



**ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

**FarmIt: Ανάπτυξη ανοιχτής πλατφόρμας αγοράς γεωργικών  
προϊόντων μέσω κινητού τηλεφώνου**

Πτυχιακή εργασία

**Αδαμάντιος Ζάρας**



Αθήνα, Φεβρουάριος 2017



# **ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ**

**ΣΧΟΛΗ ΨΗΦΙΑΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ**

## **Τριμελής Εξεταστική Επιτροπή**

**Βαρλάμης Ηρακλής  
Επίκουρος Καθηγητής,  
Τμήμα Πληροφορικής και Τηλεματικής,  
Χαροκόπειο Πανεπιστήμιο**

**Δημητρακόπουλος Γεώργιος  
Επίκουρος Καθηγητής,  
Τμήμα Πληροφορικής και Τηλεματικής,  
Χαροκόπειο Πανεπιστήμιο**

**Αναγνωστόπουλος Δημοσθένης  
Καθηγητής,  
Τμήμα Πληροφορικής και Τηλεματικής,  
Χαροκόπειο Πανεπιστήμιο**

# Πνευματικά δικαιώματα

Copyright © Αδαμάντιος Ζάρας, 2017

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου Αθηνών δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ο Αδαμάντιος Ζάρας

δηλώνω υπεύθυνα ότι:

- 1) Είμαι ο κάτοχος των πνευματικών δικαιωμάτων της πρωτότυπης αυτής εργασίας και από όσο γνωρίζω η εργασία μου δε συκοφαντεί πρόσωπα, ούτε προσβάλει τα πνευματικά δικαιώματα τρίτων.
- 2) Αποδέχομαι ότι η ΒΚΠ μπορεί, χωρίς να αλλάξει το περιεχόμενο της εργασίας μου, να τη διαθέσει σε ηλεκτρονική μορφή μέσα από τη ψηφιακή Βιβλιοθήκη της, να την αντιγράψει σε οποιοδήποτε μέσο ή/και σε οποιοδήποτε μορφότυπο καθώς και να κρατά περισσότερα από ένα αντίγραφα για λόγους συντήρησης και ασφάλειας.
- 3) Ο κώδικας που αναπτύχθηκε για την πτυχιακή μου εργασία και αποτελεί τμήμα του FarmIt, είναι ανοιχτός και διαθέσιμος για κάθε χρήση.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον κ. *Ηρακλή Βαρλάμη* για τη βοήθειά του και τις εύστοχες υποδείξεις του κατά τη διάρκεια ανάπτυξης της πτυχιακής μου εργασίας.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Περίληψη .....	7
Abstract.....	8
Κατάλογος εικόνων.....	9
Κατάλογος διαγραμμάτων.....	10
Συντομογραφίες .....	12
Εισαγωγή.....	13
Κεφ.1: Σχετική δουλειά.....	14
Κεφ.2: Θεωρητικό υπόβαθρο .....	15
2.1. Backend – Frontend .....	15
2.2. Υβριδικές εφαρμογές.....	15
2.3. Βάση δεδομένων .....	16
2.4. RESTful Web Service .....	17
2.5. Framework .....	17
2.6. Model-View-Controller (MVC) .....	18
2.7. Server .....	19
2.8. IDE .....	19
2.9. Version Control System (VCS) .....	20
2.10. Πρωτόκολλα επικοινωνίας – HTTPS .....	20
Κεφ.3: Απαιτήσεις εφαρμογής .....	23
3.1. Λειτουργικές απαιτήσεις .....	23
3.1.1. Εγγραφή χρηστών .....	23
3.1.2. Είσοδος και αποσύνδεση χρηστών .....	23
3.1.3. Αναζήτηση παραγωγών .....	24
3.1.4. Ενημερώσεις .....	24
3.1.5. Αξιολογήσεις.....	24
3.1.6. Μηνύματα.....	24
3.1.7. Ανάρτηση αγγελιών ζήτησης .....	25
3.1.8. Προφίλ .....	25
3.1.9. Στατιστικά .....	25
3.2. Μη λειτουργικές απαιτήσεις .....	26
3.2.1. Απόδοση και απόκριση.....	26
3.2.2. Χρηστικότητα .....	27
3.2.3. Ασφάλεια .....	27

Κεφ.4: Εργαλεία που χρησιμοποιήθηκαν .....	28
4.1. Servers.....	28
4.2. Frameworks .....	29
4.3. Εργαλεία για τον έλεγχο της λειτουργίας .....	30
4.4. Integrated Development Environments (IDEs).....	30
4.5. Version Control System.....	30
4.6. Κοινοποίηση εφαρμογής .....	31
Κεφ.5: Υλοποίηση εφαρμογής.....	32
5.1. Βάση Δεδομένων .....	32
5.2. Backend.....	34
5.2.1. Controllers.....	34
5.2.2. Models .....	36
5.3. Frontend.....	37
5.3.1. Templates.....	38
5.3.2. Controllers.....	40
5.3.3. Services .....	43
Κεφ.6: Ιδιαίτερα σημεία .....	45
6.1. Transactions .....	45
6.2. Infinite Scroller .....	45
6.3. Workaround .....	46
Κεφ.7: Παραδείγματα Χρήσης.....	47
7.1. Αναζήτηση.....	47
7.2. Αγγελία.....	48
7.3. Άμεση Αγορά.....	49
7.4. Αλλαγή των στοιχείων προφίλ .....	50
7.5. Προσθήκη νέας διεύθυνσης .....	51
7.6. Στατιστικά χρήσης.....	52
Κεφ.8: Η ολοκλήρωση του FarmIt .....	53
Κεφ.9: Μελλοντικές επεκτάσεις .....	55
Κεφ.10: Επίλογος .....	56
Βιβλιογραφία .....	57

## Περίληψη

Η χρήση των κινητών τηλεφώνων ολοένα και αυξάνεται. Οι άνθρωποι χρησιμοποιούν περισσότερες ώρες το κινητό τους τηλέφωνο, παρά τον υπολογιστή τους ή άλλα μέσα δικτύωσης. Το λειτουργικό του σύστημα σε συνδυασμό με όσα αυτό μπορεί να προσφέρει και η φορητότητά του, το καθιστούν ως την πλέον χρήσιμη ηλεκτρονική συσκευή.

Συνεπώς οι εφαρμογές των κινητών τηλεφώνων είναι απαραίτητες σε τεχνικό, αλλά και σε διαφημιστικό ακόμη επίπεδο. Υπάρχουν πολλές προσεγγίσεις για την ανάπτυξή τους, μία εκ των οποίων είναι ο προγραμματισμός υβριδικών εφαρμογών, με τη χρήση τεχνολογιών διαδικτύου, όπως HTML5, Javascript και CSS. Τα χαρακτηριστικά αυτής της προσέγγισης, είναι ο γρηγορότερος προγραμματισμός και η χρήση της εφαρμογής σε δύο ή περισσότερα διαφορετικά λειτουργικά συστήματα, με αντάλλαγμα την ταχύτητα εκτέλεσής της. Ωστόσο με τη ραγδαία εξέλιξη της τεχνολογίας, η αρνητική επίδραση στην ταχύτητα εκτέλεσης των υβριδικών εφαρμογών έχει ελαχιστοποιηθεί σημαντικά στα κινητά νέας γενιάς.

Μία ακόμη τάση στις μέρες μας, είναι η ανάπτυξη Ελεύθερου Λογισμικού/Λογισμικού Ανοιχτού Κώδικα (ΕΛ/ΛΑΚ), για λόγους οι οποίοι μπορούν εύκολα να γίνουν κατανοητοί, αν αναλογιστεί κανείς τα οφέλη που απορρέουν, όπως η διαλειτουργικότητα και η ύπαρξη πληθώρας επιλογών και δοκιμασμένων λύσεων.

Αξιοποιώντας τις προσφορές της σύγχρονης τεχνολογίας, με την ανάπτυξη ΕΛ/ΛΑΚ, μπορούμε να κάνουμε ευκολότερη και οικονομικότερη τη διαδικασία ορισμένων πραγμάτων, όπως για παράδειγμα τη διαφήμιση ή την προώθηση προϊόντων καθώς και την αγοραπωλησία τους.

Για τους παραπάνω λόγους, αποφάσισα να αναπτύξω μία υβριδική εφαρμογή ανοιχτού κώδικα για κινητά τηλέφωνα και όχι παραδείγματος χάρη μια ιστοσελίδα, μέσω της οποίας θα μπορούν όλοι να προμηθεύονται προϊόντα γεωργικής παραγωγής χωρίς μεσάζοντες, σε χαμηλότερο κόστος. Η εφαρμογή αυτή ονομάζεται FarmIt.

**Λέξεις κλειδιά:** FarmIt, υβριδική εφαρμογή, ΕΛ/ΛΑΚ.

## Abstract

The usage of smartphones is increasingly growing. People put more hours into using their smartphones than their personal computers or other networking media. Their operating systems, in conjunction with what they can offer and their portability make them the most useful electronic devices.

Therefore, smartphone applications are necessary in a technical, and even in a promotional level. There are many approaches to their development, one of which is the development of hybrid applications using web technologies such as HTML5, JavaScript and CSS. The characteristics of this approach, are shorter development time and the ability to run the application in two or more different operating systems, in exchange for execution speed. However, owing to the rapid evolution of technology, the negative effect on the execution speed of hybrid applications has significantly minimized in new smartphones.

Another current trend is the development of Free Software/Open Source Software (FS/OSS), which can be easily understood if one considers the rewards, such as interoperability and the presence of a multitude of options and proven solutions.

By utilizing what modern technology has to offer, with FS/OSS development we can make certain processes, such as advertising or promotion of products, as well as trading them, easier and cheaper.

For these reasons, I decided to develop a hybrid open source mobile application, over, for instance, a website, through which anyone can procure agricultural produce without intermediaries, at a lower cost. This application is called FarmIt.

**Keywords:** FarmIt, hybrid app, FS/OSS.



## ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Do not reinvent the wheel .....	28
Εικόνα 2: WampServer .....	28
Εικόνα 3: CodeIgniter.....	29
Εικόνα 4: Ionic.....	30
Εικόνα 5: Με εφαρμογή του workaround .....	46
Εικόνα 6: Χωρίς εφαρμογή του workaround.....	46
Εικόνα 7: Παράδειγμα αναζήτησης.....	47
Εικόνα 8: Παράδειγμα αγγελίας.....	48
Εικόνα 9: Παράδειγμα άμεσης αγοράς .....	49
Εικόνα 10: Παράδειγμα αλλαγής στοιχείων του προφίλ .....	50
Εικόνα 11: Παράδειγμα προσθήκης νέας διεύθυνσης .....	51
Εικόνα 12: Παράδειγμα προβολής στατιστικών χρήσης.....	52
Εικόνα 13: Η τελική μορφή του FarmIt.....	53

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Σχήμα 1: Σύγκριση Native, Web και Hybrid εφαρμογών .....	15
Σχήμα 2: Αναπαράσταση ενός RESTful Web Service.....	17
Σχήμα 3: Μέθοδος σχεδίασης MVC .....	18
Σχήμα 4: Αναπαράσταση της λειτουργίας των VCS .....	20
Σχήμα 5: Η διαδικασία του HTTPS.....	21
Σχήμα 6: Βάση Δεδομένων του FarmIt.....	32

## ΤΜΗΜΑΤΑ ΚΩΔΙΚΑ

Snippet 1: Infinite scroller.....	45
Snippet 2: los Workaround .....	46

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΕΛ/ΛΑΚ	Ελεύθερο Λογισμικό/Λογισμικό Ανοικτού Κώδικα
FS/OSS	Free Software/Open Source Software
DBMS	Database Management System
REST	Representational State Transfer
MVC	Model View Controller
HMVC	Hierarchical Model View Controller
API	Application Programming Interface
VCS	Version Control System
TLS	Transport Layer Security
SSL	Secure Sockets Layer
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDE	Integrated Development Environment

## ΕΙΣΑΓΩΓΗ

Το Fatmlt, είναι ένα έργο ΕΛ/ΛΑΚ που ξεκίνησαν οι φοιτητές του τμήματος Πληροφορικής και Τηλεματικής *Χρήστος Σαρδιανός* και *Γιάννης Κατάκης*. Στο κομμάτι που αφορά την εφαρμογή για τον παραγωγό έχει συνεισφέρει ο φοιτητής του τμήματος *Στέφανος Φαφαλιός*.

Στην πτυχιακή μου εργασία, επικεντρώθηκα στο κομμάτι των λειτουργιών που αφορούν τους πελάτες του Farmlt. Συγκεκριμένα, ανασχεδίασα τη βάση δεδομένων και υλοποίησα ένα Restful Web Service (backend) και μια εφαρμογή για κινητά (frontend), τα οποία συμβάλουν στην εύκολη αγορά και αναζήτηση προϊόντων ανάμεσα στους αγρότες του Farmlt.

Αρχικά, αναφέρονται παρόμοιες εφαρμογές με τα χαρακτηριστικά τους και τις διαφορές που κάνουν το Farmlt ξεχωριστό. Στη συνέχεια, παρέχονται οι βασικές γνώσεις που χρειάζεται να διαθέτει κάποιος, προκειμένου να αναπτύξει την εφαρμογή. Έστερα, αναφέρονται οι απαιτήσεις που τέθηκαν προτού μελετηθούν τα εργαλεία ανάπτυξης και οι τεχνολογίες που θα εφαρμόζονταν και στη συνέχεια αναφέρονται αυτά που τελικά επιλέχθηκαν. Επιπλέον, αναλύονται λεπτομερώς τα βήματα για την ανάπτυξη του Farmlt και τονίζονται ορισμένα ιδιαίτερα χαρακτηριστικά ή προβλήματα, καθώς και ο τρόπος με τον οποίο επιλύθηκαν. Τέλος, παρέχονται εικόνες με παραδείγματα χρήσης, από τις βασικότερες λειτουργίες της εφαρμογής.

## Κεφ.1: Σχετική δουλειά

Στο παρελθόν, έχουν γίνει και άλλες προσπάθειες για την ανάπτυξη ιστοσελίδων ή εφαρμογών μέσω των οποίων θα μπορούν να προωθούνται προϊόντα μεταξύ απλών χρηστών. Παρακάτω αναλύονται ορισμένα παραδείγματα:

- ebay.com: Είναι μια ιστοσελίδα, η οποία δίνει τη δυνατότητα σε ιδιώτες αλλά και επιχειρήσεις να πουλήσουν αντικείμενα καινούρια ή μεταχειρισμένα, από ρούχα, κοσμήματα, ηλεκτρικά και ηλεκτρονικά είδη, μέχρι σπίτια και αυτοκίνητα. Οι πωλήσεις μπορούν να είναι είτε με τη μορφή δημοπρασίας, όπου την αγορά ολοκληρώνει εκείνος που έκανε τη μεγαλύτερη προσφορά στη λήξη του χρόνου δημοπράτησης, ή με τη μορφή άμεσης αγοράς.
- emarket.gr: Είναι μία ελληνική ιστοσελίδα, με παρόμοια λειτουργία με το e-bay.
- e-click.gr: Είναι μία ελληνική ιστοσελίδα, με παρόμοια λειτουργία με το e-bay, όπου οι πωλήσεις δεν μπορούν να είναι με τη μορφή δημοπρασιών.
- car.gr: Είναι μια ελληνική ιστοσελίδα, η οποία επικεντρώνεται σε αγορές και πωλήσεις με τη μορφή αγγελιών, κυρίως μεταχειρισμένων αυτοκινήτων, μηχανών και γενικά οχημάτων, καθώς και ανταλλακτικών για αυτά. Επίσης υπάρχει η δυνατότητα δημιουργίας αγγελιών ζήτησης από τους χρήστες.

Παρόλαυτα, το Farmlt έχει το χαρακτηριστικό ότι αναφέρεται σε παραγωγούς – αγρότες, πράγμα το οποίο είναι ιδιαίτερα σημαντικό, καθώς ο κλάδος της γεωργίας απασχολεί ολοένα και περισσότερα άτομα. Συγκεκριμένα για την Ελλάδα, ο γεωργικός της πλούτος είναι μεγάλος, επομένως η προώθηση των προϊόντων γεωργικής παραγωγής είναι ζωτικής σημασίας. Επιπλέον, το Farmlt συμβάλλει στη φθηνότερη αγορά τους και είναι εφαρμογή ανοιχτού κώδικα, έτσι ώστε να μπορεί να συνεισφέρει όποιος επιθυμεί σε αυτή την προσπάθεια.

Αρχικά, προϋπήρχε το κομμάτι που αφορά τους παραγωγούς και ανέλαβα να επεκτείνω το έργο, δημιουργώντας τη λειτουργικότητα που αφορά τους πελάτες. Το πρώτο, περιλαμβάνει τη δυνατότητα από τους παραγωγούς να αναρτούν τα προϊόντα τους ώστε να είναι διαθέσιμα στην εφαρμογή, καθώς και τη δυνατότητα αντίστροφων δημοπρασιών μεταξύ τους για τη δημιουργία προσφορών σε αγγελίες των πελατών. Το δεύτερο, αφορά τη δυνατότητα προβολής των παραγωγών και των προϊόντων, τη δημιουργία αγγελιών που θα μπορούν να προβληθούν από τους παραγωγούς και την αποδοχή ή την απόρριψη των προσφορών τους.

## Κεφ.2: Θεωρητικό υπόβαθρο

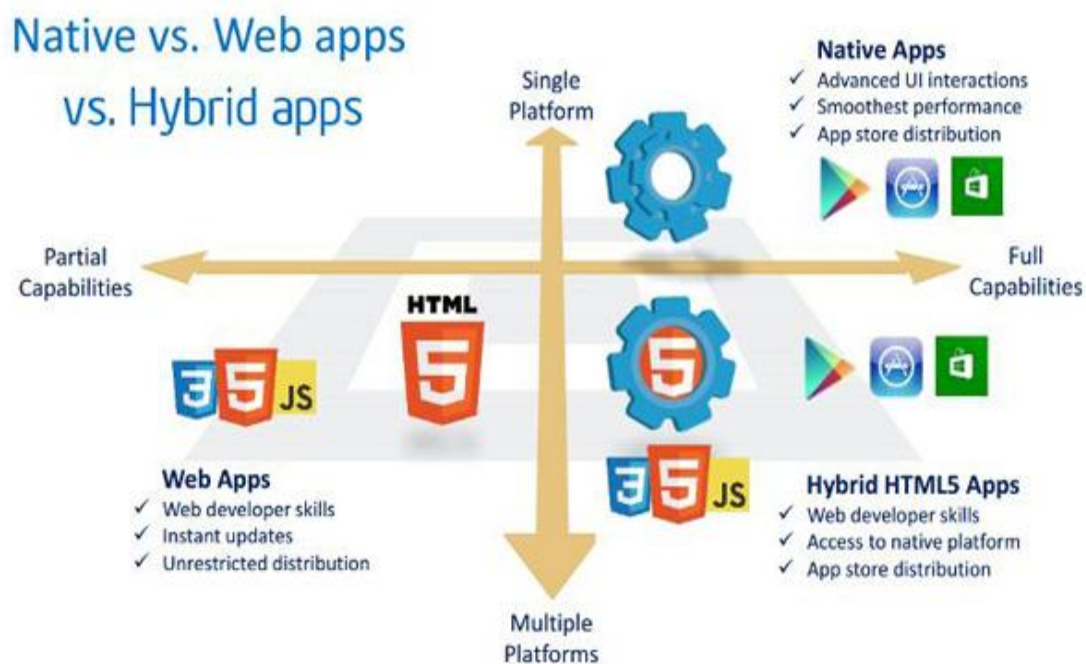
Παρακάτω, αναλύονται όλες οι απαραίτητες γνώσεις για τη δημιουργία μιας εφαρμογής όπως το FarmIt.

### 2.1. Backend – Frontend

- Backend, ονομάζεται το μέρος μιας εφαρμογής, το οποίο αλληλεπιδρά με τα δεδομένα.
- Frontend, ονομάζεται το μέρος μιας εφαρμογής, το οποίο αλληλεπιδρά με το χρήστη.

Το Backend και το Frontend, μπορεί να είναι χωρισμένα σε ένα ή περισσότερα συστήματα.

### 2.2. Υβριδικές εφαρμογές



Σχήμα 1: Σύγκριση Native, Web και Hybrid εφαρμογών

Υβριδικές ονομάζονται οι εφαρμογές που μπορούν να τρέξουν σε παραπάνω από ένα λειτουργικά συστήματα, πχ android, ios, windows, κτλ. Αυτό επιτυγχάνεται με τη χρήση ορισμένων πλατφόρμων, όπως το phonegap.

Με αυτόν τον τρόπο, μειώνεται η πολυπλοκότητα της ανάπτυξης των εφαρμογών, καθώς και ο χρόνος που απαιτείται, αφού με την ανάπτυξη μίας εφαρμογής, μπορούμε να έχουμε εκτελέσιμα αρχεία για παραπάνω από μία πλατφόρμες. Επιπλέον, η τεχνογνωσία που

απαιτείται είναι μικρότερη, αφού δεν απαιτείται γνώση προγραμματισμού native εφαρμογών, παρά μόνο διαδικτυακών.

Ωστόσο, ένα αρνητικό χαρακτηριστικό των υβριδικών εφαρμογών, είναι η ταχύτητα εκτέλεσής τους, καθώς μία native εφαρμογή που εξυπηρετεί τους ίδιους σκοπούς, πάντα θα τρέχει γρηγορότερα. Αυτό το πρόβλημα περιορίζεται με την ραγδαία ανάπτυξη της τεχνολογίας των κινητών τηλεφώνων, καθώς με τα χρόνια οι δυνατότητές τους αυξάνονται.

## 2.3. Βάση δεδομένων

Η βάση δεδομένων, είναι μία συλλογή δεδομένων η οποία έχει συγκεκριμένη δομή και είναι οργανωμένη με τέτοιον τρόπο, ώστε ένας υπολογιστής να μπορεί εύκολα να βρει τις πληροφορίες που αναζητάει σε αυτή.

Γενικότερα, μία βάση δεδομένων, μπορεί να περιγραφεί ως μια συλλογή από εγγραφές, οι οποίες περιλαμβάνουν πεδία που τις χαρακτηρίζουν. Για παράδειγμα, μία εγγραφή θα μπορούσε να είναι ένα αυτοκίνητο και τα πεδία του να είναι η μάρκα, το χρώμα, το βάρος, η υποδύναμη κτλ.

Σε βάθος χρόνου, έχουν αναπτυχθεί διάφορα μοντέλα βάσεων δεδομένων. Κάθε μοντέλο περιγράφει τη δομή τους, καθώς και τις λειτουργίες που μπορούν να πραγματοποιηθούν σε αυτές. Τυπικά κάθε βάση έχει ένα schema, το οποίο είναι μια περιγραφή του μοντέλου, συμπεριλαμβανομένων των τύπων των πεδίων του, καθώς και τις σχέσεις μεταξύ τους.

Για την αποτελεσματική διαχείριση των βάσεων δεδομένων, μπορούμε να χρησιμοποιήσουμε ένα Database Management System (DBMS), ένα λειτουργικό ανεπτυγμένο με τέτοιον τρόπο ώστε να εξυπηρετεί αυτόν το σκοπό.

Το FarmIt χρησιμοποιεί το μοντέλο της σχεσιακής βάσης δεδομένων, το οποίο έχει τα δεδομένα οργανωμένα με τέτοιον τρόπο, ώστε να δίνει την εντύπωση ότι είναι αποθηκευμένα σε μια σειρά από αλληλοσχετιζόμενους πίνακες.



## 2.4. RESTful Web Service



Σχήμα 2: Αναπαράσταση ενός RESTful Web Service

Με τη χρήση των RESTful Web Services, ελαχιστοποιούμε τους φραγμούς κατά την υλοποίηση και ανεξαρτητοποιούμε το server από τον client, καθώς αν επιθυμούμε μπορούμε να ενημερώσουμε το server και να αφήσουμε την εφαρμογή του client ως έχει. Πολύ συχνά στις εφαρμογές κινητών τηλεφώνων είναι απαραίτητη η χρήση τους.

Τα RESTful Web Services, απαντούν στους clients με αναπαραστάσεις κειμένου και όχι γραφικό τρόπο όπως για παράδειγμα μια ιστοσελίδα. Υπάρχουν διαφόρων ειδών αναπαραστάσεις κειμένου, όπως Raw text, XML, HTML, JSON κτλ. Οι απαντήσεις μπορεί να επιβεβαιώνουν μία αλλαγή στη βάση, να επιστρέφουν δεδομένα ή ακόμη και μηνύματα προειδοποίησης για μη εξουσιοδοτημένη πρόσβαση, αναλόγως πάντα με τα αιτήματα των clients.

Η επιτυχία στην ανάπτυξη ενός RESTful Web Service έγκειται στην ασφάλειά του, δηλαδή στη σωστή κωδικοποίηση των δεδομένων, στη χρήση API Keys και την απαγόρευση πρόσβασης σε μη εξουσιοδοτημένους χρήστες.

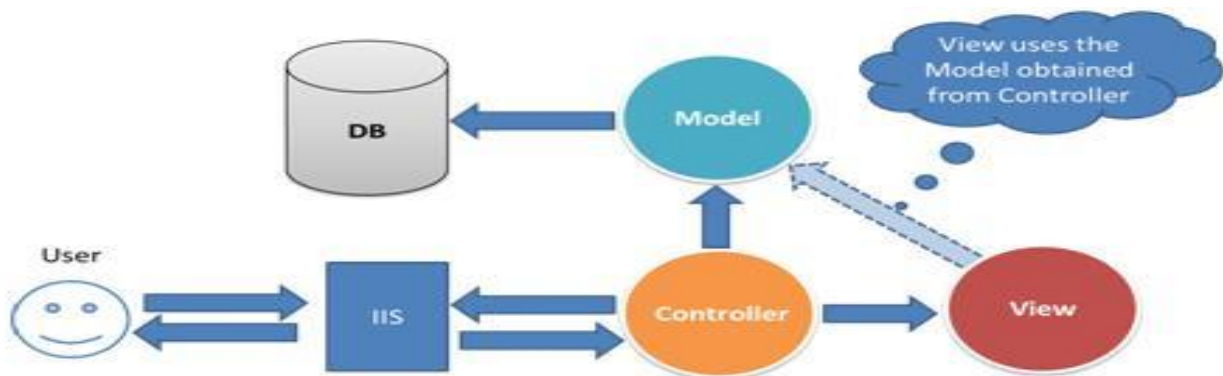
## 2.5. Framework

Τα frameworks αποτελούνται από μία αρχιτεκτονική η οποία έχει σχεδιαστεί με βασικό σκοπό τη διευκόλυνση της ανάπτυξης μιας εφαρμογής με τη χρήση τους. Υπάρχουν πολλά διαφορετικά frameworks, τα οποία εξυπηρετούν ποικίλους σκοπούς και έχουν σχεδιαστεί για να προσφέρουν, όπως ο δημιουργός τους πιστεύει καλύτερα. Ορισμένα από τα πράγματα που μπορεί να παρέχουν, αναλύονται παρακάτω:

- Μπορούν να συνδεθούν σε βάσεις δεδομένων, πράγμα το οποίο σημαίνει πως απλά παρέχοντας τα στοιχεία της βάσης μας, όπως την IP διεύθυνση, τον τύπο της βάσης κτλ., το πρόγραμμά μας αποκτά πρόσβαση σε αυτή, χωρίς να χρειαστεί καθόλου κώδικας.

- Αναλαμβάνουν τη διαχείριση και την πιστοποίηση των χρηστών, με την υψηλότερη δυνατή ασφάλεια. Μπορούν επίσης να διαχειριστούν τα δεδομένα των session ενός χρήστη, καθώς και τα δικαιώματα πρόσβασής του.
- Προσφέρουν φορητότητα, καθώς η εφαρμογή μπορεί να εγκατασταθεί σε πολλούς server και να χρησιμοποιούν, ακόμη, διαφορετικές βάσεις. Αυτό είναι αρκετά θετικό σε εφαρμογές ανοιχτού κώδικα όπως το Farmlt, εφόσον μπορούν να χρησιμοποιούν την εφαρμογή, για μελέτη και βελτίωση, πολλοί διαφορετικοί χρήστες.
- Ένα ακόμη πλεονέκτημα, είναι η ταχύτερη και ευκολότερη ανάπτυξη μιας εφαρμογής, καθώς παρέχουν πολύ μεγάλες βιβλιοθήκες οι οποίες μπορούν να χρησιμοποιηθούν, ελαχιστοποιώντας τις γραμμές του κώδικα.
- Μεγάλο θετικό, είναι η ασφάλεια που παρέχουν. Αποτρέπουν μορφές επιθέσεων, όπως το SQL Injection, με τη χρήση των έτοιμων κλάσεων και συναρτήσεων που διαθέτουν.
- Υποστηρίζονται από κοινότητες στις οποίες μπορεί κανείς να απευθυνθεί για λύσεις σε τυχόν προβλήματα που αντιμετωπίζει. Επιπλέον είναι επεκτάσιμα, βελτιώνονται συνεχώς και η αναβάθμιση σε νέα έκδοση είναι πολύ απλή.
- Τέλος, με τη χρήση τους, ακολουθούμε μοντέλα προγραμματισμού, όπως το MVC, πράγμα το οποίο καθιστά ποιο ποιοτικό το τελικό πρόγραμμα.

## 2.6. Model-View-Controller (MVC)



Σχήμα 3: Μέθοδος σχεδίασης MVC

Ως Model-View-Controller, ορίζουμε μία μέθοδο σχεδίασης λειτουργικού, με το οποίο δημιουργούμε μία γραφική διεπαφή χρήστη. Παλαιότερα χρησιμοποιούνταν για τη δημιουργία προγραμμάτων σε υπολογιστές, ενώ στις μέρες μας έχει γίνει δημοφιλής η χρήση του για την ανάπτυξη διαδικτυακών εφαρμογών και εφαρμογών κινητών τηλεφώνων.

Σε αυτή τη μέθοδο, τα πάντα ξεκινούν από τον Controller, ο οποίος δέχεται κάτι ως είσοδο και το μετατρέπει σε εντολές για το Model. Το Model κυριαρχεί σε μια εφαρμογή, καθώς διαχειρίζεται τα δεδομένα και τη λογική. Αντιθέτως το View, διαχειρίζεται απλώς αναπαραστάσεις των πληροφοριών που παρέχει σε αυτό το Model, όπως είναι για παράδειγμα η εμφάνιση μιας λίστας με ονόματα. Μία εφαρμογή, μπορεί να αποτελείται από

πολλά Models, Views και Controllers, όπως επίσης τα Views και σπάνια τα Models, μπορούν να παραλειφθούν.

Στην περίπτωση του Farmlt, κάθε πελάτης έχει μια εφαρμογή στο κινητό του. Η εφαρμογή αυτή, επικοινωνεί όποτε είναι απαραίτητο με το RESTful Web Server, προκειμένου να στέλνει αιτήματα στους Controllers, οι οποίοι αναθέτουν λειτουργίες στα Models που έχουν τη δυνατότητα να επικοινωνούν με τη βάση δεδομένων. Στη συνέχεια τα Models, επιστρέφουν το αποτέλεσμα των λειτουργιών που έχουν πραγματοποιήσει στους Controllers, οι οποίοι με τη σειρά τους απαντούν στο αίτημα της εφαρμογής. Τέλος, η εφαρμογή αναπαριστά γραφικά αυτή την πληροφορία στον πελάτη, αντικαθιστώντας τα Views.

Υπάρχουν αρκετά frameworks τα οποία υιοθετούν την MVC λογική. Ένα από αυτά είναι το CodeIgniter, το οποίο χρησιμοποιήθηκε κατά την ανάπτυξη του Farmlt.

## 2.7. Server

Ένας server, αλλιώς γνωστός ως εξυπηρετητής ή διακομιστής, είναι ένα λογισμικό το οποίο προσφέρει υπηρεσίες σε υπολογιστές, οι οποίοι ονομάζονται clients (πελάτες). Ανάλογα με τους σκοπούς που εξυπηρετεί, εκτελεί και τις αντίστοιχες λειτουργίες. Για παράδειγμα ένας mail server έχει τη δυνατότητα αποστολής και λήψης μηνυμάτων μεταξύ των clients. Ένας server μπορεί να εξυπηρετεί πολλούς clients ταυτόχρονα, όπως επίσης ένας client, μπορεί να είναι ταυτόχρονα συνδεδεμένος σε πολλούς servers. Ο server και ο client, μπορεί να βρίσκονται στην ίδια συσκευή. Συνήθως, οι servers είναι εγκατεστημένοι σε υπολογιστές αυξημένων δυνατοτήτων, επομένως, είναι ιδιαίτερα χρήσιμοι στην περίπτωση που θέλουμε να πραγματοποιήσουμε κάποια απαιτητική λειτουργία, ώστε να γίνεται με μεγάλη ταχύτητα και να επιστρέφεται το αποτέλεσμα στους clients. Τα επιθυμητά χαρακτηριστικά ενός server, είναι τα εξής:

- Αυξημένη χωρητικότητα και ταχύτητα των δίσκων
- Αυξημένη χωρητικότητα και ταχύτητα της μνήμης RAM
- Μεγάλη επεξεργαστική ισχύς (CPU)
- Πολύ μεγάλη χωρητικότητα δικτύου
- Αξιόπιστα υλικά

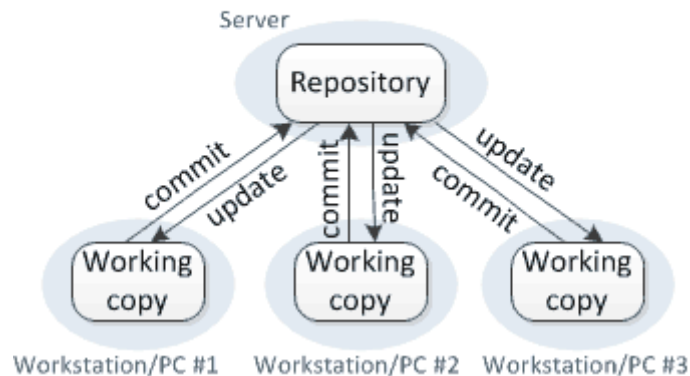
Το Farmlt, χρησιμοποιεί έναν database server για τη βάση δεδομένων και έναν web server για το backend.

## 2.8. IDE

Το IDE είναι μία εφαρμογή λογισμικού, η οποία διευκολύνει τη διαδικασία του προγραμματισμού. Έχει εύκολο γραφικό περιβάλλον και ένα πλήθος χρήσιμων εργαλείων, για αυτόματη συμπλήρωση και μορφοποίηση κώδικα, debugging, version control κτλ.

## 2.9. Version Control System (VCS)

### Centralized version control



Σχήμα 4: Αναπαράσταση της λειτουργίας των VCS

Version Control System, ονομάζεται ένα είδος λογισμικού, το οποίο διευκολύνει τη διαχείριση αρχείων και φακέλων, καθώς και των αλλαγών που έχουν γίνει σε αυτά σε βάθος χρόνου. Επομένως, επιτρέπει την ανάκτηση παλαιότερων εκδόσεων των δεδομένων και την εξέταση του ιστορικού των αλλαγών τους. Μπορεί να λειτουργεί μέσω διαδικτύου, πράγμα το οποίο επιτρέπει τη χρήση του από διαφορετικούς υπολογιστές. Η ικανότητα της διαχείρισης των ίδιων δεδομένων και της εφαρμογής αλλαγών σε αυτά από διαφορετικά άτομα, διευκολύνει τη συνεργασία τους. Έτσι ο προγραμματισμός γίνεται γρηγορότερα, χωρίς όμως να ανταλλάσσεται η ποιότητα του κώδικα, εφόσον διατηρείται ιστορικό των αλλαγών και ανά πάσα στιγμή, σε περίπτωση λάθους, μπορεί να γυρίσει σε προηγούμενη έκδοση.

Στον πυρήνα του VCS, βρίσκεται ένα repository, μια μορφή αποθήκης, όπου είναι το κεντρικό σημείο αποθήκευσης των δεδομένων του συστήματος. Εκεί συνήθως βρίσκονται πληροφορίες με τη μορφή δέντρου. Σε αυτό, μπορούν να συνδέονται οι χρήστες και να πραγματοποιούν αλλαγές στα αρχεία.

## 2.10. Πρωτόκολλα επικοινωνίας – HTTPS

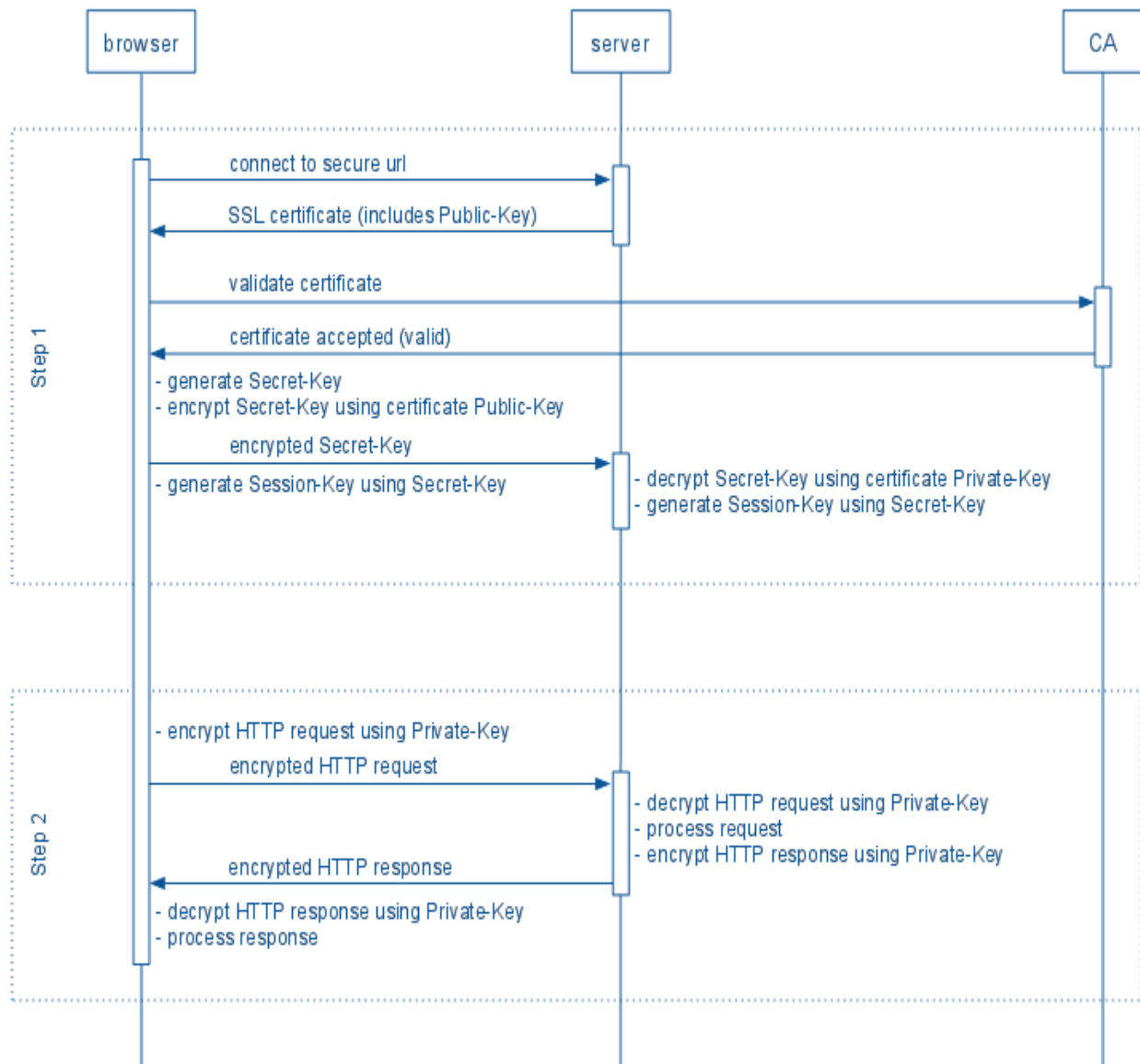
Ένα πρωτόκολλο επικοινωνίας, είναι ένα σύνολο από κανόνες, οι οποίοι είναι απαραίτητοι για την ανταλλαγή πληροφοριών μεταξύ υπολογιστών. Υλοποιούνται σε επίπεδο υλικού και λογισμικού. Υπάρχουν πολλά πρωτόκολλα επικοινωνίας, δίχως τα οποία δε θα μπορούσαν να υπάρξουν τα δίκτυα υπολογιστών. Ένα από αυτά, είναι το HTTPS.

# HTTPS Sequence Diagram

HTTPS uses SSL/TLS for secure communication between the browser and server.

Step 1: Public-Key encryption is first used to setup a secure connection

Step 2: Private-Key encryption is then used for all subsequent communication



\* SSL key exchange is complete after Step 1. Subsequent communication will use Private-Key encryption, the Session-Key will be used as the Private-Key.

\* Private-Key Encryption = Symmetric Encryption

\* Public-Key Encryption = Asymmetric Encryption

\* CA = Certificate Authority

\* HTTPS TCP Port 443

Σχήμα 5: Η διαδικασία του HTTPS

Το πρωτόκολλο HTTPS, αλλιώς γνωστό και ως HTTP Secure, HTTP over TLS και HTTP over SSL, επικοινωνεί πάνω από το Hypertext Transfer Protocol (HTTP), μέσω κρυπτογραφημένης σύνδεσης από το Transport Layer Security (TLS) ή το Secure Sockets Layer (SSL). Τα πλεονεκτήματά του συγκριτικά με άλλα πρωτόκολλα επικοινωνίας, συμπεριλαμβανομένου και του απλού HTTP, αναφέρονται παρακάτω:

- Αυθεντικότητα του περιεχόμενου των σελίδων που επισκέπτεται ο χρήστης, καθώς δε μπορεί να τροποποιηθεί από τρίτους με τεχνικές όπως Man In The Middle (MITM) επιθέσεις. Αυτό έχει ως αποτέλεσμα τη μεγαλύτερη ασφάλεια των χρηστών.
- Προσωπικότητα, όσον αφορά το ιστορικό των αναζητήσεων και των σελίδων που επισκέφθηκε ο χρήστης.
- Πιο φιλικές σελίδες προς τους χρήστες, καθώς ενέργειες όπως είναι η είσοδος σε έναν λογαριασμό, μπορούν να γίνουν στην ίδια σελίδα, χωρίς να χρειάζεται ανακατεύθυνση σε κάποια άλλη που χρησιμοποιεί HTTPS.
- Δυνατότητα χρήσης περισσότερων third party επεκτάσεων, οι οποίες συχνά απαιτούν τη χρήση HTTPS από τις σελίδες στις οποίες χρησιμοποιούνται.

## **Κεφ.3: Απαιτήσεις εφαρμογής**

Προκειμένου να σχεδιαστεί και στη συνέχεια να αναπτυχθεί σωστά το FarmIt, χρειάστηκε να οριστούν κάποιες απαιτήσεις/προσδοκίες για αυτό. Στη συνέχεια, με βάση αυτές, δημιουργήθηκε ένα use case diagram, προκειμένου να διευκολύνει και να επιταχύνει την ανάπτυξη της εφαρμογής.

### **3.1. Λειτουργικές απαιτήσεις**

Παρακάτω αναλύονται οι απαιτήσεις που ορίστηκαν αρχικά και αφορούν τις επιθυμητές λειτουργίες της εφαρμογής.

#### **3.1.1. Εγγραφή χρηστών**

Προκειμένου να χρησιμοποιήσει κάποιος το FarmIt ως πελάτης, θα πρέπει να δημιουργήσει ένα προφίλ, παρέχοντας βασικά στοιχεία, όπως το ονοματεπώνυμό του, το τηλέφωνο επικοινωνίας του κτλ.

Στη συνέχεια, θα πρέπει να εισάγει μία διεύθυνση προκειμένου να ολοκληρώσει τη διαδικασία της εγγραφής του και να χρησιμοποιήσει πλήρως την εφαρμογή, αποκτώντας μεταξύ άλλων τη δυνατότητα να αγοράζει προϊόντα και να δημιουργεί αγγελίες ζήτησης.

#### **3.1.2. Είσοδος και αποσύνδεση χρηστών**

Σε περίπτωση που κάποιος έχει ήδη λογαριασμό, θα πρέπει να δίνεται η δυνατότητα εισόδου στο σύστημα με το email και τον κωδικό που συμπλήρωσε κατά την εγγραφή του. Επιπλέον, αν ο χρήστης επιθυμεί, η εφαρμογή θα πρέπει να είναι σε θέση να τον συνδέει αυτόματα κάθε φορά που ανοίγει. Συνεπώς, θα πρέπει να παρέχεται και η δυνατότητα αποσύνδεσης από το λογαριασμό.

### **3.1.3. Αναζήτηση παραγωγών**

Οι παραγωγοί του FarmIt, θα πρέπει να εμφανίζονται στην αρχική σελίδα των πελατών, προκειμένου να μπορούν να περιηγηθούν μεταξύ αυτών, να δουν τα προϊόντα, τις αξιολογήσεις και τα στοιχεία επικοινωνίας τους, καθώς επίσης, να πραγματοποιήσουν άμεσες αγορές.

Επιπλέον, οι πελάτες θα πρέπει να έχουν τη δυνατότητα να αναζητήσουν παραγωγούς, προϊόντα, κατηγορίες και προέλευση προϊόντων.

### **3.1.4. Ενημερώσεις**

Οι πελάτες θα πρέπει να λαμβάνουν ενημερώσεις σχετικά με την κατάσταση μιας παραγγελίας τους και με τυχόν εκκρεμείς αξιολογήσεις. Οι ενημερώσεις αυτές θα πρέπει να έχουν δύο καταστάσεις. Αναγνωσμένες ή μη αναγνωσμένες.

### **3.1.5. Αξιολογήσεις**

Μετά την αγορά προϊόντων μέσω του FarmIt, θα πρέπει να δίνεται η δυνατότητα στους πελάτες να αξιολογούν την ποιότητά τους και τον τρόπο εξυπηρέτησης των παραγωγών, ώστε να διασφαλίζεται ο αξιοκρατικός τρόπος λειτουργίας.

### **3.1.6. Μηνύματα**

Μία χρήσιμη λειτουργία, είναι η δυνατότητα αποστολής άμεσων μηνυμάτων μέσω της εφαρμογής προς τους παραγωγούς, για τυχόν ερωτήσεις ή διευκρινήσεις σχετικά με τα προϊόντα τους. Επομένως θα πρέπει να υπάρχει ένα «κιβώτιο» μηνυμάτων.



### 3.1.7. Ανάρτηση αγγελιών ζήτησης

Μία από τις βασικότερες λειτουργίες της εφαρμογής, είναι η προοπτική ανάρτησης αγγελιών ζήτησης από τους πελάτες, για προϊόντα που ενδιαφέρονται, στην ποσότητα που αποζητούν. Επιπλέον θα μπορούν να ορίζουν τη διάρκεια ισχύος της αγγελίας και τη μέγιστη τιμή στην οποία μπορούν να αγοράσουν τα προϊόντα που αυτή περιέχει.

Στη συνέχεια, μέχρι τη λήξη της αγγελίας, ο πελάτης θα λαμβάνει προσφορές από τους παραγωγούς, μεταξύ των οποίων θα γίνεται μια διαδικασία αντίστροφης δημοπρασίας. Δηλαδή, ανταγωνίζονται για το ποιος θα παρέχει τα προϊόντα που ζητήθηκαν, στη χαμηλότερη τιμή. Αν ο πελάτης δεχθεί κάποια από τις προσφορές, η δημοπρασία θα τερματίζεται.

### 3.1.8. Προφίλ

Απαραίτητη για τη σωστή λειτουργία της εφαρμογής, είναι η ύπαρξη προφίλ των πελατών, προκειμένου να έχουν πρόσβαση σε όλα τα στοιχεία τους και να τα επεξεργάζονται. Κάθε πελάτης, αφού εισέλθει στο λογαριασμό του, θα έχει τη δυνατότητα να προβάλει τα προσωπικά του στοιχεία, καθώς και τις διευθύνσεις που έχει καταχωρήσει. Επιπλέον, θα μπορεί να κάνει οποιαδήποτε αλλαγή επιθυμεί, καθώς επίσης να προσθέσει ή να διαγράψει διευθύνσεις, με την προϋπόθεση ότι πάντα θα υπάρχει τουλάχιστον μία. Τέλος, απαραίτητη κρίνεται και η δυνατότητα αλλαγής του κωδικού πρόσβασης ενός χρήστη, όπως επίσης και ένας σύνδεσμος επαναφοράς του, σε περίπτωση που δεν τον θυμάται.

### 3.1.9. Στατιστικά

Ένα χαρακτηριστικό που μπορεί να φανεί χρήσιμο στους πελάτες του FarmIt, είναι η δημιουργία στατιστικών από την χρήση τους. Κάθε πελάτης θα πρέπει να είναι σε θέση να γνωρίζει ανά πάσα στιγμή ορισμένα στοιχεία, όπως ο αριθμός των αγορών που έχει κάνει, ο αριθμός των αγγελιών που έχει δημιουργήσει, ο αριθμός των προσφορών που έχει αποδεχθεί κτλ.

Επιπλέον, αυτό το χαρακτηριστικό θα μπορούσε να φανεί χρήσιμο στην περίπτωση εφαρμογής ενός συστήματος επιβράβευσης μέσω πόντων, καθώς οι χρήστες θα είχαν τη δυνατότητα ανά πάσα στιγμή να προβάλλουν τον αριθμό των πόντων τους από τα στατιστικά τους. Για περισσότερες λεπτομέρειες, δείτε στο κεφάλαιο 9, στην ενότητα Μελλοντικές επεκτάσεις, την περιγραφή του συστήματος επιβράβευσης.

## 3.2. Μη λειτουργικές απαιτήσεις

Παρακάτω αναλύονται όλες οι απαιτήσεις που ορίστηκαν αρχικά και δεν αφορούν λειτουργίες της εφαρμογής.

### 3.2.1. Απόδοση και απόκριση

Στις μέρες μας, οι χρήστες των tablet και των κινητών, παραδόξως, έχουν περισσότερες απαιτήσεις από τους χρήστες των υπολογιστών. Αν η συσκευή δεν αποκρίνεται άμεσα κατά τη χρήση μιας εφαρμογής, δυσσαρעστούνται και υπάρχουν αρκετές πιθανότητες να τη διαγράψουν. Αυτό καθιστά τη διαδικασία βελτιστοποίησης ιδιαίτερα σημαντική για τους προγραμματιστές των εφαρμογών, αλλά ταυτόχρονα δύσκολη, καθώς η υπολογιστική ισχύς των φορητών συσκευών είναι μικρότερη από αυτή των υπολογιστών.

Επομένως, κατά τη δημιουργία μιας εφαρμογής, πρέπει να μελετηθεί η απόδοση και η απόκρισή της. Ωστόσο, οι δύο αυτοί όροι δεν πρέπει να συγχέονται, καθώς ο πρώτος αναφέρεται στην ικανότητα της εφαρμογής να απαντάει άμεσα στα αιτήματα του χρήστη, ενώ ο δεύτερος στην ικανότητά της να ανταποκρίνεται γενικά. Επιπλέον, υπάρχουν ορισμένα όρια, τα οποία προκύπτουν από τις δυνατότητες κάθε συσκευής, του λειτουργικού της και άλλους παράγοντες, όπως το δίκτυο. Για παράδειγμα μια εφαρμογή είναι πιθανό να χρειάζεται να κατεβάσει ορισμένα δεδομένα μέσω internet. Παρόλαυτα, θα πρέπει να υπάρχει άμεση ανταπόκριση στο αίτημα του χρήστη, με τεχνικές όπως η χρήση κινούμενων σχεδίων, που υποδεικνύουν το χρόνο που απομένει, διατηρώντας την απόκριση ως το σημαντικότερο παράγοντα. Σε αντίθετη περίπτωση, η εφαρμογή θα έδινε την αίσθηση της αργής απόδοσης, αφού η απάντηση θα δινόταν μετά το πέρας μιας χρονοβόρας διαδικασίας.

Για τους παραπάνω λόγους, αποφάσισα ότι το FarmIt θα πρέπει να έχει τα εξής χαρακτηριστικά:

- Άμεση απάντηση στο αίτημα του χρήστη, όπου αυτό είναι δυνατό και δημιουργία ψευδαισθήσεων, όπως η κίνηση ενός κουμπιού, όταν αυτό επιλέγεται.
- Ο χρήστης θα πρέπει να μπορεί να διακόψει οποιαδήποτε στιγμή επιθυμεί τη λειτουργία που πραγματοποιείται. Σε αντίθετη περίπτωση η εφαρμογή θα φαίνεται να μην αποκρίνεται.
- Ενημέρωση του χρήστη σχετικά με την πρόοδο της διεργασίας που εκτελείται, σε περίπτωση που αυτή είναι χρονοβόρα και δεν μπορεί να αποφευχθεί. Αυτό επιτυγχάνεται με τη χρήση progress bars και άλλων οπτικών ερεθισμάτων.

### 3.2.2. Χρηστικότητα

Μία εφαρμογή, προκειμένου να είναι επιτυχημένη, πρέπει να είναι εύκολη και απλή στο χειρισμό της, όσο πολύπλοκες κι αν είναι οι λειτουργίες και οι απαιτήσεις της, ώστε να έχει τη δυνατότητα να τη χρησιμοποιήσει οποιοσδήποτε, χωρίς ιδιαίτερη εκμάθηση.

Επομένως, το FarmIt, το οποίο απευθύνεται σε ανθρώπους χωρίς τεχνικές γνώσεις, είναι απαραίτητο να είναι απλό στη χρήση του. Για να επιτευχθεί αυτό, θα πρέπει να υπάρχει ένα απλό και σύντομο tutorial, το οποίο θα ενεργοποιείται κατά την πρώτη εκκίνηση της εφαρμογής. Επιπλέον, πολλές λειτουργίες, όπως η πρόσβαση στο κιβώτιο εισερχομένων, μπορούν να απλοποιηθούν με τη χρήση εικονιδίων.

### 3.2.3. Ασφάλεια

Σε μια εφαρμογή όπως το FarmIt, κρίνεται εξίσου σημαντική η ασφάλεια των προσωπικών δεδομένων των χρηστών. Θα πρέπει να αποτρέπονται τυχόν απώλειες ή διαρροές τους, με τη χρήση μηχανισμών προστασίας. Όλα τα ευαίσθητα δεδομένα όπως διευθύνσεις και κωδικοί πρόσβασης, θα πρέπει να αποθηκεύονται κωδικοποιημένα στη βάση και η εφαρμογή να μην τα «γνωρίζει». Επιπλέον η επικοινωνία θα πρέπει να γίνεται με https σύνδεση, για τη μέγιστη ασφάλεια κατά τη διάρκεια των αιτημάτων προς το server. Τέλος θα πρέπει να στέλνονται Application Programming Interface(API) keys για τις συνεδρίες των πελατών που εισέρχονται στο λογαριασμό τους, τα οποία είναι κάποιοι μοναδικοί κωδικοί, που χρησιμοποιούνται προκειμένου να αναγνωρίζονται οι χρήστες σε μια εφαρμογή και να διαχωρίζονται τα δικαιώματά τους σε αυτή.

## Κεφ.4: Εργαλεία που χρησιμοποιήθηκαν



Εικόνα 1: Do not reinvent the wheel

Σε αυτό το κεφάλαιο αναλύονται τα εργαλεία που χρησιμοποιήθηκαν κατά την ανάπτυξη του Farmlt. Η χρήση ορισμένων είναι προαιρετική και μια παρόμοια εφαρμογή μπορεί να υλοποιηθεί χωρίς τη χρήση τους ή με τη χρήση άλλων. Παρόλαυτα διευκολύνουν τον προγραμματισμό και δίνουν περισσότερες δυνατότητες. Ύστερα από προσεκτική μελέτη, επιλέχθηκαν αυτά που κρίθηκαν πιο συμβατά με τις απαιτήσεις του Farmlt.

### 4.1. Servers

Απαραίτητα εργαλεία αποτελούν οι servers, οι οποίοι θα φιλοξενήσουν τη βάση δεδομένων και το RESTful Web Service.



Εικόνα 2: WampServer

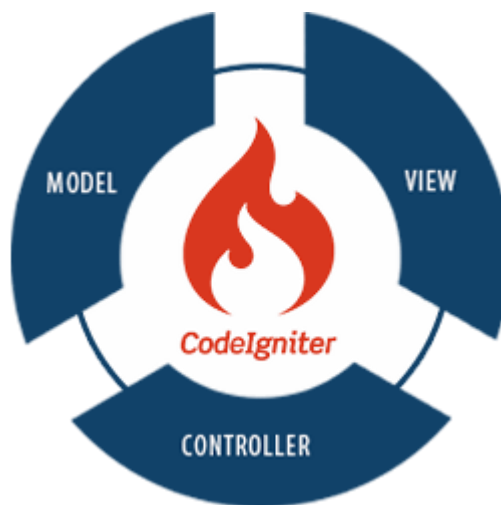
Για την ανάπτυξη του Farmlt, χρησιμοποιήθηκε ένα περιβάλλον που επιτρέπει την ανάπτυξη εφαρμογών διαδικτύου, ο WampServer, ο οποίος περιλαμβάνει Apache Server, PHP και MySQL Server. Επιπλέον περιλαμβάνει το PhpMyAdmin, το οποίο είναι ένα εργαλείο γραμμένο σε PHP και επιτρέπει την εύκολη διαχείριση της MySQL βάσης. Τέλος, με τις κατάλληλες ρυθμίσεις, υπάρχει δυνατότητα αυτόματου συγχρονισμού των αρχείων της βάσης δεδομένων σε cloud ή άλλους server, για τη διατήρηση αντιγράφων ασφαλείας.

Παρόμοια λογισμικά που μπορούν να χρησιμοποιηθούν εναλλακτικά, είναι τα παρακάτω:

- XAMPP: MariaDB, PHP, Perl
- UwAmp: Apache, MySQL, PHP, SQLite
- EasyPHP: Apache, MySQL, PHP
- AMPPS: Apache, Mysql, PHP, Perl, Python
- WnMp: Nginx, MariaDB, PHP

## 4.2. Frameworks

Τα frameworks που χρησιμοποιήθηκαν, είναι το CodeIgniter για το backend και το Ionic για το frontend.



Εικόνα 3: CodeIgniter

Το CodeIgniter είναι ένα ισχυρό framework ανοιχτού κώδικα, το οποίο επιτρέπει την ταχύτερη ανάπτυξη σε PHP. Βασίζεται στη μέθοδο σχεδίασης MVC, όπου υποχρεωτική είναι μόνο η χρήση των controllers. Επιπλέον υπάρχει δυνατότητα προσαρμογής, ώστε να χρησιμοποιεί Hierarchical Model View Controller (HMVC). Τέλος περιλαμβάνει ένα πλήθος βιβλιοθηκών, για ενέργειες όπως ερωτήματα στη βάση δεδομένων, κάνοντάς τις πολύ πιο απλές και ασφαλείς.



Εικόνα 4: Ionic

Το Ionic, είναι ένα framework, επίσης ανοιχτού κώδικα, το οποίο επιτρέπει την ταχύτερη ανάπτυξη υβριδικών εφαρμογών για κινητά τηλέφωνα με τη χρήση τεχνολογιών διαδικτύου. Χρησιμοποιεί το Apache Cordova, πρώην γνωστό ως PhoneGap και την AngularJS.

### 4.3. Εργαλεία για τον έλεγχο της λειτουργίας

Προκειμένου να διαπιστωθεί η σωστή λειτουργία του backend, ώστε να προχωρήσω στην ανάπτυξη του frontend, χρειάστηκε να ελεγχθούν οι λειτουργίες του REST Service, μέσω http κλήσεων. Για αυτόν το σκοπό, χρησιμοποίησα το Postman, το οποίο λειτουργεί ως REST Client και είναι add-on του Chrome.

Στη συνέχεια, προκειμένου να επιτηρηθεί η λειτουργία του frontend προτού εγκατασταθεί η εφαρμογή σε κινητό τηλέφωνο, χρησιμοποίησα το Ionic Lab, ένα λογισμικό το οποίο επιτρέπει την προβολή της εφαρμογής που προκύπτει από ένα πρόγραμμα που έχει αναπτυχθεί με τη χρήση του Ionic.

### 4.4. Integrated Development Environments (IDEs)

Για την ανάπτυξη του Farmlt, χρησιμοποιήθηκαν το PhpStorm για το backend και το WebStorm για το frontend.

### 4.5. Version Control System

Κατά την ανάπτυξη του Farmlt, χρησιμοποιήθηκε το git.

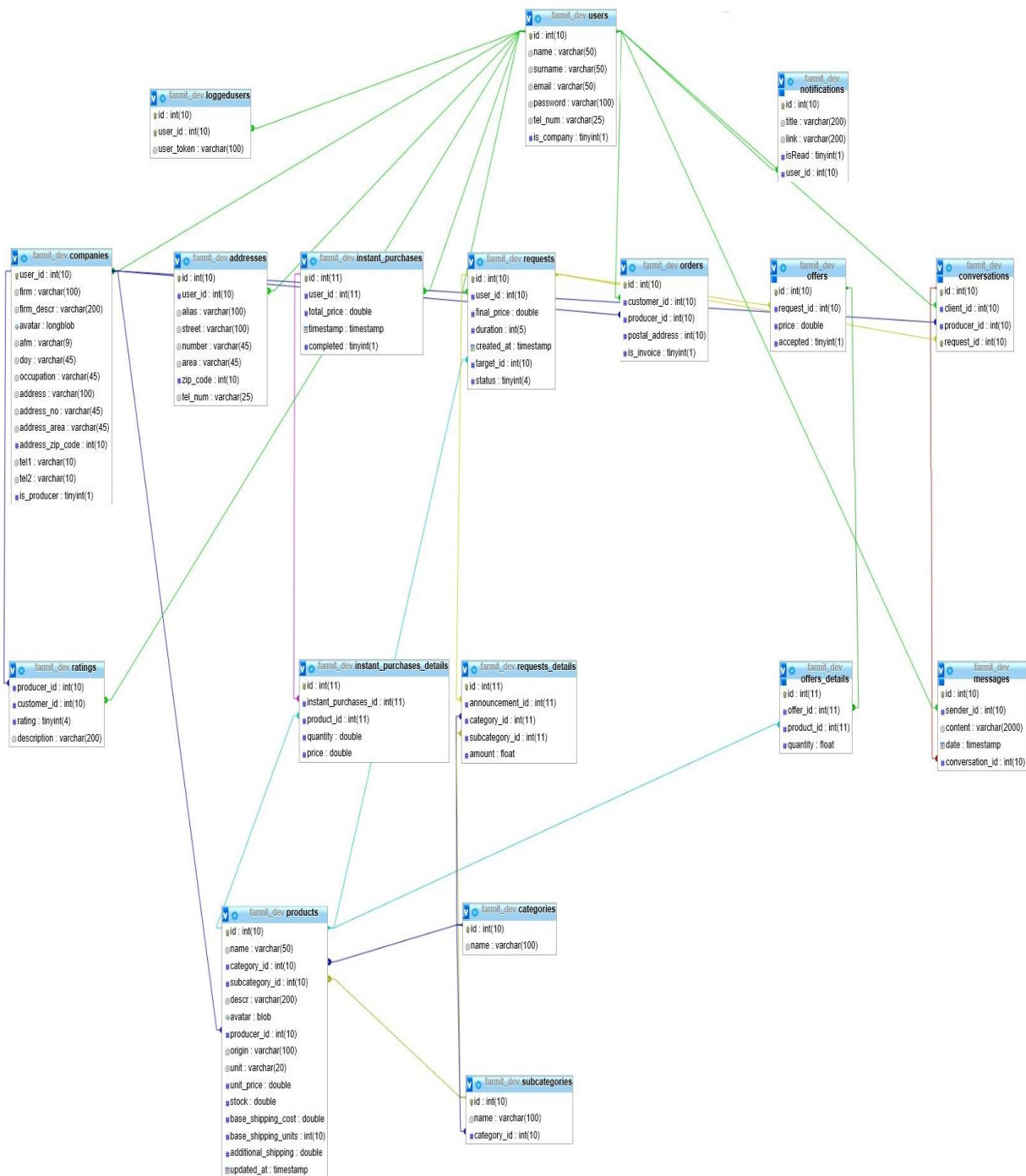
## 4.6. Κοινοποίηση εφαρμογής

Προκειμένου να δημοσιεύσει κάποιος μια εφαρμογή η οποία έχει αναπτυχθεί με τη χρήση του Ionic, αρκεί να την ανεβάσει στο Ionic View, το οποίο μπορεί να γίνει πολύ απλά μέσω του Ionic Lab, είτε μέσω του command line πληκτρολογώντας `ionic upload`. Μπορεί να φανεί ιδιαίτερα χρήσιμο στην περίπτωση που κάποιος θέλει να παρουσιάσει μια εφαρμογή στον πελάτη για τον οποίο προορίζεται, προτού δημιουργήσει native εφαρμογές για android, ios κτλ. ή τις ανεβάσει στα αντίστοιχα stores.

## Κεφ.5: Υλοποίηση εφαρμογής

Σε αυτό το κεφάλαιο, αναλύονται όλα τα δομικά στοιχεία που συνδέθηκαν για να αποτελέσουν το FarmIt.

### 5.1. Βάση Δεδομένων



Σχήμα 6: Βάση Δεδομένων του FarmIt



Η βάση δεδομένων συνοψίζεται ως εξής:

- **users:** είναι όλο οι χρήστες του FarmIt, είτε παραγωγοί, είτε πελάτες.
- **loggedusers:** είναι όλοι οι συνδεδεμένοι χρήστες.
- **notifications:** είναι οι ειδοποιήσεις των χρηστών, όπως για παράδειγμα μία αξιολόγηση που εκκρεμεί, μία νέα προσφορά κτλ.
- **companies:** είναι οι εταιρίες του FarmIt, δηλαδή οι παραγωγοί.
- **addresses:** είναι οι διευθύνσεις των χρηστών.
- **instant\_purchases:** είναι οι άμεσες αγορές των πελατών.
- **instant\_purchases\_details:** είναι οι λεπτομέρειες των άμεσων αγορών. Περιλαμβάνουν στοιχεία για κάθε προϊόν που περιέχει μία άμεση αγορά. Αυτή η τεχνική χρησιμοποιήθηκε γιατί διαφορετικά, θα ήταν αδύνατο να υπάρξει μία άμεση αγορά με περισσότερα από ένα προϊόντα και ο χρήστης θα αναγκαζόταν να πραγματοποιεί τη διαδικασία ξεχωριστά για κάθε προϊόν, χωρίς να διαθέτει καλάθι αγορών.
- **cart:** είναι το καλάθι αγορών των πελατών.
- **requests:** είναι οι αγγελίες που δημιουργούν οι πελάτες.
- **requests\_details:** είναι οι λεπτομέρειες των αγγελιών.
- **categories:** είναι οι κατηγορίες των προϊόντων, μεταξύ των οποίων μπορούν να επιλέξουν οι πελάτες κατά την ανάρτηση των αγγελιών τους.
- **subcategories:** είναι οι υποκατηγορίες των προϊόντων, μεταξύ των οποίων μπορούν να επιλέξουν οι πελάτες κατά την ανάρτηση των αγγελιών τους.
- **orders:** είναι οι παραγγελίες που λαμβάνουν οι παραγωγοί.
- **offers:** είναι οι προσφορές που γίνονται στις αγγελίες των πελατών.
- **offers\_details:** είναι οι λεπτομέρειες των προσφορών που γίνονται στις αγγελίες των πελατών.
- **products:** είναι τα προϊόντα που έχουν αναρτήσει οι παραγωγοί, τα οποία διαθέτουν κατηγορία και υποκατηγορία.
- **conversations:** είναι οι συνομιλίες των πελατών.
- **messages:** είναι τα μηνύματα των συνομιλιών.
- **ratings:** είναι οι βαθμολογίες των παραγωγών του FarmIt, σχετικά με τις αγορές που πραγματοποιούν οι πελάτες.

## 5.2. Backend

Για το Backend, χρησιμοποιήθηκε η βιβλιοθήκη CodeIgniter Rest Server, η οποία είναι επέκταση του framework. Περιλαμβάνει έτοιμες υλοποιήσεις για τους κώδικες κατάστασης των http αιτημάτων, τη διαχείριση και την επιστροφή τους, την ασφάλεια κτλ. Ιδιαίτερα χρήσιμη είναι η υλοποίηση των API keys που διαθέτει. Κάθε χρήστης του FarmIt, κατά τη σύνδεσή του λαμβάνει ένα API key στο κινητό του, το οποίο λήγει αν είναι αδρανής για μεγάλο χρονικό διάστημα ή αποσυνδεθεί. Αυτό χρησιμοποιείται από την εφαρμογή του frontend κατά την αποστολή των αιτημάτων και ελέγχεται στο backend κατά τη λήψη του, για την επιβεβαίωση της ταυτότητάς του. Σε περίπτωση που είναι λανθασμένο, επιστρέφεται αντίστοιχο μήνυμα.

Όλοι οι κωδικοί κρυπτογραφούνται προτού αποθηκευτούν στη βάση δεδομένων. Με αυτόν τον τρόπο αυξάνεται η ασφάλεια των προσωπικών δεδομένων των χρηστών, καθώς είναι δύσκολο να υποκλαπούν και η εφαρμογή δεν τους «μαθαίνει» ποτέ.

Το backend είναι κοινό για τους παραγωγούς και τους πελάτες. Παρακάτω, αναλύονται τα βασικά κομμάτια του REST Service, που χρησιμοποιούνται από την εφαρμογή των πελατών.

### 5.2.1. Controllers

Οι Controllers που υλοποιήθηκαν στο backend, αναλύονται λεπτομερώς παρακάτω:

- **Addresses:** Αναλαμβάνει τα αιτήματα που αφορούν τις διευθύνσεις. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Δημιουργία διεύθυνσης για έναν χρήστη.
  - Εύρεση όλων των διευθύνσεων ενός χρήστη.
  - Επεξεργασία μίας διεύθυνσης.
  - Διαγραφή μίας διεύθυνσης.
- **Announcements:** Περιλαμβάνει μέθοδο η οποία αναλαμβάνει τα αιτήματα που αφορούν την ανάρτηση νέων αγγελιών από τους πελάτες.
- **Cart:** Αναλαμβάνει τα αιτήματα που αφορούν τα καλάθια αγορών των χρηστών. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εύρεση του καλαθιού ενός πελάτη.
  - Τροποποίηση του καλαθιού ενός πελάτη.
  - Διαγραφή του καλαθιού ενός πελάτη.
- **Categories:** Αναλαμβάνει τα αιτήματα που αφορούν τις κατηγορίες των προϊόντων. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εύρεση των κατηγοριών των προϊόντων.
  - Εύρεση των υποκατηγοριών μιας κατηγορίας προϊόντων.

- **InstantPurchases:** Περιλαμβάνει μέθοδο η οποία αναλαμβάνει τα αιτήματα που αφορούν τη δημιουργία άμεσων αγορών από τους πελάτες.
- **Offers:** Αναλαμβάνει τα αιτήματα που αφορούν τις προσφορές των πελατών. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Λήψη των προσφορών ενός πελάτη.
  - Αποδοχή μιας προσφοράς από έναν πελάτη.
- **Products:** Αναλαμβάνει τα αιτήματα που αφορούν τα προϊόντα. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Αναζήτηση παραγωγών, προϊόντων και προέλευσης προϊόντων με μία ή περισσότερες λέξεις ως κλειδί και επιστροφή αποτελεσμάτων που μοιάζουν με αυτό. Αναζητά με βάση έναν αριθμό που υποδεικνύει από πού θα ξεκινάει η αναζήτηση και επιστρέφει κάθε φορά έως δέκα αποτελέσματα, ώστε να περιορίζεται ο όγκος των δεδομένων στο κινητό, καθώς οι παραγωγοί είναι πολλοί.
  - Λήψη των προϊόντων ενός παραγωγού.
- **Users:** Αναλαμβάνει τα αιτήματα που αφορούν τους χρήστες της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Δημιουργία λογαριασμού χρήστη.
  - Είσοδος σε έναν λογαριασμό.
  - «Κρυφή» είσοδος ενός χρήστη. Χρησιμοποιείται στην περίπτωση που κάποιος έχει επιλέξει τη δυνατότητα να συνδέεται αυτόματα όταν ανοίγει την εφαρμογή από το κινητό του.
  - Αλλαγή των στοιχείων του λογαριασμού ενός χρήστη.
  - Αλλαγή του κωδικού του λογαριασμού ενός χρήστη.
  - Έξοδος από έναν λογαριασμό.
  - Διαγραφή ενός λογαριασμού.
  - Δημιουργία API key για έναν χρήστη.
  - Διαγραφή του API key ενός χρήστη.
  - Εύρεση όλων των παραγωγών. Χρησιμοποιείται πάλι η μέθοδος για τον περιορισμό του αριθμού των δεδομένων που επιστρέφονται. Λαμβάνονται κάθε φορά έως δέκα παραγωγοί.
  - Εύρεση ενός συγκεκριμένου χρήστη.
  - Υπολογισμός των στατιστικών χρήσης ενός λογαριασμού.

## 5.2.2. Models

Τα Models που υλοποιήθηκαν στο backend, αναλύονται λεπτομερώς παρακάτω:

- **Addresses\_model:** Αναλαμβάνει τη διαχείριση των διευθύνσεων των χρηστών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εισαγωγή μίας διεύθυνσης στη βάση δεδομένων.
  - Εύρεση μίας διεύθυνσης με βάση το id ενός χρήστη.
  - Επεξεργασία μίας διεύθυνσης στη βάση δεδομένων.
  - Υπολογισμός του αριθμού των διευθύνσεων ενός χρήστη. Χρησιμοποιείται κατά την είσοδο στο λογαριασμό ενός χρήστη, στον «Users» Controller, ώστε να διαπιστωθεί αν έχει ολοκληρώσει τη διαδικασία εγγραφής του. Επίσης, χρησιμοποιείται ως ένδειξη στη σελίδα του προφίλ του χρήστη.
  - Διαγραφή μίας διεύθυνσης με βάση το id ενός χρήστη.
- **Announcements\_model:** Αναλαμβάνει τη διαχείριση των αγγελιών των πελατών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εισαγωγή μίας αγγελίας στη βάση για έναν πελάτη.
  - Διαγραφή των αγγελιών ενός πελάτη από τη βάση. Χρησιμοποιείται κατά τη διαγραφή του προφίλ του.
- **Cart\_model:** Αναλαμβάνει τη διαχείριση του καλαθιού αγορών των πελατών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Δημιουργία ενός καλαθιού για έναν πελάτη.
  - Εύρεση του καλαθιού ενός πελάτη.
  - Τροποποίηση του καλαθιού ενός πελάτη.
  - Διαγραφή του καλαθιού ενός πελάτη.
- **Categories\_model:** Αναλαμβάνει τη διαχείριση των κατηγοριών και των υποκατηγοριών των προϊόντων της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εύρεση όλων των κατηγοριών των προϊόντων από τη βάση δεδομένων.
  - Εύρεση όλων των υποκατηγοριών μιας κατηγορίας προϊόντων από τη βάση δεδομένων.
- **Instant\_purchases\_model:** Αναλαμβάνει τη διαχείριση των άμεσων αγορών των πελατών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Δημιουργία άμεσης αγοράς για έναν πελάτη.
  - Διαγραφή όλων των άμεσων αγορών ενός πελάτη. Χρησιμοποιείται κατά τη διαγραφή του προφίλ του.
- **Offers\_model:** Αναλαμβάνει τη διαχείριση των προσφορών που αφορούν τους πελάτες της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εύρεση των προσφορών που έχουν γίνει σε έναν πελάτη, από τη βάση δεδομένων.
  - Χαρακτηρισμός μιας προσφοράς ως αποδεκτή στη βάση δεδομένων.

- **Products\_model:** Αναλαμβάνει τη διαχείριση των προϊόντων της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Αναζήτηση παραγωγών, προϊόντων και προέλευσης προϊόντων με μία ή περισσότερες λέξεις ως κλειδί, στη βάση δεδομένων. Ο όγκος των δεδομένων που επιστρέφονται, περιορίζονται όπως αναφέρθηκε προηγουμένως στον «Products» Controller.
  - Εύρεση των προϊόντων ενός παραγωγού.
- **Users\_model:** Αναλαμβάνει τη διαχείριση των χρηστών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Εύρεση του id ενός χρήστη με συγκεκριμένο email, από τη βάση δεδομένων.
  - Προσθήκη ενός νέου χρήστη στη βάση δεδομένων.
  - Εύρεση ενός χρήστη από τη βάση δεδομένων.
  - Εύρεση του κωδικού ενός χρήστη με συγκεκριμένο email, από τη βάση δεδομένων. Όπως αναφέρθηκε παραπάνω, οι κωδικοί είναι κρυπτογραφημένοι.
  - Τροποποίηση ενός χρήστη στη βάση δεδομένων.
  - Διαγραφή ενός χρήστη από τη βάση δεδομένων.
  - Τροποποίηση ενός προϊόντος στη βάση δεδομένων.
  - Διαγραφή ενός προϊόντος από τη βάση δεδομένων.
  - Εύρεση όλων των παραγωγών από τη βάση δεδομένων. Χρησιμοποιείται η μέθοδος για τον περιορισμό του αριθμού των δεδομένων που επιστρέφονται. Λαμβάνονται κάθε φορά έως δέκα παραγωγοί.
  - Υπολογισμός των στατιστικών χρήσης ενός λογαριασμού.
- **Logged\_users\_model:** Αναλαμβάνει τη διαχείριση των συνδεδεμένων χρηστών της εφαρμογής. Περιλαμβάνει μεθόδους για τις παρακάτω λειτουργίες:
  - Προσθήκη ενός χρήστη στους συνδεδεμένους.
  - Αφαίρεση ενός χρήστη από τους συνδεδεμένους.
  - Εύρεση του API key ενός χρήστη.

### 5.3. Frontend

Στο Frontend όλα τα αιτήματα αποστέλλονται μέσω https. Επιπλέον οι κωδικοί πρόσβασης δεν αποθηκεύονται με κανέναν τρόπο, αυξάνοντας έτσι την ασφάλεια των προσωπικών δεδομένων των χρηστών. Κατά τη διάρκεια των αιτημάτων προς το REST Service, όλες οι χρονοβόρες και πολύπλοκες διαδικασίες, γίνονται σε αυτό και όχι στο κινητό του χρήστη. Επιστρέφονται μόνο τα αποτελέσματα, ώστε να είναι ταχύτερη η εφαρμογή.

Το frontend αποτελείται από δύο ξεχωριστές εφαρμογές, μία για τους παραγωγούς και μία για τους πελάτες. Παρακάτω, αναλύεται η εφαρμογή των πελατών.

### 5.3.1. Templates

- **farmit:** Η αρχική σελίδα της εφαρμογής. Περιλαμβάνει δύο κουμπιά, ένα για τη δημιουργία λογαριασμού πελάτη και ένα για την είσοδο ως πελάτης στο farmit.
- **signup:** Η σελίδα για τη δημιουργία ενός λογαριασμού.
- **signup-form:** Η σελίδα που περιέχει τη φόρμα δημιουργίας λογαριασμού. Περιέχει πεδία για όλα τα απαραίτητα στοιχεία που ο πελάτης αιτείται να συμπληρώσει, μηνύματα λάθους σε περιπτώσεις εισαγωγής μη έγκυρων πεδίων και κουμπιά για τη δυνατότητα αυτόματης σύνδεσης του πελάτη στο λογαριασμό του στο μέλλον στη συγκεκριμένη συσκευή, την εμφάνιση και την απόκρυψη του κωδικού πρόσβασης που πληκτρολογεί και τη δημιουργία του λογαριασμού.
- **login:** Η σελίδα για την είσοδο σε λογαριασμό.
- **login-form:** Η σελίδα που περιέχει τη φόρμα εισόδου σε λογαριασμό. Είναι όμοια με τη σελίδα δημιουργίας λογαριασμού, με τη διαφορά ότι τα αιτούμενα πεδία είναι μόνο το email και ο κωδικός του πελάτη και όχι όλα τα στοιχεία του. Επιπλέον διαθέτει επιλογή ανάκτησης του κωδικού πρόσβασης μέσω email.
- **home:** Η κεντρική σελίδα, μετά την είσοδο σε κάποιο λογαριασμό. Περιέχει ένα μενού στα αριστερά, το οποίο ανοιγοκλείνει με σύρσιμο ή με την επιλογή ενός κουμπιού, και ένα κουμπί για εμφάνιση των ενημερώσεων του πελάτη.
- **home-left-menu:** Το αριστερό μενού της σελίδας home. Περιέχει επιλογές για άνοιγμα του προφίλ, του καλαθιού αγορών, των προσφορών και των στατιστικών, για αποσύνδεση και διαγραφή προφίλ και για περισσότερες πληροφορίες σχετικά με το FarmIt.
- **home-menu-content:** Το περιεχόμενο της σελίδας home. Διαθέτει πεδίο για την αναζήτηση παραγωγών και προϊόντων. Σε περίπτωση που ο χρήστης έχει αναζητήσει κάτι, εμφανίζονται τα αποτελέσματα. Διαφορετικά, εμφανίζονται όλοι οι παραγωγοί. Επιπλέον υπάρχει δυνατότητα ανανέωσης με σύρσιμο. Τέλος υπάρχουν δύο αιωρούμενα κουμπιά στα κάτω άκρα της σελίδας. Ένα για να πηγαίνει ο χρήστης ξανά στην κορυφή, το οποίο εμφανίζεται αν φτάσει σε ένα κατώτατο σημείο που έχει οριστεί και ένα για τη δημιουργία αγγελίας.
- **providers-card:** Καρτέλες που περιέχουν τους παραγωγούς. Περιέχει έναν infinite-scroller, δηλαδή κάθε φορά που ο χρήστης φτάνει στο κατώτερο σημείο της σελίδας, φορτώνονται περισσότερα δεδομένα και στη συγκεκριμένη περίπτωση, περισσότερες καρτέλες με παραγωγούς. Αυτό γίνεται για λόγους απόδοσης, ώστε να μη φορτώνονται περιττά δεδομένα.
- **providers-search-card:** Καρτέλες που περιέχουν τους παραγωγούς και τα προϊόντα που προκύπτουν μετά από μία αναζήτηση. Περιέχει έναν infinite-scroller.
- **provider:** Η σελίδα που περιέχει στοιχεία και προϊόντα που αφορούν έναν συγκεκριμένο παραγωγό, σε περίπτωση που κάποιος τον επιλέξει. Αν ο πελάτης έχει αναζητήσει κάτι, εμφανίζονται μόνο τα σχετικά προϊόντα. Διαφορετικά, εμφανίζονται όλα τα προϊόντα του παραγωγού. Επιπλέον, στην πάνω δεξιά γωνία, υπάρχει κουμπί για πρόσβαση στο καλάθι αγορών, με ένδειξη του αριθμού των διαφορετικών προϊόντων που περιέχει.

- provider-info-card: Καρτέλες που περιέχουν τα στοιχεία και τα προϊόντα του παραγωγού.
- provider-info-after-search-card: Καρτέλες που περιέχουν τα στοιχεία και τα προϊόντα του παραγωγού, που ταιριάζουν με μία αναζήτηση. Επιπλέον, περιέχει κουμπί για τη φόρτωση όλων των προϊόντων του παραγωγού.
- profile: Η σελίδα που περιέχει τα στοιχεία του πελάτη. Επίσης, στην πάνω δεξιά γωνία, υπάρχει κουμπί για πρόσβαση στις διευθύνσεις που έχει εισάγει, με ένδειξη του αριθμού τους.
- profile-form: Η φόρμα με τα στοιχεία του πελάτη, με δυνατότητα επεξεργασίας τους και τα απαραίτητα μηνύματα λάθους σε περιπτώσεις εισαγωγής μη έγκυρων πεδίων. Επιπλέον, περιέχει εικονίδιο αποθήκευσης των αλλαγών, το οποίο εμφανίζεται μόνο αν κάποιο πεδίο έχει διαφοροποιηθεί. Τέλος περιέχει κουμπί για την αλλαγή του κωδικού πρόσβασης του πελάτη.
- change-password: Η φόρμα για την αλλαγή του κωδικού πρόσβασης. Περιέχει πεδία για την εισαγωγή του παλιού κωδικού πρόσβασης και δύο φορές του καινούργιου, για λόγους επιβεβαίωσης. Επιπλέον, περιέχει τα απαραίτητα μηνύματα λάθους.
- delete-profile: Η σελίδα για τη διαγραφή του προφίλ του συνδεδεμένου πελάτη.
- delete-profile-form: Η φόρμα που περιέχει τα απαραίτητα πεδία και μηνύματα λάθους για τη διαγραφή του προφίλ. Επιπλέον διαθέτει επιλογή για αποστολή email σε περίπτωση που ο πελάτης ξέχασε τον κωδικό πρόσβασής του, ώστε να δημιουργήσει νέο, μέσω ενός ασφαλούς συνδέσμου.
- addresses: Η σελίδα που περιέχει όλες τις διευθύνσεις του πελάτη. Υπάρχει δυνατότητα για επεξεργασία, διαγραφή και προσθήκη διευθύνσεων.
- first-address: Η σελίδα για τη δημιουργία της πρώτης διεύθυνσης του πελάτη.
- new-address-form: Φόρμα για τη δημιουργία διευθύνσεων. Περιέχει όλα τα απαραίτητα πεδία και μηνύματα λάθους.
- edit-address-form: Φόρμα για την επεξεργασία μιας διεύθυνσης. Επιπλέον, περιέχει εικονίδιο αποθήκευσης των αλλαγών, το οποίο εμφανίζεται μόνο αν κάποιο πεδίο έχει διαφοροποιηθεί.
- announcement: Η σελίδα για την προσθήκη προϊόντων με σκοπό τη δημιουργία μίας αγγελίας ζήτησης. Επιπλέον, στην πάνω δεξιά γωνία, υπάρχει κουμπί για πρόσβαση στην αγγελία που έχει δημιουργήσει ο πελάτης, με ένδειξη του αριθμού των διαφορετικών προϊόντων που έχει εισάγει.
- announcement-form: Η φόρμα που περιέχει τα απαραίτητα πεδία και μηνύματα λάθους για την προσθήκη ενός νέου προϊόντος στη φόρμα αγγελίας του πελάτη.
- announcement-modal: Ένα modal, δηλαδή μία σελίδα που εμφανίζεται με animation πάνω από την υπάρχουσα και περιέχει την αγγελία που έχει δημιουργήσει ο πελάτης. Επιπλέον, υπάρχουν πεδία για την επιλογή του χρονικού ορίου κατά το οποίο θα παραμείνει ενεργή η αγγελία και προαιρετικά, την ανώτερη τιμή προσφοράς που επιθυμεί να λάβει ο πελάτης. Τέλος υπάρχει δυνατότητα επεξεργασίας της ποσότητας των προϊόντων της αγγελίας που έχει δημιουργηθεί.

- **offers:** Η σελίδα που περιέχει όλες τις προσφορές που έχουν γίνει στον πελάτη. Υπάρχει δυνατότητα ανανέωσης με σύρσιμο.
- **offers-list:** Η λίστα που περιέχει τις προσφορές. Οι ενεργές εμφανίζονται με πράσινη σκίαση και οι ανενεργές με γκρι. Επιπλέον ο πελάτης μπορεί να αποδεχθεί μία ενεργή προσφορά.
- **cart:** Ένα modal, με το καλάθι αγορών του πελάτη. Περιέχει όλα τα προϊόντα που έχει προσθέσει για άμεση αγορά, με την ποσότητά τους και την τιμή που αντιστοιχεί σε αυτήν. Επιπλέον, αναφέρονται τα συνολικά έξοδα αποστολής, τυχόν πρόσθετοι φόροι και η τελική τιμή τους. Τέλος, υπάρχει δυνατότητα επεξεργασίας της ποσότητας ή διαγραφής ενός προϊόντος και κουμπί για την άμεση αγορά των προϊόντων που βρίσκονται στο καλάθι αγορών.
- **add-to-cart:** Περιλαμβάνει τα απαραίτητα πεδία και μηνύματα λάθους για την εισαγωγή ενός προϊόντος στο καλάθι αγορών.
- **edit-product-quantity:** Περιλαμβάνει πεδίο για την αλλαγή της ποσότητας ενός προϊόντος.
- **chat:** Η σελίδα που περιλαμβάνει τα μηνύματα μίας συζήτησης.
- **evaluation:** Η σελίδα για την αξιολόγηση ενός παραγωγού και των προϊόντων του. Περιλαμβάνει βαθμολόγηση με αστέρια, πεδίο για κείμενο και ένα κουμπί για την υποβολή της αξιολόγησης.
- **loader:** Περιγράφει τη μορφή που έχουν οι loaders.
- **notifications-popover:** Ένα popover, δηλαδή μια σελίδα που καλύπτει ένα μικρό τμήμα του περιεχομένου της εφαρμογής, το οποίο περιλαμβάνει όλες τις νέες ειδοποιήσεις του πελάτη.
- **statistics:** Η σελίδα που περιέχει τα στατιστικά χρήσης του πελάτη.
- **about:** Η σελίδα που περιέχει πληροφορίες για την εφαρμογή.

Εκτός από τα templates, υπάρχει και η σελίδα εκκίνησης της εφαρμογής. Αυτή περιέχει όλα τα meta data, τον τίτλο της εφαρμογής, τον ορισμό των css και javascript αρχείων, το body της εφαρμογής και την μπάρα που χρησιμοποιείται από τα templates, η οποία είναι απενεργοποιημένη στην κεντρική σελίδα για αισθητικούς λόγους. Επιπλέον, όλες οι σελίδες εκτός από το home, διαθέτουν κουμπί για επιστροφή στην προηγούμενη σελίδα, πάνω αριστερά στην μπάρα.

### 5.3.2. Controllers

- **FarmitItCtrl:** Είναι ο controller της αρχικής σελίδας. Αναλαμβάνει την είσοδο στο λογαριασμό του πελάτη και την ανακατεύθυνση στην κεντρική σελίδα σε περίπτωση που έχει επιλέξει να συνδέεται αυτόματα.
- **SignUpCtrl:** Αναλαμβάνει την εγγραφή ενός νέου πελάτη. Επιπλέον, αποθηκεύει στη local storage το email του και το API key, που λαμβάνεται από το REST Service, καθώς και την επιλογή του για το αν θέλει να συνδέεται αυτόματα στο μέλλον.



- **LoginCtrl:** Αναλαμβάνει την είσοδο του πελάτη στο λογαριασμό του. Επιπλέον, αποθηκεύει στη local storage το email του και το API key, που λαμβάνεται από το REST Service, καθώς και την επιλογή του για το αν θέλει να συνδέεται αυτόματα στο μέλλον.
- **LogoutCtrl:** Αναλαμβάνει την έξοδο του πελάτη από το λογαριασμό του. Επιπλέον, αρχικοποιεί τη local storage.
- **DeleteProfileCtrl:** Αναλαμβάνει τη διαγραφή του λογαριασμού του πελάτη. Επιπλέον, αρχικοποιεί τη local storage.
- **ProfileCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με το προφίλ του πελάτη. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Αρχικοποίηση του προφίλ με τα στοιχεία του πελάτη και τον αριθμό των διευθύνσεών του.
  - Άνοιγμα και κλείσιμο του modal με τις διευθύνσεις του πελάτη.
  - Ανίχνευση αλλαγών στα στοιχεία του προφίλ και των διευθύνσεων.
  - Αλλαγή του κωδικού πρόσβασης.
  - Επεξεργασία μίας διεύθυνσης.
  - Άνοιγμα και κλείσιμο ενός popup για τη δημιουργία νέας διεύθυνσης.
  - Δημιουργία νέας διεύθυνσης.
  - Διαγραφή μίας διεύθυνσης.
  - Αποθήκευση των αλλαγών.
- **HomeCtrl:** Είναι ο controller της κεντρικής σελίδας. Αναλαμβάνει τη λειτουργικότητα για το αριστερό μενού και την εμφάνιση του διαλόγου about.
- **HomeMenuCtrl:** Αναλαμβάνει τη λειτουργικότητα για το βασικό μενού της κεντρικής σελίδας. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Εμφάνιση δέκα παραγωγών και αυτόματη συμπλήρωση δέκα επιπλέον, κάθε φορά που ο πελάτης φτάνει στο κατώτερο σημείο της σελίδας.
  - Ανανέωση της σελίδας. Συνεπώς ανανέωση των παραγωγών και έλεγχος νέων ειδοποιήσεων.
  - Άνοιγμα και κλείσιμο του popover με τις ειδοποιήσεις.
  - Κάθε φορά που ο πελάτης πληκτρολογεί κάτι στην αναζήτηση, αυτό πραγματοποιείται αυτόματα. Συνεπώς μπορεί να βλέπει κάθε στιγμή τα αποτελέσματα, χωρίς να χρειάζεται να πατήσει κάποιο κουμπί.
  - Ανακατεύθυνση στην κορυφή της σελίδας.
  - Ανακατεύθυνση στη σελίδα για τη δημιουργία αγγελίας.
  - Ανακατεύθυνση σε σελίδα με τα στοιχεία ενός παραγωγού που επιλέχθηκε.
- **AddressCtrl:** Είναι ο controller της σελίδας για την πρώτη διεύθυνση του πελάτη και αναλαμβάνει τη δημιουργία της.

- **ProviderCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με έναν παραγωγό και τις άμεσες αγορές. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Αρχικοποίηση του καλαθιού αγορών του πελάτη με τα προϊόντα του.
  - Λήψη των προϊόντων του παραγωγού.
  - Προσθήκη ενός προϊόντος στο καλάθι.
  - Άνοιγμα και κλείσιμο του modal με το καλάθι αγορών του πελάτη.
  - Επεξεργασία της ποσότητας ενός προϊόντος.
  - Διαγραφή ενός προϊόντος.
  - Πραγματοποίηση άμεσης αγοράς όλων των προϊόντων που βρίσκονται στο καλάθι αγορών.
- **CartCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με το καλάθι αγορών του πελάτη. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Αρχικοποίηση του καλαθιού αγορών του πελάτη με τα προϊόντα του.
  - Άνοιγμα και κλείσιμο του καλαθιού αγορών.
  - Επεξεργασία της ποσότητας ενός προϊόντος στο καλάθι αγορών.
  - Διαγραφή ενός προϊόντος από το καλάθι αγορών.
  - Πραγματοποίηση άμεσης αγοράς όλων των προϊόντων που βρίσκονται στο καλάθι αγορών.
- **AnnouncementCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με τις αγγελίες του πελάτη. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Λήψη των διαθέσιμων κατηγοριών.
  - Λήψη των διαθέσιμων υποκατηγοριών μίας συγκεκριμένης κατηγορίας.
  - Προσθήκη ενός προϊόντος στην αγγελία του πελάτη.
  - Άνοιγμα και κλείσιμο του modal της αγγελίας του πελάτη.
  - Επεξεργασία της ποσότητας ενός προϊόντος στην αγγελία.
  - Διαγραφή ενός προϊόντος από την αγγελία.
  - Δημοσίευση της αγγελίας του πελάτη.
- **OffersCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με τις προσφορές του πελάτη. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Λήψη όλων των προσφορών του πελάτη.
  - Αποδοχή μίας προσφοράς.
  - Ανανέωση της σελίδας των προσφορών.
- **EvaluationCtrl:** Αναλαμβάνει τη λειτουργικότητα για την αξιολόγηση των παραγωγών και των προϊόντων τους.
- **ChatCtrl:** Αναλαμβάνει τη λειτουργικότητα σχετικά με τις συνομιλίες του πελάτη. Περιλαμβάνει μεθόδους για τις παρακάτω ενέργειες:
  - Λήψη όλων των συνομιλιών του πελάτη.
  - Λήψη όλων των μηνυμάτων μίας συνομιλίας.
  - Αποστολή ενός μηνύματος.
  - Αυτόματη, περιοδική ανανέωση των μηνυμάτων ανά 5 δευτερόλεπτα.
- **StatisticsCtrl:** Αναλαμβάνει τη λήψη των στατιστικών χρήσης του πελάτη.

### 5.3.3. Services

- **SignUpService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη δημιουργία του λογαριασμού ενός πελάτη.
- **LoginService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη σύνδεση στο λογαριασμό ενός πελάτη ή τη σύνδεση στο παρασκήνιο, για την περίπτωση που ο χρήστης έχει επιλέξει να συνδέεται αυτόματα.
- **RememberMeService:** Περιλαμβάνει μέθοδο η οποία ελέγχει αν ο χρήστης έχει επιλέξει να συνδέεται αυτόματα στο λογαριασμό του και αναλόγως, καλεί ή όχι την αντίστοιχη μέθοδο του LoginService.
- **LogoutService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για την αποσύνδεση ενός πελάτη από το λογαριασμό του.
- **DeleteProfileService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη διαγραφή του λογαριασμού ενός πελάτη.
- **ProfileService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αιτήματα που αφορούν το προφίλ του πελάτη. Τα αιτήματα αυτά ανάγονται στις παρακάτω ενέργειες:
  - Λήψη των στοιχείων του προφίλ ενός πελάτη.
  - Τροποποίηση των στοιχείων του προφίλ ενός πελάτη.
  - Αλλαγή του κωδικού χρήσης ενός πελάτη.
- **AddressService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αιτήματα που αφορούν τις διευθύνσεις του πελάτη. Τα αιτήματα αυτά ανάγονται στις παρακάτω ενέργειες:
  - Δημιουργία διεύθυνσης.
  - Λήψη όλων των διευθύνσεων του πελάτη.
  - Επεξεργασία μίας διεύθυνσης ενός πελάτη.
  - Διαγραφή μίας διεύθυνσης ενός πελάτη.
- **GetProvidersService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αιτήματα που αφορούν τους παραγωγούς. Τα αιτήματα αυτά ανάγονται στις παρακάτω ενέργειες:
  - Λήψη του επιτρεπόμενου αριθμού παραγωγών, από ένα συγκεκριμένο σημείο αναφοράς και μετά.
  - Λήψη του επιτρεπόμενου αριθμού των αποτελεσμάτων αναζήτησης, από ένα συγκεκριμένο σημείο αναφοράς και μετά.
  - Λήψη όλων των προϊόντων ενός παραγωγού.
- **CartService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αιτήματα που αφορούν το καλάθι αγορών του πελάτη. Τα αιτήματα αυτά ανάγονται στις παρακάτω ενέργειες:
  - Άδειασμα του καλαθιού αγορών.
  - Λήψη του καλαθιού αγορών.
  - Τροποποίηση του καλαθιού αγορών.
  - Άμεση αγορά των προϊόντων που βρίσκονται στο καλάθι αγορών.

- **PurchasesService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη δημιουργία της άμεσης αγοράς ενός πελάτη.
- **CategoriesService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα για τη λήψη όλων των κατηγοριών των προϊόντων.
- **SubcategoriesService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα για τη λήψη όλων των υποκατηγοριών ενός συγκεκριμένου προϊόντος.
- **AnnouncementService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα, μαζί με τα απαραίτητα στοιχεία, για τη δημιουργία μίας αγγελίας.
- **OffersService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αίτημα στο REST Service για τη λήψη όλων των προσφορών του πελάτη και την αποδοχή μίας εξ αυτών.
- **EvaluationService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη δημιουργία μίας αξιολόγησης.
- **ChatService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αιτήματα που αφορούν τις συνομιλίες του πελάτη. Τα αιτήματα αυτά ανάγονται στις παρακάτω ενέργειες:
  - Λήψη όλων των συνομιλιών του πελάτη.
  - Λήψη όλων των μηνυμάτων μίας συνομιλίας.
  - Δημιουργία νέου μηνύματος.
- **StatisticsService:** Περιλαμβάνει μέθοδο η οποία στέλνει αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για τη λήψη των στατιστικών χρήσης ενός πελάτη.
- **EmailService:** Περιλαμβάνει μεθόδους οι οποίες στέλνουν αίτημα στο REST Service, μαζί με τα απαραίτητα στοιχεία, για την αποστολή email στον πελάτη σχετικά με την ανάκτηση του κωδικού πρόσβασής του ή την επιβεβαίωση μίας αγοράς που μόλις πραγματοποιήσε.
- **NetworkHelperService:** Περιλαμβάνει μεθόδους οι οποίες ελέγχουν αν υπάρχει σύνδεση στο internet τη δεδομένη στιγμή ή προσθέτουν έναν listener, ο οποίος ελέγχει συνεχόμενα τη σύνδεση και σε περίπτωση που διακοπεί, εμφανίζουν αντίστοιχο μήνυμα.

## Κεφ.6: Ιδιαίτερα σημεία

Κατά την υλοποίηση του FarmIt, χρειάστηκε να εφαρμοστούν τεχνικές που θα εξασφάλιζαν τη βέλτιστη λειτουργία. Οι πιο βασικές τεχνικές, αναλύονται σε αυτό το κεφάλαιο.

### 6.1. Transactions

Στο backend, σε ορισμένες μεθόδους, χρησιμοποιήθηκαν transactions με τη βάση δεδομένων, ώστε αν προκύψει κάτι στην πορεία των αιτημάτων, να υπάρχει δυνατότητα να επιστρέψει στην αρχική κατάσταση που βρισκόταν (rollback).

### 6.2. Infinite Scroller

Στο frontend, όπως αναφέρθηκε προηγουμένως, χρησιμοποιήθηκε infinite scroller στην αρχική σελίδα, για την προβολή των παραγωγών. Γι' αυτό το λόγο, δημιουργήθηκε μία μέθοδος η οποία καλείται κάθε φορά που ο χρήστης φτάνει στο κατώτερο σημείο της σελίδας. Η μέθοδος αυτή αναλύεται παρακάτω:

```
$scope.loadMoreData = function () {
  $scope.moreDataCanBeLoaded = false;
  GetProvidersService.getProviders($scope.start)
    .then(function ($success) {
      $scope.start += $success.data.data.length;
      $scope.providers.push($success.data.data);
      $scope.moreDataCanBeLoaded = true;
    },
    function () {
      $scope.moreDataCanBeLoaded = false;

      // Reset value after one minute.
      $interval(function () {
        $scope.moreDataCanBeLoaded = true;
      }, 1000 * 60);
    });

  $scope.$broadcast('scroll.infiniteScrollComplete');
};
```

Snippet 1: Infinite scroller

1. Αλλαγή ενός flag που υποδεικνύει αν μπορούν να φορτωθούν κι άλλα δεδομένα σε false.
2. Με τη βοήθεια του GetProvidersService λήψη περισσότερων παραγωγών.
3. Σε περίπτωση που το αίτημα επιστρέψει επιτυχές μήνυμα, αύξηση της μεταβλητής start, η οποία υποδεικνύει από πού θα ξεκινούν οι λήψεις των επόμενων παραγωγών, κατά τον αριθμό αυτών που λήφθηκαν και επαναφορά του flag σε true.

4. Σε περίπτωση που το αίτημα αποτύχει, απενεργοποίηση των προσπαθειών για 1 λεπτό.
5. Τέλος, δήλωση του τερματισμού της φόρτωση δεδομένων από τον infinite scroller.

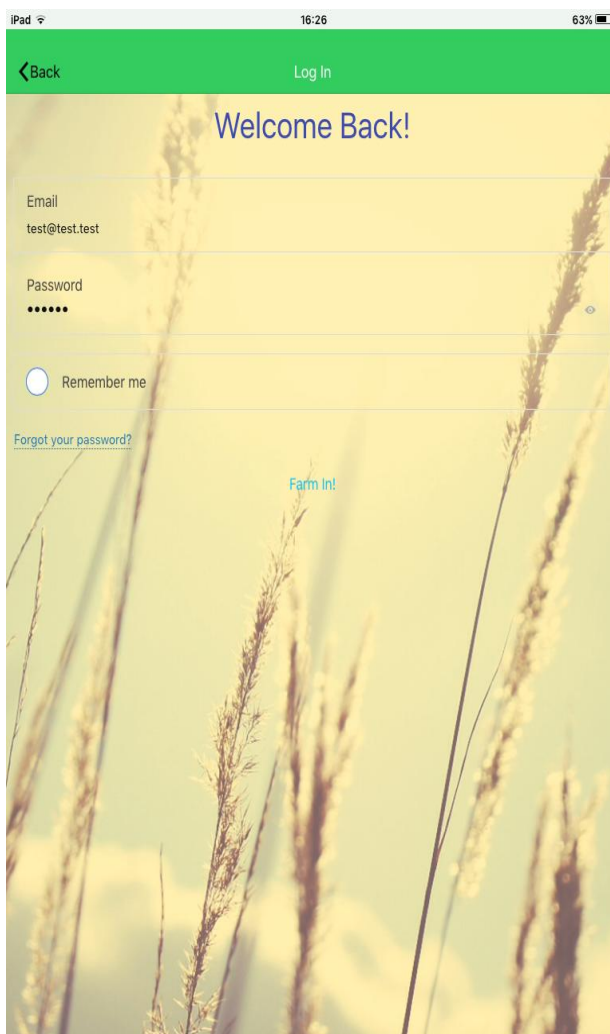
## 6.3. Workaround

Σε ορισμένες συσκευές με ios λειτουργικό, παρατήρησα ότι το περιεχόμενο κρυβόταν από την μπάρα. Επομένως, προκειμένου να παρακάμψω αυτό το πρόβλημα, χρησιμοποίησα τον παρακάτω css κώδικα:

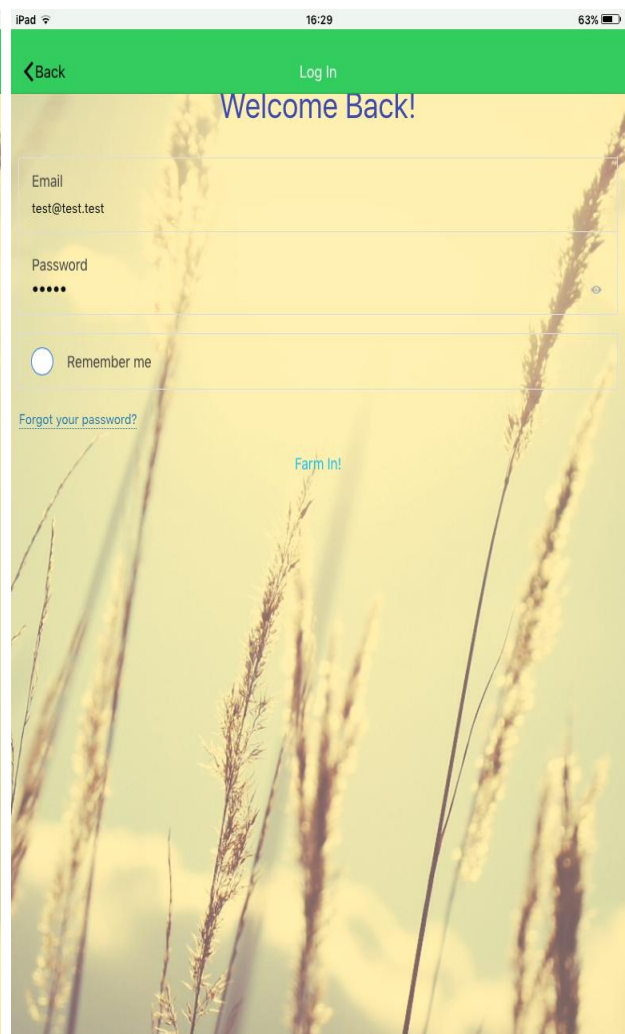
```
.platform-ios.platform-cordova ion-nav-view {  
  top: 2%;  
}
```

Snippet 2: Ios Workaround

Το αποτέλεσμα φαίνεται στις παρακάτω εικόνες:



Εικόνα 5: Με εφαρμογή του workaround

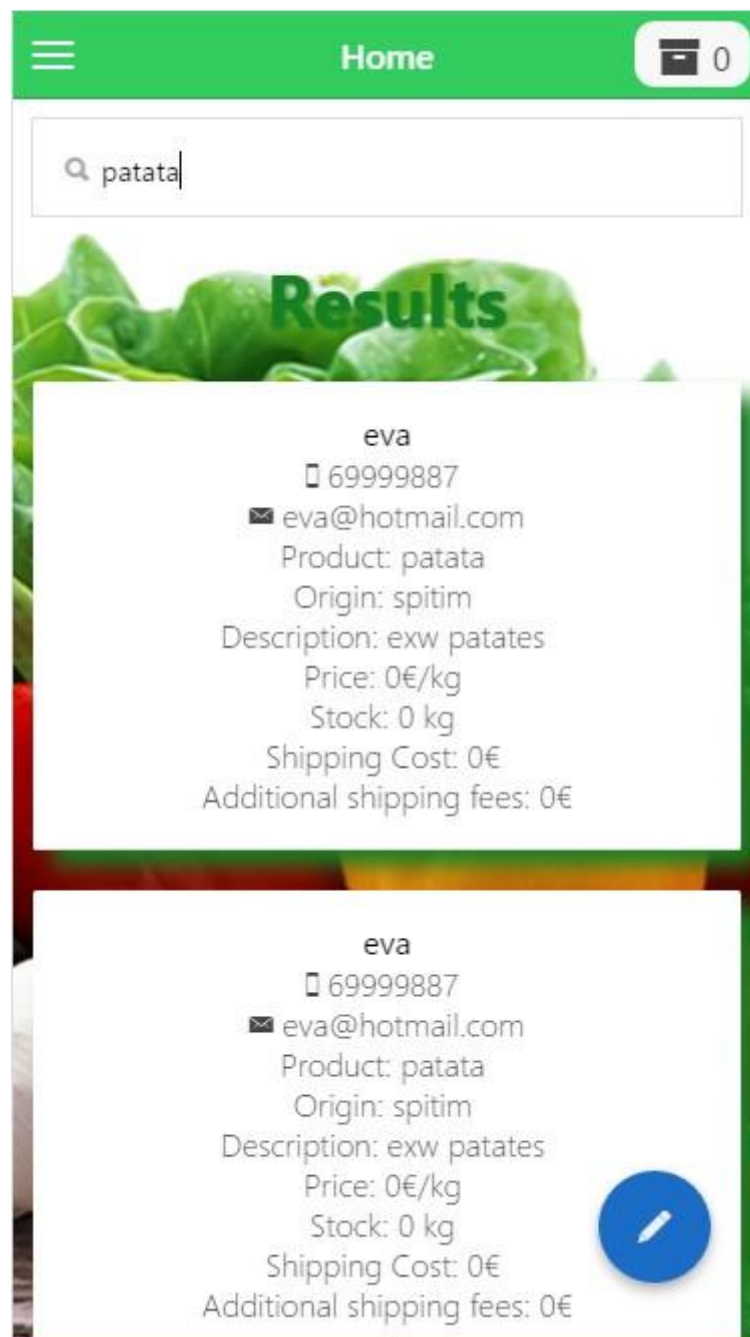


Εικόνα 6: Χωρίς εφαρμογή του workaround

## Κεφ.7: Παραδείγματα Χρήσης

### 7.1. Αναζήτηση

Παρακάτω παρουσιάζεται η περίπτωση που κάποιος πελάτης επιθυμεί να αναζητήσει πατάτες. Εμφανίζονται μόνο οι παραγωγοί που διαθέτουν αυτό το προϊόν, μαζί με τα στοιχεία επικοινωνίας τους και πληροφορίες όπως η διαθεσιμότητα του προϊόντος, η τιμή του κτλ.



Εικόνα 7: Παράδειγμα αναζήτησης

## 7.2. Αγγελία

Παρακάτω παρουσιάζεται μία ενδεικτική αγγελία ζήτησης προϊόντων. Εμφανίζονται η κατηγορία, η υποκατηγορία και η ποσότητα από τα προϊόντα που έχει εισάγει ο πελάτης στη λίστα του. Επιπλέον υπάρχουν δύο κουμπιά κάτω από κάθε προϊόν, ένα για την αλλαγή της ποσότητάς του και ένα για τη διαγραφή του από τη λίστα. Τέλος, υπάρχουν επιλογές για την υψηλότερη τιμή προσφοράς που θέλει να λάβει ο πελάτης και τη διάρκεια για την οποία επιθυμεί να παραμείνει ενεργή η αγγελία του. Αφού συμπληρωθούν όλα, μπορεί να επιλέξει το κουμπί για τη δημοσίευσή της.

Your AnnouncementAdd More

Categoryfruits

Subcategoryaktinidio

Quantity (kg)2.5

Edit QuantityRemove Product

Categorysuperfoods

Subcategorygoji berry

Quantity (kg)2

Edit QuantityRemove Product

Highest Price € (optional)4

Duration (days)7

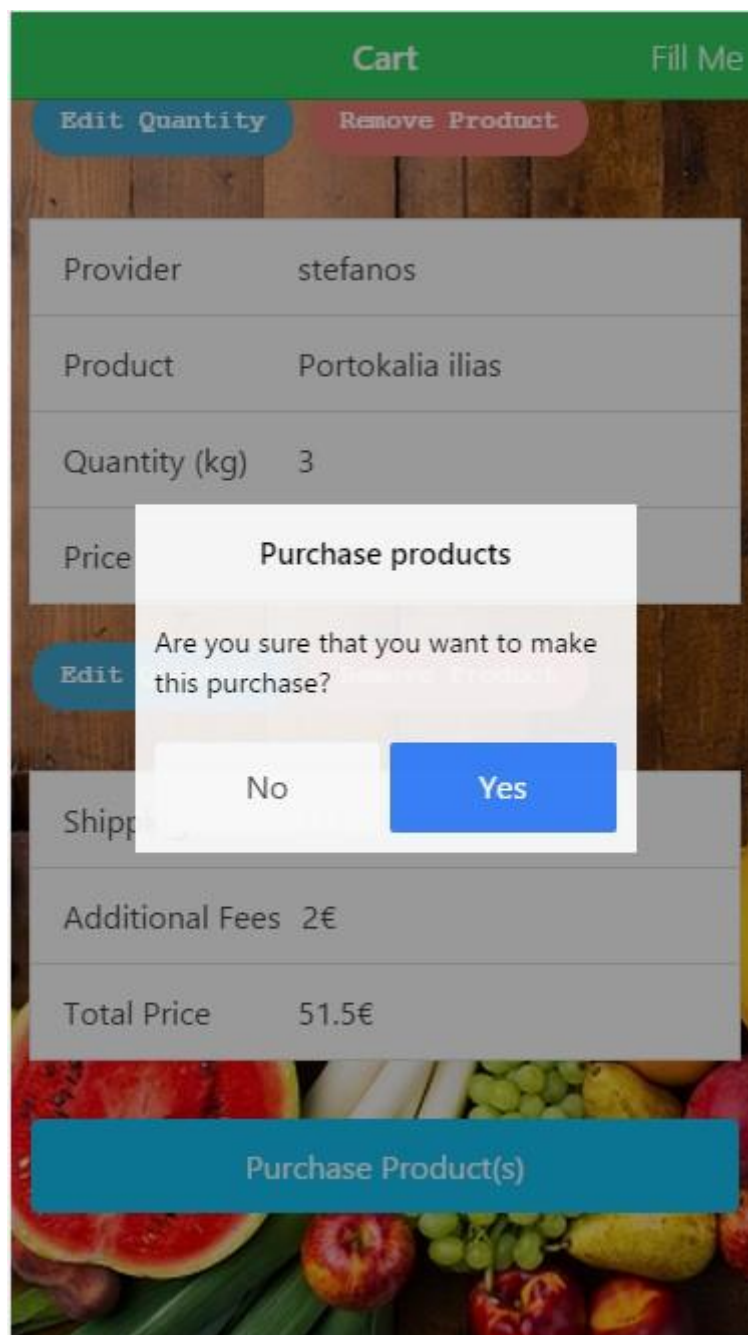
Upload Your Announcement

Εικόνα 8: Παράδειγμα αγγελίας



### 7.3. Άμεση Αγορά

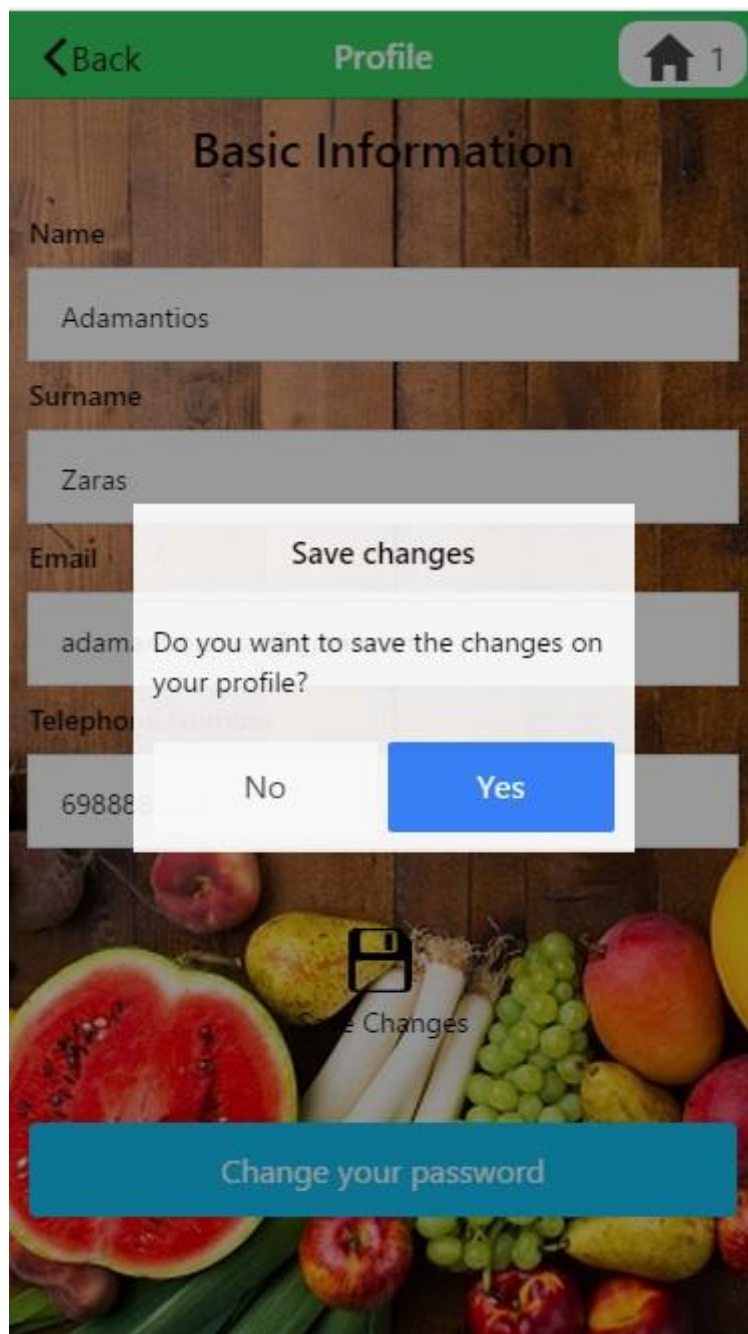
Παρακάτω, παρουσιάζεται το τελικό στάδιο στη διαδικασία μίας άμεσης αγοράς. Ζητείται από τον πελάτη να επιβεβαιώσει την αγορά του για λόγους ασφάλειας. Οι άμεσες αγορές πραγματοποιούνται από το καλάθι αγορών, όπου υπάρχουν όλα τα προϊόντα που έχει προσθέσει ο πελάτης, καθώς και κουμπιά για την επεξεργασία της ποσότητάς τους και τη διαγραφή τους. Στο τέλος αναφέρονται οι πρόσθετες επιβαρύνσεις, όπως τα μεταφορικά και τα συνολικό κόστος. Αφού συμπληρωθούν όλα, μπορεί να επιλέξει το κουμπί για την πραγματοποίησή της.



Εικόνα 9: Παράδειγμα άμεσης αγοράς

## 7.4. Αλλαγή των στοιχείων προφίλ

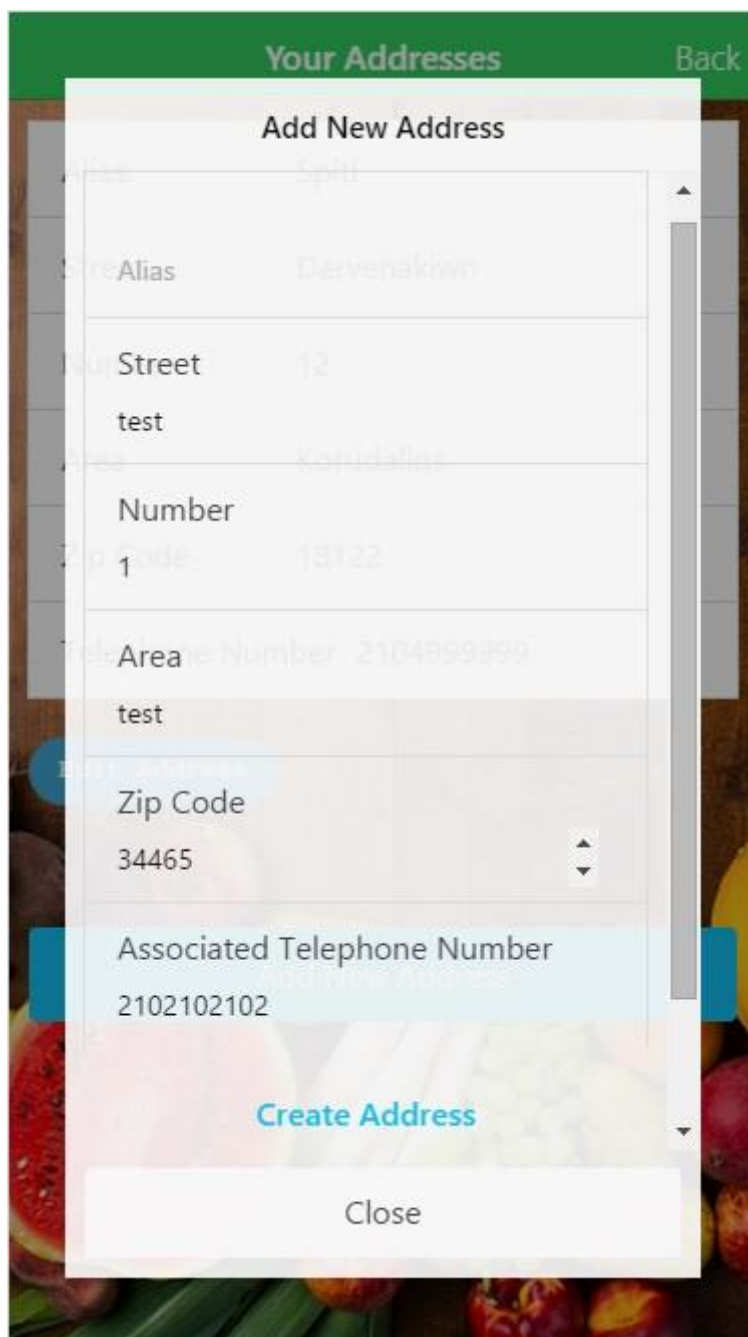
Παρακάτω παρουσιάζεται η περίπτωση της πραγματοποίησης αλλαγών στο προφίλ ενός πελάτη. Ζητείται να επιβεβαιώσει τις αλλαγές του για λόγους ασφάλειας. Όταν κάποιο πεδίο διαφοροποιείται, εμφανίζεται αυτόματα εικονίδιο, το οποίο ο χρήστης μπορεί να επιλέξει για να αποθηκεύσει τις νέες εισαγωγές. Επιπλέον υπάρχει κουμπί για την αλλαγή του κωδικού πρόσβασής του.



Εικόνα 10: Παράδειγμα αλλαγής στοιχείων του προφίλ

## 7.5. Προσθήκη νέας διεύθυνσης

Παρακάτω παρουσιάζεται η περίπτωση προσθήκης νέας διεύθυνσης στο λογαριασμό ενός πελάτη. Εμφανίζεται μία φόρμα, που περιλαμβάνει όλα τα απαραίτητα πεδία και αφού συμπληρωθούν, μπορεί να επιλεγεί η δημιουργία της.



The screenshot shows a mobile application interface with a green header bar containing the text "Your Addresses" and a "Back" button. A white modal dialog box titled "Add New Address" is centered on the screen. The dialog contains several input fields: "Alias" (with the placeholder "Dervenakiwn"), "Street" (with the placeholder "12 test"), "Number" (with the placeholder "1"), "Area" (with the placeholder "test"), "Zip Code" (with the placeholder "34465"), and "Associated Telephone Number" (with the placeholder "2102102102"). Below the input fields are two buttons: "Create Address" in blue and "Close" in white. The background of the application is a blurred image of various fruits.

Εικόνα 11: Παράδειγμα προσθήκης νέας διεύθυνσης

## 7.6. Στατιστικά χρήσης

Παρακάτω παρουσιάζονται τα στατιστικά χρήσης ενός πελάτη. Παρέχονται στοιχεία για τις αγγελίες που έχει δημιουργήσει, τις προσφορές που έχει αποδεχτεί ως προς τις συνολικές προσφορές που έχει λάβει από τους παραγωγούς και τις άμεσες αγορές που έχει πραγματοποιήσει



Εικόνα 12: Παράδειγμα προβολής στατιστικών χρήσης

## Κεφ.8: Η ολοκλήρωση του FarmIt



Εικόνα 13: Η τελική μορφή του FarmIt

Το FarmIt, είναι ένα έργο ΕΛ/ΛΑΚ που ξεκίνησαν οι *Χρήστος Σαρδιανός* και *Γιάννης Κατάκης*. Αποτελείται από δύο τμήματα, ένα για τους παραγωγούς και ένα για τους πελάτες. Για κάθε τμήμα, υπάρχει και η αντίστοιχη εφαρμογή. Το backend και η βάση δεδομένων είναι κοινά, όμως υπάρχουν χαρακτηριστικά που αφορούν μόνο τους παραγωγούς ή μόνο τους πελάτες. Στο τμήμα που αφορά τους παραγωγούς έχει συνεισφέρει ο *Στέφανος Φαφαλιός* και υλοποίησε λειτουργίες που παρέχουν δυνατότητες όπως η πραγματοποίηση δημοπρασιών μεταξύ των παραγωγών, η ανάρτηση προϊόντων, η αναζήτηση αγγελιών κτλ.

Στην πτυχιακή μου εργασία ασχολήθηκα με το τμήμα που αφορά τους πελάτες και υλοποίησα λειτουργίες που παρέχουν δυνατότητες όπως η δημιουργία αγγελιών, η αναζήτηση παραγωγών και προϊόντων, η αποδοχή ή απόρριψη προσφορών κτλ. Χρειάστηκαν μετατροπές στη βάση δεδομένων και το backend, καθώς και η προσθήκη συναρτήσεων στο δεύτερο, για τη διαχείριση των αιτημάτων που αφορούν τους πελάτες. Επιπλέον ανέπτυξα μία υβριδική εφαρμογή η οποία αποτελεί εξολοκλήρου το frontend, μαζί με αυτή των παραγωγών.

Η τελική μορφή φαίνεται στην παραπάνω εικόνα. Οι παραγωγοί και οι πελάτες, με τη χρήση της αντίστοιχης εφαρμογής του κινητού τους τηλεφώνου και του διαδικτύου, επικοινωνούν με ασφάλεια, διαθέτοντας ένα μοναδικό κλειδί, με το backend, το οποίο συνεργάζεται με τη βάση δεδομένων και στην πραγματικότητα μοιάζει σαν να συνδέονται άμεσα μεταξύ τους.

Έτσι ολοκληρώθηκε το FarmIt σαν έργο και είναι πλέον διαθέσιμο για ανεξάρτητους παραγωγούς, λιανέμπορους, συνεταιρισμούς, σωματεία και πρωτίστως μικρομεσαίες αγροτικές επιχειρήσεις, οι οποίες δεν έχουν τη δυνατότητα να επενδύσουν χρήματα για τη διάθεση των προϊόντων τους ηλεκτρονικά αλλά επιθυμούν να πραγματοποιήσουν μόνοι τους τη διαχείριση και την προώθηση τους. Η επιχειρησιακή στρατηγική των εφαρμογών βασίζεται στην ενεργοποίηση εμπόρων και πελατών με απώτερο σκοπό την επίτευξη της καλύτερης δυνατής τιμής αγοράς για τους τελικούς καταναλωτές καθώς και την διευκόλυνση της παροχής των προϊόντων από τους παραγωγούς, ειδικά για εκείνους που οι τρόποι παροχής των προϊόντων τους είναι περιορισμένοι.

Τέλος, το FarmIt είναι ΕΛ/ΛΑΚ και διαθέσιμο για κάθε χρήση, προκειμένου να επεκτείνεται συνεχώς και να ερευνάται για πιθανά σφάλματα ή back doors που μπορεί να το καθιστούν ευάλωτο απέναντι σε κακόβουλες επιθέσεις. Είναι δυνατή η παραμετροποίησή του ώστε να προσαρμοστεί και να επεκταθεί, προκειμένου να εξυπηρετήσει τους σκοπούς οποιουδήποτε χρήστη. Μπορεί να αποτελέσει εκπαιδευτικό εργαλείο.

## Κεφ.9: Μελλοντικές επεκτάσεις

Το FarmIt, ως λογισμικό ανοιχτού κώδικα, έχει πολλές ευκαιρίες ανάπτυξης, εξέλιξης και προσθήκης επιπλέον χρήσιμων χαρακτηριστικών. Ορισμένες ιδέες που θα μπορούσαν να αποτελέσουν τη μελλοντική επέκτασή του, είναι οι παρακάτω:

1. Εφαρμογή συστήματος επιβράβευσης με την εξαργύρωση πόντων. Οι πελάτες θα λαμβάνουν έναν αριθμό πόντων, ανάλογα με το κόστος των αγορών που πραγματοποιούν. Οι πόντοι αυτοί θα μπορούν να εξαργυρωθούν για αγορές, με τη μορφή έκπτωσης, σε συγκεκριμένους αγρότες που συνεργάζονται με το σύστημα. Αυτό το χαρακτηριστικό, θα ενθαρρύνει πολλούς ανθρώπους να χρησιμοποιήσουν το FarmIt για τις αγορές τους και επιπλέον θα τους δώσει τη δυνατότητα να αγοράσουν περισσότερα προϊόντα. Τέλος, θα δώσει την ευκαιρία σε κάποιους παραγωγούς να αυξήσουν τις πωλήσεις τους, πραγματοποιώντας δελεαστικές εκπτώσεις και παράλληλα αυξάνοντας τις αγορές, ώστε να μην έχουν ζημία.
2. Αναζήτηση παραγωγών με βάση την αξιολόγησή τους, τη δημοτικότητά τους και άλλα χαρακτηριστικά, όπως για παράδειγμα αν διαθέτει προϊόντα βιολογικής καλλιέργειας.
3. Δυνατότητα αναφοράς κάποιου χρήστη για κακή χρήση της εφαρμογής. Για παράδειγμα κάποιος παραγωγός μπορεί να ανεβάζει παραπλανητικές τιμές στο FarmIt ή άσχετα προϊόντα. Οι πελάτες θα μπορούν να το αναφέρουν, ώστε να εξεταστεί από τους διαχειριστές.
4. Προσθήκη περισσότερων τρόπων πληρωμής, όπως για παράδειγμα με τη χρήση πιστωτικών καρτών ή Payral. Προς το παρόν όλα τα προϊόντα πληρώνονται με τη μέθοδο της αντικαταβολής.
5. Οργάνωση δρώμενων, ανά τακτά χρονικά διαστήματα, στα οποία θα μπορούν να παρευρεθούν παραγωγοί, ώστε να παρουσιάσουν και να πουλήσουν τα προϊόντα τους στους πελάτες. Έτσι μπορεί να δημιουργηθεί καλύτερη σχέση εμπιστοσύνης μεταξύ των παραγωγών και των πελατών του FarmIt.
6. Προσθήκη αγγελιών ζήτησης και εύρεσης εργασίας που αφορά τον αγροτικό τομέα.
7. Συνεχής παρακολούθηση και εξέλιξη των μέτρων ασφαλείας, ώστε οι χρήστες να μην κινδυνεύουν από υποκλοπή των προσωπικών τους δεδομένων.



## Κεφ.10: Επίλογος

Σε αυτή την πτυχιακή, αναπτύχθηκε μία εφαρμογή ΕΛ/ΛΑΚ για πελάτες, με στόχο την αγορά αγροτικών προϊόντων απευθείας από παραγωγούς, η οποία συνεργάζεται σαν μία οντότητα με μία ήδη υπάρχουσα για τους τελευταίους και μαζί αποτελούν το FarmIt. Βασικός στόχος ήταν η δυνατότητα δημιουργίας αγγελιών από τους χρήστες, προκειμένου να πραγματοποιούνται δημοπρασίες μεταξύ των παραγωγών και να λαμβάνονται τα αιτούμενα προϊόντα στη χαμηλότερη δυνατή τιμή.

Αρχικά, μελετήθηκαν οι απαιτήσεις της εφαρμογής, ώστε να επιλεχθούν σωστά τα εργαλεία και τα frameworks που ταίριαζαν περισσότερο. Αφού επιλέχθηκαν με ιδιαίτερη προσοχή, σχεδιάστηκε η βάση δεδομένων με τρόπο που θα εξασφάλιζε τη σωστή λειτουργία. Στη συνέχεια, αναπτύχθηκε ένα RESTful Web Service, το οποίο επικοινωνεί με τη βάση δεδομένων και παρέχει απαντήσεις σε συγκεκριμένα αιτήματα. Με τη βοήθειά του, αναπτύχθηκε μία υβριδική εφαρμογή κινητών τηλεφώνων, φιλική προς του χρήστες που θα αποτελέσουν τους πελάτες του FarmIt, της οποίας το περιεχόμενο εξαρτάται σε μεγάλο βαθμό από τις απαντήσεις που λαμβάνει. Τέλος, μελετήθηκαν τα αποτελέσματα της συνεργασίας των παραπάνω, καθώς και η απόδοσή τους και πραγματοποιήθηκαν βελτιώσεις όπου ήταν απαραίτητο.

Η χρήση των framework επιτάχυνε σημαντικά τη διαδικασία ανάπτυξης και η πρώτη έκδοση του FarmIt ήταν έτοιμη σε σύντομο χρονικό διάστημα. Επιπλέον προσφέρουν ασφάλεια και υποστηρίζονται από μεγάλες, ανοιχτές κοινότητες. Χωρίς τη χρήση τους, η δημιουργία διαφορετικών εφαρμογών για τα διαφορετικά λειτουργικά συστήματα των φορητών συσκευών και ο προγραμματισμός διαδικασιών που ήδη παρέχονται από τις βιβλιοθήκες τους, θα ήταν απαραίτητα.



## Βιβλιογραφία

1. Ben Collins-Sussman, B. W. (2011). *Version Control with Subversion For Subversion 1.7 (Compiled from r5267)*. California, USA.
2. *blog.expressionsoftware.com*. (2011, Φεβρουάριος 27). Ανάκτηση Ιανουάριος 10, 2017, από <http://blog.expressionsoftware.com/2011/02/https-sequence-diagram.html>
3. *CodeIgniter*. (n.d.). Ανάκτηση Ιανουάριος 10, 2017, από <https://codeigniter.com/>: <https://codeigniter.com/>
4. Geet, B. (2016, Μάρτιος 16). <http://falconhive.com/>. Ανάκτηση Ιανουάριος 10, 2017, από <http://falconhive.com/host-website-wamp-server/>
5. *Ionic*. (2013). Ανάκτηση Ιανουάριος 10, 2017, από <http://ionicframework.com/>: <http://ionicframework.com/>
6. Mario Korf, Eugene Oksman (2016, Ιούνιος). <https://developer.salesforce.com/>. Ανάκτηση Ιανουάριος 10, 2017, από [https://developer.salesforce.com/page/Native,\\_HTML5,\\_or\\_Hybrid:\\_Understanding\\_Your\\_Mobile\\_Application\\_Development\\_Options](https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options)
7. Matheus, Abner (2015, Οκτώβριος 13). <http://picoledelimao.github.io/>. Ανάκτηση Ιανουάριος 10, 2017, από <http://picoledelimao.github.io/blog/2015/10/13/building-a-restful-web-service-in-c-plus-plus/>
8. The Linux Information Project. (2006, Ιούνιος 22). *Database Definition*. Ανάκτηση Ιανουάριος 10, 2017, από <http://linfo.org/database.html>