



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη συστήματος ηλεκτρονικών αγορών με υποστήριξη παρακολούθησης τοποθεσίας κατά την παραλαβή των προϊόντων Development of an online shopping system that supports location tracking during the product delivery
Όνοματεπώνυμο Φοιτητή	Παπαδάκης Λουκάς
Πατρώνυμο	Μιχαήλ
Αριθμός Μητρώου	ΜΠΠΛ/ 13062
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Περίληψη

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός συστήματος ηλεκτρονικών αγορών το οποίο θα εκμεταλλεύεται τις δυνατότητες των σύγχρονων φορητών συσκευών (κινητά τηλέφωνα, ταμπλέτες) και ιδιαίτερα αυτή της εύρεσης τοποθεσίας, προκειμένου να παρέχει καλύτερες υπηρεσίες προς τους πελάτες αλλά και σημαντικά οφέλη για τα καταστήματα κατά την διαδικασία παράδοσης των προϊόντων.

Δημιουργήθηκε εφαρμογή για συσκευές με λειτουργικό σύστημα Android, με την οποία οι χρήστες μπορούν να πραγματοποιούν αγορές από το σύστημα και μέσω της οποίας γίνεται παρακολούθηση της τοποθεσίας του πελάτη κατά την παραλαβή των προϊόντων.

Κατασκευάστηκε ακόμα ένας ιστότοπος μέσω του οποίου τα καταστήματα έχουν τη δυνατότητα να διαχειρίζονται τα προϊόντα που προσφέρουν προς πώληση, καθώς και να παρακολουθούν σε πραγματικό χρόνο τις παραλαβές προϊόντων που βρίσκονται σε εξέλιξη από πελάτες.

Abstract

The purpose of this thesis is the development of an online shopping system that will take advantage of the abilities of the modern portable devices (mobile phones, tablets) and especially location tracking in order to offer better services to the customers and benefits for the shops during the delivery of the bought products.

An android application was developed, which let customer buy products from the system and through which customer's location is tracked live throughout the pickup of bought products.

A website was also developed that lets shops manage the products they sell and monitor in real time product pickups from their customers.

Περιεχόμενα

1. Βάση Δεδομένων	5
1.1 Τεχνικά Χαρακτηριστικά	5
1.2 Δομή Βάσης Δεδομένων	5
2. Web Services	9
2.1 Τεχνικά Χαρακτηριστικά	9
2.2 Δομή του Solution	9
2.3 Διαθέσιμα web-services και Λειτουργίες	10
3. Εφαρμογή Android	14
3.1 Τεχνικά Χαρακτηριστικά	14
3.2 Βασικά Στοιχεία Εφαρμογής	14
3.3 Οθόνη Παρουσίασης Προϊόντων	15
3.4 Καλάθι Αγορών - Οθόνη ολοκλήρωσης Αγοράς	18
3.5 Οθόνη Σύνδεσης - Εγγραφής Χρήστη	22
3.6 Οθόνη Στοιχείων Χρήστη	24
3.7 Οθόνη Έναρξης Παραλαβής Προϊόντων	25
3.8 Οθόνη Παρακολούθησης Παραλαβής Προϊόντων	28
3.9 Παρακολούθηση Τοποθεσίας	30
3.10 Web Requests	31
4. Ιστότοπος Συστήματος	32
4.1 Τεχνικά Χαρακτηριστικά	32
4.2 Δομή του Solution	33
4.3 Οθόνη Σύνδεσης Καταστήματος	34
4.4 Κεντρική Σελίδα Καταστήματος	34
4.5 Σελίδα Διαθέσιμων Προϊόντων	35
4.6 Σελίδα Παρακολούθησης Παραλαβών	38
5. Συμπεράσματα	41

Ευρετήριο Εικόνων

Εικόνα 1: Δομή της βάσης δεδομένων	8
εικόνα 2 : Δομή του solution	10
εικόνα 3: Συρόμενο μενού πλοήγησης, γραμμή ενεργειών εφαρμογής.....	15
εικόνα 4: dialog ενημέρωσης χρήστη για σύνδεση στο δίκτυο	16
εικόνα 5: Αντικείμενο λίστας διαθέσιμων προϊόντων	16
εικόνα 6: Αναζήτηση στη λίστα διαθέσιμων προϊόντων	17
εικόνα 7: Αντικείμενο λίστας καταστημάτων	17
εικόνα 8: Λίστα καταστημάτων που προσφέρουν ένα προϊόν	18
εικόνα 9: Καλάθι αγορών χρήστη	19
εικόνα 10: Μήνυμα ενημέρωσης καλαθιού χωρίς προϊόντα	20
εικόνα 11: Αντικείμενο λίστας προϊόντων στο καλάθι χρήστη	20
εικόνα 12: Μήνυμα σφάλματος κατά την προσπάθεια ολοκλήρωσης παραγγελίας	21
εικόνα 13: dialog ολοκλήρωσης παραγγελίας	22
εικόνα 14: Οθόνη σύνδεσης χρήστη	23
εικόνα 15: Οθόνη εγγραφής χρήστη	23
εικόνα 16: Οθόνη στοιχείων χρήστη	24
εικόνα 17: dialog αλλαγής κωδικού πρόσβασης	25
εικόνα 18: Κενή λίστα προϊόντων προς παραλαβή	26
εικόνα 19: Αντικείμενο λίστας προϊόντων στο καλάθι χρήστη	27
εικόνα 20: Εμφάνιση - απόκρυψη κουμπιού εκκίνησης παραλαβής	27
εικόνα 21: Προτροπή χρήστη για αλλαγή ρυθμίσεων τοποθεσίας	28
εικόνα 22: Τοποθέτηση marker στη θέση του καταστήματος παραλαβής	29
εικόνα 23: dialog επιβεβαίωσης ολοκλήρωσης παραλαβής	30
εικόνα 24: dialog αιτήματος για άδεια παρακολούθησης τοποθεσίας κατά	31
εικόνα 25: Δομή του Solution ιστότοπου	33
εικόνα 26: Οθόνη σύνδεσης καταστήματος	34
εικόνα 27: Κεντρική σελίδα καταστήματος	35
εικόνα 28: Πίνακας παρουσίασης διαθέσιμων προϊόντων καταστήματος	36
εικόνα 29: Παράθυρο επεξεργασία στοιχείων προϊόντος	37
εικόνα 30: Παράθυρο επιλογής προϊόντος προς διάθεση	37
εικόνα 31: Παράθυρο εισαγωγής στοιχείων προϊόντος προς διάθεση	38
εικόνα 32: Οθόνη παρακολούθησης παραλαβών καταστήματος	39
εικόνα 33: Παράθυρο στοιχείων πελάτη παραλαβής	40

1. Βάση Δεδομένων

Χρησιμοποιείται μια κεντρική βάση δεδομένων η οποία περιέχει όλα τα στοιχεία σχετικά με τα καταστήματα που συμμετέχουν στο σύστημα, τα προϊόντα που προσφέρονται, τους πελάτες και τις αγορές.

Η ιστοσελίδα διαχείρισης επικοινωνεί απευθείας με τη βάση δεδομένων για την προβολή στοιχείων κάθε επιχείρησης, ενώ η επικοινωνία με την android εφαρμογή γίνεται μέσω web services.

1.1 Τεχνικά χαρακτηριστικά

Επιλέχτηκε η χρήση βάσης δεδομένων MSSQL και για την για τη διαχείριση της χρησιμοποιείται Microsoft Sql Server.

Η βάση δεδομένων φιλοξενείται στον ιστότοπο smarterasp.net, μαζί με τα web services μέσω των οποίων η εφαρμογή Android επικοινωνεί με την βάση δεδομένων).

1.2 Δομή Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από τους εξής πίνακες:

Customer : Περιέχει τους πελάτες του συστήματος, που έχουν κάνει εγγραφή μέσω της android εφαρμογής.

- Id (int): Πρωτεύον κλειδί εγγραφής
- Name (Nvarchar30): το όνομα του πελάτη
- Password (Nvarchar10): ο κωδικός του πελάτη διατηρείται ως string (στα πλαίσια αυτής της εργασίας δεν εφαρμόζεται κάποιο hashing για την κρυπτογράφηση/αποκρυπτογράφηση του κωδικού)
- Email (Nvarchar20) : το email του χρήστη που λειτουργεί και ως αναγνωριστικό
- Age (tinyInt) : ηλικία του χρήστη σαν αριθμός (δίνεται τιμή 0 αν ο χρήστης δεν έχει καταχωρήσει ηλικία)
- Address (int FK) : Ξένο κλειδί στον πίνακα Location. Κρατάει την τοποθεσία του χρήστη ως διεύθυνση του.

Shop : Περιέχει τα καταστήματα που συμμετέχουν στο σύστημα και μπορούν να προσφέρουν προϊόντα.

- Id (int): Πρωτεύον κλειδί εγγραφής
- Name (nvarchar30) : το όνομα του καταστήματος

- Location (int FK) : Ξένο κλειδί στον πίνακα Location. Κρατάει την τοποθεσία του καταστήματος

Product : Περιέχει τα προϊόντα του συστήματος που μπορούν να προσφέρονται από κάποιο κατάστημα που συμμετέχει στο σύστημα.

- Id (int): Πρωτεύον κλειδί εγγραφής
- Name (nvarchar 30) : Το όνομα του προϊόντος
- Description (nvarchar 50) : Περιγραφή του προϊόντος
- Category (int FK) : Ξένο κλειδί στον πίνακα Category. Αναπαριστά την κατηγορία του προϊόντος

Category : Περιέχει τις κατηγορίες στις οποίες μπορεί να ανήκει ένα προϊόν του συστήματος.

- Id (int) : Πρωτεύον κλειδί εγγραφής
- Name (nvarchar30) : Το όνομα της κατηγορίας

AvailableProducts : Ο πίνακας αυτός συσχετίζει τα προϊόντα και τα καταστήματα του συστήματος. Πιο συγκεκριμένα περιλαμβάνει τα προϊόντα του συστήματος που προσφέρονται από κάθε κατάστημα, σε ποια τιμή και ποσότητα.

- Id (int) : Πρωτεύον κλειδί εγγραφής
- Product (int FK) : Ξένο κλειδί στον πίνακα Product. Αναπαριστά το προϊόν που προσφέρεται.
- Shop (int FK) : Ξένο κλειδί στον πίνακα Shop. Αναπαριστά το κατάστημα που προσφέρει το συγκεκριμένο προϊόν.
- Ammount (smallint) : Αριθμός τεμαχίων του προϊόντος που είναι διαθέσιμα προς πώληση από το συγκεκριμένο κατάστημα.
- Price (decimal19,4) : Τιμή στην οποία προσφέρεται το προϊόν
- Discount (int): ποσοστό έκπτωσης στην τιμή που προσφέρεται το προϊόν

BoughtProducts : Πρόκειται για τον αντίστοιχο πίνακα με αυτόν των AvailableProducts που περιέχει αρχείο με όλα τα προϊόντα που έχουν αγοραστεί από κάποιον πελάτη.

- Id (int) : Πρωτεύον κλειδί εγγραφής
- Customer (int FK) : Ξένο κλειδί στον πίνακα Customer. Ο πελάτης που έκανε τη συγκεκριμένη αγορά.
- Product (int FK) : Ξένο κλειδί στον πίνακα Product. Το προϊόν που αγοράστηκε.
- Shop (int FK): Ξένο κλειδί στον πίνακα Shop. Το κατάστημα από το οποίο εκτελέστηκε η συγκεκριμένη αγορά.
- Ammount (smallint) : Αριθμός τεμαχίων που αγοράστηκαν
- Price (decimal 18,4) : Η τιμή μονάδας του προϊόντος όταν έγινε η αγορά
- Discount (int) : Το ποσοστό έκπτωσης στην τιμή όταν έγινε η αγορά

- BoughtOn (datetime) : Η ημερομηνία που έγινε η αγορά
- DeliveredOn (datetime) : Η ημερομηνία που έγινε η παραλαβή του προϊόντος από το κατάστημα. Το πεδίο αυτό δέχεται κενές τιμές (null) και χρησιμοποιείται για την αναγνώριση των προϊόντων που μπορεί να παραλάβει ο πελάτης
- CustomerLocation (int FK) : Ξένο κλειδί στον πίνακα Location. Η διεύθυνση του χρήστη όταν πραγματοποιήθηκε η αγορά

Location : Χρησιμοποιείται ως βοηθητικός πίνακας για την καταχώρηση τοποθεσίας από διάφορους πίνακες.

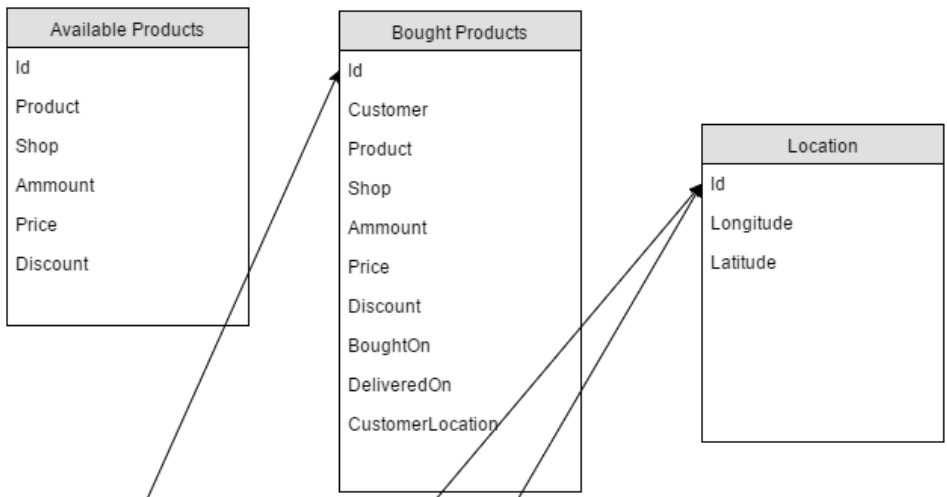
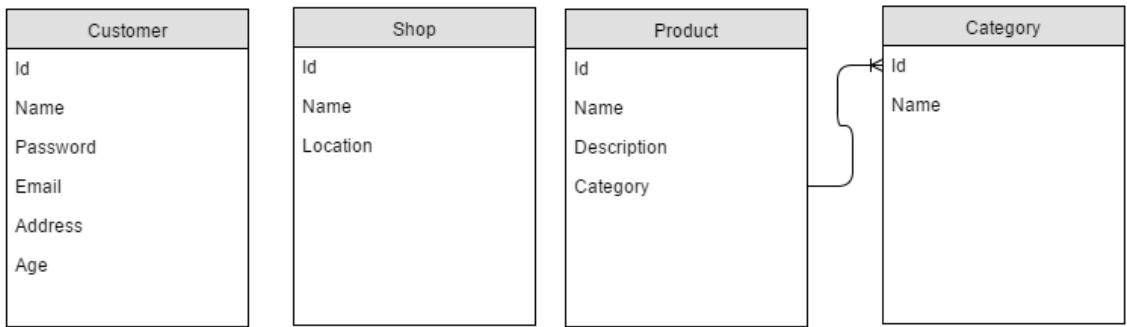
- Id (int) : Πρωτεύον κλειδί εγγραφής
- Longitude (nvarchar) : Γεωγραφικό μήκος της τοποθεσίας
- Latitude (nvarchar) : Γεωγραφικό πλάτος της τοποθεσίας

Pickup : Περιέχει τις ενεργές διαδικασίες παραλαβής αγορασμένων προϊόντων από πελάτες.

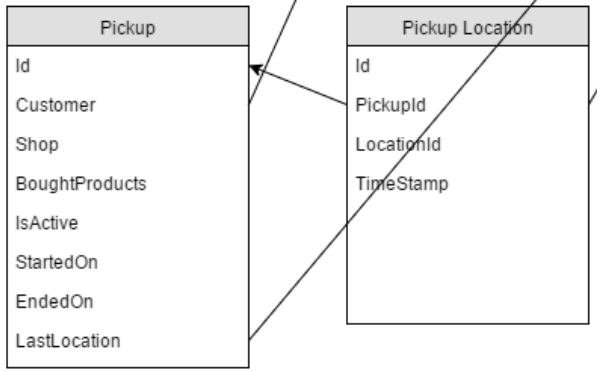
- Id (int) : Πρωτεύον κλειδί εγγραφής
- Customer (int FK) : ξένο κλειδί στον πίνακα Customer. Ο πελάτης που συμμετέχει στην παραλαβή προϊόντων
- Shop (int FK) : ξένο κλειδί στον πίνακα Shop. Το κατάστημα από το οποίο θα γίνει η παραλαβή προϊόντων
- BoughtProduct (nvarchar FK) : Λίστα με τα Id των αγορασμένων προϊόντων του πελάτη που θα παραληφθούν. Αποθηκεύονται σαν λεκτικού χωρίζοντας τα Id με κόμμα.
- IsActive (bool): ένδειξη για το αν η παραλαβή είναι ενεργή. Χρησιμοποιείται για γρηγορότερο φιλτράρισμα των παραλαβών κατά την ζωντανή παρακολούθηση
- StartedOn (datetime): χρόνος έναρξης της παραλαβής
- EndedOn (datetime): χρόνος ολοκλήρωσης παραλαβής
- LastLocation (int FK) : Ξένο κλειδί στον πίνακα Location. Η τελευταία τοποθεσία πελάτη που καταχωρήθηκε κατά την παραλαβή (χρησιμοποιείται κατά την ζωντανή παρακολούθηση παραλαβής για γρηγορότερη εύρεση της τοποθεσίας πελάτη).

PickupLocation : Περιέχει τις διαδοχικές τοποθεσίες στις οποίες βρέθηκε ο πελάτης κατά την παραλαβή.

- Id (int) : Πρωτεύον κλειδί εγγραφής
- PickupId (int FK): Ξένο κλειδί στον πίνακα Pickup. Η παραλαβή της οποίας τις τοποθεσίες αποθηκεύουμε.
- LocationId (int FK): Ξένο κλειδί στον πίνακα Location. Μια συγκεκριμένη τοποθεσία της παραλαβής.
- TimeStamp(datetime): χρόνος της συγκεκριμένης καταχώρησης



Location Tracking



εικόνα 1: Δομή της βάσης δεδομένων

2. Web Services

Για την επικοινωνία της Android εφαρμογής με τη βάση δεδομένων που φιλοξενείται στο cloud χρησιμοποιούνται Web Services.

Τα services φιλοξενούνται στο cloud και έχουν πρόσβαση στη βάση δεδομένων, παίρνουν δεδομένα από αυτήν και τα στέλνουν στην android εφαρμογή σε μορφή json. Επιπλέον καταχωρούν και νέα δεδομένα, όπως για παράδειγμα εγγραφή νέων χρηστών, αγορά προϊόντων από πελάτες κτλ.

2.1 Τεχνικά χαρακτηριστικά

Τα web services δημιουργήθηκαν σε περιβάλλον .net.

Χρησιμοποιήθηκε Visual Studio 2013[20] και φιλοξενήθηκαν στον ιστότοπο smarterasp.net ώστε να μπορούμε να έχουμε πρόσβαση σε αυτά από συσκευές Android με σύνδεση στο διαδίκτυο.

2.2 Δομή του Solution

Προκειμένου να επιτευχθεί συντηρησιμότητα το solution χωρίζεται σε 4 projects, ώστε κάνοντας αλλαγή σε κάποιο από αυτά να μην επηρεάζει όλο το solution.

Πιο συγκεκριμένα, το κύριο project, στο οποίο υλοποιούνται τα web services, είναι το ShopServices. Το project αυτό περιλαμβάνει το αρχείο IShopServices.cs, στο οποίο δηλώνονται τα web services τα οποία είναι διαθέσιμα, ενώ οι υλοποίηση των services σε κώδικα C# γίνεται στο αρχείο ShopServices.cs.

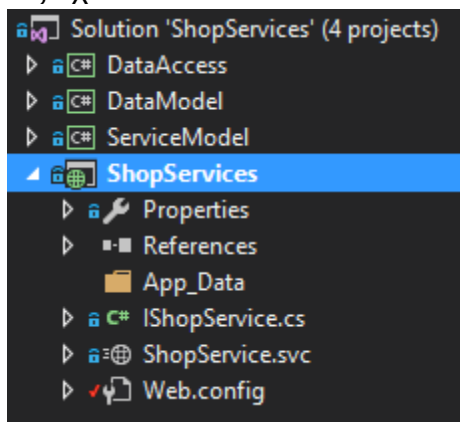
Το project ServiceModel περιέχει τις κλάσεις που επιστρέφουν τα web services (σε μορφή json). Απαιτείται επομένως το project ShopServices να έχει αναφορά στο project αυτό.

Η επικοινωνία με τη βάση δεδομένων γίνεται χρησιμοποιώντας το Entity Framework του περιβάλλοντος .NET. Το project DataModel περιέχει τις κλάσεις του Entity Framework που αντιστοιχούν στους πίνακες της βάσης δεδομένων.

Το project DataAccess έχει αναφορά στο DataModel και περιλαμβάνει μεθόδους για την επικοινωνία με τη βάση δεδομένων. Το project έχει επίσης αναφορά στο ServiceModel και χρησιμοποιεί έναν mapper για να μετατρέπει τα DataModel που παίρνει από τη βάση δεδομένων σε ServiceModels τα οποία και επιστρέφει στο βασικό project που περιλαμβάνει τα web services.

Με τη δομή αυτή επιτυγχάνουμε το βασικό μας project να είναι ανεξάρτητο από το είδος της βάσης δεδομένων που χρησιμοποιούμε, καθώς σε περίπτωση που απαιτηθεί

αλλαγή της βάσης δεδομένων θα γίνουν αλλαγές μόνο στα project DataModel και DataAccess ενώ το βασικό project που περιλαμβάνει τα web services θα παραμείνει ως έχει.



εικόνα 2 : Δομή του solution

2.3 Διαθέσιμα web-services και λειτουργίες τους

Τα services που δημιουργήθηκαν μπορούν να χωριστούν σε τέσσερις κατηγορίες με βάση τη λειτουργία που επιτελούν.

Αρχικά υπάρχουν τέσσερα services για τη διαχείριση πελατών, και τα οποία επιτελούν λειτουργίες δημιουργία - ανανέωσης (create - update) στον πίνακα των πελατών της βάσης δεδομένων, καθώς και ένα service για την σύνδεση του χρήστη (login). Πιο αναλυτικά, υπάρχουν τα εξής services:

- Η δημιουργία νέου χρήστη(πελάτη) υλοποιείται μέσω του service Register. Είναι μορφής POST και δέχεται σαν όρισμα ένα object Customer το οποίο περιλαμβάνει τα στοιχεία ενός πελάτη που θέλει να εγγραφεί. Αποστέλλονται λοιπόν όνομα, email το οποίο δρα και ως αναγνωριστικό του χρήστη, κωδικός ο οποίος χρησιμοποιείται κατά της σύνδεση του χρήστη, ηλικία και γεωγραφικό πλάτος και μήκος της συσκευής που καταχωρείται ως διεύθυνση χρήστη. Το service αρχικά ελέγχει αν υπάρχει καταχωρημένος χρήστης με την ίδια διεύθυνση ηλεκτρονικού ταχυδρομείου διαφορετικά αποθηκεύει τα στοιχεία του χρήστη στη βάση δεδομένων. Το service επιστρέφει μια μεταβλητή τύπου bool που σηματοδοτεί αν η καταχώρηση του πελάτη ήταν επιτυχής.
- Η σύνδεση χρήστη (πελάτη) υλοποιείται μέσω του service LoginCustomer. Είναι μορφής GET και δέχεται παραμέτρους το username και το password του χρήστη. Ως αναγνωριστικό χρησιμοποιείται η διεύθυνση email που χρησιμοποιήθηκε κατά τη δημιουργία του χρήστη. Το service ελέγχει αν στη βάση δεδομένων υπάρχει χρήστης με αυτά τα στοιχεία και επιστρέφει μια μεταβλητή τύπου Customer του project ServiceModels, το οποίο περιλαμβάνει

τα στοιχεία του χρήστη που καταχωρήθηκαν κατά την εγγραφή του χρήστη και επιπλέον το Id του χρήστη στη βάση δεδομένων, το οποίο χρησιμοποιείται ως παράμετρος σε άλλες web services που θα περιγραφούν παρακάτω. Σε περίπτωση που δεν βρεθεί στη βάση δεδομένων χρήστης με τον συγκεκριμένο συνδυασμό email - κωδικού το service επιστρέφει null.

- Το service UpdateCustomer υλοποιεί την αλλαγή στοιχείων ενός εγγεγραμμένου χρήστη (πελάτη). Είναι μορφής POST και δέχεται σαν παραμέτρους το Id του χρήστη του οποία τα στοιχεία θέλουμε να μεταβάλουμε, και τα νέα στοιχεία που θέλουμε να καταχωρηθούν στη βάση δεδομένων, όνομα, διεύθυνση ηλεκτρονικού ταχυδρομείου, ηλικία και γεωγραφικό πλάτος και μήκος της διεύθυνσης του. Το service επιστρέφει μια μεταβλητή μορφής bool που σηματοδοτεί την επιτυχή ανανέωση των στοιχείων του χρήστη.
- Τέλος, μέσω του service ChangePassword υλοποιείται η αλλαγή κωδικού πρόσβασης ενός χρήστη. Το service είναι μορφής POST και δέχεται ως παραμέτρους το Id του χρήστη, τον παλιό κωδικό πρόσβασης και έναν νέο κωδικό που θέλει να καταχωρήσει ο χρήστης. Το service σηματοδοτεί την επιτυχή αλλαγή του κωδικού πρόσβασης του χρήστη επιστρέφοντας μια μεταβλητή τύπου bool.

Η δεύτερη κατηγορία περιλαμβάνει services μέσω των οποίων μπορούμε να έχουμε πρόσβαση στα προϊόντα που είναι αποθηκευμένα στη βάση δεδομένων και είναι διαθέσιμα προς πώληση από τα καταστήματα, προκειμένου αυτά να παρουσιαστούν στους πελάτες. Τα services είναι τα εξής:

- Μέσω του service GetShops παρέχεται πρόσβαση στα καταστήματα που προσφέρουν προϊόντα και βρίσκονται στη βάση δεδομένων. Το service είναι μορφής GET και δεν δέχεται παραμέτρους. Επιστρέφει μια λίστα από αντικείμενα τύπου Shop του project ServiceModels, που περιέχουν στοιχεία για τα καταστήματα όπως το Id της εγγραφής στη βάση δεδομένων, το όνομα του καταστήματος και τη γεωγραφική του θέση (γεωγραφικό μήκος και πλάτος).
- Με το service GetCategories παρέχεται πρόσβαση στις διαθέσιμες κατηγορίες προϊόντων. Είναι μορφής GET και δεν δέχεται παραμέτρους ενώ επιστρέφει μια λίστα από αντικείμενα τύπου Category του project ServiceModels, που περιέχουν το Id της κατηγορίας στη βάση δεδομένων και το όνομα της.
- Το service GetProducts δίνει τη δυνατότητα πρόσβασης στη προϊόντα που είναι καταχωρημένα στη βάση δεδομένων και τα οποία μπορούν να παρέχονται από κάποιο κατάστημα. Το service είναι μορφής GET και δεν δέχεται παραμέτρους. Επιστρέφει μια λίστα από τύπου Product του project ServiceModels τα οποία περιέχουν πληροφορίες για τα προϊόντα, όπως το Id

κάθε εγγραφής στη βάση, το όνομα του προϊόντος, το Id της κατηγορίας στην οποία ανήκει και μια περιγραφή.

- Τέλος μέσω του service `GetAvailableProducts` παρέχεται πρόσβαση στον κατάλογο προϊόντων των καταστημάτων. Το service είναι μορφής GET, δεν δέχεται παραμέτρους και επιστρέφει μια λίστα από αντικείμενα τύπου `AvailableProduct` του project `ServiceModels`. Η κλάση `AvailableProduct` περιέχει στοιχεία για τη διαθεσιμότητα κάθε προϊόντος σε ένα συγκεκριμένο κατάστημα. Περιέχει το Id της εγγραφής στη βάση δεδομένων, τα Id του προϊόντος και του καταστήματος που προσφέρει το προϊόν αυτό, και στοιχεία για τη διαθεσιμότητα όπως τιμή, αριθμό τεμαχίων που προσφέρονται για αγορά καθώς και την έκπτωση η οποία προσφέρεται στο προϊόν.

Συνδυάζοντας τα service που παρουσιάστηκαν παραπάνω είναι δυνατόν να εμφανιστούν όλα τα στοιχεία των προϊόντων που προσφέρει κάθε κατάστημα. Στην τέταρτη κατηγορία ανήκουν τα services που σχετίζονται με την αγορά προϊόντων από εγγεγραμμένους πελάτες καθώς και την προβολή των προϊόντων που έχει αγοράσει κάθε χρήστης.

Τα services αυτά χρησιμοποιούν την κλάση `BoughtProduct` του project `ServiceModels`, η οποία περιλαμβάνει στοιχεία μιας αγοράς που πρόκειται ή έχει ήδη πραγματοποιηθεί. Η κλάση αυτή περιέχει το Id του πελάτη που κάνει την αγορά, το Id του προϊόντος το οποίο αγοράζεται και το κατάστημα το οποίο προσφέρει το προϊόν. Περιέχει ακόμα τον αριθμό των τεμαχίων που αγοράστηκαν, την τιμή και την έκπτωση που επικρατούσε κατά την αγορά. Τέλος, περιέχει τον χρόνο που πραγματοποιήθηκε η αγορά και σε περίπτωση που το προϊόν έχει παραληφθεί από τον πελάτη τον χρόνο που πραγματοποιήθηκε η παραλαβή.

- Η αγορά διαθέσιμων προϊόντων υλοποιείται μέσω του service `BuyProducts`. Είναι μορφής POST και δέχεται σαν παραμέτρους μια λίστα με αντικείμενα τύπου `BoughtProduct` τα οποία πρέπει να περιέχουν τα στοιχεία πελάτη, καταστήματος και προϊόντος που αφορά η αγορά, καθώς και την ποσότητα που επιθυμεί να αγοράσει ο πελάτης. Το service αρχικά επιβεβαιώνει την εγκυρότητα των στοιχείων ελέγχοντας την διαθεσιμότητα του προϊόντος στο κατάστημα καθώς και την ύπαρξη του πελάτη και στη συνέχεια εισάγει στη βάση δεδομένων την συγκεκριμένη αγορά προϊόντος, ενώ παράλληλα μειώνει τον αριθμό των διαθέσιμων τεμαχίων προϊόντος στο κατάστημα. Επιστρέφεται μια μεταβλητή τύπου `bool` η οποία σηματοδοτεί εάν η αγορά πραγματοποιήθηκε επιτυχώς και έχει εισαχθεί στη βάση δεδομένων.
- Προκειμένου να παρουσιαστούν τα προϊόντα που έχει αγοράσει κάποιος πελάτης χρησιμοποιείται το service `GetBoughtProducts`. Είναι μορφής POST και δέχεται σαν παράμετρο το Id του χρήστη του οποίου τις αγορές θέλουμε να μας επιστραφούν. Το service επιστρέφει μια λίστα με αντικείμενα τύπου `BoughtProduct` τα οποία έχουν τιμές σε όλα τα πεδία που περιγράψαμε παραπάνω.

Η τελευταία κατηγορία περιλαμβάνει τα services μέσω των οποίων πραγματοποιείται η διαδικασία παραλαβής αγορασμένων προϊόντων από κάποιο κατάστημα. Πρόκειται για τα εξής services:

- Το service StartPickup δίνει τη δυνατότητα εκκίνησης της διαδικασίας παραλαβής από κάποιον πελάτη. Είναι μορφής POST και δέχεται σαν παραμέτρους το Id του πελάτη που πρόκειται να παραλάβει τα προϊόντα, το Id του καταστήματος από το οποίο θα γίνει η παραλαβή καθώς και μια λίστα με τα Id των αγορών του πελάτη που πρόκειται να παραληφθούν (αναφέρεται στο BoughtProducts που αναλύθηκαν παραπάνω). Το service ελέγχει αν τα δεδομένα που έχουν αποσταλεί είναι έγκυρα και στη συνέχεια εισάγει μια νέα εγγραφή στον πίνακα παραλαβών στη βάση δεδομένων, επιστρέφεται τελικά μια μεταβλητή τύπου ακεραίου (int) που αντιστοιχεί στο Id της εγγραφής που εισήχθη.
- Χρησιμοποιώντας το Id της παραλαβής που επέστρεψε το service StartPickup μπορούμε να συγχρονίσουμε τη θέση του πελάτη με το service UpdatePickupLocation. Είναι μορφής POST και δέχεται σαν παραμέτρους το Id της παραλαβής και το γεωγραφικό μήκος και πλάτος της θέσης του πελάτη. Το service επιστρέφει μια λογική μεταβλητή (bool) η οποία σηματοδοτεί εάν η θέση αποθηκεύτηκε επιτυχώς.
- Η διαδικασία παραλαβής μπορεί να ακυρωθεί όσο είναι σε εξέλιξη χρησιμοποιώντας το service CancelPickup. Είναι μορφής POST και δέχεται σαν παράμετρο το Id της ενεργής παραλαβής. Το service επιστρέφει μια μεταβλητή τύπου bool προκειμένου να ενημερώσει για την επιτυχή διαγραφή της παραλαβής από τη βάση δεδομένων.
- Τέλος, όταν ο χρήστης έχει φτάσει στο κατάστημα ενημερώνεται ο server για την ολοκλήρωση της παραλαβής μέσω του service CompletePickup. Είναι μορφής POST και δέχεται σαν παράμετρο το Id της παραλαβής. Το service ανανεώνει τη βάση δεδομένων ώστε η παραλαβή να μην θεωρείται ενεργή (διατηρείται ωστόσο στη βάση ως ιστορικό) και αποθηκεύει το χρόνο παραλαβής των αγορών που σχετίζονταν με την παραλαβή. Επιστρέφει μια μεταβλητή τύπου bool που σηματοδοτεί την επιτυχή ανανέωση των δεδομένων στη βάση.

3. Εφαρμογή Android

Ο κύριος όγκος της εργασίας αφορά τη δημιουργία μιας εφαρμογής android μέσω της οποίας ο χρήστης μπορεί να δημιουργήσει λογαριασμό, να δει και να αγοράσει τα προϊόντα που προσφέρουν τα καταστήματα του συστήματος καθώς και να εκκινήσει τη διαδικασία παραλαβής κάποιων από αυτά που έχει αγοράσει. Οι λειτουργίες αυτές υλοποιούνται στην εφαρμογή χρησιμοποιώντας τα web services που περιγράφηκαν σε προηγούμενο κεφάλαιο.

Στο κεφάλαιο αυτό παρουσιάζονται τα βασικά στοιχεία της εφαρμογής, οι επιμέρους οθόνες και οι λειτουργίες που προσφέρονται στον χρήστη.

3.1 Τεχνικά Χαρακτηριστικά

Η ανάπτυξη της εφαρμογής έγινε με το πρόγραμμα Android Studio[19]. Η ελάχιστη έκδοση λειτουργικού Android στην οποία μπορεί να εγκατασταθεί η εφαρμογή είναι Android 4.0 (ICE_CREAM_SANDWICH) (API level 14). Επιπλέον η εφαρμογή χρησιμοποιεί ορισμένες εξωτερικές βιβλιοθήκες οι οποίες δηλώνονται ως εξαρτήσεις (dependancy) στο αρχείο build.gradle[31]. Συγκεκριμένα χρησιμοποιείται η βιβλιοθήκη Android Support Library[32] με την οποία η google δίνει τη δυνατότητα υποστήριξης σε παλαιότερες εκδόσεις λειτουργικού, η βιβλιοθήκη RecyclerView[33] με την οποία διευκολύνεται η διαχείριση και παρουσίαση μεγάλων συνόλων δεδομένων, η βιβλιοθήκη gson[25] η οποία επιτρέπει την αυτόματη μετατροπή json αντικειμένων σε αντικείμενα java και τέλος η βιβλιοθήκη υπηρεσιών Google Play για εμφάνιση χαρτών.

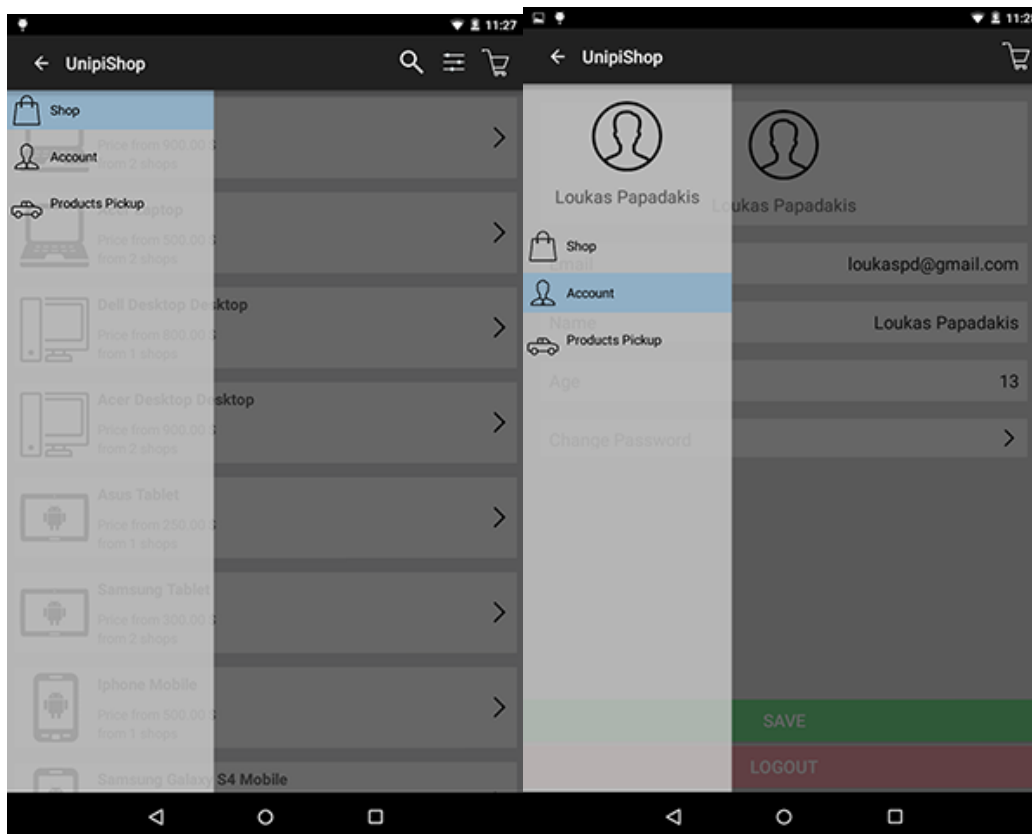
3.2 Βασικά στοιχεία εφαρμογής

Για την πλοήγηση μεταξύ των οθονών της εφαρμογής χρησιμοποιείται συρόμενο πλαϊνό μενού Navigation Drawer [1]. Πρόκειται για ένα μενού το οποίο περιέχει σαν επιλογές τις κύριες οθόνες της εφαρμογής στις οποίες ο χρήστης μπορεί να πλοηγηθεί κάνοντας κλικ. Προκειμένου να εξοικονομηθεί χώρος το μενού παραμένει κρυμμένο και εμφανίζεται μόνο όταν ο χρήστης σύρει την άκρη της οθόνης είτε πατώντας το κουμπί στην μπάρα ενεργειών της εφαρμογής.

Για την υλοποίηση του μενού χρησιμοποιείται ένα βασικό Activity (MainActivity) το οποίο περιέχει DrawerLayout [2] με πρώτο στοιχείο ένα LinearLayout, ενώ σαν δεύτερο τοποθετείται ένα FrameLayout το οποίο περιλαμβάνει το περιεχόμενο του μενού και μια λίστα με τις επιλογές πλοήγησης. Χρησιμοποιείται ακόμα μια Singleton κλάση η οποία παρακολουθεί τα κλικ στην λίστα του μενού και αναλαμβάνει των οθονών. Οι επιμέρους οθόνες υλοποιούνται ως Fragments [3,4] τα οποία τοποθετούνται στη θέση του LinearLayout του βασικού Activity.

Τέλος, στο μενού εμφανίζονται τα στοιχεία του χρήστη σε περίπτωση που αυτός είναι συνδεδεμένος.

Η εφαρμογή διαθέτει επίσης γραμμή ενεργειών Actionbar [5,6] η οποία προσφέρει στο χρήστη γρήγορη πρόσβαση σε συνήθεις ενέργειες της εφαρμογής όπως η εμφάνιση του μενού πλοήγησης ή η μετάβαση στο καλάθι αγορών (θα παρουσιαστεί αργότερα). Επιπλέον μπορούμε να εισάγουμε επιπλέον ενέργειες που σχετίζονται με τις επιμέρους οθόνες, όταν αυτές είναι ενεργές.



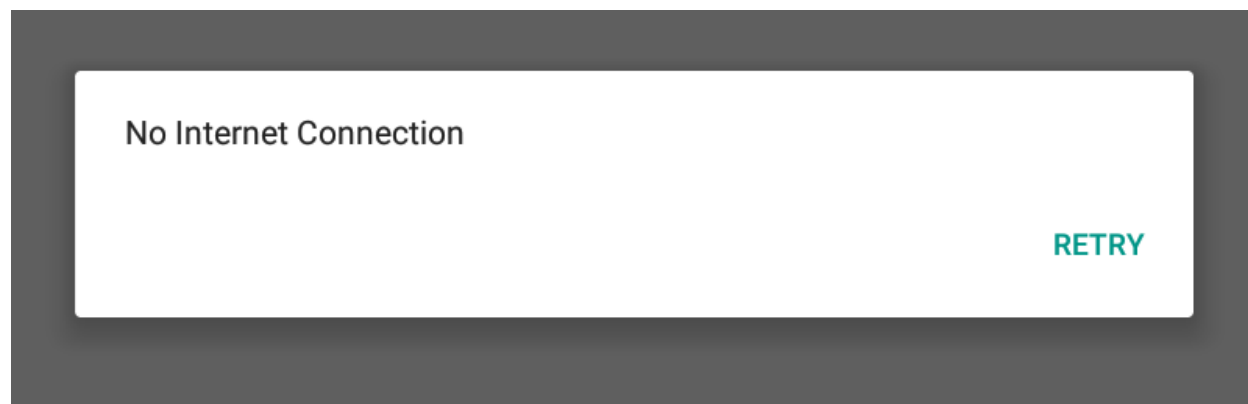
εικόνα 3: Συρόμενο μενού πλοήγησης, γραμμή ενεργειών εφαρμογής

Τα παραπάνω στοιχεία χρησιμοποιούνται σε όλες τις οθόνες της εφαρμογής, με εξαίρεση οθόνες που λειτουργούν αυτόνομα και οι οποίες έχουν υλοποιηθεί ως ξεχωριστά Activities. Τέτοια είναι για παράδειγμα η οθόνη παρακολούθησης παραλαβής προϊόντων που θα παρουσιαστεί παρακάτω.

3.3 Οθόνη Παρουσίασης Προϊόντων

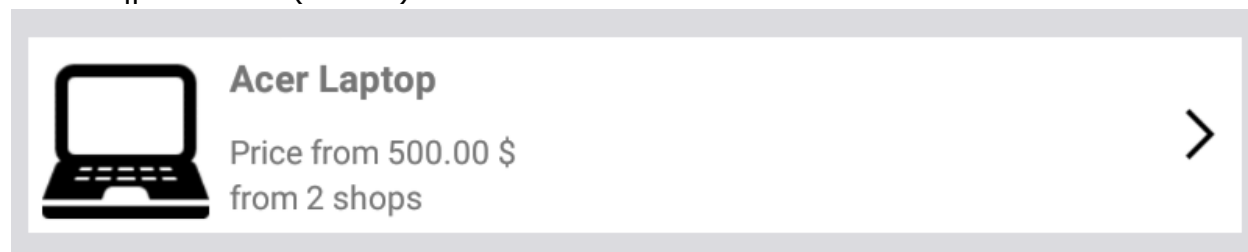
Η κεντρική οθόνη της εφαρμογής είναι η οθόνη παρουσίασης προϊόντων που είναι διαθέσιμα προς πώληση από τα καταστήματα του συστήματος. Τα δεδομένα λαμβάνονται μέσω των web services που περιγράφηκαν σε προηγούμενη

παράγραφο και αποθηκεύονται στη μνήμη της συσκευής, προκειμένου να μην εκτελούνται επαναλαμβανόμενες κλήσεις στα web services. Για το λόγο αυτό είναι απαραίτητο η συσκευή να διαθέτει σύνδεση στο δίκτυο, ενώ σε διαφορετική περίπτωση εμφανίζεται μήνυμα σφάλματος το οποίο διαθέτει κουμπί με το οποίο μπορεί να γίνει εκ νέου προσπάθεια φόρτωσης των δεδομένων μόλις η συσκευή συνδεθεί στο δίκτυο.



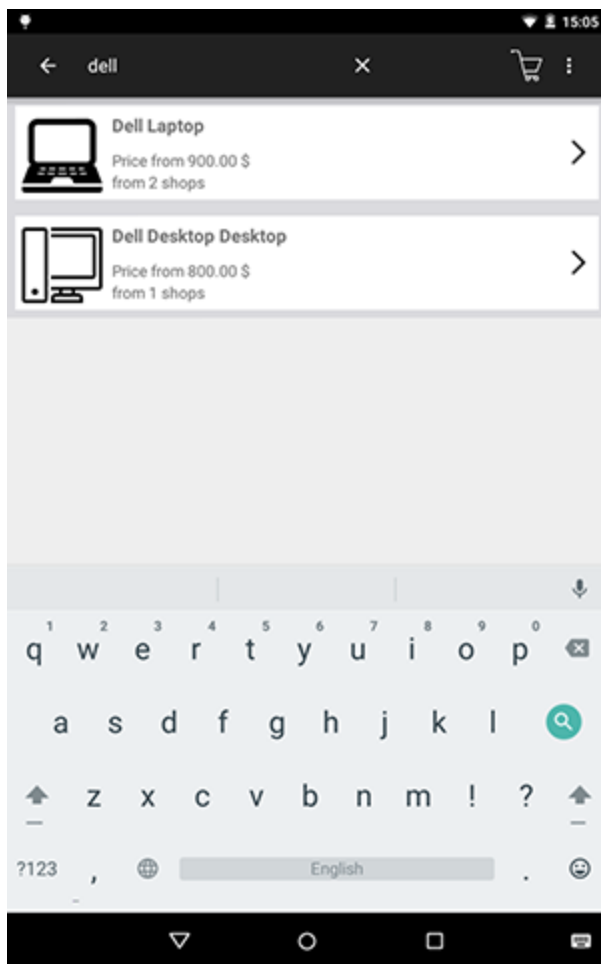
εικόνα 4: dialog ενημέρωσης χρήστη για σύνδεση στο δίκτυο

Σε περίπτωση επιτυχούς φόρτωσης των δεδομένων εμφανίζεται λίστα, κάθε αντικείμενο της οποίας περιέχει πληροφορίες για ένα από τα διαθέσιμα προς πώληση προϊόντα. Κάθε αντικείμενο περιέχει μια εικόνα που αντιστοιχεί στην κατηγορία του προϊόντος, το όνομα του προϊόντος, τον αριθμό των καταστημάτων που προσφέρουν το προϊόν και την ελάχιστη τιμή στην οποία είναι διαθέσιμο το προϊόν από τα καταστήματα αυτά (εικόνα).



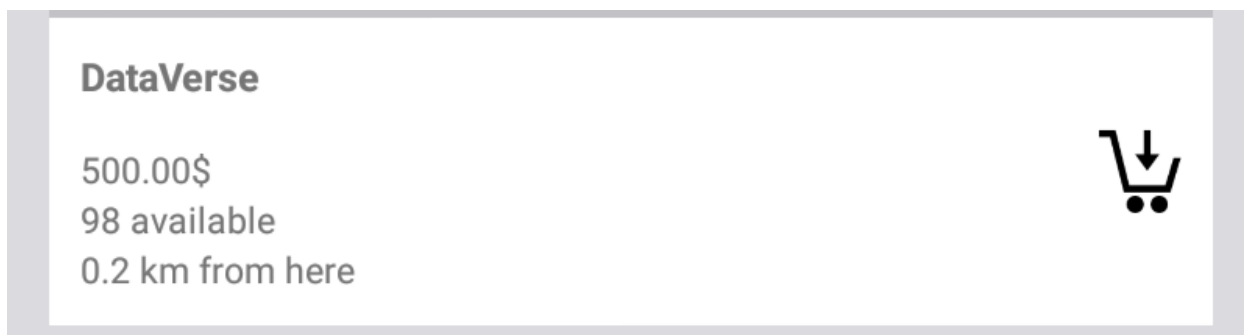
εικόνα 5: Αντικείμενο λίστας διαθέσιμων προϊόντων

Στην οθόνη εμφανίζεται η δυνατότητα αναζήτησης στη μπάρα ενεργειών [7] μέσω της οποίας ο χρήστης μπορεί να αναζητήσει κάποιο προϊόν στη λίστα. Πατώντας το κουμπί εμφανίζεται περιοχή εισαγωγής κειμένου και καθώς ο χρήστης εισάγει κείμενο, στη λίστα εμφανίζονται μόνο τα προϊόντα που περιέχουν το κείμενο αυτό.



εικόνα 6: Αναζήτηση στη λίστα διαθέσιμων προϊόντων

Ο χρήστης μπορεί να επιλέξει κάποιο προϊόν της λίστας κάνοντας κλικ οπότε εμφανίζεται νέα λίστα με λεπτομέρειες για τα καταστήματα που προσφέρουν το συγκεκριμένο προϊόν. Κάθε αντικείμενο της λίστας περιέχει το όνομα του καταστήματος, την τιμή στην οποία προσφέρει το επιλεγμένο προϊόν, τον αριθμό των τεμαχίων που διαθέτει το κατάστημα και τέλος, την απόσταση του καταστήματος από την θέση της συσκευής, σε περίπτωση που έχει βρεθεί η θέση της συσκευής.



εικόνα 7: Αντικείμενο λίστας καταστημάτων

Ο χρήστης μπορεί να προσθέσει ένα προϊόν στο καλάθι του πατώντας το αντίστοιχο κουμπί στο αντικείμενο της λίστας που αντιστοιχεί στο κατάστημα της επιθυμίας του. Στον χρήστη εμφανίζεται μήνυμα επιβεβαίωσης σε μορφή Snackbar [8,9] ενώ δίνεται και η δυνατότητα αναίρεσης της ενέργειας.



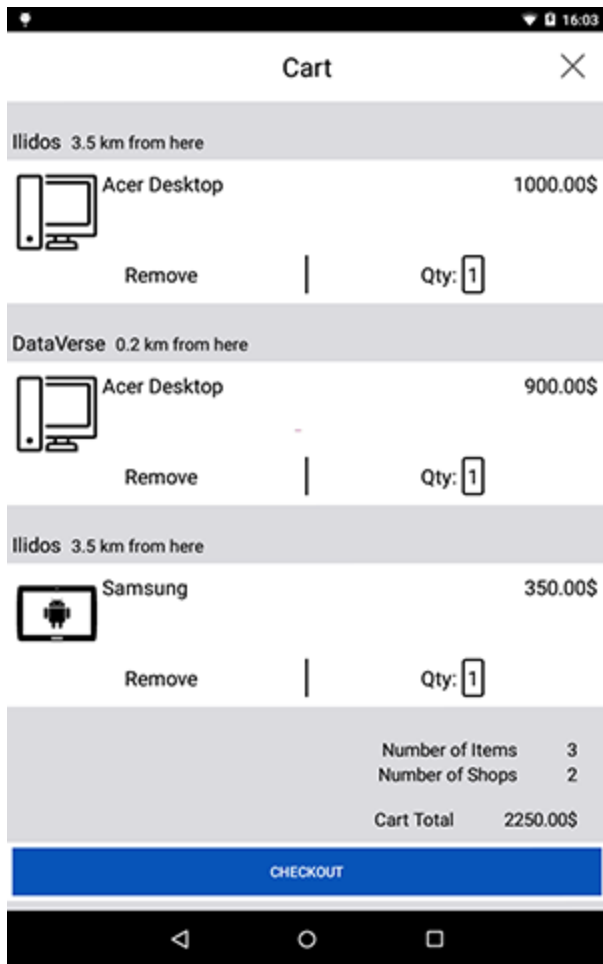
εικόνα 8: Λίστα καταστημάτων που προσφέρουν ένα προϊόν

Τέλος, πατώντας το βελάκι επιστροφής ο χρήστης έχει τη δυνατότητα να ξαναδεί τη λίστα των προϊόντων.

3.4 Καλάθι Αγορών - Οθόνη ολοκλήρωσης αγοράς

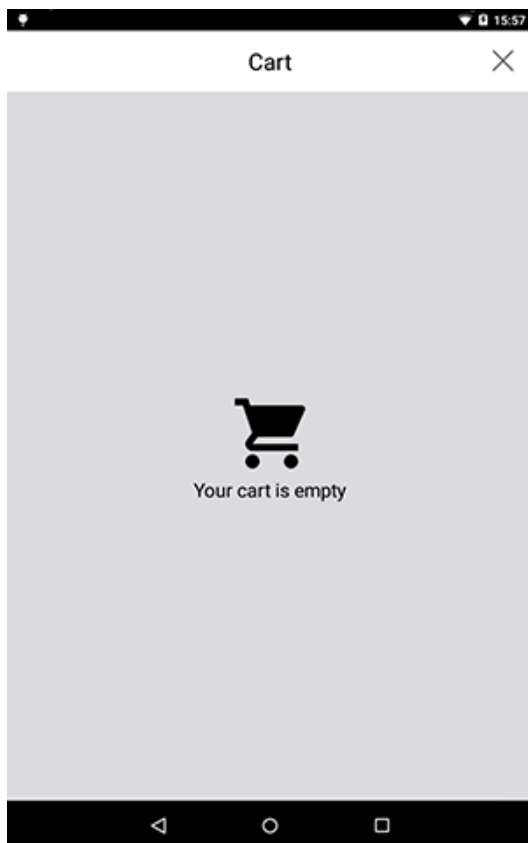
Στην οθόνη αυτή ο χρήστης μπορείς να επεξεργαστεί τα στοιχεία της παραγγελίας που έχει δημιουργήσει μέχρι στιγμής καθώς και να ολοκληρώσει την αποστολή της παραγγελίας στο σύστημα. Η οθόνη υλοποιείται ως ξεχωριστό Activity και δεν διαθέτει συρόμενο μενού πλοήγησης, αλλά η επιστροφή στη βασική εφαρμογή γίνεται πατώντας το κουμπί X με το οποίο και τερματίζεται το παρόν Activity.

Η οθόνη περιέχει λίστα με στοιχεία για τα προϊόντα που έχουν προστεθεί στο καλάθι του χρήστη στο τέλος της οποίας υπάρχει μια σύνοψη για το κόστος της παραγγελίας και το σύνολο των προϊόντων και καταστημάτων που συμμετέχουν στην παραγγελία.



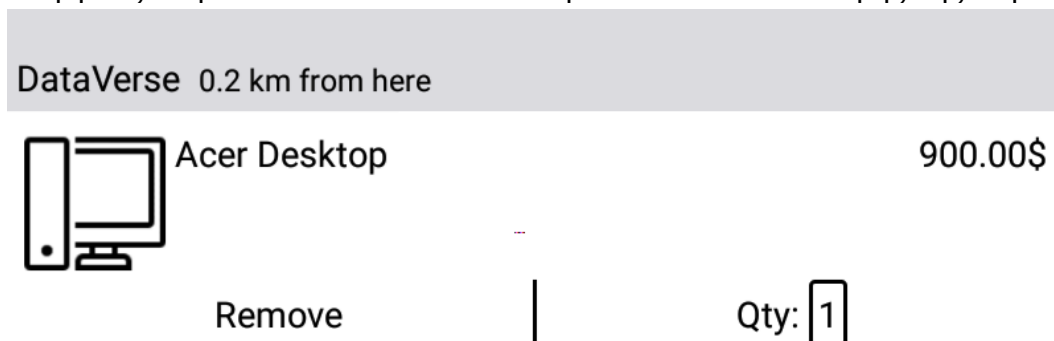
εικόνα 9: Καλάθι αγορών χρήστη

Σε περίπτωση που το καλάθι είναι δεν περιέχει προϊόντα εμφανίζεται ενημερωτικό μήνυμα, ενώ στο χρήστη δεν δίνεται η δυνατότητα εκτέλεσης κάποιας ενέργειας πέρα από την επιστροφή στην βασική εφαρμογή.



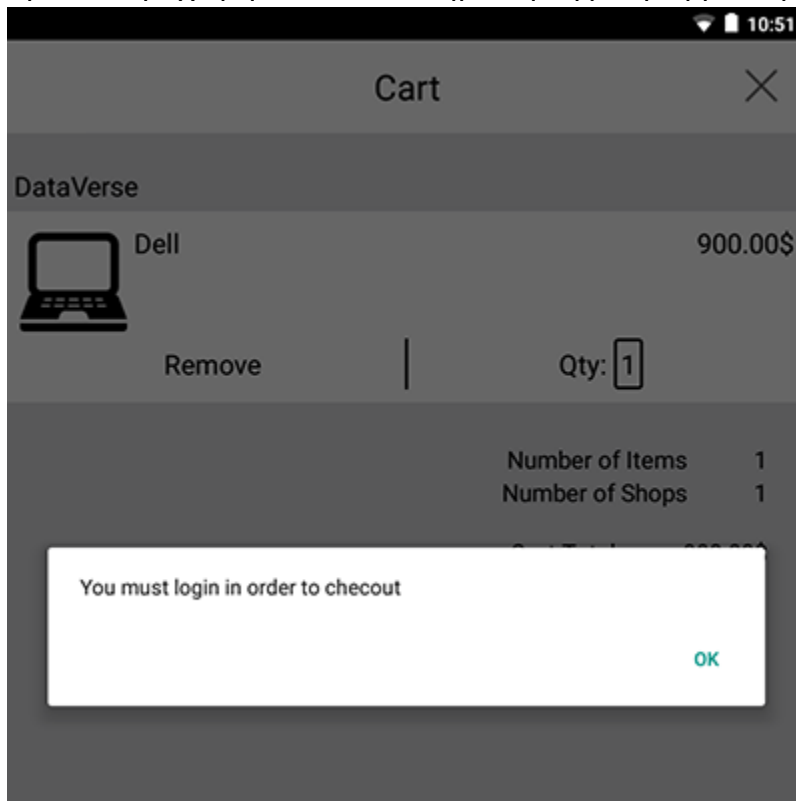
εικόνα 10: Μήνυμα ενημέρωσης καλαθιού χωρίς προϊόντα

Σε κάθε αντικείμενο της λίστας εμφανίζεται σαν τίτλος το όνομα του καταστήματος από το οποίο θα αγοραστεί το προϊόν καθώς και η απόστασή από αυτό σε περίπτωση που έχει βρεθεί η θέση της συσκευής. Στη συνέχεια υπάρχει ένα εικονίδιο που αντιστοιχεί στην κατηγορία του προϊόντος, το όνομα του προϊόντος καθώς και η τιμή στην οποία αυτό προσφέρεται από το κατάστημα. Μέσω του κουμπιού Remove ο χρήστης μπορεί να αφαιρέσει ένα προϊόν από τη λίστα ενώ μέσω του πεδίου ποσότητα (quantity) μπορεί να αλλάξει τον αριθμό των τεμαχίων ενός προϊόντος στο καλάθι. Κάθε φορά που ο χρήστης πραγματοποιεί κάποια από τις παραπάνω ενέργειες στη λίστα ανανεώνεται αυτόματα το πεδίο σύνοψης της παραγγελίας.



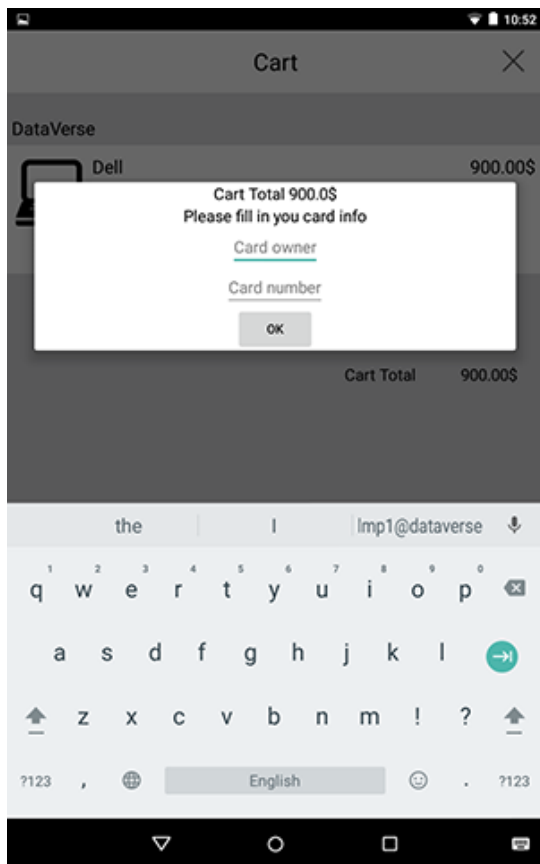
εικόνα 11: Αντικείμενο λίστας προϊόντων στο καλάθι χρήστη

Αφού ο χρήστης επεξεργαστεί τα προϊόντα στο καλάθι μπορεί να προχωρήσει στην ολοκλήρωση της παραγγελία πατώντας το αντίστοιχο κουμπί (checkout). Σε περίπτωση που ο χρήστης δεν έχει συνδεθεί στην εφαρμογή εμφανίζεται μήνυμα σφάλματος που ενημερώνει τον χρήστη ότι πρέπει να πραγματοποιήσει σύνδεση προτού προχωρήσει σε ολοκλήρωση της παραγγελίας.



εικόνα 12: Μήνυμα σφάλματος κατά την προσπάθεια ολοκλήρωσης παραγγελίας

Σε περίπτωση που ο χρήστης είναι συνδεδεμένος εμφανίζεται dialog στο οποίο εμφανίζεται το συνολικό κόστος της παραγγελίας, και ζητείται από τον χρήστη να εισάγει τα στοιχεία της πιστωτικής κάρτας με την οποία θα πραγματοποιηθεί η πληρωμή. Τα στοιχεία στέλνονται στον server μέσω του αντίστοιχου web service, ο χρήστης ενημερώνεται για την ολοκλήρωση της παραγγελίας και οδηγείται πίσω στο βασικό Activity της εφαρμογής, ενώ αδειάζει και το καλάθι των αγορών.

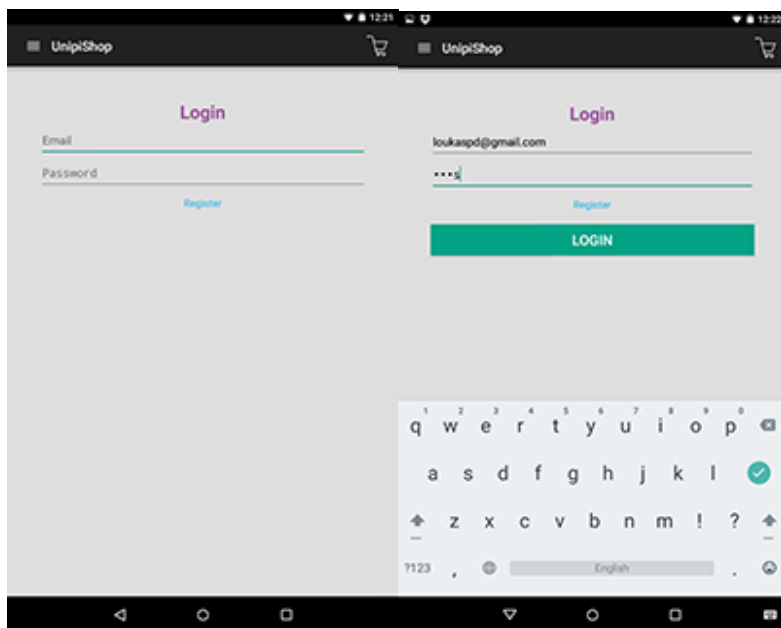


εικόνα 13: dialog ολοκλήρωσης παραγγελίας

3.5 Οθόνη Σύνδεσης - Εγγραφής Χρήστη

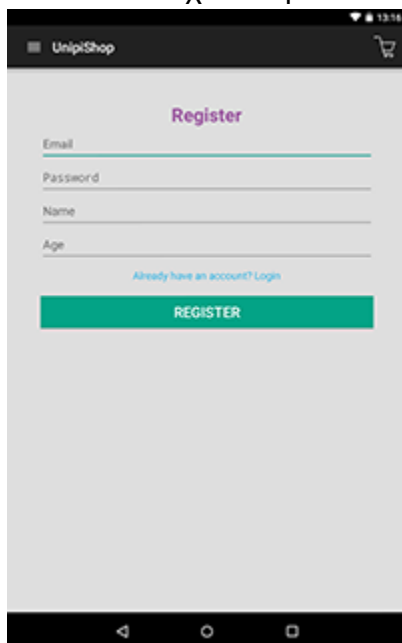
Η οθόνη αυτή εμφανίζεται όταν ο χρήστης θελήσει να μεταβεί στην οθόνη στοιχείων χρήστη ή στην οθόνη παραλαβής προϊόντων χωρίς να έχει προηγουμένως συνδεθεί και δίνει τη δυνατότητα σύνδεσης ή δημιουργίας λογαριασμού στο σύστημα.

Αρχικά εμφανίζεται η οθόνη σύνδεσης στην οποία ο χρήστης καλείται να εισάγει το email και τον κωδικό πρόσβασης του λογαριασμού του, ενώ παρέχεται και η δυνατότητα μετάβασης στη σελίδα εγγραφής μέσω του αντίστοιχου κουμπιού. Όταν έχουν εισαχθεί τα στοιχεία και στα δυο πεδία εμφανίζεται με ένα animation το κουμπι σύνδεσης.



εικόνα 14: Οθόνη σύνδεσης χρήστη

Η οθόνη εγγραφής απαιτεί από τον χρήστη να εισάγει τα στοιχεία με τα οποία θα δημιουργηθεί λογαριασμός στο σύστημα. Εκτός των πεδίων που εμφανίζονται η εφαρμογή στέλνει μέσω του web service εγγραφής και την θέση της συσκευής. Στην οθόνη δίνεται η δυνατότητα στο χρήστη να μεταβεί στην οθόνη σύνδεσης πατώντας στο αντίστοιχο κουμπί.



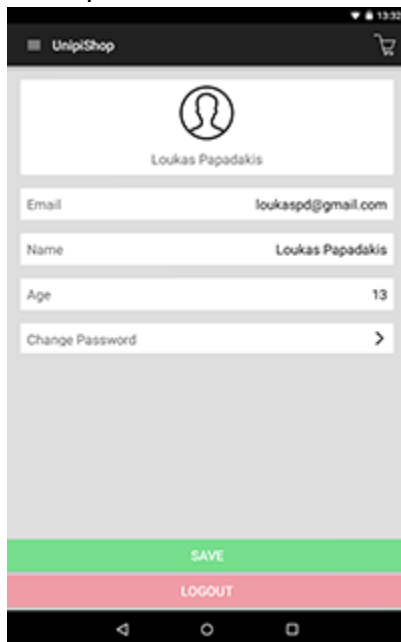
εικόνα 15: Οθόνη εγγραφής χρήστη

Κατά τη σύνδεση / εγγραφή χρήστη σε κάποιον λογαριασμό τα στοιχεία του αποθηκεύονται από την εφαρμογή ώστε να μην απαιτείται συμπλήρωση των

στοιχείων σύνδεσης από τον χρήστη σε κάθε εκκίνηση της εφαρμογής. Η αποθήκευση γίνεται χρησιμοποιώντας το SharedPreferences API [17,18] του λειτουργικού Android το οποίο μας δίνει τη δυνατότητα εύκολης αποθήκευσης δεδομένων με κάποιο κλειδί.

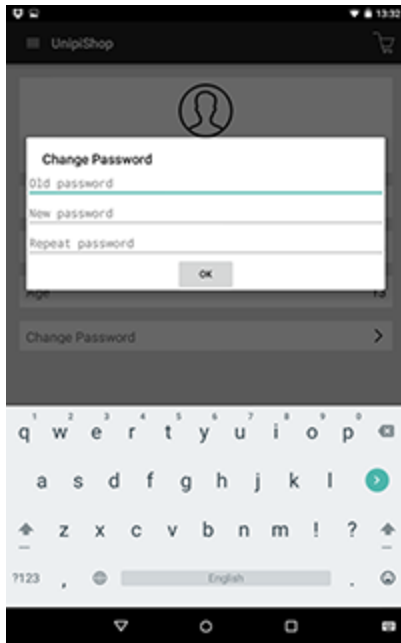
3.6 Οθόνη Στοιχείων Χρήστη

Στην οθόνη στοιχείων ο χρήστης μπορεί να δει τα στοιχεία του λογαριασμού με τον οποίο έχει συνδεθεί. Ο χρήστης έχει τη δυνατότητα να αλλάξει τις τιμές των πεδίων και να ανανεώσει το προφίλ του πατώντας το αντίστοιχο κουμπί (save). Υπάρχει ακόμα η δυνατότητα αποσύνδεσης του χρήστη από το συγκεκριμένο λογαριασμό, πατώντας το κουμπί logout, στην περίπτωση αυτή ο χρήστης μετακινείται στην οθόνη σύνδεσης προκειμένου να συνδεθεί σε κάποιον άλλο λογαριασμό αν το επιθυμεί.



εικόνα 16: Οθόνη στοιχείων χρήστη

Ο χρήστης έχει ακόμα τη δυνατότητα αλλαγής του κωδικού πρόσβασης του λογαριασμού. Πατώντας στο αντίστοιχο πεδίο εμφανίζεται dialog που περιέχει πεδία εισαγωγής του υπάρχοντος και του νέου κωδικού πρόσβασης.

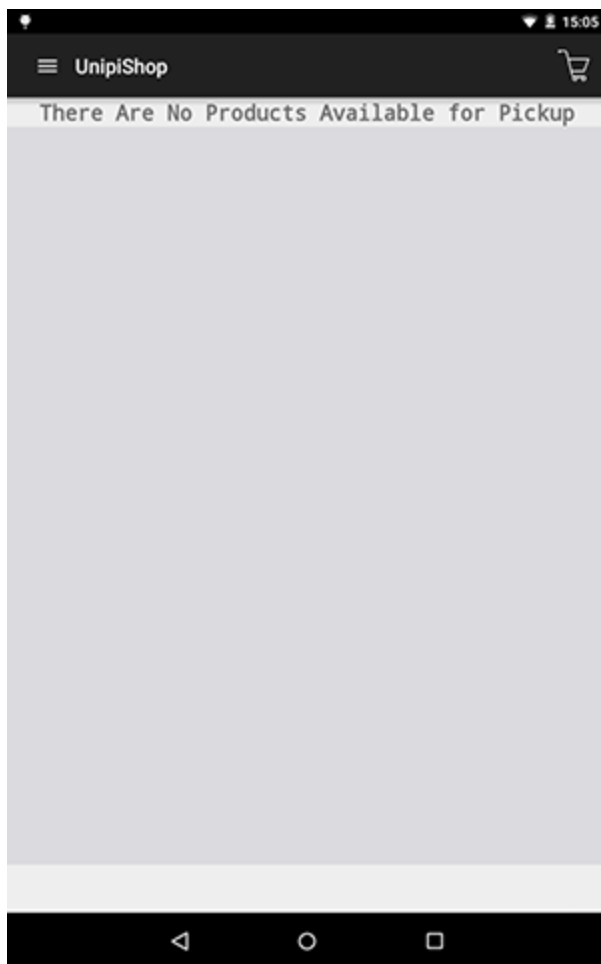


εικόνα 17: dialog αλλαγής κωδικού πρόσβασης

3.7 Οθόνη Έναρξης Παραλαβής Προϊόντων

Η οθόνη αυτή δίνει τη δυνατότητα στο χρήστη να δει τα προϊόντα που έχει αγοράσει από το σύστημα και βρίσκονται διαθέσιμα στο κατάστημα και να επιλέξει κάποια από αυτά για παραλαβή.

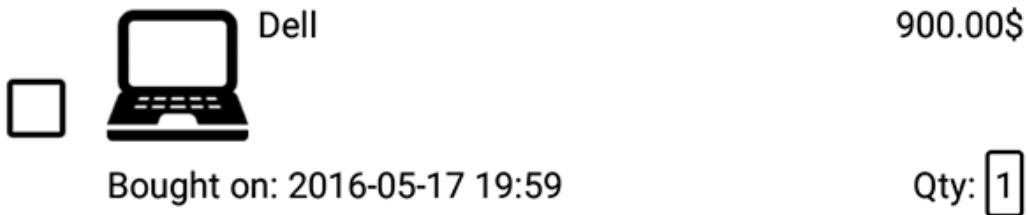
Όπως είναι λογικό η οθόνη απαιτεί ο χρήστης να είναι συνδεδεμένος με κάποιον λογαριασμό στο σύστημα, αν αυτό δεν συμβαίνει εμφανίζεται η οθόνη σύνδεσης. Με την είσοδο του χρήστη στην οθόνη γίνεται επίσης έλεγχος για σύνδεση της συσκευής στο δίκτυο και εμφανίζεται το αντίστοιχο μήνυμα σφάλματος σε dialog. Χρησιμοποιώντας το κατάλληλο web service φορτώνονται τα αγορασμένα προϊόντα με τον συγκεκριμένο λογαριασμό. Σε περίπτωση που δεν υπάρχουν τέτοια εμφανίζεται το αντίστοιχο μήνυμα.



εικόνα 18: Κενή λίστα προϊόντων προς παραλαβή

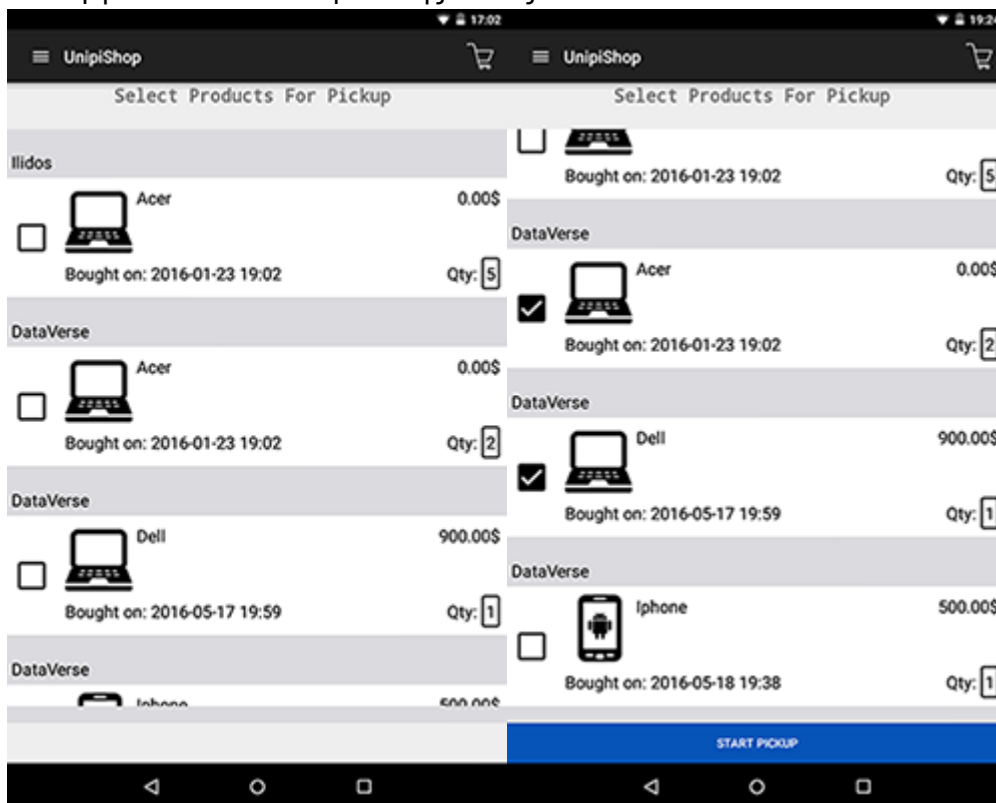
Τα προϊόντα εμφανίζονται σε λίστα ώστε να δίνεται η δυνατότητα στον χρήστη να επιλέξει ορισμένα από αυτά για παραλαβή. Μια παραλαβή μπορεί να περιέχει διάφορα προϊόντα από ξεχωριστές παραγγελίες αρκεί όλα να ανήκουν στο ίδιο κατάστημα. Για το λόγο αυτό, κάθε αντικείμενο της λίστα περιλαμβάνει πληροφορίες για ένα προϊόν από ένα συγκεκριμένο κατάστημα. Πιο συγκεκριμένα κάθε αντικείμενο παρουσιάζει το όνομα του καταστήματος από το οποίο οποίο έχει αγοραστεί το προϊόν, ένα εικονίδιο που αντιστοιχεί στο είδος του προϊόντος, το όνομα του προϊόντος και τον αριθμό των τεμαχίων που έχουν αγοραστεί από το συγκεκριμένο κατάστημα. Εμφανίζονται επιπλέον κάποιες ενημερωτικές πληροφορίες όπως το κόστος στο οποίο αγοράστηκε το προϊόν και την ημερομηνία πραγματοποίησης της αγοράς. Τέλος, σε κάθε αντικείμενο της λίστας υπάρχει ένα checkbox που δείχνει αν το προϊόν έχει επιλεγεί προς παραλαβή ή όχι.

DataVerse



εικόνα 19: Αντικείμενο λίστας προϊόντων στο καλάθι χρήστη

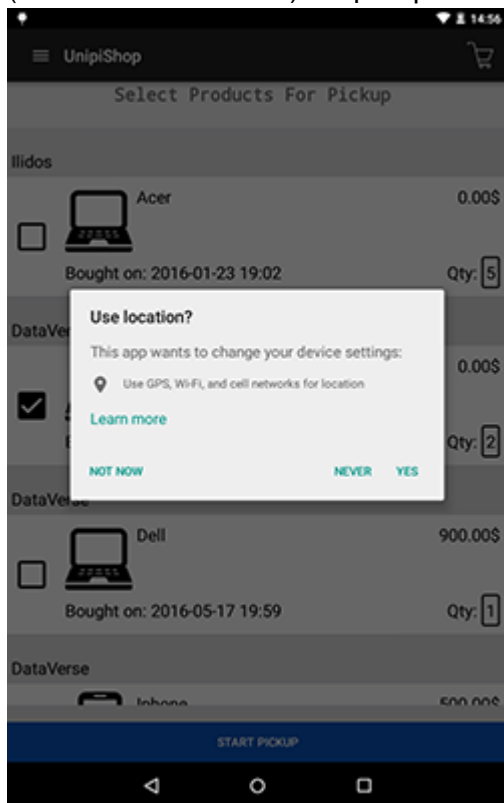
Ο χρήστης μπορεί πατώντας στα αντικείμενα της λίστας να επιλέξει ορισμένα από αυτά προς παραλαβή. Ένα αντικείμενο μπορεί να επιλεγεί μόνο αν έχει αγοραστεί από το ίδιο κατάστημα με τα προϊόντα που έχουν ήδη επιλεγεί σε διαφορετική περίπτωση το πάτημα του χρήστη αγνοείται και το αντικείμενο δεν εμφανίζεται ως επιλεγμένο. Όταν επιλεγεί ένα αντικείμενο της λίστας εμφανίζεται στην οθόνη το κουμπί το οποίο δίνει στο χρήστη τη δυνατότητα να εκκινήσει τη διαδικασία παραλαβής των προϊόντων που έχει επιλέξει. Μέχρι τότε το κουμπί δεν εμφανίζεται στην οθόνη όπως επίσης το κουμπί εξαφανίζεται εάν ο χρήστης καταργήσει την επιλογή σε όλα τα αντικείμενα της λίστας.



εικόνα 20: Εμφάνιση - απόκρυψη κουμπιού εκκίνησης παραλαβής

Όταν ο χρήστης έχει επιλέξει τα προϊόντα που θέλει να παραλάβει μπορεί να ξεκινήσει τη διαδικασία παραλαβής πατώντας στο αντίστοιχο κουμπί. Πριν την εκκίνηση της διαδικασίας η

εφαρμογή ελέγχει εάν ο χρήστης έχει ενεργοποιήσει της υπηρεσίες τοποθεσίας στην συσκευή και σε περίπτωση που αυτό δεν συμβαίνει η εφαρμογή τον προτρέπει να το κάνει μέσω κατάλληλου dialog του συστήματος. Η διαδικασία ελέγχου και προτροπής αλλαγής ρυθμίσεων υλοποιείται χρησιμοποιώντας τις υπηρεσίες Google Play, του API των υπηρεσιών τοποθεσίας (location services API) και με την κλάση LocationSettingsRequest [15,16].



εικόνα 21: Προτροπή χρήστη για αλλαγή ρυθμίσεων τοποθεσίας

Αφού επιβεβαιωθεί ότι η συσκευή έχει ενεργοποιημένες τις υπηρεσίες τοποθεσίας στις ρυθμίσεις γίνεται κλήση του web service μέσω του οποίου δηλώνονται στο σύστημα τα στοιχεία της παραλαβής, τα προϊόντα που θα παραληφθούν, το κατάστημα και τον χρήστη στον οποίο ανήκουν τα προϊόντα αυτά. Η εφαρμογή παίρνει σαν απάντηση από το σύστημα το Id της παραλαβής το οποίο χρησιμοποιείται σε επόμενες κλήσεις για την ενημέρωση θέσης του χρήστη και εκκινεί το Activity παρακολούθησης παραλαβής.

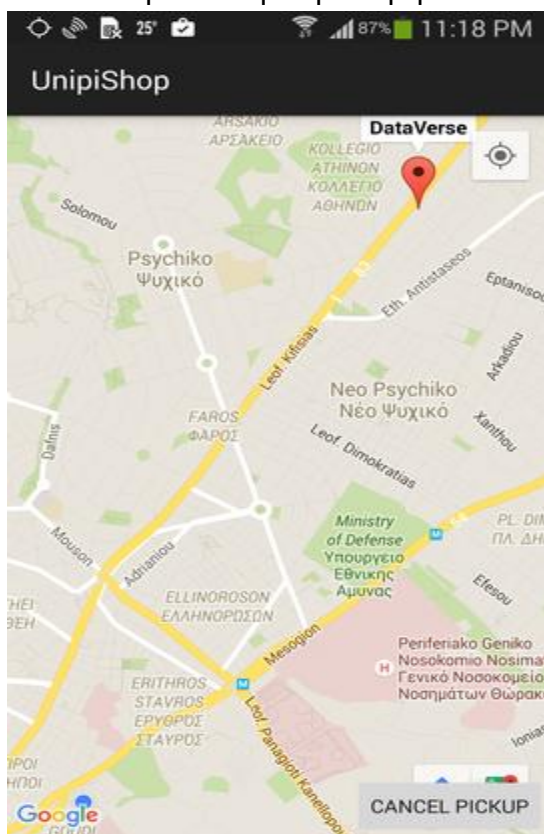
3.8 Οθόνη Παρακολούθησης Παραλαβής Προϊόντων

Η οθόνη αυτή δεν παρέχει πολλές δυνατότητες στον χρήστη αλλά σκοπός της είναι η εφαρμογή να παρακολουθεί τη θέση της συσκευής και να ενημερώνει το σύστημα ώστε τα καταστήματα να μπορούν να γνωρίζουν σε πραγματικό χρόνο την θέση των πελατών που παραλαμβάνουν κάποια προϊόντα.

Η οθόνη υλοποιείται ως ξεχωριστό Activity και δεν περιέχει παινό μενού πλοήγησης, ενώ έχει απενεργοποιηθεί και η δυνατότητα τερματισμού και επιστροφής

στο βασικό Activity πατώντας το κουμπί επιστροφής (back button) αντικαθιστώντας την μέθοδο `onBackPressed` της κλάσης Activity. Η δυνατότητα αυτή δίνεται στο χρήστη μέσω του κουμπιού ακύρωσης παραλαβής. Όταν το κουμπί πατηθεί από τον χρήστη η εφαρμογή τερματίζει το Activity αφού πρώτα καλέσει το web service με το οποίο ενημερώνει το σύστημα για την ακύρωση της παραλαβής.

Για την προβολή χάρτη χρησιμοποιούμε το API χαρτών της Google για λειτουργικό Android[26]. Αρχικά, προκειμένου να έχουμε δικαίωμα χρήσης του API θα πρέπει να δηλώσουμε την εφαρμογή μας στο Google Developers Console και να πάρουμε ένα κλειδί[27] το οποίο δηλώνεται στη συνέχεια στο αρχείο `manifest.xml` της εφαρμογής. Χρησιμοποιούμε το API μέσω της κλάσης `SupportMapFragment` [28]. Συγκεκριμένα, καλούμε τη μέθοδο `getMapAsync` για την φόρτωση του χάρτη η οποία με την ολοκλήρωση της φόρτωσης επιστρέφει ένα αντικείμενο της κλάσης `GoogleMap`[29]. Στον χάρτη τοποθετείται `Marker` στην θέση του καταστήματος από το οποίο γίνεται η παραλαβή.

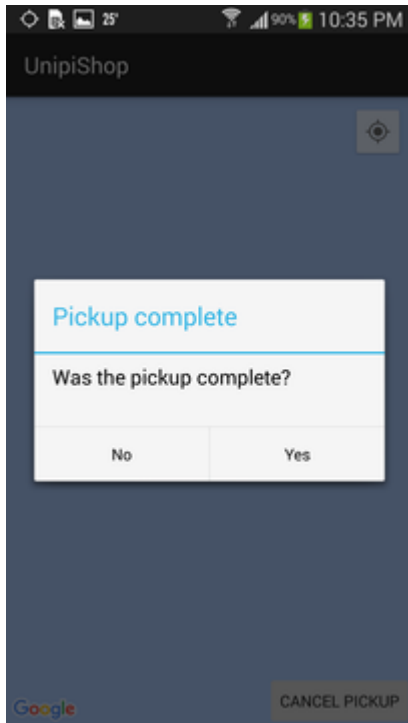


εικόνα 22: Τοποθέτηση marker στη θέση του καταστήματος παραλαβής

Στη συνέχεια παρακολουθούμε την θέση της συσκευής και σε κάθε αλλαγή της ελέγχουμε αν πρέπει να ενημερωθεί το σύστημα καλώντας το αντίστοιχο web service. Αυτό θα συμβεί αν έχει περάσει αρκετός χρόνος ή η συσκευή έχει μετακινηθεί αρκετά σε σχέση με τον χρόνο και τη θέση της τελευταίας ενημέρωσης. Στην περίπτωση αυτή, εκτός από την κλήση του web service αλλάζουμε και το επίπεδο μεγένθυσης του χάρτη ώστε να φαίνονται ακριβώς η θέση της συσκευής

και του καταστήματος παραλαβής, χρησιμοποιώντας τη μέθοδο `animateCamera` της κλάσης `GoogleMap`.

Τέλος, όταν η συσκευή έχει φτάσει αρκετά κοντά στο κατάστημα παραλαβής εμφανίζεται dialog με το οποίο ζητάμε από τον χρήστη να επιβεβαιώσει την ολοκλήρωση της παραλαβής



εικόνα 23: dialog επιβεβαίωσης ολοκλήρωσης παραλαβής

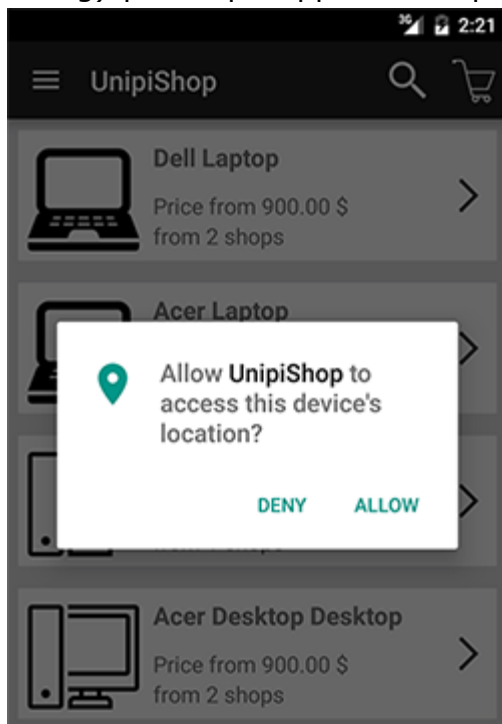
Εάν ο χρήστης απαντήσει θετικά καλείται το `service` με το οποίο ενημερώνουμε το σύστημα για την ολοκλήρωση της παραγγελίας και αφού λάβουμε απάντηση τερματίζουμε το `Activity` και ο χρήστης επιστρέφει στο κεντρικό `Activity`, στην οθόνη έναρξης παραλαβής.

3.9 Παρακολούθηση Τοποθεσίας

Όπως περιγράφηκε στα προηγούμενα κεφάλαια η εφαρμογή χρησιμοποιεί την θέση της συσκευής σε διάφορες λειτουργίες όπως εγγραφή χρήστη και φυσικά κατά την παρακολούθηση παραλαβής προϊόντων. Σύμφωνα με την `google` προτείνεται στους προγραμματιστές να χρησιμοποιούν το API των `Google Play Services` αντί του `framework` τοποθεσίας του `Android`[11]. Η σύνδεση με τις υπηρεσίες του `Google Play` γίνονται μέσω της κλάσης `GoogleApiClient`[12]. Η εφαρμογή χρησιμοποιεί μια `Singleton` κλάση η οποία αναλαμβάνει να παρακολουθεί την κατάσταση της σύνδεσης με τις υπηρεσίες και δημιουργεί αίτημα για ενημερώσεις στην αλλαγή

τοποθεσίας της συσκευής όταν αυτές απαιτούνται από κάποιο Activity της εφαρμογής. Η κλάση αυτή

Προκειμένου η εφαρμογή να μπορεί να παρακολουθεί την θέση της συσκευής (με τον τρόπο που περιγράφηκε προηγουμένως) θα πρέπει να έχει την άδεια του χρήστη. Ξεκινώντας από την έκδοση Android 6.0 (API level 23) οι χρήστες δίνουν άδεια για χρήση της τοποθεσίας της συσκευής τους όταν η εφαρμογή τρέχει και όχι κατά την εγκατάσταση, όπως συνέβαινε σε προηγούμενες εκδόσεις[13]. Κατά την εκκίνηση της εφαρμογής γίνεται έλεγχος και αν απαιτείται παρουσιάζεται dialog του συστήματος με το οποίο ζητείται από τον χρήστη να δώσει την παραπάνω άδεια. Ο έλεγχος αφορά την έκδοση Android της συσκευής καθώς και το εάν η εφαρμογή διαθέτει ήδη την άδεια χρήστη. Κατά τη διαδικασία αυτή χρησιμοποιείται η κλάση ContextCompat[14] καθώς ο έλεγχος απόκτησης άδειας γίνεται μέσω της μεθόδου checkSelfPermission, ενώ αίτηση για την άδεια (και η εμφάνιση του αντίστοιχου dialog) γίνεται με τη μέθοδο requestPermissions.



εικόνα 24: dialog αιτήματος για άδεια παρακολούθησης τοποθεσίας κατά

3.10 Web Requests

Η εφαρμογή χρησιμοποιεί τα web services που αναπτύχθηκαν σε προηγούμενο κεφάλαιο προκειμένου να επικοινωνεί με το σύστημα. Για την κλήση έχει δημιουργηθεί μια java κλάση (RequestClass) στον constructor της οποίας περνάμε όλες τις πληροφορίες που απαιτούνται από το web service, όπως την μέθοδο του

αιτήματος (GET, POST..) το url και τα δεδομένα που θα πρέπει να περάσουν σαν παράμετροι κατά την κλήση. Η κλάση αναλαμβάνει στη συνέχεια να καλέσει το service και να επιστρέψει την απάντηση σε μορφή string. Η υλοποίηση γίνεται μέσω της κλάσης HttpURLConnection[24]. Αρχικά γίνεται σύνδεση με το url του service καλώντας τη μέθοδο openConnection. Αν απαιτείται να περάσουμε παραμέτρους αυτό γίνεται χρησιμοποιώντας το OutputStream που παίρνουμε καλώντας τη μέθοδο getOutputStream. Τέλος, διαβάζουμε την απάντηση χρησιμοποιώντας ένα InputStream που παίρνουμε με τη μέθοδο getInputStream και μετατρέπουμε τα bytes σε String. Στο τέλος πάντα κλείνουμε τη σύνδεση καλώντας τη μέθοδο disconnect. Η απάντηση γνωρίζουμε ότι είναι σε μορφή json οπότε την μετατρέπουμε στην κλάση που αντιστοιχεί. Για τη μετατροπή χρησιμοποιούμε τη βιβλιοθήκη gson της google[25].

Η όλη διαδικασία είτε διαρκεί αρκετά είτε δεν μπορούμε να είμαστε βέβαιοι πόσο θα διαρκέσει, για το λόγο αυτό οι κλήσεις των web services δε θα πρέπει να συμβαίνουν στο κύριο Thread, καθώς σε αυτή την περίπτωση θα είχαμε πάγωμα της εφαρμογής πράγμα που θα οδηγούσε ακόμα και στο κλείσιμο της από το σύστημα και σίγουρα πολύ κακή εμπειρία χρήσης. Η μεταφορά σε background thread γίνεται χρησιμοποιώντας την abstract κλάση AsyncTask[30]. Για την ακρίβεια δημιουργείται μια κλάση για κάθε request η οποία επεκτίνει την AsyncTask. Η κλάση αυτή δέχεται στον constructor τις παραμέτρους που απαιτούνται από το web service και επιπλέον σε περίπτωση που η εφαρμογή πρέπει να εκτελέσει κάποια ενέργεια μετά την ολοκλήρωση του Request η κλάση δηλώνει ένα Interface το οποίο και εκτελείται στη μέθοδο onPostExecute η οποία τρέχει στο κύριο thread.

4. Ιστότοπος Συστήματος

Μέσω του ιστότοπου του συστήματος δίνεται η δυνατότητα αλληλεπίδρασης των καταστημάτων με αυτό. Στο κεφάλαιο αυτό παρουσιάζονται τα τεχνικά χαρακτηριστικά του ιστότοπου και οι οθόνες μέσα από τις οποίες κάθε κατάσταση που συμμετέχει στο σύστημα μπορεί να δει μια συνοπτική εικόνα της κατάστασής της, να δει τις πωλήσεις που έχουν πραγματοποιηθεί, να επεξεργαστεί τα προϊόντα που προσφέρει και να παρακολουθήσει σε πραγματικό χρόνο τις παραλαβές προϊόντων που βρίσκονται σε εξέλιξη από πελάτες - χρήστες της Android εφαρμογής.

4.1 Τεχνικά Χαρακτηριστικά

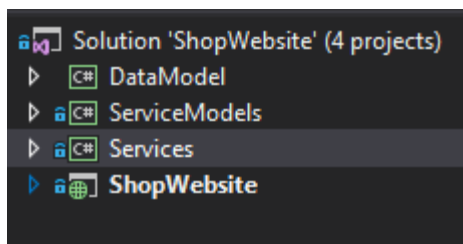
Η ανάπτυξη του website έγινε στο πρόγραμμα Visual Studio 2013[20]. Πρόκειται για ένα ASP.NET MVC website[21]. Προκειμένου να ελαχιστοποιήσουμε τον κώδικα που απαιτείται χρησιμοποιούμε τη βιβλιοθήκη Bootstrap[22] για να διαμορφώσουμε

στιλιστικά το site. Πρόκειται για ένα Framework ανοιχτού κώδικα το οποίο αποτελείται από κώδικα HTML, CSS και Javascript και έχει σχεδιαστεί για να διευκολύνει το σχεδιασμό της διεπαφής ενός site με το χρήστη. Τέλος, χρησιμοποιείται η βιβλιοθήκη Angular Js[23] κυρίως για να εκμεταλλευτούμε την δυνατότητα που προσφέρει να αλλάζουμε το περιεχόμενο μίας σελίδας σε πραγματικό χρόνο, ιδιαίτερα χρήσιμο κατά την παρακολούθηση παραλαβής από πελάτη.

Τόσο το ASP.NET MVC όσο και η βιβλιοθήκη Angular JS βασίζονται στην αρχιτεκτονική Model - View - Controller (MVC) σύμφωνα με την οποία γίνεται διαχωρισμός μιας εφαρμογής στα τρία αυτά βασικά στοιχεία. Ακολουθώντας την αρχιτεκτονική αυτή, επιτυγχάνουμε τον διαχωρισμό των διαφόρων πτυχών της εφαρμογής (ενέργειες χρήστη, λογική υλοποίησης, διεπαφή χρήστη) χρησιμοποιώντας μια χαλαρή σύζευξη μεταξύ τους. Ο διαχωρισμός αυτός μας βοηθάει να ελαττώνουμε την πολυπλοκότητα κατά τη δημιουργία μιας εφαρμογής καθώς μπορούμε να συγκεντρωθούμε σε κάθε κομμάτι ξεχωριστά. Διευκολύνει ακόμα την παράλληλη ανάπτυξη της εφαρμογής καθώς και δημιουργία δοκιμών κατά την ανάπτυξη (test-driven development).

4.2 Δομή του Solution

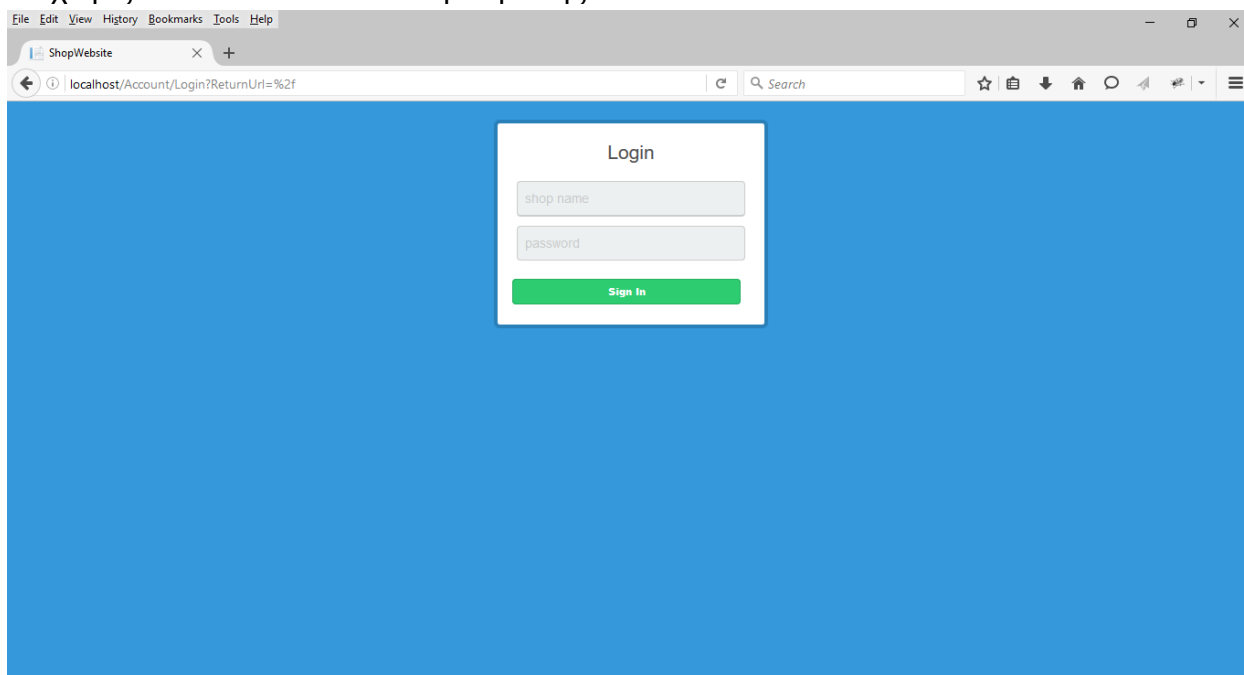
Το site δημιουργήθηκε με το πρόγραμμα Visual Studio ως ένα ASP web application project. Το project αυτό περιλαμβάνει όλες τα views και τους controllers του site, τον κώδικα css και javascript. Το solution περιλαμβάνει ωστόσο ακόμα τρία project. Το project DataModel δίνει τη δυνατότητα επικοινωνίας με τη βάση δεδομένων και είναι το ίδιο project το οποίο χρησιμοποιείται και στο Solution των web services που περιγράφηκε σε προηγούμενο κεφάλαιο. Το project ServiceModels περιλαμβάνει τα μοντέλα που θα χρησιμοποιηθούν από το site για την εμφάνιση του περιεχομένου στις σελίδες. Το project Services λειτουργεί ως ενδιάμεσος μεταξύ του project του website και αυτού της βάσης δεδομένων, χρησιμοποιεί το project DataModel για να επικοινωνεί με τη βάση δεδομένων και περιλαμβάνει κώδικα για τη μετατροπή των κλάσεων της βάσης δεδομένων στις αντίστοιχες του project ServiceModels. Τα πλεονεκτήματα της ανάλυσης του Solution σε επιμέρους project έγιναν εμφανή ήδη με την χρησιμοποίηση του κοινού project και επομένως κοινού κώδικα για την επικοινωνία με τη βάση δεδομένων από το solution των web services και του web site, ενώ διευκολύνει πολύ στη συντηρησιμότητα του project και στην υλοποίηση αλλαγών.



εικόνα 25: Δομή του Solution ιστότοπου

4.3 Οθόνη Σύνδεσης Καταστήματος

Μέσω του ιστοτόπου δίνεται η δυνατότητα στα καταστήματα να αλληλεπιδρούν με τα στοιχεία του συστήματος, καμία ενέργεια δεν είναι δυνατή αν πρώτα δεν αναγνωριστεί από το σύστημα το κατάστημα του οποίου τα στοιχεία θέλει να δει ο χρήστης. Με την πλοήγηση του χρήστη στον ιστότοπο οδηγείται στη σελίδα σύνδεσης όπου καλείται να εισάγει το όνομα του καταστήματος το οποίο διαχειρίζεται και τον κωδικό πρόσβασης.

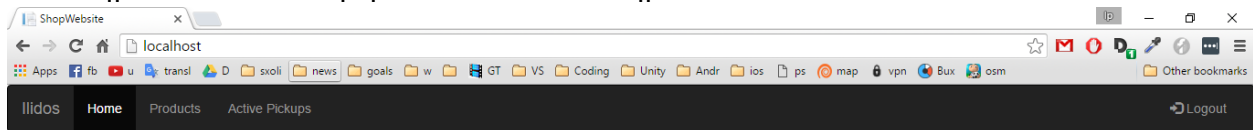


εικόνα 26: Οθόνη σύνδεσης καταστήματος

4.4 Κεντρική Σελίδα Καταστήματος

Μετά τη σύνδεση του χρήστη και την ταυτοποίηση του καταστήματος, οδηγείται στην κεντρική σελίδα του ιστοτόπου. Στην οθόνη αυτή παρουσιάζεται μια συνοπτική εικόνα των στοιχείων του καταστήματος. Συγκεκριμένα, εμφανίζεται το πλήθος των προϊόντων που προσφέρει το κατάστημα, το πλήθος των παραλαβών προϊόντων που

βρίσκονται σε εξέλιξη, ο συνολικός αριθμός παραλαβών που έχουν ολοκληρωθεί, το πλήθος των προϊόντων που έχουν αγοραστεί και ο αριθμός των πελατών που έχουν ολοκληρώσει κάποια αγορά από το κατάστημα.



Iidos

Available Products : 4

Active Pickups : 0

Total Number of Pickups : 0

Bought Products : 3

Number of customers : 2

εικόνα 27: Κεντρική σελίδα καταστήματος

4.5 Σελίδα Διαθέσιμων προϊόντων

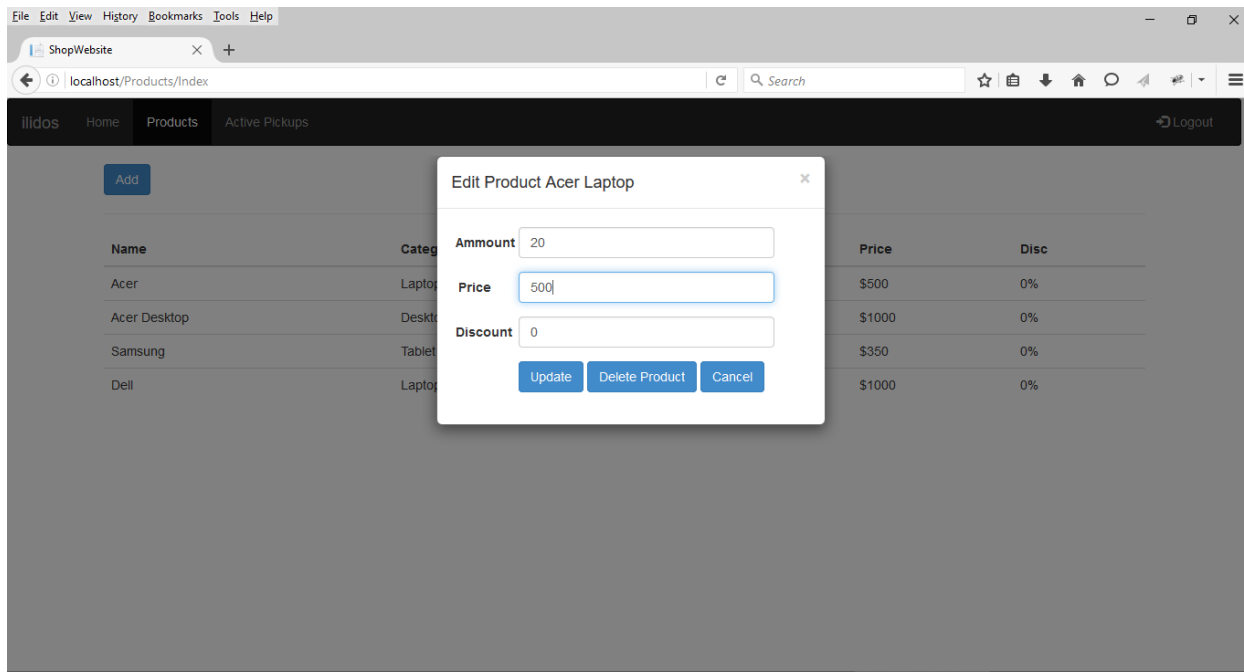
Στη σελίδα αυτή εμφανίζονται τα προϊόντα του συστήματος τα οποία διαθέτει προς πώληση το κατάστημα, λεπτομέρειες σχετικά με αυτά, ενώ δύναται δυνατότητα προσθήκης νέων προϊόντων ή επεξεργασίας των υπάρχοντων.

Τα προϊόντα παρουσιάζονται σε πίνακα, κάθε για κάθε ένα εμφανίζεται το όνομα, η κατηγορία στην οποία ανήκει, ο αριθμός που βρίσκεται σε διαθέσιμα στο κατάστημα, η τιμή στην οποία διατίθεται καθώς και το ποσοστό έκπτωσης που προσφέρεται επί της αρχικής τιμής.

Name	Category	Ammount	Price	Disc
Acer	Laptop	20	\$500	0%
Acer Desktop	Desktop	100	\$1000	0%
Samsung	Tablet	60	\$350	0%
Dell	Laptop	20	\$1000	0%

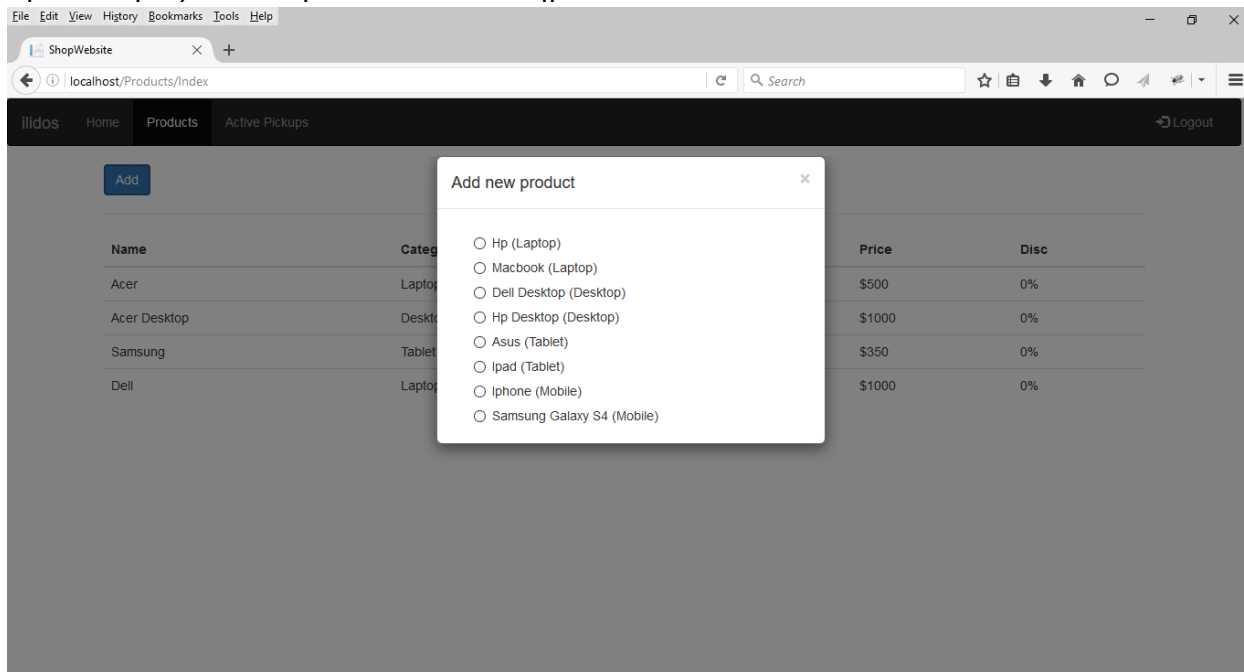
εικόνα 28: Πίνακας παρουσίασης διαθέσιμων προϊόντων καταστήματος

Ο χρήστης έχει τη δυνατότητα να επεξεργαστεί τα δεδομένα ενός προϊόντος που διατίθεται από το κατάστημα, κάνοντας κλικ πάνω στην αντίστοιχη εγγραφή του πίνακα. Στην περίπτωση αυτή εμφανίζεται ένα παράθυρο στο οποίο ο χρήστης μπορεί να μεταβάλει τον αριθμό των προϊόντων που διατίθενται στο κατάστημα, την τιμή και το ποσοστό έκπτωσης. Όταν ο χρήστης είναι ικανοποιημένος με τις αλλαγές πρέπει να πατήσει το κουμπί ανανέωσης, με το οποίο καταχωρούνται οι αλλαγές στο σύστημα και ο χρήστης μεταφέρεται στον κεντρικό πίνακα προϊόντων όπου πλέον τα δεδομένα έχουν ανανεωθεί. Σε περίπτωση που ο χρήστης αποφασίσει ότι δεν θέλει να γίνουν αλλαγές μπορεί να επιστρέψει στον κεντρικό πίνακα πατώντας το κουμπί ακύρωσης. Τέλος, με το κουμπί διαγραφής το σύστημα ενημερώνεται ότι το κατάστημα σταματάει να προσφέρει το συγκεκριμένο προϊόν.



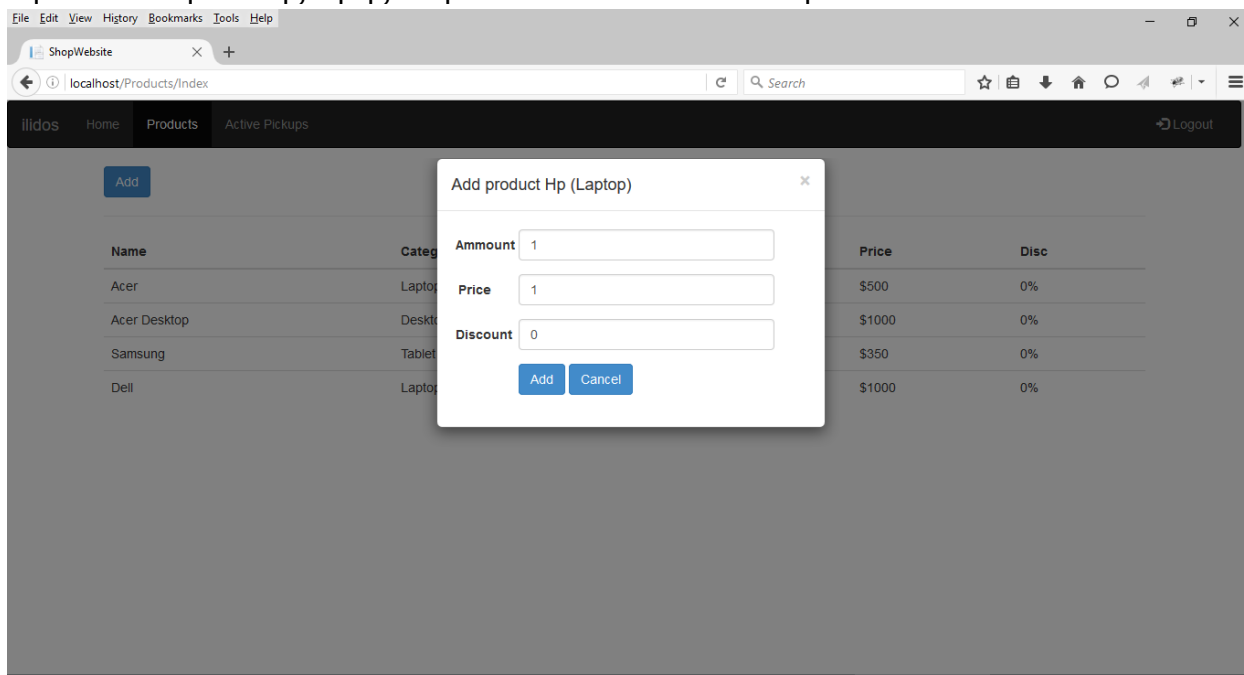
εικόνα 29: Παράθυρο επεξεργασία στοιχείων προϊόντος

Ο χρήστης μπορεί να προσθέσει στη λίστα διαθέσιμων προϊόντων του καταστήματος κάποιο από τα προϊόντα του συστήματος πατώντας το αντίστοιχο κουμπί. Στην περίπτωση αυτή εμφανίζεται παράθυρο στο οποίο ο χρήστης καλείται να επιλέξει το προϊόν προς διάθεση από το κατάστημα.



εικόνα 30: Παράθυρο επιλογής προϊόντος προς διάθεση

Στη συνέχεια, ο χρήστης καλείται να εισάγει τα στοιχεία στα οποία θα διαθέσει το προϊόν, και πιο συγκεκριμένα, τον αριθμό των διαθέσιμων προϊόντων, την τιμή και την έκπτωση επί της τιμής στην οποία θα διατεθεί το προϊόν.



εικόνα 31: Παράθυρο εισαγωγής στοιχείων προϊόντος προς διάθεση

Τέλος, πατώντας το κουμπί προσθήκης το σύστημα ενημερώνεται για την διάθεση του προϊόντος, ο χρήστης επιστρέφει στην κεντρική λίστα προϊόντων όπου πλέον τα δεδομένα έχουν ανανεωθεί.

4.6 Σελίδα Παρακολούθησης Παραλαβών

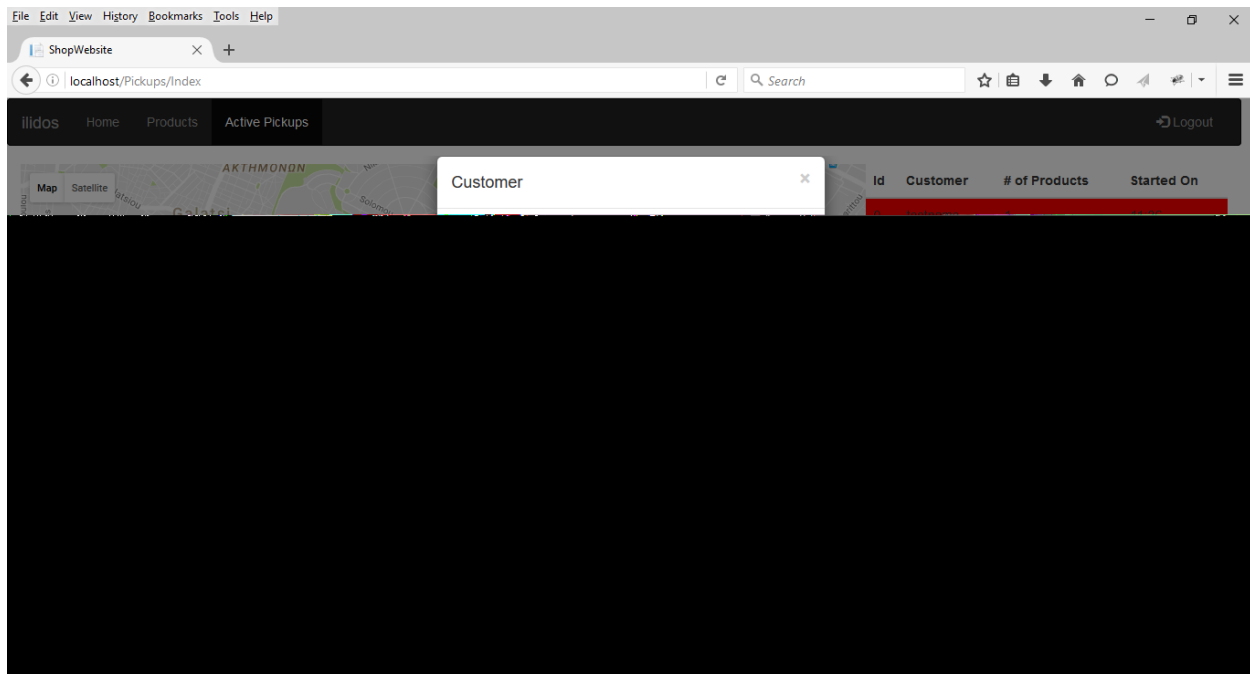
Μέσω της σελίδας αυτής ο χρήστης μπορεί να παρακολουθήσει σε πραγματικό χρόνο τη θέση των πελατών του καταστήματος που πραγματοποιούν κάποια παραλαβή από αυτό.

Βασικό στοιχείο της σελίδας είναι ο χάρτης. Με τη φόρτωση της σελίδας εισάγεται marker με κατάλληλο εικονίδιο στη θέση του καταστήματος ενώ ο χάρτης κεντράρεται στη θέση αυτή. Στη συνέχεια φορτώνονται οι θέσεις όλων των πελατών που πραγματοποιούν κάποια παραλαβή και εισάγονται markers αντίστοιχες θέσεις. Τα δεδομένα ανανεώνονται αυτόματα κάθε τριάντα δευτερόλεπτα, οπότε ενημερώνεται η θέση των markers στον χάρτη, εισάγονται οι νέες παραλαβές και αφαιρούνται αυτές που έχουν ολοκληρωθεί. Κάθε φορά ο χάρτης εστιάζει σε τέτοιο βαθμό ώστε να είναι εμφανείς όλοι οι markers των ενεργών παραλαβών.

Id	Customer	# of Products	Started On
0	testname	1	11:36

εικόνα 32: Οθόνη παρακολούθησης παραλαβών καταστήματος

Τα στοιχεία των παραλαβών εμφανίζονται σε πίνακα δίπλα από τον χάρτη. Εκεί ο χρήστης μπορεί να δει το όνομα του πελάτη, τον αριθμό των προϊόντων που παραλαμβάνονται και την ώρα που έγινε η εκκίνηση της παραλαβής. Κάθε maker στον χάρτη έχει ένα αριθμό με τον οποίο αντιστοιχίζεται στην αντίστοιχη εγγραφή του πίνακα. Κάνοντας κλικ σε κάποια από τις εγγραφές του πίνακα εμφανίζεται παράθυρο στο οποίο παρουσιάζεται το όνομα, η διεύθυνση ηλεκτρονικού ταχυδρομείου και η ηλικία του πελάτη που πραγματοποιεί την παραλαβή.



εικόνα 33: Παράθυρο στοιχείων πελάτη παραλαβής

5. Συμπεράσματα

Με την εργασία αυτή παρουσιάστηκε ένα σύστημα ηλεκτρονικών αγορών στο οποίο τα καταστήματα μπορούν να παρακολουθούν σε πραγματικό χρόνο την θέση των πελατών καθώς παραλαμβάνουν προϊόντα που έχουν αγοράσει από το κατάστημα.

Τα οφέλη από ένα τέτοιο σύστημα είναι πολλά τόσο για τα καταστήματα, όσο και για τους πελάτες. Αρχικά οι πελάτες έχουν τη δυνατότητα πραγματοποίησης αγορών οποιαδήποτε στιγμή και από οποιοδήποτε μέρος, μπορούν επιπλέον να ενημερώνονται για νέα προϊόντα καθώς και να διαχειρίζονται αγορές που έχουν πραγματοποιήσει. Τέλος, μέσω της ζωντανής παρακολούθησης της θέσης επιτυγχάνεται καλύτερη εμπειρία κατά την παραλαβή των προϊόντων, ένα τομέα που αποτελεί έναν από τους λόγους επιβράδυνσης της ανάπτυξης του ηλεκτρονικού εμπορίου. Τα καταστήματα μπορούν να ελέγχουν καλύτερα τα αποθέματά τους, την πορεία των πωλήσεων, μπορούν να προωθούν ευκολότερα τα προϊόντα και τις προσφορές τους, ενώ μέσω της ζωντανής παρακολούθησης της τοποθεσίας προσφέρουν καλύτερες υπηρεσίες κατά την παραλαβή των προϊόντων και μπορούν να προσελκύσουν νέους πελάτες.

Τέλος παρατίθενται κάποιες προτάσεις για μελλοντική έρευνα. Αρχικά το σύστημα θα πρέπει να δοκιμαστεί σε μεγαλύτερο αριθμό πελατών και καταστημάτων. Στη συνέχεια, διατηρώντας τα δεδομένα των αγορών που πραγματοποιούνται θα μπορούσε να δημιουργηθεί ένα σύστημα προτάσεων προϊόντων προς τους πελάτες με βάση τις προηγούμενες αγορές τους, ένα πρόγραμμα για αναλυτική παρουσίαση των πωλήσεων των καταστημάτων και πολλά ακόμα. Ακόμα, θα πρέπει να υλοποιηθεί ένα ασφαλές σύστημα ηλεκτρονικών πληρωμών προκειμένου να μπορούν να ολοκληρώνονται οι αγορές μέσω κινητών συσκευών, ενώ υπάρχουν πολλές επιπλέον προσθήκες που μπορούν να υλοποιηθούν για να προσφέρουν καλύτερες υπηρεσίες τόσο στους πελάτες όσο και στα καταστήματα που χρησιμοποιούν το σύστημα.

Βιβλιογραφία

1. Navigation Drawer : <http://developer.android.com/training/implementing-navigation/nav-drawer.html>
2. Drawer Layout :
<http://developer.android.com/reference/android/support/v4/widget/DrawerLayout.html>
3. Android Fragments :
<http://developer.android.com/guide/components/fragments.html>
4. Fragment Class :
<http://developer.android.com/reference/android/app/Fragment.html>
5. Adding the App Bar :
<http://developer.android.com/training/appbar/index.html>
6. ActionBar Class :
<https://developer.android.com/reference/android/support/v7/app/ActionBar.html>
7. Search Interface in app bar :
<http://developer.android.com/training/search/setup.html>
8. Display Popup Message with Snackbar:
<http://developer.android.com/training/snackbar/showing.html>
9. Snackbar Class :
<http://developer.android.com/reference/android/support/design/widget/Snackbar.html>
10. Google Play Services Location API :
<https://developer.android.com/reference/com/google/android/gms/location/package-summary.html>
11. Android location Framework :
<https://developer.android.com/reference/android/location/package-summary.html>
12. GoogleApiClient :
<https://developers.google.com/android/reference/com/google/android/gms/common/api/GoogleApiClient>
13. Requesting Permissions at Runtime :
<https://developer.android.com/training/permissions/requesting.html>
14. ContextCompat Class :
<https://developer.android.com/reference/android/support/v4/content/ContextCompat.html>
15. Changing Location Settings :
<https://developer.android.com/training/location/change-location-settings.html>

16. LocationSettingsRequest class :
<https://developers.google.com/android/reference/com/google/android/gms/location/LocationSettingsRequest.Builder>
17. Saving Key - Value Sets :
<https://developer.android.com/training/basics/data-storage/shared-preferences.html>
18. SharedPreferences Class :
<https://developer.android.com/reference/android/content/SharedPreferences.html>
19. Android Studio : <https://developer.android.com/studio/intro/index.html>
20. Visual Studio : <https://www.visualstudio.com>
21. ASP.NET MVC : <http://www.asp.net/mvc>
22. Bootstrap : <http://getbootstrap.com/>
23. Angular Js : <https://angularjs.org/>
24. HttpURLConnection class :
<https://developer.android.com/reference/java/net/HttpURLConnection.html>
25. Βιβλιοθήκη Gson : <https://github.com/google/gson>
26. Google Maps API :
<https://developers.google.com/maps/documentation/android-api/>
27. <https://developers.google.com/maps/documentation/android-api/signup#release-cert>
28. SupportMapFragment class :
<https://developer.android.com/reference/com/google/android/gms/maps/SupportMapFragment.html>
29. GoogleMap class :
<https://developers.google.com/android/reference/com/google/android/gms/maps/GoogleMap>
30. AsyncTask class :
<https://developer.android.com/reference/android/os/AsyncTask.html>
31. Gradle Build Dependancies :
<https://developer.android.com/studio/build/build-variants.html#dependencies>
32. Android Support Library :
<https://developer.android.com/topic/libraries/support-library/features.html>
33. RecyclerView class :
<https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>