



**Τμήμα Μηχανικών Πληροφορικής**  
**ΤΕΙ ΚΡΗΤΗΣ**

Πτυχιακή Εργασία

**Εφαρμογή κινητού Android για έλεγχο εσόδων και εξόδων**

Ιωάννης Μελαμπιανάκης 2971

2015



## Περίληψη

Στην εξής πτυχιακή ασχολούμαστε με εφαρμογή Android, η οποία ασχολείται με τα Έσοδα και Έξοδα του χρήστη. Η εφαρμογή απευθύνεται σε απλό χρήστη αλλά και σε επαγγελματία.

Ο χρήστης περνώντας μια εγγραφή στην εφαρμογή του μπορεί να σημειώσει τι κατηγορία και τύπος είναι, ποια η τιμή της συναλλαγής του και βέβαια αν το επιθυμεί να γράψει κάποια σημειώσει.

Επίσης μπορεί να παρακολουθεί το σύνολο των εσόδων και εξόδων του και το τελικό ισοζύγιο ή να διαλέξει από συγκεκριμένες ημερομηνίες να δει τα προηγούμενα αποτελέσματα.

### **Abstract**

In the following thesis we deal with application Android, which deals with the income and expenses of the user. The application is aimed at casual user but also professional.

The user passing a registration on application may note what category and type is, what the transaction price and of course if he wishes to write a note.

It can also monitor all the revenue and expenditure of the final balance or choose from specific dates to see previous results.

<b>1. Εισαγωγή</b> .....	9
1.1 Περίληψη.....	9
1.2 Κίνητρο για την διεξαγωγή της Εργασίας.....	9
1.3 Σκοπός και στόχοι της Εργασίας.....	9
<b>2. Ανάπτυξη εφαρμογών για κινητές συσκευές</b> .....	10
2.1 Κινητές συσκευές.....	10
2.1.1 Smartphone.....	10
2.1.2 Tablet.....	10
2.2 Είδη Λογισμικού.....	11
2.2.1 IOS.....	11
2.2.2 Android.....	12
2.2.3 Windows Phone.....	12
2.3 Μερίδιο Αγοράς.....	12
2.4 Είδη εφαρμογών για κινητές συσκευές.....	14
2.4.1 Native Εφαρμογές.....	14
2.4.2 Web Εφαρμογές.....	16
2.4.3 Υβριδικές Εφαρμογές.....	17
<b>3. Τεχνολογίες Υλοποίησης</b> .....	20
3.1 HTML5.....	20
3.2 Javascript.....	20
3.3 CSS.....	21
3.4 Jquery & Jquery Mobile.....	21
3.5 Sqlite.....	22
3.6 Bootstrap.....	22
<b>4. Κύριο μέρος Πτυχιακής</b> .....	23
4.1 Ανάλυση εφαρμογής.....	23
4.2 Σχεδιασμός εφαρμογής.....	23
4.3 Υλοποίηση.....	27
<b>5. Αποτελέσματα</b> .....	43
5.1 Συμπεράσματα.....	43

## Κατάλογος Εικόνων

Εικόνα 1: Smartphone.....	10
Εικόνα 2: Tablet Ipad.....	11
Εικόνα 3: Tablet Samsung.....	11
Εικόνα 4: IOS.....	12
Εικόνα 5: Android.....	12
Εικόνα 6: Windows Phone.....	12
Εικόνα 7: Native Εφαρμογές.....	15
Εικόνα 8: Web Εφαρμογές.....	16
Εικόνα 9: Υβριδικές Εφαρμογές.....	17
Εικόνα 10: Homepage.....	24
Εικόνα 11: Add new.....	25
Εικόνα 12: Balance.....	26
Εικόνα 13: About.....	27
Εικόνα 14: Κώδικας 1.....	28
Εικόνα 15: Κώδικας 2.....	28
Εικόνα 16: Κώδικας 3.....	29
Εικόνα 17: Button.....	29
Εικόνα 18: Homepage #2.....	29
Εικόνα 19: Add new #2.....	30
Εικόνα 20: Κώδικας 4.....	31
Εικόνα 21: Κώδικας 5.....	31
Εικόνα 22: Κώδικας 6.....	31
Εικόνα 23: Κώδικας 7.....	32
Εικόνα 24: Edit.....	32
Εικόνα 25: Κώδικας 8.....	32
Εικόνα 26: Κώδικας 9.....	33

Εικόνα 27: Όλες οι εγγραφές.....	33
Εικόνα 28: Έσοδα.....	34
Εικόνα 29: Έξοδα.....	34
Εικόνα 30: Αναζήτηση.....	34
Εικόνα 31: Ισοζύγιο.....	35
Εικόνα 32: Κώδικας 10.....	35
Εικόνα 33: Κώδικας 11.....	36
Εικόνα 34: Διαγραφεί .....	36
Εικόνα 35: Κώδικας 12.....	37
Εικόνα 36: Edit #2.....	38
Εικόνα 37: Edit #3.....	38
Εικόνα 38: Κώδικας 13.....	39
Εικόνα 39: Delete.....	39
Εικόνα 40: Confirm.....	40
Εικόνα 41: Delete #2.....	40
Εικόνα 42: Κώδικας 14.....	40
Εικόνα 43: Balance #2.....	41
Εικόνα 44: Balance #3.....	41
Εικόνα 45: Κώδικας 15.....	42

## **Κατάλογος Πινάκων**

Πίνακας 1: Χαρακτηριστικά Λογισμικών.....13-14

Πίνακας 2: Τεχνολογίες Εφαρμογών.....18-19



# 1. Εισαγωγή

## 1.1 Περίληψη

Ο σκοπός της πτυχιακής ήταν η μελέτη πάνω σε Υβριδική Εφαρμογή (Hybrid App), η οποία έχει την δυνατότητα να είναι συμβατή με το μεγαλύτερο μέρος των διαθέσιμων κινητών όπως IOS, Android, Windows Phone. Οι συγκεκριμένες εφαρμογές χρησιμοποιούνται όλο και πιο συχνά λόγω της δυνατότητας να χρησιμοποιείται σε πολλά λογισμικά. Όμως δεν μπορεί να καθοριστεί αν είναι καλύτερη μια Hybrid App ή μια Native App, επειδή αυτό εξαρτάται από την εκάστοτε εφαρμογή και το βαθμό δυσκολίας της. Έτσι κατέληξα στο συμπέρασμα ότι δεν μπορώ να πω με σιγουριά τα πλεονεκτήματα και μειονεκτήματα των Υβριδικών εφαρμογών.

Για την ολοκλήρωση της εφαρμογής χρησιμοποιήθηκε το IDE Intel Xdk, το οποίο έχει τα πλεονεκτήματα ότι είναι ελαφρύ σαν πρόγραμμα, έχει δικό του emulator κάτι που κάνει την δουλειά του προγραμματιστή πιο εύκολη και τέλος μπορεί να κάνει build την εφαρμογή πολύ γρήγορα και εύκολα. Επίσης, απαραίτητο τεχνολογικό συστατικό υλοποίησης της είναι η χρήση της τεχνολογίας PhoneGap η οποία επιτρέπει τη χρήση γλωσσών προγραμματισμού και σήμανσης (HTML5, CSS, Javascript, JQuery, JQuery Mobile, Bootstrap) καθώς και η χρήση βάσεις δεδομένων Sqlite μιας και είναι offline εφαρμογή.

Το θέμα της εφαρμογής είναι ένα πρόγραμμα «Εσόδων & Εξόδων» το οποίο θα κρατούσε σε μια βάση δεδομένων το ισοζύγιο (έσοδα, έξοδα) του χρήστη (απλού είτε επαγγελματία), ώστε να μπορεί να ελέγχει τα οικονομικά του και να τα βάζει σε μια τάξη. Επίσης θα μπορεί να βλέπει την ημερομηνία που έκανε την πίστωση ή χρέωση και βέβαια μπορεί να κρατεί και σημειώσεις σε περίπτωση που αν θα θέλει να ξανά ελέγξει την συγκεκριμένη κίνηση να μπορεί να θυμάται τι έκανε.

## 1.2 Κίνητρο για την Διεξαγωγή της Εργασίας

Ο λόγος που επέλεξα την συγκεκριμένη πτυχιακή εργασία είναι λόγω ότι η τεχνολογία αυτή είναι καινούργια και επειδή ήθελα να δω τις δυνατότητες μιας Υβριδικής εφαρμογής. Επίσης ο τομέας προγραμματισμού και συγκεκριμένα τα Smartphone App με ενδιαφέρει πάρα πολύ για τον μελλοντικό επαγγελματικό που προορισμό μου.

## 1.3 Σκοπός και Στόχοι Εργασίας

Σκοπός της πτυχιακής είναι να γνωρίσω την τεχνολογία προγραμματισμού μιας εφαρμογής κινητού που μια από τις πρώτες επιλογές όλων των μεγάλων και μικρών εταιρειών προγραμματισμού Site και εταιρειών κινητής τηλεφωνίας. Στόχος εργασίας μου είναι να μπορεί η εφαρμογή να εξυπηρετεί τον κάθε χρήστη ανεξαρτήτως ηλικίας και επαγγελματικής ιδιότητας.

## 2. Ανάπτυξη εφαρμογών για κινητές συσκευές

### 2.1 Κινητές συσκευές

Στο κεφάλαιο αυτό παρουσιάζονται τα είδη των συσκευών και τα πιο διαδεδομένα είδη λογισμικών που κυκλοφορούν στην αγορά αυτή την στιγμή.

#### 2.1.1 Smartphone

Ένα έξυπνο κινητό, όπως ορίζεται σήμερα διαθέτει λειτουργικό σύστημα που υποστηρίζει τον πολυνηματισμό, ασύρματη συνδεσιμότητα στο διαδίκτυο, υποστήριξη 3G/4G σύνδεσης, υποστήριξη εφαρμογών, δυνατότητα αναπαραγωγής πολυμέσων και πολλά άλλα. Τα πιο διαδεδομένα λογισμικά που υπάρχουν στα έξυπνα κινητά είναι: IOS, Android, Windows, Blackberry

- GPS
- Bluetooth
- Wi-fi
- Κάμερα
- Λειτουργίες Αφής



Εικόνα 1: Smartphone

#### 2.1.2 Tablet

Το tablet θεωρείται μια επίπεδη κινητή συσκευή, η οποία όμως είναι λίγο μεγαλύτερη από κινητό και λίγο μικρότερο από έναν φορητό υπολογιστή.

Τα μεγέθη ξεκινάνε από 5 ίντσες έως 13.3 ίντσες η οποία υποστηρίζει λειτουργία οθόνης αφής. Οι δυνατότητες που παρέχουν είναι σχεδόν ίδιες με αυτές που παρέχει ένα έξυπνο κινητό εκτός από την μεγάλη διαφορά τους ότι το Tablet δεν υποστηρίζει τηλεφωνικές κλήσεις και αποστολή μηνυμάτων,

βέβαια και αυτό αναπληρώνεται από διάφορες εφαρμογές που έχουν δημιουργηθεί όπως Viber, Whats App , Skype , Facebook .

Όπως και στα έξυπνα κινητά έτσι και στα Tablet τα πιο διαδεδομένα λογισμικά που χρησιμοποιούνται είναι: IOS, Android , Windows .



Εικόνα 2:Tablet Ipad



Εικόνα 3:Tablet Samsung

## 2.2. Είδη Λογισμικού

### 2.2.1 IOS

IOS είναι λογισμικό της Apple το οποίο κυκλοφόρησε για πρώτη φορά στις 29 Ιουνίου 2007 μαζί με το πρώτο iPhone. Αυτή την στιγμή υπάρχουν πάνω από 1.000.000 εφαρμογές στο App Store. Η μόνη διάφορα του IOS με τα υπόλοιπα λογισμικά είναι ότι η Apple δεν δίνει άδεια εγκατάστασης του λογισμικού της σε κινητό που δεν είναι IOS. [1]



Εικόνα 4: IOS

### 2.2.2 Android

Το Android είναι το λογισμικό που τρέχει στις περισσότερες έξυπνες συσκευές. Είναι το λειτουργικό το οποίο τρέχει με πυρήνα του λειτουργικού Linux. Οι συσκευές με Android λογισμικό είναι οι περισσότερες σε πώλησης ανά κόσμο.

Κυκλοφόρησε για πρώτη φορά στις 23 Σεπτεμβρίου 2008 . [2]



Εικόνα 5:Android

### 2.2.3 Windows Phone

Το Windows Phone είναι το λειτουργικό που εκδίδει η Microsoft , το οποίο τρέχει σε συσκευές Microsoft , Nokia , Huawei. Η πρώτη έκδοση βγήκε το 2000 και από τότε έχει αυξησει κατά πολύ τις πώλησης του. [3]



Εικόνα 6:Windows Phone

## 2.3 Μερίδιο Αγοράς

Βέβαια να αναπτύξεις μια εφαρμογή έχει δυσκολίες και μετά την υλοποίηση της γιατί θα πρέπει έχουμε υπολογίσει τον κατακερματισμό που εμφανίζεται στην αγορά των κινητών συσκευών.

Επίσης για την ανάπτυξη μιας εφαρμογής πρέπει να αποφασίσεις για ποιο λειτουργικό θα την χρησιμοποιήσεις ώστε να χρησιμοποιήσεις την κατάλληλη γλώσσα προγραμματισμού και τα συγκεκριμένα εργαλεία ανάπτυξης που παρέχονται από τον κατασκευαστή.

Ο παρακάτω πίνακας περιέχει τις διαφορές που παρουσιάζονται στις γλώσσες προγραμματισμού , τις καταλήξεις εκτελέσιμου αρχείου , το περιβάλλον ανάπτυξης και το Store της κάθε μιας.

Χαρακτηριστικά	iOS	Android	Windows Phone	Symbian	Blackberry/RIM
Λειτουργικό Σύστημα	iOS 1.x, iOS 2.x, iOS 3.x, iOS 4.x, iOS 5.x, iOS 6.x	1.5 Cupcake, 1.6 Donut, 2.0,2.0.1,2.1 Eclair, 2.2 – 2.2.3 Froyo, 2.3 – 2.3.7 Gingerbread, 3.0 – 3.2 Honeycomb, 4.0 – 4.0.4 Ice Cream Sandwich, 4.1 – 4.3 Jelly Bean	WinMo 2003, WinMo 5, WinMo 6.5, Windows Phone 7.x, Windows Phone 8	Symbian OS 6.0, Symbian OS 6.1, Symbian OS 7.0, Symbian OS 8.0, Symbian OS 9.0, Symbian EPOC32 (εκδόσεις 1 έως 5)	RIM 4.5, RIM 4.6, RIM 5.0, RIM 6.0, RIM 7.0, RIM 7.1, Blackberry 10

Γλώσσα Προγραμματισμού	Objective-C, Objective C++, C, C++	Java και XML	C#, Visual Basic	Java, C++	Java
Κατάληξη εκτελέσιμου αρχείου	.app	.apk	.xap	.pkg	.cod
Περιβάλλον ανάπτυξης	Xcode	Android SDK	Visual Studio, Windows Phone development tools	Symbian SDK	BB Java Eclipse Plug-in
Application Stores	Apple Store/App Store	Android Marketplace/Google Play	Windows Phone Marketplace	Nokia Store	BlackBerry App World

Πίνακας 1: Χαρακτηριστικά Λογισμικών

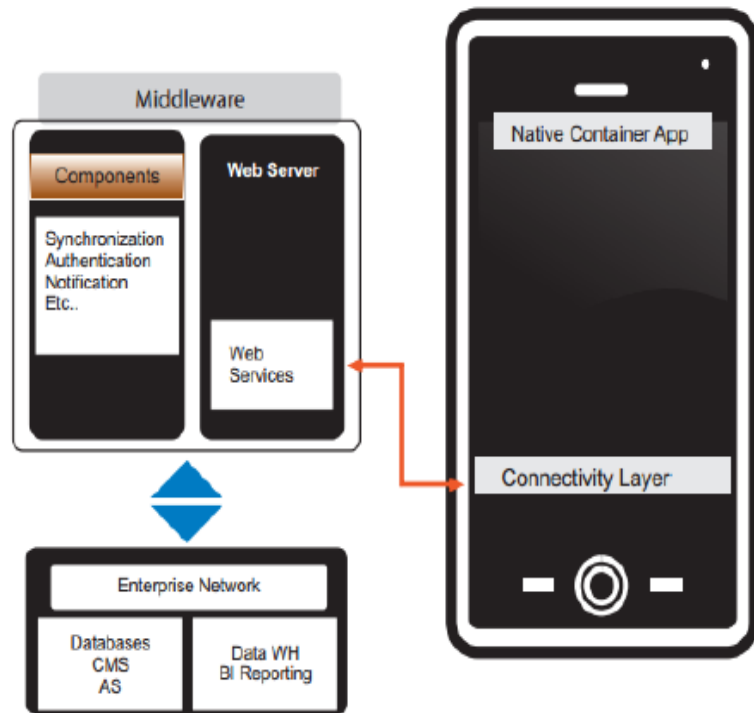
## 2.4 Είδη εφαρμογών για κινητές συσκευές

Πριν την υλοποίηση μιας εφαρμογής πρέπει να πάρουμε μια σημαντική απόφαση, ποια αρχιτεκτονική προσέγγιση θα χρησιμοποιούμε. Για να το κάνουμε αυτό πρέπει να υπολογίσουμε κάποιους παραμέτρους .

- ✓ Προϋπολογισμός
- ✓ Η τεχνογνωσία του προσωπικού που θα αναλάβει την ολοκλήρωση του
- ✓ Σε ποιούς πελάτες & χρήστες απευθύνεται η εφαρμογή
- ✓ Ποία κινητά θα υποστηρίζουν την εφαρμογή
- ✓ Χρονοδιάγραμμα

Ας δούμε αναλυτικά κάθε τεχνογνωσία:

### 2.4.1 Native εφαρμογές



Εικόνα 7: Native Εφαρμογές

Οι native εφαρμογές είναι εκτελέσιμα αρχεία τα οποία αποθηκεύονται στην συσκευή. Τις εφαρμογές αυτές μπορούμε να τις βρούμε και να τις κατεβάσουμε από τα αντίστοιχα store του κάθε λειτουργικού συστήματος όπως π.χ. Android market για τα android. Μόλις κατεβάσουμε την εφαρμογή και την εγκαταστήσουμε δημιουργείται ένα εικονίδιο στην επιφάνεια εργασίας. Μετά από αυτό μπορούμε να τρέξουμε την εφαρμογή μας. [4]

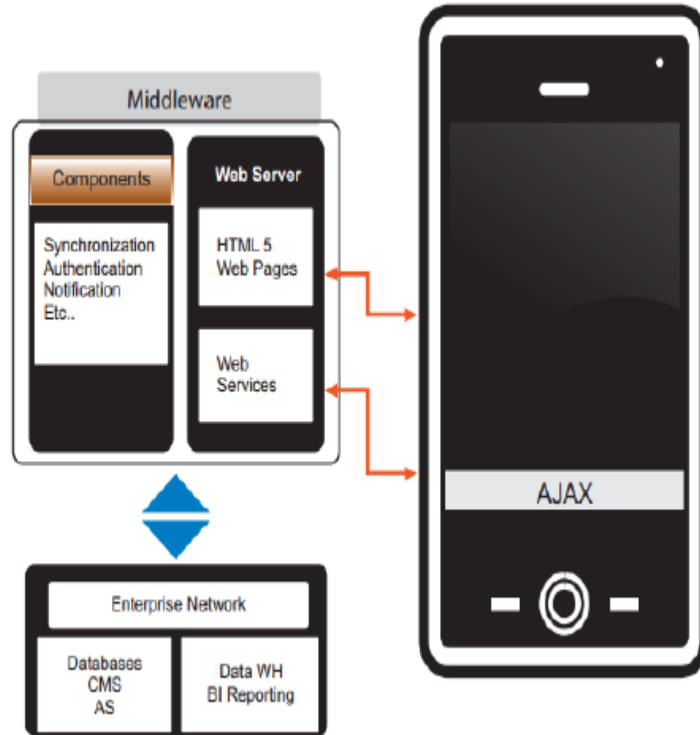
Για τη δημιουργία μιας native εφαρμογής ο προγραμματιστής πρέπει να γράψει τον κώδικα σε γλώσσα προγραμματισμού που υποστηρίζει το συγκεκριμένο λειτουργικό σύστημα της συσκευής. Η εταιρία του κάθε λειτουργικού συστήματος παρέχει συγκεκριμένα εργαλεία. Μια native εφαρμογή έχει το αρνητικό να μπορεί να χρησιμοποιείται μόνο σε ένα λειτουργικό σύστημα και συγκεκριμένα σε αυτό που έχει γραφτεί. Έτσι αν θέλουμε μια εφαρμογή να τρέχει σε περισσότερα από ένα λειτουργικό σύστημα θα πρέπει να την δημιουργήσουμε υβριδικά κάτι που θα το δούμε παρακάτω.

Το ερώτημα είναι γιατί συνεχίζουν και χρησιμοποιούνται native εφαρμογές ?

Ο πιο σημαντικός λόγος είναι ότι τις δυνατότητες που έχει μια Native εφαρμογή δεν τις έχει μια υβριδική εφαρμογή, μιας και η native γράφεται συγκεκριμένα για το λειτουργικό σύστημα αυτό.

Επίσης γραφικά , μπάρες , κουμπιά κλπ. Είναι συγκεκριμένα σε κάθε λειτουργικό σύστημα και αυτά πρέπει να γίνουν σε native.

## 2.4.2 Web εφαρμογές



Εικόνα 8: Web Εφαρμογές

Οι web εφαρμογές δίνουν την λύση στο πρόβλημα που δημιουργείται από τις natives εφαρμογές και έτσι χρησιμοποιώντας HTML5, CSS, Javascript δημιουργήθηκαν εφαρμογές όπως gmail οι οποίες λειτουργούν με τεχνολογία διαδικτύου. Έτσι αποφεύγουν οι προγραμματιστές χρόνο και χρήμα για να μάθουν νέες γλώσσες προγραμματισμού. Οι εφαρμογές αυτές θα φιλοξενοούνται στον εξυπηρετητή και θα μπορούν να εκτελεστούν από τον χρήστη με την επίσκεψη στο συγκεκριμένο URL, έτσι αποφεύγουμε τα ηλεκτρονικά καταστήματα του κάθε λειτουργικού συστήματος. [4]

Πρέπει να αναφερθεί ότι σε αντίθεση με τις native εφαρμογές οι διαδικτυακές εκτελούνται στον browser και όχι στο περιβάλλον του λειτουργικού συστήματος. Ενώ οι native εφαρμογές έχουν πλήρη πρόσβαση στη συσκευή , οι web εφαρμογές έχουν περιορισμένη πρόσβαση σε πολλά χαρακτηριστικά της συσκευής και κάποιες φορές έχουν πλήρη απουσία πρόσβασης. Η απουσία πρόσβασης οφείλεται σε δυο λόγους.

- Η ασφάλεια της συσκευής :

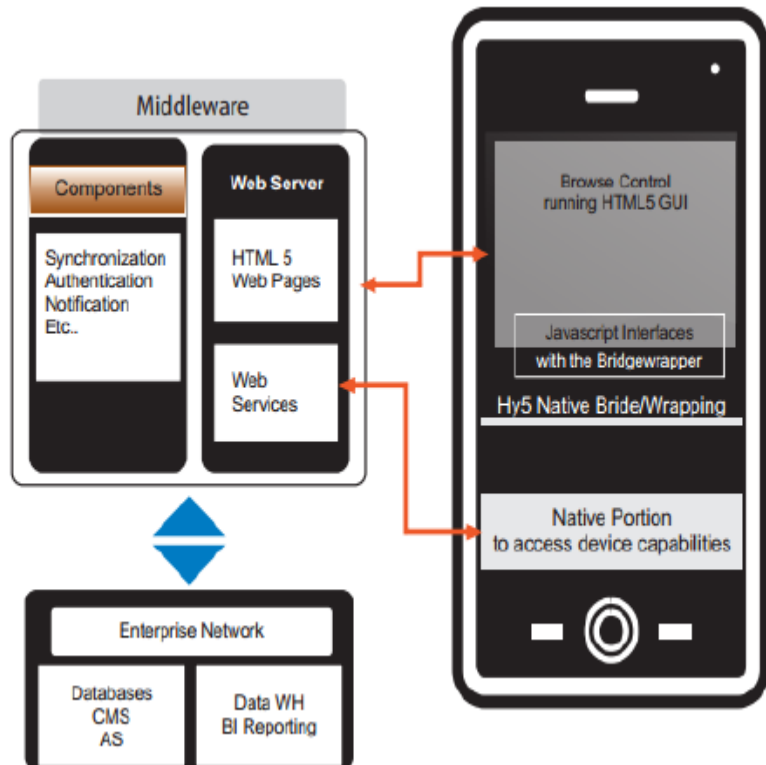


Οι native εφαρμογές είναι διαθέσιμες στα market κάτι το οποίο εξασφαλίζει την λειτουργία της εφαρμογής και αν έχει τα κατάλληλα χαρακτηριστικά ώστε να μην είναι επιβλαβή για την συσκευή.

Στις web εφαρμογές αυτό απουσιάζει.

- Η απουσία υλοποίησης διάφορων προγραμματιστικών διεπαφών, έχοντας ως αποτέλεσμα την απουσία πρόσβασης σε διάφορες λειτουργίες της συσκευής.

### 2.4.3 Υβριδικές εφαρμογές



Εικόνα 9: Υβριδικές Εφαρμογές

Κάπου εδώ έρχονται οι υβριδικές εφαρμογές οι οποίες ικανοποιούν και συνδυάζουν τα χαρακτηριστικά των native και web εφαρμογών. Με αυτό τον τρόπο οι προγραμματιστές μπορούν να εκμεταλλευτούν τις κύριες λειτουργίες των web εφαρμογών και την πρόσβαση σε χαρακτηριστικά της συσκευής που είναι διαθέσιμες μόνο για native. [4]

Το μόνο που μένει είναι να χρησιμοποιηθεί κάποια γέφυρα τύπου ανοιχτό κώδικα, ανάμεσα στις δυο τεχνολογίες ώστε να δημιουργηθεί μια υβριδική εφαρμογή. Μια τέτοια είδος

τεχνολογία είναι το PhoneGap, το οποίο μέσω Javascript επιτρέπει να κληθούν API.

Με δυο τρόπους μπορούμε να αναπαραστήσουμε μια υβριδική εφαρμογή ,είτε με σύνολο αρχείων HTML ,CSS ,Javascript είτε με web σελίδα.

Εδώ είναι ένα πινακάκι που συγκρίνει τις 3 τεχνολογίες που αναφέραμε και μια ακόμα που δεν την έχω αναφέρει.

Χαρακτηριστικά	Native	Mobile Web	Hybrid	Cross-Compiled
Εργαλεία/δεξιότητες που απαιτούνται για ανάπτυξη εφαρμογών σε διάφορες πλατφόρμες	Objective-C, Java, C, C++, C#, VB.net etc	HTML, CSS, Javascript	HTML, CSS, Javascript, Mobile development framework (όπως το PhoneGap)	Javascript, Ruby, C#, Java, C, C++, Objective-C
Διανομή	App Store/Market	Internet	App Store/Market	App Store/Market
Ταχύτητα υλοποίησης	Αργό	Γρήγορο	Ικανοποιητικό	Ικανοποιητικό
Συντήρηση (Updates)	Δύσκολη	Χαμηλή	Ικανοποιητική	Ικανοποιητική

Προσβασιμότητα στα χαρακτηριστικά της συσκευής	Πλήρης πρόσβαση: (Camera, microphone, GPS, gyroscope, accelerometer, file upload, etc...)	Μερική πρόσβαση: (GPS, gyroscope, accelerometer, file upload.)	Πλήρης πρόσβαση: (Camera, microphone, GPS, gyroscope, accelerometer, file upload, etc...)	Πλήρης πρόσβαση: (Camera, microphone, GPS, gyroscope, accelerometer, file upload, etc...)
Λειτουργία offline	ΝΑΙ	ΝΑΙ (app-cache) / ΟΧΙ εάν υλοποιηθεί ως mobile web site	ΝΑΙ	ΝΑΙ
Πλεονεκτήματα	Επιτρέπει την δημιουργία εφαρμογών με πλούσια ή/και απαιτητικά γραφικά.	Προσφέρει γρήγορη υλοποίηση, εύκολη συντήρηση και υποστήριξη του ίδιου κώδικα από μεγάλο αριθμό συσκευών και λειτουργικών συστημάτων.	Συνδυάζει την ταχύτητα υλοποίησης των web εφαρμογών με την πρόσβαση σε χαρακτηριστικά της συσκευής και τον τρόπο διανομής των native εφαρμογών.	Γρήγορη υλοποίηση με πρόσβαση σε χαρακτηριστικά της συσκευής. Επαναχρησιμοποίηση κώδικα που αφορά την λειτουργικότητα της εφαρμογής
Μειονεκτήματα	Χρόνος υλοποίησης, Κόστος υλοποίησης, Συντήρηση και αναβάθμιση, Υποστήριξη από ένα λειτουργικό σύστημα.	Δε μπορεί να διαχειριστεί απαιτητικά γραφικά. Δεν παρέχεται πρόσβαση σε όλα τα χαρακτηριστικά της συσκευής.	Δε μπορεί να διαχειριστεί απαιτητικά γραφικά, Απαιτεί την γνώση κάποιου mobile framework.	Στοχεύει σε συγκεκριμένα λειτουργικά συστήματα. Απαιτεί τη γνώση κάποιου framework. Περιορισμένος αριθμός διαθέσιμων frameworks. Εξάρτηση από το framework που επιλέχθηκε.

Ιδανικές κατηγορίες	Παιχνίδια, πελατοκεντρικές εφαρμογές με πλούσιο γραφικό περιβάλλον χρήστη. Εφαρμογές απαιτητικές σε μνήμη και ταχύτητα.	Εφαρμογές που παρέχουν κυρίως πληροφόρηση, εφαρμογές που δεν απαιτούν μεγάλη επεξεργαστική ισχύ και πρόσβαση σε όλα τα τεχνικά χαρακτηριστικά της συσκευής.	Πελατοκεντρικές εφαρμογές με ικανοποιητικό γραφικό περιβάλλον χρήστη. Εφαρμογές που απαιτούν πρόσβαση στα χαρακτηριστικά της συσκευής.	Ομοίως με native
---------------------	---	---	--	------------------

Πινάκκι 2: Τεχνολογίες εφαρμογών

## 3. Τεχνολογίες υλοποίησης

### 3.1 HTML5

Η HTML αποτελεί την κύρια γλώσσα σήμανσης για την ανάπτυξη διαδικτυακών εφαρμογών και ιστοσελίδων. Η HTML γράφεται υπό την μορφή στοιχείων τα οποία αποτελούνται από ετικέτες οι οποίες περικλείονται μέσα σε σύμβολα λειτουργώντας ανά ζεύγη <html> </html>. Ο σκοπός ενός browser είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει. Όταν ένας διαδικτυακός εξυπηρετητής ανοίγει ένα αρχείο HTML τα στοιχεία μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας. [5]

Η HTML5 αποτελεί ένα νέο πρότυπο ιστοσελίδων, προσφέρει νέες λειτουργίες για τον ορισμό αυτό που θα προσφέρει στο Διαδίκτυο, εφαρμογές πιο ισχυρές και πιο ευέλικτες. Συγκεκριμένα επιτρέπει στους browsers να υποστηρίζουν την αποθήκευση δεδομένων στην client μεριά. Η προσωρινή αποθήκευση των δεδομένων παίζει μεγάλη σημασία στις υβριδικές εφαρμογές. Έκτος από τα cookies και τις βάσεις δεδομένων στην μεριά του server έρχεται μια νέα τεχνολογία να αλλάξει τα δεδομένα, η οποία είναι διαθέσιμη για τους browsers παρέχοντας τοπική αποθήκευση. Γνωστές ως WebSql Database. Μερικές από αυτές είναι οι εξής :

- Localstorage
- Indexedb
- Web storage

### 3.2 Javascript

Η Javascript δημιουργήθηκε το 1995, το αρχικό της όνομα ήταν Mocha και ύστερα έγινε LiveScript και τέλος Javascript. Η Javascript είναι γλώσσα σεναρίων και όχι σήμανσης όπως η HTML. Η δουλειά της Javascript είναι η επικοινωνία με τον πελάτη (client-side) , αν και τα τελευταία χρόνια έχουν δημιουργηθεί μοντέλα τύπου node.js που κάνουν και την server-side. [6]

Η Javascript έρχεται να στηρίξει την δουλειά της HTML με τα script της, όπως δημιουργώντας εφέ, κάνοντας υπολογισμούς, περνώντας από τις φόρμες στοιχεία στην βάση μας. Ειδικότερα στις διαδικτυακές εφαρμογές είναι από τα πιο σημαντικά κομμάτια στην υλοποίηση τους, λόγω ότι διαχειρίζεται όπως είπαμε και πιο πάνω την client μεριά και επειδή σε κάποιες εφαρμογές που χρησιμοποιούνται Web-storage database (localstorage, indexedb, session storage, κλπ) διαχειρίζεται και την server – side

μεριά ουσιαστικά και αυτό είναι ένας από τους πολλούς λόγους που την κάνει σημαντική.

Ωστόσο υπάρχουν και περιορισμοί. Για λόγους ασφαλείας δεν παρέχει τη δυνατότητα προσπέλασης χειρισμού αρχείων του υπολογιστή του χρήστη. Επίσης δεν υποστηρίζει δυνατότητες δικτύωσης οποιασδήποτε μορφής ούτε διαθέτει εντολές δημιουργίας γραφικών.

### 3.3 CSS

Η CSS αποτελεί μια γλώσσα φύλλων στυλ και χρησιμοποιείται για τον έλεγχο της εμφάνισης μιας διαδικτυακής εφαρμογής. Οι αλλαγές μπορούν να γίνουν κατευθείαν στο έγγραφο HTML χρησιμοποιώντας το tag `<style>` είτε φτιάχνοντας ένα αρχείο CSS και περνώντας κάποια ετικέτα σε αυτό που θέλουμε να αλλάξουμε είτε class είτε id μπορούμε να κάνουμε την αλλαγή μας, βέβαια μπορούμε να χρησιμοποιήσουμε και τους δυο ταυτόχρονα. [7]

Το κύριο πλεονέκτημα της CSS είναι το ότι μια μορφοποίηση μπορεί να εφαρμοστεί σε ένα σύνολο ιστοσελίδων. Αν απαιτούνται αλλαγές στη μορφοποίηση τότε με μια επεξεργασία του CSS εγγράφου όλες οι σελίδες του ισότοπου θα λάβουν την ίδια αλλαγή. Η τελευταία έκδοση είναι η CSS3.

### 3.4 JQuery & JQuery Mobile

Στις περισσότερες γλώσσες ανοιχτού κώδικα υπάρχουν βιβλιοθήκες ώστε να κάνουν την ζωή του προγραμματιστή πιο εύκολη και να γλιτώνει πολλές γραμμές κώδικα. Το JQuery είναι μια βιβλιοθήκη της Javascript η οποία μας δίνει την δυνατότητα να δημιουργούμε κάποια εφέ στη σελίδα μας με πολύ λιγότερο κώδικα σε αντίθεση αν χρησιμοποιούσαμε σκέτη Javascript.

Για να χρησιμοποιήσουμε την JQuery πρέπει να κατεβάσουμε το αρχείο JQuery.js και να το βάλουμε στο tag `<script>` μέσα.

Κάποια από τα πράγματα που μπορεί να κάνει το framework αυτό είναι να περάσει δεδομένα από την φόρμα μου στον πίνακα μου, να εκτελέσει ερωτήματα AJAX έως και τα πιο απλά πράγματα όπως να αλλάξει μέγεθος στην γραμματοσειρά μας. [8] [9]

Η σύνταξη είναι :

**`$`(επιλογή).δράση()**

Η βιβλιοθήκη JQuery δημιούργησε μια ειδική έκδοση για την εφαρμογή του σε ανάπτυξη εφαρμογών κινητών τηλεφώνων, την JQuery Mobile. Το πλεονέκτημα που δίνει στον προγραμματιστή το JQuery mobile είναι ότι δημιουργεί γραφικά τα οποία παραμετροποιούν την γραφική υπόσταση της εφαρμογής ώστε να

προσαρμόζεται στην οθόνη του κινητού ή tablet. Τα πλεονεκτήματα του JQuery mobile είναι τα εξής:

1. Είναι δωρεάν
2. Μαθαίνεται εύκολα
3. Υποστηρίζει το μεγαλύτερο αριθμό συσκευών
4. Μπορεί να συνδυαστεί εύκολα με το σύστημα Phonegap για την δημιουργία υβριδικών εφαρμογών

### **3.5 Sqlite**

Η Sqlite είναι μια βάση δεδομένων που σε αντίθεση με την mysql είναι στην μεριά του πελάτη και χρησιμοποιείται και offline. Χρησιμοποιεί τα ίδια query που χρησιμοποιεί και η mysql, και γενικά χρησιμοποιείται σε διαδικτυακές εφαρμογές κινητών λόγω της ευκολίας χρήσης και αποθήκευσης και του μικρού χώρου που καταλαμβάνει στην εφαρμογή. [10]

### **3.6 Bootstrap**

Το bootstrap είναι ένα framework ανοιχτού κώδικα για την δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Χρησιμοποιείται όλο και πιο συχνά πια λόγω των πολύ όμορφων γραφικών που προσφέρει και της πολύ εύκολης χρήση του. Έχει .css και .js αρχείο. Να μην ξεχάσω να αναφέρω το πολύ καλό documentation που προσφέρει το Site του. [11]

## 4 . Κύριο Μέρος Πτυχιακής

### 4.1 Ανάλυση εφαρμογής

Ο σκοπός της πτυχιακής είναι να δημιουργηθεί μια εφαρμογή που θα καλύπτει τις ανάγκες των χρηστών στα έσοδα και έξοδα τους.

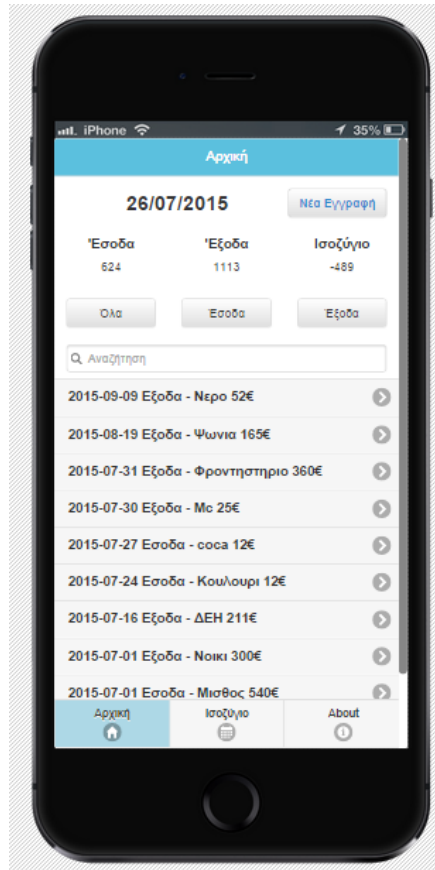
Αρχικώς μου σκοπός ήταν να μπορεί να το κατεβάσουν και να το χρησιμοποιήσουν χρήστες όλων των λογισμικών ή τουλάχιστον των περισσότερων. Δεύτερος μου στόχος ήταν να είναι μια «ελαφριά» εφαρμογή την οποία θα μπορούσε ο καθένας να περάσει στο κινητό του και να την χρησιμοποιήσει. Τέλος στόχος μου ήταν να μπορεί ο χρήστης να την χρησιμοποιεί και χωρίς internet, δηλαδή offline.

### 4.2 Σχεδιασμός εφαρμογής

Το πρώτο πράγμα που έκανα ήταν να σχεδιάσω τον πίνακα της βάσης μου. Το όνομα της βάσης μου είναι money.db. Στην εφαρμογή μου χρειάστηκα έναν πίνακα μονό τον οποίο ονόμασα και form\_charge. Τα πεδία του πίνακα μου είναι τα εξής:

- Int id (primary key)
- Text category
- Text type
- Text date
- Real charge
- Text notes

Επόμενο βήμα ήταν να σχεδιάσω στο χαρτί τις «οθόνες» της εφαρμογής μου. Να σημειώσω εδώ ότι jquery mobile σου δίνει την δυνατότητα να χρησιμοποιείς ένα αρχείο html με πολλές σελίδες ενσωματωμένες. Επιστρέφω στην εφαρμογή μου, το αρχείο μου το έχω ονομάσει index.html. Η πρώτη σελίδα που εμφανίζεται μόλις ανοίξω την εφαρμογή μου είναι η homepage



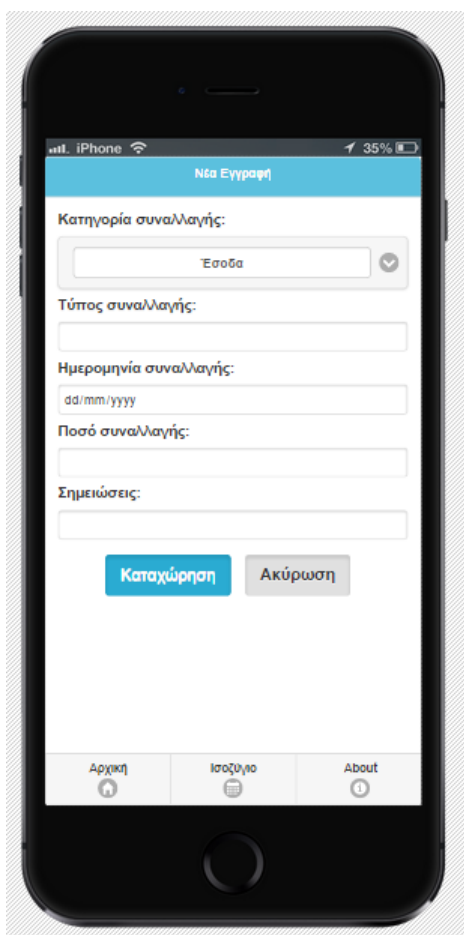
Εικόνα 10:Homepage

Η οποία εμφανίζει όλες τις εγγραφές που έχει κάνει ο χρήστης, δίνοντας την ευκαιρία στον χρήστη να του δείχνει μόνο έσοδα ή έξοδα.

Επίσης έχει πάνω δεξιά το κουμπάκι για νέα έγγραφη, το οποίο μας πάει σε άλλη σελίδα την οποία θα δούμε πιο μετά. Στο footer μας έχουμε τρία κουμπιά-λινκ τα οποία μας πάνε στις σελίδες homepage, balance και about. Τελειώνοντας έχω πρόσθεση με jquery πάνω script το οποίο να μας δείχνει την τωρινή ημερομηνία πάντα.

Επόμενη οθόνη που σχεδίασα ήταν η add\_new

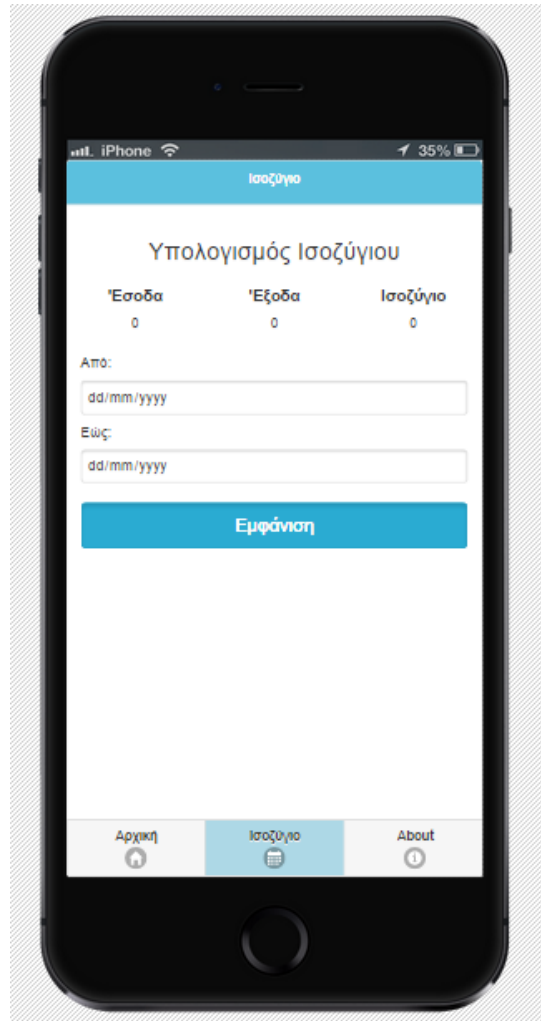




Εικόνα 11: Add new

Στην οποία είναι η φόρμα μου που καταχωρώ τις τιμές μου στον πίνακα μου τον form\_charge.

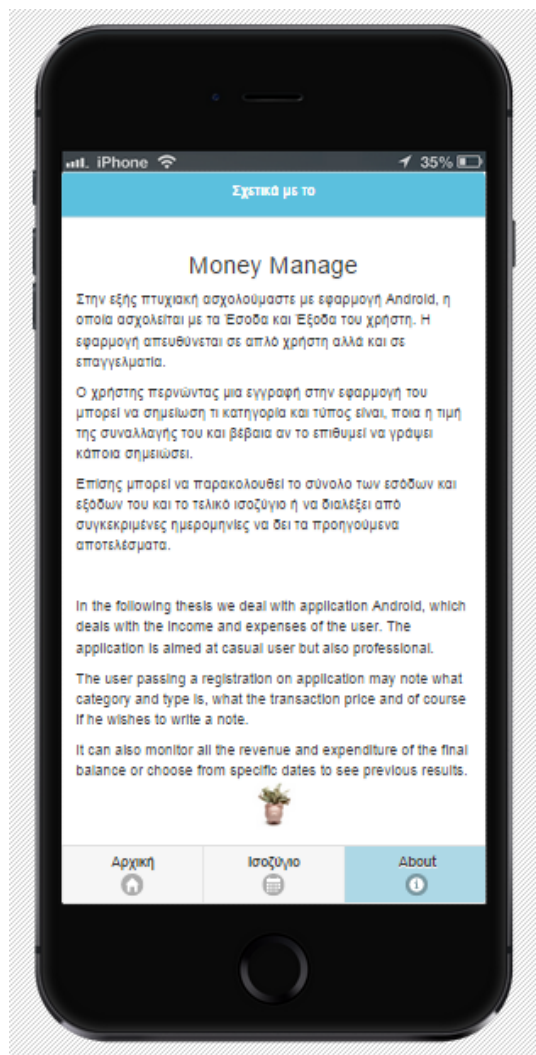
Επομένη οθόνη είναι η balance



Εικόνα 12: Balance

Στην οποία βάζουμε δυο ημερομηνίες στα textbox μας και μας εμφανίζει σε αυτό το διάστημα τις εγγραφές και επίσης μας υπολογίζει τα συνολικά έσοδα, έξοδα και το ισοζύγιο μας.

Τέλος έβαλα την about η οποία έχει μια περιγραφή της εφαρμογής στα ελληνικά και αγγλικά για τους χρηστές.



Εικόνα 13: About

### 4.3 Υλοποίηση

Από την στιγμή που είχα σχεδιάσει στο χαρτί τις «οθόνες» μου και είχα αποφασίσει τα παιδιά του πίνακα μου ξεκίνησα να γράφω κώδικα στο Intel Xdk.

Σαν αρχή πέρασα τα tag των .js και .css (Jquery-mobile.css, style.css, bootstrap.css, Jquery.js, Jquery-mobile.js) ώστε να μπορώ να χρησιμοποιήσω τα framework μου.

Αμέσως μετά άνοιξα ένα tag <script> και ξεκίνησα να φτιάχνω την βάση μου σε sqlite. Προτίμησα τα queries μου να τα περνάω σε μεταβλητές ώστε να μην έχω τεράστιο κώδικα και να αποφύγω τυχόν λάθος.

```

var createStatement = "CREATE TABLE IF NOT EXISTS form_charge (id INTEGER PRIMARY KEY AUTOINCREMENT, category TEXT NOT NULL, type TEXT NOT NULL, date TEXT NOT NULL, charge REAL NOT NULL, notes TEXT)";
var selectAllStatement = "SELECT * FROM form_charge ORDER BY date DESC";
var insertStatement = "INSERT INTO form_charge (category, type, date, charge, notes) VALUES (?, ?, ?, ?, ?)";
var selectDate = "SELECT * FROM form_charge WHERE date BETWEEN ? AND ?";
var selectAllDate = "SELECT * FROM form_charge";
var updateStatement = "UPDATE form_charge SET type = ?, date = ?, charge = ?, notes = ? WHERE id=?";
var deleteStatement = "DELETE FROM form_charge WHERE id=?";

```

Εικόνα 14: Κώδικας 1

Τι κάνει το καθένα θα το δούμε σιγά σιγά στην πορεία.

Για να ανοίξει η βάση δεδομένων της sqlite πρέπει να προσθέσω τον εξής κώδικα

```
var db = openDatabase("money.db", "1.0", "Money Manage", 200000);
```

Το επόμενο βήμα είναι να ελέγξω την βάση μου να υποστηρίζεται από τον browser ή την συσκευή μου και αν ναι να μου δημιουργήσει τον πίνακα μου.

```

function initDatabase() // Function Call When Page is ready.
{
    try {
        if (!window.openDatabase) // Check browser is supported SQLite or not.
        {
            alert('Databases are not supported in this browser.');
```

Εικόνα 15: Κώδικας 2

Επειδή όμως αυτό θέλω να γίνεται κάθε φορά που ανοίγω την εφαρμογή μου, βάζω την συγκεκριμένη function στο window.onload, του οποίου μας δίνει την δυνατότητα να εκτελούμε script με το άνοιγμα της σελίδας ή της εφαρμογής μας.

Αρα αν έχουνε πάει όλα σωστά θα έχει δημιουργηθεί ο πίνακας μου form\_charge.

Το αμέσως επόμενο πράγμα που έκανα ήταν να γράψω στην html την homepage σελίδα μου. Αποφάσισα να κάνω το footer και το header μου ίδια σε όλες τις σελίδες μου ώστε να είναι ομοιόμορφο. Στο σημείο αυτό να αναφέρω ότι στο footer μου δηλαδή το κάτω μέρος της εφαρμογής μου πέρασα τα link των σελίδων μου σε μορφή κουμπιών ώστε να μπορώ να εναλλάσσω τις σελίδες μου χωρίς πρόβλημα και καθυστέρηση.

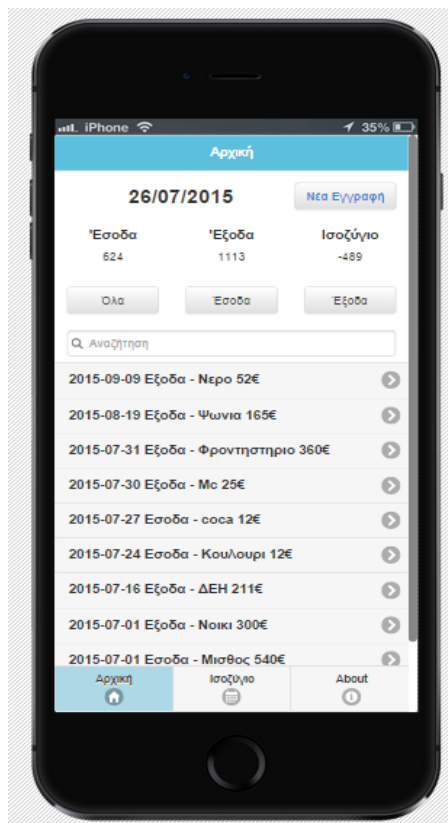
```
<div data-role="footer" data-position="fixed">
  <div data-role="navbar" data-iconpos="bottom">
    <ul>
      <li><a href="#homepage" id='tab1' data-icon="home" style="background-color:lightblue">Αρχική</a></li>
      <li><a href="#balance" id='tab2' data-icon="calendar">Ισοζύγιο</a></li>
      <li><a href="#about" id='tab3' data-icon="info">About</a></li>
    </ul>
  </div>
</div>
```

Εικόνα 16: Κώδικας 3



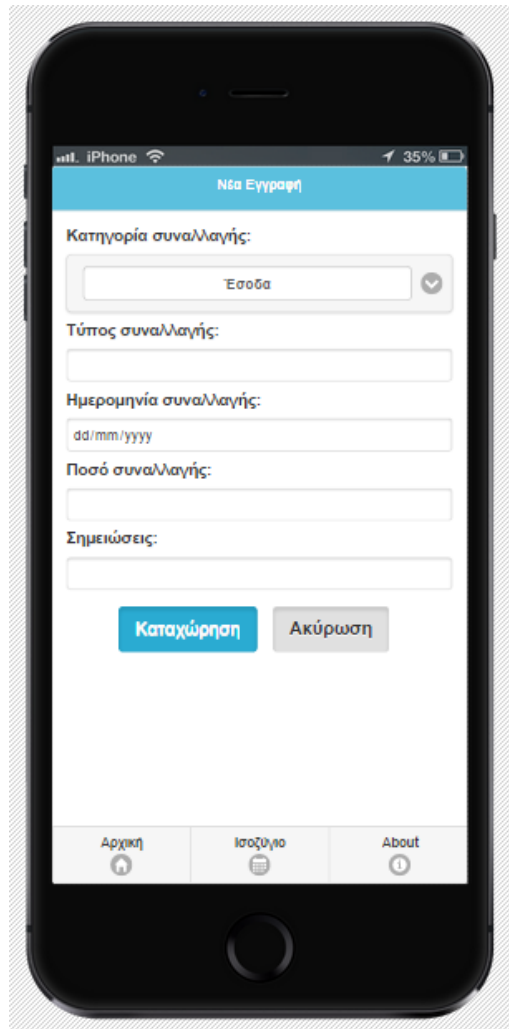
Εικόνα 17: Button

Ας ξεκινήσουμε με την πρώτη σελίδα της εφαρμογής μου.



Εικόνα 18: Homepage #2

Κάτω από το header και στα δεξιά όπως βλέπουμε από την πάνω εικόνα έχω βάλει ένα κουμπί το οποίο γράφει στα περιεχόμενα του “Νέα Εγγραφή”. Πατώντας το θα μας οδηγήσει στην σελίδα add\_form,



Εικόνα 19: Add new #2

Στην οποία έχω βάλει πέντε (5) πεδία input τα οποία είναι τα πεδία του πίνακα μου form\_charge.

Το input «Κατηγορία συναλλαγής» το έχω κάνει “option”, δηλαδή να μπορείς να επιλέγεις μεταξύ συγκεκριμένων τιμών που έχει επιλέξει να προσθέσει ο προγραμματιστής. Στην δικιά μου περίπτωση έβαλα μόνο δυο

- Έσοδα
- Έξοδα

Επίσης το input «Ημερομηνία συναλλαγής» το έχω κάνει date, είναι μια δυνατότητα που προσφέρει η HTML5.

Άλλη μια σημαντική λεπτομέρεια που πρέπει να αναφέρω είναι ότι για το input «Ποσό συναλλαγής» έφτιαξα μια function η οποία δεν επιτρέπει στο χρήστη να πατάει στο πεδίο αυτό εκτός από νούμερα και την υποδιαστολή τίποτα άλλο.

```
<script>
    //allow only numbers, no characters
    function validate(evt) {
        var theEvent = evt || window.event;
        var key = theEvent.keyCode || theEvent.which;
        key = String.fromCharCode( key );
        var regex = /[0-9]|\./;
        if( !regex.test(key) ) {
            theEvent.returnValue = false;
            if(theEvent.preventDefault) theEvent.preventDefault();
        }
    }
</script>
```

Εικόνα 20: Κώδικας 4

Στο κάτω μέρος έχω βάλει το κουμπί μου «Καταχώρηση», το οποίο αποθηκεύει τα δεδομένα μου στην βάση μου. Τέλος έχω προσθέσει και το κουμπί «Ακύρωση» το οποίο πατώντας το θα με γυρίσει στην homepage σελίδα μου.

Βέβαια για να περαστούν τα δεδομένα μου στην βάση μου χρειάζομαι JQuery κώδικα.

Το πρώτο που πράγμα που έκανα ήταν να φτιάξω την function insertRecord(),

```
function insertRecord()
{
    var usercategorytemp = $('#category').val();
    var usertypetemp = $('#type').val();
    var userdatetemp = $('#date').val();
    var userchargetemp = $('#charge').val();
    var usernotestemp = $('#notes').val();

    db.transaction(function (tx) { tx.executeSql(insertStatement, [usercategorytemp, usertypetemp, userdatetemp, userchargetemp, usernotestemp], loadAndReset, onError); });
}
```

Εικόνα 21: Κώδικας 5

Η εντολή \$('#id).val();

Παίρνει την τιμή από το πεδίο με το id που θα δώσουμε. Εγώ το παίρνω εδώ και το περνάω σε μια νέα μεταβλητή και μετά κάνω executeSql το querye “insertStatement”

```
var insertStatement = "INSERT INTO form_charge (category, type, date, charge, notes) VALUES (?, ?, ?, ?, ?)";
```

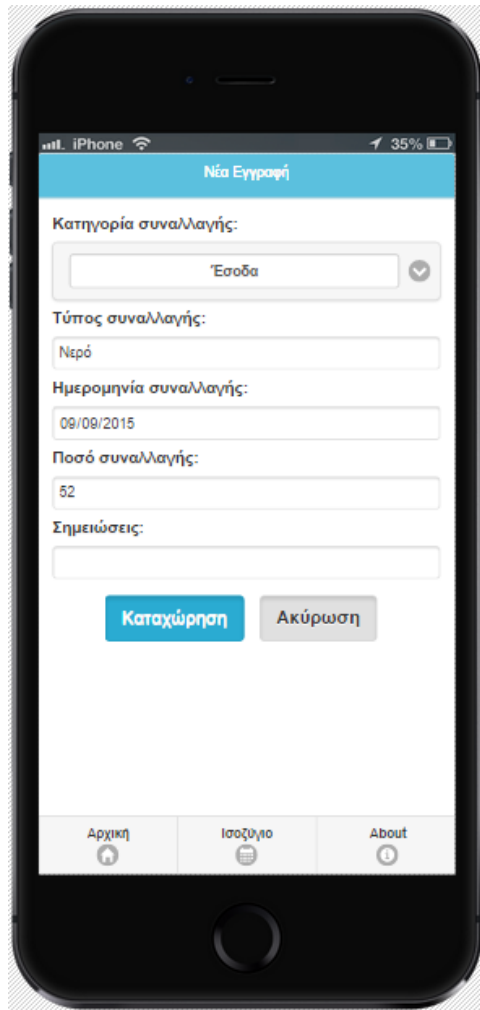
Εικόνα 22: Κώδικας 6

Για να ολοκληρωθεί η διαδικασία πρέπει να πατήσουμε το κουμπί “submitButton”

```
$("#submitButton").click(insertRecord);
```

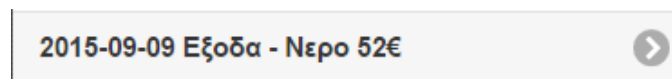
Εικόνα 23: Κώδικας 7

Παρακάτω θα κάνω ένα παράδειγμα,



Εικόνα 24: Edit

Και μόλις πατήσω «Καταχώρηση»



Εικόνα 25: Κώδικας 8



Εμφανίζεται η εξής εγγραφή στην homepage σελίδα μου.

Ας γυρίσουμε στην homepage σελίδα μου, ώστε να συνεχίσουμε εκεί που είχαμε σταματήσει.

Δίπλα από το κουμπί «Νέα Εγγραφή» έχω βάλει ένα <div> στο οποίο περνάω μέσω μιας function την τωρινή ημερομηνία.

```
var myDate = new Date(Date.now());

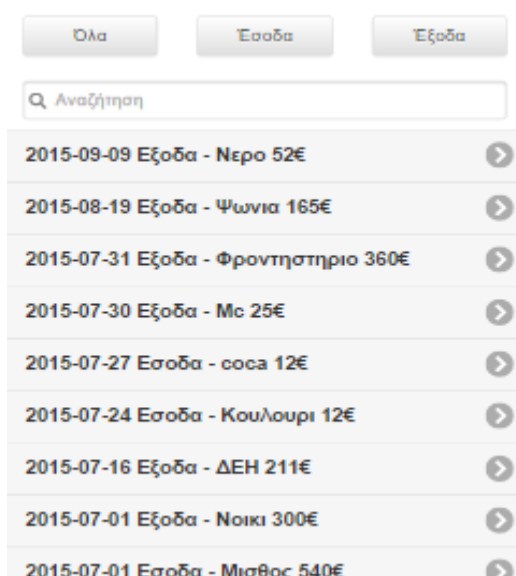
var year = myDate.getFullYear();
var month = myDate.getMonth() + 1;
if(month<10)
{
    month = "0" + month;
}

var date = myDate.getDate();
if(date<10)
{
    date = "0" + date;
}

var formattedDate = date + "/" + month + "/" + year;
document.getElementById("date_result").innerHTML = formattedDate;
```

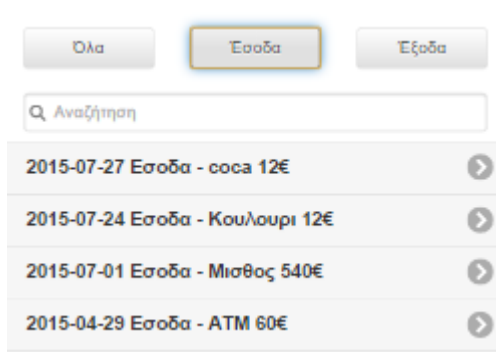
Εικόνα 26: Κώδικας 9

Μελετώντας αντίστοιχες εφαρμογές με την δικιά μου, θεώρησα πολύ σημαντικό ο χρήστης να μπορεί να παρακολουθεί τις εγγραφές του και να μπορεί να τις βλέπει κατηγοριοποιημένα αν το επιθυμεί.

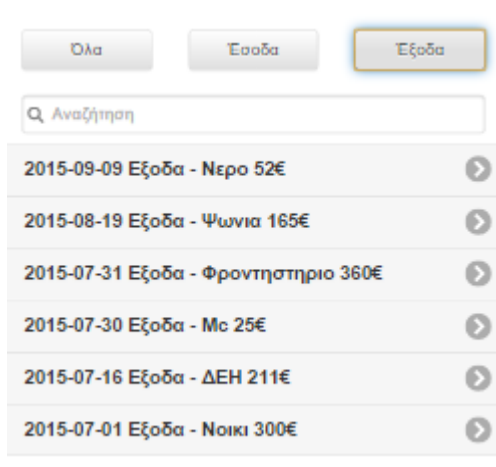


Όλα	Εσοδα	Εξοδα
Αναζήτηση		
2015-09-09	Εξοδα - Νερο	52€
2015-08-19	Εξοδα - Ψωνια	165€
2015-07-31	Εξοδα - Φροντιστήριο	360€
2015-07-30	Εξοδα - Mc	25€
2015-07-27	Εσοδα - coca	12€
2015-07-24	Εσοδα - Κουλουρι	12€
2015-07-16	Εξοδα - ΔΕΗ	211€
2015-07-01	Εξοδα - Νοικι	300€
2015-07-01	Εσοδα - Μισθος	540€

Εικόνα 27: Όλες οι εγγραφές



Εικόνα 28: Έσοδα



Εικόνα 29: Έξοδα

Για να το πραγματοποιήσω πήγα και έβαλα id σε όλα τα κουμπιά, μετά έφτιαξα ένα script το οποίο θα παίρνει όλες τις εγγραφές και αναλόγως ποιο κουμπί θα πατήσω θα πάρει και θα συγκρίνει το id του κουμπιού με το id του τύπου της εγγραφής. Με αυτό τον τρόπο μου γυρνάει τα αποτελέσματα που θέλω. Το κάθε κουμπί έχει την δικιά του function.

Επίσης χρησιμοποίησα την δυνατότητα που μου προσφέρει το framework JQuery-mobile και πρόσθεσα την αναζήτηση.



Εικόνα 30: Αναζήτηση

Κάτι πολύ σημαντικό σε εφαρμογές εσόδων-εξόδων είναι να μπορεί ο χρήστης να παρακολουθεί συνεχώς τα συνολικά του έσοδα, έξοδα και οπωσδήποτε το ισοζύγιο του.

Έσοδα	Έξοδα	Ισοζύγιο
624	1113	-489

Εικόνα 31: Ισοζύγιο

Για να δουλέψει όμως αυτό σωστά πρέπει να μπει στο window.onload ο κώδικας του, ώστε κάθε φορά που φορτώνει η σελίδα να φορτώνει και τα ποσά.

```
//take income,outcome,sum
db.transaction(function(tx) {
    tx.executeSql(selectAllDate, [], function(tx, result) {
        dataset = result.rows;

        var x=0,inc=0,out=0,sum=0;

        for (var i = 0, item = null; i < dataset.length; i++) {
            item = dataset.item(i);

            x=item['charge'];

            if(item['category']=='Έσοδα')
            {
                inc=inc+x;
            }
            else
            {
                out=out+x;
            }
        }

        sum= inc - out;

        document.getElementById("sum").innerHTML = sum;
        document.getElementById("inc").innerHTML = inc;
        document.getElementById("out").innerHTML = out;

    });
});
```

Εικόνα 32: Κώδικας 10

Τρέχω πάλι όλες τις εγγραφές του πίνακα μου και συγκρίνω πάλι το id του Input category με το αν είναι Έσοδα, αν είναι έσοδα τότε περνάω την χρέωση του σε μια νέα μεταβλητή με όνομα Inc αν όχι την περνάω σε μια μεταβλητή με όνομα out.

Στο τέλος της if μου αφαιρώ από το inc το out και το αποτέλεσμα το περνάω στην μεταβλητή sum ώστε να πάρω το ισοζύγιο μου. Το μόνο που είχα να κάνω μετά είναι να πάρω αυτές τις τιμές με το document.getelementbyid και να τις εμφανίσω.

Βέβαια αντιμετώπισα πρόβλημα σε αυτό το κομμάτι, επειδή όταν περνούσα μια νέα εγγραφή δεν μου την υπολόγιζε κατευθείαν. Έτσι χρειάστηκε να προσθέσω κάποιον κώδικα jquery στην function inserRecord() ώστε να κάνει refresh πρώτου με πάει στην homepage.

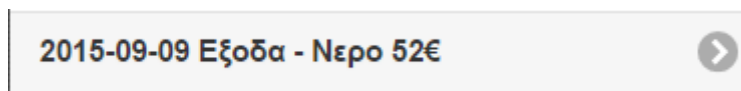
```
$('#my_new_form').on('submit', function (e) {  
    if (e.preventDefault) e.preventDefault();  
    return false;  
});  
  
var serializedFormData = $('#my_new_form').serialize();  
  
$.ajax({  
    url: '#add_form',  
    type: 'POST',  
    data: serializedFormData  
});  
location.reload();  
$.mobile.changePage("#homepage");  
}
```

Εικόνα 33: Κώδικας 11

Το \$.mobile.changePage(); είναι δυνατότητα που προσφέρει το JQuery-mobile.

Τελειώνοντας την homepage θεώρησα υποχρεωτικό ο χρήστης να έχει δικαίωμα να κάνει αλλαγές αν το επιθυμεί σε κάποια από τις εγγραφές του ή ακόμα και να μπορεί να διαγράψει μια.

Ας πάρουμε σαν παράδειγμα την εξής εγγραφή



Εικόνα 34: Διαγραφεί

Πατώντας πάνω της μας πάει σε μια άλλη σελίδα γνώστη ως «δυναμική σελίδα». Τι είναι μια δυναμική σελίδα? Είναι το εξής πράγμα

```

for (var i = 0, item = null; i < dataset.length; i++) {
    item = dataset.item(i);
    var linkeditdelete = '<li id="' + item['id'] + '">' + '<a href="#" + item['id'] + '" data-icon="edit">' + item['date'] + '
' + item['category'] + ' - ' + item['type'] + ' ' + item['charge'] + ' €' + '</a></li>';
    $("#form_list").append(linkeditdelete);
    var form_page_Element_HTML = '<div class="form_page" id="item" + item['id'] + '" data-role="page" data-category="' + item['category'] +
"'" >' + '<div data-role="header" class="header" style="background-color:#5bc0de; color: #ffffff;">' + '<h1>' + item['category'] + '</h1>' +
'<a href="#" data-icon="delete" class="ui-btn-right" onclick="deleteRecord(' + item['id'] + ');">Διαγραφή</a>' + '</div>' + '<div data-
role="content">' + '<p>' + '<label for="type"><b>Τύπος συναλλαγής:</b></label>' + '<input type="text" data-clear-btn="true" name="typeInput"
id="typeInput" value="' + item['type'] + '" />' + '</p>' + '<p>' + '<label for="date"><b>Ημερομηνία συναλλαγής:</b></label>' + '<input
type="text" data-clear-btn="true" name="dateInput" id="dateInput" value="' + item['date'] + '" />' + '</p>' + '<p>' + '<label for="charge">
<b>Ποσό συναλλαγής:</b></label>' + '<input type="text" data-clear-btn="true" name="chargeInput" id="chargeInput" onkeypress="validate(event)"
value="' + item['charge'] + '" />' + '</p>' + '<p>' + '<label for="notes"><b>Σημειώσεις:</b></label>' + '<input type="text" data-clear-
btn="true" name="notesInput" id="notesInput" value="' + item['notes'] + '" />' + '</p>' + '<center><button type="button" class="btn btn-info
btn-lg active" id="btnUpdate" data-inline="true" onclick="updateRecord(' + item['id'] + ');">Ανανέωση</button>' + '<a href="#" + item['id'] + '" data-
role="button" data-inline="true" class="btn btn-default btn-lg active"> Ακύρωση </a></center>' + '</div>' + '</div>';
    $('body').append(form_page_Element_HTML);
    $("#form_list").listview("refresh");
}
});
});
}

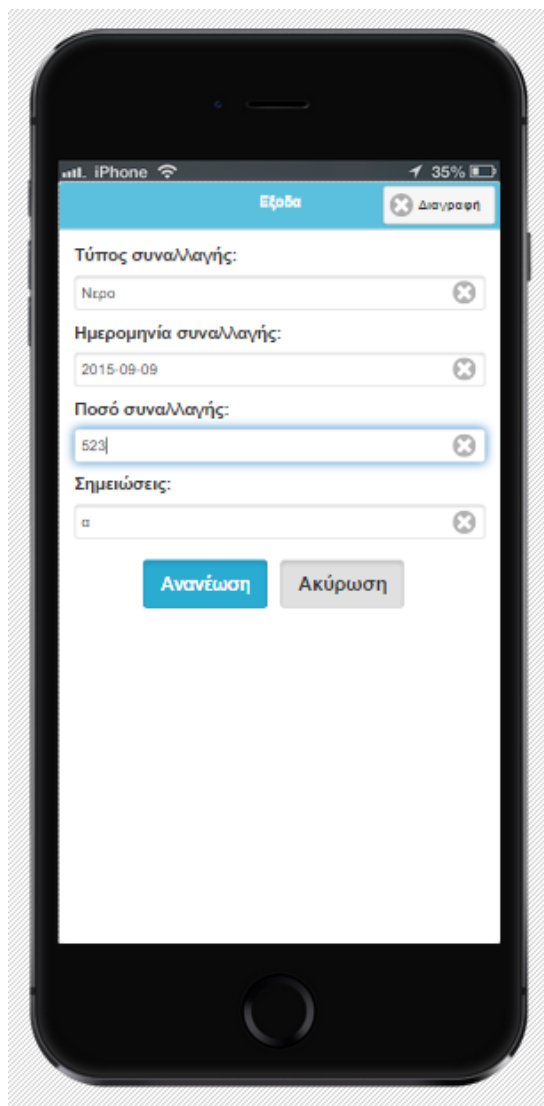
```

Εικόνα 35: Κώδικας 12

Η δυναμική σελίδα ουσιαστικά δεν υπάρχει αλλά δημιουργείται κάθε φορά που ο χρήστης θα κάνει μια ενέργεια είτε πατώντας ένα κουμπί. Ο τρόπος που ξεχωρίζουμε μια δυναμική σελίδα είναι με τα διπλά ή μονά αυτάκια που βάζουμε στην αρχή και στο τέλος του tag μας. Επίσης πολύ σημαντικό είναι να παίρνουμε το id της εγγραφής μας.

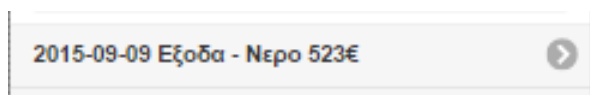
Πάμε πίσω την δικιά μας περίπτωση, ουσιαστικά αυτό που κάνω εδώ είναι ότι φτιάχνω την ίδια σελίδα με την add\_form εξωτερικά και το μόνο που διαφέρει είναι ότι περνάω μέσα στα πεδία μου τα δεδομένα μου το οποίο το κάνω τρέχοντας πρώτα μια for.

Ας δοκιμάσουμε να κάνουμε μια αλλαγή στην εγγραφή μας



Εικόνα 36: Edit #2

Προσθέτω στο «Ποσό συναλλαγής» ένα 3 στο τέλος και πατάω Ανανέωση.



Εικόνα 37: Edit #3

Δουλεύει μια χαρά. Πάμε τώρα να δούμε πίσω στον κώδικα τι έχω κάνει.

```

function updateRecord(id)
{
    var usertypeupdate = $('#item' + id).find('#typeInput').val();
    var userdateupdate = $('#item' + id).find('#dateInput').val();
    var userchargeupdate = $('#item' + id).find('#chargeInput').val();
    var usernotesupdate = $('#item' + id).find('#notesInput').val();
    var category = $('#item' + id).attr("data-category");
    var useridupdate = id;

    db.transaction(function (tx) { tx.executeSql(updateStatement, [usertypeupdate, userdateupdate, userchargeupdate, usernotesupdate,
    Number(useridupdate)], updateLoad, onError); });

    $('#li' + id).html("<a href='#item'+ id +'> data-icon='edit'> + userdateupdate + ' ' + category + ' - ' + usertypeupdate + ' ' +
    userchargeupdate + ' €' + '</a>");

    $('#form_list').listview("refresh");
}

```

Εικόνα 38: Κώδικας 13

Μοιάζει αρκετά με τον κώδικα του insertRecord() *άλλα έχει κάποιες σημαντικές διαφορές.*

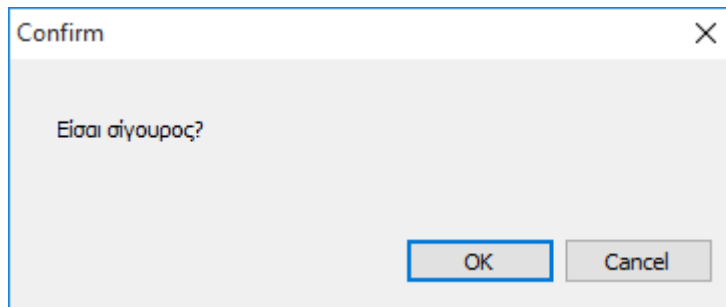
Η πρώτη διαφορά είναι ότι έχω προσθέσει το \$('#item' + id).find(...).val();

Χωρίς αυτό μου έβρισκε πάντα την πρώτη εγγραφή και της άλλα τις τιμές, άρα έπρεπε να βρώ έναν τρόπο να παίρνω την εγγραφή που ήθελα κάθε φορά εγώ και αυτό γίνεται με τον τρόπο από πάνω. Τα υπόλοιπα παραμένουν τα ίδια, κάνω exequitesql το queirie μου και στο τέλος κάνω refresh την λίστα μου.

Τώρα πάμε να διαγράψουμε την εγγραφή μας.

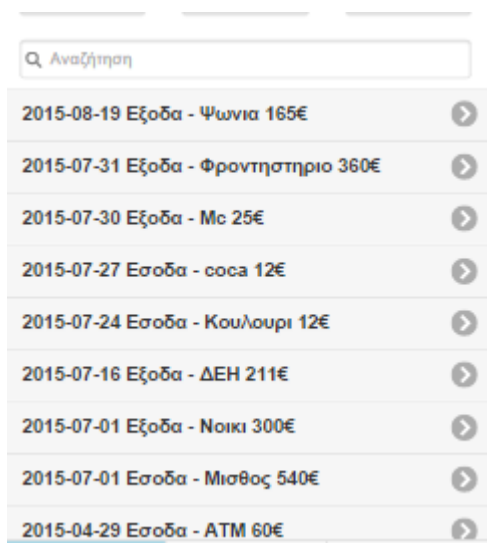
Εικόνα 39: Delete

Πατάμε το κουμπί Διαγραφή πάνω δεξιά και μας πετάει ένα παράθυρο επιβεβαίωσης



Εικόνα 40: Confirm

Πατάμε OK



Εικόνα 41: Delete #2

Και η εγγραφή μας διαγράφηκε. Ο κώδικα που χρησιμοποιήθηκε είναι ο εξής

```
function deleteRecord(id)
{
  if (confirm("Είσαι σίγουρος?")) {
    var iddelete = id.toString();

    db.transaction(function (tx) { tx.executeSql(deleteStatement, [id], deleteLoad, onError); });

    resetForm();

    $.mobile.changePage("#homepage");
  }
  return false;
}
```

Εικόνα 42: Κώδικας 14



Ουσιαστικά φτιάχνω μια function και επιλέγοντας το id εκτελώ το query DELETE. Έχω προσθέσει και το confirm(“Είσαι σίγουρος?”) σε περίπτωση που ο χρήστης πατήσει κατά λάθος το κουμπί να μπορεί να ακυρώσει την διαγραφή.

Τελειώνοντας με την homepage ασχολήθηκα με την balance σελίδα, στην οποία ο χρήστης έχει την δυνατότητα να αναζητήσει κινήσεις που έχει κάνει σε ημερομηνίες που ο ίδιος θα επιλέξει, έτσι του δίνεται η δυνατότητα να βλέπει μεμονωμένα έσοδα, έξοδα και ισοζύγιο.

Έσοδα	Έξοδα	Ισοζύγιο
0	0	0

Από:  
dd/mm/yyyy

Εώς:  
dd/mm/yyyy

**Εμφάνιση**

Εικόνα 43: Balance #2

Πάνω έχω πάλι τα 3 στοιχεία μου (έσοδα, έξοδα, ισοζύγιο) τα οποία μόλις επιλέξω τις ημερομηνίες και πατήσω «Εμφάνιση» θα μου υπολογίσει από τις εγγραφές που θα πάρω τα ανάλογα ποσά. Επίσης θα μου εμφανίσει από κάτω από το κουμπί όλες εγγραφές που είναι στο διάστημα αυτό. Ας κάνουμε ένα παράδειγμα

Έσοδα	Έξοδα	Ισοζύγιο
24	761	-737

Από: 16/07/2015

Εώς: 30/08/2015

**Εμφάνιση**

- 2015-07-16 Έξοδα - ΔΕΗ 211€
- 2015-07-31 Έξοδα - Φροντιστήριο 360€
- 2015-08-19 Έξοδα - Ψωνια 165€
- 2015-07-30 Έξοδα - Mc 25€
- 2015-07-24 Έσοδα - Κουλουρι 12€
- 2015-07-27 Έσοδα - coca 12€

Εικόνα 44: Balance #3

## Πάμε και πίσω στον κώδικα

```
//Note: 00 is month i.e. January
var dateOne = $('#date_start').val();
var dateTwo = $('#date_end').val();
if (dateOne < dateTwo) {
    db.transaction(function(tx) {
        tx.executeSql(selectDate, [dateOne, dateTwo], function(tx, result) {
            dataset = result.rows;

            var x=0,income=0,outcome=0,balance=0;

            for (var i = 0, item = null; i < dataset.length; i++) {
                item = dataset.item(i);

                var linkeditdelete = '<li id="li' + item['id'] + '>' + '<a href="#item'+ item['id'] +'> data-icon='edit'>' + item['date'] + '
' + item['category'] + ' - ' + item['type'] + ' ' + item['charge'] + '€' + '</a></li>';

                $('#form_compare').append(linkeditdelete);

                x=item['charge'];

                if(item['category']=='Εσοδα')
                {
                    income=income+x;
                }
                else
                {
                    outcome=outcome+x;
                }
            }

            $('#form_compare').listview("refresh");
            balance= income - outcome;

            document.getElementById("balance_sum").innerHTML = balance;
            document.getElementById("income").innerHTML = income;
            document.getElementById("outcome").innerHTML = outcome;

        });

    });

}
else {
    alert("Λάθος επιλογή ημερομηνιών!!! Ξαναπροσπαθήστε.");
}
}
```

Εικόνα 45: Κώδικας 15

Στην αρχή παίρνω τις δυο ημερομηνίες από τα πεδία μας και τις περνάμε σε νέες μεταβλητές, αμέσως μετά συγκρίνω τις δυο ημερομηνίες ώστε η πρώτη να είναι πιο παλιά από την δεύτερη αλλιώς αν δεν είναι έτσι έχω βάλει να πετάει ένα alert και να λέει ότι «Λάθος επιλογή ημερομηνιών!!! Ξαναπροσπαθήστε», αν όμως οι ημερομηνίες είναι σωστές τότε τρέχω μια for και μέσα της μια if ώστε να ελέγξει όλες τις ημερομηνίες να είναι στο χρονικό περιθώριο που έχω δώσει και να τις περνάει αναλόγως αν είναι έσοδα ή έξοδα σε δυο νέες μεταβλητές income, outcome αντίστοιχα. Μετά κάνει τον υπολογισμό ώστε να μας βγάλει το ισοζύγιο και τέλος εμφανίζουμε τις τιμές μας στην εφαρμογή μας.

Στην about σελίδα μου έχω γράψει δυο κείμενα το ένα στα ελληνικά το άλλο στα αγγλικά, είναι περιγράφει της εφαρμογής ώστε ο νέος χρήστης που περνάει την εφαρμογή στο κινητό του να ξέρει τις δυνατότητες της.

## 5. Αποτελέσματα

Το αποτέλεσμα της πτυχιακής εργασίας αυτής είναι η επίτευξη να καταλάβουμε πως λειτουργούν οι υβριδικές εφαρμογές και ταυτόχρονα να καταλάβουμε πως λειτουργεί η αγορά των εφαρμογών κινητών. Να δούμε τις διάφορες τεχνολογίες που χρησιμοποιούνται και να πάρουμε εικόνα μια από αυτές.

Τα ευρήματα της εργασίας κατέστησαν σαφές ότι παρά την πρόοδο της διαδικτυακής αρχιτεκτονικής, δεν είναι ακόμη εφικτό να αντικαταστήσει τις native εφαρμογές σε όλα τα επίπεδα. Υπάρχουν ακόμη πολλές ελλείψεις που αφορούν την πρόσβαση στα χαρακτηριστικά της συσκευής. Παρόλα αυτά έχει υπάρξει μεγάλη πρόοδος. Τα διάφορα εργαλεία που προσφέρουν έτοιμη λειτουργικότητα όπως η jQuery Mobile, προσφέρουν εξαιρετικά σημαντικές ευκολίες.

Η υβριδική προσέγγιση

δίνει λύση στο πρόβλημα της απουσίας πρόσβασης σε κάποια χαρακτηριστικά της συσκευής όμως το πρόβλημα της απόδοσης αλλά και το γεγονός ότι το μέρος της εφαρμογής που αφορά αυτήν την πρόσβαση θα πρέπει να αλλαχθεί για κάθε λειτουργικό σύστημα, αποτελεί πρόβλημα. Οι cross-compiled εφαρμογές μπορεί να αποτελούν λύση για κάποιες περιπτώσεις όμως το κόστος που συνοδεύει τα απαιτούμενα εργαλεία, ο χρόνος εκμάθησης αυτών, αλλά και η περιορισμένη υποστήριξη εφαρμογών για διαφορετικές συσκευές, τις καθιστά λίγο καλύτερες από την native προσέγγιση.

Αυτή η πτυχιακή μπορεί να ωφελήσει κάποιον που δεν έχει ξανά ασχοληθεί με προγραμματισμό εφαρμογής κινητού σε βαθμό να πάρει μια εικόνα για το πώς λειτουργούν, που χρησιμεύει το καθένα και τι δυνατότητες μας δίνει το καθένα. Είναι σημαντικό προτού ξεκινήσει κάποιος να ασχολείται με την ανάπτυξη εφαρμογών κινητού να ξέρει τα θετικά και αρνητικά τις κάθε τεχνολογίας. Άλλωστε όσο περνάει ο καιρός η τεχνολογία κινητών συσκευών μπαίνει όλο και πιο πολύ στην ζωή μας, μιας και ότι κινητό κυκλοφορεί στην αγορά είναι κατά 90% smartphone, και αυτό έχει ως αποτέλεσμα η ζήτηση των εφαρμογών γίνεται όλο και πιο πολύ αναγκαία και ο ανταγωνισμός πιο μεγάλος.

Άρα θεωρώ την πτυχιακή μια καλή βάση και μια καλή αρχή για έναν αρχάριο πάνω στις εφαρμογές κινητών.

Άλλο ένα στόχο που έχω με την πτυχιακή μου είναι να κινήσω το ενδιαφέρον των νέων χρηστών ώστε να δούνε πόσο ενδιαφέρον είναι ο προγραμματισμός εφαρμογών κινητών.

### 5.1 Συμπεράσματα

Το συμπέρασμα που έβγαλα υλοποιώντας την εργασία μου είναι ότι οι υβριδικές εφαρμογές έχουν πολλές δυνατότητες αλλά και πολλά αρνητικά ταυτόχρονα. Το ζήτημα είναι το τι θέλει να κάνει ο προγραμματιστής κάθε φορά, γιατί μπορεί μια υβριδική εφαρμογή να έχει την δυνατότητα να χρησιμοποιείται σε πολλά λειτουργικά αλλά ταυτόχρονα έχει το αρνητικό ότι δεν μπορούμε να κάνουμε τα πράγματα που κάνουμε σε native.

Επίσης άλλο ένα συμπέρασμα που έβγαλα είναι ότι με το framework JQuery mobile αν έχεις κάποιες γνώσεις σε Html5, Css, Javascript μπορείς να μπεις γρήγορα στο κλίμα και να

φτιάξεις μικρές εφαρμογές σε σύντομο χρόνο, γιατί υπάρχουν framework όπως Ionic τα οποία πρέπει ο χρήστης να γνωρίζει angular.js και MVC.

## 5.2 Μελλοντικές επεκτάσεις

Στα πλαίσια της πτυχιακής εργασίας, μελλοντικές επεκτάσεις εντοπίζονται κυρίως στην εφαρμογή. Η εφαρμογή διαθέτει ένα πλήθος από υπηρεσίες που απευθύνονται σε χρήστες απλούς και επαγγελματίες, αλλά θέλω να προσθέσω και άλλα πράγματα όπως ειδοποιήσεις, κάποια μενού επιπλέον και κάποιες δυνατότητες στις αναζητήσεις.

Είναι αναγκαίες οι επεκτάσεις στην εφαρμογή μου, λόγω ότι από τις άλλες εφαρμογές που είδα πρέπει να προστεθούν κάποια πράγματα ακόμα ώστε να μπορεί αν είναι ανταγωνιστική στο android market που θα την ανεβάσω.

## Βιβλιογραφία

- [1] <https://el.wikipedia.org/wiki/IOS>
- [2] <https://el.wikipedia.org/wiki/Android>
- [3] [https://el.wikipedia.org/wiki/Windows\\_Phone](https://el.wikipedia.org/wiki/Windows_Phone)
- [4] Ptyxiakh\_Alqis\_Kalogeris.pdf
- [5] <https://el.wikipedia.org/wiki/HTML5>
- [6] <https://el.wikipedia.org/wiki/JavaScript>
- [7] <https://el.wikipedia.org/wiki/CSS>
- [8] <https://el.wikipedia.org/wiki/JQuery>
- [9] [https://en.wikipedia.org/wiki/JQuery\\_Mobile](https://en.wikipedia.org/wiki/JQuery_Mobile)
- [10] <https://el.wikipedia.org/wiki/SQLite>
- [11] <https://el.wikipedia.org/wiki/Bootstrap>
- [12] Sofios.pdf