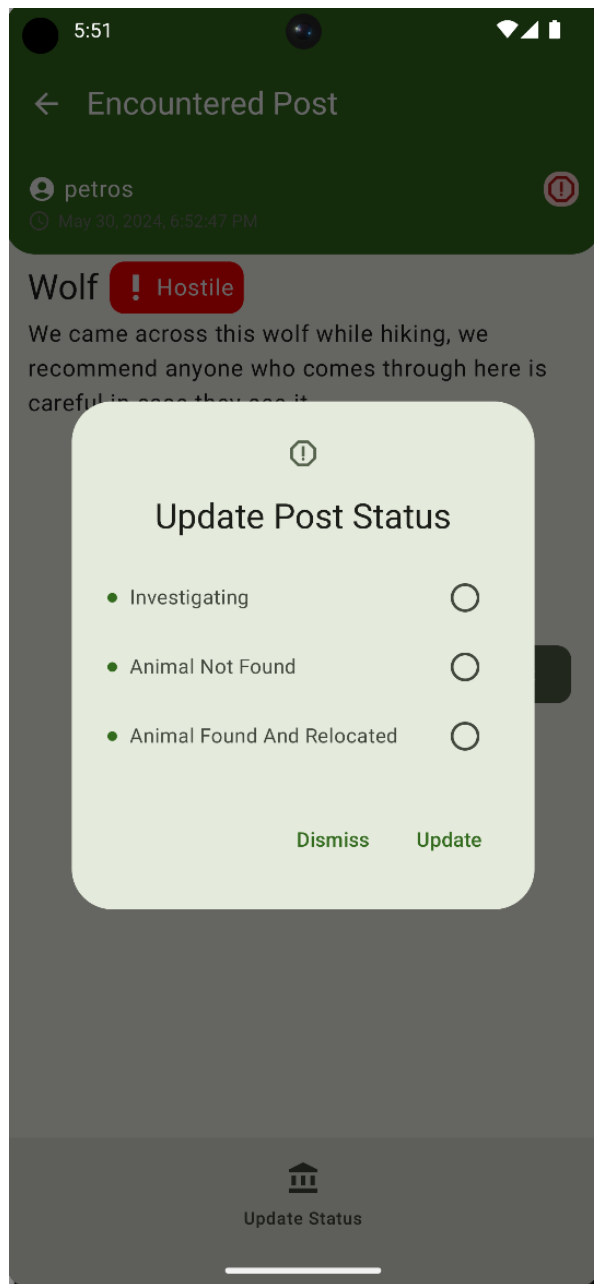
Οι επιλογές για το είδος του ζώου αλλάζουν στην περίπτωση του χαμένου κατοικίδιου, παράλα αυτά ο χρήστης έχει και πάλι την δυνατότητα να εισάγει χειροκίνητα το είδος του ζώου εάν αυτό δεν υπάρχει στην λίστα.



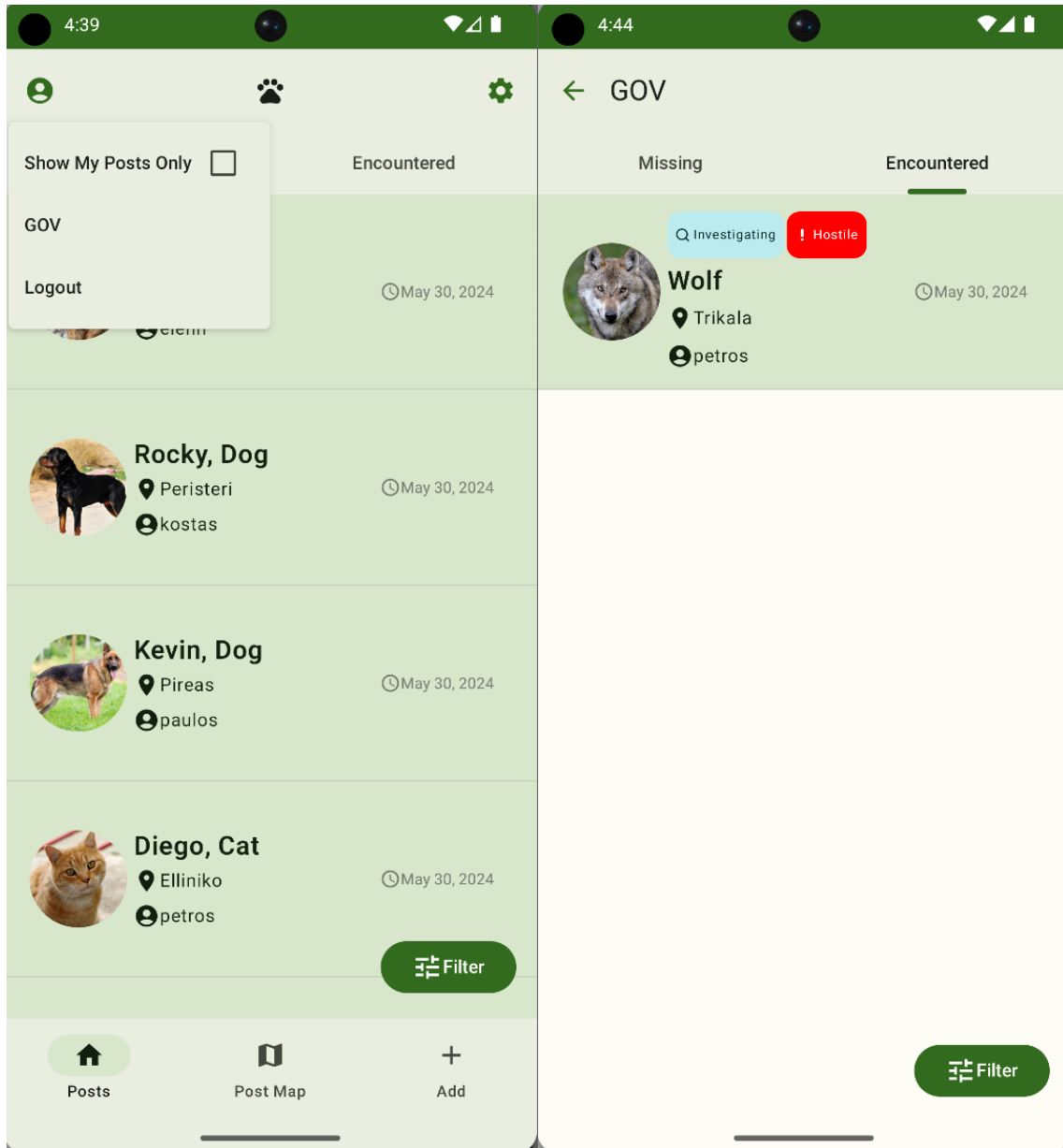
Οι χρήστες μπορούν να λάβουν ένα από τα τρία είδη ειδοποιήσεων που έχει η εφαρμογή, με βάση και τις επιλογές ειδοποιήσεων που έχουν κάνει. Η ειδοποίηση που φαίνεται αριστερά αφορά δημοσιεύσεις για χαμένα κατοικίδια που βρίσκονται σε μέγιστη απόσταση 5 χιλιομέτρων από την θέση του χρήστη ενώ η μεσαία για ζώα που έχει δει κάποιος χρήστης πάλι σε απόσταση 5 χιλιομέτρων από την θέση του χρήστη. Η ειδοποίηση που φαίνεται δεξιά στέλνεται προσωπικά σε έναν χρήστη όταν ένα μέλος της τοπικής κυβέρνησης ενημερώσει την κατάσταση της δημοσίευσής του. Πατώντας πάνω σε οποιαδήποτε από τις ειδοποιήσεις ο χρήστης θα μεταφερθεί στην οθόνη προβολής της δημοσίευσής στην οποία αντιστοιχεί.



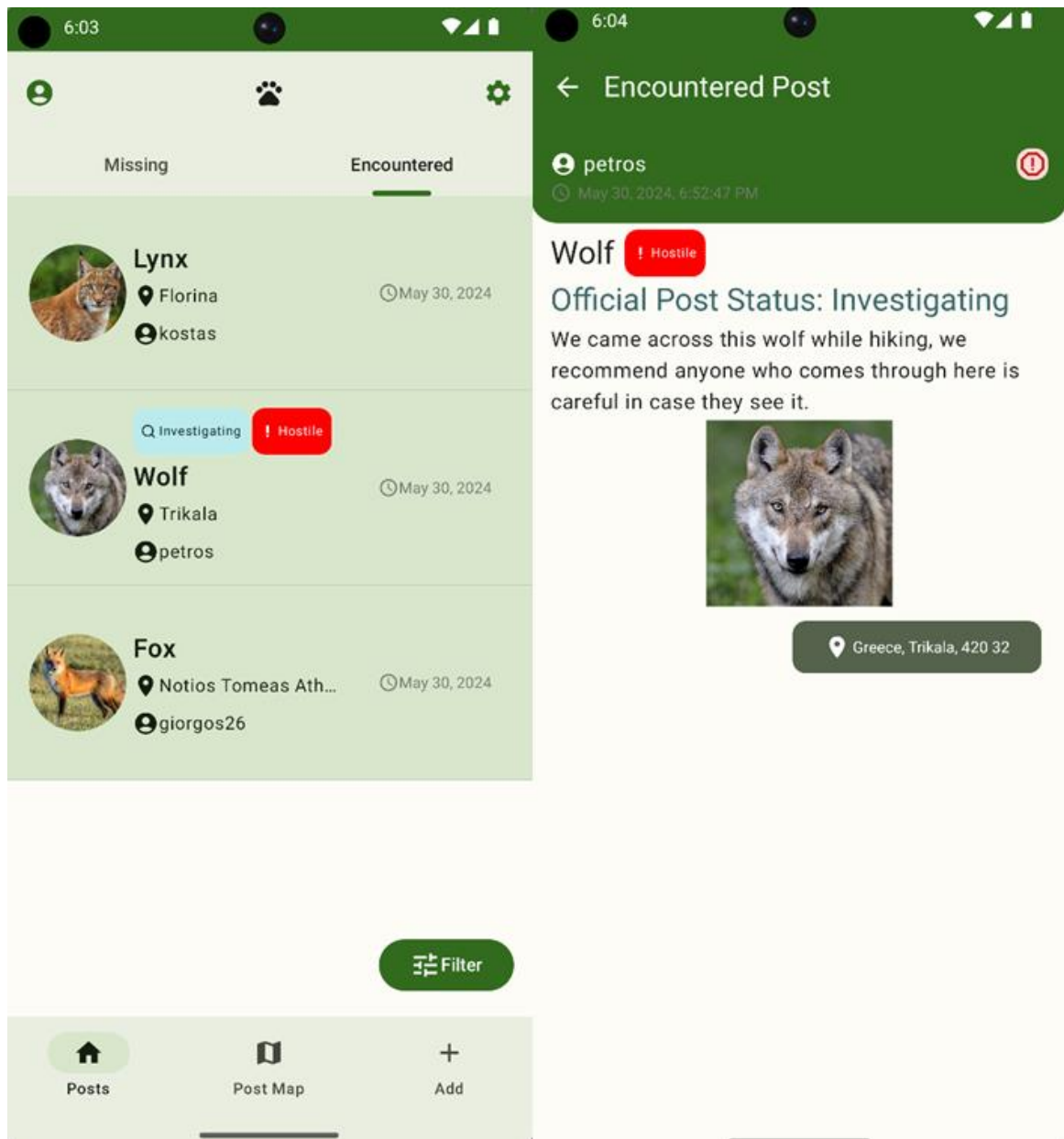
Η εφαρμογή δίνει την δυνατότητα και σε μέλη της τοπικής κυβέρνησης να έχουν πρόσβαση σε αυτήν με σκοπό να ενημερώνουν τους χρήστες για τις ενέργειές που έχουν λάβει για την διαχείριση της κατάστασης που περιγράφει μια δημοσίευση. Ένας χρήστης του οποίου ο λογαριασμός έχει οριστεί ως μέλος της τοπικής κυβέρνησης έχει την επιλογή να ενημερώσει την κατάσταση μιας δημοσίευσης όπως φαίνεται από την παραπάνω εικόνα.



Μια δημοσίευση μπορεί να έχει μια από τις τρεις καταστάσεις “Ερευνάται”, “Το Ζώο Δεν Βρέθηκε” και “Το Ζώο Βρέθηκε Και Μετακινήθηκε”. Τα μέλη της τοπικής κυβέρνησης μπορούν να επιλέξουν μια από αυτές τις καταστάσεις για να ενημερώσουν τον χρήστη που δημιούργησε την δημοσίευση αλλά και τους υπόλοιπους χρήστες που θα την δούνε για την κατάσταση του ζώου.



Οι χρήστες που είναι συνδεδεμένοι ως χρήστες με ρόλο μέλους τοπικής κυβέρνησης έχουν πρόσβαση στην σε μια οθόνη η οποία περιέχει μόνο δημοσιεύσεις των οποίων η κατάσταση έχει ενημερωθεί είτε από αυτούς ή από άλλο μέλος της τοπικής κυβέρνησης.



Μια δημοσίευση που έχει ενημερωθεί από μέλος της τοπικής κυβέρνησης λαμβάνει μια αντίστοιχη ετικέτα στην λίστα των δημοσιεύσεων αλλά και στην οθόνη προβολής της εκάστοτε δημοσίευσης.

5. Συμπεράσματα Και Μελλοντικές Επεκτάσεις

Η εφαρμογή που δημιουργήσαμε προσφέρει την δυνατότητα στους χρήστες της να ενημερώνονται εύκολα ανά πάσα στιγμή για χαμένα κατοικίδια και για ζώα που έχουν εντοπιστεί από άλλους χρήστες στην περιοχή τους. Χρησιμοποιώντας την εφαρμογή ένας χρήστης μπορεί να βοηθήσει κάποιον να βρει το κατοικίδιο του, να αναζητήσει ο ίδιος το κατοικίδιο του και να ενημερώνει και να ενημερώνεται σχετικά με ζώα που έχουν βρεθεί στην περιοχή του και τα οποία μπορεί να είναι επικίνδυνα. Η εφαρμογή δίνει επιπλέον την

δυνατότητα στους χρήστες να αναγνωρίσουν τι ζώο βλέπουνε βγάζοντας το μια φωτογραφία με την χρήση της τεχνητής νοημοσύνης.

Η εφαρμογή μας θα μπορούσε να επεκταθεί με διάφορους τρόπους έτσι ώστε να βοηθάει περισσότερο τους χρήστες της. Μερικές από αυτές τις επεκτάσεις είναι οι εξής:

- **Επέκταση Δυνατοτήτων Αναγνώρισης Ζώων:** Προσθήκη περισσότερων ζώων στο μοντέλο αναγνώρισης από εικόνες αλλά και δυνατότητας αναγνώρισης ράτσας ζώου.
- **Προφίλ Χρηστών:** Μια οθόνη στην οποία φαίνονται μερικά στοιχεία ενός χρήστη μαζί με τις δημοσιεύσεις που έχει κάνει. Εφόσον ο χρήστης έχει επιλέξει να κάνει την σελίδα προφίλ του δημόσια θα έχουν πρόσβαση σε αυτήν όλοι οι χρήστες της εφαρμογής. Ένας χρήστης θα μπορεί να προσθέτει και μερικές χρήσιμες πληροφορίες στο προφίλ του όπως τις τροφές που προτιμάει για τα κατοικίδια του ή έναν κτηνίατρο που προτείνει.
- **Απευθείας Μηνύματα:** Δυνατότητα άμεσης επικοινωνίας μεταξύ χρηστών της εφαρμογής για να μην υπάρχει ανάγκη να παρέχουν προσωπικά στοιχεία επικοινωνίας.
- **Αποθήκευση Κατοικίδιων:** Όλοι οι εγγεγραμμένοι χρήστες θα έχουν την δυνατότητα να προσθέτουν τα κατοικίδια τους στην εφαρμογή με κάποια βασικά στοιχεία σχετικά με αυτά. Τα κατοικίδια αυτά θα φαίνονται στην σελίδα προφίλ ενός χρήστη και σε περίπτωση που κάποιο από αυτά χαθεί θα μπορεί αμέσως να το επιλέξει μαζί με την τοποθεσία στην οποία χάθηκε και να δημιουργήσει μια δημοσίευση, προσπερνώντας την διαδικασία εισαγωγής στοιχείων για την δημιουργία της.
- **Ημερολόγιο Επιτευγμάτων/Υποχρεώσεων:** Προσθήκη ενός ημερολογίου για κάθε κατοικίδιο του χρήστη στο οποίο αυτός θα μπορεί να αποθηκεύει στιγμές με το κατοικίδιο του που θέλει να θυμάται καθώς και να σημειώνει υποχρεώσεις που έχει όπως για παράδειγμα επισκέψεις σε κτηνίατρο.
- **Φόρουμ:** Δημιουργία ενός φόρουμ στο οποίο θα έχουν πρόσβαση οι χρήστες για να κάνουν συζητήσεις σχετικές με κατοικίδια και γενικότερα με ζώα.

6. Βιβλιογραφία

Kotlin:

<https://kotlinlang.org/>

Kotlin Coroutines:

<https://developer.android.com/kotlin/coroutines>

Declarative Programming:

https://en.wikipedia.org/wiki/Declarative_programming

Jetpack Compose:

<https://developer.android.com/develop/ui/compose>

Jetpack Compose Recomposition:

<https://developer.android.com/develop/ui/compose/mental-model#recomposition>

Compose View Models:

<https://developer.android.com/topic/libraries/architecture/viewmodel>

Spring:

<https://spring.io/>

Spring JPA:

<https://docs.spring.io/spring-data/jpa/reference/jpa.html>

Hibernate:

<https://hibernate.org/>

Tensorflow:

<https://www.tensorflow.org/>

Convolutional Neural Networks:

<https://www.ibm.com/topics/convolutional-neural-networks>

ResNet50:

<https://blog.roboflow.com/what-is-resnet-50/>

Crowdsourcing Recognized Image Objects In Mobile Devices Through Machine Learning:

<https://ieeexplore.ieee.org/document/9643157>

Supervised machine learning models for student performance prediction;

<https://content.iospress.com/articles/intelligent-decision-technologies/idt210251>

Machine Learning Paradigms:

https://link.springer.com/chapter/10.1007/978-3-030-13743-4_1