

Ατομική Διπλωματική Εργασία

**Implementing an E-Commerce
system using a Peer-To-Peer network**

Μαρία Ραγιά

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2016

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Implementing an E-Commerce
system using a Peer-To-Peer network**

Μαρία Ραγιά

Επιβλέπων Καθηγητής

Παρασκευάς Ευριπίδου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2016

Περίληψη

Η διπλωματική μου εργασία σχετίζεται με την ανάπτυξη μιας εφαρμογής διαδικτύου μέσω της οποίας οι χρήστες θα έχουν την δυνατότητα πώλησης αλλά και αγοράς διάφορων προϊόντων. Κάθε χρήστης έχει την δυνατότητα εγγραφής είτε ως χρήστης, είτε ως πωλητής, είτε ως και τα δύο. Επιπλέον η εφαρμογή μου παρέχει την δυνατότητα ανταλλαγής μηνυμάτων μεταξύ των χρηστών.

Η εφαρμογή έχει αναπτυχθεί στο περιβάλλον Android Studio λόγω του ότι προσφέρει μια πολύ καλή διαπροσωπία ανάπτυξης εφαρμογών. Οι δυνατότητες που προσφέρει το Android Studio είναι απεριόριστες αφού είναι η πλατφόρμα που δημιούργησε και υποστηρίζει η Google για την ανάπτυξη εφαρμογών.

Επίσης, για τις απαιτήσεις μιας εφαρμογής διαδικτύου υλοποίησα έναν εξυπηρετητή σε Java με την βοήθεια του γραφικού εργαλείου Eclipse (Eclipse IDE – Integrated Development Environment). Για την διατήρηση των δεδομένων ανέπτυξα μια βάση δεδομένων στην MySQL.

Τέλος, βασικός στόχος της εφαρμογής είναι να γίνουν πιο γρήγορες και εύκολες οι συναλλαγές μεταξύ πωλητή και αγοραστή.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	6
	1.1 Σκοπός	6
	1.2 Δομή εργασίας	7
Κεφάλαιο 2	Τεχνολογίες που χρησιμοποιήθηκαν.....	8
	2.1 Android	8
	2.2 Eclipse	10
	2.3 Android Studio	12
	2.4 MySql	12
	2.5 Peer-to-Peer	13
	2.6 Client-Server	16
	2.7 Java	18
	2.8 JSON	18
Κεφάλαιο 3	Υλοποίηση	19
	3.1 Αρχιτεκτονική εφαρμογής	19
	3.2 Περιγραφή εφαρμογή	20
	3.3 Κατηγορίες Χρηστών	21
	3.4 Λειτουργίες εφαρμογής	25

Κεφάλαιο 4	Βάση δεδομένων και Εξυπηρετητής.....	42
	4.1 Βάση δεδομένων	42
	4.2 Πολυνηματικός εξυπηρετητής	44
	4.3 Inverted Index	46
 Κεφάλαιο 5	 Επέκταση.....	 48
	5.1 Μελλοντική Δουλειά	48
	5.2 Συμπεράσματα	49
 Βιβλιογραφία		 50
 Παράρτημα Α.....		 A-1

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός	6
1.2 Δομή εργασίας	7

1.1 Σκοπός

Ως φυσικό επακόλουθο της γρήγορης ανάπτυξης της τεχνολογίας, ο άνθρωπος εφηύρε πολλές ασύρματες συσκευές όπως τα έξυπνα τηλέφωνα. Πλέον, οι περισσότεροι άνθρωποι έχουν τουλάχιστο μια έξυπνη συσκευή στην κατοχή τους μέσω των οποίων έχουν την δυνατότητα με το πάτημα ενός κουμπιού να πετύχουν τους στόχους τους.

Σήμερα, υπάρχει μεγάλη ποικιλία εφαρμογών ηλεκτρονικών καταστημάτων διαθέσιμα τα οποία είναι φτιαγμένα με αρχιτεκτονική Client-Server όπου ο εξυπηρετητής διαχειρίζεται την συναλλαγή μεταξύ του πωλητή και του αγοραστή.

Η παρούσα διπλωματική εργασία ασχολείται με την ανάπτυξη μίας εφαρμογής διαδικτύου που θα δίνει την δυνατότητα σε κάθε χρήστη να αγοράζει ή να πουλάει προϊόντα σε γρήγορο και εύκολο χρόνο. Για την λειτουργία της συναλλαγής υλοποιήθηκε αρχιτεκτονική Peer-to-Peer όπου οι δύο χρήστες, ο πωλητής και ο αγοραστής διενεργούν μεταξύ τους την συναλλαγή. Ουσιαστικά ένα Peer-to-Peer δίκτυο αποτελεί μια πιο φτηνή λύση. Για τις υπόλοιπες λειτουργίες έχει υλοποιηθεί ένας πολυνηματικός εξυπηρετητής σε Java στο γραφικό περιβάλλον (Eclipse IDE – Integrated Development Environment). Για την διατήρηση των δεδομένων του κάθε χρήστη έχω δημιουργήσει μια βάση δεδομένων στην MySQL.

Ο τομέας της ανάπτυξης εφαρμογών έχει επεκταθεί σε πολύ μεγάλο βαθμό τα τελευταία χρόνια και είναι ένας από τους πιο ενδιαφέροντες τομείς, προσφέροντας

πολλές προοπτικές εργοδότησης. Πλέον, έχω αποφασίσει ότι μετά την ολοκλήρωση των σπουδών θα ήθελα να ασχοληθώ περεταίρω με την ανάπτυξη εφαρμογών.

1.2 Δομή εργασίας

Σε αυτό το σημείο θα περιγράψω την δομή της διπλωματικής μου εργασίας. Αρχικά, γίνεται μια περιγραφή του τι έχω υλοποιήσει στα πλαίσια της διπλωματικής μου εργασίας.

Στο πρώτο κεφάλαιο γίνεται η εισαγωγή όπου περιγράφω το σκοπό και τα κίνητρα που με οδήγησαν στην επιλογή αυτής της εργασίας.

Στο δεύτερο κεφάλαιο επεξηγώ το λογισμικό Android και τα εργαλεία που χρησιμοποίησα στην ανάπτυξη της εφαρμογής τα οποία είναι το Android Studio, Eclipse, MySQL.

Στο τρίτο κεφάλαιο γίνεται επεξήγηση των αρχιτεκτονικών Peer-to-Peer και Client-Server. Στην συνέχεια, γίνεται η περιγραφή της εφαρμογής που ανέπτυξα.

Στο τέταρτο κεφάλαιο γίνεται η περιγραφή του Server που υλοποιήθηκε σε Java και η ανάλυση της βάσης δεδομένων που υλοποιήθηκε σε MySQL.

Τέλος, στο πέμπτο και τελευταίο κεφάλαιο αναφέρονται τα συμπεράσματα και οι τομείς που μπορεί να γίνει βελτίωση της εφαρμογής.

Κεφάλαιο 2

Τεχνολογίες που χρησιμοποιήθηκαν

2.1 Android	8
2.2 Eclipse	10
2.3 Android Studio	12
2.4 MySQL	12
2.5 Peer-to-Peer	13
2.6 Client-Server	16
2.7 Java	18
2.8 JSON	18

2.1 Android

Το Android είναι ίσως το πιο διαδεδομένο λειτουργικό σύστημα το οποίο έχει δημιουργηθεί από την Google. Είναι βασισμένο στον πυρήνα του Linux και έχει σχεδιαστεί κυρίως για έξυπνες συσκευές όπως τα κινητά τηλέφωνα και τα tablet με οθόνες αφής. Η αλληλεπίδραση μεταξύ χρηστών και λογισμικού γίνεται με την επαφή του χεριού του χρήστη με την οθόνη της έξυπνης συσκευής. Ανάλογα με την περιοχή που γίνεται η επαφή το λογισμικό εκτελεί διαφορετική λειτουργία.



Σχήμα 2.1 Το λογότυπο του λογισμικού Android.

Αρχικά, αναπτύχθηκε από την εταιρία Android Inc., η οποία εξαγοράστηκε από την Google το 2005. Το Android παρουσιάστηκε για πρώτη φορά στο κοινό το 2007 μαζί με την ίδρυση της «κοινοπραξίας» εταιρειών λογισμικού, υλικού και κινητής τηλεφωνίας «Open Handset Alliance 3» με σκοπό την προώθηση ανοιχτών λογισμικών για κινητά τηλέφωνα. Κυριότερος ανταγωνιστής του λογισμικού Android είναι το λογισμικό IOS της Apple.

Το Android είναι λογισμικό ανοιχτού κώδικα, αν και οι περισσότερες συσκευές Android έχουν ένα συνδυασμό λογισμικού ανοιχτού κώδικα με ιδιόκτητο κώδικα. Το γεγονός ότι αποτελεί λογισμικό ανοιχτού κώδικα έχει ενθαρρύνει ένα μεγάλο αριθμό προγραμματιστών να χρησιμοποιούν τον ανοιχτό κώδικα ως βάση για την ανάπτυξη δικών τους εφαρμογών, όπου στην συνέχεια τις μοιράζονται με το κοινό.

Οι εφαρμογές που υλοποιούνται για το Android είναι γραμμένες στη γλώσσα προγραμματισμού Java. Το κύριο γραφικό περιβάλλον που ανέπτυξε η Google για δημιουργία εφαρμογών είναι το Android Studio το οποίο έχει ενσωματωμένο το Android software development kit (SDK). Υπάρχουν δύο τρόποι για εγκατάσταση των εφαρμογών στο Android. Η εγκατάσταση μπορεί να γίνει απευθείας με την λήψη και την εγκατάσταση του APK (Android Application Package) αρχείου. Ο δεύτερος τρόπος είναι με τη λήψη τους από ένα κατάστημα εφαρμογών. Το βασικό κατάστημα εφαρμογών στο Android είναι το Google Play Store το οποίο τον Ιούλιο του 2013 ξεπέρασε το ένα εκατομμύριο διαθέσιμες εφαρμογές.

Οι εφαρμογές σε Android εκτελούνται σε μια εικονική μηχανή Dalvik στις συσκευές Android, αφού προηγουμένως έχουν μετατραπεί από κώδικα Java σε bytecode και αργότερα σε εκτελέσιμα αρχεία Dalvik. Από την έκδοση Android 5.0 χρησιμοποιείται το Android Runtime (RTA) για να μεταγλωττίσει τα bytecode της εφαρμογής σε κώδικα μηχανής.

Κάθε έκδοση του Android έχει πάρει το όνομα της από ένα γλυκό εκτός από τις δύο πρώτες που ονομάστηκαν Alpha και Beta αντίστοιχα:

- Android 1.6. Donut.
- Android 2.1. Eclair.
- Android 2.2. Froyo.
- Android 2.3. Gingerbread.
- Android 3.0. Honeycomb.
- Android 4.0. Ice Cream Sandwich.
- Android 4.1. Jelly Bean.
- Android 4.4. KitKat.
- Android 5.0 Lollipop
- Android 6.0 Marshmallow

Ένα άλλο ενδιαφέρον χαρακτηριστικό στα ονόματα των εκδόσεων είναι ότι για κάθε έκδοση το πρώτο γράμμα του ονόματος της ακολουθεί την σειρά του Αγγλικού αλφαβήτου.



Σχήμα 2.2 Το λογότυπο της τελευταίας έκδοσης του Android, το Android 6.0 Marshmallow.

Το 2014 η Google ανακοινώσε πως υπάρχουν πάνω από 1 δισεκατομμύριο διαφορετικές ενεργές συσκευές Android σε περίοδο ενός μηνός. Με βάση αυτά τα στατιστικά αυξάνεται ραγδαία ο αριθμός των προγραμματιστών που αποκλειστικά δημιουργούν εφαρμογές για Android συσκευές.

2.2 Eclipse

Το γραφικό περιβάλλον Eclipse είναι ένα περιβάλλον ανάπτυξης προγραμμάτων. Περιέχει μια βασική διπροσωπία για ανάπτυξη προγραμμάτων και ένα επεκτάσιμο σύστημα από plug-ins. Το Eclipse είναι ως επί το πλείστον γραμμένο σε Java και χρησιμοποιείται κυρίως για την ανάπτυξη προγραμμάτων σε Java, αλλά μέσω του

πλούσιου συστήματος από plug-ins που διαθέτει είναι δυνατή η ανάπτυξη προγραμμάτων και σε άλλες γλώσσες προγραμματισμού.



Σχήμα 2.3 Το λογότυπο του γραφικού περιβάλλον Eclipse.

Ο αρχικός πηγαίος κώδικας του Eclipse προήλθε από την εταιρία IBM VisualAge. Το 2004 δημιουργήθηκε το Eclipse Foundation το οποίο ανέλαβε την ανάπτυξη του. Την ίδια χρονιά ανακοινώθηκε και η πρώτη έκδοση του, το Eclipse 3.0. Μέχρι στιγμής έχουν κυκλοφορήσει οι εξής εκδόσεις:

- Callisto
- Europa
- Ganymede
- Galileo
- Helios
- Indigo
- Juno
- Kepler
- Luna
- Mars
- Neon
- Oxygen

Το software development kit (SDK) του Eclipse, είναι λογισμικό ανοιχτού κώδικα και αρχικά προοριζόταν μόνο για προγραμματιστές της Java, όμως οι χρήστες μπορούν να επεκτείνουν τις δυνατότητες του Eclipse με την εγκατάσταση plug-ins που είναι γραμμένα για αυτό και να το χρησιμοποιήσουν για ανάπτυξη προγραμμάτων και σε άλλες γλώσσες προγραμματισμού. Επίσης, μπορούν να γράψουν τα δικά τους plug-ins και να τα ενσωματώσουν.

2.3 Android Studio

Το Android Studio είναι το επίσημο εργαλείο για ανάπτυξη εφαρμογών (IDE) για τις πλατφόρμες που χρησιμοποιούν Android. Δημιουργήθηκε από την Google για να αντικαταστήσει το Eclipse Android Development Tools (ADT) ως το βασικό περιβάλλον για ανάπτυξη εφαρμογών Android. Η πρώτη σταθερή έκδοση κυκλοφόρησε

το 2014 και είναι συμβατό και με τα Windows και με το Mac OS X, αλλά και με τα Linux.



Σχήμα 2.4 Το λογότυπο του περιβάλλοντος Android Studio.

2.4 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων ανοικτού κώδικα. Το 2013 ήταν το δεύτερο σε χρήση σύστημα για διαχείριση βάσεων δεδομένων ανά το παγκόσμιο και το πρώτο σε χρήση σύστημα ανοιχτού κώδικα.

Αναπτύχθηκε από την Σουηδική εταιρία MySQL AB η οποία στη συνέχεια αγοράστηκε από την εταιρία Oracle.

Η MySQL είναι η πιο δημοφιλής επιλογή στην κοινότητα των προγραμματιστών για χρήση σε διαδικτυακές εφαρμογές και είναι βασικό κομμάτι του ευρέως χρησιμοποιημένου λογισμικού LAMP (Linux Apache MySQL Php) Stack. Πολλές μεγάλες εταιρίες χρησιμοποιούν την MySQL ως βάση δεδομένων για τις ιστοσελίδες και τις εφαρμογές τους όπως η Google, το Facebook, το YouTube και το Twitter.



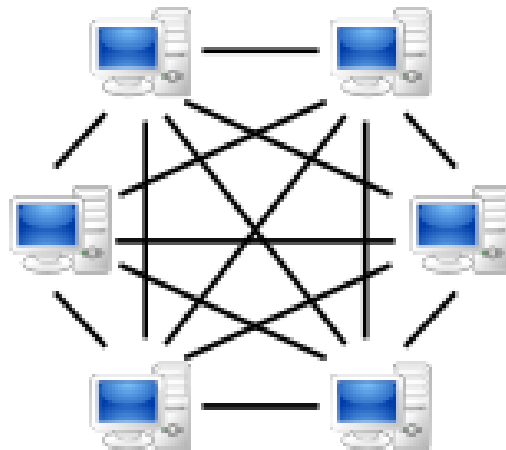
Σχήμα 2.5 Το λογότυπο του συστήματος διαχείρισης βάσεων δεδομένων MySQL.

Η MySQL είναι γραμμένη σε γλώσσες προγραμματισμού C και C++. Προσφέρει ποικίλες δυνατότητες όπως την υποστήριξη μεταξύ διαφορετικών πλατφόρμων, την

αποθήκευση διαδικασιών, triggers, cursors, ανανεώσιμα views, Information Schema, Query caching, υποστήριξη Unicode χαρακτήρων.

2.5 Peer-to-Peer

Βασική λειτουργία της αρχιτεκτονικής Peer-to-Peer είναι να διαμοιράζει τις εργασίες που πρέπει να εκτελεστούν σε ένα δίκτυο μεταξύ των peers. Οι peers έχουν τα ίδια δικαιώματα ο ένας με τον άλλο και συμμετέχουν ισοδύναμα στην εφαρμογή.



Σχήμα 3.1 Μια απλή απεικόνιση ενός δικτύου Peer-to-Peer

Κάθε peer που είναι μέλος του δικτύου παρέχει ένα μέρος από τους πόρους του, όπως επεξεργαστική ισχύ, χωρητικότητα δίσκου, bandwidth, αρχεία κ.α. σε άλλους συμμετέχοντες του δικτύου για να τους χρησιμοποιήσουν με σκοπό να πετύχουν τον στόχο τους. Η απόφαση για το πώς θα διανεμηθούν σωστά και με δίκαιο τρόπο οι πόροι γίνεται μεταξύ των peers και όχι με την χρήση κάποιου κεντρικού εξυπηρετητή.

Στην αρχιτεκτονική Peer-to-Peer κάθε peer μπορεί να ζητήσει πόρους από άλλους peers αλλά και να διαθέσει τους δικούς του πόρους σε άλλους peers. Όλους τους πόρους που θα δεσμεύσει έχει την δυνατότητα να τους εκμεταλλευτεί στο εκατόν της εκατόν για πετύχει τον σκοπό του, σε αντίθεση με την αρχιτεκτονική Client-Server όπου ο κεντρικός εξυπηρετητής διαμοιράζει τους πόρους που έχει στην διάθεση του σε όλους τους clients.

Αρχικά, κάθε peer όταν ήθελε να κατεβάσει ένα αρχείο ενωνόταν με κάποιον άλλο peer που είχε το αρχείο στην διάθεση του και το κατέβαζε εξ ολοκλήρου από τον συγκεκριμένο peer. Αυτό είχε ως αποτέλεσμα η ταχύτητα με την οποία κατέβαινε το αρχείο να εξαρτιόταν αποκλειστικά από το bandwidth της σύνδεσης του peer που

διέθετε το αρχείο. Για να επιλυθεί αυτό το πρόβλημα και να μειωθεί ο χρόνος που χρειάζεται για να κατεβεί ένα αρχείο οι εφαρμογές που χρησιμοποιούν Peer-to-Peer αρχιτεκτονική επιτρέπουν στον peer που θέλει να κατεβάσει ένα αρχείο να συνδεθεί όχι μόνο με ένα αλλά με πολλούς peer, οι οποίοι διαθέτουν τον συγκεκριμένο αρχείο και να κατεβάσει κομμάτια του αρχείου από διαφορετικούς peers. Άρα, όσο περισσότεροι peers είναι διαθέσιμοι τόσο μικρότερο χρόνο χρειάζεται για να κατεβεί ένα αρχείο.

Αν και αρκετές εφαρμογές χρησιμοποιούσαν την αρχιτεκτονική Peer-to-Peer, η πρώτη εφαρμογή που ήταν ευρέως διαθέσιμη ήταν το Napster το 1999. Το Napster ήταν μια εφαρμογή μέσω της οποίας οι χρήστες τις μπορούσαν να διαμοιράσουν αρχεία μουσικής. Κάθε χρήστης ανέβαζε τα δικά του αρχεία μουσικής τα οποία γίνονταν αμέσως διαθέσιμα στους άλλους χρήστες για να τα κατεβάσουν. Για να κατεβάσει κάποιος ένα αρχείο, ενωνόταν απευθείας με τον χρήστη που το είχε και το κατέβαζε. Στην συνέχεια αναπτύχθηκαν πολλές παρόμοιες εφαρμογές με το Napster. Μια από αυτές προκάλεσε επανάσταση στον τρόπο που διαμοιράζονταν τα αρχεία μεταξύ των χρηστών, το BitTorrent. Το BitTorrent ήταν μια εφαρμογή διαμοιρασμού αρχείων torrent με τον εξής τρόπο, ο χρήστης που ήθελε να κατεβάσει ένα αρχείο ενωνόταν με πολλούς χρήστες που διέθεταν το αρχείο που ήθελε και κατέβαζε κομμάτια του από κάθε χρήστη με αποτέλεσμα να μειώνεται ο συνολικός χρόνος που χρειαζόταν για να κατεβεί ένα αρχείο αλλά και σε περίπτωση που η διακοπτόταν η σύνδεση με ένα χρήστη τότε κάποιος άλλος που είχε το αρχείο έπαιρνε την θέση του.



Σχήμα 3.2 Το λογότυπο της εφαρμογής BitTorrent η οποία χρησιμοποιούσε αρχιτεκτονική Peer-to-Peer για την υλοποίηση της.

Η χρήση της αρχιτεκτονικής Peer-to-Peer σε εφαρμογές έχει τα εξής πλεονεκτήματα:

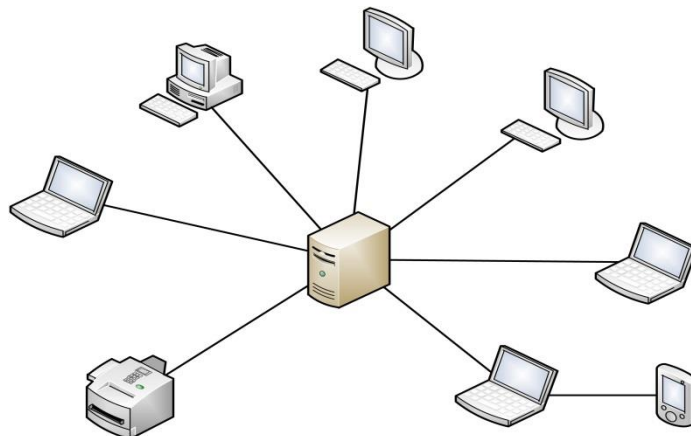
- Δεν χρειάζεται κεντρικό εξυπηρετητή για να οργανωθεί και να λειτουργήσει ένα δίκτυο.
- Είναι πιο εύκολο και πιο φτηνό το στήσιμο ενός δικτύου Peer-to-Peer παρά ενός δικτύου Client-Server, λόγω του ότι δεν χρειάζεται η αγορά και εγκατάσταση ενός ακριβού εξυπηρετητή.
- Όταν ένας peer διακόψει την σύνδεση του με το δίκτυο, τότε το δίκτυο θα συνεχίσει να λειτουργεί κανονικά απλά τα αρχεία που είχε ο peer που αποσυνδέθηκε δεν θα είναι διαθέσιμα πλέον στους άλλους peer.

Πέραν των πλεονεκτημάτων η χρήση της αρχιτεκτονικής Peer-to-Peer έχει και μειονεκτήματα όπως:

- Οι ηλεκτρονικοί υπολογιστές στους οποίους θα έχουν πρόσβαση άλλοι, για να κατεβάσουν κάποιο αρχείο, θα έχουν μειωμένη απόδοση στις άλλες λειτουργίες που θα κάνει ο ιδιοκτήτης τους.
- Τα αρχεία είναι διαθέσιμα κατά την διάρκεια που ο κάτοχος τους είναι ενωμένος με το δίκτυο.
- Προκύπτουν θέματα ασφαλείας αφού κάθε χρήστης είναι υπεύθυνος για να μην ανεβάσει αρχεία μολυσμένα με ιούς αλλά και για το ποια δικαιώματα έχουν οι χρήστες που είναι ενωμένοι μαζί του.

2.6 Client-Server

Η αρχιτεκτονική Client-Server είναι ένα μοντέλο που χρησιμοποιείται για την ανάπτυξη εφαρμογών. Σκοπός της αρχιτεκτονικής Client-Server είναι ο διαμοιρασμός των πόρων μεταξύ ενός εξυπηρετητή και των χρηστών οι οποίοι λειτουργούν ως πελάτες και αιτούνται κάποιον πόρο, όπως αρχεία, επεξεργαστική ισχύ, χωρητικότητα δίσκου, bandwidth κ.α..



Σχήμα 3.3 Μια απλή απεικόνιση ενός δικτύου Client-Server.

Σε ένα δίκτυο Client-Server εγκαθιστάτε ένας εξυπηρετητής πάνω στον οποίο ενώνονται πολλοί χρήστες, clients. Οι χρήστες δεν γνωρίζουν ποιοι άλλοι χρήστες

βρίσκονται στο δίκτυο και είναι ενωμένοι με τον εξυπηρετητή. Όταν ένας χρήστης θέλει να χρησιμοποιήσει ένα πόρο που έχει ο εξυπηρετητής, τότε πρέπει να ξεκινήσει την διαδικασία δημιουργίας σύνδεσης είτε μέσω local area network (LAN), είτε μέσω wide-area network (WAN). Αρχικά, ο χρήστης αποστέλλει αίτημα για δημιουργία σύνδεσης και ο εξυπηρετητής του απαντά ανάλογα αν είναι δυνατή η δημιουργία σύνδεσης ή όχι. Αν η απάντηση του εξυπηρετητή είναι καταφατική τότε δημιουργείται η σύνδεση μεταξύ πελάτη και εξυπηρετητή. Στην συνέχεια, ο χρήστης αποστέλλει αίτημα για τον πόρο που θέλει στον εξυπηρετητή, ο οποίος του δίνει τον πόρο και τερματίζει την σύνδεση. Ο χρήστης έχει την δυνατότητα να ανοίξει πολλές παράλληλες συνδέσεις με τον εξυπηρετητή με αποτέλεσμα να μπορεί να πάρει πολλούς πόρους ταυτόχρονα, άρα μειώνει τον χρόνο που χρειάζεται για να περιμένει. Από την πλευρά του ο εξυπηρετητής είναι υπεύθυνος για να διαμοιράζει σωστά και δίκαια τους διαθέσιμους πόρους μεταξύ των χρηστών.

Ο ίδιος ο Παγκόσμιος Ιστός είναι υλοποιημένος με το μοντέλο Client-Server, όπου κάθε χρήστης που θέλει να έχει πρόσβαση σε μια ιστοσελίδα χρησιμοποιεί ένα πλοηγό, ο οποίος είναι ο πελάτης. Όταν δώσει την διεύθυνση της σελίδας που θέλει να δει στην οθόνη του τότε ο πλοηγός στέλνει ένα αίτημα στον εξυπηρετητή που φιλοξενεί την συγκεκριμένη σελίδα για δημιουργία σύνδεσης μέσω του πρωτοκόλλου επικοινωνίας HTTP. Στην συνέχεια, δημιουργείται η σύνδεση και ο εξυπηρετητής αποστέλλει την σελίδα στον πλοηγό ο οποίος την παρουσιάζει στον χρήστη. Άλλα προγράμματα που χρησιμοποιούν το μοντέλο Client-Server είναι το e-mail, το πρωτόκολλο Ftp κ.α..



Σχήμα 3.4 Η αρχιτεκτονική που χρησιμοποιεί ο Παγκόσμιος Ιστός είναι Client-Server.

Ως επέκταση ενός απλού δικτύου με αρχιτεκτονική Client-Server ο εξυπηρετητής μπορεί να συνδέσει το δίκτυο με κάποιο άλλο δίκτυο. Όπως για παράδειγμα, όταν ένας χρήστης ελέγχει από τον υπολογιστή του τον λογαριασμό του σε μια τράπεζα, το πρόγραμμα πελάτη στον υπολογιστή του αποστέλλει αίτημα στον κεντρικό

εξυπηρετητή της τράπεζας. Μετά, ο κεντρικός εξυπηρετητής της τράπεζας μεταφέρει το αίτημα του στον εξυπηρετητή του υποκαταστήματος στο οποίο είναι ο λογαριασμός και στην συνέχεια ο εξυπηρετητής του υποκαταστήματος απαντά στον κεντρικό εξυπηρετητή και αυτός με την σειρά του επιστρέφει στο πρόγραμμα πελάτη τον λογαριασμό του.

Στην εφαρμογή μου χρησιμοποιήθηκε η αρχιτεκτονική Client-Server για τις εξής λειτουργίες:

- Login
- Register
- Add Product
- Search
- Cart
- Insert Online User
- Update User
- Update Product
- Delete Product

Η αρχιτεκτονική Client-Server η πιο διαδεδομένη για την ανάπτυξη εφαρμογών. Κύρια πλεονεκτήματα της είναι:

- Όλα τα αρχεία είναι αποθηκευμένα σε ένα κεντρικό επεξεργαστή.
- Η ασφάλεια του δικτύου ελέγχεται από τον κεντρικό εξυπηρετητή.
- Οι χρήστες μπορούν οποιαδήποτε στιγμή να έχουν πρόσβαση στους πόρους που βρίσκονται στον εξυπηρετητή.

Εκτός από τα πλεονεκτήματα παρουσιάζει και κάποια μειονεκτήματα όπως:

- Οι εξυπηρετητές είναι ακριβοί.
- Πρέπει να υπάρχει κάποιος υπεύθυνος που θα διαχειρίζεται τον εξυπηρετητή.
- Αν υπάρξει κάποιο πρόβλημα με τον εξυπηρετητή τότε το δίκτυο θα σταματήσει να λειτουργεί.

2.7 Java

Η γλώσσα προγραμματισμού που επιλέχθηκε για την ανάπτυξη του εξυπηρετητή είναι η αντικειμενοστρεφής γλώσσα Java. Είναι σχεδιασμένη να έχει όσο το δυνατό λιγότερες εξαρτήσεις υλοποίησης γίνεται. Σκοπός του είναι να δώσει την δυνατότητα στους προγραμματιστές να γράψουν ένα πρόγραμμα το οποίο θα τρέχει παντού, το οποίο

σημαίνει ότι ο μεταγλωττισμένος κώδικας Java μπορεί να τρέξει σε όλες τις πλατφόρμες που έχουν εγκατεστημένη την Java. Η εφαρμογές Java μεταγλωττίζονται σε bytecode που μπορεί να τρέξει σε όλες τις Java Virtual Machines (JVM) άσχετα από την αρχιτεκτονική του υπολογιστή.

Από το 2016 η Java είναι η δημοφιλέστερη γλώσσα προγραμματισμού παγκοσμίως και κυρίως για Client-Server εφαρμογές. Η Java αρχικά αναπτύχθηκε από τον James Gosling στην εταιρία Sun Microsystems και κυκλοφόρησε για πρώτη φορά το 1995. Η τελευταία έκδοση της, Java 8, κυκλοφόρησε από την εταιρία Oracle και είναι ανοικτού κώδικα.

2.8 JSON

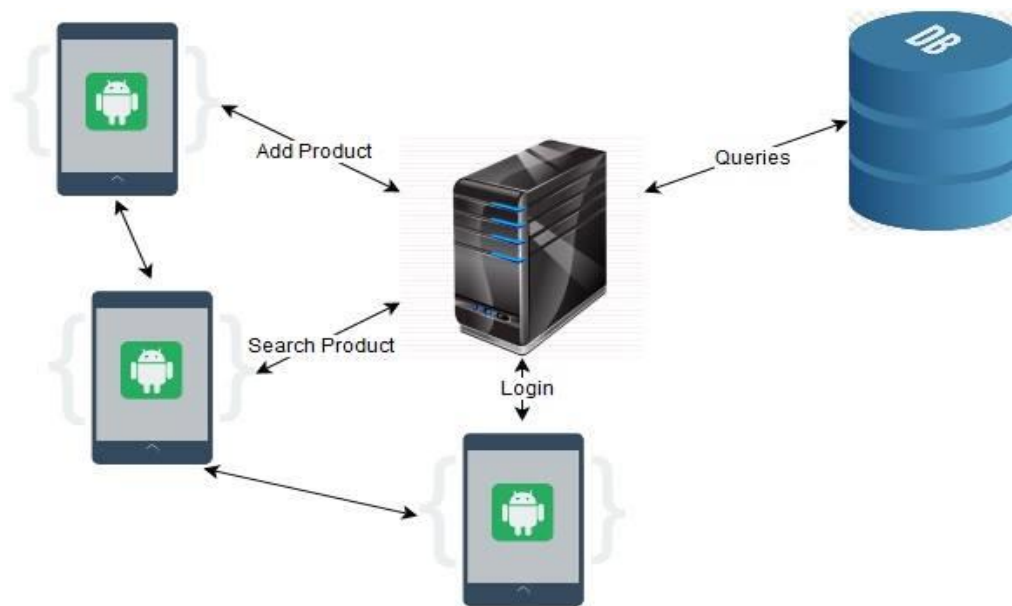
Τόσο τα αιτήματα που λαμβάνει ο εξυπηρετητής, όσο και οι απαντήσεις που στέλλει είναι κωδικοποιημένες σε αντικείμενα τύπου JSON. Αυτό επιτυγχάνεται με την βοήθεια της εξωτερικής βιβλιοθήκης gson-2.3.1.jar. Τα JSON αντικείμενα είναι μια αναγνωρίσιμη μορφή δεδομένων οι οποία χρησιμοποιείται κυρίως για την μετάδοση δεδομένων μεταξύ εφαρμογών διαδικτύου και εξυπηρετητών. Η μορφή που έχουν τα αντικείμενα JSON είναι Key/Value Pair δηλαδή ένα ζευγάρι τιμών όπου το Key είναι το όνομα του πεδίου θέλουμε να κωδικοποιήσουμε και το Value είναι η τιμή που έχει το συγκεκριμένο πεδίο.

Κεφάλαιο 3

Υλοποίηση

3.1 Αρχιτεκτονική Εφαρμογής	19
3.2 Περιγραφή Εφαρμογής	20
3.3 Κατηγορίες Χρηστών	21
3.4 Περιγραφή Λειτουργιών	25

3.1 Αρχιτεκτονική εφαρμογής



Σχήμα 3.1 Η απεικόνιση της αρχιτεκτονικής της εφαρμογής που ανέπτυξα.

Όπως παρατηρούμε στο σχήμα 3.1 οι χρήστες θα κατεβάζουν την εφαρμογή πάνω στο smartphone ή tablet τους. Οι ανταλλαγή μηνυμάτων και η αγορά των προϊόντων έγινε με την υλοποίηση ενός μοντέλου Peer-to-Peer όπου κάθε χρήστης που θέλει να αγοράσει κάτι θα ενώνεται απευθείας με τον πωλητή και θα γίνεται η συναλλαγή. Όλες οι υπόλοιπες λειτουργίες όπως η είσοδος, η εγγραφή, η αναζήτηση προϊόντων, το καλάθι κ.α. θα γίνονται με την βοήθεια του εξυπηρετητή όπου κάθε εφαρμογή του χρήστη θα στέλνει ένα κατάλληλα κωδικοποιημένο αίτημα στον εξυπηρετητή και αυτός με την σειρά του θα έχει πρόσβαση στην βάση δεδομένων και θα επιστρέφει τα δεδομένα στην εφαρμογή του χρήστη.

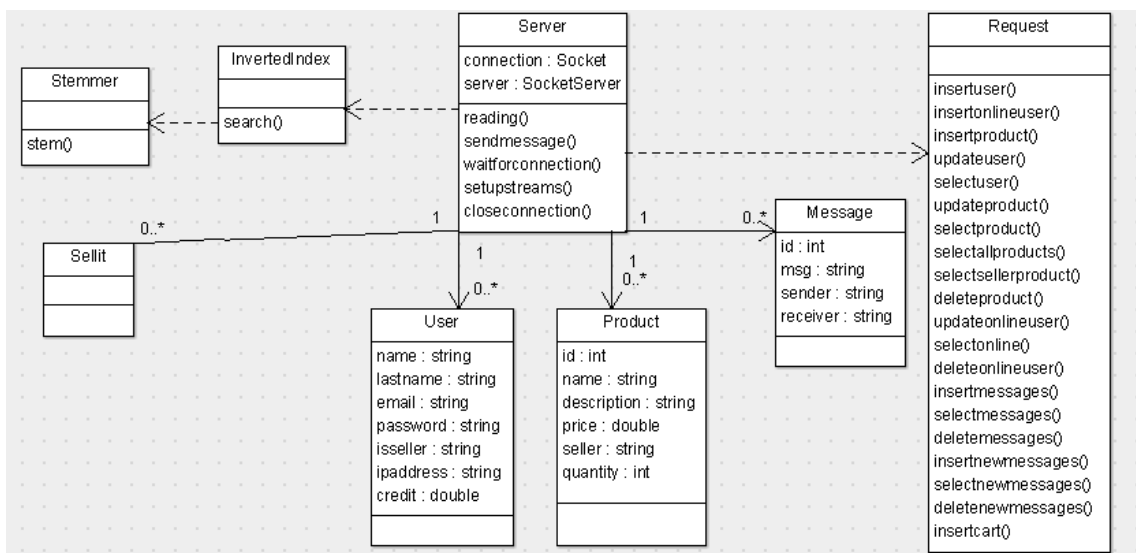
3.2 Περιγραφή εφαρμογής

Η εφαρμογή που ανέπτυξα είναι ένα ηλεκτρονικό κατάστημα πώλησης προϊόντων. Το ηλεκτρονικό κατάστημα θα είναι ελεύθερο για πρόσβαση από όλους τους ενδιαφερόμενους κατεβάζοντας και εγκαθιστώντας την εφαρμογή είτε στο smartphone είτε στο tablet τους. Οι διάφοροι χρήστες του συστήματος αποτελούν τους πελάτες του μοντέλου αλλά και τους εξυπηρετητές της ιστοσελίδας αφού έχουν την δυνατότητα να είναι και πελάτες αλλά και πωλητές και να διαθέτουν τα προϊόντα τους στην εφαρμογή.

Συγκεκριμένα η όψη που θα παρέχεται στον χρήστη θα είναι συγκεκριμένη μέσω της διεπαφής, η οποία θα αλληλοεπιδρά με αυτόν. Ο χρήστης έχει τη δυνατότητα πρόσβασης στις πληροφορίες οι οποίες είναι επιτρεπτές από το σύστημα, ενώ δύναται, σε διάφορα χρονικά διαστήματα, να ενημερώνεται άμεσα από την βάση δεδομένων για καινούρια στοιχεία, τα οποία ενσωματώνονται στο σύστημα και αφορούν τον ίδιο.

Αναλυτικά, για διαδικασίες του χρήστη όπως η προβολή προϊόντων μέσω αναζήτησης, η αγορά προϊόντων, η τροποποίηση καλαθιού αγορών, η επικύρωση αγορών, η δημιουργία λογαριασμού και άλλες, που θα εξηγηθούν αναλυτικότερα πιο κάτω, που αποτελούν γεγονότα από τον χρήστη θα υπάρχει ενημέρωση σε αυτόν μέσω του εξυπηρετητή και της βάση δεδομένων.

Η βάση δεδομένων του συστήματος δύναται να έχει όλες τις δυνατές πληροφορίες που αφορούν τους πελάτες, τις συναλλαγές τους, τις ενέργειες από τους διαχειριστές του συστήματος και κυρίως στοιχεία που αφορούν τα προϊόντα που βρίσκονται για πώληση προς τους πελάτες-ενδιαφερομένους του ηλεκτρονικού καταστήματος.



Σχήμα 3.2 Το UML Class Diagram των κλάσεων που χρησιμοποιούνται και αλληλοεπιδρούν με την εφαρμογή.

3.3 Κατηγορίες Χρηστών

Η διπλωματική μου θα αποτελεί ένα Android εφαρμογή μέσω της οποίας θα υποστηρίζονται τρεις κατηγορίες χρηστών. Κατά την εγγραφή του χρήστη στο

σύστημα θα ζητείται το email του, το όνομα και το επίθετό του, η κατηγορία του αλλά και ένας κωδικός πρόσβασης. Ο κάθε χρήστης θα αποκτά πρόσβαση πληκτρολογώντας το email του αλλά και ένα κωδικό πρόσβασης.

Κατηγορία 1 - Πωλητής (Seller)

- Θα καταγράφει τις πληροφορίες του όπως:

- Όνομα
- Επίθετο
- email
- Κωδικός πρόσβασης
- Κατηγορία (Πωλητής)
- Ip Address
- Πίστωση

- Θα καταγράφει τα αντικείμενα που θα πωλεί:

- Όνομα αντικειμένου
- Ποσότητα
- Τιμή
- Γενική περιγραφή που θα περιλαμβάνει γενικές πληροφορίες για το προϊόν όπως διαθέσιμα μεγέθη, διαθέσιμα χρώματα, μέρες παράδοσης και άλλα.
- Φωτογραφία

- Θα έχει την δυνατότητα αναζήτησης των δικών του προϊόντων για ανανέωση των πληροφοριών και διαγραφή των προϊόντων.

- Μετά την εγγραφή του ο χρήστης θα έχει την δυνατότητα να αλλάξει τις πληροφορίες που έδωσε αν επιθυμεί.

-Θα μπορεί να συνομιλεί με τους πελάτες για να τους λύσει τυχόν απορίες ή προβλήματα που θα έχουν με τα προϊόντα τους.

Κατηγορία 2 - Πελάτης (Customer)

- Θα καταγράφει τις πληροφορίες του όπως:

- Όνομα
- Επίθετο
- email
- Κωδικός πρόσβασης
- Κατηγορία (Πελάτης)
- Ip Address
- Πίστωση

- Θα μπορεί να κάνει αναζήτηση για τα πράγματα που αγοράζει μέσω φιλτραρίσματος των λέξεων που υπάρχουν στο όνομα και την περιγραφή των προϊόντων. Τα προϊόντα που θα επιλέγει να δει θα παρουσιάζουν τα εξής χαρακτηριστικά:

- Όνομα αντικειμένου
- Ποσότητα
- Τιμή
- Γενική περιγραφή που θα περιλαμβάνει γενικές πληροφορίες για το προϊόν όπως διαθέσιμα μεγέθη, διαθέσιμα χρώματα, μέρες παράδοσης και άλλα.
- Φωτογραφία

- Θα περιέχει καλάθι στο οποίο θα φυλάσσονται τα αντικείμενα που θέλει να αγοράσει ο πελάτης.

- Μετά την εγγραφή του ο χρήστης θα έχει την δυνατότητα να αλλάξει τις πληροφορίες που έδωσε αν επιθυμεί.

-Θα μπορεί να συνομιλεί με τους πωλητές για να του λύσουν τυχόν απορίες ή προβλήματα που θα έχει με τα προϊόντα τους.

Κατηγορία 3 – Πωλητής και Πελάτης (Seller and Customer)

- Θα καταγράφει τις πληροφορίες του όπως:

- Όνομα
- Επίθετο
- email
- Κωδικός πρόσβασης

- Κατηγορία (Πελάτης)
- Ip Address
- Πίστωση

Θα καταγράφει τα αντικείμενα που θα πωλεί:

- Όνομα αντικειμένου
- Ποσότητα
- Τιμή
- Γενική περιγραφή που θα περιλαμβάνει γενικές πληροφορίες για το προϊόν όπως διαθέσιμα μεγέθη, διαθέσιμα χρώματα, μέρες παράδοσης και άλλα.
- Φωτογραφία

- Θα έχει την δυνατότητα αναζήτησης των δικών του προϊόντων για ανανέωση των πληροφοριών και διαγραφή των προϊόντων.

- Μετά την εγγραφή του ο χρήστης θα έχει την δυνατότητα να αλλάξει τις πληροφορίες που έδωσε αν επιθυμεί.

-Θα μπορεί να συνομιλεί με τους πελάτες για να τους λύσει τυχόν απορίες ή προβλήματα που θα έχουν με τα προϊόντα τους.

- Θα μπορεί να κάνει αναζήτηση για τα πράγματα που αγοράζει μέσω φιλτραρίσματος των λέξεων που υπάρχουν στο όνομα και την περιγραφή των προϊόντων. Τα προϊόντα που θα επιλέγει για να δει του παρουσιάζουν τα εξής χαρακτηριστικά:

- Όνομα αντικειμένου
- Ποσότητα
- Τιμή
- Γενική περιγραφή που θα περιλαμβάνει γενικές πληροφορίες για το προϊόν όπως διαθέσιμα μεγέθη, διαθέσιμα χρώματα, μέρες παράδοσης και άλλα.
- Φωτογραφία

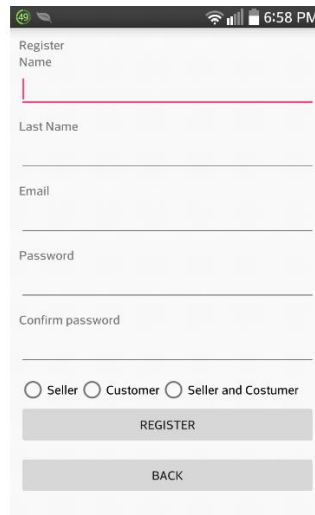
- Θα περιέχει καλάθι στο οποίο θα φυλάσσονται τα αντικείμενα που θέλει να αγοράσει ο πελάτης.

-Θα μπορεί να συνομιλεί με τους πωλητές για να του λύσουν τυχόν απορίες ή προβλήματα που θα έχει με τα προϊόντα τους.

3.3 Περιγραφή Λειτουργιών

1.Εγγραφή καινούριου μέλους στο ηλεκτρονικό κατάστημα

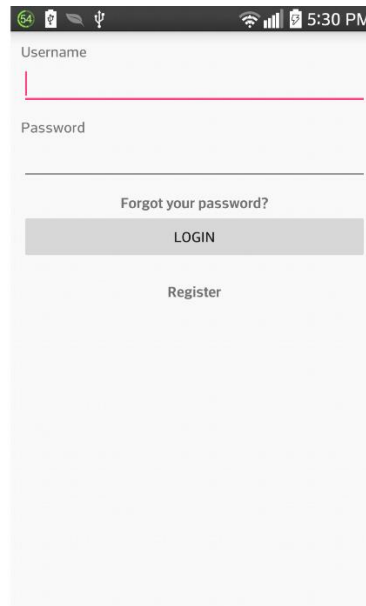
Η εγγραφή καινούριου μέλους στο ηλεκτρονικό κατάστημα αποτελεί μια από τις βασικότερες λειτουργίες τις οποίες θα υποστηρίζει το σύστημα. Συγκεκριμένα πρέπει να προηγηθεί η είσοδος κάποιου ενδιαφερόμενου στο σύστημα για να μπορέσει να καταχωρήσει προϊόντα, να αγοράσει προϊόντα και να εκτελέσει τις διάφορες συναλλαγές του με το σύστημα. Για να γίνει αυτό θα πρέπει να δημιουργήσει ένα προσωπικό λογαριασμό ο οποίος θα χρησιμοποιείται από το σύστημα καθ' όλη την διάρκεια των συνδέσεων του από την δημιουργία του λογαριασμού και μετά. Κατά την εγγραφή του θα μπορεί να καθορίσει σε ποια κατηγορία θα ανήκει αν θα είναι πωλητής, πελάτης ή και τα δύο. Συγκεκριμένα ένας ενδιαφερόμενος πελάτης θα μπορεί να καταχωρεί σε μια φόρμα τις απαραίτητες πληροφορίες που απαιτούνται για την δημιουργία προσωπικού λογαριασμού. Τα πεδία που πρέπει να συμπληρώσει είναι το όνομα του, το επίθετο, το email, τον κωδικό, να επιβεβαιώσει τον κωδικό του και να επιλέξει σε ποια κατηγορία θα ανήκει αν θα είναι πελάτης, αγοραστής ή και τα δύο. Στη συνέχεια, η εφαρμογή ελέγχει πρώτα αν έχουν συμπληρωθεί όλα τα πεδία και ακολούθως ελέγχει αν το πεδίο του κωδικού πρόσβασης και το πεδίο επικύρωσης του κωδικού ταυτίζονται. Επίσης, η εφαρμογή μετατρέπει τους χαρακτήρες που εισάγονται στον κωδικό σε κουκίδες για να μην είναι ορατός. Αν υπάρχει οποιοδήποτε λάθος κατά την εισαγωγή στα πεδία τότε εμφανίζονται τα κατάλληλα μηνύματα, αλλιώς αποστέλλονται οι πληροφορίες στον εξυπηρετητή με την μορφή JSON αντικειμένου. Στον εξυπηρετητή γίνεται ο έλεγχος για το αν το email που έδωσε ο χρήστης χρησιμοποιείται ήδη από κάποιον άλλο χρήστη και ο εξυπηρετητής ενημερώνει ανάλογα την εφαρμογή. Αν ο εξυπηρετητής ενημερώσει αρνητικά την εφαρμογή τότε εμφανίζεται το κατάλληλο μήνυμα στην εφαρμογή, αλλιώς γίνεται η εγγραφή και ο χρήστης πλέον βλέπει το προφίλ του. Οι πληροφορίες του χρήστη αποθηκεύονται και στην τοπική βάση της εφαρμογής για να μην χρειάζεται συνέχεια να επικοινωνεί με τον εξυπηρετητή.



Σχήμα 3.6 Η οθόνη εγγραφής όπου περιέχει τα εξής πεδία name, last name, email, password, confirm password, κατηγορία χρήστη και τα κουμπιά Register και Cancel. Στο συγκεκριμένο σχήμα φαίνονται τα μηνύματα που εμφανίζονται όταν πατηθεί το κουμπί Register και κάποια πεδία δεν είναι συμπληρωμένα.

2.Είσοδος μελών

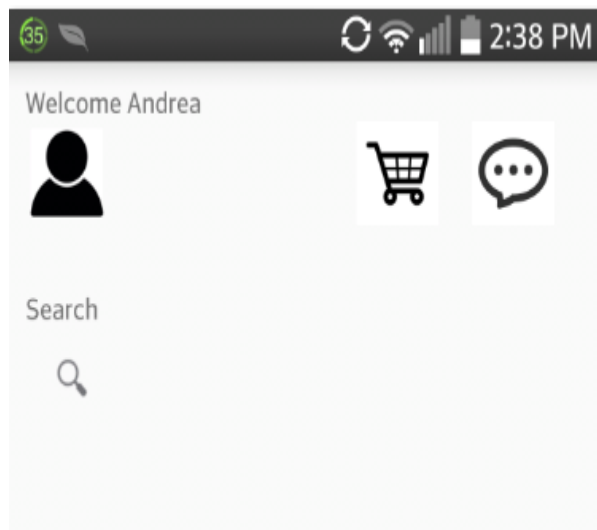
Η πρώτη οθόνη που εμφανίζεται όταν ανοίξει ο χρήστης την εφαρμογή είναι η οθόνη που ζητά τα διαπιστευτήρια του χρήστη, email και κωδικό πρόσβασης, για να μπορέσει να συνδεθεί και να χρησιμοποιήσει την εφαρμογή. Ο χρήστης πρέπει να συμπληρώσει τα δύο πεδία ,τα πεδία του email και του κωδικού πρόσβασης. Στην συνέχεια, η εφαρμογή ελέγχει αν είναι συμπληρωμένα και τα δύο πεδία ενώ αν δεν είναι, εμφανίζονται τα κατάλληλα μηνύματα στον χρήστη που του ζητούν να συμπληρώσει σωστά τα πεδία. Αν όλα τα πεδία είναι συμπληρωμένα σωστά τότε στέλνονται τα στοιχεία του χρήστη στον εξυπηρετητή αφού πρώτα κωδικοποιηθούν σε JSON αντικείμενο. Ο εξυπηρετητής με τη σειρά του κάνει την ταυτοποίηση ελέγχοντας αν υπάρχει το ίδιο email με τον ίδιο κωδικό στη βάση δεδομένων και ενημερώνει την εφαρμογή ανάλογα: αν έγινε η ταυτοποίηση ή όχι. Η εφαρμογή λαμβάνει την απάντηση του εξυπηρετητή και αν είναι θετική τότε μεταφέρει το χρήστη στο προφίλ του, αλλιώς εμφανίζει το κατάλληλο μήνυμα. Για τους χρήστες που χρησιμοποιούν για πρώτη φορά την εφαρμογή, στη οθόνη εισόδου υπάρχει και η επιλογή της εγγραφής. Οι πληροφορίες του χρήστη αποθηκεύονται και στην τοπική βάση της εφαρμογής, για να μην χρειάζεται συνέχεια να επικοινωνεί με τον εξυπηρετητή.



Σχήμα 3.7 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη εισόδου στην οποία υπάρχουν τα εξής πεδία email, κωδικός πρόσβασης και τα κουμπιά Login και Register. Στο συγκεκριμένο σχήμα φαίνονται τα μηνύματα που εμφανίζονται όταν πατηθεί το κουμπί Login και κάποια πεδία δεν είναι συμπληρωμένα.

3. Προφίλ Πελάτη

Ο κάθε χρήστης-πελάτης του συστήματος αφού εισέλθει στο σύστημα θα μπορεί να ενημερώνει το καλάθι αγορών του μέσω των διαφόρων προϊόντων τα οποία έχει εισάγει πατώντας το εικονίδιο για το καλάθι. Πιο συγκεκριμένα, ο πελάτης θα έχει τη δυνατότητα μέσω αναζήτησης με βάση λέξεων κλειδιών, να του εμφανίζονται όλα τα σχετικά προϊόντα παρέχοντάς του τις εξής πληροφορίες: όνομα προϊόντος, τιμή και το όνομα του πωλητή. Με την επιλογή κάποιου προϊόντος εμφανίζονται οι πληροφορίες του. Όταν πατηθεί το εικονίδιο του χρήστη τότε του εμφανίζεται ένα μενού με τις επιλογές για τροποποίηση των πληροφοριών του προφίλ του, για πίστωση χρημάτων ενώ θα έχει και την επιλογή να αποσυνδεθεί. Με το πάτημα του εικονιδίου των μηνυμάτων θα εισέρχεται στα εισερχόμενα μηνυμάτα του και θα μπορεί να δει ποιος του έχει στείλει μήνυμα. Τέλος, όταν ο χρήστης κλείσει την εφαρμογή χωρίς να αποσυνδεθεί την επόμενη φορά που θα ενεργοποιήσει την εφαρμογή θα εμφανιστεί η οθόνη με το προφίλ του και όχι η αρχική οθόνη της σύνδεσης. Η εφαρμογή κωδικοποιεί τις πληροφορίες του χρήστη σε αντικείμενο JSON και της αποστέλλει στον εξυπηρετητή για να τον εισάγει στην λίστα με τους ενεργούς χρήστες, στη βάση δεδομένων και να ελέγξει αν υπάρχουν νέα μηνύματα στη βάση δεδομένων που αφορούν το χρήστη.

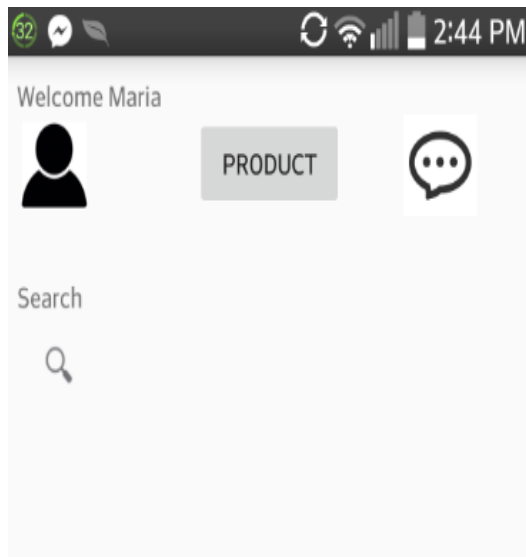


Σχήμα 3.8 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη του προφίλ ενός χρήστη-πελάτη η οποία περιέχει το εικονίδιο του χρήστη που όταν πατηθεί εμφανίζει τις επιλογές: τροποποίηση προφίλ, πίστωση χρημάτων και αποσύνδεση λογαριασμού. Επίσης, παρουσιάζονται το εικονίδιο του καλαθιού, το εικονίδιο των μηνυμάτων και η δυνατότητα αναζήτησης προϊόντων.

4. Προφίλ Πωλητή

Ο κάθε χρήστης-πωλητής του συστήματος αφού εισέλθει στο σύστημα θα μπορεί να ενημερώνει τα στοιχεία τα οποία έχει εισαγάγει στο σύστημα. Ποιο συγκεκριμένα ο πωλητής εφόσον επιλέξει τα προϊόντα του μέσω αναζήτησης με βάση λέξεις κλειδιά, θα του δίνεται η δυνατότητα να ελέγχει τα προϊόντα του για τη διαθέσιμη ποσότητα του κάθε προϊόντος. Επίσης, θα έχει την δυνατότητα να τροποποιήσει της πληροφορίες των προϊόντων αλλά και να τα διαγράψει. Μέσα από το προφίλ του μπορεί να προσθέσει νέα προϊόντα πατώντας το κουμπί Product. Όταν πατήσει το εικονίδιο του χρήστη τότε εμφανίζεται ένα μενού με τις επιλογές για τροποποίηση των πληροφοριών του προφίλ του καθώς και με την επιλογή της αποσύνδεσης. Δε διαθέτει όμως τη δυνατότητα πίστωσης χρημάτων, αφού αποτελεί μόνο πωλητή. Με το πάτημα του εικονιδίου των μηνυμάτων θα εισέρχεται στα εισερχόμενα μηνύματά του και θα μπορεί να δει ποιος του έχει στείλει μήνυμα. Τέλος, όταν ο χρήστης κλείσει την εφαρμογή χωρίς να

αποσυνδεθεί την επόμενη φορά που θα ενεργοποιήσει την εφαρμογή θα εμφανιστεί η οθόνη με το προφίλ του και όχι η αρχική οθόνη τις σύνδεσης. Η εφαρμογή κωδικοποιεί τις πληροφορίες του χρήστη σε αντικείμενο JSON και της αποστέλλει στον εξυπηρετητή για να τον εισαγάγει στη λίστα με τους ενεργούς χρήστες στη βάση δεδομένων και να ελέγξει αν υπάρχουν νέα μηνύματα στη βάση δεδομένων που αφορούν τον χρήστη.

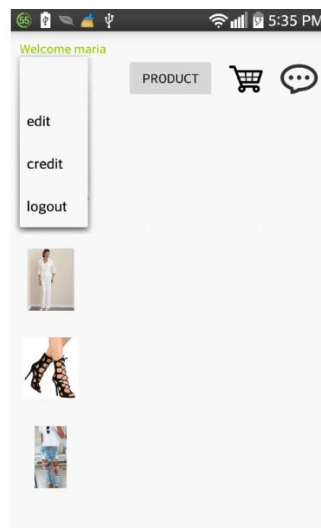


Σχήμα 3.9 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη του προφίλ ενός χρήστη-πωλητή η οποία περιέχει το εικονίδιο του χρήστη που όταν πατηθεί εμφανίζει τις επιλογές τροποποίηση προφίλ και αποσύνδεση λογαριασμού. Επίσης, παρουσιάζονται το κουμπί Product για πρόσθεση νέων προϊόντων, το εικονίδιο των μηνυμάτων και η δυνατότητα αναζήτησης των προϊόντων του με σκοπό την επεξεργασία ή την διαγραφή τους.

5. Προφίλ Πωλητή-Πελάτη

Η τρίτη κατηγορία που μπορεί να επιλέξει ο χρήστης είναι να είναι και πωλητής και πελάτης. Ποιο συγκεκριμένα ο πωλητής-πελάτης εφόσον επιλέξει τα προϊόντα του μέσω αναζήτησης με βάση λέξεις κλειδιά θα του δίνεται η δυνατότητα να ελέγχει τα προϊόντα του για τη διαθέσιμη ποσότητα του κάθε προϊόν. Επίσης, θα έχει τη δυνατότητα να τροποποιήσει τις πληροφορίες των προϊόντων αλλά και να τα διαγράψει. Μέσα από το προφίλ του μπορεί να προσθέσει νέα προϊόντα πατώντας το κουμπί Product. Ο κάθε χρήστης-πωλητής-πελάτης του συστήματος αφού εισέλθει στο σύστημα θα μπορεί να ενημερώνει το καλάθι αγορών του μέσω των διαφόρων προϊόντων τα οποία έχει εισαγάγει πατώντας το εικονίδιο για το καλάθι. Πιο συγκεκριμένα ο πελάτης θα έχει τη δυνατότητα μέσω αναζήτησης με βάση λέξεων

κλειδιών να εμφανίζονται όλα τα σχετικά προϊόντα παρέχοντάς του τις εξής πληροφορίες όνομα προϊόντος, τιμή και όνομα του πωλητή. Με το να επιλέξει κάποιο προϊόν ο χρήστης εμφανίζονται οι πληροφορίες του. Όταν πατήσει το εικονίδιο του χρήστη τότε εμφανίζεται ένα μενού με τις επιλογές για τροποποίηση των πληροφοριών του προφίλ του, για πίστωση λεφτών και η επιλογή της αποσύνδεσης. Με το πάτημα του εικονιδίου των μηνυμάτων θα εισέρχεται στα εισερχόμενα μηνύματά του και θα μπορεί να δει ποιος του έχει στείλει μήνυμα. Τέλος, όταν ο χρήστης κλείσει την εφαρμογή χωρίς να αποσυνδεθεί την επόμενη φορά που θα ενεργοποιήσει την εφαρμογή θα εμφανιστεί η οθόνη με το προφίλ του και όχι η αρχική οθόνη τις σύνδεσης. Η εφαρμογή κωδικοποιεί της πληροφορίες του χρήστη σε αντικείμενο JSON και της αποστέλλει στον εξυπηρετητή για να τον εισάγει στην λίστα με του ενεργούς χρήστες στην βάση δεδομένων και να ελέγξει αν υπάρχουν νέα μηνύματα στη βάση δεδομένων που αφορούν το χρήστη.

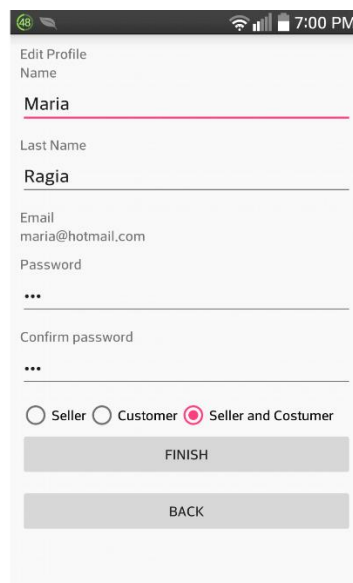


Σχήμα 3.10 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη του προφίλ ενός χρήστη-πωλητή-πελάτη η οποία περιέχει το εικονίδιο του χρήστη που όταν πατηθεί εμφανίζει τις επιλογές τροποποίηση προφίλ, πίστωση χρημάτων και αποσύνδεση λογαριασμού. Επίσης παρουσιάζονται το κουμπί Product για πρόσθεση, επεξεργασία και διαγραφή προϊόντων, το εικονίδιο του καλαθιού, το εικονίδιο των μηνυμάτων και η δυνατότητα αναζήτησης προϊόντων.

6. Τροποποίηση Πληροφοριών Χρήστη

Μετά την είσοδο του χρήστη στο αντίστοιχο του προφίλ, ο χρήστης αν θέλει μπορεί να κάνει αλλαγές στα στοιχεία του αλλά και να αλλάξει κατηγορία στην οποία ανήκει. Όταν πατήσει το εικονίδιο του χρήστη και επιλέξει το edit, τότε εμφανίζεται η οθόνη με τα πεδία που συμπλήρωσε κατά την εγγραφή, τα οποία είναι τα πεδία που πρέπει να συμπληρώσει: το όνομα του, το επίθετο, το email, τον κωδικό, να επιβεβαιώσει τον κωδικό του και να επιλέξει σε ποια κατηγορία θα ανήκει, αν θα είναι πελάτης ή αγοραστής ή και τα δύο. Στην συνέχεια, η εφαρμογή ελέγχει πρώτα αν είναι συμπληρωμένα όλα τα πεδία και αν τα πεδία όπου ζητήθηκε ο κωδικός πρόσβασης και η επικύρωση του κωδικού είναι ίδια.

Επίσης, η εφαρμογή μετατρέπει τους χαρακτήρες που εισάγονται στον κωδικό σε κουκίδες για να μην είναι ορατός. Αν υπάρχει οποιοδήποτε λάθος εισαγωγής στα πεδία τότε εμφανίζονται τα κατάλληλα μηνύματα, αλλιώς αποστέλλονται οι πληροφορίες στον εξυπηρετητή με την μορφή JSON αντικειμένου και εκεί γίνεται ο έλεγχος για το αν το email που έδωσε ο χρήστης χρησιμοποιείται ήδη από κάποιον άλλο χρήστη και ο εξυπηρετητής ενημερώνει ανάλογα την εφαρμογή. Αν ο εξυπηρετητής ενημερώσει αρνητικά την εφαρμογή τότε εμφανίζεται το κατάλληλο μήνυμα στην εφαρμογή, αλλιώς γίνεται η αλλαγή και ο χρήστης πλέον βλέπει το προφίλ του. Οι νέες πληροφορίες του χρήστη αποθηκεύονται και στην τοπική βάση της εφαρμογής για να μην χρειάζεται συνέχεια να επικοινωνεί με τον εξυπηρετητή.

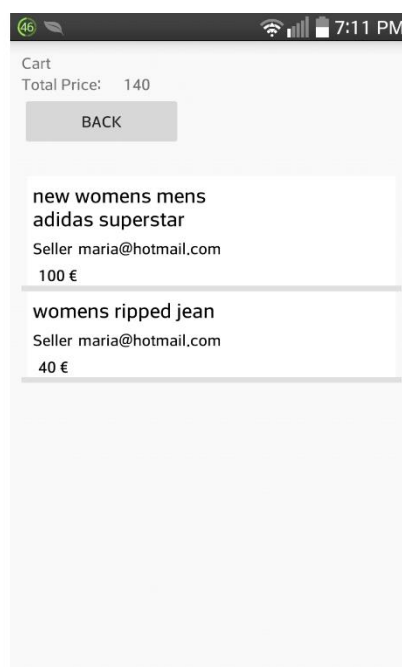


Σχήμα 3.11 Η οθόνη τροποποίησης πληροφοριών χρήστη όπου περιέχει τα εξής πεδία name, last name, email, password, confirm password, κατηγορία χρήστη και τα κουμπιά Finish και Cancel. Το πεδίο email δεν μπορεί να τροποποιηθεί.

7. Καλάθι Προϊόντων

Κάθε χρήστης-πελάτης και κάθε χρήστης-πωλητής-πελάτης έχει το δικό του καλάθι αγορών μέσα από το οποίο μπορεί να διαχειριστεί τα προϊόντα τα οποία τον ενδιαφέρουν και θα ήθελε να αγοράσει. Μέσα από την οθόνη του προφίλ του χρήστη όταν πατηθεί το εικονίδιο του καλαθιού μεταφέρεται στην οθόνη του καλαθιού. Εκεί η εφαρμογή κωδικοποιεί τις πληροφορίες του χρήστη σε JSON αντικείμενο και το αποστέλλει στον εξυπηρετητή ο οποίος με την σειρά του βρίσκει από την βάση δεδομένων τα αντικείμενα που είχε προσθέσει ο

χρήστης στο καλάθι του και τα αποστέλλει στην εφαρμογή. Στην εφαρμογή δημιουργείται μια λίστα με τα προϊόντα με την χρήση του Adapter που δημιούργησα για να παρουσιάζονται τα εξής στοιχεία για κάθε προϊόν όνομα, πωλητής και τιμή. Για να παρουσιαστούν όλες οι πληροφορίες ενός προϊόντος, ο χρήστης πρέπει απλά να πατήσει στο προϊόν που θέλει να δει και μεταφέρεται στην σελίδα του προϊόντος. Επίσης, ο χρήστης έχει τη δυνατότητα να διαγράψει κάποιο προϊόν από το καλάθι πιέζοντας εκτεταμένα το προϊόν με αποτέλεσμα να εμφανίζεται η επιλογή διαγραφής όπου αν επιλεγθεί η εφαρμογή κωδικοποιεί το μοναδικό αριθμό του προϊόντος σε αντικείμενο JSON και το αποστέλλει στον εξυπηρετητή ο οποίος διαγράφει από την βάση δεδομένων το προϊόν από το καλάθι του συγκεκριμένου χρήστη. Τέλος, στο πάνω μέρος της οθόνης του καλαθιού εμφανίζεται η συνολική τιμή των προϊόντων που είναι μέσα στο καλάθι. Όταν ένα προϊόν διαγραφεί από τον κάτοχο του τότε αυτομάτως διαγράφεται και από τα καλάθια όλων των χρηστών.

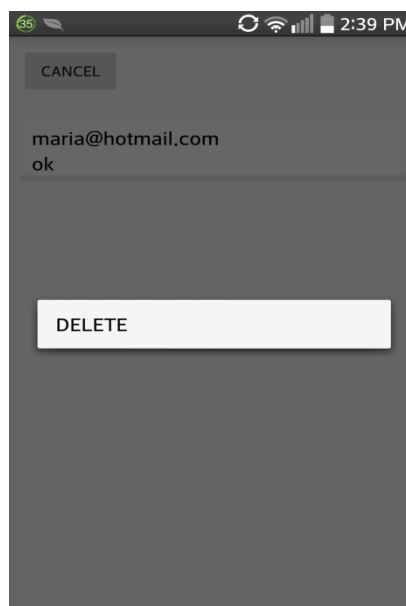


Σχήμα 3.12 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη ου καλαθιού στην οποία περιέχονται δύο προϊόντα. Στο πάνω μέρος της οθόνης εμφανίζεται και η συνολική τιμή των προϊόντων που βρίσκονται στο καλάθι.

8.Εισερχόμενα Μηνύματα

Στην εφαρμογή μου έχω υλοποιήσει ένα σύστημα ανταλλαγής μηνυμάτων σε περίπτωση που κάποιος από τον πωλητή και τον πελάτη θέλουν να επικοινωνήσουν σχετικά με την συναλλαγή τους. Ο χρήστης στο προφίλ του έχει ένα εικονίδιο ανταλλαγής μηνυμάτων που όταν το πατήσει μεταφέρεται στην οθόνη των εισερχόμενων μηνυμάτων. Σε αυτό το σημείο η

εφαρμογή κωδικοποιεί τις πληροφορίες του χρήστη σε αντικείμενα JSON και αποστέλλονται στον εξυπηρετητή, που με την σειρά του- με βάση τα δεδομένα του χρήστη- επιστρέφει στην εφαρμογή όλα τα εισερχόμενα μηνύματα του χρήστη. Στην εφαρμογή και πάλι υλοποίησα τον δικό μου Adapter για να εμφανίζονται τα μηνύματα σε μια λίστα με τα εξής πεδία, μέρος του μηνύματος και όνομα αποστολέα. Για να δει μια συνομιλία ή να απαντήσει σε κάποιο μήνυμα ο χρήστης απλά πατά πάνω στο μήνυμα που τον ενδιαφέρει και μεταφέρεται στην οθόνη της ανταλλαγής μηνυμάτων. Επίσης, ο χρήστης έχει την δυνατότητα να διαγράψει κάποιο μήνυμα πιέζοντας εκτεταμένα το προϊόν με αποτέλεσμα να εμφανίζεται η επιλογή διαγραφής όπου αν επιλεγθεί η εφαρμογή κωδικοποιεί τον μοναδικό αριθμό του μηνύματος σε αντικείμενο JSON και το αποστέλλει στον εξυπηρετητή ο οποίος διαγράφει από την βάση δεδομένων το μήνυμα του συγκεκριμένου χρήστη.

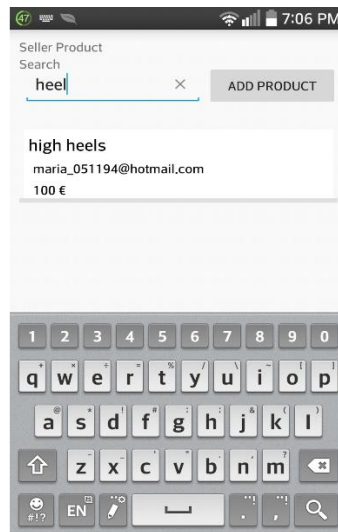


Σχήμα 3.13 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη εισερχόμενων μηνυμάτων όπου εμφανίζονται τα εισερχόμενα μηνύματα και ο χρήστης έχοντας πατημένο το μήνυμα εμφανίζεται η επιλογή Delete όπου μπορεί να το διαγράψει.

9. Προϊόντα πωλητή

Κάθε χρήστης-πωλητής και χρηστής-πωλητής-πελάτης μπορούν να δουν, να προσθέσουν και να τροποποιήσουν τα προϊόντα τους. Με το πάτημα του κουμπιού Product που βρίσκεται στην οθόνη του προφίλ του χρήστη μεταφέρονται στην οθόνη των προϊόντων τους. Εκεί μπορούν μέσω αναζήτησης με βάση λέξεις- κλειδιά να εμφανιστούν τα σχετικά προϊόντα τους σε μια

λίστα στην οποία έχω υλοποιήσει τον δικό μου Adapter για να εμφανίζονται οι εξής πληροφορίες του χρήστη όνομα και τιμή. Σε αυτό το σημείο για να εμφανιστούν οι υπόλοιπες πληροφορίες κάποιου προϊόντος ο χρήστης πρέπει απλά να επιλέξει το προϊόν που θέλει δει και να πατήσει πάνω του. Στην συνέχεια, η εφαρμογή αποθηκεύει τις πληροφορίες του προϊόντος που επιλέχθηκε στην τοπική βάση δεδομένων της εφαρμογής για να μπορέσει να εμφανίσει όλες τις πληροφορίες του στην οθόνη των επεξεργασίας των προϊόντων. Μια ακόμα δυνατότητα που έχει ο χρήστης είναι να προσθέσει νέα προϊόντα προς πώληση. Αυτό επιτυγχάνεται με το πάτημα του κουμπιού Add Product το οποίο μεταφέρει τον χρήστη στη οθόνη πρόσθεσης νέου προϊόντος για να εισάγει της σχετικές πληροφορίες για το νέο του προϊόν.

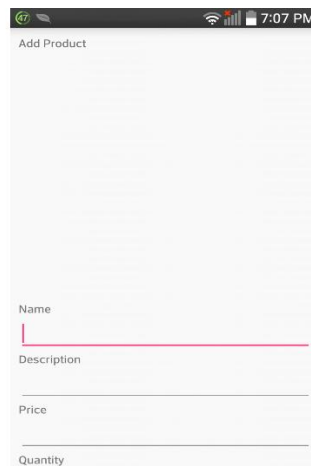


Σχήμα 3.14 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη των προϊόντων πωλητή στην οποία ο πωλητής μπορεί να προσθέσει νέα προϊόντα με το κουμπί Add Product και να αναζητήσει τα προϊόντα του.

10. Προσθήκη νέου προϊόντος

Κάθε ηλεκτρονικό κατάστημα πρέπει να παρέχει την δυνατότητα στους χρήστες του, που θέλουν να πουλήσουν ένα προϊόν, να μπορούν με εύκολο και γρήγορο τρόπο να διαθέσουν τα προϊόντα τους. Έτσι και στην εφαρμογή που ανέπτυξα κάθε χρήστης-πωλητής και χρηστής-πωλητής-πελάτης όταν πατήσει το κουμπί Add Product από την οθόνη προϊόντος μεταφέρεται στην οθόνη πρόσθεσης νέου προϊόντος. Εκεί η εφαρμογή ζητά από τον χρήστη να συμπληρώσει τις εξής πληροφορίες που αφορούν το νέο προϊόν: το όνομα, μια περιγραφή με τα χαρακτηριστικά του νέου προϊόντος, την τιμή και την ποσότητα των διαθέσιμων προϊόντων. Επίσης, ο χρήστης έχει την δυνατότητα να προσθέσει και φωτογραφία του προϊόντος είτε επιλέγοντας φωτογραφία από την μνήμη της έξυπνης συσκευής με το πάτημα του κουμπιού Gallery είτε με τη χρήση της κάμερας της έξυπνης συσκευής με το πάτημα του κουμπιού Camera. Με το πάτημα του κουμπιού Add Product, η εφαρμογή ελέγχει πρώτα αν είναι

συμπληρωμένα όλα τα πεδία. Αν όλα τα πεδία είναι συμπληρωμένα τότε κωδικοποιούνται οι πληροφορίες του νέου προϊόντος σε ένα αντικείμενο JSON και αποστέλλεται στον εξυπηρετητή ο οποίος με την σειρά του προσθέτει το νέο προϊόν στη βάση δεδομένων και ενημερώνει την εφαρμογή με μήνυμα ανάλογα αν ήταν επιτυχής η εισαγωγή του προϊόντος στη βάση δεδομένων. Η εφαρμογή ανάλογα με το μήνυμα που θα παραλάβει από τον εξυπηρετητή ενημερώνει και το χρήστη. Αν κάποιο πεδίο δεν είναι συμπληρωμένο και ο χρήστης πατήσει το κουμπί Add Product τότε εμφανίζεται μήνυμα που ζητά από τον χρήστη να συμπληρώσει το κενό πεδίο.



Σχήμα 3.15 Στο πιο πάνω σχήμα φαίνεται η οθόνη προσθήκης νέου προϊόντος όπου πρέπει να συμπληρωθούν τα εξής πεδία όνομα, περιγραφή, τιμή, διαθέσιμη ποσότητα. Επίσης, η οθόνη περιλαμβάνει και τα κουμπιά Camera, Gallery, Add Product, Cancel.

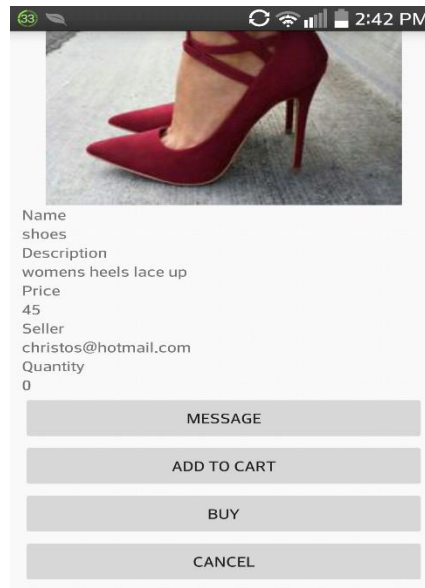
11. Τροποποίηση προϊόντων πωλητή

Μετά την εισαγωγή των προϊόντων τους κάθε χρήστης-πωλητής και χρήστης-πωλητής-πελάτης μπορεί να τροποποιήσει και να διαγράψει ένα προϊόν από την εφαρμογή. Αφού πρώτα αναζητήσει κάποιο προϊόν με βάση λέξεις- κλειδιά και μετά επιλέξει το προϊόν του που τον ενδιαφέρει πατώντας πάνω του, τότε ο μοναδικός αριθμός του επιλεγμένου αριθμού αποστέλλεται στον εξυπηρετητή ο οποίος βρίσκει το προϊόν μέσα στην βάση δεδομένων και το επιστρέφει στην εφαρμογή όπου παρουσιάζεται στη οθόνη προϊόντων του πωλητή. Σε αυτό το σημείο παρουσιάζονται στο χρήστη τα εξής πεδία: όνομα, περιγραφή με τις πληροφορίες του προϊόντος, τιμή, διαθέσιμη ποσότητα αλλά και η φωτογραφία του προϊόντος. Ο χρήστης πλέον, μπορεί να μην κάνει κάποια αλλαγή πατώντας το Back. Στην περίπτωση που αλλάξει κάποια από τα πεδία ή προσθέσει κάποια άλλη φωτογραφία, είτε από την μνήμη της έξυπνης συσκευής

είτε από την κάμερα, και πατήσει το κουμπί Update τότε αν όλα τα πεδία είναι συμπληρωμένα κωδικοποιούνται οι πληροφορίες του προϊόντος σε ένα αντικείμενο JSON και αποστέλλεται στον εξυπηρετητή, ο οποίος με την σειρά του ανανεώνει το προϊόν στη βάση δεδομένων και ενημερώνει την εφαρμογή με μήνυμα ανάλογα αν ήταν επιτυχής η ανανέωση του προϊόντος στη βάση δεδομένων. Η εφαρμογή ανάλογα με το μήνυμα που θα παραλάβει από τον εξυπηρετητή ενημερώνει και το χρήστη. Αν κάποιο πεδίο δεν είναι συμπληρωμένο και ο χρήστης πατήσει το κουμπί Update τότε εμφανίζεται μήνυμα που ζητά από τον χρήστη να συμπληρώσει το κενό πεδίο. Όταν ο χρήστης θέλει να διαγράψει κάποιο από τα προϊόντα του τότε το μόνο που έχει να κάνει είναι να πατήσει το κουμπί Delete και η εφαρμογή θα κωδικοποιήσει τις πληροφορίες του προϊόντος σε ένα αντικείμενο JSON και θα το αποστείλει στον εξυπηρετητή ο οποίος με την σειρά του θα διαγράψει από την βάση δεδομένων το προϊόν.

12.Παρουσίαση προϊόντων

Μια από τις βασικές λειτουργίες της εφαρμογής μου είναι η δυνατότητα που προσφέρει στους χρήστες-πελάτες να μπορούν να βλέπουν τα διάφορα προϊόντα και να έχουν την δυνατότητα να τα αγοράσουν αλλά και να στείλουν κάποιο μήνυμα στον πωλητή για τυχόν διευκρινήσεις για τα προϊόντα. Αφού πρώτα ο χρήστης αναζητήσει κάποιο προϊόν με βάση λέξεις- κλειδιά και μετά επιλέξει το προϊόν του που τον ενδιαφέρει πατώντας πάνω του, τότε ο μοναδικός αριθμός του επιλεγμένου αριθμού αποστέλλεται στον εξυπηρετητή, ο οποίος βρίσκει το προϊόν μέσα στη βάση δεδομένων και το επιστρέφει στην εφαρμογή, όπου παρουσιάζεται στη οθόνη παρουσίασης προϊόντων. Σε αυτό το σημείο παρουσιάζονται στο χρήστη οι πληροφορίες του προϊόντος δηλαδή το όνομα, η περιγραφή, η τιμή, η διαθέσιμη ποσότητα αλλά και η φωτογραφία του. Ο χρήστης μπορεί να επιλέξει να αγοράσει το προϊόν με το πάτημα του κουμπιού Buy, όπου θα ελεγχθεί πρώτα αν υπάρχει αρκετή πίστωση από το χρήστη για να αγοραστεί το προϊόν και μετά θα αποσταλεί ένα αίτημα αγοράς απευθείας- και όχι μέσω του εξυπηρετητή - στην εφαρμογή του πωλητή. Η εφαρμογή του τελευταίου μόλις λάβει το αίτημα θα αποστείλει στον εξυπηρετητή τον μοναδικό αριθμό του προϊόντος και αυτός θα ανακτήσει από τη βάση δεδομένων τη διαθέσιμη ποσότητα και θα την επιστρέψει στον πωλητή. Αν η διαθέσιμη ποσότητα είναι μεγαλύτερη του μηδενός τότε ο πωλητής ενημερώνει απευθείας των πελάτη ότι ολοκληρώθηκε η αγορά, αλλιώς τον ενημερώνει ότι δεν μπορεί να γίνει η αγορά. Στην περίπτωση που ο πωλητής δεν είναι συνδεδεμένος με την εφαρμογή του, τότε εμφανίζεται μήνυμα στον χρήστη ότι δεν είναι διαθέσιμος ο πωλητής και έχει τη δυνατότητα να προσθέσει το προϊόν στο καλάθι και να το αγοράσει αργότερα. Μια άλλη δυνατότητα που παρέχεται στην εφαρμογή είναι η ανταλλαγή μηνυμάτων μεταξύ του πελάτη και του πωλητή με το πάτημα του κουμπιού Chat όπου οι πληροφορίες του πωλητή θα αποθηκευτούν στην τοπική βάση τις εφαρμογής και ο χρήστης θα μεταφερθεί στην οθόνη ανταλλαγής μηνυμάτων.

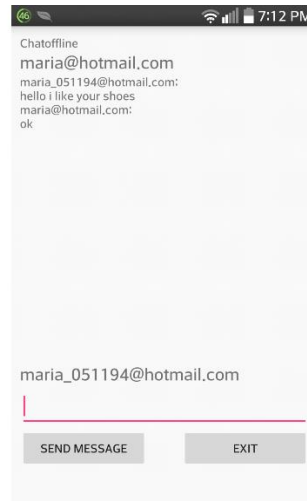


Σχήμα 3.16 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη παρουσίασης προϊόντων όπου περιέχει φωτογραφία του προϊόν, το όνομα, την περιγραφή, την τιμή, το email του πωλητή και την διαθέσιμη ποσότητα. Επίσης, περιέχει και τα κουμπιά Message, Add to Cart, Buy, Cancel.

13.Ανταλλαγή μηνυμάτων εκτός σύνδεσης

Η ανάδραση μεταξύ των ανθρώπων σε διαδικτυακό επίπεδο έχει γίνει πολύ διάσημη στην σύγχρονη εποχή με την προσβασιμότητα στο διαδίκτυο να έχει γίνει καθημερινότητα στις ζωές των ανθρώπων. Στην εφαρμογή μου έχω υλοποιήσει και την δυνατότητα ανταλλαγής μηνυμάτων μεταξύ χρηστών-πελατών και χρηστών-πωλητών που τους επιτρέπει να επικοινωνήσουν μεταξύ τους. Είτε από την οθόνη των εισερχόμενων μηνυμάτων είτε από την οθόνη παρουσίασης των προϊόντων μπορούν να μεταφερθούν στην οθόνη ανταλλαγής μηνυμάτων. Σε αυτό το σημείο η εφαρμογή κωδικοποιεί σε αντικείμενο JSON τις πληροφορίες του συνομιλητή, για να ελέγξει στη βάση δεδομένων των ενεργών χρηστών αν είναι ενεργός την στιγμή της ανταλλαγής μηνυμάτων. Σε αυτή την περίπτωση θα περιγραφεί η περίπτωση που δεν είναι ενεργός ο συνομιλητής. Σε αυτό το σημείο η εφαρμογή κωδικοποιεί σε δύο αντικείμενα JSON τις πληροφορίες -του χρήστη και του συνομιλητή- και τις αποστέλλει στον εξυπηρετητή ο οποίος με την σειρά του ανακτά από την βάση δεδομένων των μηνυμάτων- αν υπάρχει ολόκληρη- την συνομιλία μεταξύ των δύο και την επιστρέφει στην εφαρμογή. Η εφαρμογή με τη σειρά της παρουσιάζει στον χρήστη την συνομιλία. Ο χρήστης μπορεί να γράψει κάποιο μήνυμα και να πατήσει το κουμπί Send για να αποσταλεί το μήνυμα. Το μήνυμα μαζί με τις πληροφορίες του αποστολέα και παραλήπτη κωδικοποιούνται σε ένα αντικείμενο

JSON και αποστέλλονται στον εξυπηρετητή για εισαχθούν στην βάση δεδομένων των μηνυμάτων και επιστρέφει στην εφαρμογή μήνυμα ανάλογα με το αν είναι επιτυχής η εισαγωγή του μηνύματος στη βάση δεδομένων. Αν το μήνυμα που θα λάβει η εφαρμογή είναι θετικό τότε εμφανίζεται στην οθόνη με τα υπόλοιπα μηνύματα, το μήνυμα που στάλθηκε.



Σχήμα 3.17 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη ανταλλαγής μηνυμάτων. Όπως βλέπουμε εμφανίζονται όλα τα μηνύματα της συνομιλίας μεταξύ δύο χρηστών και στο κάτω μέρος ο χρήστης μπορεί να στείλει ένα νέο μήνυμα εισάγοντας το στο πεδίο και πατώντας το κουμπί Send.

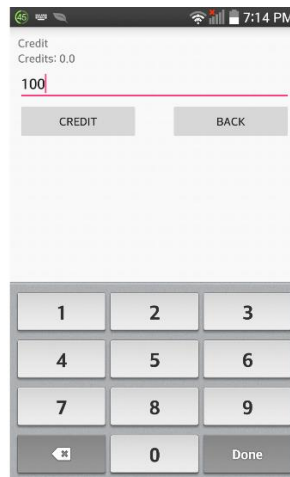
14.Ανταλλαγή μηνυμάτων εντός σύνδεσης

Στην εφαρμογή μου έχω υλοποιήσει και την δυνατότητα ανταλλαγής μηνυμάτων μεταξύ χρηστών-πελατών και χρηστών-πωλητών που τους επιτρέπει να επικοινωνήσουν μεταξύ τους. Είτε από την οθόνη των εισερχόμενων μηνυμάτων είτε από την οθόνη παρουσίασης των προϊόντων μπορούν να μεταφερθούν στην οθόνη ανταλλαγής μηνυμάτων. Σε αυτό το σημείο η εφαρμογή κωδικοποιεί σε αντικείμενο JSON τις πληροφορίες του συνομιλητή για να ελέγξει στη βάση δεδομένων των ενεργών χρηστών, αν είναι ενεργός τη στιγμή της ανταλλαγής μηνυμάτων. Σε αυτή την περίπτωση θα περιγραφεί η περίπτωση που είναι ενεργός ο συνομιλητής. Σε αυτό το σημείο η εφαρμογή κωδικοποιεί σε δύο αντικείμενα JSON τις πληροφορίες του χρήστη και του συνομιλητή και τις αποστέλλει στον εξυπηρετητή, ο οποίος με τη σειρά του ανακτά από τη βάση δεδομένων των μηνυμάτων, αν υπάρχει, ολόκληρη τη συνομιλία μεταξύ των δύο και την επιστρέφει στην εφαρμογή. Η εφαρμογή με τη σειρά της παρουσιάζει στο χρήστη τη συνομιλία. Ο χρήστης μπορεί να γράψει κάποιο μήνυμα και να πατήσει το κουμπί Send για να αποσταλεί το μήνυμα. Το μήνυμα μαζί με τις πληροφορίες του αποστολέα και παραλήπτη κωδικοποιούνται σε ένα αντικείμενο JSON και αποστέλλονται στον

εξυπηρετητή για εισαχθούν στη βάση δεδομένων των μηνυμάτων έτσι ώστε να διατηρηθούν όλες οι συνομιλίες μεταξύ του αποστολέα και του παραλήπτη, και επιστρέφει στην εφαρμογή μήνυμα ανάλογα με το αν είναι επιτυχής η εισαγωγή του μηνύματος στη βάση δεδομένων. Στη συνέχεια, ο αποστολέας και ο παραλήπτης συνδέονται απευθείας και το μήνυμα στέλνεται απευθείας το οποίο εμφανίζεται στην οθόνη με τα υπόλοιπα μηνύματα και των δύο .

15. Πίστωση χρημάτων από τους χρήστες

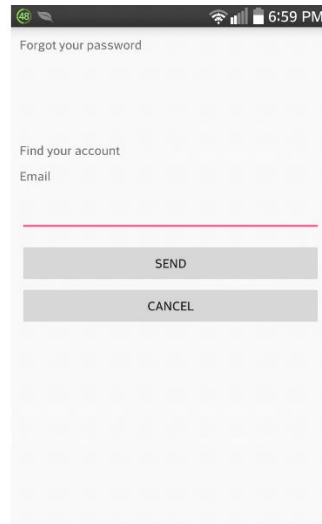
Για να υπάρχει η δυνατότητα συναλλαγής μεταξύ των χρηστών πρέπει να υπάρχει κάποιος τρόπος με τον οποίο οι χρήστες-πελάτες και οι χρήστες-πωλητές-πελάτες να πιστώνουν χρήματα. Για τον σκοπό τις λειτουργικότητας της εφαρμογής μου έχω αναπτύξει ένα απλό σύστημα πίστωσης εικονικών χρημάτων χωρίς να δίνονται λεπτομέρειες για κάρτες αφού δεν είχαμε σύνδεση με την JJC ούτε και άλλη δυνατότητα ασφαλούς πίστωσης πραγματικών χρημάτων. Ο κάθε χρήστης-πελάτης και ο χρήστης-πωλητή-πελάτης μέσω του προφίλ τους όταν επιλέξουν το εικονίδιο χρήστη, τους εμφανίζεται μια λίστα με επιλογές από την οποία επιλέγοντας την επιλογή Credit μεταφέρονται στην οθόνη πίστωσης χρημάτων. Σε αυτό το σημείο μπορούν να δουν το συνολικό ποσό που έχουν ήδη πιστώσει αλλά έχουν και την δυνατότητα, συμπληρώνοντας με ένα ποσό το πεδίο και πατώντας το κουμπί Credit, να πιστώσουν επιπλέον χρήματα. Όταν πατηθεί το κουμπί Credit τότε οι πληροφορίες του χρήστη μαζί με το σύνολο των ήδη πιστωμένων χρημάτων και των αριθμό των χρημάτων που θέλει ο χρήστης να πιστώσει, κωδικοποιούνται σε ένα αντικείμενο JSON και αποστέλλονται στον εξυπηρετητή. Στην συνέχεια, ο εξυπηρετητής ανανεώνει τις πληροφορίες του χρήστη στην βάση δεδομένων του χρήστη και επιστρέφει στην εφαρμογή ανάλογο μήνυμα για επιτυχία ή αποτυχία της ανανέωσης.



Σχήμα 3.18 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη στην οποία μπορεί ένας χρήστης να πιστώσει χρήματα στον λογαριασμό του

16. Αποστολή email για υπενθύμιση κωδικού πρόσβασης

Με το άνοιγμα της εφαρμογής ο χρήστης καλείται να δώσει το email του και τον κωδικό πρόσβασης του. Σε περίπτωση που δεν θυμάται τον κωδικό του τότε έχει την δυνατότητα να πατήσει στο Forgot your password?. Όταν επιλεγθεί το Forgot your password? εμφανίζεται στον χρήστη ένα πεδίο στο οποίο του ζητείται να δώσει το email του και στην συνέχεια αποστέλλεται ένα email στον χρήστη με τον κωδικό πρόσβασης του. Για την αποστολή του κωδικού έχω δημιουργήσει ένα λογαριασμό στο Gmail με το όνομα sellitapp1@gmail.com.



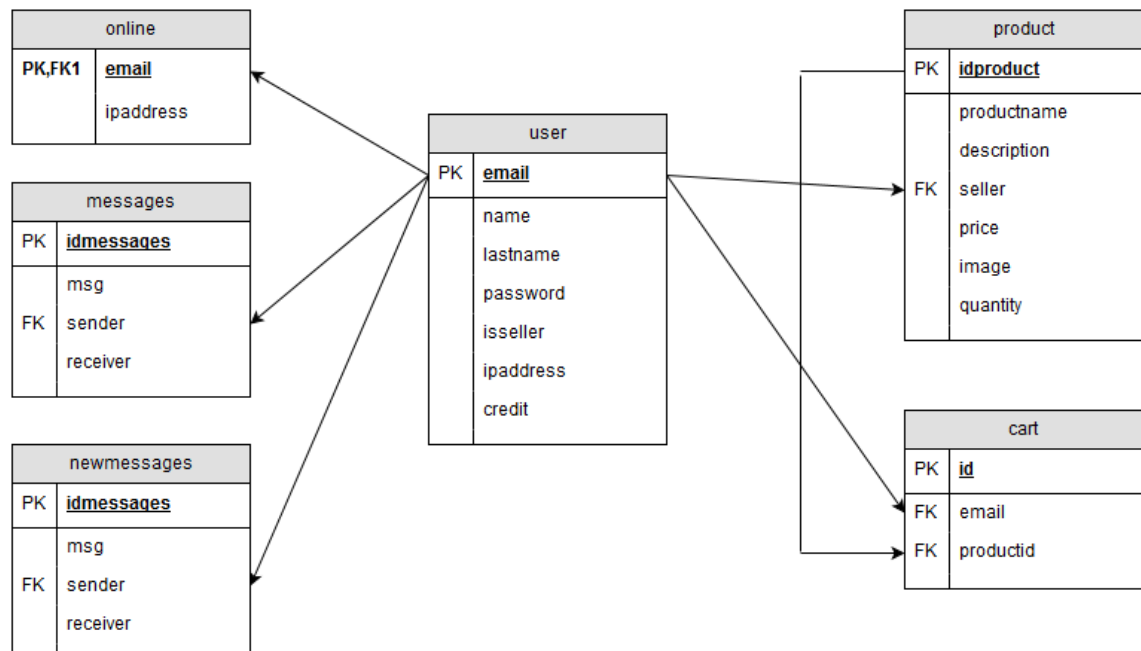
Σχήμα 3.19 Στο πιο πάνω σχήμα παρουσιάζεται η οθόνη στην οποία ο χρήστης σε περίπτωση που ξεχάσει τον κωδικό του εισάγει το email του για να λάβει ένα email με τον κωδικό του.

Κεφάλαιο 4

Βάση δεδομένων και εξυπηρετητής

4.1 Βάση δεδομένων	42
4.2 Πολυνηματικός εξυπηρετητής	44
4.3 Inverted Index	46

4.1 Βάση δεδομένων



Σχήμα 4.1 Στο πιο πάνω σχήμα παρουσιάζεται το σχήμα της βάσης δεδομένων.

Για να μπορέσω να διατηρήσω τα δεδομένα της εφαρμογής μου έχω υλοποιήσει μια βάση δεδομένων σε MySQL την οποία θα περιγράψω πιο κάτω:

User:

- Name
- Last name
- Email
- Password
- Isseller
- Ip address
- Credit

Ο πίνακας user χρησιμοποιείται για να κρατά τις πληροφορίες όλων των εγγεγραμμένων χρηστών. Το πεδίο name που είναι το όνομα του χρήστη. Το πεδίο last name που είναι το επίθετο του χρήστη. Το πεδίο password που είναι ο κωδικός πρόσβασης του χρήστη. Το πεδίο credit που είναι τα πιστωμένα χρήματα του χρήστη. Το πεδίο isseller διατηρεί την κατηγορία στην οποία ανήκει ο χρήστης. Το πεδίο ip address διατηρεί την ip address του χρήστη για να μπορεί κάποιος άλλος χρήστης να την ξέρει και να ενωθεί απευθείας μαζί του είτε για ανταλλαγή μηνυμάτων είτε για την αγορά προϊόντος. Το μοναδικό στοιχείο του πίνακα είναι το email.

Online user:

- Email
- Ip address

Ο πίνακας online user χρησιμοποιείται για να κρατά τους ενεργούς χρήστες της εφαρμογής. Κάθε φορά που κάποιος χρήστης εισέρχεται στο προφίλ του οι πληροφορίες του, κυρίως το ip address, ανανεώνονται και γίνεται εισαγωγή στον πίνακα online user. Το πεδίο ip address διατηρεί την ip address του χρήστη για να μπορεί κάποιος άλλος χρήστης να την ξέρει και να ενωθεί απευθείας μαζί του είτε για ανταλλαγή μηνυμάτων είτε για την αγορά προϊόντος. Το μοναδικό στοιχείο του πίνακα είναι το email.

Product:

- Idproduct
- Name
- Description
- Price
- Seller
- Quantity
- Image

Ο πίνακας product χρησιμοποιείται για να διατηρούνται οι πληροφορίες για όλα τα προϊόντα που είναι διαθέσιμα στην εφαρμογή. Περιέχει τα εξής πεδία: idproduct που είναι ο μοναδικός αριθμός για κάθε προϊόν, name που είναι το όνομα του προϊόντος, το description που είναι η περιγραφή των χαρακτηριστικών του προϊόντος, price που είναι η τιμή, seller το οποίο είναι το email του χρήστη που πουλά το προϊόν, quantity που είναι η διαθέσιμη ποσότητα του προϊόντος και image το οποία είναι η φωτογραφία του κάθε προϊόντος.

Messages:

- Idmessage
- Msg
- Sender
- Receiver

Ο πίνακας messages περιέχει όλα τα μηνύματα που έχουν σταλεί μεταξύ των χρηστών της εφαρμογής μου. Περιέχει τα εξής πεδία idmessage το οποίο είναι ο μοναδικός αριθμός του κάθε μηνύματος, msg που είναι το περιεχόμενο του μηνύματος που στάλθηκε, sender ο οποίος είναι ο αποστολέας του μηνύματος, receiver ο οποίος είναι ο παραλήπτης του μηνύματος.

New messages:

- Idmessage
- Msg
- Sender
- Receiver

Ο πίνακας new messages περιέχει όλα τα νέα μηνύματα που έχουν σταλεί μεταξύ των χρηστών και δεν τα έχουν δει ακόμη οι παραλήπτες τους. Κάθε φορά που κάποιος χρήστης στέλνει ένα νέο μήνυμα σε κάποιον άλλο χρήστη τότε το μήνυμα εισάγεται και στον πίνακα messages αλλά και στον πίνακα new messages με σκοπό να ειδοποιείται ο

χρήστης ότι έχει παραλάβει ένα νέο μήνυμα. Όταν ο χρήστης μπει στα εισερχόμενα μηνύματα του και δει τα νέα μηνύματα, αυτά τα μηνύματα διαγράφονται από τον πίνακα new messages. Περιέχει τα εξής πεδία idmessage το οποίο είναι ο μοναδικός αριθμός του κάθε μηνύματος, msg που είναι το περιεχόμενο του μηνύματος που στάλθηκε, sender ο οποίος είναι ο αποστολέας του μηνύματος, receiver ο οποίος είναι ο παραλήπτης του μηνύματος.

cart :

- email
- productid
- id

Ο πίνακας cart περιέχει τα προϊόντα που έχει ο κάθε χρήστης στο καλάθι του. Περιέχει τα εξής πεδία email το οποίο είναι το email του κάθε χρήστη, το productid που είναι το id του προϊόντος, το id το οποίο είναι το μοναδικό στοιχείο του πίνακα.

4.2 Πολυνηματικός Εξυπηρετητής

Στα πλαίσια της διπλωματικής μου εργασίας έχω αναπτύξει ένα πολυνηματικό εξυπηρετητή σε Java με την βοήθεια του γραφικού περιβάλλοντος Eclipse, με σκοπό να εξυπηρετεί τους χρήστες τις εφαρμογές μου.

Αρχικά, έχω υλοποιήσει μια δημόσια κλάση με το όνομα Server.java μέσα στην οποία, χρησιμοποιώντας κυρίως την βιβλιοθήκη Socket της Java, υλοποιώ των πολυνηματικό εξυπηρετητή. Δημιουργώ ένα αντικείμενο ServerSocket και του δίνω ως όρισμα τον αριθμό 8080, δηλαδή με αυτό τον τρόπο καθορίζω ότι ο εξυπηρετητής θα δέχεται και θα απαντά αιτήματα από την πόρτα 8080 του υπολογιστή μου. Στη συνέχεια, με την εντολή connection.accept, όπου το connection είναι ένα αντικείμενο τύπου ServerSocket, ο εξυπηρετητής ξεκινά να δέχεται αιτήματα για σύνδεση από πελάτες. Μετά, ορίζω μια εσωτερική κλάση την MultithreadServer μέσα στην οποία γίνεται η διαχείριση των αιτημάτων δηλαδή, γίνεται το διάβασμα του αιτήματος, αποφασίζεται ποια συνάρτηση θα εκτελεστεί από τον εξυπηρετητή σε σχέση με τη βάση δεδομένων και γίνεται η απάντηση στο αίτημα επιστρέφοντας τα δεδομένα που ανακτήθηκαν από τη βάση δεδομένων. Για να γίνει πολυνηματικός ο εξυπηρετητής μου χρησιμοποιώ την

βιβλιοθήκη Thread της Java, όπου για κάθε νέα σύνδεση η βιβλιοθήκη Thread δημιουργεί ένα νέο νήμα της εσωτερικής κλάσης MultithreadServer για κάθε σύνδεση. Με αυτό τον τρόπο ο πολυνηματικός εξυπηρετητής μπορεί να εξυπηρετήσει πολλούς χρήστες παράλληλα και όχι σειριακά όπως θα έκανε ένας απλός εξυπηρετητής. Αυτό έχει ως αποτέλεσμα να μειώνεται ο χρόνος απόκρισης των αιτημάτων από τον εξυπηρετητή και έτσι να μειώνεται και ο χρόνος που περιμένουν οι χρήστες όταν εκτελούν κάποια εργασία πάνω στην εφαρμογή.

Τόσο τα αιτήματα που λαμβάνει ο εξυπηρετητής, όσο και οι απαντήσεις που στέλλει είναι κωδικοποιημένες σε αντικείμενα τύπου JSON. Αυτό επιτυγχάνεται με την βοήθεια της εξωτερικής βιβλιοθήκης gson-2.3.1.jar. Τα JSON αντικείμενα είναι μια αναγνωρίσιμη μορφή δεδομένων οι οποία χρησιμοποιείται κυρίως για τη μετάδοση δεδομένων μεταξύ εφαρμογών διαδικτύου και εξυπηρετητών. Η μορφή που έχουν τα αντικείμενα JSON είναι Key/Value Pair δηλαδή ένα ζευγάρι τιμών όπου το Key είναι το όνομα του πεδίου θέλουμε να κωδικοποιήσουμε και το Value είναι η τιμή που έχει το συγκεκριμένο πεδίο.

Επίσης, μια άλλη κλάση που χρησιμοποιεί ο πολυνηματικός εξυπηρετητής είναι η κλάση Request.java που με την βοήθεια της εξωτερικής βιβλιοθήκης mysql-connector-java-5.1.36.jar, την οποία ανέπτυξε η Microsoft για να είναι δυνατή η επικοινωνία μεταξύ προγραμμάτων σε Java και MySql βάσεων δεδομένων, επιτρέπει στον εξυπηρετητή να ενωθεί με την βάση δεδομένων και να εκτελέσει τα απαραίτητα queries για κάθε αίτημα.

4.3 Inverted Index

Ο εξυπηρετητής μου για όλες τις αναζητήσεις προϊόντων των χρηστών χρησιμοποιεί την κλάση Inverted Index.java μέσα στην οποία έχω υλοποιήσει ένα αντεστραμμένο ευρετήριο. Η κλάση Inverted Index.java δέχεται ως είσοδο ένα Array List από προϊόντα και το query του χρήστη. Με τη βοήθεια των εξωτερικών βιβλιοθηκών της **Apache Lucene lucene-query parser-5.4.1.jar, Lucene-queries-5.4.1.jar, Lucene-core-5.4.1.jar** και **lucene-analyzers-common-5.4.1.jar** αφαιρώ τα stop words τόσο από το όνομα και την περιγραφή όλων των προϊόντων αλλά και από το query του χρήστη. Ακολουθώς η κλάση Stemmer στην οποία υλοποιείται ο αλγόριθμος Porter την οποία έχω πάρει έτοιμη και έχω τροποποιήσει, δημιουργεί όλες τις δυνατές λέξεις από τη ρίζα της κάθε λέξης που δέχεται ως είσοδο. Ο αλγόριθμος Porter είναι ο πιο κοινός αλγόριθμος για Stemming αγγλικών λέξεων. Οι τυπικοί κανόνες τους οποίους εκτελεί ο αλγόριθμος είναι οι εξής:

- sses→ss
- ies→i
- ational→ate
- tional→tion

Δημιουργείται ένα `HashMap<String, HashSet<InvertedIndexProduct2>>` δηλαδή ένα `HashMap` που έχει για κλειδί ένα `String` και για κάθε κλειδί περιέχει `HashSet` από τα `id` των προϊόντων μόνο. Κάθε μια από τις λέξεις που βρίσκονται, είτε στο όνομα είτε στην περιγραφή των προϊόντων, γίνεται μοναδικό κλειδί στο `HasMap` και τα `a id` των προϊόντων που περιέχουν αυτή τη λέξη εισάγονται στο κάθε ένα `HashSet`. Σε αυτό το σημείο γίνεται η αναζήτηση των προϊόντων. Κάθε μια από της λέξεις του `query` του χρήστη ελέγχεται αν είναι ίδια με κάποιο από τα κλειδιά στο `HashMap` και γίνει ταύτιση τότε όλα τα προϊόντα που βρίσκονται στο `HashSet`, το οποίο αντιστοιχεί στο κλειδί που είναι ίδιο με τη λέξη από το `Query` του χρήστη και επιστρέφονται τα `id` των προϊόντων για να ανευρεθούν από την βάση δεδομένων τα προϊόντα και να επιστραφούν οι πληροφορίες τους στην εφαρμογή. Το `InvertedIndex` κάθε φορά που ξεκινά ο εξυπηρετητής δημιουργείτε με την βοήθεια ενός πίνακα στην βάση δεδομένων ο οποίος περιέχει το `id` των προϊόντων και τις λέξεις που αντιστοιχούν σε αυτά. Όταν γίνει εισαγωγή νέου προϊόντος στην βάση δεδομένων τότε το προϊόν προσθέτετε και στο `Inverted Index`. Όταν διαγραφεί κάποιο προϊόν από την βάση δεδομένων τότε διαγράφεται και από το `Inverted Index`.

Κεφάλαιο 5

Επέκταση

5.1 Μελλοντική Δουλειά	48
5.2 Συμπεράσματα	49

5.1 Μελλοντική Δουλειά

Μέσα από την διαδικασία ολοκλήρωσης της διπλωματικής μου εργασίας έχω διδαχθεί πολλά και ενδιαφέροντα πράγματα γύρω από το πεδίο ανάπτυξης διαδικτυακών εφαρμογών, με αποτέλεσμα να έχω αποφασίσει ότι θέλω να εργαστώ σε αυτόν τον τομέα της Πληροφορικής.

Η εφαρμογή που έχω αναπτύξει ανταποκρίνεται αρκετά καλά στα πλείστα σενάρια για τα οποία την έχω δοκιμάσει, όμως υπάρχουν κάποιες επεκτάσεις οι οποίες μπορούν να γίνουν σε μελλοντικό στάδιο από κάποιον άλλο φοιτητή.

Αρχικά, πρέπει να βελτιωθεί η εμφάνιση των γραφικών της εφαρμογής αφού πλέον οι δυνατότητες που προσφέρουν τα εργαλεία ανάπτυξης εφαρμογών, είναι απεριόριστες.

Ένα άλλο θέμα που πρέπει να επιλυθεί είναι το πώς θα γίνεται ασφαλής και πραγματική πίστωση χρημάτων από το χρήστη. Ίσως να είναι δυνατό να αγοραστεί από το Πανεπιστήμιο λογισμικό από την εταιρία JCC για να ενσωματωθεί στην εφαρμογή.

Τέλος, καλό θα ήταν όταν εγγράφεται ο χρήστης, να αποστέλλεται ένα μήνυμα στο email του για πιστοποίηση του email του. Επίσης, ως επέκταση του προηγούμενου σημείου να υπάρχει και η δυνατότητα σε περίπτωση που ξεχάσει τον κωδικό του ο χρήστης να του να αποστέλλεται στο email ο κωδικός του.

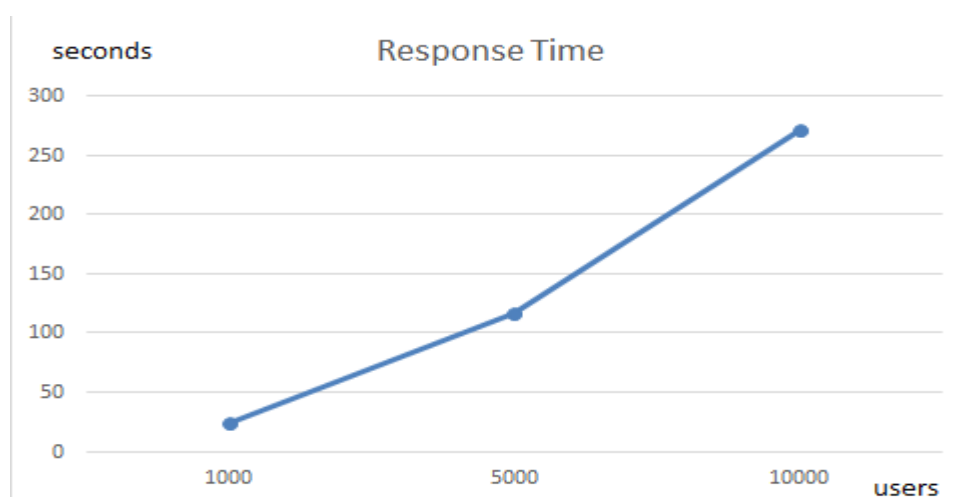
5.2 Συμπεράσματα

Ο μεγάλος αριθμός διαθέσιμων εφαρμογών ηλεκτρονικών καταστημάτων με αρχιτεκτονική Client-Server με οδήγησε στην δημιουργία μιας εφαρμογής διαδικτύου με αρχιτεκτονική Peer-to-Peer η οποία δίνει την δυνατότητα στους χρήστες της να αγοράσουν ή να πουλήσουν προϊόντα με απλό και γρήγορο τρόπο.

Αρχικά, έγινε η ανάπτυξη της εφαρμογής διαδικτύου σε Android με την βοήθεια του περιβάλλοντος Android Studio. Στην συνέχεια, υλοποιήθηκε ένας πολυνηματικός εξυπηρετητής σε Java στο Eclipse ο οποίος έχει σκοπό να ικανοποιήσει όλες τις ανάγκες των χρηστών της εφαρμογής και να συνενώσει την εφαρμογή με την βάση δεδομένων. Τέλος, αναπτύχθηκε μια βάση δεδομένων μέσα στην οποία φυλάγονται τα δεδομένα των χρηστών και των προϊόντων και την οποία διαχειρίζεται ο εξυπηρετητής.

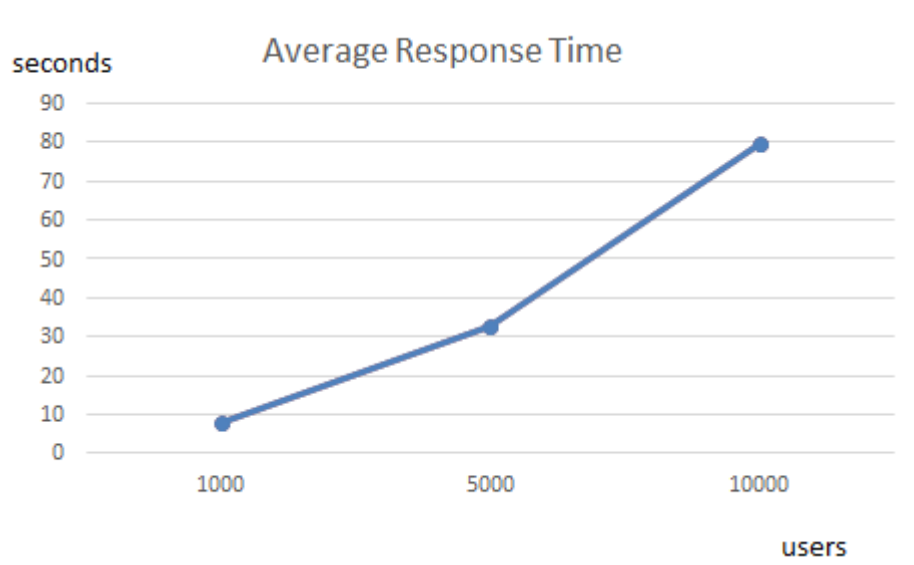
Έχω αναπτύξει μια λειτουργική εφαρμογή η οποία ανταποκρίθηκε σε υψηλό βαθμό σε όλα τα σενάρια για τα οποία την δοκίμασα. Εξυπηρέτησε χωρίς να παρατηρηθεί αισθητοί μύωση στον χρόνο απόκρισης περισσότερους από ένα ενωμένους χρήστες ταυτόχρονα.

Για την αξιολόγηση της απόδοσης του εξυπηρετητή έχω δημιουργήσει δύο γραφικές στις οποίες ο εξυπηρετητής δέχεται αιτήματα από 1000 ,5000 και 10000 πελάτες. Κάθε πελάτης αποστέλλει ένα μόνο αίτημα για τον λόγο ότι ήθελα να προσομοιάσω τις συνθήκες με τις οποίες θα εκτελείται οι εφαρμογή στην οποία ο χρήστης για κάθε λειτουργία δημιουργεί νέα σύνδεση και αποστέλλει ένα μόνο αίτημα.



Σχήμα 5.1 Στο πιο πάνω σχήμα παρουσιάζεται η γραφική του Response Time του εξυπηρετητή.

Στην πρώτη γραφική οι οποία παρουσιάζεται στο σχήμα 5.1 έχει στον κάθετο άξονα τον μέσο όρο των χρόνων που χρειάστηκε ο εξυπηρετητής για να απαντήσει στα αιτήματα των πελατών. Στον οριζόντιο άξονα αναγράφονται ο αριθμός των πελατών τους οποίους εξυπηρέτησε ο εξυπηρετητής. Το Response Time είναι ο χρόνος που χρειάζεται για να εξυπηρετήσει συγκεκριμένο αριθμό αιτημάτων.



Σχήμα 5.2 Στο πιο πάνω σχήμα παρουσιάζεται η γραφική του Communication Latency του εξυπηρετητή.

Στην δεύτερη γραφική οι οποία παρουσιάζεται στο σχήμα 5.2 έχει στον κάθετο άξονα τον μέσο όρο των χρόνων που χρειάστηκε ο πελάτης για να αποστείλει το αίτημα του και να παραλάβει την απάντηση του. Στον οριζόντιο άξονα αναγράφονται ο αριθμός των πελατών τους οποίους εξυπηρέτησε ο εξυπηρετητής. Το Communication Latency είναι ο χρόνος από την στιγμή που ο πελάτης θα αποστείλει ένα νέο αίτημα μέχρι την στιγμή που θα παραλάβει την απάντηση του εξυπηρετητή.

Βιβλιογραφία

- [1] <https://www.android.com/>
- [2] https://en.wikipedia.org/wiki/Android_%28operating_system%29
- [3] <https://eclipse.org/>
- [4] https://en.wikipedia.org/wiki/Eclipse_%28software%29
- [5] <http://developer.android.com/tools/studio/index.html>
- [6] <https://en.wikipedia.org/wiki/MySQL>
- [7] <https://en.wikipedia.org/wiki/Peer-to-peer>
- [8] <http://www.makeuseof.com/tag/p2p-peer-peer-file-sharing-works/>
- [9] http://www.teach-ict.com/gcse_new/networks/peer_peer/miniweb/pg5.htm
- [10] https://en.wikipedia.org/wiki/Client%E2%80%93server_model
- [11] http://www.teach-ict.com/gcse_new/networks/peer_peer/miniweb/pg3.htm

- [12] <http://www.ianswer4u.com/2011/05/client-server-network-advantages-and.html#axzz47DA7UGCT>
- [13] <http://tartarus.org/martin/PorterStemmer/java.txt>