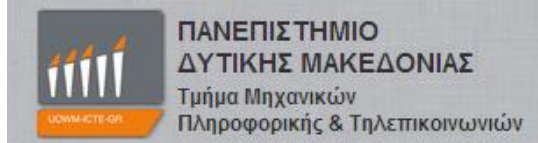


2013

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Καψάλης Αθανάσιος



Επιβλέπων καθηγητής: Μηνάς Δασυγένης

[ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ ΚΑΙ ΕΛΕΓΧΟΥ ΠΩΛΗΤΩΝ ΚΑΙ ΔΗΜΙΟΥΡΓΙΑ ΕΦΑΡΜΟΓΗΣ ΣΕ ΕΞΥΠΝΟ ΚΙΝΗΤΟ]

Κοζάνη, Οκτώβριος 2013

Περιεχόμενα

Περιεχόμενα.....	3
Κατάλογος εικόνων.....	7
Κατάλογος Πινάκων	9
Κατάλογος σχημάτων	11
Ευχαριστίες.....	13
Περίληψη	15
Abstract.....	17
Διάρθρωση Κειμένου.....	19
Κεφάλαιο 1	21
Εισαγωγή.....	21
1.1 Έξυπνα Κινητά	21
1.2 Εφαρμογές.....	21
1.3 Λειτουργικά συστήματα	23
1.4 Android, IOS, Windows Phone	24
1.5 Δημιουργία ιδέας της διπλωματικής.....	25
Κεφάλαιο 2	27
Ανάλυση και σχεδίαση.....	27
2.1 Αρχιτεκτονική.....	27
2.2 Ρόλοι του συστήματος.....	28
2.3 Απαιτήσεις.....	29
2.4 Περιπτώσεις χρήσης	30
2.4.1 Εισαγωγή/Διαγραφή/Επεξεργασία χρηστών	30
2.4.2 Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή πωλητών	31
2.4.3 Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή παραγγελιών	32
2.4.5 Εύρεση θέσης πωλητή	33
2.4.5 Εύρεση διαδρομής που ακολούθησε ο πωλητής.....	34
2.4.6 Έλεγχος σημείων στάσης του πωλητή.....	34

2.5 Βάση δεδομένων.....	35
2.5.1 Πίνακας Users.....	36
2.5.2 Πίνακας Apothiki.....	39
2.5.3 Πίνακας Salesman.....	40
2.5.4 Πίνακας Orders.....	42
2.5.5 Πίνακας Orderdetails.....	44
2.5.5 Πίνακας Gps.....	45
2.6 Πλατφόρμα ανάπτυξης βάσης δεδομένων.....	47
2.7 Διάγραμμα οντοτήτων συσχετίσεων.....	48
2.7 Προσχέδιο βάσης δεδομένων.....	49
Κεφάλαιο 3.....	51
Σχεδιασμός του ιστοχώρου.....	51
3.1 Εγγραφή χρήστη.....	51
3.2 Σύνδεση-αποσύνδεση χρήστη.....	52
3.2.1 Σημαντικά σημεία κώδικα.....	53
3.3 Αλλαγή κωδικού.....	54
3.3.1 Σημαντικά σημεία κώδικα.....	54
3.4 Προϊόντα.....	55
3.4.1 Σημαντικά σημεία κώδικα.....	57
3.5 Πωλητές.....	57
3.5.1 Σημαντικά σημεία κώδικα.....	59
3.6 Παραγγελίες.....	61
3.6.1 Σημαντικά σημεία κώδικα.....	63
3.7 Εντοπισμός θέσης.....	64
3.7.1 Σημαντικά σημεία κώδικα.....	66
3.8 Διαχειριστής.....	68
3.9 Στοιχεία κώδικα ιστοχώρου.....	68
3.10 Ασφάλεια.....	69
Κεφάλαιο 4.....	71
Android εφαρμογή.....	71
4.1 Εισαγωγικά.....	71

4.1.1 Συσκευή Smartphone	71
4.2 Ανατομία μίας εφαρμογής Android.....	73
4.2.1 Κατάλογοι και αρχεία μίας εφαρμογής.....	73
4.2.2 Activity	74
4.2.3 Service.....	75
4.3 Προγραμματισμός-Λειτουργία της εφαρμογής	76
4.3.1 Είσοδος χρήστη.....	76
4.3.2 Κεντρικό μενού.....	79
4.3.3 Λίστα παραγγελιών	80
4.3.4 Καταχώρηση αλλαγών-Διαγραφή παραγγελιών.....	82
4.3.5 Νέα παραγγελία	83
4.4 Στοιχεία κώδικα εφαρμογής	84
Κεφάλαιο 5	85
Τεχνολογίες που χρησιμοποιήθηκαν	85
5.1 Html	85
5.1.1 Ιστορική αναδρομή	86
5.2 CSS	87
5.2.1 Ιστορική αναδρομή	88
5.3 Php.....	89
5.3.1 Ιστορική αναδρομή	89
5.4 MySql	90
5.4.1 Ιστορική αναδρομή	91
5.5 JavaScript.....	92
5.5.1 Ιστορική αναδρομή	93
5.6 Xampp	93
5.7 FreeBsd.....	94
5.7.1 Ιστορική αναδρομή	94
5.8 Notepad++	95
5.8.1 Ιστορική αναδρομή	96
5.9 Google Maps-API.....	97
5.9.1 Ιστορική αναδρομή	97

5.10 Android.....	98
5.10.1 Ιστορική αναδρομή	98
5.11 Eclipse	99
5.11.1 Ιστορική αναδρομή	99
5.12 http://zafora.ict.e.uowm.gr	100
5.13 JAVA.....	100
5.13.1 Ιστορικά	100
Κεφάλαιο 6	103
Επίλογος.....	103
6.1 Συμπεράσματα.....	103
6.2 Μελλοντικά	104
6.3 Συνεργασίες.....	104
Βιβλιογραφία	106

Κατάλογος εικόνων

Εικόνα 1.1: Λειτουργικά συστήματα έξυπνων κινητών	24
Εικόνα 1.2: Πωλήσεις Smartphone παγκοσμίως	25
Εικόνα 2.3: Login	24
Εικόνα 2.4: Διαθέσιμες Επιλογές	31
Εικόνα 2.5: Διαγραφή πωλητή.....	32
Εικόνα 2.6: Ημερολόγιο	33
Εικόνα 2.7: Εμφάνιση στοιχείων παραγγελίας.....	33
Εικόνα 3.1: Επικοινωνία.....	51
Εικόνα 3.2: Εγγραφή χρήστη.....	52
Εικόνα 3.3: Login	53
Εικόνα 3.4: Προϊόντα	55
Εικόνα 3.5: Έλεγχος σωστής εισαγωγής στοιχείων.....	56
Εικόνα 3.6: Πωλητές.....	58
Εικόνα 3.6: Εγγραφή πωλητή	58
Εικόνα 3.7: Παραγγελίες πωλητών.....	59
Εικόνα 3.8: Προϊόντα παραγγελιών.....	59
Εικόνα 3.9: Παραγγελίες	61
Εικόνα 3.10: Εγγραφή νέας παραγγελίας	62
Εικόνα 3.11: Στοιχεία παραγγελίας	63
Εικόνα 3.12: Εντοπισμός θέσης.....	65
Εικόνα 3.13: Τελευταία γνωστή θέση	58
Εικόνα 3.14: Σημεία τελευταίας μέρας.....	66
Εικόνα 4.1: Samsung Galaxy S4.....	72
Εικόνα 4.2: Επικοινωνία Android-Mysql Database	77
Εικόνα 4.3: Login	80
Εικόνα 4.4: Main Menu	71
Εικόνα 4.5: Διαγραφή παραγγελίας	71
Εικόνα 4.6: Λίστα παραγγελιών	83
Εικόνα 4.7: Προσθήκη παραγγελίας	74
Εικόνα 4.8: Προσθήκη στοιχείων παραγγελίας.....	84
Εικόνα 5.1: Το λογότυπο της Html.....	86
Εικόνα 5.2: Παράδειγμα Html	86
Εικόνα 5.3: Παράδειγμα μορφοποίησης με CSS.....	88
Εικόνα 5.4: Το λογότυπο της Php.....	90
Εικόνα 5.5: Επιλογή στοιχείων πίνακα με PHP.....	90

Εικόνα 5.6: Το λογότυπο της MySql	92
Εικόνα 5.7: Δημιουργία πίνακα με MySql	92
Εικόνα 5.8: Script δημιουργίας κουμπιού για εμφάνιση ημερομηνίας με JavaScript	93
Εικόνα 5.9: Xampp control panel	94
Εικόνα 5.10: FreeBSD Logo	95
Εικόνα 5.11: FreeBSD.....	95
Εικόνα 5.12: Notepad++ Logo	96
Εικόνα 5.13: Περιβάλλον Notepad++	96
Εικόνα 5.14: Google Maps Logo	98
Εικόνα 5.15: Android logo.....	99
Εικόνα 5.16: Java Logo	101

Κατάλογος Πινάκων

Πίνακας 2.1: Users.....	37
Πίνακας 2.2: Apothiki.....	39
Πίνακας 2.3: Salesman.....	41
Πίνακας 2.4: Orders	42
Πίνακας 2.5: Orderdetails	44
Πίνακας 2.6: Gps	45

Κατάλογος σχημάτων

Σχήμα 2.1: Διάγραμμα οντοτήτων συσχετίσεων	48
Σχήμα 2.2: Προσχέδιο βάσης δεδομένων	49

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον κ. Δασυγένη Μηνά για την εμπιστοσύνη που μου έδειξε δίνοντας μου την δυνατότητα να ασχοληθώ με θέμα της επιλογής μου και για την πολύτιμη βοήθεια κατά την διάρκεια της συγγραφής και δημιουργίας.

Επίσης ευχαριστώ πολύ τους γονείς μου που ήταν δίπλα μου κάθε στιγμή και τους φίλους μου για την υποστήριξη που μου έδειξαν.

Καψάλης Αθανάσιος

Περίληψη

Για να μπορέσει μία εταιρία να εξελιχθεί πρέπει να συμβαδίζει με την τεχνολογία. Στην παρούσα διπλωματική δίνεται η δυνατότητα σε μία εταιρία να παρακολουθεί τις παραγγελίες που γίνονται από τους πωλητές αλλά και τους ίδιους τους πωλητές και την αποθήκη μέσα από μία ιστοσελίδα. Επίσης χρησιμοποιώντας την εφαρμογή σε έξυπνο κινητό με λειτουργικό Android δίνεται η δυνατότητα στον πωλητή να καταγράφει άμεσα τις παραγγελίες και να τις διαχειρίζεται.

Λέξεις κλειδιά: Smartphone, παρακολούθηση πωλητή, παραγγελίες, πωλητές, GPS, Android, ιστοσελίδα.

Abstract

In order for a company to evolve it must keep its pace with the current technology. This thesis enables a company to keep track of its orders and its salesmen, and have a live result of product availability through a website. Moreover using an Android application it enables the salesman to sent live orders to the company and manage them.

Keywords: Smartphone, salesman tracking, orders, salesmen, GPS, Android, Website.

Διάρθρωση Κειμένου

Στο Κεφάλαιο 1 γίνεται μία αναφορά στην τεχνολογία των έξυπνων κινητών και τον λόγο επιλογής της παρούσας διπλωματικής.

Στο Κεφάλαιο 2 αναφέρεται ο τρόπος σχεδιασμού της βάσης δεδομένων και αναλύεται το σύστημα για να κατανοήσουμε πως φτάσαμε στο τελικό αποτέλεσμα.

Το Κεφάλαιο 3 αναλύει την λειτουργία της διαδικτυακής εφαρμογής και περιέχει σημαντικά σημεία κώδικα.

Στο Κεφάλαιο 4 παρουσιάζεται η εφαρμογή σε έξυπνη συσκευή με λειτουργικό Android, ο τρόπος υλοποίησής της και οι δυνατότητες που παρέχει στον χρήστη.

Το Κεφάλαιο 5 αναλύει τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής.

Το Κεφάλαιο 6 αποτελεί σύνοψη της διπλωματικής και παρουσιάζονται τα συμπεράσματα και οι μελλοντικές εξελίξεις.

Κεφάλαιο 1

Εισαγωγή

Σε αυτό το κεφάλαιο θα γίνει μία εισαγωγή στα έξυπνα τηλέφωνα και τις εφαρμογές και θα αναλυθεί ο λόγος που επιλέχθηκε το συγκεκριμένο θέμα.

1.1 Έξυπνα Κινητά

Το 2007 έκανε την εμφάνιση του το πρώτο έξυπνο κινητό. Μια συσκευή που δημιουργήθηκε γύρω από ένα λειτουργικό σύστημα με μεγαλύτερες υπολογιστικές δυνατότητες από τα συμβατικά κινητά. Τα πρώτα έξυπνα κινητά είχαν την δυνατότητα αναπαραγωγής αρχείων ήχου και μουσικής. Στα επόμενα μοντέλα προστέθηκαν ψηφιακές κάμερες και συστήματα πλοήγησης. Τα τελευταίας τεχνολογίας έξυπνα κινητά έχουν δυνατότητες αντίστοιχες ενός υπολογιστή.

1.2 Εφαρμογές

Αυτό όμως που αξίζει να σημειωθεί με την εμφάνιση των έξυπνων κινητών είναι η εμφάνιση των εφαρμογών για κινητά. Μια εφαρμογή για κινητά είναι μια εφαρμογή λογισμικού που έχει σχεδιαστεί για να "τρέχει" σε έξυπνα κινητά και άλλες φορητές συσκευές. Συνήθως είναι διαθέσιμες μέσω των πλατφόρμων διανομής εφαρμογών, όπως τα Apple App Store, Google Play και Windows Phone Store. Σχεδιάζονται έχοντας ως στόχο ένα συγκεκριμένο λειτουργικό σύστημα όπως Android, Windows Phone ή iPhone αλλά μερικές φορές μπορούν να χρησιμοποιηθούν και σε φορητούς ή επιτραπέζιους υπολογιστές.

Υπάρχουν 2 ειδών εφαρμογές οι online εφαρμογές και οι εφαρμογές σε έξυπνα κινητά.

Online εφαρμογές

Αποτελεί ταχύτερο, φθηνότερο και πιο αποτελεσματικό τρόπο για την ανάπτυξη λογισμικού σε μία επιχείρηση. Αντί να αγοραστεί άδεια χρήσης λογισμικού και να γίνει εγκατάσταση σε διακομιστές ή τοπικούς υπολογιστές μπορεί απλά να γίνει μία εγγραφή και σε λίγα λεπτά να είναι δυνατή η χρήση του λογισμικού. Συνήθως η χρήση είναι με μηνιαία εγγραφή και δεν χρειάζονται λεφτά στην αρχή για την αγορά της άδειας χρήσης.

Εφαρμογές σε έξυπνα κινητά

Επεκτείνουν την εμβέλεια και την παραγωγικότητα μίας επιχείρησης. Αφού γίνει εγκατάσταση της εφαρμογής σε μία κινητή έξυπνη συσκευή μπορούν να γίνουν όλων των ειδών λειτουργίες εκτός γραφείου.

Αρχικά οι εφαρμογές προσφερόταν για ανάκτηση πληροφοριών όπως ημερολόγιο, επαφές και η χρηματιστηριακή αγορά. Ωστόσο όσο μεγάλωνε η ζήτηση σε συνδυασμό με την διαθεσιμότητα εργαλείων για την ανάπτυξη οδήγησε στην ταχεία επέκταση των εφαρμογών και σε άλλες κατηγορίες, όπως παιχνίδια, GPS και υπηρεσίες εύρεσης θέσης, τράπεζες και αγορές εισιτηρίων. Η έκρηξη σε αριθμό και ποικιλία από εφαρμογές έκανε την ανακάλυψη μία πρόκληση. Σύμφωνα με την ABI Research η αγορά εφαρμογών θα έχει αξία 27 δισεκατομμύρια δολάρια στο τέλος του 2013. Στην Ευρωπαϊκή Ένωση δημιουργεί έσοδα άνω των 10 δισεκατομμυρίων ευρώ ετησίως με πάνω από 529.000 θέσεις εργασίας να έχουν δημιουργηθεί λόγω της ανάπτυξης της αγοράς των εφαρμογών.

Οι πιο γνωστές ηλεκτρονικές αγορές εφαρμογών είναι:

- App Store
- BlackBerry World
- Google Play
- Nokia Store
- Windows Phone Store
- Windows Store
- Samsung Apps Store

1.3 Λειτουργικά συστήματα

Λειτουργικό σύστημα ονομάζεται στην επιστήμη της πληροφορικής το λογισμικό του υπολογιστή που είναι υπεύθυνο για τη διαχείριση και τον συντονισμό των εργασιών, καθώς και την κατανομή των διαθέσιμων πόρων. Το λειτουργικό σύστημα παρέχει ένα θεμέλιο, ένα μεσολαβητικό επίπεδο λογικής διασύνδεσης μεταξύ λογισμικού και υλικού, διαμέσου του οποίου οι εφαρμογές αντιλαμβάνονται εμμέσως τον υπολογιστή. Μια από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση του υλικού, απαλλάσσοντας έτσι το λογισμικό του χρήστη από τον άμεσο και επίπονο χειρισμό του υπολογιστή και καθιστώντας ευκολότερο τον προγραμματισμό τους. Σχεδόν όλοι οι υπολογιστές (παλάμης, επιτραπέζιοι ακόμη και παιχνιδιομηχανές) χρησιμοποιούν έναν τύπο λειτουργικού συστήματος. Ορισμένα παλαιότερα μοντέλα ωστόσο βασίζονται σε ένα ενσωματωμένο λειτουργικό σύστημα, το οποίο περιέχεται σε έναν οπτικό δίσκο ή άλλες συσκευές αποθήκευσης δεδομένων.

Τα λειτουργικά συστήματα (OS) των κινητών τηλεφώνων που χρησιμοποιούνται από τα σύγχρονα smartphone μπορούν να εγκατασταθούν σε πολλά διαφορετικά μοντέλα κινητών τηλεφώνων και συνήθως κάθε συσκευή μπορεί να λάβει πολλές ενημερωμένες εκδόσεις λογισμικού λειτουργικού συστήματος κατά τη διάρκεια ζωής της.

Τα πιο γνωστά λειτουργικά συστήματα Smartphone είναι:

- Android της Google
- Symbian της Nokia
- iOS της Apple
- BlackBerry της RIM
- WebOS της Hewlett-Packard
- Windows Phone της Microsoft
- Bada της Samsung



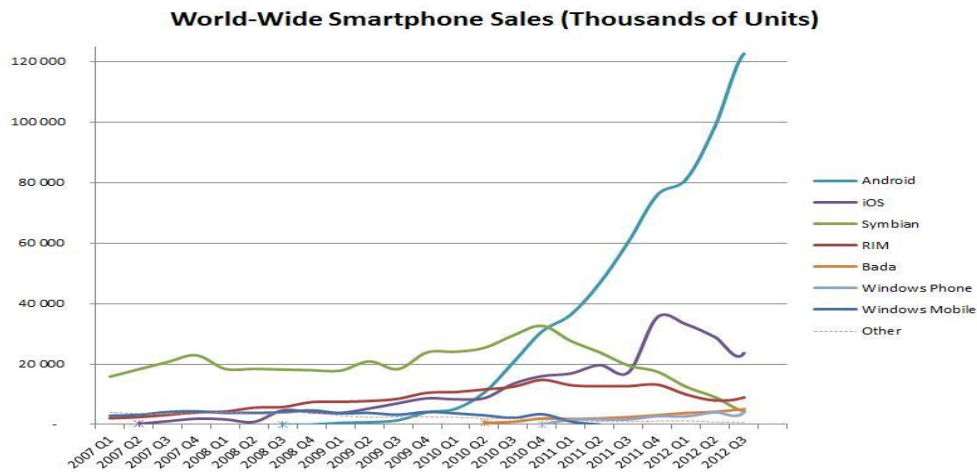
Εικόνα 1.1 Λειτουργικά συστήματα έξυπνων κινητών

Μερικά άλλα επερχόμενα λειτουργικά συστήματα είναι το Firefox OS της Mozilla, το Ubuntu Phone της Canonical Ltd's και το Tizen.

1.4 Android, IOS, Windows Phone

Τα τελευταία χρόνια δύο είναι τα λειτουργικά που μάχονται για την πρώτη θέση, το Android και το IOS της Google και της Apple αντίστοιχα με τα Windows Phone της Microsoft να ακολουθούν. Η επιλογή ενός προγραμματιστή για το λειτουργικό το οποίο θα χρησιμοποιήσει γίνεται με κριτήρια καθαρά υποκειμενικά. Μπορεί κάποιες εφαρμογές να έχουν μεγαλύτερο αντίκτυπο σε ένα συγκεκριμένο λειτουργικό και η δημιουργία τους να είναι πολύ πιο εύκολη, αλλά αυτό δεν απαγορεύει την δημιουργία τους σε κάποιο άλλο. Είναι μάταιο να προσπαθεί κανείς

στην τεχνολογία να βρει το καλύτερο, όλες οι προτάσεις που υπάρχουν κρίνονται σύμφωνα με το πόσο ικανοποιούν τις ανάγκες του κάθε χρήστη ξεχωριστά.



Εικόνα 1.2 Πωλήσεις Smartphone παγκοσμίως

1.5 Δημιουργία ιδέας της διπλωματικής

Η ιδέα της διπλωματικής εργασίας δημιουργήθηκε κατά την περίοδο εργασίας μου σε μία εταιρία η οποία προωθούσε τα προϊόντα της με πωλητές. Τα προβλήματα που παρουσιαζόταν καθημερινά ήταν η άμεση καταγραφή των παραγγελιών, η δυνατότητα άμεσου ελέγχου της αποθήκης και καταγραφή της θέσης του πωλητή τόσο για έλεγχο όσο και για καθοδήγηση. Έτσι μετά από συνεννόηση με τον υπεύθυνο της εταιρίας αποφασίστηκε η λειτουργία του συστήματος και ο τρόπος λειτουργίας του.

Κεφάλαιο 2

Ανάλυση και σχεδίαση

Στο κεφάλαιο αυτό γίνεται ανάλυση του συστήματος για να γίνουν κατανοητά τα βήματα με τα οποία φτάσαμε στο τελικό αποτέλεσμα. Γίνεται ανάλυση των απαιτήσεων που περιλαμβάνει τον καθορισμό των αναγκών ή τις προϋποθέσεις για ένα σύστημα. Λαμβάνοντας υπόψη τις απαιτήσεις των χρηστών και τις λειτουργίες του συστήματος οδηγούμαστε στην σωστή σχεδίαση της βάσης δεδομένων.

2.1 Αρχιτεκτονική

Η εφαρμογή αποτελείται από 2 μέρη. Την διαδικτυακή εφαρμογή και την εφαρμογή σε Smartphone με λειτουργικό android. Ο χρήστης(πωλητής) του κινητού σταθμού(Smartphone) μπορεί να στείλει πληροφορίες μέσω της υπηρεσίας GPRS στην βάση δεδομένων και ο χρήστης της διαδικτυακής εφαρμογής μπορεί μέσω ενός τερματικού να δει τις πληροφορίες αυτές. Κάθε χρήστης κινητού σταθμού έχει διαφορετικό id για να ξεχωρίζει αφού η εφαρμογή υποστηρίζει πολλούς χρήστες. Συγχρόνως ανά τακτά χρονικά διαστήματα στέλνονται στοιχεία για την θέση του πωλητή τα οποία μπορεί να δει μόνο ο εγγεγραμμένος χρήστης της διαδικτυακής εφαρμογής.

Ο κινητός σταθμός στέλνει σήματα σε ένα δορυφόρο ο οποίος με την σειρά του στέλνει τα δεδομένα που πήρε στο διαδίκτυο. Τα δεδομένα αυτά μπορεί να είναι είτε πληροφορίες για την θέση του είτε μία παραγγελία που καταχώρησε ο χρήστης. Ότι είδος και να είναι τα δεδομένα προωθούνται σε ένα εξυπηρετητή που είναι αποθηκευμένη η βάση δεδομένων και καταχωρούνται σε αυτήν. Ο χρήστης της διαδικτυακής εφαρμογής μπορεί να δει τις πληροφορίες αυτές μέσω ενός τερματικού παρακολούθησης.

2.2 Ρόλοι του συστήματος

Αρχίζοντας τον σχεδιασμό καταγράφουμε όλους τους χρήστες που θα μπορούν να αλληλεπιδράσουν με το σύστημα. Καταλήγουμε στους 4 που ακολουθούν:

- Διαχειριστής/admin
- Εγγεγραμμένος χρήστης
- Ανώνυμος Χρήστης
- Πωλητής

Αναλύουμε κάθε μία από τις οντότητες ξεχωριστά παρακάτω.

Ο διαχειριστής/admin είναι η οντότητα που έχει τις περισσότερες δυνατότητες στο σύστημα. Μπορεί να προσθέτει και να αφαιρεί χρήστες και να τροποποιεί τα στοιχεία τους. Είναι αυτός που ελέγχει την σωστή λειτουργία του συστήματος. Όμως δεν αλληλεπιδρά με το σύστημα τόσο συχνά.

Ο εγγεγραμμένος χρήστης έχει τον βασικότερο ρόλο στο σύστημα. Είναι αυτός που αλληλεπιδρά πιο συχνά με το σύστημα και αντλεί τις περισσότερες πληροφορίες από αυτό. Χρησιμοποιεί το σύστημα περισσότερο από όλες τις οντότητες

Ο ανώνυμος χρήστης έχει περιορισμένες δυνατότητες καθώς μπορεί απλά να ενημερωθεί για το σύστημα χωρίς να μπορεί να αλληλεπιδράσει με αυτό. Μπορεί να έρθει σε επαφή με έναν διαχειριστή και να γίνει και αυτός ένας εγγεγραμμένος χρήστης σε περίπτωση επιθυμίας του να χρησιμοποιήσει το σύστημα.

Ο πωλητής μπορεί να λαμβάνει συγκεκριμένες πληροφορίες από το σύστημα, μέσω της android εφαρμογής και να στέλνει επίσης συγκεκριμένες πληροφορίες σε αυτό. Ο πωλητής σαν χρήστης μπορεί μόνο να δει τα διαθέσιμα προϊόντα στην αποθήκη και να καταχωρήσει μία νέα παραγγελία ή να επεξεργαστεί μια που αυτός καταχώρησε.

2.3 Απαιτήσεις

Γνωρίζοντας πλέον ποιοι μπορούν να χρησιμοποιήσουν το σύστημα και να αλληλεπιδράσουν με αυτό καθορίζουμε και τις απαιτήσεις που έχει κάθε μία από τις οντότητες από το σύστημα.

Ο διαχειριστής μπορεί

- Να κάνει Εισαγωγή/Διαγραφή/Επεξεργασία χρηστών.
- Ότι μπορεί να κάνει ο εγγεγραμμένος χρήστης

Ο εγγεγραμμένος χρήστης μπορεί

- Να κάνει Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή πωλητών.
- Να κάνει Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή προϊόντων.
- Να κάνει Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή παραγγελιών
- Να βρει την θέση ενός πωλητή.
- Να βρει την διαδρομή που ακολούθησε ο πωλητής.
- Να ελέγξει στοιχεία σχετικά με την διαδρομή του πωλητή(χρόνο στάσης σε ένα σημείο-απόσταση που διήνυσε σε μία εργάσιμη ημέρα).
- Να επικοινωνήσει με τον διαχειριστή για βοήθεια.

Ο ανώνυμος χρήστης μπορεί

- Να επικοινωνήσει με τον διαχειριστή.

Ο πωλητής μπορεί

- Να κάνει εισαγωγή νέας παραγγελίας.
- Εύρεση παλαιάς παραγγελίας και τροποποίησή της.

2.4 Περιπτώσεις χρήσης

Αναλύονται παρακάτω οι πιο σημαντικές περιπτώσεις χρήσης.

2.4.1 Εισαγωγή/Διαγραφή/Επεξεργασία χρηστών

Χρήστης

Διαχειριστής

Περιγραφή

Ο διαχειριστής μπορεί να αλλάξει στοιχεία εγγεγραμμένων χρηστών, να διαγράψει εγγεγραμμένους χρήστες, ή να προσθέσει ένα καινούριο χρήστη.

Προϋποθέσεις

Να εισάγει σωστό κωδικό και όνομα χρήστη για να μπορέσει να μπει στο σύστημα ελέγχου του διαχειριστή και να επιλέξει την αντίστοιχη λειτουργία(edit-delete-add new user).

Σενάριο

Ένας ανώνυμος χρήστης ενδιαφέρθηκε για την εφαρμογή και έρχεται σε επαφή με το διαχειριστή. Ο διαχειριστής εισάγει έναν νέο χρήστη με όλα τα απαιτούμενα στοιχεία.

Αποτέλεσμα

Ο χρήστης πλέον είναι εγγεγραμμένος χρήστης και μπορεί να χρησιμοποιήσει την εφαρμογή.

2.4.2 Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή πωλητών

Χρήστης

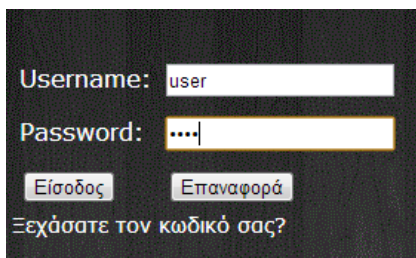
Εγγεγραμμένος χρήστης

Περιγραφή

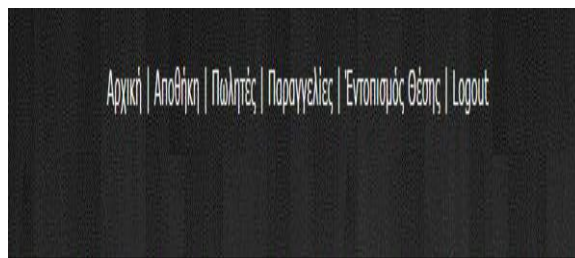
Η περίπτωση χρήσης αυτή επιτρέπει στον χρήστη να βρει έναν πωλητή του, να επεξεργαστεί τα στοιχεία του, να εισάγει ένα καινούριο ή να διαγράψει έναν ήδη υπάρχον.

Προϋποθέσεις

Να εισάγει το κωδικό του και το όνομα χρήστη για να μπει στο σύστημα που περιέχει τα στοιχεία του.(Εικόνα 2.1) Να επιλέξει την καρτέλα πωλητές(Εικόνα 2.2).



Εικόνα 2.2: Login



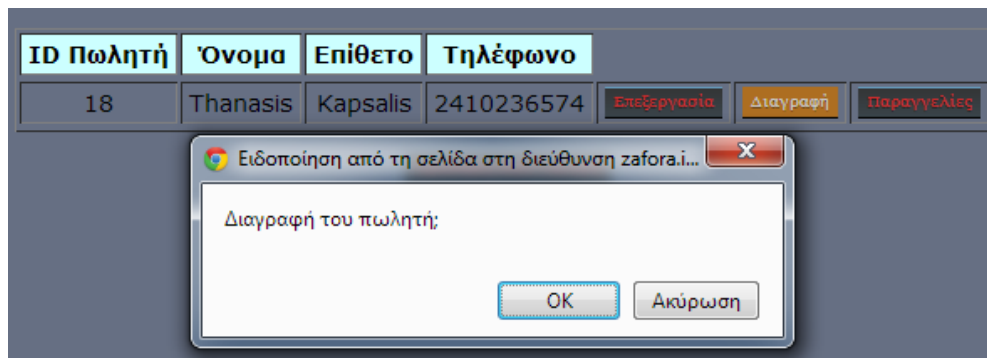
Εικόνα 2.4: Διαθέσιμες Επιλογές

Σενάριο

Ο χρήστης θέλει να διαγράψει έναν πωλητή με τον οποίο έληξε την συνεργασία του(Εικόνα 2.3).

Αποτελέσματα

Ο πωλητής δεν υπάρχει πλέον στην βάση δεδομένων.



Εικόνα 2.3: Διαγραφή πωλητή

2.4.3 Εύρεση/Επεξεργασία/Διαγραφή/Εισαγωγή παραγγελιών

Χρήστης

Εγγεγραμμένος χρήστης

Περιγραφή

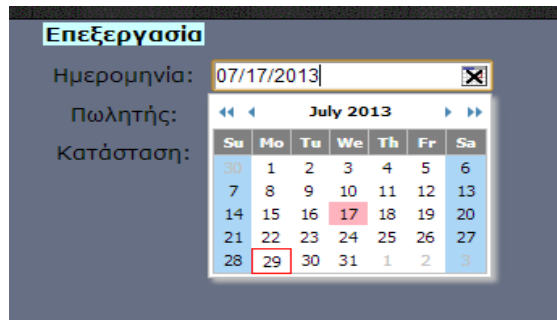
Ο εγγεγραμμένος χρήστης μπορεί να αλλάξει τα στοιχεία μιας παραγγελίας, να κάνει εισαγωγή νέας ή να βρει μία παλιά.

Προϋποθέσεις

Να πατήσει τον σύνδεσμο παραγγελίες και να επιλέξει την αντίστοιχη λειτουργία.

Σενάριο

Πρέπει να αλλαχθεί η ημερομηνία μίας παραγγελίας (Εικόνα 2.4).



Εικόνα 2.4: Ημερολόγιο

Αποτέλεσμα

Η παραγγελία είναι πλέον καταχωρημένη με την νέα της ημερομηνία(Εικόνα 2.5).

ID Παραγγελίας	Ημερομηνία ▾	Πωλητής	Κατάσταση	Επεξεργασία	Διαγραφή
215	07/29/2013	Kapsalis	Εκκρεμεί		

Εικόνα 2.5: Εμφάνιση στοιχείων παραγγελίας

2.4.5 Εύρεση θέσης πωλητή

Χρήστης

Εγγεγραμμένος χρήστης

Περιγραφή

Ο εγγεγραμμένος χρήστης μπορεί να γνωρίζει την θέση που βρίσκεται ο πωλητής ανά πάσα στιγμή.

Προϋποθέσεις

Να γίνει επιλογή του πωλητή που επιθυμεί να δει.

Σενάριο

Ο χρήστης θέλει να μάθει που βρίσκεται ο πωλητής μια συγκεκριμένη χρονική στιγμή.

Αποτέλεσμα

Η θέση του πωλητή εμφανίζεται στον χάρτη με ανανέωση ανά 1 λεπτό

2.4.5 Εύρεση διαδρομής που ακολούθησε ο πωλητής

Χρήστης

Εγγεγραμμένος χρήστης

Περιγραφή

Ο εγγεγραμμένος χρήστης μπορεί να δει την διαδρομή που ακολούθησε ο πωλητής.

Προϋποθέσεις

Ο χρήστης της διαδικτυακής εφαρμογής να πατήσει στην καρτέλα εντοπισμός θέσης και να επιλέξει τον αντίστοιχο πωλητή

Σενάριο

Ο χρήστης θέλει να δει την διαδρομή που ακολούθησε ο πωλητής κατά την διάρκεια μίας εργάσιμης ημέρας.

Αποτέλεσμα

Η διαδρομή που ακολούθησε ο πωλητής εμφανίζεται στον χάρτη

2.4.6 Έλεγχος σημείων στάσης του πωλητή

Χρήστης

Εγγεγραμμένος χρήστης

Περιγραφή

Ο εγγεγραμμένος χρήστης μπορεί να δει τα σημεία στα οποία σταμάτησε ο πωλητής

Προϋποθέσεις

Να έχει κάνει login ο χρήστης της εφαρμογής του κινητού, ο χρήστης της διαδικτυακής εφαρμογής να πατήσει στην καρτέλα εντοπισμός θέσης, να επιλέξει τον αντίστοιχο πωλητή και την επιλογή εμφάνιση σημείων στάσης

Σενάριο

Ο εγγεγραμμένος χρήστης θέλει να επαληθεύσει τα σημεία που παρακολούθησε ο πωλητής του

Αποτέλεσμα

Τα σημεία που είδε ο πωλητής εμφανίζονται στον χάρτη με σημειωμένα τα σημεία που έμεινε για χρονικό διάστημα μεγαλύτερο των 15 λεπτών και τα σημεία που έμεινε για διάστημα μικρότερο των 15 λεπτών.

2.5 Βάση δεδομένων

Μία σωστά σχεδιασμένη βάση δεδομένων παρέχει πρόσβαση σε ενημερωμένες ακριβείς πληροφορίες. Επειδή η σωστή σχεδίαση είναι ουσιαστικής σημασίας για την επίτευξη των στόχων, όταν δουλεύουμε με μια βάση δεδομένων, η επένδυση του χρόνου που απαιτείται για την εκμάθηση των αρχών της καλής σχεδίασης είναι σημαντική.

Υπάρχουν ορισμένες αρχές που καθοδηγούν τη διαδικασία σχεδίασης βάσης δεδομένων. Η πρώτη αρχή είναι ότι οι διπλότυπες πληροφορίες(πλεονάζοντα δεδομένα) είναι κακές διότι σπαταλούν χώρο και αυξάνουν την πιθανότητα σφαλμάτων και ασυνεπειών. Η δεύτερη αρχή είναι ότι η ορθότητα και η πληρότητα των πληροφοριών είναι

σημαντικές. Εάν η βάση δεδομένων περιέχει λανθασμένες πληροφορίες οι εκθέσεις που αντλούν πληροφορίες από την βάση δεδομένων θα έχουν επίσης λανθασμένες πληροφορίες. Ως αποτέλεσμα οι όποιες αποφάσεις βασίζονται σε εκθέσεις θα είναι βασισμένες σε παραπληροφόρηση.

Κατά την σχεδίαση πρέπει λοιπόν να προσέξουμε:

- Τον καθορισμό του σκοπού που θα εξυπηρετήσει η βάση δεδομένων.
- Εύρεση και οργάνωση των απαιτούμενων πληροφοριών.
- Χωρισμός των πληροφοριών.
- Μετατροπή των στοιχείων πληροφοριών σε στήλες.
- Καθορισμός πρωτευόντων κλειδίων.
- Δημιουργία σχέσεων πινάκων.
- Βελτίωση της σχεδίασης.
- Εφαρμογή των κανόνων κανονικοποίησης.

Με βάση όλα τα παραπάνω και μετά τον σχεδιασμό των πινάκων σε χαρτί το επόμενο βήμα είναι η δημιουργία τους. Για την δημιουργία της βάσης χρησιμοποιήθηκε το FreeBSD 8.2. Για την δημιουργία του σχεδιαγράμματος οντοτήτων-συσχετίσεων χρησιμοποιήθηκε το xampp.

2.5.1 Πίνακας Users

Ο πίνακας Users δημιουργήθηκε για την καταχώρηση των χρηστών ώστε να έχουν πρόσβαση στο σύστημα.

Στήλη	Τύπος	Κενό	Προεπιλογή
user_id	int(4)	Όχι	Καμία
username	varchar(20)	Όχι	Καμία
password	varchar(40)	Όχι	Καμία
name	varchar(20)	Όχι	Καμία
surname	varchar(20)	Όχι	Καμία
email	varchar(30)	Όχι	Καμία
address	varchar(50)	Όχι	Καμία
city	varchar(15)	Όχι	Καμία
phone	varchar(15)	Όχι	Καμία
tou	int(2)	Όχι	Καμία

Πίνακας 2.1: Users

user_id

Πρωτεύον κλειδί του πίνακα, προστίθεται καινούριο με κάθε νέα εισαγωγή χρήστη. Δόθηκε τύπος Integer(5) γιατί είναι ακέραιος αριθμός

username

Το συνθηματικό που χρησιμοποιεί ο εγγεγραμμένος χρήστης για να χρησιμοποιήσει την εφαρμογή. Είναι ξεχωριστό για κάθε χρήστη. Ο τύπος δεδομένων είναι Varchar(20) γιατί μπορεί να πάρει και γράμματα και αριθμούς.

password

Ο κωδικός του κάθε χρήστη που τον γνωρίζει μόνο αυτός. Είναι αλφαριθμητικό και για αυτό δόθηκε και ο τύπος δεδομένων Varchar(40). Το μέγεθος είναι 40 γιατί χρησιμοποιείται κωδικοποίηση sha1 στον κωδικό η οποία χρειάζεται 40 χαρακτήρες για να αποθηκευτεί.

name

Το όνομα του εγγεγραμμένου χρήστη αποθηκεύεται εδώ. Ο τύπος δεδομένων είναι Varchar(20) γιατί είναι αλφαριθμητικό. Το μήκος είναι 20 γιατί ζητάμε αληθινά ονόματα και τα ελληνικά ονόματα δεν ξεπερνάνε τους 20 χαρακτήρες.

surname

Εδώ αποθηκεύεται το επίθετο του εγγεγραμμένου χρήστη. Ο τύπος δεδομένων είναι Varchar(20) γιατί είναι αλφαριθμητικό.

email

Αποθηκεύει το email του εγγεγραμμένου χρήστη. Είναι ξεχωριστό για κάθε χρήστη και επειδή ο τύπος δεδομένων είναι αλφαριθμητικός χρησιμοποιήθηκε Varchar(30).

address

Αποθηκεύεται η διεύθυνση του εγγεγραμμένου χρήστη. Χρησιμοποιήθηκε Varchar(50) γιατί ο τύπος δεδομένων είναι αλφαριθμητικός.

city

Περιέχει τη πόλη στην οποία εδρεύει ο χρήστης. Χρησιμοποιήθηκε Varchar(15) γιατί ο τύπος δεδομένων είναι αλφαριθμητικός.

phone

Καταγράφει το τηλέφωνο του χρήστη. Χρησιμοποιήθηκε Varchar(15) γιατί ο τύπος δεδομένων είναι αλφαριθμητικός. Το μέγιστο μήκος που μπορεί να πάρει ένα τηλέφωνο είναι 15 σύμφωνα με το E.164 που συντάχθηκε από το ITU-T με έναν από αυτούς τους χαρακτήρες να είναι το (+,-).

του

Είναι ένας αριθμός που χρησιμεύει για να ξεχωρίσουμε τις ιδιότητες που έχει ο χρήστης. Για παράδειγμα όσοι έχουν τιμή από 0 έως 9 είναι διαχειριστές του συστήματος και όλοι οι υπόλοιποι αριθμοί χρησιμοποιούνται για τους απλούς χρήστες. Τύπος δεδομένων είναι Integer(2). Το μέγεθος είναι 2 γιατί θέλουμε να διακρίνουμε αρκετές ομάδες χρηστών ανάλογα με τα δικαιώματα που έχουνε.

2.5.2 Πίνακας Apothiki

Στο πίνακα Apothiki αποθηκεύονται τα προϊόντα που έχει στην αποθήκη ο κάθε εγγεγραμμένος χρήστης και οι πληροφορίες σχετικά με αυτά.

Στήλη	Τύπος	Κενό	Προεπιλογή
product_id	int(5)	Όχι	καμία
user_id	int(4)	Όχι	καμία
product	varchar(20)	Όχι	καμία
quantity	int(4)	Όχι	καμία
barcode	varchar(15)	Όχι	καμία
expire	varchar(10)	Όχι	καμία
description	varchar(50)	Όχι	καμία
timitemaxiou	float	Όχι	καμία

Πίνακας 2.2: Apothiki

product_id

Πρωτεύον κλειδί του πίνακα Apothiki ανεβαίνει κατά ένα με την εισαγωγή νέου προϊόντος. Τύπος δεδομένων είναι Integer(5) καθώς είναι ακέραιος αριθμός.

user_id

Πρωτεύον κλειδί του πίνακα Users. Χρησιμοποιείται σε αυτόν τον πίνακα για να ξεχωρίζουν τα προϊόντα ανά χρήστη. Ο τύπος δεδομένων είναι ίδιος με το user_id του πίνακα Users.

product

Εδώ αποθηκεύεται το όνομα του προϊόντος. Ο τύπος δεδομένων είναι Varchar(20) καθώς είναι αλφαριθμητικό. Το μέγεθος είναι 20 γιατί δεν θέλουμε να αποθηκεύεται και η περιγραφή του προϊόντος αλλά μόνο το όνομα.

quantity

Αποθηκεύεται η διαθέσιμη ποσότητα του προϊόντος στην αποθήκη. Ο τύπος δεδομένων είναι Integer(4) αφού η ποσότητα μετριέται σε αριθμό.

barcode

Κάθε προϊόν έχει έναν ξεχωριστό αναγνωριστικό αριθμό ο οποίος αποθηκεύεται εδώ. Επειδή όμως το barcode μπορεί να περιέχει και γράμματα χρησιμοποιείται τύπος Varchar(15). Το μέγεθος είναι 15 γιατί τόσοι είναι οι μέγιστοι χαρακτήρες που μπορεί να έχει το αναγνωριστικό ενός προϊόντος.

expire

Σχεδόν όλα τα προϊόντα έχουν μια ημερομηνία λήξης. Ο τύπος δεδομένων είναι Varchar(10) γιατί μπορεί η ημερομηνία λήξης να δηλώνεται μόνο με το έτος ή μπορεί και να δηλώνεται με ολόκληρη την ημερομηνία της μορφής MM/HH/EEEE.

description

Στο πεδίο αυτό καταγράφεται μία σύντομη περιγραφή του προϊόντος. Τύπος δεδομένων είναι Varchar(50) γιατί είναι αλφαριθμητικό.

timitemaxiou

Τα προϊόντα έχουν μία τιμή τεμαχίου η οποία καταγράφεται εδώ. Επειδή αυτή η τιμή μπορεί να είναι και δεκαδική επιλέχθηκε ο τύπος δεδομένων Float.

2.5.3 Πίνακας Salesman

Στον πίνακα αυτόν κρατάμε τα στοιχεία των πωλητών που εργάζονται για τον εγγεγραμμένο χρήστη.

Στήλη	Τύπος	Κενό	Προεπιλογή
salesman_id	int(4)	Όχι	καμία
user_id	int(4)	Όχι	καμία
name	varchar(20)	Όχι	καμία
surname	varchar(20)	Όχι	καμία
phone	varchar(14)	Όχι	καμία
username	varchar(20)	Όχι	καμία
password	varchar(40)	Όχι	καμία

Πίνακας 2.3: Salesman

salesman_id

Πρωτεύον κλειδί του πίνακα Salesman αυξάνεται κατά ένα κάθε φορά που εισάγεται νέος πωλητής. Ο τύπος δεδομένων είναι Int(4) γιατί είναι ακέραιος αριθμός.

user_id

Πρωτεύον κλειδί του πίνακα users. Χρησιμοποιείται σε αυτόν τον πίνακα για ως σύνδεση ξένου κλειδιού για να ξεχωρίζουν οι πωλητές του κάθε χρήστη. Ο τύπος δεδομένων ορίζεται από τον πίνακα Users.

name

Το όνομα του πωλητή αποθηκεύεται εδώ. Ο τύπος δεδομένων επιλέχθηκε ως Varchar(20) με την ίδια λογική του name του πίνακα Users.

surname

Αποθηκεύεται το επίθετο του πωλητή. Ο τύπος δεδομένων Varchar(20) είναι ίδιος με το surname του πίνακα Users.

phone

Αποθηκεύεται το τηλέφωνο του πωλητή. Ο τύπος δεδομένων Varchar(14) είναι ίδιος με το phone του πίνακα Users και επιλέχθηκε για τους ίδιους λόγους.

username

Κάθε πωλητής έχει ένα ξεχωριστό όνομα χρήστη για να μπορεί να χρησιμοποιήσει την εφαρμογή. Ο τύπος δεδομένων είναι Varchar(20) γιατί είναι αλφαριθμητικό.

password

Ο κωδικός του πωλητή που τον γνωρίζει μόνο αυτός για να χρησιμοποιήσει την εγγραφή. Χρησιμοποιείται κωδικοποίηση sha1 όπως στο password του πίνακα Users για αυτό ως τύπος δεδομένων επιλέχθηκε Varchar(40).

2.5.4 Πίνακας Orders

Ο πίνακας orders χρησιμοποιείται για την καταγραφή των παραγγελιών

Στήλη	Τύπος	Κενό	Προεπιλογή
orders_id	int(5)	Όχι	καμία
user_id	int(4)	Όχι	καμία
salesman_id	int(4)	Όχι	καμία
date	varchar(10)	Όχι	καμία
salesman	varchar(20)	Όχι	καμία
status	varchar(9)	Όχι	καμία

Πίνακας 2.4: Orders

orders_id

Πρωτεύον κλειδί του πίνακα Orders αυξάνεται αυτόματα κατά ένα με κάθε εισαγωγή νέας παραγγελίας. Ο τύπος δεδομένων είναι Integer(5) γιατί είναι ακέραιος αριθμός.

user_id

Ξένο κλειδί του πίνακα Users. Χρησιμοποιείται για να ξεχωρίζουν οι παραγγελίες το κάθε χρήστη. Ο τύπος δεδομένων είναι αυτός του αρχικού user_id από τον πίνακα Users.

salesman_id

Ξένο κλειδί του πίνακα Salesman. Χρειάζεται για να ξεχωρίζουν οι παραγγελίες που έκανε ο κάθε πωλητής. Ο τύπος δεδομένων ορίζεται από τον πίνακα salesman.

date

Καταγράφεται η ημερομηνία που έγινε η παραγγελία. Ο τύπος δεδομένων είναι Varchar(10) γιατί είναι της μορφής MM/HH/EEEE (όπου M= Μήνας, H= Ημέρα, E= Έτος).

salesman

Καταγράφεται ο πωλητής που έκανε την παραγγελία σύμφωνα με το επίθετο του. Ο τύπος δεδομένων είναι ίδιος με το surname του πίνακα salesman.

status

Κάθε παραγγελία μπορεί να πάρει μία τιμή για να ξεχωρίζει η κατάσταση της. Ο τύπος δεδομένων είναι Varchar(9) καθώς είναι αλφαριθμητικά και το μέγεθος είναι 9 γιατί ο μέγιστος αριθμός χαρακτήρων των επιλογών είναι 9. Οι 4 επιλογές είναι οι εξής:

- Εκκρεμεί
- Ακυρώθηκε
- Πληρώθηκε
- Έναντι

2.5.5 Πίνακας Orderdetails

Ο πίνακας Orderdetails αποτελεί επέκταση του πίνακα orders. Εδώ αποθηκεύονται επιπλέον πληροφορίες σχετικά με τις παραγγελίες(τα προϊόντα που παρήγγειλε και η ποσότητά τους). Χρησιμοποιήθηκε ξεχωριστός πίνακας γιατί σε μία παραγγελία μπορούν να ανήκουν πολλά προϊόντα και η εισαγωγή τους γίνεται δυναμικά κάτι που θα ήταν αδύνατο στον πίνακα Orders.

Στήλη	Τύπος	Κενό	Προεπιλογή
orders_id	int(5)	Όχι	καμία
user_id	int(4)	Όχι	καμία
product	varchar(20)	Όχι	καμία
quantity	int(4)	Όχι	καμία
orderdetails_id	int(5)	Όχι	καμία

Πίνακας 2.5: Orderdetails

orderdetails_id

Πρωτεύον κλειδί του πίνακα Orderdetails. Αυξάνεται κατά ένα αυτόματα με την εισαγωγή νέας παραγγελίας. Ο τύπος δεδομένων είναι Integer(5) γιατί είναι ακέραιος αριθμός.

orders_id

Πρωτεύον κλειδί του πίνακα Orders. Χρησιμοποιείται εδώ για να μπορέσουμε να συσχετίσουμε τις επιπλέον πληροφορίες που αποθηκεύονται στον πίνακα με τις παραγγελίες. Ο τύπος δεδομένων είναι ίδιος με το orders_id του πίνακα Orders.

user_id

Πρωτεύον κλειδί του πίνακα Users. Χρησιμοποιείται για να ξεχωρίσουμε τις παραγγελίες ανάλογα με τον χρήστη. Ίδιος τύπος δεδομένων με το user_id του πίνακα Users.

product

Αποθηκεύονται τα προϊόντα τις παραγγελίας. Ο τύπος δεδομένων είναι varchar(20) από τις τιμές από το πεδίο product του πίνακα Arothiki.

quantity

Αποθηκεύεται η ποσότητα των προϊόντων της παραγγελίας. Ο τύπος δεδομένων είναι integer(4) από το πεδίο quantity του πίνακα Arothiki.

2.5.5 Πίνακας Gps

Στον πίνακα Gps καταχωρούνται τα στοιχεία που θέλουμε σχετικά με την θέση του κάθε πωλητή.

Στήλη	Τύπος	Κενό	Προεπιλογή
id	int(11)	Όχι	καμία
user_id	int(4)	Όχι	καμία
salesman_id	int(4)	Όχι	καμία
latitude	float	Όχι	καμία
logtitude	float	Όχι	καμία
altitude	float	Όχι	καμία
date	date	Όχι	καμία
time	time	Όχι	καμία

Πίνακας 2.6: Gps

id

Πρωτεύον κλειδί του πίνακα Gps. Με κάθε νέα καταχώρηση αυξάνεται κατά ένα. Χρησιμοποιήθηκε τύπος δεδομένων Integer(11) γιατί είναι ακέραιος αριθμός.

user_id

Πρωτεύον κλειδί του πίνακα Users. Χρησιμοποιείται για να ξεχωρίζουν οι χρήστες. Ίδιος τύπος δεδομένων με το αντίστοιχο του πίνακα Users.

salesman_id

Πρωτεύον κλειδί του πίνακα Salesman. Χρησιμοποιείται για να ξεχωρίζουν οι πωλητές. Ίδιος τύπος δεδομένων με το αντίστοιχο του πίνακα Salesman.

latitude

Αποθηκεύεται το γεωγραφικό πλάτος της θέσης του πωλητή. Ο τύπος δεδομένων είναι float γιατί το πλάτος είναι δεκαδικός αριθμός .

longitude

Αποθηκεύεται το γεωγραφικό μήκος της θέσης του πωλητή. Ο τύπος δεδομένων είναι float γιατί το μήκος είναι δεκαδικός αριθμός.

altitude

Αποθηκεύεται το γεωγραφικό υψόμετρο της θέσης του πωλητή. Ο τύπος δεδομένων είναι float γιατί το υψόμετρο είναι δεκαδικός αριθμός.

date

Η ημερομηνία που βρισκόταν στην συγκεκριμένη θέση ο πωλητής αποθηκεύεται εδώ. Ο τύπος δεδομένων είναι date γιατί αποθηκεύεται ημερομηνία.

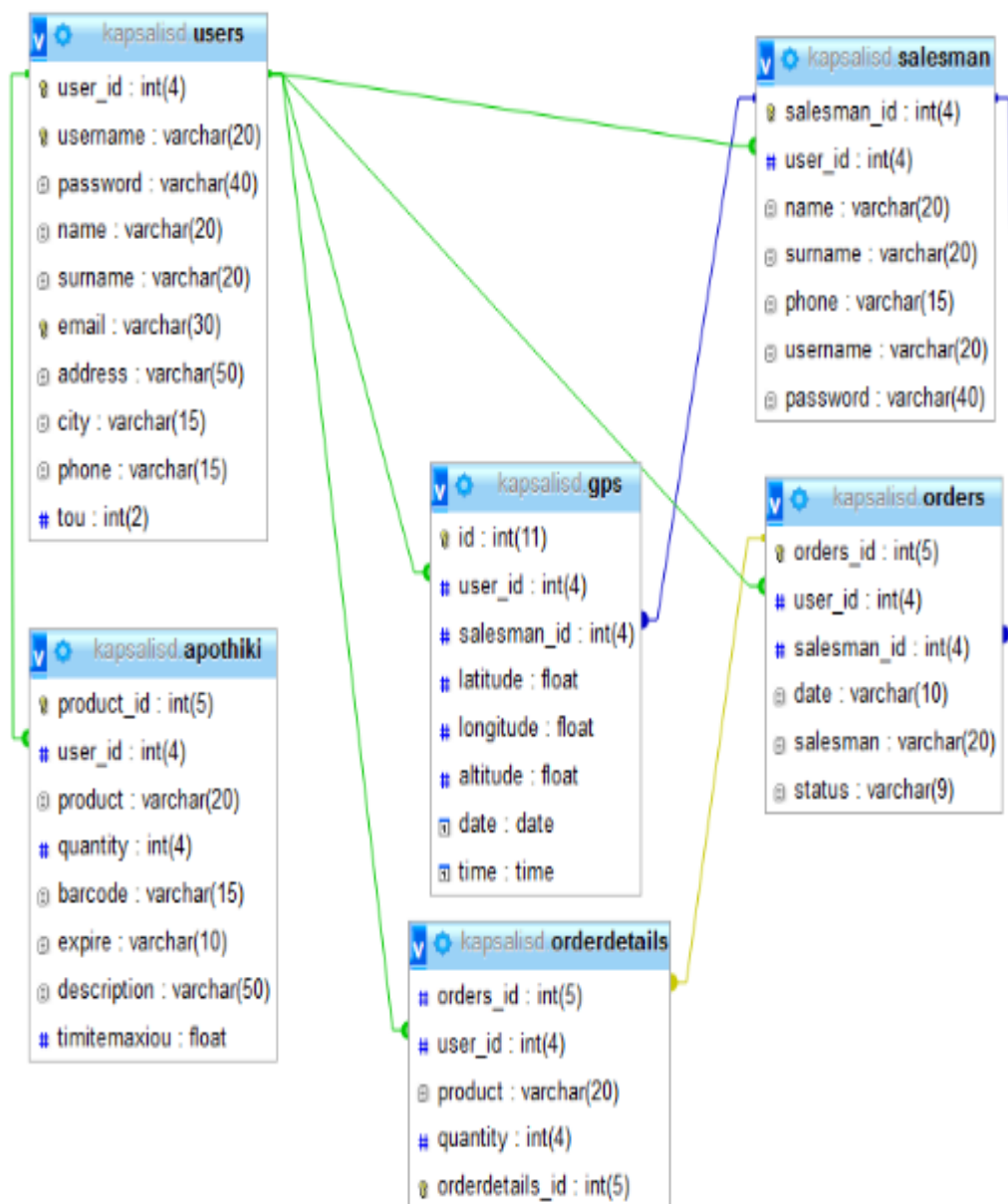
time

Η ώρα που βρισκόταν στην συγκεκριμένη θέση ο πωλητής αποθηκεύεται εδώ. Ο τύπος δεδομένων είναι time γιατί αποθηκεύεται ώρα.

2.6 Πλατφόρμα ανάπτυξης βάσης δεδομένων

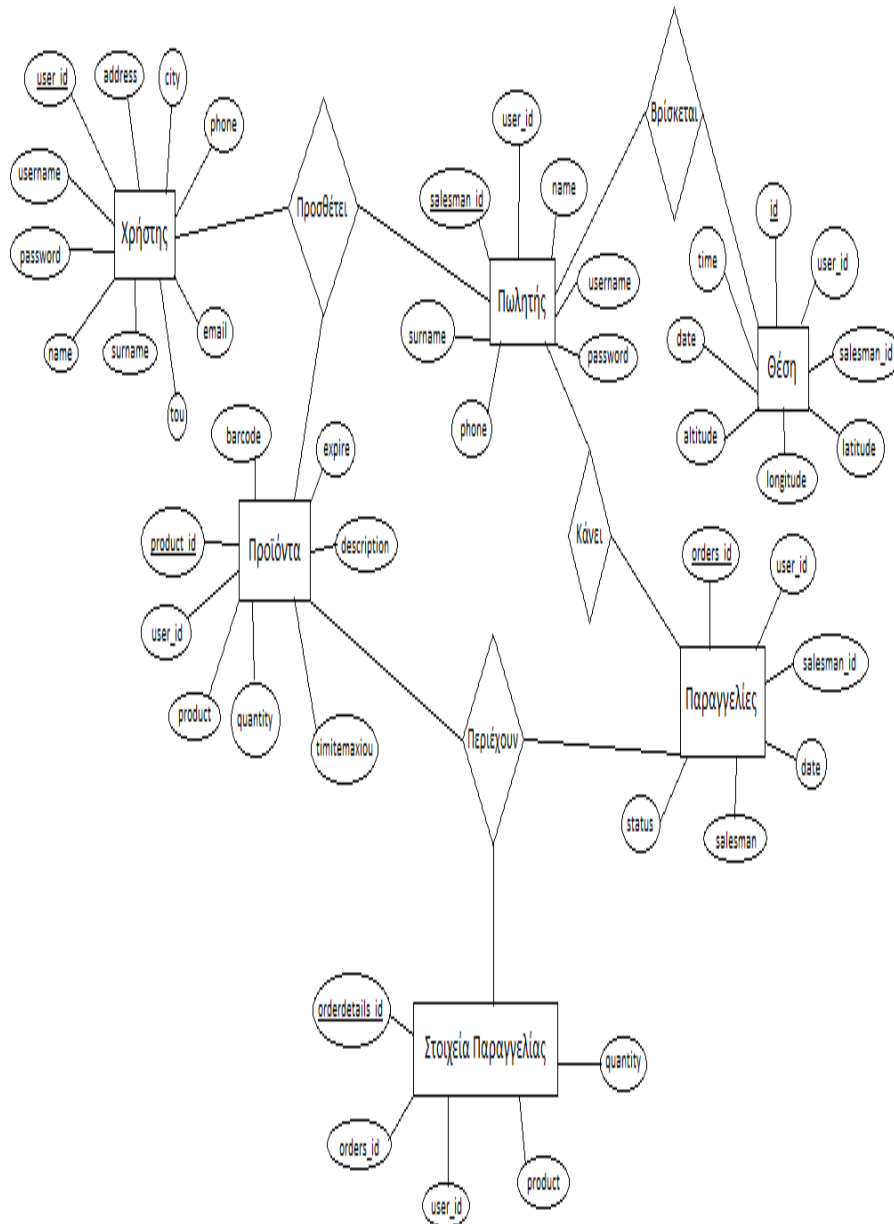
Για την δημιουργία των πινάκων αρχικά χρησιμοποιήθηκε χαρτί όπως προαναφέρθηκε. Έπειτα λόγω ευκολίας στην χρήση και λόγω των δυνατοτήτων του χρησιμοποιήθηκε το `xampp` για ένα πρώτο έλεγχο στην λειτουργικότητα της βάσης. Τέλος λόγω της χρήσης διαφορετικού διακομιστή από αυτόν που χρησιμοποιεί το `xampp` χρησιμοποιήθηκε μια δωρεάν κονσόλα η FreeBSD.

2.7 Διάγραμμα οντοτήτων συσχετίσεων



Σχήμα 2.1: Διάγραμμα οντοτήτων συσχετίσεων

2.7 Προσχέδιο βάσης δεδομένων



Σχήμα 2.2: Προσχέδιο βάσης δεδομένων

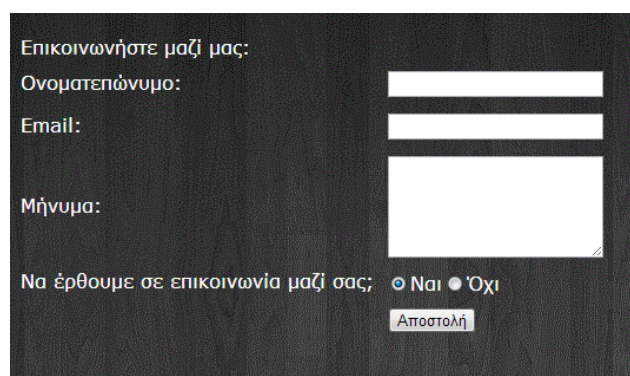
Κεφάλαιο 3

Σχεδιασμός του ιστοχώρου

Η ιστοσελίδα δημιουργήθηκε με σκοπό να δώσει στον ιδιοκτήτη της εταιρίας μία εφαρμογή εύκολη σε χρήση αλλά και συγχρόνως λειτουργική. Σε αυτό το κεφάλαιο θα γίνει παρουσίαση της λειτουργίας της διαδικτυακής εφαρμογής και θα αναλυθούν μερικά από τα πιο σημαντικά στάδια της υλοποίησής της.

3.1 Εγγραφή χρήστη

Για να χρησιμοποιήσει την εφαρμογή ο χρήστης θα πρέπει να δώσει τα στοιχεία τα οποία επιθυμεί να χρησιμοποιεί στον διαχειριστή. Για να κάνει εγγραφή ένας χρήστης πρέπει να έρθει σε επαφή με έναν διαχειριστή. Στην αρχική σελίδα της ιστοσελίδας υπάρχουν ορισμένες πληροφορίες για την εφαρμογή καθώς και τον τρόπο λειτουργίας της. Επίσης υπάρχει και η καρτέλα επικοινωνίας ώστε να μπορέσει ένας χρήστης να κάνει εγγραφή.



Επικοινωνήστε μαζί μας:

Όνοματεπώνυμο:

Email:

Μήνυμα:

Να έρθουμε σε επικοινωνία μαζί σας; Ναι Όχι

Εικόνα 3.1 Επικοινωνία

Ο χρήστης εισάγει το ονοματεπώνυμο του και το email του μαζί με ένα μήνυμα που να ενημερώνει τους διαχειριστές για τον λόγο επικοινωνίας. Αφού γίνει η επικοινωνία και ο διαχειριστής γνωρίζει τα στοιχεία του χρήστη μπορεί να τον κάνει εγγραφή ώστε να μπορεί να χρησιμοποιήσει το σύστημα. Τα στοιχεία εισαγωγής στο σύστημα στέλνονται στο προσωπικό email του μελλοντικού χρήστη ο οποίος είναι έτοιμος να χρησιμοποιήσει την ιστοσελίδα. Του αποστέλλεται επίσης και ένα μήνυμα με την διαδρομή που πρέπει να ακολουθήσει από την έξυπνη συσκευή του για να κατεβάσει και την αντίστοιχη εφαρμογή για android.



The image shows a registration form with the following fields and labels:

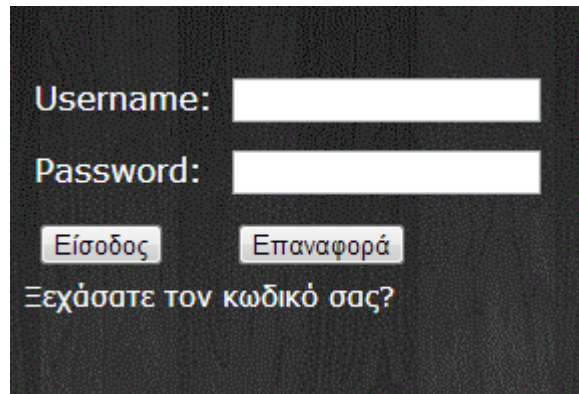
- Username: [input field]
- Password: [input field]
- Name: [input field]
- Surname: [input field]
- Email: [input field]
- Address: [input field]
- City: [input field]
- Phone: [input field]
- Type Of User: [input field]

At the bottom right of the form is a button labeled "Εγγραφή!".

Εικόνα 3.2 Εγγραφή χρήστη

3.2 Σύνδεση-αποσύνδεση χρήστη

Πηγαίνοντας στην καρτέλα login ο χρήστης μπορεί να συνδεθεί για να χρησιμοποιήσει το σύστημα. Ο χρήστης πρέπει να βάλει το όνομα χρήστη και το κωδικό τον οποίο είχε δηλώσει κατά την επικοινωνία του με τον διαχειριστή. Γίνεται έλεγχος να μην είναι κενά τα πεδία από τον εξυπηρετητή και αν η εισαγωγή είναι σωστή ανακατευθύνεται στην προσωπική αρχική του σελίδα. Συγχρόνως δημιουργείται μία συνεδρία και δημιουργούνται cookies. Σε περίπτωση λάθους εμφανίζει ανάλογο μήνυμα που ενημερώνει αν το λάθος έγινε στον κωδικό ή αν δεν υπάρχει το όνομα χρήστη.



Εικόνα 3.3 Login

Αποσύνδεση κάνει από την καρτέλα logout που υπάρχει στην προσωπική του σελίδα. Μόλις πατηθεί το logout τα cookies ακυρώνονται και ο χρήστης γυρίζει στην αρχική σελίδα του συστήματος.

3.2.1 Σημαντικά σημεία κώδικα

Ο έλεγχος αν είναι σωστά τα στοιχεία εισαγωγής είναι από τα πιο σημαντικά σημεία. Αφού σταλούν τα στοιχεία στον διακομιστή κάνει έλεγχο στην βάση αν υπάρχουν αυτά τα στοιχεία. Αν υπάρχουν γίνεται ακόμα ένας έλεγχος αυτός του είδους χρήστη. Ο διαχωρισμός που γίνεται εδώ είναι ανάλογα με έναν αριθμό. Αν αυτός ο αριθμός είναι ο 1 τότε σημαίνει ότι ο χρήστης που προσπάθησε να μπει είναι ο διαχειριστής και τον ανακατευθύνει στην κεντρική σελίδα του διαχειριστή, αλλιώς ο χρήστης είναι ο πελάτης που χρησιμοποιεί την ιστοσελίδα. Σε περίπτωση λάθους εμφανίζεται μήνυμα λάθους. Η χρήση του πεδίου του έχει γίνει για 2 λόγους. Για να ξεχωρίζει το είδος χρήστη και να ανακατευθύνει στην σωστή κεντρική σελίδα αλλά και για λόγους ασφαλείας καθώς δεν μπορεί να εισέλθει χρήστης στο σύστημα που να μην του έχει δοθεί ο αριθμός 2 από τον διαχειριστή του συστήματος

```
10 $username = stripslashes($username);
11 $password = stripslashes($password);
12 $username = mysql_real_escape_string($username);
13 $password = sha1(mysql_real_escape_string($password));
14
15
16 $result=mysql_query("SELECT * FROM users WHERE username='$username' AND password='$password'");
17 $count=mysql_num_rows($result);
18
19 while($row = mysql_fetch_array( $result )) {
20     $tou=$row['tou'];
21 }
22 if($count==1){
23     if($tou=="1"){
24         $_SESSION['username']=$username;
25         header("location:admin.php");
26     }
27     else if($tou=="2") {
28         $_SESSION['username']=$username;
29         header("location:userindex.php");
30     }
31 }
32 }
33 else{
34     ?>
35     <script type="text/javascript">
36     alert("Wrong password");
37     history.back();
38     </script>
```

3.3 Αλλαγή κωδικού

Εάν ο χρήστης έχει ξεχάσει τον κωδικό του μπορεί να πατήσει την επιλογή “Ξεχάσατε τον κωδικό σας” και μετά την εισαγωγή του προσωπικού του email και του ονόματος χρήστη να παραλάβει έναν καινούριο κωδικό.

3.3.1 Σημαντικά σημεία κώδικα

Κατά την αλλαγή κωδικού μετά από απαίτηση του χρήστη γίνονται αρκετοί έλεγχοι ώστε να σιγουρευτούμε ότι το μήνυμα που θα σταλθεί είναι ασφαλές και ότι θα πάει στο σωστό email του χρήστη. Ο πρώτος έλεγχος που γίνεται είναι για το αν το email που έχει δοθεί είναι σωστό, δηλαδή σε σωστή μορφή. Περνάει από ελέγχους όπως να υπάρχει το σύμβολο @ και οι χαρακτήρες να μην υπερβαίνουν ένα συγκεκριμένο όριο. Δεύτερος και επίσης σημαντικός έλεγχος είναι η ύπαρξη του δοθέντος email αλλά και του ονόματος χρήστη στην βάση δεδομένων για λόγους ασφάλειας κυρίως.

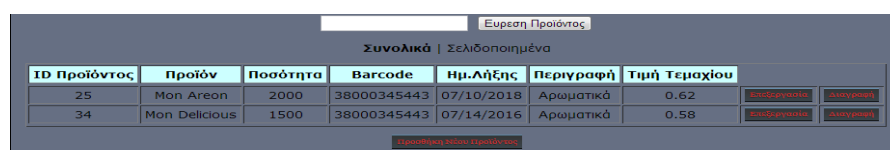
```
68 //This functions checks and makes sure the email address that is being added to database is valid in format.
69 function check_email_address($email) {
70 // First, we check that there's one @ symbol, and that the lengths are right
71 if (!ereg("^[^@]{1,64}@[^@]{1,255}$", $email)) {
72 // Email invalid because wrong number of characters in one section, or wrong number of @ symbols.
73 return false;
74 }
```

Τέλος σημαντική είναι και η δημιουργία του καινούριου κωδικού που θα σταλθεί στον χρήστη. Δημιουργείται ένας τυχαίος αριθμός χρησιμοποιώντας την συνάρτηση sha1. Από αυτόν τον αριθμό παίρνουμε τα 8 πρώτα ψηφία που θα χρησιμοποιηθούν ως καινούριος κωδικός του χρήστη. Κωδικοποιούμε αυτά τα ψηφία πάλι με την συνάρτηση sha1 για λόγους ασφαλείας για να μπορέσουμε να τα στείλουμε μέσω διαδικτύου στο email του κατόχου.

```
127 //Generate a RANDOM sha1 for a password
128 $random_password=sha1(uniqid(rand()));
129
130 //Take the first 8 digits and use them as the password we intend to email the user
131 $emailpassword=substr($random_password, 0, 8);
132
133 //Encrypt $emailpassword in SHA1 format for the database
134 $newpassword = sha1($emailpassword);
```

3.4 Προϊόντα

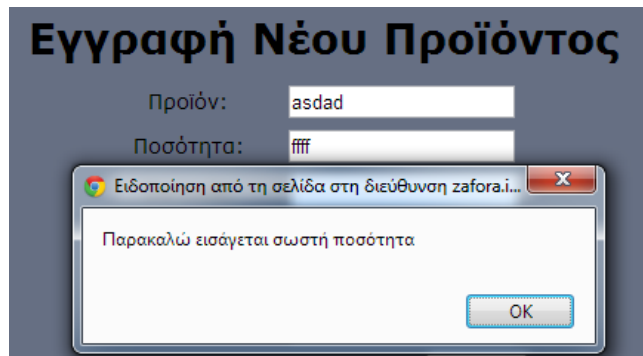
Στην καρτέλα αποθήκη ο εγγεγραμμένος χρήστης έχει την δυνατότητα του ελέγχου των προϊόντων του. Μπορεί να εισάγει νέο προϊόν να επεξεργαστεί ή να διαγράψει ένα υπάρχον. Η αρχική σελίδα της καρτέλας αποθήκη εμφανίζει μία λίστα με τα ήδη υπάρχοντα προϊόντα. Τα στοιχεία που εμφανίζει είναι το ID προϊόντος, το όνομα του, η διαθέσιμη ποσότητα στην αποθήκη, το barcode, η ημερομηνία λήξης, μια σύντομη περιγραφή και η τιμή τεμαχίου.



ID Προϊόντος	Προϊόν	Ποσότητα	Barcode	Ημ.Λήξης	Περιγραφή	Τιμή Τεμαχίου	Επιβεβαιώστε	Απογραφή
25	Mon Areon	2000	38000345443	07/10/2018	Αρωματικά	0.62	Επιβεβαιώστε	Απογραφή
34	Mon Delicious	1500	38000345443	07/14/2016	Αρωματικά	0.58	Επιβεβαιώστε	Απογραφή

Εικόνα 3.4 Προϊόντα

Στην προσθήκη νέου προϊόντος και στην επεξεργασία γίνεται έλεγχος τόσο στο περιβάλλον χρήστη όσο και στον εξυπηρετητή για να μην υπάρχουν κενά πεδία αλλά και για την σωστή εισαγωγή στοιχείων.



Εικόνα 3.5 Έλεγχος σωστής εισαγωγής στοιχείων

Το πεδίο διαγραφή σβήνει όλα τα στοιχεία ενός προϊόντος αφού πρώτα γίνει επιβεβαίωση από τον χρήστη για την διαγραφή. Η καρτέλα σελιδοποιημένα δίνει την δυνατότητα στον χρήστη να αλλάξει τον τρόπο προβολής της σελίδας έτσι ώστε όταν υπάρχουν πολλές καταχωρήσεις να μπορεί να τις χωρίσει σε σελίδες.

Υπάρχει επίσης δυνατότητα ταξινόμησης του πίνακα των προϊόντων κάνοντας κλικ σε ένα από τα στοιχεία. Κάνοντας κλικ μία φορά η ταξινόμηση γίνεται κατά αύξουσα σειρά ενώ με το δεύτερο κλικ γίνεται κατά λήγουσα.

Τέλος ο χρήστης μπορεί να κάνει εύρεση ενός προϊόντος σύμφωνα με το όνομα του. Η εύρεση γίνεται μέσα από μία συνάρτηση η οποία κοιτάει τα γράμματα ένα- ένα και φέρνει σαν αποτέλεσμα τις λέξεις που μοιάζουν ποιο πολύ με αυτό που εισήγαγε ο χρήστης. Κεφαλαία η πεζά γράμματα δεν δημιουργούν κανένα πρόβλημα στην εύρεση.

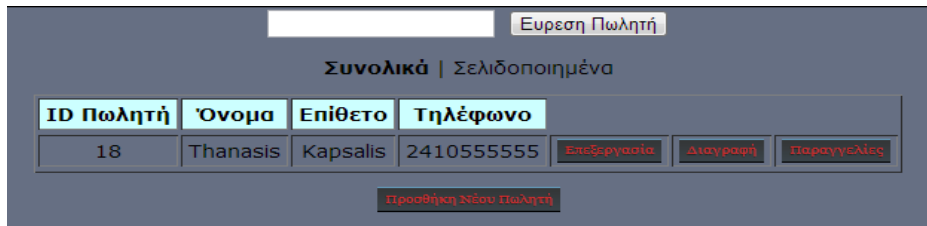
3.4.1 Σημαντικά σημεία κώδικα

Κύριο σημείο στην δημιουργία της εισαγωγής νέου προϊόντος είναι η παροχή βοήθειας στον χρήστη για σημεία που μπορεί να παραλείψει ή να κάνει λάθος κατά την εισαγωγή τους. Χρησιμοποιήθηκε javascript για τον έλεγχο από την πλευρά του πελάτη πριν αποσταλούν τα στοιχεία και χωρίς να χρειαστεί να κάνει ανανέωση η σελίδα. Έτσι όλα τα πεδία εισαγωγής ελέγχονται πριν την αποστολή τους για να μην είναι κενά και να έχουν την απαιτούμενη μορφή. Για παράδειγμα η ποσότητα απαιτείται να είναι αριθμός και αν εισαχθεί έστω και ένα γράμμα δημιουργείται μήνυμα που να ενημερώνει τον χρήστη.

```
43 function validateForm()
44 {
45     var a=document.forms["myform"]["product"].value;
46     var b=document.forms["myform"]["quantity"].value;
47     var c=document.forms["myform"]["barcode"].value;
48     var d=document.forms["myform"]["expire"].value;
49     var e=document.forms["myform"]["description"].value;
50     var f=document.forms["myform"]["timitemaxiou"].value;
51     if (a==null || a=="")
52     {
53         alert("Παρακαλώ εισάγεται όνομα προϊόντος");
54         return false;
55     }
56     if (b==null || b=="")
57     {
58         alert("Παρακαλώ εισάγεται ποσότητα");
59         return false;
60     }
61     if (isNaN(b))
62     {
63         alert("Παρακαλώ εισάγεται σωστή ποσότητα");
64         return false;
65     }
66     if (c==null || c=="")
67     {
68         alert("Παρακαλώ εισάγεται barcode");
69         return false;
70     }
```

3.5 Πωλητές

Στην καρτέλα πωλητές ο χρήστης μπορεί να δει τα στοιχεία των πωλητών του. Τα στοιχεία που εμφανίζονται είναι το ID του πωλητή, το όνομα, το επίθετο και το τηλέφωνο. Οι δυνατότητες που παρέχονται εδώ είναι παρόμοιες με την καρτέλα των προϊόντων. Δηλαδή ο χρήστης μπορεί να εισάγει έναν καινούριο πωλητή να επεξεργαστεί, να κάνει εύρεση και να διαγράψει.

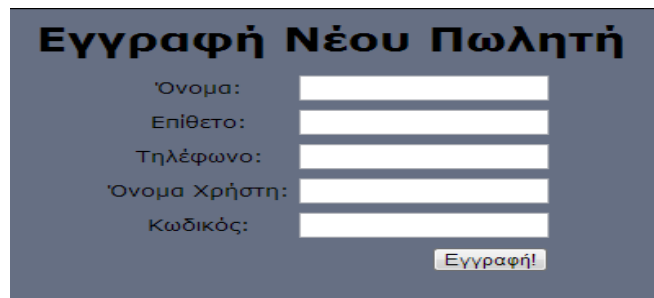


ID Πωλητή	Όνομα	Επίθετο	Τηλέφωνο	Επεξεργασία	Διαγραφή	Παραγγελίες
18	Thanasis	Kapsalis	2410555555			

Προσθήκη Νέου Πωλητή

Εικόνα 3.6 Πωλητές

Στην εισαγωγή και στην επεξεργασία γίνεται έλεγχος να είναι όλα τα πεδία συμπληρωμένα όπως επίσης και να είναι σωστά(π.χ. το πεδίο τηλέφωνο να περιέχει αριθμούς). Επιπλέον υπάρχουν δύο πεδία τα οποία δεν εμφανίζονται στην αρχική σελίδα, τα οποία είναι το όνομα χρήστη και ο κωδικός. Αυτά τα εισάγει ο χρήστης και τα δίνει στον πωλητή για να μπορεί να χρησιμοποιήσει την εφαρμογή στο smart phone στην οποία θα αναφερθούμε αργότερα.



Εγγραφή Νέου Πωλητή

Όνομα:

Επίθετο:

Τηλέφωνο:

Όνομα Χρήστη:

Κωδικός:

Εικόνα 3.6 Εγγραφή πωλητή

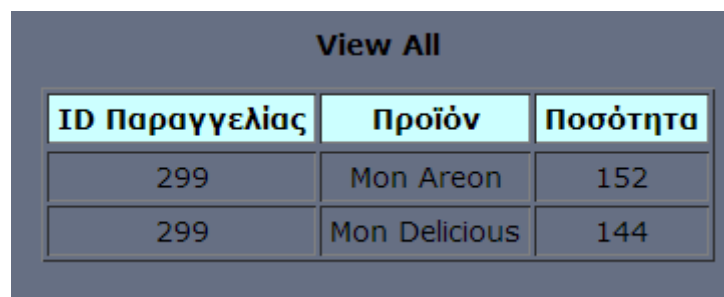
Μία επιπλέον δυνατότητα που έχει ο χρήστης είναι η παρακολούθηση των παραγγελιών που έχει κάνει μέχρι τώρα ο κάθε πωλητής ξεχωριστά πατώντας το κουμπί παραγγελίες. Μπορεί να δει την συνολική λίστα με τις παραγγελίες που έκανε μέχρι τώρα ο πωλητής με στοιχεία όπως η ημερομηνία και η πληρωμή της.



ID Παραγγελίας	Ημερομηνία	Κατάσταση	
276	07/09/2013	Εκκρεμεί	+
299	05/12/2013	Εκκρεμεί	+
300	15/04/2013	Πληρώθηκε	+
301	05/12/2013	Εκκρεμεί	+
302	28/08/2013	Εκκρεμεί	+

Εικόνα 3.7 Παραγγελίες πωλητών

Πατώντας το κουμπί με το σύμβολο συν(+) ο χρήστης μπορεί να δει τα προϊόντα που πούλησε ο πωλητής σε κάθε μια παραγγελία ξεχωριστά. Το ID παραγγελίας υπάρχει για να ξεχωρίζει σε ποια παραγγελία ανήκουν τα προϊόντα.



ID Παραγγελίας	Προϊόν	Ποσότητα
299	Mon Areon	152
299	Mon Delicious	144

Εικόνα 3.8 Προϊόντα παραγγελιών

3.5.1 Σημαντικά σημεία κώδικα

Η εμφάνιση των παραγγελιών του κάθε πωλητή ξεχωριστά είναι ένα ενδιαφέρον σημείο καθώς πρέπει να ξεχωρίσουμε τους πωλητές του κάθε χρήστη αλλά και τις παραγγελίες του κάθε πωλητή. Χρησιμοποιώντας το Session βρίσκουμε κάθε φορά ποιος χρήστης είναι συνδεδεμένος και επιλέγουμε από τον πίνακα salesman τους πωλητές που έχουν σαν username αυτόν τον χρήστη. Επόμενο βήμα είναι να επιλέξουμε τις παραγγελίες του συγκεκριμένου πωλητή. Μπορούμε να κάνουμε αναζήτηση στον πίνακα orders και να επιλέξουμε τις παραγγελίες που έχουν user_id και salesman_id αυτά του χρήστη και του πωλητή. Επειδή δεν γνωρίζουμε τον ακριβή αριθμό των παραγγελιών δεν μπορούμε να

δημιουργήσουμε έναν στατικό πίνακα για αυτό και η δημιουργία του πίνακα παρουσίασης των παραγγελιών γίνεται μέσα σε ένα while που διαρκεί όσο ελέγχει τον πίνακα και βρίσκει αποτελέσματα με τα στοιχεία που αναφέρθηκαν.

```
35 $salesman_id=$_GET['id'];
36 $username=$_SESSION['username'];
37 $res=mysql_query("SELECT user_id FROM users WHERE username='$username'");
38 $row1=mysql_fetch_array($res);
39 $user_id=$row1['user_id'];
40
41
42 $result=mysql_query("SELECT * FROM orders WHERE user_id='$user_id' && salesman_id='$salesman_id' ");
43
44
45 echo "<p><b>View All</b> </p>";
46 echo "<table cellpadding='4' border='1'>";
47 echo "<thead>";
48 echo "<tr> <th>ID Παραγγελίας</th> <th>Ημερομηνία</th> <th>Κατάσταση</th> <th></th></tr>";
49 echo "</thead>";
50 while($row = mysql_fetch_array( $result )) {
51
52     echo "<tr>";
53     echo "<td>" . $row['orders_id'] . "</td>";
54     echo "<td>" . $row['date'] . "</td>";
55     echo "<td>" . $row['status'] . "</td>";
56     echo "<td><a class='button' href='users/orderdetailsalesman.php?id=" . $row['orders_id'] . "'></td>";
57
58
59     echo "</tr>";
60 }
61
62 // close table
63 echo "</table>";
```

Εξίσου σημαντικό είναι και το σημείο που κάνουμε εύρεση στην βάση ανάλογα με τα στοιχεία που εισήγαγε ο χρήστης. Πρώτο βήμα είναι η δημιουργία φόρμας για την εισαγωγή των στοιχείων. Αυτή η φόρμα κάνει Post σε μία άλλη σελίδα η οποία θα εμφανίσει τα αποτελέσματα της αναζήτησης.

```
25 <div id="maincontent">
26 <form action="usersalesmansearch.php" method="post">
27 <input type="text" name="search" value="" />
28 <input type="submit" value="Εύρεση Πωλητή" />
29 </form>
```

Το αποτέλεσμα του Post φιλτράρεται για να φύγουν τα κενά που τυχόν εισήγαγε ο χρήστης στην αρχή ή στο τέλος της λέξης αναζήτησης με την συνάρτηση trim. Για τον έλεγχο της λέξης αναζήτησης στην Php τα μονά ή διπλά εισαγωγικά απαιτείται να αλλαχθούν από '/' για αυτό και χρησιμοποιούμε την συνάρτηση get_magic_quotes_gpc. Αν δεν

μπορεί να χρησιμοποιηθεί αυτή λόγω χαμηλής έκδοσης Php στον διακομιστή χρησιμοποιείται η συνάρτηση addslashes που κάνει ακριβώς το ίδιο. Τέλος γίνεται ένας έλεγχος στην βάση δεδομένων στον πίνακα salesman για να βρεθεί η λέξη που μόλις πέρασε από τις ανωτέρω αλλαγές.

```
33 $search=$_POST['search'];
34 trim($search);
35 if(!get_magic_quotes_gpc())
36 {
37     $search=addslashes($search);
38 }
39 $res1=mysql_query("SELECT * FROM salesman WHERE surname LIKE '%" . $search . "%' AND user_id='\$user_id'");
```

3.6 Παραγγελίες

Στην καρτέλα παραγγελίες ο χρήστης μπορεί να δει αναλυτικά όλες τις παραγγελίες που έγιναν από τους πωλητές του. Στην κεντρική σελίδα εμφανίζονται στοιχεία όπως το ID της παραγγελίας η ημερομηνία που έγινε ο πωλητής ο οποίος έκανε την παραγγελία και η κατάσταση της. Υπάρχουν και εδώ οι δυνατότητες επεξεργασίας εισαγωγής και διαγραφής παραγγελίας.

ID Παραγγελίας	Ημερομηνία	Πωλητής	Κατάσταση	Επεξεργασία	Διαγραφή
276	07/09/2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
299	05/12/2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
300	15/04/2013	Kapsalis	Πληρώθηκε	Επεξεργασία	Διαγραφή
301	05/12/2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
302	28/08/2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
304	14/12/2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
351	05/12/2013	Kapsalis	Πληρώθηκε	Επεξεργασία	Διαγραφή
352	12/05/2013	Kapsalis	Πληρώθηκε	Επεξεργασία	Διαγραφή
353	05/12/2014	Kapsalis	Πληρώθηκε	Επεξεργασία	Διαγραφή
365	7 /29 /2013	Kapsalis	Εκκρεμεί	Επεξεργασία	Διαγραφή
366	8 /30 /2013	Kapsalis	Έναντι	Επεξεργασία	Διαγραφή

Εικόνα 3.9 Παραγγελίες

Έχουν προστεθεί χρώματα ανάλογα με την κατάσταση της παραγγελίας για να μπορεί ο χρήστης να διακρίνει εύκολα αυτές που θέλει. Η εύρεση στην καρτέλα παραγγελιών μπορεί να γίνει με την χρήση τριών πεδίων όπως φαίνεται και στην εικόνα 3.9, τα οποία είναι ο πωλητής, η ημερομηνία και η κατάσταση της παραγγελίας. Στο πεδίο ημερομηνία έχει προστεθεί ημερολόγιο για να μην πληκτρολογεί ο χρήστης την ημερομηνία και υπάρξουν αλλαγές στον τρόπο γραφής και στο πεδίο κατάσταση εμφανίζεται λίστα με 4 επιλογές για να επιλέξει ο χρήστης. Η εύρεση λειτουργεί με τον ίδιο τρόπο όπως και στις καρτέλες αποθήκη και πωλητές.

Στην προσθήκη νέας παραγγελίας ο χρήστης μπορεί να εισάγει μία παραγγελία. Μετά το πάτημα του κουμπιού προσθήκη νέας παραγγελίας ο χρήστης μεταβαίνει στην επόμενη σελίδα στην οποία ξεκινάει να εισάγει τα δεδομένα της παραγγελίας. Εμφανίζονται 3 πεδία της ημερομηνίας του πωλητή και της κατάστασης. Η ημερομηνία εισάγεται με ημερολόγιο, ο πωλητής με μενού που εμφανίζει τους πωλητές που υπάρχουν στην βάση και η κατάσταση πάλι με μενού με τις 4 επιλογές. Αφού πρώτα γίνει έλεγχος ότι όλα τα πεδία είναι γεμάτα ο χρήστης μπορεί να πατήσει το κουμπί συνέχεια και να μεταβεί στην επόμενη σελίδα.

Εγγραφή Νέας Παραγγελίας

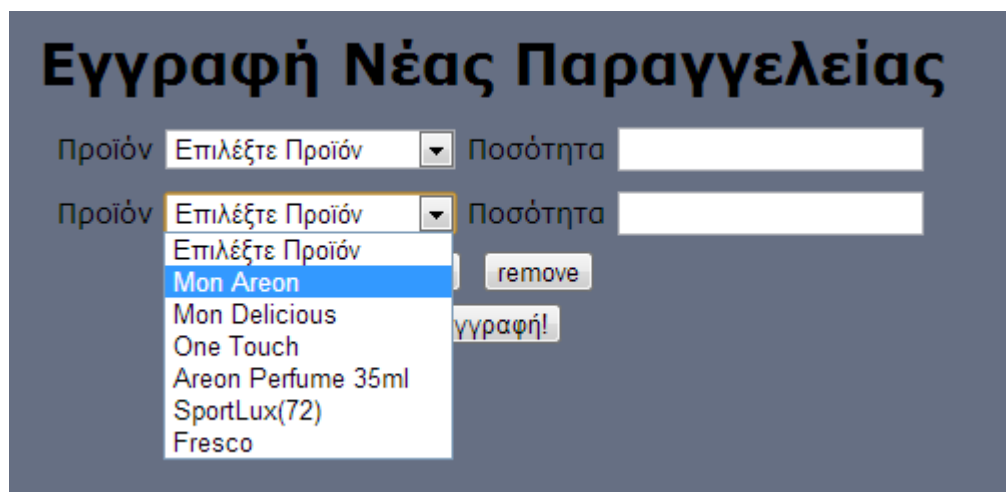
Ημερομηνία:

Πωλητής:

Κατάσταση:

Εικόνα 3.10 Εγγραφή νέας παραγγελίας

Σε αυτή την σελίδα ο χρήστης βάζει τα προϊόντα της συγκεκριμένης παραγγελίας. Επειδή το είδος των προϊόντων που παραγγέλνει ο κάθε πελάτης διαφέρει δεν μπορούσε να υπάρχει ένας σταθερός αριθμός πεδίων για την εισαγωγή προϊόντων. Δημιουργήθηκε για αυτό το λόγο ένα κουμπί το οποίο προσθέτει δυναμικά πεδία ανάλογα με τον αριθμό που χρειάζεται ο χρήστης. Υπάρχει επίσης ένα κουμπί για αφαίρεση πεδίων. Στην νέα παραγγελία υπάρχει το πεδίο προϊόν και το πεδίο ποσότητα. Στο πεδίο προϊόν ο χρήστης εισάγει τα προϊόντα που έχει στην αποθήκη τα οποία εμφανίζονται αυτόματα σε μία λίστα. Στο πεδίο ποσότητα γίνεται έλεγχος για την εισαγωγή αριθμού και μόνο. Πατώντας το κουμπί εγγραφή γίνεται η προσθήκη της νέας παραγγελίας.



Εγγραφή Νέας Παραγγελίας

Προϊόν Ποσότητα

Προϊόν Ποσότητα

Επιλέξτε Προϊόν
Μον Areon
Μον Delicious
One Touch
Areon Perfume 35ml
SportLux(72)
Fresco

remove

γγραφή!

Εικόνα 3.11 Στοιχεία παραγγελίας

3.6.1 Σημαντικά σημεία κώδικα

Για την εισαγωγή στοιχείων παραγγελίας χρειάστηκε η δημιουργία δυναμικής φόρμας, δηλαδή έπρεπε να δοθεί στον χρήστη η δυνατότητα να εισάγει με το πάτημα ενός κουμπιού νέα πεδία για εισαγωγή προϊόντων στην παραγγελία. Για την δημιουργία αυτού του κουμπιού δημιουργήθηκε ένα function με javascript κάθε φορά που πατιέται το συγκεκριμένο κουμπί.

Κάθε φορά που πατιέται το κουμπί δημιουργείται ένας κλώνος του div στο οποίο υπάρχουν τα πεδία εισαγωγής των προϊόντων και της ποσότητας. Δίνεται ένας αριθμός στον κλώνο αυτό και οι ιδιότητες του είναι ακριβώς οι ίδιες με το αρχικό. Έτσι έχει ακριβώς τις ίδιες δυνατότητες με την μόνη διαφορά ότι κατά την αποστολή των στοιχείων κάθε κλώνος έχει έναν αριθμό μεγαλύτερο στο όνομα του από τον προηγούμενο ώστε να μπορεί να ξεχωρίσει και να αποθηκευτεί σε ξεχωριστή γραμμή στον πίνακα. Τέλος με κάθε κλώνο που δημιουργείται αυξάνεται κατά ένα ένας μετρητής έτσι ώστε μόλις φτάσει στο 10 να μην μπορεί να πατηθεί το κουμπί εισαγωγής νέου πεδίου, καθώς ο μέγιστος αριθμός που συμφωνήθηκε είναι το 10.

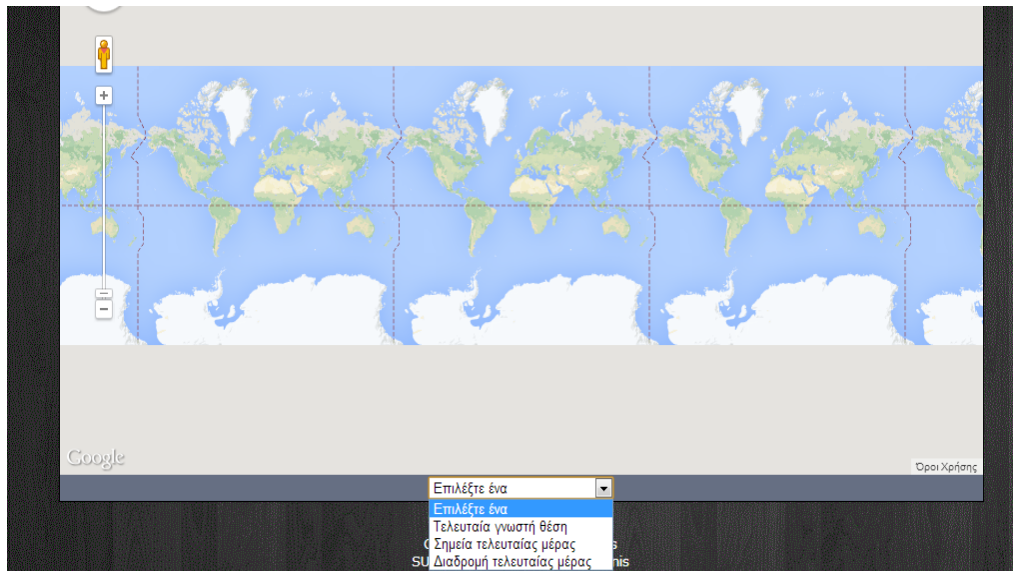
Το κουμπί remove διαγράφει τον τελευταίο κλώνο που δημιουργήθηκε χωρίς να μπορεί να διαγράψει τα αρχικά πεδία.

```
60 // manipulate the name/id values of the input inside the new element
61 newElem.children(':first').attr('id', 'ctable' + newNum).attr('name', 'ctable' + newNum);
62 newElem.children(':first').find("tbody.dynctb").attr('id', 'ctb' + newNum).attr('name', 'ctb' + newNum);
63 newElem.children(':first').find("tr.dynctr").attr('id', 'ctrow' + newNum).attr('name', 'ctrow' + newNum);
64 newElem.children(':first').find("td.dynctd").attr('id', 'item' + newNum).attr('name', 'item' + newNum);
65
66 // insert the new element after the last "duplicatable" input field
67 $('#input' + num).after(newElem);
68
69 // enable the "remove" button
70 $('#btnDel').attr('disabled', '');
71
72 // business rule: you can only add 10 names
73 if (newNum == 10)
74 |   $('#btnAdd').attr('disabled', 'disabled');
75 | });
76
77 $('#btnDel').click(function() {
78 |   var num = $('.clonedInput').length; // how many "duplicatable" input fields we currently have
79 |   $('#input' + num).remove(); // remove the last element
80 |
81 |   // enable the "add" button
82 |   $('#btnAdd').attr('disabled', '');
83 |
84 |   // if only one element remains, disable the "remove" button
85 |   if (num-1 == 1)
86 | |   $('#btnDel').attr('disabled', 'disabled');
87 | });
```

3.7 Εντοπισμός θέσης

Στην καρτέλα εντοπισμός θέσης δίνεται η δυνατότητα στον ιδιοκτήτη να παρακολουθήσει την πορεία των πωλητών κατά την διάρκεια μίας εργάσιμης ημέρας. Αυτό καθίσταται δυνατό με την χρήση της εφαρμογής σε smart phone. Η λειτουργία της εφαρμογής θα αναφερθεί στο κεφάλαιο 4. Οι 3 λειτουργίες που είναι δυνατές εδώ είναι η εύρεση της τελευταίας

γνωστής θέσης, η εύρεση όλων των σημείων της τελευταίας ημέρας και ο υπολογισμός της διαδρομής που ακολούθησε ο πωλητής.

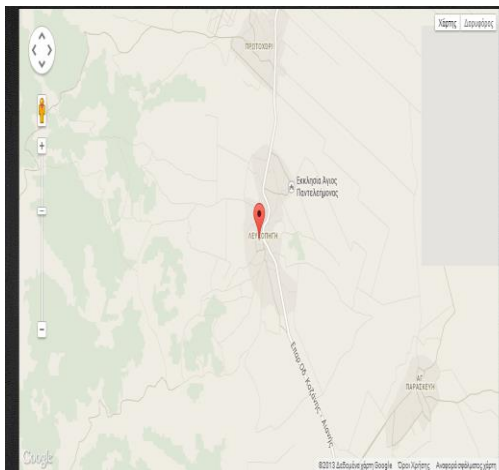


Εικόνα 3.12 Εντοπισμός θέσης

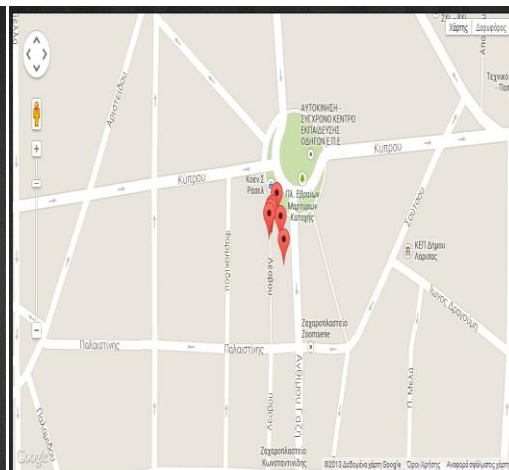
Η εφαρμογή στέλνει στον διακομιστή ο οποίος με την σειρά του αποθηκεύει τα στοιχεία της θέσης στον πίνακα gps της βάσης δεδομένων. Με κατάλληλη επιλογή από αυτά τα στοιχεία γίνεται δυνατή η απεικόνιση των θέσεων στον χάρτη.

Για να εμφανισθεί η τελευταία γνωστή θέση στον χάρτη επιλέγετε από τον πίνακα gps η εισαγωγή με το μεγαλύτερο αναγνωριστικό. Για την επιλογή όλων των σημείων της τελευταίας ημέρας επιλέγονται από τον πίνακα οι εισαγωγές που έγιναν την προηγούμενη ημέρα. Στα σημεία που εμφανίζονται στον χάρτη υπάρχουν και πληροφορίες όπως η ημερομηνία και η ώρα που ήταν ο πωλητής σε αυτό το σημείο. Για τον υπολογισμό της διαδρομής επιλέγονται τα σημεία της τελευταίας ημέρας εμφανίζονται στον χάρτη και ενώνονται με χρώμα για να εμφανισθεί η διαδρομή που ακολούθησε ο πωλητής. Όμως επειδή η ανανέωση της θέσης γίνεται ανά 5 λεπτά δεν μπορούμε να πούμε ότι η διαδρομή που ακολούθησε είναι και αυτή που φαίνεται στον χάρτη. Με μείωση του χρόνου που αποστέλλει η εφαρμογή τα στοιχεία της θέσης του πωλητή

μπορεί να εμφανισθεί η διαδρομή με μεγαλύτερη ακρίβεια. Επίσης ένα ακόμα πρόβλημα που εμφανίζεται στην εμφάνιση της διαδρομής είναι ότι λόγω της χρησιμοποίησης του API της Google για την απεικόνιση του χάρτη και των σημείων δεν είναι δυνατόν να υπολογισθεί η διαδρομή για πάνω από 7 σημεία. Για αυτόν τον λόγο δημιουργήθηκε μία συνάρτηση η οποία διαλέγει το πρώτο και το τελευταίο σημείο της ημέρας και τυχαία σημεία ανάμεσα τους και υπολογίζει μία τυχαία διαδρομή. Η διαδρομή που υπολογίζει είναι για πεζούς δηλαδή δεν λαμβάνει υπόψη μονόδρομους ώστε να μπορεί να δώσει μία πιο ρεαλιστική διαδρομή.



Εικόνα 3.13 Τελευταία γνωστή θέση



Εικόνα 3.14 Σημεία τελευταίας μέρας

3.7.1 Σημαντικά σημεία κώδικα

Για τη δημιουργία των χαρτών χρησιμοποιήθηκε το Google Api. Δημιουργήθηκε ένας λογαριασμός στα Google Services και από εκεί πήραμε ένα συγκεκριμένο κλειδί που μπορεί να χρησιμοποιηθεί μόνο στην συγκεκριμένη ιστοσελίδα.

```
9 <script src="http://maps.googleapis.com/maps/api/js?key=AIzaSyAv9KQONqQvMdc8yNiXoctZkucPUr0b0co&sensor=false">
```

Για να απεικονιστεί στον χάρτη η τελευταία γνωστή θέση χρησιμοποιείται function της Google που “φορτώνει” τον χάρτη στην ιστοσελίδα και χρησιμοποιώντας το marker μπορεί να μαρκαριστεί το σημείο στον χάρτη. Το γεωγραφικό πλάτος και μήκος έχει βρεθεί κάνοντας εύρεση στον πίνακα gps και μεταφέρονται σε μεταβλητές javascript καθώς δεν μπορεί να “διαβαστεί” απευθείας από την Php.

Για την εμφάνιση πολλών σημείων απαιτείται η αποθήκευση αυτών των σημείων σε έναν πίνακα ώστε να μπορούν να διαβαστούν ένα-ένα. Πρώτα γίνεται η εύρεση των σημείων από την βάση σύμφωνα με την ημερομηνία.(Επιλέγεται η προηγούμενη ημέρα). Αποθηκεύονται το μήκος, το πλάτος και η ημερομηνία σε πίνακα της Php με αντίστοιχα ονόματα.

```
4 $res=mysql_query("SELECT * FROM gps WHERE date >= ( CURDATE() - INTERVAL 1 DAY )");
5
6 $lat=array();
7 $lon=array();
8 $date=array();
9 while($row = mysql_fetch_array($res)) {
10     $lat[]=$row['latitude'];
11     $lon[]=$row['longtitude'];
12     $date[]=$row['date'];
13 }
```

Το επόμενο βήμα είναι να μεταφέρουμε αυτά τα σημεία σε πίνακα που μπορεί να “διαβαστεί” από javascript. Με ένα loop τα μεταφέρουμε σε μεταβλητή που έχει την μορφή πίνακα ώστε να μπορέσουμε να δημιουργήσουμε τα υποψήφια σημεία για απεικόνιση στον χάρτη. Υπάρχει ξεχωριστός πίνακας για κάθε στοιχείο που χρειαζόμαστε δηλαδή συνολικά υπάρχουν 3.

```
50 var latitude = new Array();
51 <?php
52 for ($i=0;$i<count($lat); $i++)
53 {
54     echo "latitude[$i]='".$lat[$i]."';\n";
55 }
56 ?>
```

Μετά χρησιμοποιώντας ένα for loop το οποίο έχει μέγεθος όσο και το μέγεθος των πινάκων αρχίζουμε και προσθέτουμε σημεία στον χάρτη και ως πληροφορίες που θα εμφανίζονται πάνω από κάθε σημείο επιλέγουμε την ημερομηνία και την ώρα που είναι και αυτές αποθηκευμένες σε πίνακα.

```
82 for (i = 0; i < latitude.length; i++) {  
83     marker = new google.maps.Marker({  
84         position: new google.maps.LatLng(latitude[i], longitude[i]),  
85         map: map  
86     });  
87  
88     google.maps.event.addListener(marker, 'click', (function(marker, i) {  
89         return function() {  
90             infowindow.setContent(dat[i]);  
91             infowindow.open(map, marker);  
92         }  
93     })(marker, i));  
94 }
```

3.8 Διαχειριστής

Ο κύριος σκοπός του διαχειριστή είναι να ενημερώνει τους χρήστες για τυχόν αλλαγές και να εισάγει νέους χρήστες ή να διαγράψει παλαιούς. Η νοοτροπία της ιστοσελίδας ελέγχου του διαχειριστή βασίστηκε σε αυτήν του χρήστη με την διαφορά ότι ο διαχειριστής έχει την δυνατότητα να ελέγξει τους χρήστες. Μπορεί να δει τα στοιχεία τους και να τα επεξεργαστεί. Δεν κρίθηκε απαραίτητη η δυνατότητα του διαχειριστή να βλέπει τις παραγγελίες και τα προϊόντα των χρηστών.

3.9 Στοιχεία κώδικα ιστοχώρου

Για την δημιουργία του ιστοχώρου δημιουργήθηκαν 52 αρχεία Php με συνολικό μέγεθος 140KB και γραμμές κώδικα 4673, 2 αρχεία js(javascript) με μέγεθος 28550B και γραμμές κώδικα 843, 2 αρχεία css με μέγεθος 7660B και γραμμές κώδικα 418. Τέλος υπάρχουν 11 αρχεία εικόνας .png μεγέθους 1316KB και 8 αρχεία εικόνας .gif μεγέθους 579KB. Επίσης για την επικοινωνία της εφαρμογής με την βάση δεδομένων δημιουργήθηκαν 12 αρχεία Php Με συνολικό μέγεθος 12947B και γραμμές κώδικα 411.

3.10 Ασφάλεια

Η ασφάλεια είναι πολύ σημαντική για την λειτουργία ενός ιστοχώρου. Πρέπει να προφυλάσσεται από επιθέσεις που έχουν σκοπό να κλέψουν στοιχεία χρηστών, δεδομένα ή και να προσπαθήσουν να μπουν στο σύστημα σαν χρήστες χωρίς να είναι εγγεγραμμένοι.

Πρώτο βήμα για την ασφάλεια είναι εγγραφή των χρηστών αποκλειστικά και μόνο από τον διαχειριστή για να υπάρχει έλεγχος τόσο του αριθμού των χρηστών αλλά και να μπορεί ο χρήστης να δώσει το χαρακτηριστικό type of user που διακρίνει αν ο χρήστης είναι εγγεγραμμένος και τι δικαιώματα έχει στο σύστημα. Έτσι αν προσπαθήσει να μπει κάποιος που δεν χρησιμοποιεί σωστό type of user ο έλεγχος που γίνεται στον php κώδικα τον εμποδίζει να μπει στο σύστημα.

Δεύτερο είναι η κωδικοποίηση όλων των κωδικών που χρησιμοποιούνται από όλους τους χρήστες πριν την αποστολή τους στον διακομιστή έτσι ώστε αν μπορέσει κάποιος να 'ψαρέψει' έναν κωδικό με την μέθοδο του phishing να μην μπορεί να αποκωδικοποιήσει αυτόν τον κωδικό. Η συνάρτηση που χρησιμοποιείται για κωδικοποίηση είναι η sha1 η οποία είναι One way δηλαδή αν γνωρίζει κάποιος τον αρχικό κωδικό να μπορεί να βρει τον sha1 αντίστοιχό του. Αν δεν τον γνωρίζει όμως είναι σχεδόν ακατόρθωτο να τον αποκωδικοποιήσει.

Τέλος όλα τα πεδία που χρησιμοποιούνται για εισαγωγή στοιχείων ελέγχονται κατά την εισαγωγή τους αλλά και μετά την αποστολή τους ώστε τα στοιχεία που στέλνονται να είναι αυτά που είναι αποδεκτά από κάθε φόρμα εισαγωγής.

Κεφάλαιο 4

Android εφαρμογή

4.1 Εισαγωγικά

Η εφαρμογή σε smart phone με λειτουργικό android έγινε με σκοπό την γρήγορη καταγραφή των παραγγελιών και την άμεση αποστολή τους στην αποθήκη για έλεγχο. Η εφαρμογή απαιτεί να υπάρχει σύνδεση είτε μέσω ασύρματου δικτύου είτε μέσω δορυφορικού(3G-4G). Έχει προτιμηθεί το Gps(Global Positioning System) γιατί ο χρήστης θα κινείται συνεχώς και θα είναι δύσκολο να βρίσκεται σε εμβέλεια ασυρμάτων δικτύων. Επίσης η ακρίβεια του Gps είναι μεγαλύτερη. Τα δεδομένα που στέλνει δεν είναι πολλά ώστε να υπάρχει υπερβολική χρέωση. Η κατανάλωση της μπαταρίας είναι μεγάλη, αλλά φτάνει για την διάρκεια μίας εργάσιμης ημέρας.

4.1.1 Συσκευή Smartphone

Η εφαρμογή σε αρχικό στάδιο δημιουργήθηκε χρησιμοποιώντας το eclipse αλλά λόγω της αγοράς έξυπνου κινητού με λογισμικό android κατά την διάρκεια του προγραμματισμού, συνεχίστηκε στην συσκευή. Η συσκευή στην οποία αναπτύχθηκε η εφαρμογή είναι το Samsung Galaxy S4. Η εφαρμογή είναι σχεδιασμένη να τρέχει σε εκδόσεις από 2.x μέχρι την τελευταία έκδοση πάνω στην οποία και σχεδιάστηκε την 4.2.

Τα χαρακτηριστικά της συσκευής είναι τα εξής:

- Λειτουργικό σύστημα: Android
- System on a chip(SoC): Qualcomm Snapdragon 600
- Επεξεργαστής: Τετραπύρηνος Krait 300 στα 1,9GHz
- Συνεπεξεργαστής: Adreno 300 GPU

- Μνήμη RAM: 2GB
- Αποθηκευτικός χώρος: 16GB
- Διαστάσεις 136,6 x 69,8 x 7,9 χιλιοστά
- Βάρος: 130 γραμμάρια
- Τύπος οθόνης: AMOLED
- Τεχνολογία οθόνης: Full HD Super AMOLED
- Οθόνη αφής
- Ανάλυση: 1080 X 1920 pixels FHD
- Χρώματα: 16,7 εκατ. Χρώματα
- Wi-Fi: IEEE 802.11a/b/g/n/ac
- Εντοπισμός θέσης: GPS, A-GPS Glonass
- Δίκτυα: GSM (850,900,1800,1900 Mhz)/UMTS (850,900,1900,2100 MHz)/LTE 1800MHz)
- Τύπος μπαταρίας: Lithium-Ion

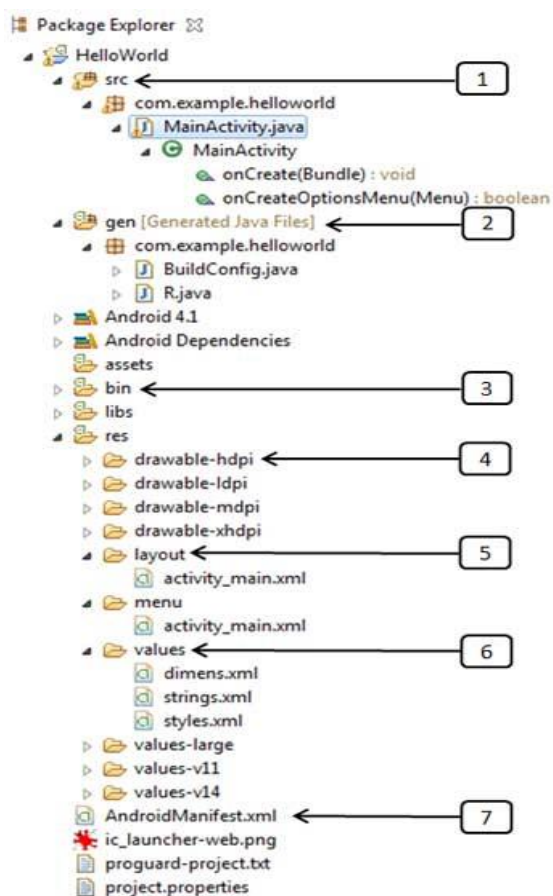


Εικόνα 4.1 Samsung Galaxy S4

4.2 Ανατομία μίας εφαρμογής Android

Για την δημιουργία μίας εφαρμογής Android είναι απαραίτητη η κατανόηση των αρχείων που αποτελούν την εφαρμογή, την κατανόηση του τι είναι δραστηριότητα(activity) τι είναι service(υπηρεσία) και πως αλληλεπιδρούν αυτά μεταξύ τους.

4.2.1 Κατάλογοι και αρχεία μίας εφαρμογής



Τα βασικά αρχεία και κατάλογοι που πρέπει να γνωρίζουμε πριν την δημιουργία μίας εφαρμογής είναι τα εξής:

- (1)SCR- Περιέχει τους πηγαίους κώδικες. Κατά την δημιουργία μίας εφαρμογής δημιουργείται αυτόματα και το αρχείο Main.Activity.java το οποίο περιέχει ένα activity(δραστηριότητα) που τρέχει μόλις πατηθεί το εικονίδιο της εφαρμογής.
- (2)GEN- Περιλαμβάνει τα αρχεία .R ένα αρχείο που έχει μεταγλωττιστεί και αναφέρει όλες τους πόρους που υπάρχουν στην εφαρμογή.
- (3)BIN- Περιέχει όλα τα αρχεία .apk τα οποία χρειάζεται για να “τρέξει” η εφαρμογή.
- (4)RES/Drawable-hdpi- Κατάλογος με σχεδιασμένα αντικείμενα για οθόνες υψηλής πυκνότητας.
- (5)RES/Layout- Αρχεία που καθορίζουν το περιβάλλον του χρήστη.
- (6)RES/Values- Κατάλογος για διάφορα άλλα αρχεία XML που περιέχουν συλλογή από πόρους, όπως παράδειγμα ορισμούς χρωμάτων.
- (7)AndroidManifest.xml- Αρχείο που περιγράφει τα θεμελιώδη χαρακτηριστικά της εφαρμογής.

4.2.2 Activity

Μία δραστηριότητα αντιπροσωπεύει μία οθόνη που μπορεί να αλληλεπιδράσει με τον χρήστη. Για παράδειγμα στην εφαρμογή το κεντρικό μενού αποτελεί ένα περιβάλλον με 2 δραστηριότητες μία για παρουσίαση των παραγγελιών και μία για εισαγωγή νέας παραγγελίας. Οι πιο βασικές μέθοδοι που χρησιμοποιούνται σε μία δραστηριότητα είναι οι ακόλουθες.

onCreate()- Καλείται όταν δημιουργείται η εφαρμογή.

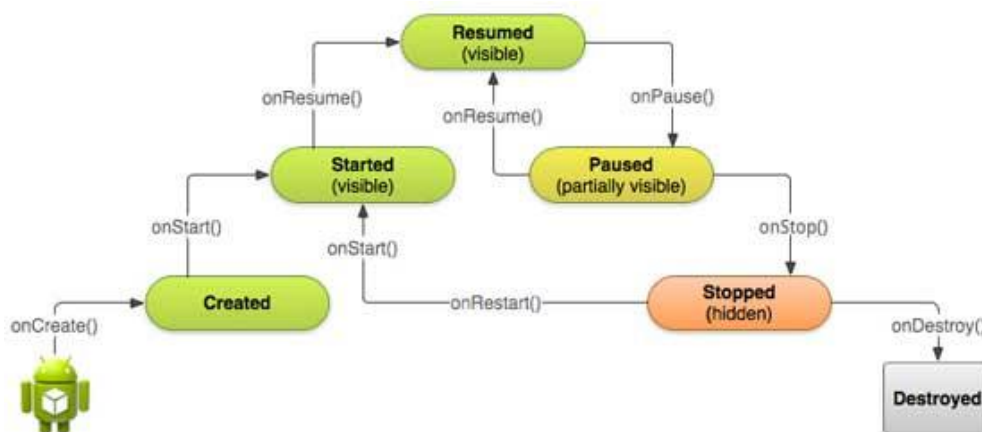
onStart()- Καλείται όταν ξεκινάει η εφαρμογή και μπορεί ο χρήστης να δει την κεντρική οθόνη.

onResume()- Καλείται όταν ξεκινάει ο χρήστης να αλληλεπιδρά με την εφαρμογή.

onStop()- Καλείται όταν ο χρήστης να δει καμία οθόνη της εφαρμογής.

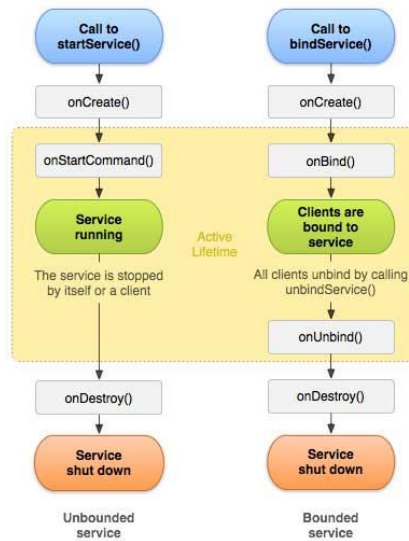
onDestroy()- Καλείται πριν σταματήσει η λειτουργία της εφαρμογής.

onRestart()- Καλείται όταν κάνει επανεκκίνηση η εφαρμογή.



4.2.3 Service

Ένα service αποτελεί ένα στοιχείο που τρέχει στο παρασκήνιο για την εκτέλεση λειτουργιών μακράς διάρκειας χωρίς να χρειάζεται να αλληλεπιδράσει με τον χρήστη. Για παράδειγμα στην εφαρμογή η υπηρεσία που στέλνει το σημείο του χρήστη αποτελεί ένα service.



4.3 Προγραμματισμός-Λειτουργία της εφαρμογής

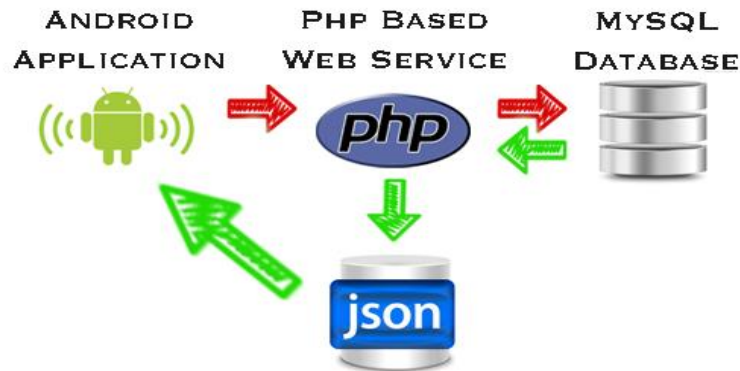
Παρακάτω αναφέρεται αναλυτικά η λειτουργία της εφαρμογής και προστίθενται ενδιάμεσα σημαντικά κομμάτια κώδικα τα οποία αναλύονται.

4.3.1 Είσοδος χρήστη

Χρήστης της εφαρμογής είναι ο πωλητής. Για να μπορέσει να λειτουργήσει την εφαρμογή θα πρέπει να του έχει δώσει ένα όνομα χρήστη και έναν κωδικό ο εργοδότης του. Αφού ανοίξει την εφαρμογή μπορεί να εισάγει τα στοιχεία του ώστε να συνδεθεί και να ξεκινήσει η λειτουργία της. Μπορεί να επιλέξει και το κουμπί "Remember me" έτσι ώστε να μην χρειάζεται να γράφει κάθε φορά το όνομα χρήστη και το κωδικό. Πατώντας το κουμπί login εισέρχεται στην εφαρμογή. Με το κουμπί logout κλείνει την εφαρμογή και σταματάει την λειτουργία της.

Πατώντας το κουμπί login γίνεται η εξής διαδικασία: Τα στοιχεία που έβαλε ως όνομα χρήστη και κωδικό στέλνονται στον διακομιστή με τον οποίο είναι συνδεδεμένος. Επειδή η επικοινωνία μεταξύ android εφαρμογής και βάσης δεδομένων δεν γίνεται, χρησιμοποιείται

ενδιάμεσος κώδικας που μετατρέπει τα δεδομένα σε κατανοητή μορφή και από τις δύο τις πλευρές.



Εικόνα 4.2 Επικοινωνία Android-Mysql Database

Τα δεδομένα στέλνονται από την εφαρμογή, ένας κώδικας PhP τα επεξεργάζεται και συνδέεται με την βάση δεδομένων για να κάνει τον έλεγχο. Αφού γίνει ο έλεγχος τα στοιχεία αποθηκεύονται σε μεταβλητές με κωδικοποίηση Json τα οποία μπορεί να επεξεργαστεί η εφαρμογή.

Συγκεκριμένα το όνομα χρήστη και ο κωδικός στέλνονται με HTTP_POST στον διακομιστή. Ο διακομιστής λαμβάνει τα δεδομένα και ενεργοποιείται ένα αρχείο το οποίο κάνει τον έλεγχο για το αν υπάρχουν τα στοιχεία στην βάση δεδομένων και αν είναι σωστά. Σε περίπτωση που είναι σωστά αποθηκεύεται σε μία μεταβλητή η τιμή 'Y' από το αγγλικό Yes η οποία λαμβάνεται από την εφαρμογή και επιτρέπει τον χρήστη να χρησιμοποιήσει την εφαρμογή. Σε περίπτωση που δεν υπάρχουν τα στοιχεία ή είναι λανθασμένα η τιμή της μεταβλητής γίνεται 'N' από το αγγλικό No και η εφαρμογή βγάζει μήνυμα λάθους.

```
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
  
login.setOnClickListener(new View.OnClickListener() {  
    public void onClick(View v) {  
        name = username.getText().toString();  
        pass = password.getText().toString();  
        String Str_check2 = app_preferences.getString("checked", "no");  
        if (Str_check2.equals("yes")) {  
            SharedPreferences.Editor editor = app_preferences.edit();  
            editor.putString("username", name);  
            editor.putString("password", pass);  
            editor.commit();  
        }  
        if (name.equals("") || pass.equals("")) {  
            Toast.makeText(Login.this, "Blank Field..Please Enter",  
                Toast.LENGTH_LONG).show();  
        } else {  
            try {  
                httpClient = new DefaultHttpClient();  
                httpPost = new HttpPost(  
                    "http://zafora.ictc.uowm.gr/~ictest00011/diplomatiki/android_connect/main.php");  
                // Add your data  
                nameValuePairs = new ArrayList<NameValuePair>(2);  
                nameValuePairs.add(new BasicNameValuePair("UserEmail",  
                    name.trim()));  
                nameValuePairs.add(new BasicNameValuePair("Password",  
                    pass.trim()));  
                httpPost.setEntity(new UrlEncodedFormEntity(  
                    nameValuePairs));  
                // Execute HTTP Post Request  
                response = httpClient.execute(httpPost);  
                inputStream = response.getEntity().getContent();  
            }  
        }  
    }  
});
```

Αφού γίνει είσοδος του πωλητή μετά από τον έλεγχο των στοιχείων ενεργοποιείται ένα Service. Αυτό το Service στέλνει τα στοιχεία της θέσης του πωλητή. Ο πρώτος έλεγχος που κάνει είναι αν είναι ανοιχτό ή όχι το GPS και ειδοποιεί με αντίστοιχο μήνυμα. Για να ενεργοποιηθεί η λειτουργία πρέπει να ενεργοποιηθεί το GPS. Όταν ενεργοποιηθεί ξεκινάει και στέλνει την θέση του πωλητή στον διακομιστή που με την σειρά του τα αποθηκεύει σε έναν πίνακα. Ο χρόνος αποστολής είναι καθορισμένος στα 5 λεπτά. Αν δεν ενεργοποιηθεί το GPS τότε το service δεν λειτουργεί. Τα στοιχεία της θέσης που στέλνει είναι το γεωγραφικό μήκος και το γεωγραφικό πλάτος. Το service ανά 5 λεπτά κάνει έλεγχο για την θέση του πωλητή. Αν υπάρξει αλλαγή στην θέση τότε στέλνει τα νέα δεδομένα αλλιώς για λόγους οικονομίας δεν στέλνει τα ήδη υπάρχοντα στοιχεία.

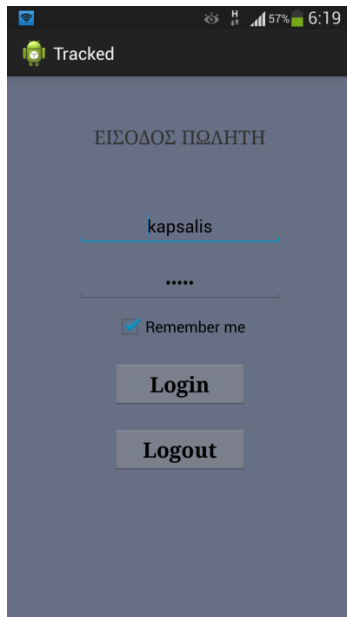
```
49     @Override
50     public void onLocationChanged(Location loc) {
51         // Toast.makeText(
52         //     getBaseContext(),
53         //     String.valueOf(loc.getLatitude()) + "\n"
54         //     + String.valueOf(loc.getLongitude()),
55         //     Toast.LENGTH_SHORT).show();
56
57         Double latitude = loc.getLatitude();
58         Double longitude = loc.getLongitude();
59         String slatitude = String.valueOf(latitude);
60         String slongitude = String.valueOf(longitude);
61         List<NameValuePair> params = new ArrayList<NameValuePair>();
62         params.add(new BasicNameValuePair("latitude", slatitude));
63         params.add(new BasicNameValuePair("longitude", slongitude));
64         JSONObject json = jsonParser.makeHttpRequest(url_post_loc, "POST",
65             params);
66         Log.d("Create Response", json.toString());
67
68
69     }
```

Το διάστημα των 5 λεπτών επιλέχθηκε μετά από έλεγχο και έρευνα. Κάθε πώληση διαρκεί το λιγότερο 5 λεπτά οπότε με αυτό το χρονικό διάστημα δίνεται η δυνατότητα ελέγχου των σημείων που έγιναν οι πωλήσεις και η διάρκεια ζωής της μπαταρίας είναι αρκετή για τον χρόνο που χρειάζεται η εφαρμογή.

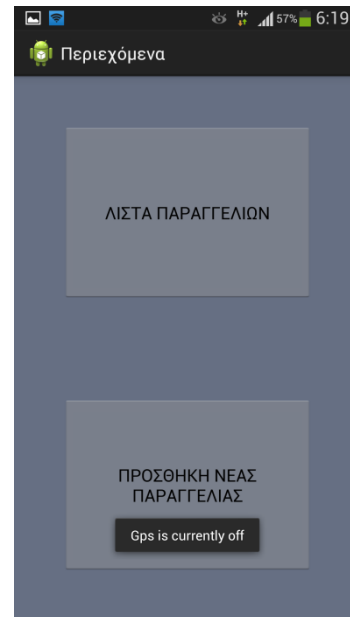
```
35     @Override
36     public void onCreate() {
37         super.onCreate();
38         locMgr = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
39         locMgr.requestLocationUpdates(LocationManager.GPS_PROVIDER,
40             5 * 1000 * 60, 0, this);
41     }
```

4.3.2 Κεντρικό μενού

Στο κεντρικό μενού υπάρχουν δύο διαθέσιμες επιλογές για τον χρήστη. Η μία είναι να βλέπει την λίστα με τις παραγγελίες που έκανε την τρέχουσα μέρα και η άλλη να προσθέσει καινούργια παραγγελία. Η επιλογή για την λίστα παραγγελιών είναι διαθέσιμη ώστε να μπορεί ο πωλητής να επεξεργαστεί τις παραγγελίες για τυχόν αλλαγές μετά την καταχώρηση τους και να μπορεί να ελέγξει αν έχει πετύχει τον στόχο που ίσως του έχει δοθεί από την εταιρία.



Εικόνα 4.3 :Login



Εικόνα 4.4 Main Menu

4.3.3 Λίστα παραγγελιών

Πατώντας το κουμπί λίστα παραγγελιών ο πωλητής έχει την δυνατότητα να δει τις παραγγελίες της τρέχουσας ημέρας. Εμφανίζεται μία λίστα με την ημερομηνία τις παραγγελίας(η ημερομηνία δεν είναι κύριο χαρακτηριστικό και δεν μπορεί εύκολα ο πωλητής να ξεχωρίσει τις παραγγελίες από αυτήν, αλλά επειδή η διπλωματική δεν ασχολείται με τους πελάτες αλλά με τους πωλητές μπήκε σαν χαρακτηριστικό επιλογής η ημερομηνία) η οποία είναι ενεργή και μπορεί να πατηθεί μεταφέροντας τον χρήστη στην επόμενη καρτέλα η οποία εμφανίζει όλα τα στοιχεία τις παραγγελίας και δύο επιπλέον κουμπιά, την καταχώρηση αλλαγών και την διαγραφή.


```
48     @Override
49     public void onCreate(Bundle savedInstanceState) {
50         super.onCreate(savedInstanceState);
51         setContentView(R.layout.all_orders);
52
53         // Hashmap for ListView
54         ordersList = new ArrayList<HashMap<String, String>>();
55
56         // Loading products in Background Thread
57         new LoadAllOrders().execute();
58
59         // Get listview
60         ListView lv = getListView();
61
62         // on seleting single product
63         // launching Edit Product Screen
64         lv.setOnItemClickListener(new OnItemClickListener() {
65
66             @Override
67             public void onItemClick(AdapterView<?> parent, View view,
68                 int position, long id) {
69                 // getting values from selected ListItem
70                 String pid = ((TextView) view.findViewById(R.id.pid)).getText()
71                     .toString();
72
73                 // Starting new intent
74                 Intent in = new Intent(getApplicationContext(),
75                     EditOrderActivity.class);
76                 // sending pid to next activity
77                 in.putExtra(TAG_PID, pid);
78
79                 // starting new activity and expecting some response back
80                 startActivityForResult(in, 100);
81             }
82         });
83
84     }
```

Για να εμφανίσει τις παραγγελίες όταν πατηθεί το κουμπί ενεργοποιείται η μέθοδος LoadAllOrders, η οποία “ζητάει” όλες τις παραγγελίες της τελευταίας ημέρας από τον διακομιστή. Η διαδικασία ανάκτησης των παραγγελιών είναι η εξής. Το αρχείο Php κάνει έλεγχο στον πίνακα orders και βρίσκει της παραγγελίες που ανήκουν στον εκάστοτε πωλητή και την τρέχουσα ημέρα. Αποθηκεύονται σε ένα πίνακα όλα τα στοιχεία που απαιτούνται και στέλνονται κωδικοποιημένα με json στην εφαρμογή. Η εφαρμογή δέχεται μία μεταβλητή success η οποία αν είναι 1 σημαίνει ότι υπάρχουν στοιχεία και τα εμφανίζει αλλιώς εμφανίζει ότι δεν υπάρχουν διαθέσιμες εγγραφές και παραπέμπει στην δημιουργία νέας παραγγελίας.

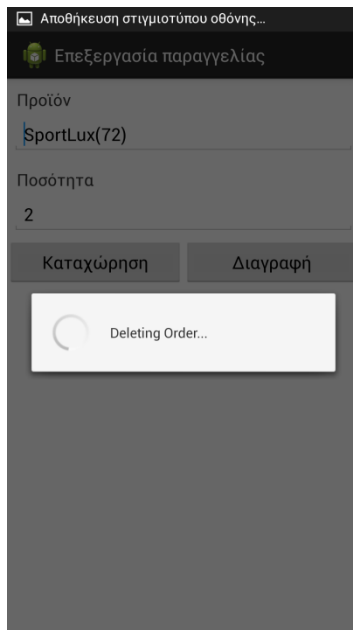
4.3.4 Καταχώρηση αλλαγών-Διαγραφή παραγγελιών

Αφού εμφανιστεί η λίστα ο χρήστης έχει τη δυνατότητα να πατήσει πάνω σε μία παραγγελία και να δει επιπλέον στοιχεία της παραγγελίας. Επίσης του δίνεται η δυνατότητα να αλλάξει στοιχεία της παραγγελίας ή και να την διαγράψει.

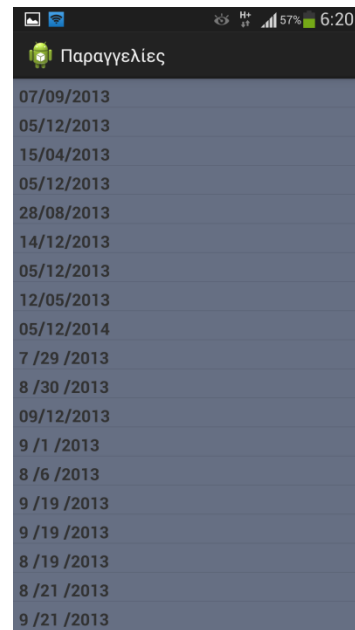
Τα στοιχεία της παραγγελίας λαμβάνονται από την βάση δεδομένων με τον τρόπο που αναφέρθηκε προηγουμένως. Μετά από την εμφάνιση των στοιχείων της παραγγελίας ο χρήστης μπορεί να επεξεργαστεί ένα ή και περισσότερα στοιχεία της παραγγελίας. Αφού αλλάξει κάτι και πατήσει καταχώρηση αλλαγών τα καινούρια στοιχεία στέλνονται στην βάση δεδομένων μέσω ενός PHP αρχείου που προσπαθεί να αλλάξει τα παλιά στοιχεία με τα καινούρια. Αν η αλλαγή ήταν επιτυχής ο χρήστης ενημερώνεται με ένα μήνυμα επιτυχίας αλλιώς ενημερώνεται για την αποτυχία αλλαγής των στοιχείων.

```
202 // getting updated data from EditTexts
203 String product = txtProduct.getText().toString();
204 String quantity = txtQuantity.getText().toString();
205 // Building Parameters
206 List<NameValuePair> params = new ArrayList<NameValuePair>();
207 params.add(new BasicNameValuePair(TAG_PID, pid));
208 params.add(new BasicNameValuePair(TAG_PRODUCT, product));
209 params.add(new BasicNameValuePair(TAG_QUANTITY, quantity));
210
211 // sending modified data through http request
212 // Notice that update product url accepts POST method
213 JSONObject json = jsonParser.makeHttpRequest(url_update_order,
214     "POST", params);
215
216 // check json success tag
217 try {
218     int success = json.getInt(TAG_SUCCESS);
219
220     if (success == 1) {
221         // successfully updated
222         Intent i = getIntent();
223         // send result code 100 to notify about product update
224         setResult(100, i);
225         finish();
226     } else {
227         // failed to update product
228     }
229 } catch (JSONException e) {
230     e.printStackTrace();
231 }
232
233 return null;
234 }
```

Η διαγραφή λειτουργεί με τον ίδιο τρόπο. Πατώντας ο χρήστης τον κουμπί διαγραφή το PHP αρχείο προσπαθεί να σβήσει την υπάρχουσα διαγραφή από την βάση. Σε περίπτωση επιτυχίας ή αποτυχίας ο χρήστης ενημερώνεται με αντίστοιχο μήνυμα.



Εικόνα 4.5 Διαγραφή παραγγελίας

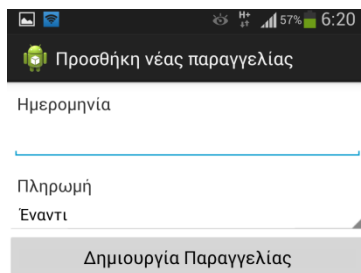


Εικόνα 4.6 Λίστα παραγγελιών

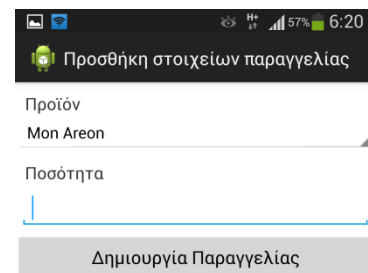
4.3.5 Νέα παραγγελία

Στην προσθήκη νέας παραγγελίας ο χρήστης έχει την δυνατότητα να προσθέσει μία νέα παραγγελία η οποία θα καταχωρηθεί στην βάση δεδομένων. Πατώντας το κουμπί προσθήκη νέας παραγγελίας εμφανίζεται μία νέα οθόνη με 2 πεδία εισαγωγής. Την ημερομηνία και τον τρόπο πληρωμής. Πατώντας το πεδίο ημερομηνία εμφανίζεται ένα ημερολόγιο για την καταχώρηση ημερομηνίας. Στο πεδίο πληρωμή εμφανίζεται μία λίστα με μία από τους 4 δυνατούς τρόπους πληρωμής που έχουν αναφερθεί στο προηγούμενο κεφάλαιο. Πατώντας συνέχεια ο χρήστης μεταβαίνει στην επόμενη οθόνη η οποία είναι για να εισάγει τα προϊόντα τα οποία παρήγγειλε ο πελάτης. Τα πεδία και εδώ είναι 2 το προϊόν και η ποσότητα. Στο πεδίο προϊόν εμφανίζεται μία λίστα με τα διαθέσιμα προϊόντα που είναι αποθηκευμένα στην εφαρμογή και δεν

γίνεται σύνδεση κάθε φορά με την βάση για να μην υπάρχει τόσο μεγάλος όγκος δεδομένων που χρειάζονται αποστολή. Το άλλο πεδίο είναι η ποσότητα των προϊόντων που παραγγέλνει ο πελάτης. Πατώντας δημιουργία παραγγελίας ο χρήστης μεταβαίνει στην αρχική οθόνη με τις παραγγελίες που είναι ανανεωμένη με την καινούρια παραγγελία.



Εικόνα 4.7 Προσθήκη παραγγελίας



Εικόνα 4.8 Προσθήκη στοιχείων παραγγελίας

4.4 Στοιχεία κώδικα εφαρμογής

Για την δημιουργία της εφαρμογής δημιουργήθηκαν 9 αρχεία xml μεγέθους 13KB με γραμμές κώδικα 453. Επίσης 9 αρχεία java με συνολικό μέγεθος 39,1KB και συνολικές γραμμές κώδικα 1389. Από τα 9 αρχεία xml τα 8 δημιουργήθηκαν για την εμφάνιση των διεπαφών του χρήστη το ένα που μένει είναι το `Android.Mainifest.xml` που καθορίζει τον ρόλο όλων των αρχείων.

Κεφάλαιο 5

Τεχνολογίες που χρησιμοποιήθηκαν

5.1 Html

Η Html (1) δεν είναι μία γλώσσα προγραμματισμού (αν και ονομάζουμε όσους την χρησιμοποιούν Html programmers). Είναι μια περιγραφική γλώσσα. Δηλαδή ένας ειδικός τρόπος γραφής κειμένου και κλήσης άλλων αρχείων ή εφαρμογών βασισμένος σε οδηγίες. Στην παρούσα διπλωματική χρησιμοποιήθηκε για την κατασκευή της ιστοσελίδας.

Για τη δημιουργία ενός ιστοχώρου υπάρχουν δύο βασικοί τρόποι. Η χρησιμοποίηση ενός αυτόματου προγράμματος για να δημιουργηθεί η σελίδα με τρόπο ανάλογο της δημιουργίας ενός εγγράφου στο word, ή να δημιουργηθούν τα πάντα από το μηδέν με την χρήση κώδικα Html.

Με την χρήση ενός αυτόματου προγράμματος θα μπορούσε να γίνει η κατασκευή του ιστοχώρου σε πολύ μικρό χρονικό διάστημα και η ταχύτητα ανανέωσης θα ήταν πολύ μεγάλη αφού το πρόγραμμα φροντίζει για την αυτόματη ενημέρωση των ιστοσελίδων με τις αλλαγές που έχουν γίνει μετά την τελευταία ανανέωση. Όμως κάθε αυτόματο πρόγραμμα αποτελεί ουσιαστικά έναν μεταφραστή που μετατρέπει τη σχεδίαση και τις επιθυμίες μας σε γλώσσα Html και καμία μετάφραση δεν είναι άψογη, ειδικά αν γίνεται από υπολογιστή. Επίσης κάθε αυτόματο πρόγραμμα υποστηρίζει μόνο έναν υποσύνολο όλων των δυνατοτήτων της γλώσσας Html. Τέλος επειδή η δημιουργία της ιστοσελίδας έγινε από το μηδέν και γίνονται κατανοητά οι ιδιομορφίες και τα χαρακτηριστικά της, είναι πολύ πιο εύκολο να βρεθούν και να διορθωθούν τα όποια προβλήματα εμφανιστούν σε αυτήν.

5.1.1 Ιστορική αναδρομή

Ο φυσικός Τιμ Μπέρνερς Λι, επινόησε ένα σύστημα χρήσης και διαμοιρασμού εγγράφων για τους ερευνητές του CERN και κατασκεύασε ένα πρωτότυπό του. Αργότερα το 1989, πρότεινε ένα σύστημα βασισμένο στο διαδίκτυο, το οποίο θα χρησιμοποιούσε υπερκείμενο.

Η πρώτη δημόσια διαθέσιμη περιγραφή της Html ήταν ένα έγγραφο με το όνομα Ετικέτες HTML, το οποίο πρωτοαναφέρθηκε στο Διαδίκτυο από τον Μπέρνερς Λι στα τέλη του 1991. Περιέγραφε τα 20 στοιχεία τα οποία αποτελούσαν τον αρχικό και σχετικά απλό σχεδιασμό της Html. Η Html είναι βασισμένη στην γλώσσα SGML που ήταν φτιαγμένη στο CERN. Λόγω έλλειψης χρηματοδότησης η ανάπτυξη της Html ανατίθεται στην ομάδα εργασίας IIR του IETF(Internet Engineering Task Force). Σήμερα οι προδιαγραφές Html τηρούνται, από το World Wide Web Consortium(W3C) και το 2000 έγινε παγκόσμιο πρότυπο.



Εικόνα 5.1: Το λογότυπο της Html

```
3 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4 <html xmlns="http://www.w3.org/1999/xhtml">
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <link href="userindex.css" rel="stylesheet" type="text/css" />
8 <title>Target tracking</title>
9 </head>
```

Εικόνα 5.2: Παράδειγμα Html

5.2 CSS

Η Css (2) (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μία γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες Html και Xhtml, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Χρησιμοποιήθηκε για τον σχεδιασμό του ιστοχώρου και την εμφάνιση του. Η Css έχει:

- Πολύ μεγαλύτερη ευελιξία. Κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασσική Html.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου ιστοχώρου μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο Css. Έτσι κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μια μοναδική αλλαγή σε αυτό το αρχείο
- Μικρότερο μέγεθος αρχείου καθώς κάθε κανόνας μορφοποίησης γράφεται μόνο μία φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO(Search engine optimization). Οι μηχανές αναζήτησης δεν «μπερδεύονται» ανάμεσα σε περιεχόμενο και τη μορφοποίηση του, αλλά έχουν πρόσβαση στο περιεχόμενο απομονωμένο από την μορφοποίηση.
- Όταν χρησιμοποιείται εξωτερικό αρχείο Css ο browser την πρώτη φορά που θα φορτώσει κάποια σελίδα του ιστοχώρου θα το αποθηκεύσει στην cache, οπότε δεν χρειάζεται να το κατεβάσει ξανά κάθε φορά που κατεβάζει ο χρήστης κάποια άλλη σελίδα της ιστοχώρου.

5.2.1 Ιστορική αναδρομή

Τα φύλλα στυλ υπήρχαν σε κάποια μορφή από την αρχή δημιουργίας της Sgml το 1980. Η Css αναπτύχθηκε για να δώσει έναν συγκεκριμένο τρόπο προσέγγισης του στυλ παρουσίασης πληροφοριών σε αρχεία του διαδικτύου. Όσο μεγάλωνε η χρήση της Html χρειάστηκαν περισσότερες στιλιστικές δυνατότητες. Αυτή η εξέλιξη έδινε περισσότερες δυνατότητες στην παρουσίαση της ιστοσελίδας με κόστος όμως πολύ πιο σύνθετα Html αρχεία. Ο Robert Cailliau ήθελε να διαχωρίσει τη δομή της ιστοσελίδας από την παρουσίαση. Ο ιδανικός τρόπος θα ήταν να υπάρχουν τρία διαφορετικά φύλλα μορφοποίησης: ένα για εκτύπωση ένα για την παρουσίαση στην οθόνη και ένα για τον συντάκτη.

Έτσι προτάθηκαν εννέα διαφορετικές γλώσσες φύλλων μορφοποίησης στο W3C από τις οποίες δύο επιλέχθηκαν οι οποίες αποτέλεσαν την βάση της Css, η Chss και η Ssp. Το 1995 ο Lie και ο Bos που δουλεύανε στην ανάπτυξη του browser Arena δημιούργησαν την Css η οποία μπορούσε να εφαρμοστεί και σε άλλες γλώσσες πέραν της Html. Η πρόταση έγινε δεκτή από τον οργανισμό W3C που συνέχισε και τελειοποίησε την γλώσσα και το 1996 παρουσιάστηκε στο κοινό. Το 1997 παρουσιάστηκε η Css επιπέδου 2 η οποία είχε διορθώσεις και το 1998 έγινε η Css επιπέδου 3 που χρησιμοποιείται μέχρι σήμερα.

```
1 body {
2   margin: 0;
3   background-image: url('bodybg.gif');
4   /*background-color: #AAAAAA;*/
5   background-repeat: repeat-x;
6   height: 100%;
7   padding: 0;
8   text-align: center;
9 }
10
```

Εικόνα 5.3: Παράδειγμα μορφοποίησης με CSS

5.3 Php

Η Php (3) είναι μία γλώσσα προγραμματισμού για την δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα Php περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού, ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα Html. Η Php χρησιμοποιήθηκε για την επεξεργασία των δεδομένων που στέλνει ο χρήστης στον διακομιστή και το αντίστροφο.

- Είναι ανοιχτού κώδικα με υποστήριξη πολλαπλών πλατφόρμων και ευκολία στην χρήση.
- Τα αποτελέσματα που παράγει, αλλάζουν σύμφωνα με τις ανάγκες του χρήστη.
- Έχει διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων.
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες διαδικασίες διαδικτύου.
- Χαμηλό κόστος-Παρέχεται δωρεάν.
- Φορητότητα -Είναι διαθέσιμη για πολλά λειτουργικά συστήματα.
- Διαθεσιμότητα του κώδικα προέλευσης.
- Η Php είναι γλώσσα δυναμική και παρουσιάζει πολλές ομοιότητες με την γλώσσα C++.

5.3.1 Ιστορική αναδρομή

Η ιστορία της Php ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl ένα απλό script με όνομα php.cgi για προσωπική χρήση. Τα script αυτό είχε σαν σκοπό να διατηρεί μία λίστα στατιστικών για τα άτομα που έβλεπαν το βιογραφικό του σημείωμα. Η γλώσσα τότε ονομαζόταν

Php/Fi από τα αρχικά Personal Home Page/Form Interpreter. Το 1997 η γλώσσα έφθασε τη έκδοση 2.0, βασιζόμενη αυτή την φορά στη γλώσσα C και αριθμώντας περισσότερους από 50000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η Php έφθασε στην έκδοση 3.0 που θύμιζε περισσότερο τη σημερινή της μορφή. Το 1998 ακολούθησε η έκδοση 4 και τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης Php 6. Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της Php.



Εικόνα 5.4: Το λογότυπο της Php

```
26 <?php
27 db_open();
28 $username=$_SESSION['username'];
29 $res=mysql_query("SELECT user_id FROM users WHERE username='$username'");
30 $row1=mysql_fetch_array($res);
31 $user_id=$row1['user_id'];
32
33 $result=mysql_query("SELECT * FROM orders WHERE user_id='$user_id'");
```

Εικόνα 5.5: Επιλογή στοιχείων πίνακα με PHP

5.4 MySql

Η MySql (4) είναι ένα πολύ γρήγορο και δυνατό, σύστημα διαχείρισης βάσεων δεδομένων. Ο MySql διακομιστής ελέγχει την πρόσβαση στα δεδομένα για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι

μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Χρησιμοποιεί την Sql(Structured Query Language) την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια. Είναι ανοιχτού κώδικα, οποιοσδήποτε μπορεί να την κατεβάσει και να την διαμορφώσει ανάλογα με τις ανάγκες του σύμφωνα πάντα με την γενική άδεια που υπάρχει. Μπορεί να λειτουργήσει σε περιβάλλον Linux, Unix και Windows. Η βάση δεδομένων που αποθηκεύονται όλα τα στοιχεία δημιουργήθηκε με Mysql.

- Είναι πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων.
- Μπορούν να υπάρχουν ταυτόχρονα περισσότερες από μία συνδέσεις με τη βάση χωρίς να υπάρχουν πολλαπλά αντίγραφα της, όπως συμβαίνει με άλλα συστήματα βάσεων δεδομένων.
- Παρέχει ευκολίες στο backup.
- Είναι ιδιαίτερα βελτιωμένη σε ταχύτητα για την ανάκτηση δεδομένων.
- Είναι οικονομική.
- Ανοιχτού κώδικα.
- Συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και για διάφορα εργαλεία ανάπτυξης.
- Υποστηρίζεται από πολλές διαδικτυακές κοινότητες.
- Αναπτύσσεται συνεχώς.
- Συνεργάζεται άψογα με την Php.

5.4.1 Ιστορική αναδρομή

Σχεδιαστής της MySQL είναι ο Michael Widenius και ο David Axmark στις αρχές του 1995. Υπήρξαν αρκετές δικαστικές διαμάχες ανάμεσα σε δύο εταιρείες για το ποιος ήταν ο ιδρυτής της ιδέας οι οποίες κατέληξαν σε συμβιβασμό.



Εικόνα 5.6: Το λογότυπο της MySql

```
1 CREATE TABLE table_name
2 (
3   column_name1 data_type (size),
4 );
```

Εικόνα 5.7: Δημιουργία πίνακα με MySql

5.5 JavaScript

Η JavaScript (5) είναι γλώσσα προγραμματισμού η οποία έχει σαν σκοπό την παραγωγή δυναμικού περιεχομένου και την εκτέλεση κώδικα στην πλευρά του πελάτη(client-side) σε ιστοσελίδες. Είναι γλώσσα σεναρίων που βασίζεται στα πρωτότυπα, είναι δυναμική με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C.

- Είναι πολύ «ελαφριά» γλώσσα.
- Προγραμματίζει κώδικα που μπορεί να εισαχθεί σε ιστοσελίδες φτιαγμένες με Html.
- Μπορεί να εκτελεστεί από οποιονδήποτε σύγχρονο browser.
- Είναι εύκολη στην εκμάθηση.
- Πολλά διαθέσιμα εργαλεία.
- Χρήση Ajax.

5.5.1 Ιστορική αναδρομή

Δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα μετονομάστηκε σε LiveScript και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξη της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. Κυκλοφόρησε πρώτη φορά σαν LiveScript σε δοκιμαστικές εκδόσεις τον Σεπτέμβριο του 1995. Μετονομάστηκε σε JavaScript το Δεκέμβριο του 1995 από την εταιρεία Sun Microsystems.

Απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά(client-side) του πελάτη για εκτέλεση κώδικα και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε Jscript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων.

```
44 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.3/jquery.min.js"></script>
45 <script type="text/javascript">
46 $(document).ready(function() {
47     $('#btnAdd').click(function() {
48         var num = $('.clonedInput').length; // how many "duplicatable" input fields we currently have
49         var newNum = new Number(num + 1); // the numeric ID of the new input field being added
50         var c = ('ctable' + newNum);
51         // create the new element via clone(), and manipulate it's ID using newNum value
52         var newElem = $('#input' + num).clone().attr('id', 'input' + newNum);
```

Εικόνα 5.8: Script δημιουργίας κουμπιού για εμφάνιση ημερομηνίας με JavaScript

5.6 Xampp

Το Xampp (6) είναι ένα πακέτο προγραμμάτων ελεύθερου λογισμικού, ανοιχτού κώδικα και ανεξαρτήτου πλατφόρμας το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων http Apache, την βάση δεδομένων MySql και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού Php και Perl.

- Η εφαρμογή είναι ανεξάρτητη λειτουργικού συστήματος.
- Ανοιχτού κώδικα.
- Προσιτή στην διαχείριση.
- Δοκιμή των λειτουργιών της ιστοσελίδας χωρίς να υπάρχει πρόσβαση στο διαδίκτυο.





Εικόνα 5.10: FreeBSD Logo

```
FreeBSD 6.2-RELEASE-p4 (GENERIC) #0: Thu Apr 26 17:40:53 UTC 2007
Welcome to FreeBSD!

Before seeking technical support, please use the following resources:

o Security advisories and updated errata information for all releases are
  at http://www.FreeBSD.org/releases/ - always consult the ERRATA section
  for your release first as it's updated frequently.

o The Handbook and FAQ documents are at http://www.FreeBSD.org/ and,
  along with the mailing lists, can be searched by going to
  http://www.FreeBSD.org/search/. If the doc distribution has
  been installed, they're also available formatted in /usr/share/doc.

If you still have a question or problem, please take the output of
'uname -a', along with any relevant error messages, and email it
as a question to the questions@FreeBSD.org mailing list. If you are
unfamiliar with FreeBSD's directory layout, please refer to the hier(7)
manual page. If you are not familiar with manual pages, type 'man man'.

You may also use sysinstall(8) to re-enter the installation and
configuration utility. Edit /etc/motd to change this login announcement.

$
```

Εικόνα 5.11: FreeBSD

5.8 Notepad++

Το Notepad++ (8) αποτελεί έναν από τους διασημότερους
επεξεργαστές κειμένου με πάνω από 27εκατομμύρια χρήστες.

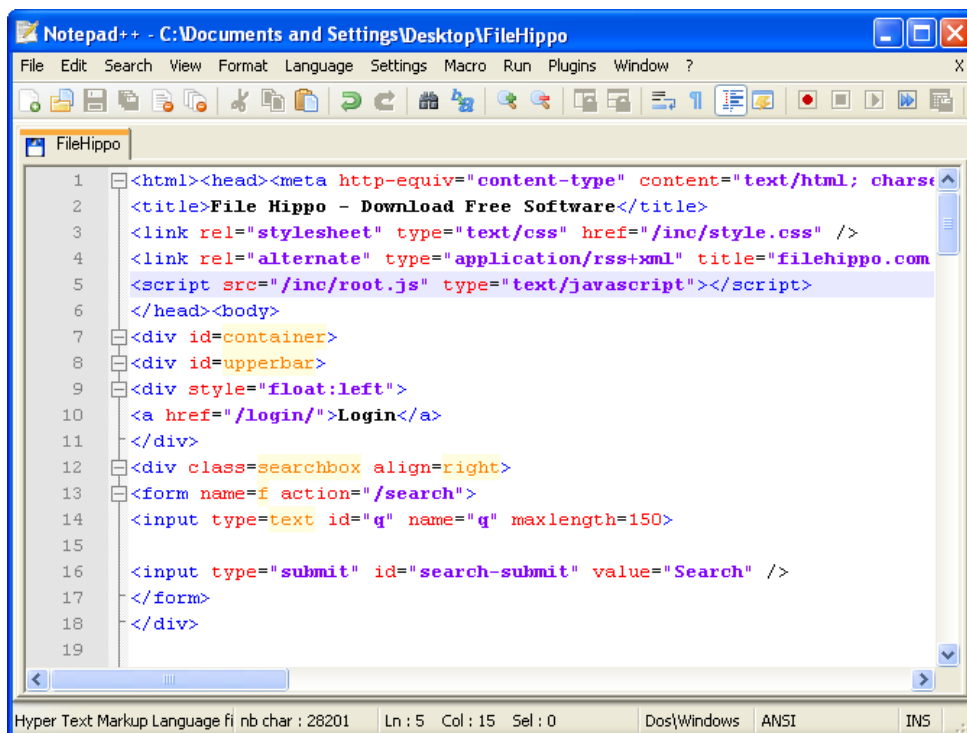
- Υποστηρίζει καρτέλες.
- Επισήμανση λαθών για πλειάδα γλωσσών προγραμματισμού.
- Εκτύπωση χρωματισμένου κώδικα.
- Υποστήριξη επεκτάσεων.
- Φορητή έκδοση

5.8.1 Ιστορική αναδρομή

Ιδρυτής του Notepad++ είναι ο Don Ho και είναι γραμμένο σε C++. Σκοπός είναι να έχει μεγάλη απόδοση μειώνοντας το μέγεθος του προγράμματος και καταναλώνοντας όσο το δυνατόν λιγότερη Cpu.



Εικόνα 5.12: Notepad++ Logo



Εικόνα 5.13: Περιβάλλον Notepad++

5.9 Google Maps-API

Το Google maps είναι μια πλατφόρμα βασισμένη στο διαδίκτυο που τροφοδοτεί πολλές υπηρεσίες που βασίζονται σε χάρτες. Προσφέρει, δρόμους, σχεδιάζει δρομολόγια για ταξίδια με αμάξι ή με τα πόδια ή με δημόσια μέσα μαζικής μεταφοράς και εντοπίζει επιχειρήσεις σε όλο τον κόσμο. Η ανανέωση των χαρτών από τους δορυφόρους δεν γίνεται σε πραγματικό χρόνο αλλά είναι αρκετών μηνών ή και χρόνων παλαιοί.

Το Google Maps Api (9) επιτρέπει σε σχεδιαστές να χρησιμοποιήσουν την πλατφόρμα Google Maps στις ιστοσελίδες του. Είναι δωρεάν και επιτρέπει στον σχεδιαστή να ενσωματώσει την ιστοσελίδα του Google Maps στην δικιά του ιστοσελίδα.

Δεν υπάρχει παρόμοια πλατφόρμα που να υποστηρίζει την παρουσίαση χαρτών τόσο αναλυτικά και με τόσες πληροφορίες όσο τα Google maps. Έχουν βέβαια παρουσιαστεί χάρτες από άλλες εταιρίες όπως είναι η Apple αλλά η λειτουργία τους δεν είναι ακόμα σε τόσο ικανοποιητικό επίπεδο.

5.9.1 Ιστορική αναδρομή

Τα Google Maps ξεκίνησαν σαν ένα πρόγραμμα σε C++ σχεδιασμένο από δύο αδέρφια από την Δανία. Η ιδέα αρχικά ήταν να σχεδιαστεί για να μπορεί να το κατεβάσει ο χρήστης αλλά υπερνίκησε η ιδέα να είναι καθαρά διαδικτυακό προϊόν της Google, η οποία και αγόρασε την εταιρεία των δύο αδερφών. Πρώτη παρουσίαση έγινε το 2005 και έκτοτε αναπτύσσεται συνεχώς. Προστέθηκαν υπηρεσίες όπως η θέαση πρώτου προσώπου του δρόμου για μεγάλες πόλεις και πρόσθεση φωτογραφιών από τοπία.

Το Google Maps Api δημιουργήθηκε το 2005 και αρχικά ήταν μόνο JavaScript Api, στην συνέχεια όμως επεκτάθηκε ώστε να μπορεί να χρησιμοποιηθεί και από Adobe Flash εφαρμογές. Πάνω από 350.000 ιστοσελίδες το Google Maps Api που το κάνει την πιο διαδεδομένη διαδικτυακή εφαρμογή Api.



Εικόνα 5.14: Google Maps Logo

5.10 Android

Το Android (10) είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών ανεπτυγμένων από την Google.

- Ανοιχτή πλατφόρμα.
- Απλοποιημένη ανάπτυξη. Τόσο οι πελάτες όσο και οι προγραμματιστές ζητούν εύκολα στη χρήση εργαλεία.
- Ισχυρό λειτουργικό σύστημα
- Πλήρες σύνολο βιβλιοθηκών, με πολλαπλές δυνατότητες χρήσης πολυμέσων και πλήρες σετ εφαρμογών τηλεφωνίας.

5.10.1 Ιστορική αναδρομή

Πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι

αφιερωμένες στην ανάπτυξη και εξέλιξη προτύπων στις συσκευές κινητής τηλεφωνίας.



Εικόνα 5.15: Android logo

5.11 Eclipse

Το Eclipse (11) αποτελεί ένα εργαλείο ανάπτυξης λογισμικού, ένα ολοκληρωμένο δηλαδή περιβάλλον ανάπτυξης, μέσα από το οποίο μπορεί να εκτελεστεί κώδικας. Είναι σχεδιασμένο κατά κύριο λόγο με Java.

- Απλό περιβάλλον.
- Υποστηρίζει πολλές γλώσσες προγραμματισμού.
- Υποστήριξη σε διαδικτυακές κοινότητες.
- Προαπαιτούμενο για προγραμματισμό σε Android

5.11.1 Ιστορική αναδρομή

Ξεκίνησε ως ένα εργαλείο ανοιχτού κώδικα από την IBM. Η IBM επένδυσε πάνω από 40 εκατομμύρια δολάρια προτού μία συννομοσπονδία αποτελούμενη από 80 μέλη μαζί με την IBM ιδρύσει το ίδρυμα Eclipse. Από τότε το Eclipse έγινε ένα πολύ διαδεδομένο εργαλείο.

5.12 <http://zafora.icte.uowm.gr>

Διακομιστής του τμήματος Μηχανικών Πληροφορικής Και Τηλεπικοινωνιών που είναι ελεύθερος για όλους τους φοιτητές. Το ανέβασμα των αρχείων γίνεται σε αυτόν τον διακομιστή ώστε να υπάρχει άμεσος έλεγχος και βοήθεια από τον επιβλέπων καθηγητή.

5.13 JAVA

Η Java (12) είναι αντικειμενοστραφής γλώσσα προγραμματισμού που σχεδιάστηκε από την εταιρεία πληροφορικής Sun Microsystems. Ένα από τα βασικά πλεονεκτήματα της Java έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε Java τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh. Για να επιτευχθεί όμως αυτό χρειαζόταν κάποιος τρόπος έτσι ώστε τα προγράμματα γραμμένα σε Java να μπορούν να είναι «κατανοητά» από κάθε υπολογιστή ανεξάρτητα του είδους επεξεργαστή αλλά και λειτουργικού συστήματος. Ο λόγος είναι ότι κάθε κεντρική μονάδα επεξεργασίας κατανοεί διαφορετικό κώδικα μηχανής. Ο συμβολικός κώδικας (assembly) που μεταφράζεται και εκτελείται σε Windows είναι διαφορετικός από αυτόν που μεταφράζεται και εκτελείται σε έναν υπολογιστή Macintosh. Η λύση δόθηκε με την ανάπτυξη της Εικονικής Μηχανής. Η επιλογή της Java έγινε γιατί ο προγραμματισμός της εφαρμογής σε smart phone γίνεται υποχρεωτικά σε αυτή την γλώσσα

5.13.1 Ιστορικά

Στις αρχές του 1991, η Sun αναζητούσε το κατάλληλο εργαλείο για να αποτελέσει την πλατφόρμα ανάπτυξης λογισμικού σε μικρό-συσκευές (έξυπνες οικιακές συσκευές έως πολύπλοκα συστήματα παραγωγής γραφικών). Τα εργαλεία της εποχής ήταν γλώσσες όπως η C++ και η C. Μετά από διάφορους πειραματισμούς προέκυψε το συμπέρασμα ότι οι

,0

υπάρχουσες γλώσσες δεν μπορούσαν να καλύψουν τις ανάγκες τους. Το όνομά της το πήρε από το ομώνυμο δένδρο (βελανιδιά).



Εικόνα 5.16 Java Logo

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Η εφαρμογή που δημιουργήθηκε έχει ως κύριο στόχο την απλοποίηση της καταγραφής των παραγγελιών, τον έλεγχο της αποθήκης και της δυναμικότητας των πωλητών. Πρώτος στόχος ήταν η δημιουργία ιστοχώρου τον οποίο θα ελέγχει ο ιδιοκτήτης της εταιρίας. Τα στοιχεία που μπορεί να ελέγχει είναι τα διαθέσιμα προϊόντα, οι πωλητές, οι παραγγελίες και στοιχεία σχετικά με την θέση των πωλητών. Δεύτερος στόχος ήταν η δημιουργία εφαρμογής σε smart phone με λειτουργικό android. Στην εφαρμογή ο πωλητής θα μπορεί να στείλει τις παραγγελίες κατευθείαν στον διακομιστή και ο ιδιοκτήτης θα μπορεί να ελέγχει άμεσα την πορεία των παραγγελιών. Επίσης θα στέλνεται το στίγμα της θέσης του πωλητή το οποίο θα μπορεί να δει άμεσα ο ιδιοκτήτης.

Η αρχική ιδέα ήταν η δημιουργία ιστοχώρου που θα εξυπηρετούσε έναν χρήστη αλλά κατά την διάρκεια της δημιουργίας της κρίθηκε σκόπιμο να εξυπηρετεί μεγαλύτερο όγκο χρηστών, όπως και έγινε. Η εφαρμογή αντιθέτως μπορεί να ελέγξει έναν πωλητή για κάθε ιδιοκτήτη.

Τα αποτελέσματα ήταν ικανοποιητικά πετυχαίνοντας τον μεγαλύτερο όγκο των αρχικών στόχων. Η εφαρμογή δοκιμάστηκε από εταιρία μεσαίου μεγέθους με 24 προϊόντα και κρίθηκε λειτουργική με περιθώρια βελτίωσης. Στα θετικά ήταν η εύκολη χρήση της εφαρμογής σε smart phone η δυνατότητα παρακολούθησης των πωλητών χωρίς μεγάλη χρέωση και η λειτουργικότητα της ιστοσελίδας. Στα αρνητικά ήταν ο μικρός όγκος δεδομένων που καταχωρούνται σχετικά με τα προϊόντα για της αποθήκης και οι λίγες δυνατότητες της εφαρμογής σε smart phone.

Το αποτέλεσμα κρίθηκε ικανοποιητικό και η εφαρμογή μπορεί να χρησιμοποιηθεί έχοντας καλά αποτελέσματα για εταιρίες μικρού προς μεσαίου μεγέθους.

6.2 Μελλοντικά

Μελλοντικά είναι απαραίτητη η υποστήριξη πολλών πωλητών για κάθε χρήστη ώστε να είναι δυνατή η παρακολούθηση όλων. Επίσης η ιστοσελίδα μπορεί να αλλαχθεί κυρίως στα προϊόντα στον τομέα της αποθήκης για να μπορεί να κρατάει μεγαλύτερο αριθμό πληροφοριών για κάθε προϊόν. Η σύνδεση της βάσης δεδομένων με αρχεία άλλου τύπου όπως κυρίως word ή excel για την άμεση εισαγωγή και ανανέωση της αποθήκης κρίνεται απαραίτητη. Δηλαδή θα είναι εφικτό να "ανεβάσεις" ένα αρχείο που θα έχει σχεδιασμένο πίνακα με τα δεδομένα της καρτέλας αποθήκη και να ανανεώνει την αποθήκη αυτόματα. Θα μπορούσε η εφαρμογή να σχεδιαστεί και για άλλες πλατφόρμες με διαφορετικά λειτουργικά συστήματα (IOS, Window Phone). Για την επαναπαρακολούθηση των σημείων που είναι πελάτες της εταιρίας θα μπορούσε να δημιουργηθεί ένα πεδίο στην βάση που θα ανανεώνεται κάθε φορά που ο πελάτης θα ξαναπερνάει από εκείνο το σημείο. Τέλος η εφαρμογή θα μπορούσε να δουλεύει και χωρίς σύνδεση στο διαδίκτυο. Δηλαδή να κρατάει τα δεδομένα σε τοπικό χώρο αποθήκευσης και κατά την επιστροφή του πωλητή στην έδρα του να συνδέεται με το ασύρματο δίκτυο και να στέλνει τα δεδομένα τότε. Έτσι θα μειώνονταν πολύ τα έξοδα αλλά δεν θα ήταν δυνατή η παρακολούθηση του πωλητή κατά την διάρκεια της εργασίας.

6.3 Συνεργασίες

Η παρούσα διπλωματική εργασία συνεργάζεται με την διπλωματική εργασία του κ. Αργυρίου Λιούτσια. Η μία μπορεί να συμπληρώσει την άλλη καθώς η παρούσα διπλωματική εργασία ελέγχει τους πωλητές ενώ η άλλη ελέγχει τους οδηγούς που διανέμουν τις παραγγελίες και τους πελάτες. Έτσι ο έλεγχος που θα υπάρχει από τον ιδιοκτήτη θα είναι σε μεγαλύτερο και ικανοποιητικότερο βαθμό. Οι ιστοσελίδες θα μπορούσαν να ενωθούν σε μία η οποία θα περιέχει όλα τα στοιχεία από τους πελάτες μέχρι τους πωλητές, τους οδηγούς και τα προϊόντα. Θα είναι δυνατός ο

εντοπισμός των οδηγών αλλά και η δημιουργία διαδρομής που θα ακολουθήσει ο πωλητής κατά την διάρκεια της ημέρας με τον αλγόριθμο δημιουργίας μονοπατιού της διπλωματικής του κ Λιούτσια.

Βιβλιογραφία

1. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 15 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Html>.
2. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 15 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Css>.
3. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 14 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Php>.
4. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 14 Ιούλιος 2013.]
<http://el.wikipedia.org/wiki/MySQL>.
5. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 16 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Javascript>.
6. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 15 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Xampp>.
7. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 15 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Freebsd>.
8. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 16 Ιούλιος 2013.]
<http://en.wikipedia.org/wiki/Notepad%2B%2B>.
9. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 15 Ιούλιος 2013.]
http://en.wikipedia.org/wiki/Google_Maps_API#Google_Maps_API.
10. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 14 Ιούλιος 2013.]
<http://el.wikipedia.org/wiki/Android>.
11. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 14 Ιούλιος 2013.]
[http://en.wikipedia.org/wiki/Eclipse_\(software\)](http://en.wikipedia.org/wiki/Eclipse_(software)).

12. wikipedia. [Ηλεκτρονικό] [Παραπομπή: 10 Σεπτέμβριος 2013.]
<http://el.wikipedia.org/wiki/Java>.
13. Webdesignergeeks. *Android Login with remote database*.
[Ηλεκτρονικό] <http://webdesignergeeks.com/mobile/android/android-login-authentication-with-remote-db/>.
14. **Michael Burton, Donn Felker**. *Android Application Development For Dummies*.
15. **Meier, Reto**. *Professional Android 4 Application Development*.
16. Stackoverflow. *General questions over programming*. [Ηλεκτρονικό]
<http://stackoverflow.com/>.
17. MyBringBack. *Android Development Tutorials*. [Ηλεκτρονικό]
<http://www.mybringback.com/tutorial-series/>.
18. Compus. *Html-Php-Css Courses by Minas Dasygenis*. [Ηλεκτρονικό]
<https://compus.uowm.gr/claroline/document/document.php>.
19. AndroidHive. *How to connect android with php mysql*.
[Ηλεκτρονικό] <http://www.androidhive.info/2012/05/how-to-connect-android-with-php-mysql/>.
20. Coderzheaven. *Android-phpmysql connection*. [Ηλεκτρονικό]
<http://www.coderzheaven.com/2011/04/08/android-phpmysql-connection/>.