

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Όνομα εφαρμογής Android :

PlantApp

Μάθημα:

Ανάπτυξη Εφαρμογών Κινητών Συσκευών

Διδάσκων: Κατσάνος Χρήστος

Μέλη ομάδας:

- Μιχαλίτσος Σπύρος, Α.Ε.Μ.: 3211
- Παπαδόπουλος Ξενοφών, Α.Ε.Μ.: 3200
- Τερζής Ηλίας, Α.Ε.Μ.: 3170

ΘΕΣΣΑΛΟΝΙΚΗ
Εαρινό εξάμηνο 2019-2020

Πίνακας περιεχομένων

Εισαγωγή.....	3
Γενική περιγραφή.....	3
Σκοπός και κίνητρα.....	3
Τυπικοί χρήστες της εφαρμογής.....	3
Λειτουργικότητα της εφαρμογής.....	4
Αρχιτεκτονική της εφαρμογής.....	12
Περιγραφή του κώδικα της εφαρμογής.....	13
Σημαντικά σημεία.....	14
Ιδέες για μελλοντική εξέλιξη.....	15

Εισαγωγή

Γενική περιγραφή

Η εφαρμογή PlantApp αποτελεί εργαλείο βοήθειας για την οργάνωση και την φροντίδα των φυτών κήπου ή βεράντας. Δίνει την δυνατότητα στον χρήστη να αποθηκεύει σε μια βάση δεδομένων όσα φυτά διατηρεί στον πραγματικό κόσμο έτσι ώστε να καταγράφει βασικές πληροφορίες τους, οι οποίες είναι απαραίτητες για την καλύτερη διαχείρισή τους. Στη βάση δεδομένων δημιουργούνται εγγραφές με το όνομα του κάθε φυτού, την ημερομηνία που φυτεύτηκε καθώς και την ημερομηνία της τελευταίας φοράς που λιπάνθηκε. Όποτε το επιθυμεί ο χρήστης-κηπουρός μπορεί να ανατρέξει στα περιεχόμενα της βάσης για επισκόπηση των δεδομένων που έχει καταγράψει, για να διαγράψει ένα φυτό ή να δει τις πληροφορίες μεμονωμένα για αυτό το φυτό. Τέλος, του δίνεται η δυνατότητα να ενημερώσει την ημερομηνία λίπανσης, είτε με μαζικό τρόπο για όλα τα φυτά είτε για κάποιο συγκεκριμένο φυτό.

Σκοπός και κίνητρα

Η εφαρμογή PlantApp δημιουργήθηκε για να αποτελέσει εργαλείο για τους ανθρώπους που φροντίζουν φυτά στον κήπο ή στο μπαλκόνι τους. Η ιδέα προέρχεται κυρίως από την αγάπη και το μεράκι των μελών της ομάδας για την κηπουρική αλλά και από το γεγονός ότι μια εφαρμογή διαχείρισης κήπου λείπει από το ελληνικό αποθετήριο εφαρμογών Android. Πιστεύουμε ότι μια εφαρμογή διαχείρισης κήπου είναι απαραίτητη για τον σύγχρονο κηπουρό καθώς με αυτό τον τρόπο μπορεί να εκμεταλλευτεί τις δυνατότητες που του προσφέρει η κινητή συσκευή του, ώστε να διευκολύνεται στην οργάνωση της φροντίδας των φυτών του. Με την αποθήκευση των δεδομένων του και την εύκολη και αποδοτική διαχείρισή τους η ενασχόλησή του με τα φυτά μπορεί να γίνει ακόμη πιο αποτελεσματική και να επικεντρωθεί στα πιο σημαντικά σημεία της που τον ευχαριστούν περισσότερο. Πιστεύουμε ότι ο κηπουρός αξίζει να έχει στη διάθεσή του και ψηφιακά εργαλεία εκτός από τα υλικά (π.χ. φτυάρια και τσάπες) που να τον βοηθούν καθώς απασχολείται με την κηπουρική.

Τυπικοί χρήστες της εφαρμογής

Η εφαρμογή απευθύνεται κυρίως σε ερασιτέχνες κηπουρούς ή ακόμη και σε οποιονδήποτε που διατηρεί μερικές γλάστρες με φυτά στο σπίτι του και θέλει να αξιοποιήσει το κινητό του και σε αυτό το κομμάτι της καθημερινότητάς του, καταγράφοντας τις πληροφορίες των φυτών που φροντίζει.

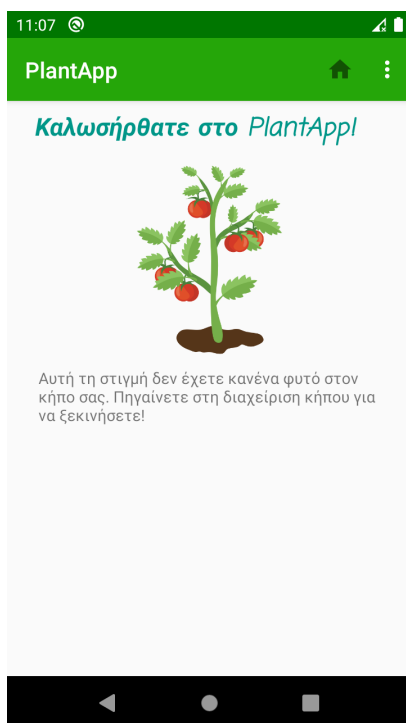
Είναι γεγονός ότι το αντικείμενο της εφαρμογής PlantApp είναι τέτοιο, που δημιουργούνται πάρα πολλές ιδέες για μεγαλύτερη αξιοποίηση της συσκευής στην υπηρεσία του σκοπού της (βλ. ενότητα

“Ιδέες για μελλοντική εξέλιξη”). Αν αυτές οι ιδέες γίνουν πραγματικότητα κάποια στιγμή, αναμφίβολα το PlantApp θα μπορούσε να αποτελέσει σημαντικό βοήθημα για κάθε γεωργό.

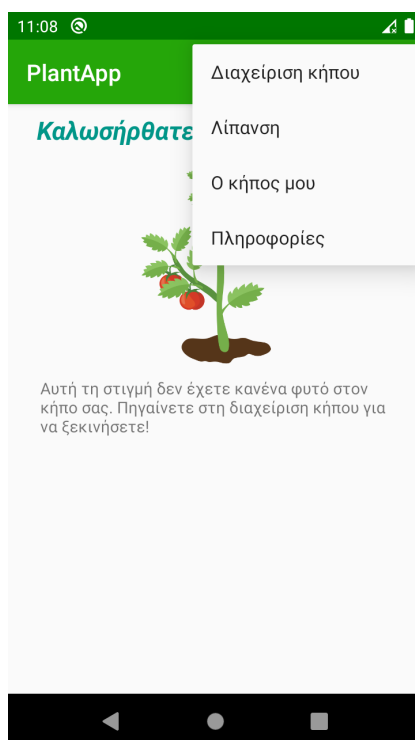
Λειτουργικότητα της εφαρμογής

Στη συνέχεια παρουσιάζεται ο τρόπος χρήσης της εφαρμογής με τη χρήση στιγμιοτύπων οθόνης (screenshots) από την εκτέλεσή της στον εξομοιωτή κανονικού μεγέθους “Galaxy Nexus API 29” των 4,65”.

- Στο πρώτο screenshot παρουσιάζεται η αρχική οθόνη που βλέπει ο χρήστης πριν προσθέσει κάποιο φυτό στη βάση δεδομένων:



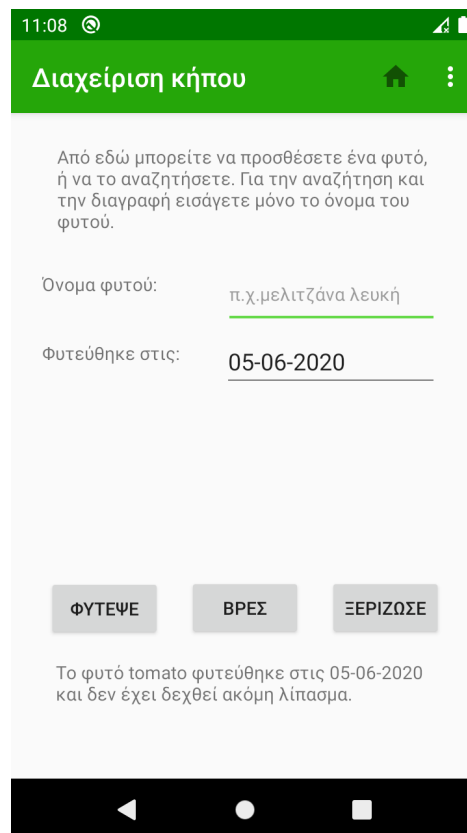
- Το μενού που εμφανίζεται φαίνεται παρακάτω:



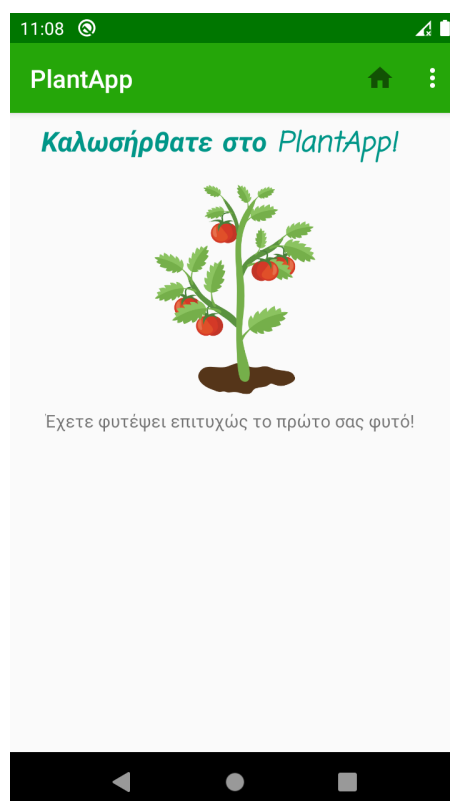
- Στη συνέχεια βλέπουμε την οθόνη διαχείρισης κήπου όπου ο χρήστης μπορεί να προσθέσει, αναζητήσει και να διαγράψει ένα φυτό από τη βάση δεδομένων. Όπου ζητείται ημερομηνία για είσοδο στην εφαρμογή αρχικοποιείται αυτόματα με την τρέχουσα ημερομηνία αλλά μπορεί φυσικά να αλλάξει από τον χρήστη.

- Με την επιτυχή εισαγωγή του φυτού εμφανίζεται κατάλληλο μήνυμα.

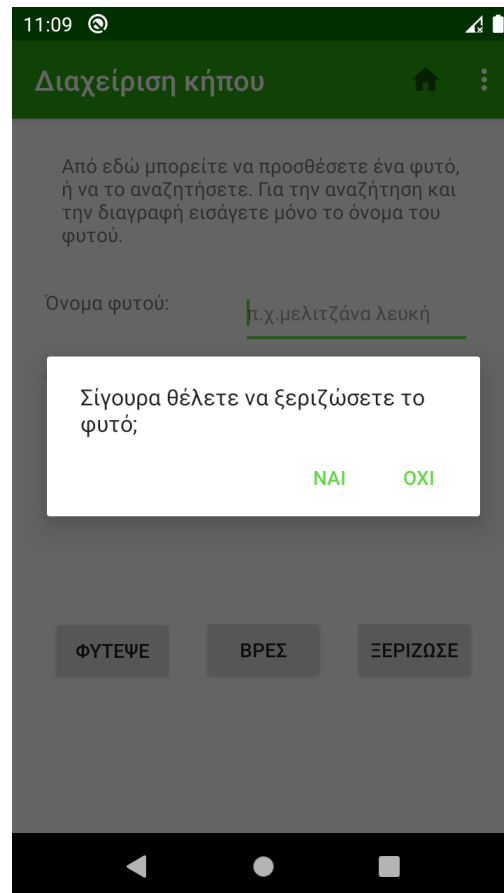
- Η αναζήτηση του φυτού με το όνομά του, κάνει ορατό ένα textView για την εμφάνιση των πληροφοριών του.



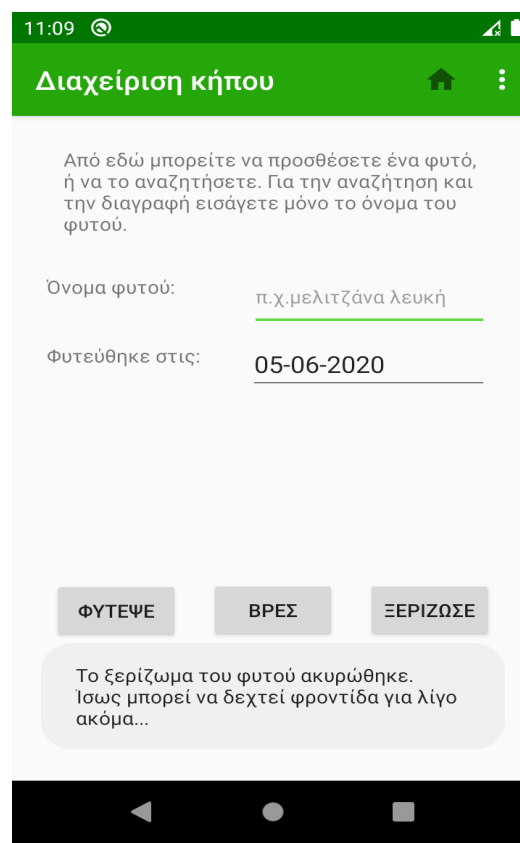
- Το μήνυμα της αρχικής οθόνης προσαρμόζεται και ο χρήστης ενημερώνεται ότι στην βάση δεδομένων υπάρχει πλέον ένα φυτό.



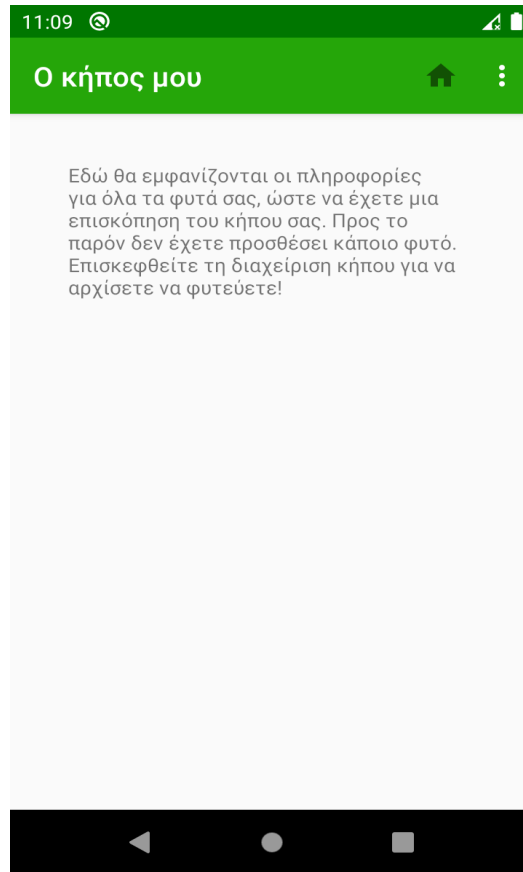
- Όταν ο χρήστης επιλέξει την διαγραφή φυτού εμφανίζεται διάλογος επιβεβαίωσης.



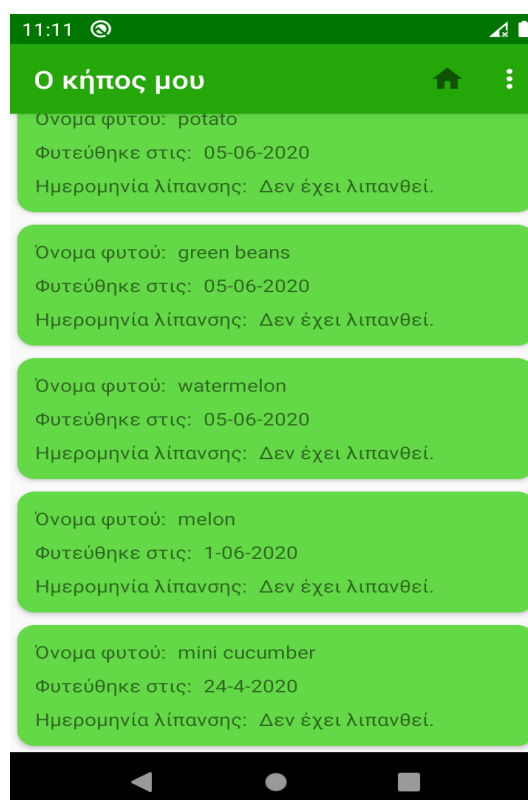
- Εάν στο παράθυρο που εμφανίζεται η επιλογή του χρήστη είναι αρνητική, η διαδικασία ακυρώνεται και εμφανίζεται σχετικό μήνυμα.



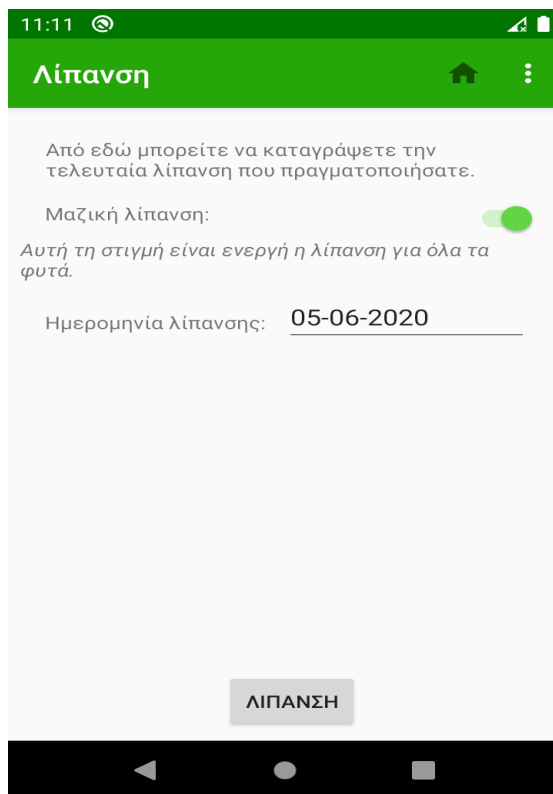
- Στη συνέχεια παρουσιάζεται η οθόνη που βλέπει ο χρήστης εάν επισκεφθεί την επιλογή “Ο κήπος μου” για να δει ουσιαστικά τα περιεχόμενα της βάσης δεδομένων. Εδώ βλέπουμε το μήνυμα που εμφανίζεται όταν δεν υπάρχει κάποια εγγραφή στη βάση.



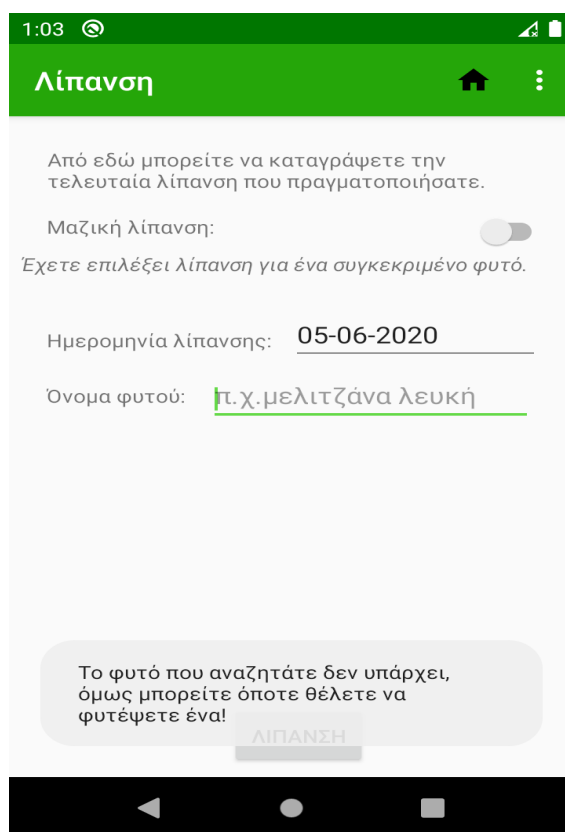
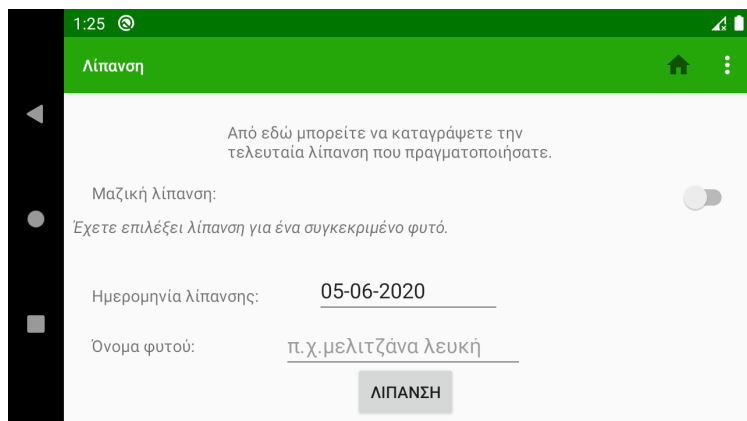
Τώρα βλέπουμε τον τρόπο με τον οποίο εμφανίζονται τα περιεχόμενα της βάσης στην οθόνη: χρησιμοποιούνται CardViews που εμφανίζονται σε ένα RecyclerView.



- Στα επόμενα screenshots παρακολουθούμε την διαδικασία που ακολουθεί ο χρήστης για την λίπανση. Η προεπιλογή είναι η μαζική λίπανση του κήπου αλλά με το αντικείμενο switch μπορεί ο χρήστης να μεταβεί σε κατάσταση ατομικού λιπάσματος φυτού.



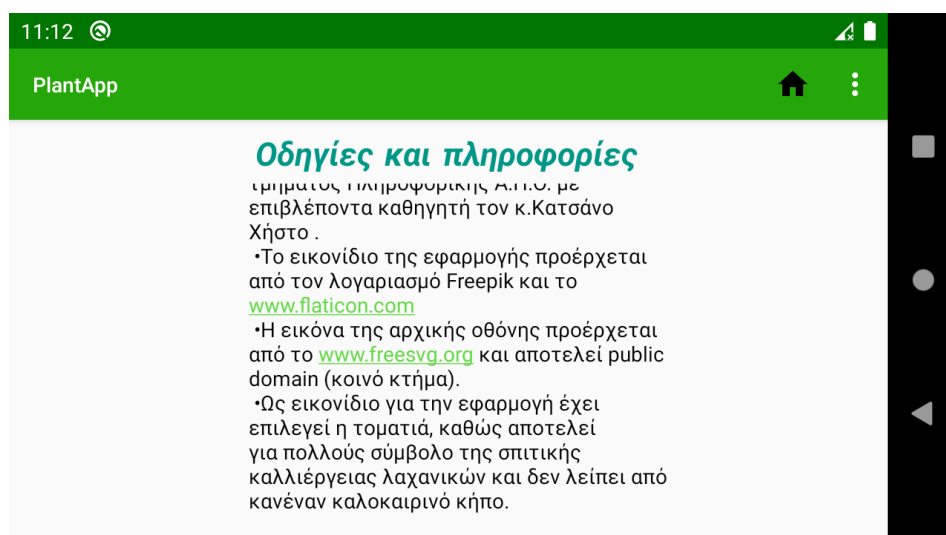
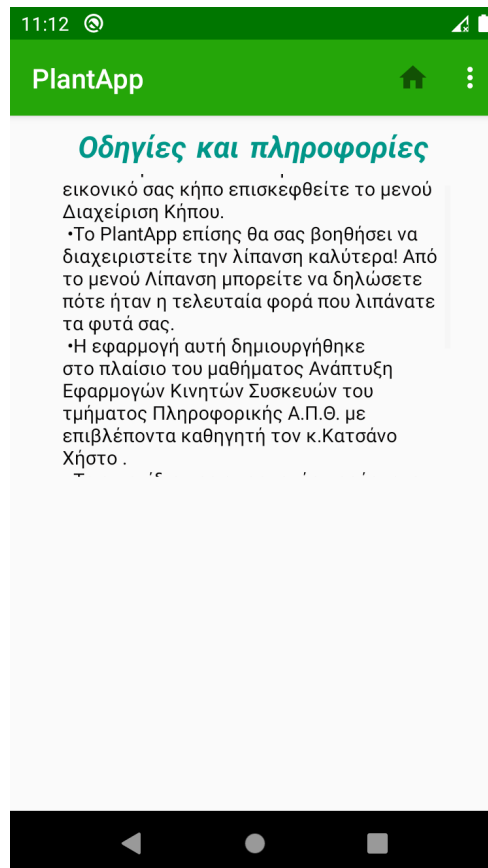
- Βλέπουμε τώρα πώς φαίνεται η οθόνη στην ατομική λίπανση σε κάθετο και οριζόντιο προσανατολισμό και το μήνυμα που εμφανίζεται στο χρήστη στην περίπτωση που εισάγει όνομα φυτού που δεν έχει εισαχθεί στη βάση (το πεδίο εισαγωγής ονόματος γίνεται κενό).



- Το μήνυμα στην αρχική οθόνη είναι διαρκώς ενημερωμένο γιατί ο αριθμός των φυτών παίρνεται από τη βάση κάθε φορά που εκκινείται η δραστηριότητα. Εδώ παρατίθεται στιγμιότυπο της οθόνης τόσο σε οριζόντιο όσο και σε κάθετο προσανατολισμό.



- Τέλος, βλέπουμε την οθόνη “Πληροφορίες”. Έχει δοθεί η δυνατότητα στο κείμενο να είναι scrollable, να μπορεί δηλαδή ο χρήστης να μετακινεί πάνω και κάτω την προεπισκόπηση του κειμένου. Κάναμε αυτή την επιλογή ώστε να μην χάνεται μέρος από την εμφάνιση του κειμένου σε συσκευές διαφορετικών μεγεθών, όπως συνέβαινε σε κάποιες περιπτώσεις όσο είχαμε το μέγεθος του textView καθορισμένο. Επίσης δεν δώσαμε στο κείμενο όλη την περιοχή της οθόνης γιατί σε αυτήν την περίπτωση υπήρχε πρόβλημα στην εμφάνισή του όταν το κινητό έχει οριζόντιο προσανατολισμό.



Αρχιτεκτονική της εφαρμογής

Η εφαρμογή αποτελείται από πέντε δραστηριότητες (Activities) :

1. **MainActivity:** Η δραστηριότητα αυτή υλοποιεί την λειτουργικότητα του μενού και όλες οι κλάσεις των υπόλοιπων δραστηριοτήτων της εφαρμογής που το περιλαμβάνουν επεκτείνουν αυτή την κλάση. Επίσης, είναι υπεύθυνη για την εμφάνιση των συστατικών της αρχικής οθόνης της εφαρμογής: την εικόνα, το μήνυμα καλωσορίσματος και την εμφάνιση του πλήθους των φυτών που έχει καταχωρήσει ο χρήστης στη βάση.
2. **DataBaseActivity:** Η δραστηριότητα αυτή αναλαμβάνει να υλοποιήσει τις απαραίτητες λειτουργίες διαχείρισης της βάσης δεδομένων. Από εδώ ο χρήστης μπορεί να προσθέσει στη βάση δεδομένων μία εγγραφή για ένα νέο φυτό, να εντοπίσει το φυτό αυτό στη βάση ώστε να εμφανιστούν οι πληροφορίες του, καθώς και να το διαγράψει από τη βάση. Για τις δυνατότητες αυτές η κλάση της δραστηριότητας χρησιμοποιεί την MyDBHandler, που είναι η κλάση διαχείρισης της βάσης.
3. **FertilizerActivity:** Η δραστηριότητα αυτή δίνει τη δυνατότητα στο χρήστη να ενημερώσει το περιεχόμενο της στήλης “Ημερομηνία λίπανσης” του πίνακα της βάσης δεδομένων, έτσι ώστε να δηλώσει πότε ήταν η τελευταία φορά που λίπανε τα φυτά του. Η λίπανση μπορεί να γίνει είτε σε ένα συγκεκριμένο φυτό αναζητώντας το με το όνομά του είτε σε όλα τα φυτά που έχουν καταχωρηθεί. Η επιλογή του χρήστη για τον τρόπο λίπανσης αλλάζει με ένα στοιχείο διακόπτη Switch.
4. **GalleryActivity:** Αυτή η δραστηριότητα αναλαμβάνει να εμφανίσει στον χρήστη τα περιεχόμενα της βάσης δεδομένων χρησιμοποιώντας ένα RecyclerView που περιλαμβάνει CardViews, ένα για κάθε φυτό. Για την λειτουργία της, αυτή η δραστηριότητα χρησιμοποιεί την κλάση RecyclerViewAdapter που υλοποιεί τον μηχανισμό του RecyclerView.
5. **AboutActivity:** Μέσω αυτής της δραστηριότητας εμφανίζονται στο χρήστη οι πληροφορίες της εφαρμογής και οι οδηγίες χρήσης της.

Οι επιπλέον κλάσεις (εκτός από τις κλάσεις των δραστηριοτήτων που αναφέρθηκαν) είναι οι εξής:

1. **Plant:** Σύμφωνα με το μοντέλο MVC (Model, View, Controller) που παρουσιάστηκε στο μάθημα, η κλάση αυτή αποτελεί την κλάση μοντέλου δεδομένων που είναι απαραίτητη για την εφαρμογή. Ουσιαστικά αναπαριστά το αντικείμενο του φυτού περιλαμβάνοντας τις απαραίτητες για αυτό πληροφορίες όπως είναι το όνομα και ημερομηνία φύτευσής του. Τα αντικείμενα αυτής της κλάσης αποθηκεύονται ως εγγραφές στον πίνακα της βάσης δεδομένων.
2. **MyDBHandler:** Έχοντας το ρόλο του controller, η κλάση αυτή χρησιμοποιείται για να πραγματοποιούνται όλες εκείνες οι λειτουργίες διαχείρισης της βάσης δεδομένων τις οποίες επιθυμεί να εκτελέσει ο χρήστης αλληλεπιδρώντας με την DatabaseActivity. Για τη δημιουργία αυτής της κλάσης ακολουθήθηκε ο τρόπος δόμησης που παρουσιάστηκε στο κεφάλαιο 4 του μαθήματος με το παράδειγμα δημιουργίας εφαρμογής διαχείρισης βάσης δεδομένων. Εκτός από τις μεθόδους για την προσθήκη, την αναζήτηση και την διαγραφή των δεδομένων από τη βάση, κάποιες επιπλέον μέθοδοι δημιουργήθηκαν για τις ανάγκες της

εφαρμογής, όπως για παράδειγμα για την τροποποίηση και ενημέρωση του περιεχομένου της στήλης “Ημερομηνία Λίπανσης” του πίνακα.

3. **RecyclerAdapter:** Αυτή είναι η κλάση που υποχρεωτικά πρέπει να περιλαμβάνεται στην εφαρμογή για να μπορεί να εμφανίζεται το RecyclerView στην δραστηριότητα “GalleryActivity”.

Περιγραφή του κώδικα της εφαρμογής

Στη συνέχεια αναλύονται κάποια σημαντικά σημεία του project και του κώδικα των κλάσεων που αναφέρθηκαν παραπάνω. Σε αυτό το σημείο πρέπει να αναφερθεί ότι για την υλοποίηση αρκετών από των μεθόδων που χρησιμοποιούνται στις κλάσεις - ειδικά για τις λειτουργίες που δεν αναφέρθηκαν στο μάθημα - έγινε αναζήτηση στο διαδίκτυο και άντληση ιδεών και γραμμών κώδικα από διάφορους ιστοτόπους όπως το [stackoverflow](#) και το κανάλι [Coding in Flow](#) στο YouTube.

- **Το μενού της εφαρμογής** υλοποιήθηκε ακολουθώντας τις οδηγίες που βρίσκονται στις διαφάνειες του κεφαλαίου 8 του μαθήματος. Ένα σημαντικό σημείο για το μενού που αξίζει να αναφερθεί είναι πως κάθε κλάση δραστηριότητας για να μπορεί να εμφανίσει το μενού και να περιλαμβάνει τη λειτουργικότητά του κληρονομεί από την MainActivity ώστε να είναι δυνατή η πλοήγηση του χρήστη στην εφαρμογή. Μία μέθοδος έχει προστεθεί σε κάθε μία από αυτές τις δραστηριότητες που περιλαμβάνουν το μενού και χρησιμοποιείται για να δείχνει ως μη επιλέξιμη την επιλογή από το μενού που οδηγεί στην τρέχουσα δραστηριότητα. Αφενός γίνεται τροποποίηση του μενού κατά το χρόνο εκτέλεσης της δραστηριότητας για να μην είναι η επιλογή που αντιστοιχεί σε αυτή τη δραστηριότητα επιλέξιμη αλλά και για την ρύθμιση της εμφάνισης του item μειώνεται ο αριθμός Alpha , ώστε να επικοινωνεί στο χρήστη ότι δεν είναι επιλέξιμο.
- Στον πίνακα της βάσης δεδομένων όλα τα πεδία (εκτός από το πεδίο id) είναι τύπου “TEXT” και για την μετατροπή **από String σε ημερομηνία** και αντίστροφα (όπου είναι απαραίτητο) χρησιμοποιείται η μέθοδος format της κλάσης SimpleDateFormat. Η ημερομηνία φύτευσης εισάγεται από τον χρήστη και κατά την προσθήκη του στη βάση δεδομένων η στήλη “Ημερομηνία λίπανσης” αρχικοποιείται με το μήνυμα “Δεν έχει λιπανθεί.” Στην δραστηριότητα FertilizerActivity αυτή η τιμή αλλάζει από το χρήστη μέσω του κατάλληλου πεδίου εισαγωγής ημερομηνίας.
- Στην **DataBaseActivity** υπάρχουν όλες οι onClick μέθοδοι για τα buttons του layout της δραστηριότητας. Εκεί γίνονται και οι έλεγχοι εισαγωγής δεδομένων, όπως ο έλεγχος για την προϋπαρξη ενός φυτού στη βάση πριν αυτό προστεθεί. Ο χρήστης ενημερώνεται για τα αποτελέσματα των ελέγχων αυτών και των ενεργειών του μέσα από την εμφάνιση Toast μηνυμάτων. Ένα σημείο που μπορούμε να αναφέρουμε εδώ είναι η **εμφάνιση διαλόγου** επιβεβαίωσης πριν την διαγραφή φυτού, καθώς η διαγραφή δεδομένων από τη βάση ως μη αναστρέψιμη ενέργεια καλό είναι να γίνεται ύστερα από την επιβεβαίωση του χρήστη.

- Στην **FertilizerActivity** η εμφάνιση των συστατικών του layout προσαρμόζεται κατά την εκτέλεση της εφαρμογής. Πιο συγκεκριμένα κατά την εκκίνηση της δραστηριότητας, αρχικά τα πεδία εισόδου ονόματος φυτού είναι κρυφά και αυτό γιατί θεωρούμε ότι η μαζική λίπανση του κήπου είναι πιο συχνή ενέργεια από την λίπανση συγκεκριμένου φυτού. Έτσι η εμφάνιση των στοιχείων που απαιτούνται για την λίπανση κάποιου συγκεκριμένου φυτού γίνεται όταν ο χρήστης αλλάξει την κατάσταση του διακόπτη switch έτσι ώστε να δηλώσει ότι θέλει να προβεί στην διενέργεια λίπανσης του φυτού που αναζητά στη βάση δεδομένων.
- Στην κλάση **MyDBHandler** η **ενημέρωση** της στήλης με την ημερομηνία της τελευταίας λίπανσης του πίνακα πραγματοποιείται με την χρήση δύο μεθόδων που δημιουργήθηκαν για την λίπανση στους δύο διαφορετικούς τρόπους της. Εκείνες οι μέθοδοι με τη σειρά τους χρησιμοποίησαν την update της κλάσης SQLiteDatabase.
- Στο αρχείο activity_about.xml που αποτελεί το layout της AboutActivity η **αλλαγή γραμμής** γίνεται δυνατή με την προσθήκη των χαρακτήρων “\ \n” και η προσθήκη των **συνδέσμων** προς τις πηγές των εικόνων με την προσθήκη της ετικέτας συνδέσμου της HTML και της μεθόδου LinkMovementMethod στην onCreate της δραστηριότητας.
- Για την εμφάνιση του **RecyclerView** με τις κάρτες των φυτών, υλοποιήθηκε η μέθοδος getPlants στην βοηθητική κλάση MyDBHandler η οποία δημιουργεί και επιστρέφει ένα αντικείμενο ArrayList που περιέχει τα αντικείμενα των φυτών. Αυτό το ArrayList περνιέται σαν παράμετρος στον κατασκευαστή της κλάσης RecyclerViewAdapter ώστε ο ViewHolder να μπορεί να πάρει ένα-ένα τα φυτά από εκεί, και στη συνέχεια κάνοντας χρήση των μεθόδων get της κλάσης Plant να εμφανίσει τα στοιχεία του κάθε φυτού στην κάρτα που δημιουργείται για αυτό.

Σημαντικά σημεία

Στη συνέχεια παρατίθενται μερικά χαρακτηριστικά του έργου που αξίζει να αναφερθούν.

- ✓ Έχει ληφθεί πρόνοια ώστε να μην χάνονται τα δεδομένα που εισάγει ή βλέπει ο χρήστης σε μια οθόνη αν αλλάξει η κατάσταση του κινητού (π.χ. αν αλλάξει τον προσανατολισμό της συσκευής από κάθετο σε οριζόντιο). Κατά την ανάπτυξη της εφαρμογής είχαμε να αντιμετωπίσουμε την πρόκληση που εμφανίζεται κάθε φορά σε αυτό το είδος ανάπτυξης λογισμικού: η προσαρμογή της διάταξης της οθόνης. Δοκιμάσαμε την εφαρμογή σε αρκετές συσκευές (τόσο φυσικές όσο και εικονικές μέσα από την δυνατότητα που προσφέρει το περιβάλλον του Android Studio) και διαπιστώσαμε ότι το πιο δύσκολο κομμάτι της σχεδίασης της εμφάνισής της ήταν να κάνουμε την διάταξη προσαρμόσιμη σε οριζόντιο προσανατολισμό. Η αλήθεια είναι ότι δεν μπορέσαμε να το πετύχουμε για όλες τις συσκευές που δοκιμάσαμε οπότε δώσαμε έμφαση στο να εμφανίζεται ορθά η διάταξη τουλάχιστον σε κάθετο προσανατολισμό (portrait) σε όσο το δυνατόν περισσότερες συσκευές μπορούσαμε. Ο εξομοιωτής που χρησιμοποιήθηκε για την αποτύπωση της λειτουργικότητας της εφαρμογής σε screenshots είναι ο “Galaxy Nexus API 29” (με τον χαρακτηρισμό normal size από τον AVD manager του Android Studio). Στον εξομοιωτή αυτόν εμφανίζονται ορθά οι διατάξεις τόσο σε κάθετο όσο και σε οριζόντιο προσανατολισμό (portrait και landscape

mode). Μια από τις επεκτάσεις της εφαρμογής είναι η σχεδίαση διατάξεων για διαφορετικά μεγέθη οθονών σύμφωνα με τον τρόπο που διδαχθήκαμε στο μάθημα.

- ✓ Χρησιμοποιείται η διαχείριση των strings με το αρχείο strings.xml του φακέλου values για την εμφάνιση κειμένων.
- ✓ Χρησιμοποιήθηκε ο διαχειριστής χρωμάτων της [Google](#) για την δημιουργία των χρωμάτων που χρησιμοποιεί η εφαρμογή. Το πράσινο με τις δύο διαφορετικές αποχρώσεις για τα primary και complementary χρώματα ήταν αναμφίβολα η πρώτη σκέψη λόγω της θεματικής του application.
- ✓ Η ελάχιστη έκδοση του Android που απαιτείται για την εγκατάσταση της εφαρμογής είναι η Android 5.0 (API level 21).

Ιδέες για μελλοντική εξέλιξη

Σε αυτή την ενότητα παρουσιάζονται κάποιες ιδέες που θα μπορούσαν να υλοποιηθούν σε ενδεχόμενες επεκτάσεις της εφαρμογής. Σίγουρα οι λειτουργίες της σε αυτή την πρώτη έκδοση για την εργασία του μαθήματος είναι λίγες και απλές. Με την προσδοκία όμως να βελτιωθεί η σχεδίαση και να προστεθούν επιπλέον λειτουργίες αποφασίσαμε με το πέρας της εξέτασης της εργασίας να δημοσιεύσουμε το project στο github.

- Δημιουργία ειδοποιήσεων της εφαρμογής για λίπανση. Ο χρήστης θα επιλέγει κάθε πότε θέλει να λιπάνει τα φυτά του και όταν ολοκληρώνεται ο κύκλος ημερών που έχει προσδιορίσει θα λαβαίνει ειδοποιήσει για πραγματοποίηση λίπανσης.
- Νέα σχεδίαση της εφαρμογής ώστε να εφαρμόζει τις οδηγίες σχεδίασης και να αξιοποιεί τα συστατικά του Material Design.
- Προσθήκη διατάξεων για διαφορετικά μεγέθη συσκευών.
- Δημιουργία κατηγοριών φυτών έτσι ώστε να προστεθούν στην εφαρμογή ξεχωριστές πληροφορίες και καλλιεργητικές οδηγίες για αυτά. Για παράδειγμα θα μπορεί η εφαρμογή να προτείνει ρυθμό λίπανσης ανάλογα με το είδος του φυτού.
- Χρήση της κάμερας για δημιουργία φωτογραφιών των φυτών. Ο χρήστης θα μπορεί να φωτογραφίζει τα φυτά του καθόλη τη διάρκεια της ζωής τους, έτσι ώστε να κρατάει και να έχει στη διάθεσή του σε φωτογραφικό υλικό την πορεία του φυτού.
- Λήψη πληροφοριών για τον καιρό ανάλογα με την τοποθεσία του χρήστη, για να απευθύνεται σε αγρότες. Θα μπορούσαν να προστεθούν πολλές λειτουργίες για να μετατραπεί η εφαρμογή σε χρήσιμο εργαλείο στα χέρια των γεωργών. Όπως για παράδειγμα η σύνδεση με τον γεωπόνο του γεωργού μέσω της εφαρμογής για αποστολή φωτογραφιών και δεδομένων των φυτών ώστε να λαβαίνει από αυτόν κατάλληλη συμβουλευτική υποστήριξη.