



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ  
ΠΛΗΡΟΦΟΡΙΚΗΣ

## **Κατασκευή Web & Android εφαρμογής για δημιουργία και παρουσίαση δυναμικών δεδομένων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Ζ. Αθανασόπουλος

**Επιβλέπων :** Γεώργιος Στασινόπουλος  
Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2014





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ  
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

## Κατασκευή Web & Android εφαρμογής για δημιουργία και παρουσίαση δυναμικών δεδομένων

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Παναγιώτης Ζ. Αθανασόπουλος

**Επιβλέπων :** Γεώργιος Στασινόπουλος

Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .... Οκτωβρίου 2014.  
Αθήνα, Οκτώβριος 2014

.....  
Γεώργιος Στασινόπουλος  
Καθηγητής ΕΜΠ

.....  
Ευστάθιος Συκάς  
Καθηγητής ΕΜΠ

.....  
Μιχαήλ Θεολόγου  
Καθηγητής ΕΜΠ

.....

Παναγιώτης Ζ. Αθανασόπουλος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Παναγιώτης Αθανασόπουλος, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η έρευνα, η σχεδίαση και η κατασκευή μιας Web και μιας Android εφαρμογής παρουσίασης δυναμικών δεδομένων. Κυριότερο ενδιαφέρον της εργασίας είναι η δυνατότητα η εφαρμογή να είναι πλήρως δυναμική και καθώς ο χρήστης δημιουργεί το «σχήμα» των δεδομένων του, να δημιουργούνται αυτόματα τα αντίστοιχα “views” σε web αλλά και στην Android εφαρμογή, ώστε να δίνεται η δυνατότητα προβολής και επεξεργασίας των δεδομένων του.

Επίσης σημαντική προϋπόθεση στο σχεδιασμό μας είναι η δομή μας να χρησιμοποιείται ως πρότυπο για παραγωγή ομοειδών εφαρμογών των οποίων το γενικό development να μπορεί να ξεκινήσει σε πολύ μικρό χρονικό διάστημα.

Στο πρώτο κεφάλαιο υπάρχει η ανάλυση των απαιτήσεων των εφαρμογών, που ορίζονται όλα τα χαρακτηριστικά και οι λειτουργίες που απαιτείται να έχουν και να εκτελούν.

Στο δεύτερο κεφάλαιο υπάρχει η παρουσίαση και περιγραφή των βασικών ορολογιών που είναι απαραίτητες να γνωρίζει κάποιος για τη λειτουργία ή και τη δημιουργία αυτών των εφαρμογών.

Στο τρίτο κεφάλαιο περιγράφονται όλες οι τεχνολογίες, βιβλιοθήκες καθώς και εφαρμογές που χρησιμοποιούν οι εφαρμογές μας κατά τη δημιουργία αλλά και τη λειτουργία τους.

Τέλος στο τέταρτο κεφάλαιο πραγματοποιείται εκτενής περιγραφή των λειτουργιών των δυο εφαρμογών (PHP Web app & Android application) καθώς και τα πιο σημαντικά μέρη του κώδικα που εκτελείται σε κάθε περίπτωση.

## Λέξεις κλειδιά

Web εφαρμογή, Android εφαρμογή, PHP, MySQL, JSON, XAMPP, Apache, δυναμικά δεδομένα, δυναμικό σχήμα βάσης δεδομένων, jQuery, Javascript, HTML, Έξυπνο τηλέφωνο

## **Abstract**

The target of the current thesis is the research, design and development of a Web and Android application for presenting dynamic data. The main interest of this project is the ability of the application to be totally dynamic and while the user is creating the database schema, the respective views in the web and in the Android application will be created automatically so the user will be able to view and edit his data.

Another important specification in our design is that the project's structure will be able to be used as a prototype for the development of other similar applications and their development will be able to start within a small amount of time.

The first chapter includes the specifications analysis for both of the applications, where all the features and usecases that are required are set.

In the second chapter there is a presentation and description of the basic terminology that are required from someone in order to be able to use and create these applications.

In the third chapter all the technologies, libraries and applications, used by our applications during their development and usage, are described.

Finally, in the fourth chapter there is an extended description of all the usecases of the two applications (PHP Web app & Android application) and also the most important parts of the source code that is used in each case are described.

## **Key words**

Web application, Android application, PHP, MySQL, JSON, XAMPP, Apache, dynamic data, dynamic database schema, jQuery, Javascript, HTML, smartphone

## Ευχαριστίες

Πριν την παρουσίαση αυτής της εργασίας, αισθάνομαι την ανάγκη να ευχαριστήσω θερμά όλους όσους με βοήθησαν και έπαιξαν καθοριστικό ρόλο στην πραγματοποίησή της και συγκεκριμένα:

Τον κ. Γεώργιο Στασινόπουλο, καθηγητή της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ του Εθνικού Μετσόβιου Πολυτεχνείου, επιβλέπων της εργασίας για την πολύτιμη βοήθεια και καθοδήγηση σε όλη τη διάρκεια της δουλειά μου καθώς και την ενδιαφέρουσα αλλαγή που πρότεινε στο θέμα της εργασίας.

Την οικογένειά μου και τους κοντινούς μου φίλους για την υπομονή, τη συμπαράσταση και την υποστήριξη που μου παρείχαν σε όλη τη διάρκεια των σπουδών μου και γενικότερα σε όλα τα στάδια της ζωής μου.

Τέλος θα ήθελα να ευχαριστήσω τη Χάρης για την υπομονή, τη συμπαράσταση και την υποστήριξη που μου παρείχε στην προσπάθειά μου να ολοκληρώσω τις σπουδές μου.

## Περιεχόμενα

Περίληψη .....	5
Abstract .....	6
Ευχαριστίες .....	7
Περιεχόμενα .....	8
1. Ανάλυση απαιτήσεων .....	11
2. Ορολογία .....	12
2.1 Ιστοσελίδα .....	12
2.2 Web browser .....	12
2.2.1 Περιγραφή .....	12
2.2.2 Κύρια χαρακτηριστικά.....	12
2.3 Web server.....	13
2.3.1 Περιγραφή.....	13
2.3.2 Κύρια χαρακτηριστικά.....	13
2.4 Άδειες λογισμικού (Software License) .....	13
2.4.1 Περιγραφή.....	13
2.4.2 Συνηθισμένες άδειες λογισμικού.....	14
2.5 Αντικειμενοστραφής προγραμματισμός (Object Oriented Programming).....	15
2.6 Model-View-Controller (MVC) .....	15
2.7 HTML Template / Icons.....	15
2.8 SmartPhone.....	16
3. Ανάλυση τεχνολογιών και βιβλιοθηκών.....	17
3.1 HTML.....	17
3.1.1 Περιγραφή.....	17
3.1.2 Δομή μιας HTML σελίδας.....	18
3.2 CSS (Cascading Style Sheets) .....	20
3.2.1 Περιγραφή.....	20
3.2.2 Σύνταξη.....	20
3.2.3 Κληρονομικότητα Στυλ.....	21
3.3 JavaScript .....	22
3.3.1 Περιγραφή.....	22
3.3.2 Σύνταξη.....	22
3.3.3 Χρήση σε ιστοσελίδες.....	22
3.3.4 Χρήση εκτός ιστοσελίδων.....	23
3.4 JQuery.....	23
3.4.1 Περιγραφή.....	23
3.4.2 Χαρακτηριστικά.....	24
3.4.3 Χρήση.....	24
3.5 SQL (Structured Query Language) .....	25
3.5.1 Περιγραφή.....	25
3.5.2 Γλωσσικά στοιχεία.....	26
3.5.3 Operators.....	26
3.5.4 Τύποι Δεδομένων.....	28
3.6 PHP.....	28
3.6.1 Περιγραφή.....	28
3.6.2 Σύνταξη.....	29
3.6.3 License.....	30



3.7 MySQL.....	30
3.7.1 Περιγραφή.....	30
3.7.2 Σύνταξη.....	31
3.7.3 License.....	31
3.7.4 Χρήση με PHP.....	32
3.8 Java.....	33
3.8.1 Περιγραφή.....	33
3.8.2 Σύνταξη.....	34
3.9 Android.....	35
3.9.1 Περιγραφή.....	35
3.9.2 Εφαρμογές.....	35
3.9.3 Hardware.....	35
3.9.4 Διαχείριση μνήμης.....	36
3.10 Apache HTTP Server.....	36
3.10.1 Περιγραφή.....	36
3.10.2 License.....	36
3.11 XAMPP.....	36
3.11.1 Περιγραφή.....	36
3.11.2 Απαιτήσεις και χαρακτηριστικά.....	37
3.11.3 Χρήση.....	37
3.11.4 Εργαλεία - Τεχνολογίες.....	37
3.11.5 License.....	37
3.12 phpMyAdmin.....	38
3.12.1 Περιγραφή.....	38
3.12.2 Χαρακτηριστικά.....	38
3.12.3 License.....	38
3.13 JSON.....	39
3.13.1 Περιγραφή.....	39
3.13.2 Τύποι δεδομένων και σύνταξη.....	39
3.14 GSON (Βιβλιοθήκη).....	40
3.14.1 Περιγραφή.....	40
3.14.2 Χαρακτηριστικά.....	40
3.14.3 License.....	40
3.15 DataTables ( jQuery plugin ).....	41
3.15.1 Περιγραφή.....	41
3.15.2 Χαρακτηριστικά.....	41
3.15.3 Χρήση.....	41
3.15.4 License.....	41
4. Κατασκευή Εφαρμογών.....	42
4.1 Βάση δεδομένων.....	42
4.1.1 Περιγραφή.....	42
4.1.2 Πίνακας users.....	42
4.1.3 Πίνακας db_versions.....	43
4.1.4 Πίνακας tables_reference.....	43
4.1.5 Πίνακας columns_reference.....	43
4.2 Ανάλυση λειτουργίας και κώδικα εφαρμογών.....	44
4.2.1 Γενική δομή κώδικα PHP.....	44
4.2.2 Γενική δομή κώδικα Android.....	46
4.2.3 Εκκίνηση εφαρμογής – Πρόσβαση στην εφαρμογή.....	46
4.2.4 Εγγραφή χρήστη.....	50

4.2.5 Αρχική οθόνη.....	53
4.2.6 Προβολή πεδίων πίνακα.....	59
4.2.7 Επεξεργασία ονόματος πίνακα.....	62
4.2.8 Διαγραφή πίνακα.....	65
4.2.9 Προσθήκη πεδίου πίνακα.....	67
4.2.10 Επεξεργασία ονόματος πεδίου πίνακα.....	70
4.2.11 Διαγραφή πεδίου πίνακα.....	72
4.2.12 Προσθήκη Referenced πίνακα.....	74
4.2.13 Προβολή δεδομένων πίνακα.....	78
4.2.14 Αναζήτηση δεδομένων πίνακα.....	82
4.2.15 Λεπτομερής εμφάνιση εγγραφής πίνακα.....	83
4.2.16 Προσθήκη δεδομένων.....	87
4.2.17 Επιλογή δεδομένων από referenced πίνακα.....	90
4.2.18 Επεξεργασία εγγραφής πίνακα.....	93
4.2.19 Διαγραφή εγγραφής πίνακα.....	96
4.2.20 Εμφάνιση δεδομένων τύπου LOCATION στα Google Maps.....	98
4.2.21 Android-Webserver data.....	99
4.2.22 Android Social media Share.....	100
Βιβλιογραφία.....	102

## 1. Ανάλυση απαιτήσεων

Σκοπός της παρούσας διπλωματικής εργασίας είναι η έρευνα, η σχεδίαση και η κατασκευή μιας Web και μιας Android εφαρμογής παρουσίασης δυναμικών δεδομένων. Κυριότερο ενδιαφέρον της εργασίας είναι η δυνατότητα η εφαρμογή να είναι πλήρως δυναμική και καθώς ο χρήστης δημιουργεί το «σχήμα» των δεδομένων του, να δημιουργούνται αυτόματα τα αντίστοιχα “views” σε web αλλά και στην Android εφαρμογή, ώστε να δίνεται η δυνατότητα προβολής και επεξεργασίας των δεδομένων του.

Επίσης σημαντική προϋπόθεση στο σχεδιασμό μας είναι η δομή μας να χρησιμοποιείται ως πρότυπο για παραγωγή ομοειδών εφαρμογών των οποίων το γενικό development να μπορεί να ξεκινήσει σε πολύ μικρό χρονικό διάστημα.

Η web εφαρμογή θα αποτελείται από:

- Ο χρήστης θα έχει τη δυνατότητα να δημιουργήσει μέσω της εφαρμογής το αρχικό σχήμα της βάσης και στη συνέχεια να προσθέτει πεδία στους πίνακες καθώς και να προσθέτει νέους πίνακες (one-to-many) για καλύτερη οργάνωση των δεδομένων.
- Ο χρήστης θα μπορεί να προσθέσει, να αφαιρέσει, να επεξεργαστεί και να αναζητήσει δεδομένα που βρίσκονται στη βάση.
- Με τον παραπάνω σχεδιασμό, ο χρήστης θα έχει τη δυνατότητα δημιουργίας ομοειδών εφαρμογών και το development μιας τέτοιας εφαρμογής θα μπορεί να γίνεται instantiated μέσα σε πολύ μικρό χρονικό διάστημα.
- Front-end web-site παρουσίασης των δεδομένων.
- Ο επισκέπτης του web-site θα μπορεί να κάνει μια «εγγραφή-χρήστη». Θα μπορεί επίσης να κάνει και “login” σε περίπτωση που έχει ήδη εγγραφεί.

Η Android εφαρμογή θα αποτελείται από:

- Κατασκευή μιας “Native” Android εφαρμογής πάνω στο Android SDK η οποία θα παρουσιάζει στο χρήστη τα δεδομένα της βάσης με όλες τους τις πληροφορίες χρησιμοποιώντας τα Android GUI Elements.
- Θα δίνεται η δυνατότητα στο χρήστη της εφαρμογής να κάνει εγγραφή στο σύστημα καθώς και “login” σε περίπτωση που έχει ήδη εγγραφεί.
- Θα υπάρχει η δυνατότητα οι χρήστες να κάνουν “share” τα δεδομένα σε social-media όπως το facebook και το twitter.
- Ο χρήστης θα έχει τη δυνατότητα να προσθέσει, να αφαιρέσει και να επεξεργαστεί τα δεδομένα μέσα από την εφαρμογή.
- Ο χρήστης θα έχει τη δυνατότητα να δημιουργήσει μέσω της εφαρμογής το αρχικό σχήμα της βάσης και στη συνέχεια να προσθέτει πεδία στους πίνακες καθώς και να προσθέτει νέους πίνακες (one-to-many) για καλύτερη οργάνωση των δεδομένων.

## 2. Ορολογία

### 2.1 Ιστοσελίδα

Ιστοσελίδα (website) είναι ένα είδος εγγράφου του παγκόσμιου ιστού που περιλαμβάνει πληροφορίες με τη μορφή κειμένου, υπερκειμένου και πολυμέσων.

Πολλές ιστοσελίδες μαζί αποτελούν έναν ισότοπο και εμφανίζονται κάτω από το ίδιο όνομα χώρου (domain name πχ [www.google.gr](http://www.google.gr)). Οι ιστοσελίδες μπορεί να συνδέονται μεταξύ τους μέσω συνδέσμων (links) πάνω στα οποία ο χρήστης μπορεί να κάνει “click” και να οδηγηθεί από τη μία στην άλλη. <sup>[2]</sup>

### 2.2 Web browser

#### 2.2.1 Περιγραφή

Ένας φυλλομετρητής (web browser) είναι ένα λογισμικό το οποίο έχει σκοπό την ανάκτηση, την παρουσίαση και τη μετατροπή ιστοσελίδων από τον παγκόσμιο ιστό (www). Δίνοντας στον browser μία διεύθυνση URL, μπορεί να παρουσιάσει τα δεδομένα που απαιτούμε, τα οποία μπορεί να είναι μια ιστοσελίδα, εικόνες, βίντεο ή άλλα πολυμέσα. Με χρήση συνδέσμων (links) ο χρήστης μπορεί να οδηγήσει τον browser να παρουσιάσει κάποια άλλη ιστοσελίδα ή δεδομένα. <sup>[2]</sup>



Οι κυριότεροι web browsers με τους περισσότερους χρήστες είναι:

- Google Chrome
- Mozilla Firefox
- Opera
- Microsoft Internet Explorer
- Safari

#### 2.2.2 Κύρια χαρακτηριστικά

Τα κυριότερα χαρακτηριστικά των web browsers είναι τα εξής:

- **Κουμπιά Πίσω/Εμπρός.** Για την οδήγηση στην προηγούμενη ή την επόμενη σελίδα.
- **Κουμπί ανανέωσης.** Για την ανανέωση της ενεργής ιστοσελίδας.
- **Κουμπί Stop.** Για τη διακοπή φόρτωσης της ιστοσελίδας.
- **Κουμπί Αρχική.** Για την επιστροφή του χρήστη στην αρχική του σελίδα άμεσα.

- **Μπάρα διεύθυνσης** όπου ο χρήστης γράφει τη διεύθυνση της ιστοσελίδας που θέλει να επισκευθεί.
- **Μπάρα αναζήτησης** όπου ο χρήστης μπορεί να κάνει αναζήτηση σε μηχανή αναζήτησης της επιλογής του.
- **Μπάρα κατάστασης** όπου φαίνεται η κατάσταση φόρτωσης της ιστοσελίδας.
- **Η δυνατότητα παρουσίασης πηγαίου κώδικα HTML.**

## 2.3 Web server

### 2.3.1 Περιγραφή



Ο web server είναι ένα σύστημα υπολογιστή το οποίο επεξεργάζεται αιτήματα μέσω HTTP. Ο όρος αυτός μπορεί να αναφέρεται είτε σε ολόκληρο το σύστημα είτε συγκεκριμένα στο λογισμικό το οποίο αναλαμβάνει τα δέχεται και να χειρίζεται τα αιτήματα HTTP (HTTP requests).

Η πιο κοινή χρήση των web servers είναι να φιλοξενούν ιστοσελίδες, αλλά έχουν και πολλές άλλες χρήσεις όπως αποθήκευση δεδομένων, εκτέλεση enterprise εφαρμογών, διαχείριση email, FTP κ.α. Η επικοινωνία μεταξύ του server και του client γίνεται με χρήση του πρωτοκόλλου Hypertext Transfer Protocol (HTTP).<sup>[2]</sup>

### 2.3.2 Κύρια χαρακτηριστικά

Τα κυριότερα χαρακτηριστικά των web servers είναι τα εξής:

- **Virtual hosting** για τη φιλοξενία πολλών διαφορετικών websites υπό την ίδια διεύθυνση IP.
- **Υποστήριξη μεγάλων αρχείων** για να υπάρχει η δυνατότητα εξυπηρέτησης αιτήσεων για αρχεία μεγαλύτερα από 2GB σε 32-bit συστήματα.
- **Bandwidth throttling** για να περιορίζεται η ταχύτητα των απαντήσεων ώστε να μην κορεννύεται το δίκτυο και να δύναται να εξυπηρετήσει περισσότερους πελάτες.
- **Server Side Scripting** για τη δημιουργία δυναμικών ιστοσελίδων όπως πχ. με PHP.

## 2.4 Άδειες λογισμικού (Software License)

### 2.4.1 Περιγραφή

Μια άδεια λογισμικού είναι ένα νομικό έγγραφο το οποίο ορίζει τον τρόπο χρήσης και διάθεσης ενός λογισμικού. Μια τυπική άδεια λογισμικού δίνει στον τελικό χρήστη την άδεια να χρησιμοποιήσει ένα ή περισσότερα αντίγραφα του

λογισμικού με τέτοιο τρόπο ώστε να μην αποτελεί καταπάτηση των πνευματικών δικαιωμάτων του ιδιοκτήτη του λογισμικού.

Οι άδειες λογισμικού χωρίζονται σε δύο κύριες κατηγορίες. Τις “ιδιωτικές άδειες” (propriety licenses) και τις “Ελεύθερου και ανοιχτού κώδικα” (Free and open source licenses). Το κυριότερο χαρακτηριστικό τους που τις ξεχωρίζει είναι οι όροι με τους οποίους ο χρήστης μπορεί να διανείμει ή να αντιγράψει το λογισμικό.

Στην παρούσα εργασία όλες οι βιβλιοθήκες και τεχνολογίες που χρησιμοποιήσαμε εντάσσονται στην κατηγορία “Ελεύθερου και ανοιχτού κώδικα”.<sup>[15]</sup>

#### 2.4.2 Συνηθισμένες άδειες λογισμικού

Οι πιο συνηθισμένες άδειες λογισμικού είναι οι εξής:

- **Apache License 2.0.** Είναι μια Open source άδεια όπου το λογισμικό που διατίθεται με αυτή μπορεί να χρησιμοποιηθεί για εμπορική χρήση, να γίνουν μετατροπές σε αυτό, να διανεμηθεί, να διανεμηθεί με “υπό-άδεια”, να χρησιμοποιηθεί για ιδιωτική χρήση, να χρησιμοποιηθεί για αιτήσεις πατέντας καθώς και να περιλαμβάνει εγγύηση. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης του λογισμικού για ευθύνες από ζημίες ούτε να χρησιμοποιηθούν τα Trademarks αυτού. Πρέπει να περιλαμβάνεται το Copyright, η άδεια, να δηλώνονται τυχόν αλλαγές καθώς και το αρχείο NOTICE που περιλαμβάνεται.<sup>[15]</sup>
- **MIT License.** Πρόκειται για μία μικρή, ανεκτική άδεια όπου γενικά ο προγραμματιστής μπορεί να κάνει ότι θέλει αρκεί να περιλαμβάνει το αρχικό copyright και την άδεια. Γενικά μπορεί να χρησιμοποιηθεί για εμπορική χρήση, μπορούν να γίνουν μετατροπές στον αρχικό κώδικα, να διανεμηθεί, να διανεμηθεί με “υπό-άδεια” και να χρησιμοποιηθεί για ιδιωτική χρήση. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης του λογισμικού για ευθύνες από ζημίες.<sup>[15]</sup>
- **GNU General Public License v3/v2.** Ο προγραμματιστής μπορεί να αντιγράψει, διανείμει και να κάνει μετατροπές στο λογισμικό αρκεί να γίνεται καταγραφή των αλλαγών στα αρχεία και οι αλλαγές να συνεχίζουν να είναι υπό την ίδια άδεια. Μπορεί να διανεμηθεί η εφαρμογή εμπορικά, αλλά



**Free as in Freedom**

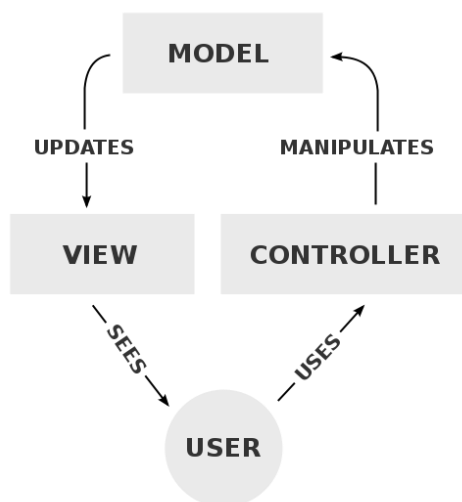
πρέπει να είναι open source υπό την GPLv3. Δεν μπορεί να κατηγορηθεί ο ιδιοκτήτης του λογισμικού για ευθύνες από ζημίες ούτε να διανεμηθεί υπό άλλη άδεια. Πρέπει να περιλαμβάνεται η αρχική έκδοση της βιβλιοθήκης, να ορίζονται οι αλλαγές, να περιλαμβάνεται η άδεια και το copyright καθώς και το παραγόμενο λογισμικό που χρησιμοποιεί βιβλιοθήκη με αυτή την άδεια να είναι open source (αν περιλαμβάνει τη βιβλιοθήκη ως ένα εκτελέσιμο, όχι αν εξυπηρετεί μέσω ενός web-portal).<sup>[15]</sup>

## 2.5 Αντικειμενοστραφής προγραμματισμός (Object Oriented Programming)

Ο αντικειμενοστραφής προγραμματισμός (Object Oriented Programming – OOP) είναι ένα μοντέλο προγραμματισμού που αντιπροσωπεύει την έννοια των “αντικειμένων” τα οποία έχουν πεδία δεδομένων και σχετισμένες διαδικασίες γνωστές ως μεθόδους. Οι πιο γνωστές γλώσσες προγραμματισμού οι οποίες βασίζονται στον αντικειμενοστραφή προγραμματισμό είναι οι Java, C++, Objective-C, C#, Javascript, Perl, Python, Ruby κ.α.

Ένα αντικείμενο είναι μια αφηρημένη μορφή δεδομένων με επιπλέον τον πολυμορφισμό και την κληρονομικότητα. Ένα αντικείμενο επίσης έχει “κατάσταση” (δεδομένα) και “συμπεριφορά” (κώδικας). Τα αντικείμενα συνήθως αναπαριστούν πράγματα του πραγματικού κόσμου, όπως πχ. ένας “πελάτης” κτλ. Τα αντικείμενα σχεδιάζονται σε ιεραρχίες κλάσεων έτσι ένα αντικείμενο μπορεί να είναι το “Προϊόν” που έχει τις γενικές πληροφορίες και αυτό να το κληρονομούν το “Προϊόν Ηλεκτρονικής” και το “Προϊόν κηπουρικής” τα οποία κληρονομούν όλα τα πεδία και μεθόδους του αρχικού αντικειμένου και προσθέτουν τα πεδία και μεθόδους που είναι σχετικά με αυτά. [2]

## 2.6 Model-View-Controller (MVC)



Το MVC είναι ένα μοντέλο αρχιτεκτονικής λογισμικού για δημιουργία διεπαφών χρήστη. Χωρίζει μια εφαρμογή σε 3 διασυνδεδεμένα κομμάτια έτσι ώστε να ξεχωρίσει τον τρόπο με τον οποίο ορίζονται τα δεδομένα εσωτερικά στην εφαρμογή σε σχέση με το πως παρουσιάζονται στο χρήστη.

Το σημαντικότερο μέρος του MVC, το **Model** αναλαμβάνει να διαχειριστεί τη συμπεριφορά της εφαρμογής και τα δεδομένα, τη λογική και τους κανόνες. Ένα **View** μπορεί να είναι οποιαδήποτε μορφή παρουσίασης δεδομένων και πληροφοριών.

Το τρίτο μέρος, ο **Controller** δέχεται την είσοδο και τη μετατρέπει σε εντολές για το model και το view.

Η αρχιτεκτονική MVC χρησιμοποιείται ευρέως σε web εφαρμογές. Επίσης οι πλατφόρμες mobile εφαρμογών (όπως στην περίπτωση μας το Android) είναι έτσι σχεδιασμένες ώστε ο προγραμματισμός να βασίζεται στην αρχιτεκτονική MVC. [2]

## 2.7 HTML Template / Icons

Η χρήση των HTML templates είναι πολύ διαδεδομένη τη σημερινή εποχή στο σχεδιασμό ιστοσελίδες. Οι προγραμματιστές μπορούν να προμηθευτούν έτοιμα templates εμφάνισης ιστοσελίδων με έτοιμα κομμάτια HTML, CSS κτλ τα οποία στη συνέχεια μπορούν να μετατρέψουν και να δημιουργήσουν τις δικές τους

ιστοσελίδες από αυτά. Στην παρούσα εργασία προμηθευτήκαμε ένα δωρεάν HTML Template από το ακόλουθο website <http://all-free-download.com/free-website-templates/>. Ο περιορισμός που μας δίνει είναι ότι για τη χρήση του οφείλουμε να **All-free-download.com** περιλαμβάνουμε ένα link προς τη σελίδα αυτή.

Επίσης πολλές φορές στις web-εφαρμογές και στις mobile εφαρμογές χρειαζόμαστε κάποια εικονίδια για να σχεδιαστεί το περιβάλλον. Μπορούμε να προμηθευτούμε από το διαδίκτυο εικονίδια επί-πληρωμή ή δωρεάν με διάφορες άδειες. Στην παρούσα εργασία χρησιμοποιήθηκαν εικονίδια από το website



<https://www.iconfinder.com/> το οποίο διαθέτει και δωρεάν και επί-πληρωμή εικονίδια. Όσα εικονίδια χρησιμοποιήθηκαν

ήταν δωρεάν χωρίς την απαίτηση ύπαρξης backlink σε αυτά. [16, 17]

## 2.8 SmartPhone

Τα smartphones είναι κινητά τηλέφωνα με αυξημένες υπολογιστικές δυνατότητες και συνδεσιμότητα από τα απλά κινητά τηλέφωνα. Περιλαμβάνουν πολλά χαρακτηριστικά των υπολογιστών καθώς και άλλων προσωπικών συσκευών, όπως mp3 player, φωτογραφική μηχανή, GPS, ατζέντα κτλ. Τα περισσότερα πρόσφατα smartphones έχουν οθόνη αφής, σύνδεση στο internet μέσω WiFi ή δικτύων κινητής τηλεφωνίας, 3<sup>rd</sup> party εφαρμογές, διάφορους αισθητήρες κτλ.

Τα κυριότερα λειτουργικά συστήματα για smartphones είναι το **Google Android** το οποίο είναι εγκατεστημένο σε πάρα

πολλές συσκευές smartphones πολλών διαφορετικών κατασκευαστών, το **iOS** της Apple το οποίο το βρίσκουμε σε κινητά τηλέφωνα Apple iPhone και tablets Apple iPad, καθώς και το **Windows Phone** το οποίο παράγει η Microsoft. [2]





### 3. Ανάλυση τεχνολογιών και βιβλιοθηκών

#### 3.1 HTML

##### 3.1.1 Περιγραφή



Κάθε σελίδα που εμφανίζεται στο Internet είναι ένα αρχείο γραμμένο με τη γλώσσα HTML (HyperText Markup Language, Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου), που περιλαμβάνει το κείμενο της σελίδας, τη δομή της και τους συνδέσμους προς άλλα έγγραφα, εικόνες ή άλλα μέσα.

Ο φυλλομετρητής (Web browser) παίρνει τις πληροφορίες από τον web server, τις μορφοποιεί και τις εμφανίζει κατάλληλα για το σύστημά μας. Διαφορετικά προγράμματα φυλλομετρητή μπορεί να μορφοποιούν και να εμφανίζουν το ίδιο αρχείο με διαφορετικό τρόπο, ανάλογα με τις δυνατότητες του συστήματος στο οποίο

τρέχουν και τις επιλογές διαμόρφωσης του προγράμματος του φυλλομετρητή. Μια web σελίδα ή ιστοσελίδα (web page) είναι ένα μεμονωμένο στοιχείο μιας παρουσίασης για το Web και περιέχεται σ' ένα αρχείο στον δίσκο, το οποίο ανακτάται από έναν web server και μορφοποιείται μέσω ενός φυλλομετρητή.

Η αρχική σελίδα (home page) είναι η πρώτη ή κορυφαία σελίδα μιας παρουσίασης για το Web, είναι δηλαδή το σημείο εισόδου ή εκκίνησης για τις υπόλοιπες σελίδες της παρουσίασης και η πρώτη σελίδα που θα συναντήσουν οι αναγνώστες της παρουσίασής μας. Η αρχική σελίδα περιέχει συνήθως μια σύνοψη του περιεχομένου της παρουσίασης με τη μορφή ενός πίνακα περιεχομένων ή μιας ομάδας εικονιδίων.

Η HTML είναι το ακρωνύμιο των λέξεων **H**yper**T**ext **M**arkup **L**anguage, δηλ. Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου και βασίζεται στη γλώσσα SGML, Standard Generalized Markup Language, που είναι ένα πολύ μεγαλύτερο σύστημα επεξεργασίας εγγράφων. Η HTML ορίζει ένα σύνολο κοινών στυλ για τις Web σελίδες, όπως τίτλοι (titles), επικεφαλίδες (headings), παράγραφοι (paragraphs), λίστες (lists) και πίνακες (tables). Ορίζει επίσης στυλ χαρακτήρων, όπως η έντονη γραφή (boldface) και οι ενότητες κώδικα.

Κάθε στοιχείο έχει ένα όνομα και περιέχεται μέσα στα σύμβολα <>, που αποκαλούνται tags (ετικέτες). Όταν γράφουμε μια Web σελίδα με την HTML, στην ουσία δίνουμε τίτλους στα διάφορα στοιχεία της σελίδας με αυτά τα tags. Οι φυλλομετρητές, μαζί με τη δυνατότητά τους να ανακτούν σελίδες από το Web, λειτουργούν επίσης και σαν μορφοποιητές για την HTML. Όταν διαβάζουμε μια σελίδα γραμμένη με την HTML σ' έναν φυλλομετρητή, ο φυλλομετρητής διαβάζει (διερμηνεύει) τα tags της HTML και μορφοποιεί το κείμενο και τις εικόνες στην

οθόνη. Διαφορετικοί φυλλομετρητές, οι οποίοι τρέχουν σε διαφορετικούς υπολογιστές, μπορεί να αντιστοιχίζουν διαφορετικά στυλ σε κάθε στοιχείο μιας σελίδας. Αυτό σημαίνει ότι οι σελίδες που δημιουργούμε με την HTML μπορεί να δείχνουν διαφορετικές από σύστημα σε σύστημα και από φυλλομετρητή σε φυλλομετρητή. Δηλαδή, οι πραγματικές πληροφορίες και οι σύνδεσμοι που περιέχουν οι σελίδες μας θα είναι πάντα εκεί, αλλά η εμφάνιση των σελίδων στην οθόνη θα είναι διαφορετική. <sup>[2, 6, 21]</sup>

### 3.1.2 Δομή μιας HTML σελίδας

Οι σελίδες που γράφουμε με την HTML είναι απλά αρχεία, που σημαίνει ότι δεν περιέχουν πληροφορίες για κάποιο λειτουργικό σύστημα ή πρόγραμμα, αλλά μπορούν να διαβαστούν από οποιονδήποτε συντάκτη υποστηρίζει απλό κείμενο. Τα αρχεία της HTML περιέχουν τα ακόλουθα :

- Το κείμενο της σελίδας.
- Τα tags (ετικέτες) της HTML, τα οποία υποδεικνύουν τα στοιχεία, τη δομή και τη μορφοποίηση των σελίδων, καθώς επίσης και τους συνδέσμους υπερκειμένου προς άλλες σελίδες ή προς αρχεία άλλων μορφών (πχ πολυμέσα).

Τα περισσότερα tags έχουν την εξής μορφή :

`<ΌνομαTag> Κείμενο ... </ΌνομαTag>`

Τα tags της HTML έχουν γενικά μια ετικέτα αρχής (ανοίγματος) και μια τέλους (κλεισίματος), τα οποία περικλείουν το κείμενο που επηρεάζουν. Το tag αρχής ενεργοποιεί μια λειτουργία ή ένα χαρακτηριστικό, όπως για παράδειγμα την έντονη γραφή, ενώ το tag τέλους την απενεργοποιεί. Τα tags τέλους έχουν το ίδιο όνομα με τα tag αρχής αλλά με πρόθεμα τον χαρακτήρα ' / ' .

Δεν αποτελούν ζευγάρι όλα τα tags της HTML, καθώς ορισμένα είναι «μονομελή», ενώ άλλα περιέχουν επιπλέον πληροφορίες και κείμενο ανάμεσα στα σύμβολα < και > . Επίσης, όλα τα tags στην HTML δεν κάνουν διάκριση μεταξύ κεφαλαίων και πεζών γραμμάτων, δηλαδή μπορούμε να γράφουμε είτε με κεφαλαία είτε με μικρά, ακόμα και με οποιονδήποτε συνδυασμό τους.

Η HTML χρησιμοποιεί τρία tags για την περιγραφή της συνολικής δομής μιας σελίδας, τα οποία περιέχουν ορισμένες απλές πληροφορίες κεφαλίδας. Αυτά προσδιορίζουν τη σελίδα μας στους φυλλομετρητές και παρέχουν απλές πληροφορίες για αυτήν, όπως τον τίτλο ή το συγγραφέα της, πριν από τη φόρτωση ολόκληρης της σελίδας.

Το tag `<HTML>` : η πρώτη ετικέτα που ελέγχει τη δομή μιας σελίδας γραμμένης σε κώδικα HTML και υποδεικνύει ότι το περιεχόμενο του αρχείου περιέχει κώδικα γραμμένο σε αυτή τη γλώσσα. Όλο το κείμενο και οι εντολές σε μια HTML σελίδα θα πρέπει να τοποθετούνται ανάμεσα σε tags αρχής και τέλους `<HTML>` , ως εξής :

`<HTML>`

.... Περιεχόμενα σελίδας ....

`<HTML>`

Το tag <HEAD> : προσδιορίζει ότι οι γραμμές που περιέχονται ανάμεσα στην αρχή και στο τέλος του είναι ο πρόλογος για το υπόλοιπο του αρχείου. Σε αυτή την ενότητα συνήθως τοποθετούμε τον τίτλο της σελίδας καθώς και την εισαγωγή εξωτερικών βιβλιοθηκών και css αρχείων.

```
<HTML>
  <HEAD>
    <TITLE> Τίτλος </TITLE>
  </HEAD>
  ....
</HTML>
```

Το tag <BODY> : το υπόλοιπο της HTML σελίδας, δηλαδή όλο το κείμενο και οποιοδήποτε άλλο περιεχόμενο, όπως σύνδεσμοι, εικόνες κτλ περικλείεται σε μια τέτοια ετικέτα.

```
<HTML>
  <HEAD>
    <TITLE> Τίτλος </TITLE>
  </HEAD>
  <BODY>
    .....Περιεχόμενα.....
  </BODY>
</HTML>
```

Όλες οι ετικέτες της HTML σχηματίζουν ξεχωριστές, ένθετες ενότητες κειμένου και θα πρέπει να δίνεται ιδιαίτερη προσοχή ώστε να μην υπάρχει επικάλυψη μεταξύ των tags.

Το tag <TITLE> : Κάθε HTML σελίδα χρειάζεται έναν τίτλο, ο οποίος θα υποδεικνύει το περιεχόμενό της, και θα εμφανίζεται στη γραμμή τίτλου των δημοφιλέστερων φυλλομετρητών. Οι ετικέτες <TITLE> τοποθετούνται πάντα μέσα στο ζευγάρι των tags <HEAD> όπως φαίνεται στο προηγούμενο σχήμα.

Τα tags επικεφαλίδων : Οι επικεφαλίδες χρησιμοποιούνται για το διαχωρισμό των ενοτήτων κειμένου, όπως ακριβώς και σε ένα βιβλίο. Έχουν τη μορφή

```
<h1> Τίτλος κειμένου </h1>
```

και ορίζονται με ετικέτες από H1 έως H6. Οι επικεφαλίδες δεν αριθμούνται όταν εμφανίζονται στην οθόνη, αλλά έχουν κάποιο χαρακτηριστικό που τις ξεχωρίζει από το υπόλοιπο κείμενο, όπως μεγαλύτερο μέγεθος, εντονότερο κείμενο ή υπογράμμιση. Συνήθως το μέγεθος του κειμένου ελαττώνεται πηγαίνοντας από το H1 στο H6.

Τα tags παραγράφων : είναι τα <p> και </p> και η αλλαγή παραγράφου σημαίνει το ξεκίνημα μιας νέας γραμμής και επιπλέον κατακόρυφη απόσταση

μεταξύ των παραγράφων. Το `<br>` είναι κενό στοιχείο, δεν έχει περιεχόμενο, ούτε χρειάζεται ετικέτα τερματισμού.

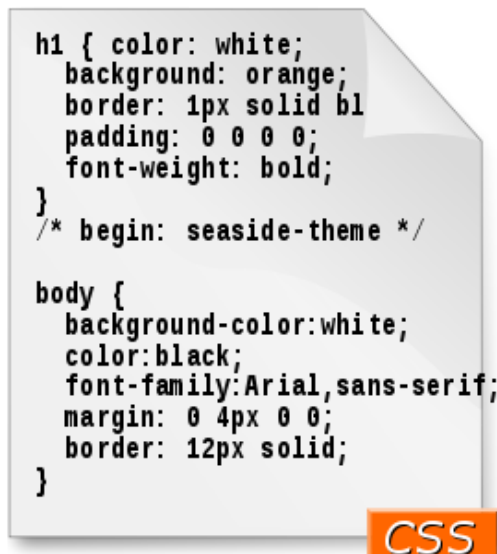
Παρουσιάσαμε εν συντομία κάποια βασικά στοιχεία της γλώσσας HTML. Επιπλέον αισθητικές παρεμβάσεις στη σελίδα μας μπορούμε να κάνουμε χρησιμοποιώντας κάποια tags, όπως το `<style>` ή χρήση CSS (Cascade Style Sheets) αρχείων. <sup>[2, 6, 21]</sup>

## 3.2 CSS (Cascading Style Sheets)

### 3.2.1 Περιγραφή

```
h1 { color: white;
background: orange;
border: 1px solid bl
padding: 0 0 0 0;
font-weight: bold;
}
/* begin: seaside-theme */

body {
background-color:white;
color:black;
font-family:Arial,sans-serif;
margin: 0 4px 0 0;
border: 12px solid;
}
```



Όπως είδαμε και σε προηγούμενη παράγραφο, η γλώσσα προγραμματισμού ιστοσελίδων και εφαρμογών HTML, χρησιμοποιείται για τη δημιουργία και την περιγραφή στοιχείων σε μια ιστοσελίδα. Όταν ο browser διαβάζει ένα κείμενο σε μια σελίδα, λαμβάνει υπόψη του τα html tags, τα οποία χρησιμοποιούνται για να περιγράψουν αντικείμενα. Ότι βρίσκεται ανάμεσα σε tags, παίρνει την αντίστοιχες ιδιότητες. Μέσω των επιλογέων (selectors) μπορούμε να επικεντρωθούμε σε συγκεκριμένα στοιχεία στην HTML σελίδα και να τους διαμορφώσουμε στυλ. <sup>[2, 6, 21]</sup>

### 3.2.2 Σύνταξη

Η σύνταξη μιας CSS εντολής είναι :



Επειδή τα tags στην HTML είναι περιορισμένα, τα CSS μας δίνουν τη δυνατότητα να δηλώσουμε τα δικά μας μοναδικά ή μη στοιχεία, και να τους διαμορφώσουμε το στυλ. Αυτό επιτυγχάνεται με τη χρήση των attributes των tags της HTML, "id" και "class". Τα id και classes είναι διαφορετικά στοιχεία και επιτρέπεται να έχουν το ίδιο όνομα. Τα ID δηλώνουν μοναδικά στοιχεία και είναι ορθότερο να μην εμφανίζουμε το ίδιο στοιχεία πάνω από μια φορά στη σελίδα μας. Αντίθετα, μπορούμε να έχουμε απεριόριστα αντίγραφα κλάσεων μέσα στη σελίδα μας. Το cascading αναφέρεται στο φαινόμενο του καταρράκτη, και ουσιαστικά υποδεικνύει την κληρονομικότητα που έχουν τα στοιχεία των CSS.

Με τα CSS μπορούμε να ορίσουμε πολλά χαρακτηριστικά της εμφάνισης της ιστοσελίδας μας. Για παράδειγμα μπορούμε να ορίσουμε το μέγεθος της γραματοσειράς, το χρώμα της, να δηλώσουμε bold, underlined, italic κτλ χαρακτηριστικά τυπογραφίας. Μπορούμε να αλλάξουμε το background σε κάποιο element της HTML μας. <sup>[2, 6, 21]</sup>

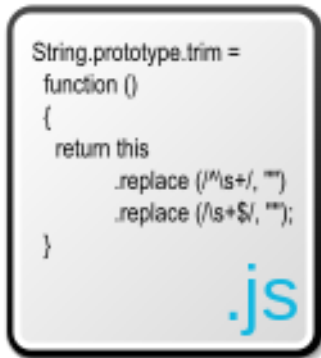
### 3.2.3 Κληρονομικότητα Στυλ

Όπως αναφέραμε προηγουμένως, στη CSS έχουμε κληρονομικότητα μεταξύ των στυλ, όπου ένα HTML element “παιδί” κάποιου άλλου HTML element κληρονομεί το στυλ του “πατέρα” element κτλ. Οι κανόνες που ορίζουν ποιο element κληρονομεί τι και με ποια σειρά υπερισχύει κάποιο στυλ σε σχέση με κάποιο άλλο αντίστοιχο είναι η ακόλουθη (από τα ισχυρότερα στα λιγότερο ισχυρά):

<b>Προτεραιότητα</b>	<b>CSS τύπος</b>	<b>Περιγραφή</b>
1	User defined	Το CSS που ορίζει απευθείας ο χρήστης
2	Inline	Ένα στυλ ορισμένο σε ένα HTML element μέσω του “style” property
3	Media Type	Ο ορισμός ενός property σε όλα τα είδη media, εκτός αν ορίζεται κάποιο συγκεκριμένο CSS
4	Importance	Το ‘!important’ διαγράφει τα προηγούμενα είδη προτεραιότητας
5	Selector specificity	Ένας συγκεκριμένος selector υπερισχύει της γενικού ορισμού
6	Rule order	Ο πιο πρόσφατος κανόνας είναι αυτός που υπερισχύει
7	Parent inheritance	Αν κάποιο property δεν είναι ορισμένο, κληρονομείται από το element “πατέρα”
8	CSS property definition in HTML document	Ένας CSS κανόνας ή CSS inline style διαγράφει την προεπιλεγμένη τιμή του browser
9	Browser default	Η χαμηλότερη προτεραιότητα: Η αρχική τιμή του browser ορίζεται από τα specifications του W3C

## 3.3 JavaScript

### 3.3.1 Περιγραφή



Η Javascript είναι μια δυναμική γλώσσα προγραμματισμού η οποία χρησιμοποιείται κυρίως στους web browsers. Η υλοποίηση της Javascript σε έναν browser επιτρέπει την εκτέλεση client-side κώδικα (κώδικα δηλαδή που τρέχει στον υπολογιστή του χρήστη και όχι στο server) για αλληλεπίδραση με το χρήστη, διαχείριση του browser, ασύγχρονη επικοινωνία με το server και δυναμική αλλαγή της HTML που παρουσιάζεται στο χρήστη.

Πρόκειται για μια “dynamic-typing” γλώσσα, όπου δηλαδή οι μεταβλητές μπορούν να έχουν οποιοδήποτε τύπο (είτε strings, είτε αριθμοί κτλ) και να αλλάζουν δυναμικά χωρίς να ορίζεται αυστηρά κάποιος συγκεκριμένος τύπος τους.

Τα τελευταία χρόνια η Javascript χρησιμοποιείται και εκτός των ιστοσελίδων, όπως σε αρχεία PDF, widgets ακόμα και σε server-site εφαρμογές μέσω frameworks όπως το Node.js και το Angular.js. <sup>[2]</sup>

### 3.3.2 Σύνταξη

Οι μεταβλητές στη Javascript ορίζονται χρησιμοποιώντας τη λέξη κλειδί “var”, για παράδειγμα:

```
var hMetablitiMou; //Ορίζει μια μεταβλητή με το όνομα “ hMetablitiMou”  
χωρίς αρχική τιμή
```

```
var myVariable = 5; //Ορίζει μια μεταβλητή με το όνομα “ myVariable ” και  
αρχική τιμή τον αριθμό 5
```

Στη Javascript μπορούμε να οργανώσουμε τον κώδικά μας σε functions, για παράδειγμα η ακόλουθη που παίρνει μια μεταβλητή και της προσθέτει ένα string:

```
function myFunction ( var myVar ) {  
    myVar = myVar + “myString”;  
    return myVar;  
}
```

### 3.3.3 Χρήση σε ιστοσελίδες

Η πιο συνηθισμένη χρήση της Javascript είναι η χρήση της σε ιστοσελίδες για εκτέλεση clien-side κώδικα. Διάφορα JS scripts περιλαμβάνονται στις ιστοσελίδες για να επηρεάζουν το DOM της σελίδας. Μερικά παραδείγματα είναι τα ακόλουθα:

- Φόρτωση νέων σελίδων ή μερών αυτών καθώς και υποβολή δεδομένων μέσω ασύγχρονων ( AJAX ) requests χωρίς να απαιτείται η επαναφόρτιση της σελίδας.

- Κινήσεις στοιχείων της σελίδας
- Διαδραστικό περιεχόμενο
- Validation της εισόδου το χρήστη πριν αποσταλεί στο χρήστη
- Αποστολή στοιχείων χρήσης του χρήστη πάνω στις διάφορες ιστοσελίδες

Το ρόλο της ερμηνείας του κώδικα Javascript τον αναλαμβάνει ο εκάστοτε Web Browser. Η Javascript μπορεί να τρέχει τοπικά στον υπολογιστή του χρήστη δίνοντας τη δυνατότητα στη σελίδα να ανταποκρίνεται γρήγορα στις πράξεις του χρήστη χωρίς να επιβαρύνει το server. <sup>[2]</sup>

### 3.3.4 Χρήση εκτός ιστοσελίδων

Το τελευταίο χρονικό διάστημα η Javascript αποκτά χρήση και εκτός ιστοσελίδων. Πολλές desktop εφαρμογές χρησιμοποιούν την εν λόγω γλώσσα για να τρέχουν διάφορα scripts για τις εσωτερικές τους λειτουργίες. Επίσης διάφορα developing frameworks κυκλοφορούν τα οποία βασίζονται στη javascript για τη δημιουργία εφαρμογών, όπως πχ. το Node.js και το Angular.js. Σε αυτή την εργασία η Javascript χρησιμοποιήθηκε μόνο εντός του web browser για χρήση στην ιστοσελίδα, οπότε δε θα γίνει περαιτέρω αναφορά στη χρήση της εκτός των ιστοσελίδων. <sup>[2]</sup>

## 3.4 JQuery

### 3.4.1 Περιγραφή



Το jQuery είναι μια μικρή σε μέγεθος, απλή, γρήγορη και περιεκτική βιβλιοθήκη Javascript ανοιχτού κώδικα και πρωτοεμφανίστηκε το 2006. Συνδυάζοντας την απλότητα, την ευελιξία και την επεκτασιμότητα, άλλαξε τον τρόπο με τον οποίο οι προγραμματιστές γράφουν Javascript.

Αυτή η βιβλιοθήκη εντολών μας παρέχει δυνατότητες επιλογής και διαμόρφωσης HTML στοιχείων, διαμόρφωσης CSS στοιχείων, διεργασιών HTML γεγονότων, εφέ και animations, χρήσης AJAX και πλήθώρας άλλων εφαρμογών.

Σε γενικές γραμμές το jQuery έχει αντικαταστήσει τον κώδικα παραδοσιακής Javascript. Οι περισσότεροι developers χρησιμοποιούν απευθείας αυτή τη βιβλιοθήκη καθώς προσφέρει πολλές επιπλέον δυνατότητες σε σχέση με την παραδοσιακή Javascript. Τα κυριότερα πλεονεκτήματά της είναι τα εξής:



- **Διαχωρισμός της Javascript και της HTML.** Αντί να γίνεται χρήση των HTML attributes για να καλούνται οι javascript functions, το jQuery επιτρέπει όλες τις functions για διαχείριση γεγονότων ( event handling ) να γίνονται πλήρως σε Javascript έτσι ώστε να μη χρειάζονται να μπλέκονται με την HTML.
- **Συντομία και σαφήνεια.** Το jQuery παρέχει διάφορα είδη σύνταξης ώστε να είναι πιο σύντομο και πιο σαφές, πχ. chaining εφέ και πράξεις, συντομεύσεις μεθόδων.
- **Εξαλείφει τις ασυμβατότητες μεταξύ των browsers.** Οι διάφοροι μηχανισμοί Javascript στους browsers μπορεί να έχουν διαφορές και Javascript κώδικα που λειτουργεί σε έναν browser να μη λειτουργεί σε κάποιον άλλο. Το jquery διαχειρίζεται όλες αυτές τις ασυμβατότητες και παρέχει ένα συνεπές interface που λειτουργεί σε όλους τους browsers που υποστηρίζει.
- **Επεκτάσιμο.** Το jQuery δίνει τη δυνατότητα επέκτασης του framework. Νέα γεγονότα, elements και μέθοδοι μπορούν να προστεθούν και να χρησιμοποιηθούν σε ένα plugin. <sup>[2, 10]</sup>

### 3.4.2 Χαρακτηριστικά

Το jQuery περιλαμβάνει τα εξής χαρακτηριστικά:

- Επιλογή DOM elements με χρήση της open source μηχανής Sizzle
- Μετατροπές στο DOM
- Διαχείριση του DOM βασισμένο σε CSS selectors που χρησιμοποιεί το όνομα του element και τα attributes id και class ως κριτήρια επιλογής
- Events (γεγονότα)
- Εφέ και animations
- AJAX ( asynchronous requests )
- JSON parsing
- Επεκτασιμότητα μέσω plugins
- Utilities ( όπως πληροφορίες του user-agent και εντοπισμός features )
- Μέθοδοι συμβατότητας οι οποίες είναι παρούσες στους νέους browsers αλλά χρειάζονται fall-backs σε παλαιότερους
- Υποστήριξη πολλαπλών browsers

### 3.4.3 Χρήση

Για να ενσωματώσουμε την βιβλιοθήκη σε ένα html αρχείο, προσθέτουμε στο <head> μέρος της σελίδας μας τον εξής κώδικα:

```
<script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
```

Η σύνταξη της είναι πολύ απλή :  $\$(\text{επιλογή element}).\text{function}()$ ;

Το  $\$$  είναι το σύμβολο που υποδηλώνει την jQuery.

Ο (επιλογέας) είναι το ερώτημα(query) για το HTML στοιχείο που θέλουμε.

Η function είναι η συνάρτηση που θα εκτελεστεί στο επιλεγμένο αντικείμενο.

Π.χ.  $\$(\#image1).hide()$  – Κρύβει το HTML element με id “image1”.



Κάποια από τα βασικά γεγονότα (events) της jQuery είναι :

`$(document).ready(function)` - Έλεγχος ολοκλήρωσης φόρτωσης μιας σελίδας  
`$(selector).click(function)` - Έλεγχος για το κλικ σε ένα στοιχείο  
`$(selector).dblclick(function)` - Έλεγχος για το διπλό κλικ σε ένα στοιχείο  
`$(selector).focus(function)` - Έλεγχος εστίασης σε ένα στοιχείο  
`$(selector).mouseover(function)` - Έλεγχος για το hover του ποντικιού σε ένα στοιχείο

Κάποια παραδείγματα εφέ της jQuery είναι :

`$(selector).hide()` – Απόκρυψη επιλεγμένων στοιχείων  
`$(selector).show()` – Εμφάνιση επιλεγμένων στοιχείων  
`$(selector).toggle()` – Εμφάνιση/Απόκρυψη επιλεγμένων στοιχείων  
`$(selector).slideDown()` – Slide down  
`$(selector).slideUp()` – Slide up  
`$(selector).slideToggle()` – Slide up/down  
`$(selector).fadeIn()` – Εμφάνιση (fade in)  
`$(selector).fadeOut()` – Απόκρυψη (fade out)  
`$(selector).fadeTo()` – Απόκρυψη σε ορισμένο opacity  
`$(selector).animation()` – Εφαρμογή animation σε επιλεγμένα στοιχεία

Συνδυάζοντας τις δυνατότητες της jQuery με τα CSS μπορούμε να δημιουργήσουμε ένα πολύ καλαίσθητο περιβάλλον, λειτουργικό και με όμορφα animations. <sup>[2, 10]</sup>

## 3.5 SQL (Structured Query Language)

### 3.5.1 Περιγραφή

Η SQL (Structured Query Language) είναι μια γλώσσα υπολογιστών, που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS), η οποία αρχικά βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα.

Η SQL αναπτύχθηκε στην IBM στην αρχή της δεκαετίας του 1970. Αυτή η πρώτη έκδοση που αρχικά ονομαζόταν SEQUEL, είχε ως σκοπό να χειριστεί και να ανακτήσει τα στοιχεία που αποθηκεύτηκαν στο πρώτο RDBMS της IBM, το System R. Την ίδια περίοδο αναπτύχθηκε στο MIT το πρώτο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) και η INGRES στο Πανεπιστήμιο Berkeley. Στα τέλη της δεκαετίας η Relational Software εισήγαγε την πρώτη διαθέσιμη εμπορικά εφαρμογή του SQL, λίγες εβδομάδες νωρίτερα από την IBM. <sup>[2, 8]</sup>

### 3.5.2 Γλωσσικά στοιχεία

Η γλώσσα SQL υποδιαιρείται σε διάφορα γλωσσικά στοιχεία που περιλαμβάνουν :

- **Clauses**, προαιρετικές σε ορισμένες περιπτώσεις, αλλά απαραίτητα συστατικά όλων των δηλώσεων και ερωτήσεων.
- **Expressions**, που παράγουν είτε κλιμακωτές τιμές, είτε πίνακες αποτελούμενους από στήλες και σειρές δεδομένων.
- **Predicates**, που διευκρινίζουν τους όρους που μπορούν να αξιολογηθούν σαν σωστό ή λάθος.
- **Queries**, που ανακτούν τα στοιχεία βασιζόμενες σε ειδικά κριτήρια.
- **Statements**, που μπορούν να επιδρούν στα σχήματα και τα στοιχεία, ή να ελέγξουν τη ροή του προγράμματος και τις συνδέσεις με άλλα προγράμματα.

Οι εκφράσεις σε SQL περιλαμβάνουν και το χαρακτήρα τερματισμού « ; ».

Το κενό γενικά αγνοείται στις statements και τις queries της SQL, αλλά είναι απαραίτητο για να ξεχωρίζει τα statements, όπως και στην κανονική γραφή κειμένου. <sup>[2,8]</sup>

### 3.5.3 Operators

<b>Operator</b>	<b>Περιγραφή</b>
=	Ισούται με
<>	Δεν ισούται με (τα περισσότερα DBMS επίσης δέχονται το != αντί για το <>)
>	Μεγαλύτερο από
<	Μικρότερο από
>=	Μεγαλύτερο ή ίσο

<=	Μικρότερο ή ίσο
BETWEEN	Μεταξύ ενός καθορισμένου εύρους
LIKE	Ταιριάζει με ένα πρότυπο χαρακτήρα
IN	Ίσο με μια από τις πολλαπλές δυνατές τιμές
IS ή IS NOT	Σύγκριση με το null (κενό δεδομένων)
IS NOT DISTINCT FROM	Ίσο με τιμή ή και τα 2 είναι null (κενό δεδομένων)
AS	Χρησιμοποιείται για αλλαγή του ονόματος πεδίου κατά την εμφάνιση

Κάποιες από τις σημαντικότερες SQL εντολές είναι :

- **SELECT** – επιλογή δεδομένων από τη βάση
- **UPDATE** – ενημέρωση δεδομένων στη βάση
- **DELETE** – διαγραφή δεδομένων από τη βάση
- **INSERT INTO** – εισαγωγή νέων δεδομένων στη βάση
- **CREATE DATABASE** – δημιουργία νέας βάσης
- **ALTER DATABASE** – τροποποίηση της βάσης
- **CREATE TABLE** – δημιουργία νέου πίνακα
- **ALTER TABLE** – τροποποίηση έναν πίνακα
- **DROP TABLE** – διαγραφή ενός πίνακα
- **CREATE COLUMN** – δημιουργία κολόνας πίνακα
- **ALTER COLUMN** – τροποποίηση κολόνας πίνακα
- **DROP COLUMN** – διαγραφή κολόνας πίνακα
- **CREATE INDEX** – δημιουργία δείκτη (κλειδί αναζήτησης)
- **DROP INDEX** – διαγραφή ενός δείκτη

Για τον καλύτερο σχεδιασμό και βελτιστοποίηση μιας βάσης δεδομένων συχνά γίνεται χρήση και των ακόλουθων εντολών :

- **FROM** – υποδεικνύει από ποιο πίνακα θα τραβήξουμε τα δεδομένα που μας ενδιαφέρουν
- **JOIN** – ορίζει τους κανόνες για την ένωση πινάκων.
- **WHERE** – θέτει όρους σύγκρισης στα δεδομένα που επιστρέφονται από τη βάση, θέτει φίλτρα αληθείας.
- **GROUP BY** – ομαδοποιεί τα δεδομένα σύμφωνα με ορισμένα κριτήρια
- **ORDER BY** – ταξινομεί τα δεδομένα σύμφωνα με ορισμένα κριτήρια

Άλλες εντολές για τη μεταχείριση των δεδομένων, διαμόρφωση των πινάκων και τις συναλλαγές μεταξύ τους, είναι πχ η **INSERT, UPDATE, DELETE, MERGE, START – SAVE TRANSACTION** κτλ.

#### 3.5.4 Τύποι Δεδομένων

- Κάθε στήλη πίνακα SQL δηλώνει τι τύπο δεδομένων περιέχει. Η ANSI SQL περιλαμβάνει τους παρακάτω τύπους δεδομένων.
  - String Χαρακτήρων – Χαρακτήρες, κενά, μέγιστου μήκους n, μπορεί να περιέχει και εθνικούς χαρακτήρες.
  - Bit Strings
  - Αριθμούς – Ακέραιους, πραγματικούς, κινητής υποδιαστολής, διπλής υποδιαστολής κτλ
  - Ημερομηνία και Χρόνο
    - DATE : τιμές για ημερομηνία (πχ 2014-2-21)
    - TIME : τιμές για χρόνο, ώρα (πχ 15:51:36)
    - TIME WITH TIMEZONE ή TIMEZ : ίδιο με το TIME, αλλά περιέχει πληροφορίες και για τη ζώνη ώρας
    - TIMESTAMP : DATE και TIME σε μια μεταβλητή (πχ 2014-2-21 15:51:36)
    - TIMESTAMP WITH TIMEZONE ή TIMESTAMPZ : ίδιο με το timestamp περιέχει και πληροφορίες για τη ζώνη ώρας.

Η SQL παρέχει διάφορες λειτουργίες για τη δημιουργία μιας μεταβλητής ημερομηνίας και ώρας, καθώς και να το διαχωρισμό στα επιμέρους στοιχεία της, όπως λεπτά, δευτερόλεπτα κτλ. <sup>[2, 8]</sup>

## 3.6 PHP

### 3.6.1 Περιγραφή



Η PHP είναι μια server-side scripting γλώσσα προγραμματισμού με κύριο σκοπό τη δημιουργία web-εφαρμογών. Επίσημο website για την PHP είναι το <http://php.net/>.

Στον προγραμματισμό με PHP δίνεται η δυνατότητα ανάμειξης κώδικα PHP με κλασική HTML καθώς και διάφορα άλλα frameworks. Ο PHP κώδικας επεξεργάζεται από έναν PHP διερμηνέα (interpreter) ο οποίος συνήθως υλοποιείται από κάποιο module ενός web-server ή από κάποιο CGI εκτελέσιμο (Common Gateway Interface). Αφού ο κώδικας ερμηνευτεί και εκτελεστεί, ο web-server στέλνει το αποτέλεσμα στον client (συνήθως με τη μορφή μιας ιστοσελίδας, εικόνας ή άλλων δεδομένων). Στην περίπτωση μας ο PHP interpreter είναι module του Apache HTTP (“httpd”) ο οποίος διαχειρίζεται από το panel του XAMPP. <sup>[2, 3, 5]</sup>

### 3.6.2 Σύνταξη

Παράδειγμα ενός “Hello World” προγράμματος σε PHP.

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php echo '<p>Hello World</p>'; ?>
  </body>
</html>
```

Παρατηρούμε τον τρόπο με τον οποίο περιλαμβάνεται ο κώδικας PHP μέσα στην κλασική HTML, ανοίγοντας με “<?php” και κλείνοντας με “>”. Η εντολή “echo” χρησιμοποιείται για να τυπωθεί κάποιο string ή κάποια μεταβλητή στην έξοδο.

Στη συνέχεια παρουσιάζουμε ένα παράδειγμα μιας PHP function. Οι συναρτήσεις στην PHP όπως και σε όλες τις γλώσσες προγραμματισμού χρησιμοποιούνται για την οργάνωση κοινών κομματιών κώδικα τα οποία χρησιμοποιούνται σε πολλά σημεία στην εφαρμογή μας. Με τη χρήση των συναρτήσεων μπορούμε να οργανώσουμε τον κώδικά μας, να τον μειώσουμε σε μέγεθος κάνοντας ευκολότερη τη συντήρησή του και τη βελτίωσή του. Η PHP περιλαμβάνει πληθώρα συναρτήσεων και βιβλιοθηκών για τις πιο συνήθεις λειτουργίες. Φυσικά ο κάθε προγραμματιστής έχει τη δυνατότητα να φτιάξει τις δικές του συναρτήσεις ανάλογα με τις απαιτήσεις του.

```
public static function isEmptyOrNull($question){
    return (!isset($question) || trim($question)=='');
}
```

Η παραπάνω συνάρτηση χρησιμοποιείται για να ελέγξουμε αν ένα string είναι κενό.

Η PHP μετά την έκδοση 4 μπορεί να χρησιμοποιηθεί και για Object Oriented προγραμματισμό. Στην παρακάτω εικόνα βλέπουμε ένα παράδειγμα μιας PHP Class η οποία στην εφαρμογή μας περιλαμβάνει κάποιες γενικές functions για διάβασμα παραμέτρων από το εκάστοτε HTTP Request. <sup>[2, 3, 5]</sup>

```

<?php

class MiscUtil {

    public static function getBoolean($input) {

        $result = false;
        if ( $input === 1 || $input === "true" || $input === "t
            $result = true;
        }

        return $result;
    }

    public static function getBooleanMessage ($input) {

        $result = "Όχι";
        if ( getBoolean($input) === true ) {
            $result = "Ναί";
        }
        return $result;
    }

    public static function test_input($data) {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        $data = mysql_real_escape_string($data);
        return $data;
    }
}
?>

```

### 3.6.3 License

Η PHP είναι δωρεάν λογισμικό και διανέμεται με την [PHP License](#) η οποία ορίζει ότι "Τα προϊόντα που προέρχονται από αυτό το λογισμικό δεν μπορούν να ονομάζονται "PHP", ούτε το "PHP" μπορεί να εμφανίζεται στο όνομά τους, χωρίς προηγούμενη γραπτή άδεια από group@php.net. Μπορείτε να δείχνετε ότι το λογισμικό σας λειτουργεί σε συνδυασμό με την PHP λέγοντας "Foo για την PHP" αντί να το καλείτε "PHP Foo" ή "phpfoo" ".

## 3.7 MySQL

### 3.7.1 Περιγραφή



Η MySQL είναι η δεύτερη πιο διαδεδομένη (RDBMS) βάση δεδομένων παγκοσμίως. Ακολουθεί την SQLite η οποία είναι εγκατεστημένη σε κάθε κινητό τηλέφωνο το οποίο τρέχει iOS ή Android, κερδίζοντας έτσι τεράστιο αριθμό χρηστών σε σχέση με τη MySQL και τις υπόλοιπες βάσεις δεδομένων. Πρόκειται για πολύ δημοφιλή

επιλογή για χρήση σε web εφαρμογές καθώς προσφέρει πάρα πολλές δυνατότητες, είναι ταχύτατη και δωρεάν. Περιέχεται στο πακέτο “LAMP” δωρεάν εφαρμογών. Πολλές γνωστές και ευρέως χρησιμοποιούμενες εφαρμογές είναι βασισμένες στη MySQL, όπως για παράδειγμα το Joomla, το Wordpress, το Drupal, το phpBB, καθώς και πολλά μεγάλα web-sites όπως το Google ( όχι στις αναζητήσεις ), το Facebook, το Twitter, το Youtube κ.α.

Τα αρχικά “SQL” που περιλαμβάνονται στο όνομα της βάσης αντιστοιχούν στη γλώσσα για βάσεις δεδομένων “[Structured Query Language](#)”. Κατά την εγκατάστασή της δεν περιλαμβάνεται κάποιο γραφικό περιβάλλον, υπάρχουν όμως αρκετά 3<sup>rd</sup> Party όπως το “MySQL Workbench” και το “PHPMysqlAdmin”, το οποίο τελευταίο χρησιμοποιήθηκε για το σχεδιασμό και τη διαχείριση της βάσης δεδομένων αυτής της εργασίας.

Ένα πολύ σημαντικό πλεονέκτημα της MySQL είναι ότι μπορεί να τρέξει στα περισσότερα περιβάλλοντα εργασίας, όπως Windows, Linux, Mac OS, Solaris κ.α. δίνοντας τη δυνατότητα στον προγραμματιστή να τη χρησιμοποιήσει σε διαφορετικές πλατφόρμες χωρίς να χρειάζεται να γράψει επιπλέον κώδικα. Για την εργασία αυτή η MySQL χρησιμοποιήθηκε σε Mac OS και σε Windows άκρως επιτυχημένα.

Επίσημο website της είναι το <http://www.mysql.com/> όπου μπορούμε να βρούμε πληροφορίες, documentation καθώς και να κατεβάσουμε το εκτελέσιμό της. <sup>[2,8]</sup>

### 3.7.2 Σύνταξη

Η σύνταξη της MySQL είναι η ίδια με τη σύνταξη της γλώσσας SQL και περιλαμβάνει επιπλέον κάποιες συναρτήσεις για συνήθειες πράξεις.

Παραδείγματα χρήσης της MySQL:

- Αναζήτηση χρήστη:

```
SELECT * FROM users u WHERE u.username = 'panos';
```

- Εισαγωγή χρήστη:

```
INSERT INTO users(`id`, `username`, `password`, `name`, `surname`, `email`, `role`)
VALUES ([value-1], [value-2], [value-3], [value-4], [value-5], [value-6], [value-7]);
```

- Επεξεργασία χρήστη:

```
UPDATE users u SET `id`=[value-1], `username`=[value-2], `password`=[value-3],
`name`=[value-4], `surname`=[value-5], `email`=[value-6], `role`=[value-7] WHERE u.id = 1
```

- Διαγραφή χρήστη:

```
DELETE FROM users u WHERE u.id = 1
```

### 3.7.3 License

Η χρήση της MySQL είναι δωρεάν κάτω από την “GNU General Public License v2” (GNU GPL ή GPL). Ο προγραμματιστής έχει το δικαίωμα να χρησιμοποιήσει, να μελετήσει, να αντιγράψει (διανείμει) και να επεξεργαστεί αυτό το software.

### 3.7.4 Χρήση με PHP

Στο documentation της MySQL, ο όρος “connector” αναφέρεται στο κομμάτι λογισμικού το οποίο επιτρέπει σε μια εφαρμογή να συνδεθεί σε μία βάση δεδομένων MySQL. Η MySQL παρέχει connectors για πολλές γλώσσες προγραμματισμού συμπεριλαμβανομένης της PHP.

Αν μια PHP εφαρμογή χρειάζεται να επικοινωνεί με μία βάση δεδομένων, ο προγραμματιστής θα πρέπει να γράψει κώδικα για να καλύψει τις περιπτώσεις σύνδεσης στη βάση, καθώς και εκτέλεσης ερωτημάτων και functions σε αυτή. Συνήθως αυτό το λογισμικό βρίσκεται σε βιβλιοθήκες και παρέχεται στους προγραμματιστές και είναι γενικά γνωστό ως “connector” καθώς επιτρέπει τη σύνδεση με τη βάση δεδομένων.

Ο “driver” είναι ένα κομμάτι λογισμικού σχεδιασμένο για να επικοινωνεί με κάποιο συγκεκριμένο τύπο βάσης δεδομένων. Ο “driver” μπορεί επίσης να καλεί βιβλιοθήκες, όπως τη MySQL Client Library ή το MySQL Native Driver. Αυτές οι βιβλιοθήκες υλοποιούν ένα low-level πρωτόκολλο το οποίο χρησιμοποιείται για την επικοινωνία με τη MySQL.

Πολλές φορές οι όροι “connector” και “driver” χρησιμοποιούνται αντίστροφα το οποίο μπερδεύει. Στην περίπτωση της MySQL, “driver” θεωρείται το λογισμικό που παρέχει το κομμάτι του connector που είναι συγκεκριμένο για αυτή τη βάση.

Η PHP παρέχει διάφορα APIs για την επικοινωνία με τη MySQL. Τα κυριότερα είναι:

- PHP's MySQL Extension
- PHP's mysqli Extension
- PHP Data Objects (PDO)

Το κάθε API έχει τα πλεονεκτήματα και τα μειονεκτήματα του. Τα κύρια χαρακτηριστικά του καθενός είναι τα ακόλουθα:

#### PHP's MySQL Extension

Πρόκειται για το αρχικό extension της PHP σχεδιασμένο για εφαρμογές που επικοινωνούν με μια MySQL βάση δεδομένων. Αυτό το extension είναι σχεδιασμένο για να λειτουργεί με εκδόσεις της MySQL ως την 4.1.3. Δεν παρέχει όλα τα χαρακτηριστικά των νέων εκδόσεων της MySQL.

#### PHP's mysqli Extension

Το extension “mysqli” είναι η βελτιωμένη έκδοση του MySQL extension (MySQL *improved* extension) και σχεδιάστηκε για να εκμεταλλεύεται τα νέα χαρακτηριστικά των βάσεων δεδομένων MySQL από την έκδοση 4.1.3 και μετά. Το extension αυτό είναι ενσωματωμένο από την PHP5 και μετά.

Οι κυριότερες βελτιώσεις του *improved* extension από το αρχικό είναι οι ακόλουθες:

- Object-oriented interface
- Υποστήριξη για Prepared Statements
- Υποστήριξη για Multiple Statements



- Υποστήριξη για Transactions
- Βελτιωμένες δυνατότητες debugging

Αν η βάση MySQL είναι έκδοση 4.1.3 ή μεταγενέστερη, συνιστάται έντονα η χρήση αυτού του extension.

### PHP Data Objects (PDO)

Το “PHP Data Objects” ή “PDO” είναι ένα “*database abstraction layer*” για εφαρμογές PHP. Το PDO παρέχει ένα συνεπές API άσχετα με τον τύπο βάσης δεδομένων που θα συνδεθεί η εφαρμογή. Θεωρητικά, με τη χρήση του PDO θα μπορούσε ο προγραμματιστής να αλλάξει τύπο βάσης δεδομένων αλλάζοντας ελάχιστα τον κώδικά του.

Παρόλο που το PDO έχει τα πλεονεκτήματα ενός καθαρού, απλού και φορητού API, έχει το μειονέκτημα ότι δεν επιτρέπει τη χρήση των προηγμένων χαρακτηριστικών που είναι διαθέσιμα στις τελευταίες εκδόσεις της MySQL.

Στην παρούσα εργασία χρησιμοποιήθηκε το “**mysqli**” για την επικοινωνία με τη βάση μας καθώς και μία συνάρτηση του αρχικού extension της MySQL για τον καθαρισμό της εισόδου του χρήστη από προσπάθειες επίθεσης με SQL Injection.<sup>[2, 3]</sup>

## 3.8 Java

### 3.8.1 Περιγραφή



Η Java είναι μια αντικειμενοστραφής (object oriented) γλώσσα προγραμματισμού βασισμένη σε Classes. Έχει σχεδιαστεί ώστε να έχει όσο το δυνατόν λιγότερες εξαρτήσεις. Ο κώδικας Java δεν τρέχει απευθείας όπως συνηθίζεται σε γλώσσες όπως η C, αλλά μετατρέπεται σε “Bytecode” και τρέχει πάνω σε ένα **Java Virtual Machine (JVM)** ανεξάρτητα από την αρχιτεκτονική του υπολογιστή. Δίνει έτσι τη δυνατότητα στους προγραμματιστές να γράψουν τον κώδικά τους μία φορά και να μπορούν να τον τρέξουν παντού (“Write once, Run anywhere”). Η Java είναι μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού με πάνω από 9 εκατομμύρια προγραμματιστές.

Ένα μεγάλο πλεονέκτημα της μετατροπής σε bytecode είναι η φορητότητα. Λόγω όμως του ότι ο κώδικας Java δεν μετατρέπεται σε κώδικα μηχανής που είναι εξαρτημένος από την εκάστοτε

αρχιτεκτονική υπολογιστή, κατηγορείται ότι τα προγράμματα γραμμένα σε αυτή είναι πιο αργά από αντίστοιχα που γίνονται compile σε κώδικα μηχανής καθώς επίσης και ότι καταναλώνει περισσότερη μνήμη. Έχουν όμως δημιουργηθεί compilers οι οποίοι μετατρέπουν το bytecode σε κώδικα μηχανής κατά τη διάρκεια

της εκτέλεσης (Just-In-Time – JST compilers ) επιταχύνοντας την εκτέλεση του κώδικα.

Ένα άλλο μεγάλο πλεονέκτημα της Java είναι η “Αυτόματη διαχείριση μνήμης”. Η ίδια η Java αναλαμβάνει να διαχειριστεί τον κύκλο ζωής των αντικειμένων. Ο προγραμματιστής δημιουργεί ένα αντικείμενο και το Java runtime αναλαμβάνει να ελευθερώσει τη μνήμη του όταν αυτό δε χρησιμοποιείται πια. Όταν δεν υπάρχουν άλλες αναφορές ενός αντικειμένου, η μνήμη που καταναλώνει μπορεί να ελευθερωθεί αυτόματα από τον garbage collector. Με αυτό το μηχανισμό ο προγραμματιστής απαλλάσσεται από τη διαδικασία του να διαθέσει μνήμη για ένα νέο αντικείμενο καθώς και του να την ελευθερώσει όταν δεν το χρειάζεται πια. Μειώνονται έτσι οι περιπτώσεις των memory-leaks.

Η Java χρησιμοποιείται σε πάρα πολλά συστήματα για πάρα πολλές εφαρμογές. Μπορεί να χρησιμοποιηθεί για command line executed εφαρμογές, σε standalone εφαρμογές με διεπαφή χρήστη (κυριότερη βιβλιοθήκη το swing), σε πληθώρα web εφαρμογών καθώς και σε πολλά embedded συστήματα. Τα τελευταία χρόνια η Java χρησιμοποιείται και σε κινητά τηλέφωνα, κυρίως στις συσκευές που τρέχουν Android καθώς αυτό χρησιμοποιεί τη Java για όλες του τις λειτουργίες.

Στην παρούσα εργασία η γλώσσα Java χρησιμοποιείται από το Android SDK για τη δημιουργία της native android εφαρμογής. <sup>[1, 2, 14, 22]</sup>

### 3.8.2 Σύνταξη

Η σύνταξη της Java έχει προέλθει κυρίως από τη C++ και είναι σχεδιασμένη ως εξολοκλήρου object oriented γλώσσα. Όλος ο κώδικα γράφεται μέσα σε “Classes” και όλα είναι ένα “αντικείμενο” με εξαίρεση κάποιους primitive τύπους δεδομένων ( πχ. integers, booleans, strings κτλ ). Αντίθετα με τη C++, η Java δεν επιτρέπει το overloading operators ή την πολλαπλή κληρονομικότητα. Μια κλάση της μπορεί να κληρονομεί από μόνο μία κλάση.

Ένα πρόγραμμα Hello-world γραμμένο σε Java έχει την ακόλουθη μορφή:

```
class HelloWorldProgram {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

Στο παράδειγμα αυτό βλέπουμε τη δήλωση της κλάσης “HelloWorldProgram” η οποία περιλαμβάνει την πιο συνηθισμένη μέθοδο σε μια Java class, τη “main” με όρισμα έναν πίνακα από Strings και τύπο void ( δεν επιστρέφει κάποιο δεδομένο δηλαδή. <sup>[1, 2, 14, 22]</sup>

## 3.9 Android

### 3.9.1 Περιγραφή



Το Android είναι ένα λειτουργικό σύστημα για φορητές συσκευές βασισμένο στο Kernel του Linux και ανεπτυγμένο από τη Google. Με το user-interface του να είναι βασισμένο στον “απευθείας χειρισμό”, έχει κυρίως σχεδιαστεί για φορητές συσκευές με οθόνη αφής, όπως τα smartphones και τα tablets καθώς και κάποιες πιο εξειδικευμένες συσκευές, όπως το Android TV, το Android Wear και το Android Auto. Το λειτουργικό χρησιμοποιεί κάποιες φυσικές κινήσεις των δακτύλων καθώς και ένα εικονικό πληκτρολόγιο για την επαφή του χρήστη με τα διάφορα αντικείμενα την οθόνη.

Από το 2013 το Android είναι το πιο δημοφιλές λειτουργικό φορητών συσκευών καθώς οι πωλήσεις του ξεπερνούν τις πωλήσεις συσκευών με iOS και Windows Mobile (των δύο μεγαλύτερων ανταγωνιστών του δηλαδή) μαζί. Τον Ιούλιο του 2013 το Google Play Store (ο ιστότοπος της Google όπου οι χρήστες του Android προμηθεύονται εφαρμογές) είχε πάνω από ένα εκατομμύριο εφαρμογές και πάνω από 50 εκατομμύρια μεταφορτώσεις εφαρμογών πράγμα το οποίο μας δείχνει ότι η πλατφόρμα του Android έχει μεγάλο εμπορικό ενδιαφέρον και προσφέρει ένα τεράστιο αγοραστικό κοινό.

Ο κώδικας του Android κυκλοφορεί από τη Google με open source licences. Οι νέες εκδόσεις παράγονται ιδιωτικά από τη Google και όταν γίνεται κάποια δημοσίευση κάποιας έκδοσης, δημοσιεύεται και ο κώδικάς της. <sup>[1, 2, 4, 14, 20, 22]</sup>

### 3.9.2 Εφαρμογές

Οι εφαρμογές (“apps”) του Android επεκτείνουν τη λειτουργικότητα των συσκευών. Είναι προγραμματισμένες σε Java με χρήση του Android Software Development Kit (SDK) το οποίο περιλαμβάνει ένα πακέτο εργαλείων, έναν ενσωματωμένο debugger, βιβλιοθήκες λογισμικού, έναν εξομοιωτή συσκευών Android, έγγραφα, παραδείγματα κώδικα και μαθήματα προγραμματισμού.

Μέσω του “Google Play store” ένας προγραμματιστής αποκτά τη δυνατότητα να δημοσιεύσει (πληρώνοντας ένα μικρό αρχικό ποσό εγγραφής) τις εφαρμογές που δημιουργεί. Με αυτό τον τρόπο μπορεί να διαθέσει το έργο του σε ένα μεγάλο κοινό. Κατά καιρούς έχουμε δει διάφορους προγραμματιστές να κάνουν τεράστιες πωλήσεις και κέρδη από εφαρμογές που δημοσιεύονται σε ηλεκτρονικά καταστήματα εφαρμογών όπως το Google Plat Store και το “App Store” της Apple.

### 3.9.3 Hardware

Η κυριότερη αρχιτεκτονική hardware που χρησιμοποιεί το Android είναι η 32-bit ARMv7. Υπάρχει και το project Android x86 το οποίο παρέχει υποστήριξη για

αρχιτεκτονική x86 καθώς έχουν αρχίσει να εμφανίζονται επεξεργαστές Intel σε διάφορες συσκευές όπως κινητά τηλέφωνα. Στην τελευταία έκδοση το Android απαιτεί τουλάχιστον 512 MBs RAM και 32-bit ARMv7 ή MIPS ή x86 αρχιτεκτονική επεξεργαστή καθώς και μονάδα επεξεργασίας γραφικών που υποστηρίζει OpenGL ES 2.0. <sup>[1, 2, 4, 14, 20, 22]</sup>

### 3.9.4 Διαχείριση μνήμης

Καθώς οι συσκευές που τρέχουν Android είναι κυρίως συσκευές που τροφοδοτούνται από μπαταρίες και όχι απευθείας παροχή ρεύματος, το Android έχει σχεδιαστεί να διαχειρίζεται τη μνήμη RAM με τέτοιο τρόπο που η κατανάλωση ρεύματος να πέφτει στο ελάχιστο δυνατό σε αντίθεση με desktop λειτουργικά συστήματα τα οποία θεωρούν ότι έχουν απεριόριστη παροχή ενέργειας. Όσο μια εφαρμογή είναι ενεργή, οι υπόλοιπες απενεργοποιημένες εφαρμογές δεν καταναλώνουν πόρους (ή τουλάχιστον στην πράξη καταναλώνουν ελάχιστους) και μένουν στο παρασκήνιο μέχρι να χρειαστούν ξανά. Το πλεονέκτημα αυτού του σχεδιασμού είναι ότι οι εφαρμογές δε χρειάζονται να κλείνονται και να επανεκινούνται από την αρχή γίνοντας έτσι πιο γρήγορες αλλά επίσης μειώνεται η κατανάλωση ενέργειας κατά την αδράνειά τους.

Όταν η διαθέσιμη μνήμη της συσκευής μειώνεται πολύ, το λειτουργικό αρχίζει να κλείνει εφαρμογές και διεργασίες που δεν έχουν χρησιμοποιηθεί για αρκετό διάστημα, έτσι με αυτό τον τρόπο το λειτουργικό διαχειρίζεται τις εφαρμογές διαφανώς χωρίς να χρειάζεται η επαφή του χρήστη. <sup>[1, 2, 4, 14, 20, 22]</sup>

## 3.10 Apache HTTP Server

### 3.10.1 Περιγραφή



Ο **Apache HTTP Server** (apache) είναι ένα project με σκοπό τη δημιουργία ενός ισχυρού, επαγγελματικού επιπέδου, με πολλά χαρακτηριστικά με ελεύθερο κώδικα HTTP (Web) Server. Το project αυτό διαχειρίζεται από εθελοντές παγκοσμίως. Από το 1996 έως και σήμερα ο apache είναι ο κυρίαρχος web-server παγκοσμίως. Χρησιμοποιείται σε πάρα πολλά λειτουργικά συστήματα, όπως Linux, Microsoft Windows, Mac OS, Solaris, Unix κ.α. Κυριότερα τον συναντάμε σε Unix based λειτουργικά συστήματα. Στην εργασία αυτή τρέχει μέσω του XAMPP. <sup>[2, 7]</sup>

### 3.10.2 License

Ο apache παρέχεται με license [Apache 2.0 License](#).

## 3.11 XAMPP

### 3.11.1 Περιγραφή



# XAMPP

Το **XAMPP** είναι το πιο δημοφιλές περιβάλλον προ-

γραμματισμού για PHP. Πρόκειται για ένα Open Source πακέτο λογισμικού το οποίο έχει σχεδιαστεί και προετοιμαστεί έτσι ώστε να είναι πάρα πολύ εύκολο κατά την εγκατάσταση και τη χρήση. Είναι cross-platform και μπορεί να λειτουργήσει σε Linux, Microsoft Windows, Solaris και Mac OS. Τα αρχικά XAMPP είναι ακρωνύμιο των αρχικών γραμμάτων των βασικών τεχνολογιών που περιλαμβάνει και είναι τα εξής: <sup>[2, 9]</sup>

- **X** ( από τη λέξη “cross” με την έννοια του cross-platform )
- **A**pache HTTP Server
- **M**ySQL
- **P**HP
- **P**erl

### 3.11.2 Απαιτήσεις και χαρακτηριστικά

Το XAMPP απαιτεί ένα αρχείο zip ή exe να εκτελεστεί. Καθόλου ή ελάχιστες ρυθμίσεις χρειάζονται για τη λειτουργία του. Είναι αυτόνομο και πολλαπλές παρουσίες του μπορούν να υπάρξουν ταυτόχρονα στον ίδιο υπολογιστή. Η κάθε εγκατάσταση μπορεί να αντιγραφεί εύκολα από έναν υπολογιστή σε έναν άλλο. <sup>[2, 9]</sup>

### 3.11.3 Χρήση

Αρχικά το XAMPP είχε σκοπό τη χρήση μόνο ως εργαλείο προγραμματισμού, δίνοντας τη δυνατότητα στους προγραμματιστές να χρησιμοποιήσουν το pc τους για να δοκιμάσουν τις εφαρμογές τους χωρίς να είναι απαραίτητο να είναι συνδεδεμένοι στο internet. Γι αυτό το λόγο έχει αρκετά χαρακτηριστικά των τεχνολογιών που περιλαμβάνει απενεργοποιημένα. Στην πραγματικότητα όμως συχνά χρησιμοποιείται ως server για πραγματικές ιστοσελίδες στο internet.

Το XAMPP περιλαμβάνει εργαλεία διαχείρισης βάσεων δεδομένων, πχ. PHPMyAdmin. Επίσης περιλαμβάνονται εφαρμογές όπως FTP server και άλλα εργαλεία για την εγκατάσταση CMS εφαρμογών κτλ. <sup>[2, 9]</sup>

### 3.11.4 Εργαλεία - Τεχνολογίες

#### XAMPP για Windows

- Apache HTTP Server
- MySQL
- PHP
- phpMyAdmin
- FileZilla FTP Server
- Tomcat
- Strawberry Perl Portable

- XAMPP Control Panel

#### XAMPP για Linux

- Apache
- MySQL
- PHP
- phpMyAdmin
- OpenSSL

### 3.11.5 License

Ο apache παρέχεται με license [GNU GPL](#).

## 3.12 phpMyAdmin

### 3.12.1 Περιγραφή



Το phpMyAdmin είναι ένα δωρεάν και open source εργαλείο για τη διαχείριση βάσεων δεδομένων MySQL μέσω ενός Web browser. Είναι γραμμένο σε PHP και είναι cross platform, μπορεί να λειτουργήσει σε όλα τα συστήματα στα οποία μπορεί να τρέξει PHP και MySQL. Δίνει τη δυνατότητα δημιουργίας, επεξεργασίας, και διαγραφής βάσεων δεδομένων, πινάκων, πεδίων και σειρών καθώς και εκτέλεσης απευθείας ερωτημάτων SQL. <sup>[2, 11]</sup>

### 3.12.2 Χαρακτηριστικά

Τα κυριότερα χαρακτηριστικά που περιλαμβάνει η εφαρμογή είναι τα εξής:

- Web interface
- Διαχείριση βάσεων MySQL
- Εισαγωγή δεδομένων από CSV και SQL
- Εξαγωγή δεδομένων σε διάφορες μορφές, όπως: CSV, SQL, XML, ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, LaTeX κ.α.
- Διαχείριση πολλαπλών servers
- Δημιουργία PDF γραφικών του σχήματος της βάσης
- Δημιουργία πολύπλοκων queries με χρήση Query-by-Example (QBE)
- Αναζήτηση σε ολόκληρη τη βάση ή υποσύνολο αυτής
- Μετατροπή αποθηκευμένων δεδομένων σε οποιαδήποτε μορφή με χρήση διάφορων συναρτήσεων, πχ. εμφάνιση BLOB-δεδομένων ως εικόνα
- Live γραφήματα για παρακολούθηση της χρήσης του MySQL όπως συνδέσεις, διεργασίες, χρήση επεξεργαστή και μνήμης κτλ.

### 3.12.3 License

Το phpMyAdmin παρέχεται με License [GNU General Public License 2](#).

## 3.13 JSON

### 3.13.1 Περιγραφή



Το JSON είναι μία open standard μορφή που χρησιμοποιεί κείμενο, το οποίο είναι δυνατόν να διαβάζεται από άνθρωπο, για τη μεταφορά αντικειμένων δεδομένων τα οποία αποτελούνται από ζευγάρια “χαρακτηριστικών-τιμών”. Χρησιμοποιείται κυρίως για τη μεταφορά δεδομένων μεταξύ των servers και των web εφαρμογών ως εναλλακτικό του XML. Στην παρούσα εργασία η μορφή JSON χρησιμοποιείται σε όλες τις επικοινωνίες της Android εφαρμογής με το web server.  
[2]

### 3.13.2 Τύποι δεδομένων και σύνταξη

Οι βασικοί τύποι δεδομένων του JSON είναι οι εξής:

- **Αριθμοί.**
- **String.** Μια σειρά από κανέναν ή περισσότερους χαρακτήρες Unicode.
- **Boolean.** Τιμές true ή false.
- **Πίνακες.** Ταξινομημένες λίστες από καμία ή περισσότερες τιμές οποιουδήποτε τύπου. Οι πίνακες περικλείονται ανάμεσα σε “square brackets” (“[” και “]”) και τα στοιχεία διαχωρίζονται με κόμμα.
- **Αντικείμενα.** Μη ταξινομημένοι “σχετιζόμενοι πίνακες” (associative array) σε ζευγάρια ονόματος-τιμής. Τα αντικείμενα χωρίζονται με “curly brackets” (“{” και “}” ) και χρησιμοποιούν κόμμα για να χωρίσουν το κάθε ζευγάρι. Το κάθε ζευγάρι ονόματος-τιμής χωρίζεται με άνω-κάτω τελεία ( “:” ) όπου αριστερά είναι το όνομα και δεξιά η τιμή. Όλα τα κλειδιά/ονόματα είναι strings και πρέπει να είναι μοναδικά σε ένα αντικείμενο.
- **Null.** Μια κενή τιμή με χρήση της λέξης null.

Ένα παράδειγμα σύνταξης ενός αντικειμένου ενός χρήστη σε μορφή JSON είναι το ακόλουθο:

```
{
  "name" : "Panos",
  "surname" : "Athanasopoulos",
  "username" : "panosath",
  "email" : "panosath@test.com",
  "phoneNumbers" : [
    {
      "type" : "home",
      "number" : '210-1112345'
    }
  ]
}
```

```

    },
    {
        "type" : "cellphone",
        "number" : "691112345"
    }
]
}

```

## 3.14 GSON (Βιβλιοθήκη)

### 3.14.1 Περιγραφή

Το GSON είναι μια open source βιβλιοθήκη για Java η οποία χρησιμοποιείται για μετατροπή αντικειμένων Java σε JSON και αντίστροφα. Χρησιμοποιείται ευρέως σε Android εφαρμογές οι οποίες απαιτούν την επικοινωνία με servers μέσω JSON. Στην παρούσα εργασία η βιβλιοθήκη αυτή χρησιμοποιείται για να μετατρέψει σε αντικείμενα Java τα αποτελέσματα από όλες τις επικοινωνίες της Android εφαρμογής με το server. <sup>[2, 12]</sup>

### 3.14.2 Χαρακτηριστικά

Η βιβλιοθήκη αυτή έχει τα εξής χαρακτηριστικά:

- Το GSON μπορεί να διαχειριστεί συλλογές, γενικούς τύπους και εμφωλευμένες classes.
- Κατά τη μετατροπή JSON σε Java Object (Deserializing) το GSON λειτουργεί με βάση τη δεντρική δομή του αντικειμένου και αγνοεί τα επιπλέον πεδία που μπορεί να περιλαμβάνει ένα JSON.
- Ο προγραμματιστής μπορεί να γράψει custom serializer/deserializer ώστε να ελέγχει πλήρως τη διαδικασία (de)serialize (μετατροπής από τη μία μορφή στην άλλη).
- Ο προγραμματιστής μπορεί να γράψει έναν "InstanceCreator " το οποίο επιτρέπει το serialization κλάσεων που δεν έχουν κάποιον constructor χωρίς παραμέτρους.
- Το GSON είναι προσαρμόσιμο
  - Συμπαγής ή «όμορφη» εμφάνιση του JSON
  - Διαχείριση null πεδίων
  - Κανόνες σχετικά με το ποια πεδία ενός αντικειμένου πρέπει να αγνοηθούν κατά τη μετατροπή σε JSON.
  - Κανόνες μετατροπής των ονομάτων των πεδίων Java (μπορούμε να έχουμε διαφορετικό όνομα πεδίου στο JSON και διαφορετικό όνομα μεταβλητής στη Java)

### 3.14.3 License

Η βιβλιοθήκη GSON παρέχεται με License [Apache License 2.0](#).



## 3.15 DataTables ( jQuery plugin )

### 3.15.1 Περιγραφή

Το DataTables είναι ένα plugin για το jQuery. Πρόκειται για ένα πολύ ευέλικτο εργαλείο το οποίο δίνει προχωρημένες λειτουργίες και ελέγχους σε πίνακες HTML. Στην παρούσα εργασία το plugin αυτό χρησιμοποιείται στους HTML πίνακες όπου απαιτείται σελιδοποίηση, αναζήτηση και ταξινομική.<sup>[13]</sup>

Show  entries Search:

Name	Position	Office	Age	Start date	Salary
Airi Satou	Accountant	Tokyo	33	2008/11/28	\$162,700
Angelica Ramos	Chief Executive Officer (CEO)	London	47	2009/10/09	\$1,200,000
Ashton Cox	Junior Technical Author	San Francisco	66	2009/01/12	\$86,000
Bradley Greer	Software Engineer	London	41	2012/10/13	\$132,000
Brenden Wagner	Software Engineer	San Francisco	28	2011/06/07	\$206,850
Brielle Williamson	Integration Specialist	New York	61	2012/12/02	\$372,000
Bruno Nash	Software Engineer	London	38	2011/05/03	\$163,500
Caesar Vance	Pre-Sales Support	New York	21	2011/12/12	\$106,450
Cara Stevens	Sales Assistant	New York	46	2011/12/06	\$145,600
Cedric Kelly	Senior Javascript Developer	Edinburgh	22	2012/03/29	\$433,060

Showing 1 to 10 of 57 entries Previous  2 3 4 5 6 Next

### 3.15.2 Χαρακτηριστικά

Τα πιο σημαντικά χαρακτηριστικά αυτού του plugin είναι τα εξής:<sup>[13]</sup>

- **Σελιδοποίηση.** Χωρισμός των αποτελεσμάτων σε σελίδες για μείωση του απαιτούμενου χώρου.
- **Άμεση αναζήτηση.** Αναζήτηση ανάμεσα σε όλα τα στοιχεία που έχει ο πίνακας και φιλτράρισμα των γραμμών που περιλαμβάνουν τον όρο αναζήτησης.
- **Ταξινόμηση.** Δυνατότητα ταξινόμησης πολλαπλών κολονών.
- **Υποστήριξη DOM, Javascript, Ajax, server-side επεξεργασία.**
- **Πολλαπλά extensions.** Editor, TableTools, FixedColumns κ.α.
- **Πλήρως μεταγλωτίσιμο.**

### 3.15.3 Χρήση

Για να ενεργοποιηθεί το plugin πάνω σε κάποιον πίνακα προσθέτουμε την παρακάτω εντολή στο event "ready" του jQuery.

```
$(document).ready(function(){  
    $('#myTable').DataTable();  
});
```

### 3.15.4 License

Το plugin DataTables παρέχεται με License "MIT License".

## 4. Κατασκευή Εφαρμογών

### 4.1 Βάση δεδομένων

#### 4.1.1 Περιγραφή

Η βάση δεδομένων της παρούσας εργασίας είναι σχεδιασμένη σε MySQL και τρέχει πάνω στο MySQL Server του XAMPP. Μπορεί να ονομαστεί όπως θέλουμε καθώς το όνομα της βάσης και τα στοιχεία σύνδεσης είναι παραμετροποιήσιμα, αλλάζοντάς τα στο αρχείο “settings.php” μέσα στο φάκελο “common”.

Η βάση αυτή αποτελείται από 4 κύριους πίνακες.

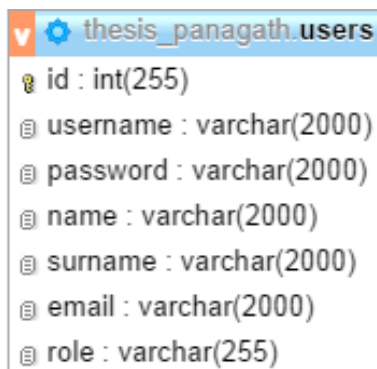
- columns\_reference
- db\_versions
- tables\_reference
- users

Έτσι όπως σχεδιαστεί η εφαρμογή ο χρήστης έχει τη δυνατότητα να παράγει νέους πίνακες και κολόνες σε αυτούς κατά τη χρήση. Παρατηρούμε έτσι όπως χρησιμοποιούμε την εφαρμογή να δημιουργούνται νέοι πίνακες και να αποκτούν κολόνες.

Οι πίνακες “tables\_reference” και “columns\_reference” είναι οι δυο κινητήριοι μοχλοί των εφαρμογών μας. Εκεί αποθηκεύονται πληροφορίες για τους πίνακες και τις κολόνες που έχει δημιουργήσει ο χρήστης και διαβάζοντας αυτές τις πληροφορίες οι εφαρμογές μας γνωρίζει αυτόματα ποιους πίνακες έχει δημιουργήσει ο χρήστης, πως τους έχει ονομάσει, τι κολόνες τους έχει προσθέσει καθώς και τι είδους πληροφορία έχουν αυτές οι κολόνες. Με αυτό τον τρόπο και αυτές τις πληροφορίες εμφανίζονται αυτόματα τα views για όλες τις λειτουργίες που απαιτούνται για τη χρήση των εφαρμογών.

#### 4.1.2 Πίνακας users

Στον πίνακα users αποθηκεύονται οι χρήστες που εγγράφονται στο σύστημά μας. Στην παρακάτω εικόνα φαίνονται τα πεδία του πίνακα. Παρατηρούμε το πεδίο id το οποίο είναι το primary key του πίνακα, τα username και password που χρησιμοποιούνται για την ταυτοποίηση του κάθε χρήστη, τα στοιχεία επικοινωνίας και το στοιχείο “role” το οποίο χρησιμοποιεί η εφαρμογή για να ξεχωρίσει αν κάποιος χρήστης είναι administrator ή απλός χρήστης.



Field	Type
id	int(255)
username	varchar(2000)
password	varchar(2000)
name	varchar(2000)
surname	varchar(2000)
email	varchar(2000)
role	varchar(255)

#### 4.1.3 Πίνακας *db\_versions*

Ο πίνακας *db\_versions* είναι βοηθητικός πίνακας για τον προγραμματιστή καθώς



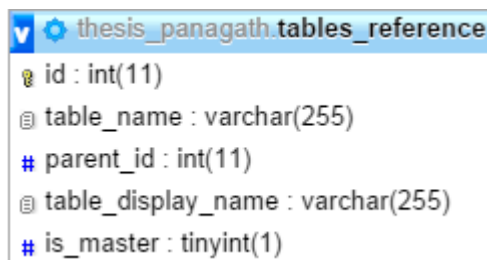
thesis_panagath.db_versions
id : int(11)
description : varchar(255)
user : varchar(255)

χρησιμοποιείται για τήρηση των στοιχείων της έκδοσης της βάσης, ώστε να μπορεί ο προγραμματιστής να καταλάβει εύκολα ποια μορφή έχει η βάση και αν χρειάζεται κάποιο update για να λειτουργήσει με την τελευταία

έκδοση του κώδικα. Κάθε φορά που γίνεται μια αλλαγή στο σχήμα της βάσης κατά τη διάρκεια του development, διατηρούνται db scripts με τα statements που εκτελούν τις απαιτούμενες αλλαγές και σώζονται ώστε να μπορούν να χρησιμοποιηθούν για τη δημιουργία νέων βάσεων δεδομένων. Με αυτό τον τρόπο όταν χρειαζόμαστε να δημιουργήσουμε μια νέα βάση για να ξεκινήσουμε το development μιας νέας εφαρμογής που θα χρησιμοποιεί την παρούσα ως σημείο αναφοράς επιταχύνουμε τη διαδικασία με αυτό τον τρόπο.

#### 4.1.4 Πίνακας *tables\_reference*

Ο πίνακας *tables\_references* είναι ο ένας από τους δυο πιο σημαντικούς πίνακες



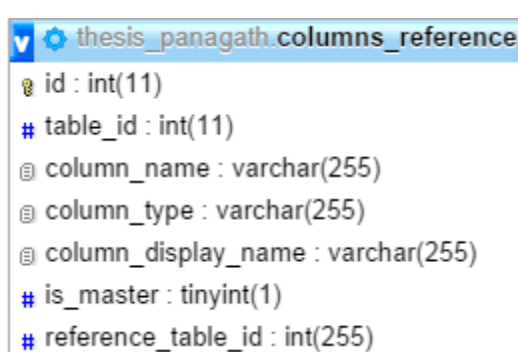
thesis_panagath.tables_reference
id : int(11)
table_name : varchar(255)
parent_id : int(11)
table_display_name : varchar(255)
is_master : tinyint(1)

της εφαρμογής μας. Σε αυτό τον πίνακα δηλώνονται όλοι οι πίνακες που δημιουργούνται από το χρήστη καθώς και χρήσιμες πληροφορίες για αυτούς. Αρχικά έχουμε το όνομα του πίνακα στη βάση, το εμφανιζόμενο όνομα αυτού το οποίο είναι ότι βλέπει ο χρήστης όταν εμφανίζονται τα

δεδομένα, καθώς και η πληροφορία αν είναι κεντρικός πίνακας, ενώ αν δεν είναι κεντρικός δηλώνεται ποιος είναι ο "parent" πίνακας (με το πεδίο *parent\_id* που δείχνει στο *id* του ίδιου πίνακα) του για να φαίνεται η one-to-many σύνδεση μεταξύ τους.

#### 4.1.5 Πίνακας *columns\_reference*

Ο πίνακας *columns\_reference* είναι ο δεύτερος από τους δυο πιο σημαντικούς



thesis_panagath.columns_reference
id : int(11)
table_id : int(11)
column_name : varchar(255)
column_type : varchar(255)
column_display_name : varchar(255)
is_master : tinyint(1)
reference_table_id : int(255)

πίνακες της εφαρμογής μας. Εδώ δηλώνονται όλες οι κολόνες που προσθέτει ο χρήστης σε κάθε πίνακα. Για την κάθε κολόνα έχουμε την πληροφορία του πίνακα στον οποίο ανήκει με one-to-many σύνδεση με το πεδίο *id* στον πίνακα *tables\_reference*. Επίσης διατηρούνται πληροφορίες όπως το εμφανιζόμενο όνομα, ο τύπος της κολόνας (πχ. αν είναι κολόνα κειμένου, κολόνα για

συντεταγμένες χάρτη, αριθμός κτλ.) καθώς και το αν πρόκειται για την κύρια κολόνα του εκάστοτε πίνακα.

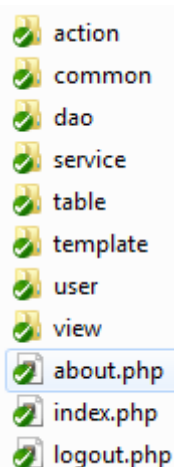
Κατά το σχεδιασμό της εφαρμογής αποφασίσθηκε κάθε φορά που δημιουργείται ένας πίνακας να αποκτά μια κεντρική κολόνα με όνομα “value” και εμφανιζόμενο όνομα το εμφανιζόμενο όνομα του πίνακα στον οποίο ανήκει. Με αυτό τον τρόπο εξασφαλίζουμε ότι κάθε πίνακας έχει ως αφετηρία μία κολόνα που περιέχει την κεντρική πληροφορία για αυτόν και στη συνέχεια ο χρήστης θα μπορεί να προσθέσει όσες άλλες θέλει.

Τέλος όταν σε έναν πίνακα δημιουργούμε έναν referenced πίνακα σε αυτόν, χρειαζόμαστε μια κολόνα που να δείχνει σε ποια γραμμή του πίνακα αντιστοιχεί η πληροφορία μας. Πχ. Όταν έχουμε έναν πίνακα που περιλαμβάνει τραγούδια και σε αυτόν δημιουργούμε έναν πίνακα με ονόματα τραγουδιστών, τότε στον πίνακα με τα τραγούδια δημιουργείται μια κολόνα η οποία δείχνει σε μία γραμμή του πίνακα με τα ονόματα τραγουδιστών, χρησιμοποιώντας το id του δεύτερου ως foreign-key. Μέσω αυτής της κολόνας μπορούμε να αντιστοιχήσουμε ότι το τραγούδι X αντιστοιχεί στον τραγουδιστή Y. Αυτή η κολόνα δημιουργείται αυτόματα από το σύστημά μας και έχει όνομα το όνομα του referenced πίνακα ακολουθούμενο από το επίθεμα “\_id” υποδηλώνοντας έτσι ότι αυτή η κολόνα δείχνει στο id του X πίνακα. Το πεδίο “referenced\_table\_id” στον πίνακα “columns\_reference” δηλώνει ότι η κολόνα είναι foreign-key προς τον πίνακα με id (από τον πίνακα “tables\_reference” ) την τιμή του πεδίου.

## 4.2 Ανάλυση λειτουργίας και κώδικα εφαρμογών

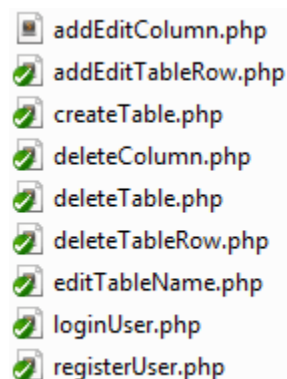
### 4.2.1 Γενική δομή κώδικα PHP

Η πρώτη εφαρμογή της παρούσας εργασίας είναι ο web-server ο οποίος ταυτόχρονα αποτελεί και τη web εφαρμογή για τον έλεγχο και την παρουσίαση των δεδομένων. Στην ακόλουθη εικόνα βλέπουμε τη δομή του κύριου φακέλου της εφαρμογής.



εφαρμογής.

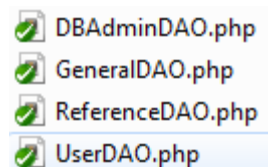
Στο φάκελο “action” περιλαμβάνονται τα αρχεία που έχουν σχέση με τις “πράξεις” που γίνονται στην εφαρμογή. Πχ. η δημιουργία πίνακα, η προσθήκη κολόνας, η προσθήκη δεδομένων κτλ. Ο λόγος για τον οποίο έχουμε ξεχωρίσει τις “πράξεις” από τα αρχεία τα οποία αναλαμβάνουν την παρουσίαση των δεδομένων είναι ότι οι ίδιες πράξεις γίνονται και μέσω της εφαρμογής Android. Έτσι είτε μια πράξη γίνεται από το website είτε από την εφαρμογή του Android και οι δύο κάνουν αιτήματα στο ίδιο αρχείο του server έχοντας έτσι κοινό τον κώδικα που



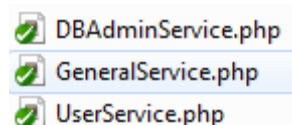
αναλαμβάνει να κάνει κάποια διεργασία. Με αυτό τον τρόπο εξασφαλίζουμε ότι η εφαρμογή θα συμπεριφέρεται με τον ίδιο τρόπο από όπου και αν γίνεται μια πράξη, αλλά και το ότι γράφουμε τον κώδικα για αυτή μία φορά μειώνοντας έτσι και το χρόνο παραγωγής αλλά κυρίως μειώνοντας το χρόνο συντήρησης.

Στο φάκελο “common” βρίσκονται κάποια βοηθητικά εργαλεία. Κυριότερα από αυτά είναι το αρχείο “settings.php” στο οποίο ορίζεται η βάση δεδομένων καθώς και τα στοιχεία σύνδεσης με αυτή. Ένα άλλο σημαντικό αρχείο είναι το “TableUtil.php” μέσα στο οποίο βρίσκονται πολλές βοηθητικές functions για την εμφάνιση των πινάκων της εφαρμογής.

Στο φάκελο “dao” βρίσκουμε όλα τα αρχεία τα οποία περιλαμβάνουν μεθόδους για την επικοινωνία με τη βάση δεδομένων. Το αρχείο “UserDAO.php” περιλαμβάνει μεθόδους για την επικοινωνία με τη βάση που έχει σχέση με τους χρήστες του συστήματος. Στο αρχείο “ReferenceDAO.php” βρίσκουμε μεθόδους που έχουν σχέση με το πίνακες “tables\_reference” και “columns\_reference” ώστε να βρίσκουμε και να προσθέτουμε πληροφορίες σε αυτούς τους πίνακες. Στο “DBAdminDAO.php” βρίσκονται όσες μέθοδοι αναλαμβάνουν τη δημιουργία/διαγράφη/επεξεργασία πινάκων και κολονών σε αυτούς, αυτές δηλαδή που επηρεάζουν το σχήμα της βάσης. Τέλος στο “GeneralDAO.php” βρίσκουμε όσες μεθόδους έχουν σχέση με τις πληροφορίες των πινάκων που δημιουργεί ο χρήστης.



Στο φάκελο “service” βρίσκουμε αρχεία τα οποία διαχειρίζονται τη “λογική” της εφαρμογής. Όταν για παράδειγμα δημιουργείται ένας referenced πίνακας τότε πρέπει να δημιουργηθεί ο νέος πίνακας, να δηλωθεί στον πίνακα tables\_reference, να δημιουργηθεί μια κολόνα στον πίνακα “πατέρα” για τη σύνδεση μεταξύ τους (όπως έχει αναφερθεί προηγουμένως) και αυτή να δηλωθεί στον πίνακα columns\_reference και τέλος στο νέο πίνακα να δημιουργηθεί η κεντρική κολόνα “value” και να δηλωθεί και αυτή στον πίνακα columns\_reference. Αυτή τη λογική αναλαμβάνουν να διαχειριστούν οι μέθοδοι που βρίσκονται στα αρχεία αυτά. Σε όσες λειτουργίες χρειάζονται πάνω από μια αλλαγές στο σχήμα της βάσης, τότε αυτές οι μέθοδοι αναλαμβάνουν να ανοίξουν ένα νέο transaction με τη βάση και να το χρησιμοποιήσουν για όλες τις ενέργειες, αν κάποια ενέργεια αποτύχει τότε γίνεται rollback των προηγούμενων ώστε να μη μένουν αποτελέσματα από ενέργειες που δεν έχουν ολοκληρωθεί.



Στους φακέλους “table”, “view”, “user” καθώς και τα περιεχόμενα του root φακέλου περιλαμβάνουν αρχεία που έχουν σχέση με αυτά που βλέπει ο χρήστης. Όπως πχ την αρχική σελίδα, τη σελίδα εγγραφής χρήστη, τη σελίδα εμφάνισης ενός πίνακα, τη σελίδα εμφάνισης δεδομένων πίνακα κτλ. Σημαντικό σημείο ο φάκελος “external” μέσα στο φάκελο “view” στον οποίο βρίσκονται διάφορα αρχεία php τα οποία χρησιμοποιεί η Android εφαρμογή για να παίρνει τα απαιτούμενα δεδομένα σε μορφή JSON. Όπως για παράδειγμα τα δεδομένα ενός πίνακα.

Στο φάκελο “template” βρίσκονται όσα αρχεία έχουν σχέση με την εμφάνιση της ιστοσελίδας. Εκεί βρίσκονται τα αρχεία CSS, τα javascript scripts, οι εικόνες καθώς και οι php σελίδες που έχουν σχέση με την κεφαλίδα και το υποσέλιδο σε όλο το project. Πειράζοντας τα css αρχεία και ρυθμίζοντας το όνομα της εφαρμογής και άλλες πληροφορίες από το αρχείο flavorSettings.php μπορούμε να αλλάξουμε σε πολύ μικρό χρονικό διάστημα την εμφάνιση της εφαρμογής μας. Έτσι μπορούμε να προετοιμάσουμε σύντομα τη βάση για μια νέα ομοειδή εφαρμογή.

#### 4.2.2 Γενική δομή κώδικα Android

Η δεύτερη εφαρμογή της παρούσας εργασίας είναι η native android εφαρμογή η οποία εμφανίζει τα ίδια δεδομένα και εκτελεί τις ίδιες εργασίες με τη web εφαρμογή. Η γενική δομή του project αυτού είναι η τυπική δομή ενός Android project.

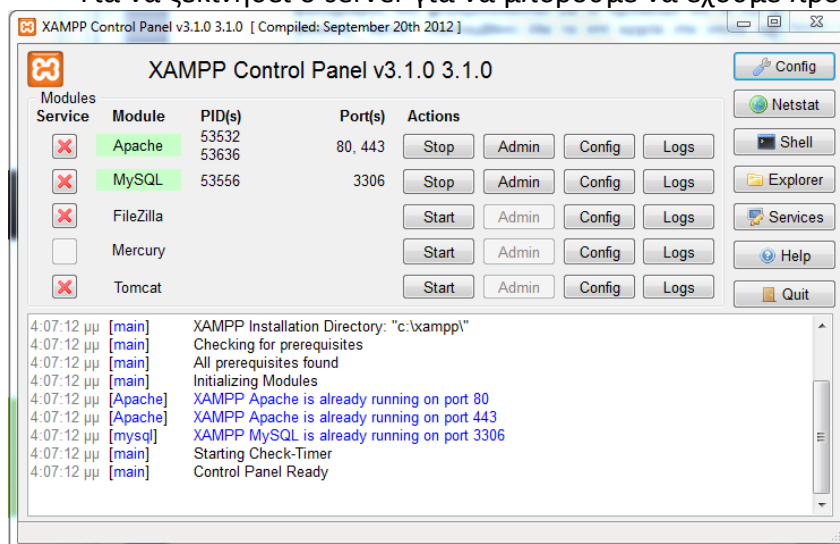
Στο root φάκελο βρίσκουμε το σημαντικό αρχείο για το Android "AndroidManifest.xml" στο οποίο δηλώνονται τα Activities που χρησιμοποιεί η εφαρμογή καθώς και τις άδειες (permissions) που χρειάζεται η εφαρμογή για να λειτουργήσει. Κάτω από το φάκελο "libs" βρίσκουμε τα αρχεία των εξωτερικών βιβλιοθηκών που χρησιμοποιούνται στο project.

Μέσα στο φάκελο "res" βρίσκουμε κάποιους φακέλους με όνομα "drawable-XXX". Στον φάκελο "drawable-mdpi" είναι τοποθετημένα όλα τα εικονίδια και φωτογραφίες που χρησιμοποιούνται για το σχεδιασμό του UI της εφαρμογής. Ο φάκελος "layout" περιλαμβάνει όλα τα xml αρχεία στα οποία σχεδιάζονται τα διάφορα views και components αυτών.

Προχωρώντας μέσα στο path "src\com\panagath\thesis" του project βρίσκουμε τη δομή του πραγματικού κώδικα της εφαρμογής. Ο φάκελος "activities" περιλαμβάνει όλες τις Java classes που αποτελούν τις ίδιες τις Activities, τους controllers δηλαδή του project μας. Στο φάκελο "adapters" έχουν τοποθετηθεί οι classes που αναλαμβάνουν να γεμίσουν τις "λίστες", όπου δηλαδή δείχνουμε δεδομένα σε μορφή λίστας ή πίνακα. Ο φάκελος "domain" έχει τις classes που αντιπροσωπεύουν τα "Αντικείμενα" της εφαρμογής. Μέσα στο φάκελο "service" θα βρούμε classes οι οποίες έχουν μεθόδους διαχείρισης της λογικής της εφαρμογής και τέλος μέσα στο φάκελο "util" βρίσκονται οι βοηθητικές classes για τη χρήση του project. Σημαντική class για τη λειτουργία της εφαρμογής είναι η NetworkUtil.java η οποία έχει τις μεθόδους για εκτέλεση HTTP GET και HTTP POST requests προς το server. Επίσης η class ThesisConstants.java περιλαμβάνει σταθερές του συστήματος τις οποίες μπορούμε να αλλάξουμε για να δημιουργήσουμε άλλη ομοειδή εφαρμογή.

#### 4.2.3 Εκκίνηση εφαρμογής – Πρόσβαση στην εφαρμογή

Για να ξεκινήσει ο server για να μπορούμε να έχουμε πρόσβαση στην εφαρμογή



πρέπει αρχικά να εκκινήσουμε τον Apache και τη MySQL από το XAMPP πατώντας τα πλήκτρα Start για το κάθε προαναφερθέν component.



Στη συνέχεια ανοίγουμε το browser της επιλογής μας και χρησιμοποιούμε το σύνδεσμο:

<http://localhost/thesis/> για να μπούμε στην εφαρμογή στον τοπικό υπολογιστή που τρέχει και το XAMPP.

Εναλλακτικά για να δούμε την εφαρμογή από τρίτο υπολογιστή αντικαθιστούμε το "localhost" με την IP του υπολογιστή στον οποίο τρέχει το XAMPP ή με κάποιο DNS name αν υπάρχει. Πχ. <http://192.168.1.55/thesis/>

Η πρώτη σελίδα που θα δούμε όταν μπούμε στη web εφαρμογή είναι η αρχική σελίδα στην οποία ζητείται από το χρήστη να κάνει login ή στην περίπτωση που δεν έχει κάποιο ενεργό λογαριασμό να κάνει εγγραφή χρήστη δίνοντας τα στοιχεία του.



© Copyright Athanasopoulos Panagiotis.

Layout by | Website Templates

This template downloaded from [free website templates](#)

Για να γίνει το login ο χρήστης καλείται να δώσει το username και το password του και να πατήσει το πλήκτρο submit. Με την εκτέλεση αυτής της ενέργειας καλείται η action "loginUser.jsp"

```
$loginUser = null;
$xmlExternal = false;
if (isset($_POST["external"])) {
    $xmlExternal = MiscUtil::test_input($_POST["external"]);
}
```

```

    if (isset($_POST["loginUsername"])) {
        $loginUsername =
MiscUtil::test_input($_POST["loginUsername"]);
    }
    if (isset($_POST["loginPassword"])) {
        $loginPassword =
MiscUtil::test_input($_POST["loginPassword"]);
    }

    if (StringUtil::notEmptyOrNull($loginUsername) &&
StringUtil::notEmptyOrNull($loginPassword) &&
StringUtil::isEmptyOrNull($loginErr) ) {
        $userService = new UserService();
        $loginUser = $userService->loginUser ($loginUsername,
$loginPassword);
        if ( $loginUser != null ) {
            session_regenerate_id();
            $_SESSION['sessionUserId'] = $loginUser['id'];
            $_SESSION['sessionUsername'] = $loginUser['username'];
            $_SESSION['sessionUserRole'] = $loginUser['role'];
            session_write_close();
        } else {
            $loginErr = 'Λάθος username ή password';
        }
    } else {
        $loginErr = 'Άκυρα δεδομένα';
    }
}

```

Παρατηρούμε στο παραπάνω απόσπασμα από τον κώδικα το διάβασμα των παραμέτρων από το HTTP POST request και στη συνέχεια την κλήση της μεθόδου loginUser από την UserService για να ελέγξει αν πράγματι υπάρχει αυτός ο χρήστης. Αν υπάρχει τότε αποθηκεύει τα στοιχεία του στο Session για μετέπειτα χρήση από την εφαρμογή. Στην περίπτωση λάθος δεδομένων ή μη υπαρκτού χρήστη επιστρέφεται το Error για να παρουσιαστεί στη σελίδα.



Αντίστοιχα στο Android ανοίγοντας την εφαρμογή οδηγούμαστε στην αρχική οθόνη όπου ζητείται από το χρήστη να δώσει username και password για να κάνει login.

Πατώντας το πλήκτρο “Σύνδεση” η εφαρμογή κάνει μια HTTP request στο server για να ελέγξει αν τα στοιχεία σύνδεσης είναι σωστά. Στην περίπτωση που η σύνδεση είναι επιτυχής οδηγείται στη Home Page. Αν αποτύχει τότε ενημερώνεται ότι έχει κάνει κάποιο λάθος και δεν υπάρχει λογαριασμός με αυτά τα στοιχεία στο σύστημα.

Στο ακόλουθο κομμάτι κώδικα βλέπουμε την ασύγχρονη κλήση της εφαρμογής στο server για να γίνει το login.



```

private class LoginTask extends AsyncTask<String, Integer, Long> {

    protected void onPreExecute() {
        //putToastMessage("Login Task Started");
    }

    protected Long doInBackground(String... progress) {

        try {
            Map<String,String> loginMap = new HashMap<String,
String>();
            loginMap.put("external", "true");
            loginMap.put("loginUsername", username);
            loginMap.put("loginPassword", password);

            loginResult =
NetworkUtil.createPost(ThesisConstants.LOGIN_ACTION_URL, loginMap);

        } catch (Exception e) {

        }

        return 0L;
    }

    protected void onProgressUpdate(Integer... progress) {
        // setProgressPercent(progress[0]);
    }

    protected void onPostExecute(Long result) {
        progDailog.dismiss();
        checkLogin();
    }
}

public void loginButton ( View v ) {
    EditText usernameET = (EditText) findViewById(R.id.usernameET);
    username = usernameET.getEditableText().toString();

    EditText passwordET = (EditText) findViewById(R.id.passwordET);
    password = passwordET.getEditableText().toString();

    if ( StringUtil.isEmptyOrNull(username) ||
StringUtil.isEmptyOrNull(password) ) {
        } else {

            progDailog = ProgressDialog.show(this , "Login", "", true);
            LoginTask loginTask = new LoginTask();
            loginTask.execute("login");

        }
}
}

```

Όταν ο χρήστης πατήσει το πλήκτρο login εκτελείται ο κώδικας της μεθόδου “loginButton” ο οποίος με τη σειρά του ελέγχει την είσοδο του χρήστη και εκτελεί το ασύγχρονο task του login.

#### 4.2.4 Εγγραφή χρήστη

Στην περίπτωση που ο χρήστης επισκέπτεται την εφαρμογή μας για πρώτη φορά την εφαρμογή χρειάζεται να κάνει μια εγγραφή χρήστη. Η αρχική οθόνη στη web εφαρμογή, όταν ο χρήστης δεν έχει κάνει login ακόμα εμφανίζεται στο μενού ένα πλήκτρο το οποίο οδηγεί στη σελίδα της εγγραφής χρήστη.

Στη σελίδα αυτή ζητείται από το χρήστη να δώσει τα στοιχεία του, όπως username, password, όνομα, επώνυμο, και email.



Thesis Panagiotis  
Athanasopoulos

Home Επιβλέπων About ΣΗΜΜΥ Register

My favourite songs

Login

Username:  Password:  Επαναφορά Υποβολή

EMΠ Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Αφού ο χρήστης δώσει τα στοιχεία του και πατήσει το πλήκτρο submit εκτελείται η action "registerUser.php". Στο ακόλουθο κομμάτι του κώδικα βλέπουμε το διάβασμα των πεδίων από το χρήστη. Ελέγχεται αν κάποιο από τα πεδία είναι κενό και αν δεν είναι κενό ελέγχεται με regular expressions αν η είσοδος του χρήστη ακολουθεί τους κανόνες που έχουμε θέσει.

Σημαντικότερα κομμάτια είναι ο έλεγχος του username και του email που έχει βάλει ο χρήστης, καθώς θέλουμε αυτά να είναι μοναδικά στο σύστημα. Οπότε όταν το email ή το username πληρούν τους κανόνες μορφής που έχουν τεθεί, τότε μέσω του UserDAO ελέγχεται αν υπάρχει χρήστης με το ίδιο username ή το ίδιο email αντίστοιχα στο σύστημα. Στην περίπτωση που αυτά δεν είναι μοναδικά, ο χρήστης ενημερώνεται με μήνυμα λάθους ότι πρέπει να τα αλλάξει.

```

$external = false;
if (isset($_POST["external"])) {
    $external = MiscUtil::test_input($_POST["external"]);
}

$userDAO = new UserDAO();
if (empty($_POST["username"])) {
    $registerErr .= "|username - Υποχρεωτικό πεδίο";
    $validationError = true;
} else {
    $username = MiscUtil::test_input($_POST["username"]);
    // check if name only contains letters and whitespace
    if (!preg_match("/^[A-Za-z0-9_-]{4,10}$/", $username)) {
        $registerErr .= "|username - Επιτρεπτή είσοδος μόνο A-Z,a-
z,0-9, -, -";
        $validationError = true;
    } else if ( $userDAO->checkIfUserNameExists( $username ) ==
'true' ) {
        $registerErr .= "|username - Υπάρχει ήδη";
        $validationError = true;
    }
}

//Do not validate empty password in edit user
$validatePass = true;
if ( empty($_POST["password"])) {
    $registerErr .= "|password - Υποχρεωτικό πεδίο";
    $validationError = true;
} else if ( $validatePass ) {
    $password = MiscUtil::test_input($_POST["password"]);
    // check if name only contains letters and whitespace
    if (!preg_match("/^[A-Za-z0-9_-]{6,12}$/", $password)) {
        $registerErr .= "|password - Επιτρεπτή είσοδος μόνο A-Z,a-
z,0-9, -, -. 6-12 Ψηφία.";
        $validationError = true;
    }
}

if (empty($_POST["name"])) {
    $registerErr .= "|Όνομα - Υποχρεωτικό πεδίο";
    $validationError = true;
} else {
    $name = MiscUtil::test_input($_POST["name"]);
}

if (empty($_POST["surname"])) {
    $registerErr .= "|Επώνυμο - Υποχρεωτικό πεδίο";
    $validationError = true;
} else {
    $surname = MiscUtil::test_input($_POST["surname"]);
}

if (empty($_POST["email"])) {
    $registerErr .= "|email - Υποχρεωτικό πεδίο";
    $validationError = true;
} else {
    $email = MiscUtil::test_input($_POST["email"]);
    $userDAO = new UserDAO();
    // check if e-mail address syntax is valid
    if (!preg_match("/([\w\_-]+\@[ \w\_-]+\.[ \w\_-]+)/", $email)) {

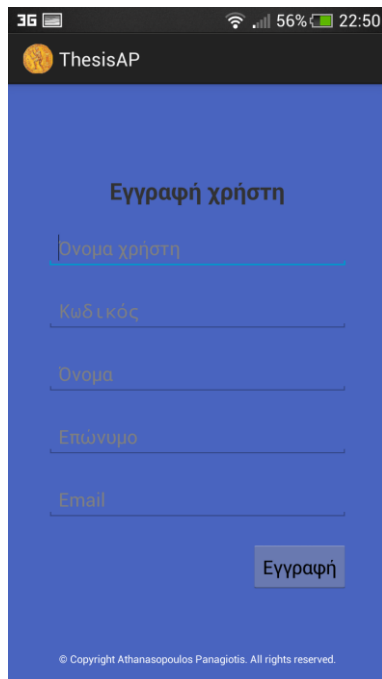
```

```

        $registerErr .= "|email - Μη έγκυρο email";
        $validationError = true;
    } else if ( $UserDAO->checkIfEmailExists($email) == 'true' ) {
        $registerErr .= "|email - Το email υπάρχει ήδη";
        $validationError = true;
    }
}
}

```

Στην περίπτωση που όλα τα πεδία περάσουν το validation χωρίς λάθη τότε δημιουργείται ένας νέος χρήστης στο σύστημα, εγγράφεται δηλαδή στον πίνακα users στη βάση δεδομένων και στη συνέχεια γίνεται αυτόματα login και οδηγείται στην αρχική σελίδα.



Αντίστοιχα στο Android όταν ο χρήστης πατήσει το κουμπί “Εγγραφή” οδηγείται στην οθόνη εγγραφής χρήστη. Στο ακόλουθο κομμάτι κώδικα βλέπουμε την κλήση στο server για την εγγραφή χρήστη.

Στην περίπτωση που η εγγραφή είναι επιτυχής ο χρήστης οδηγείται στη Home Page.

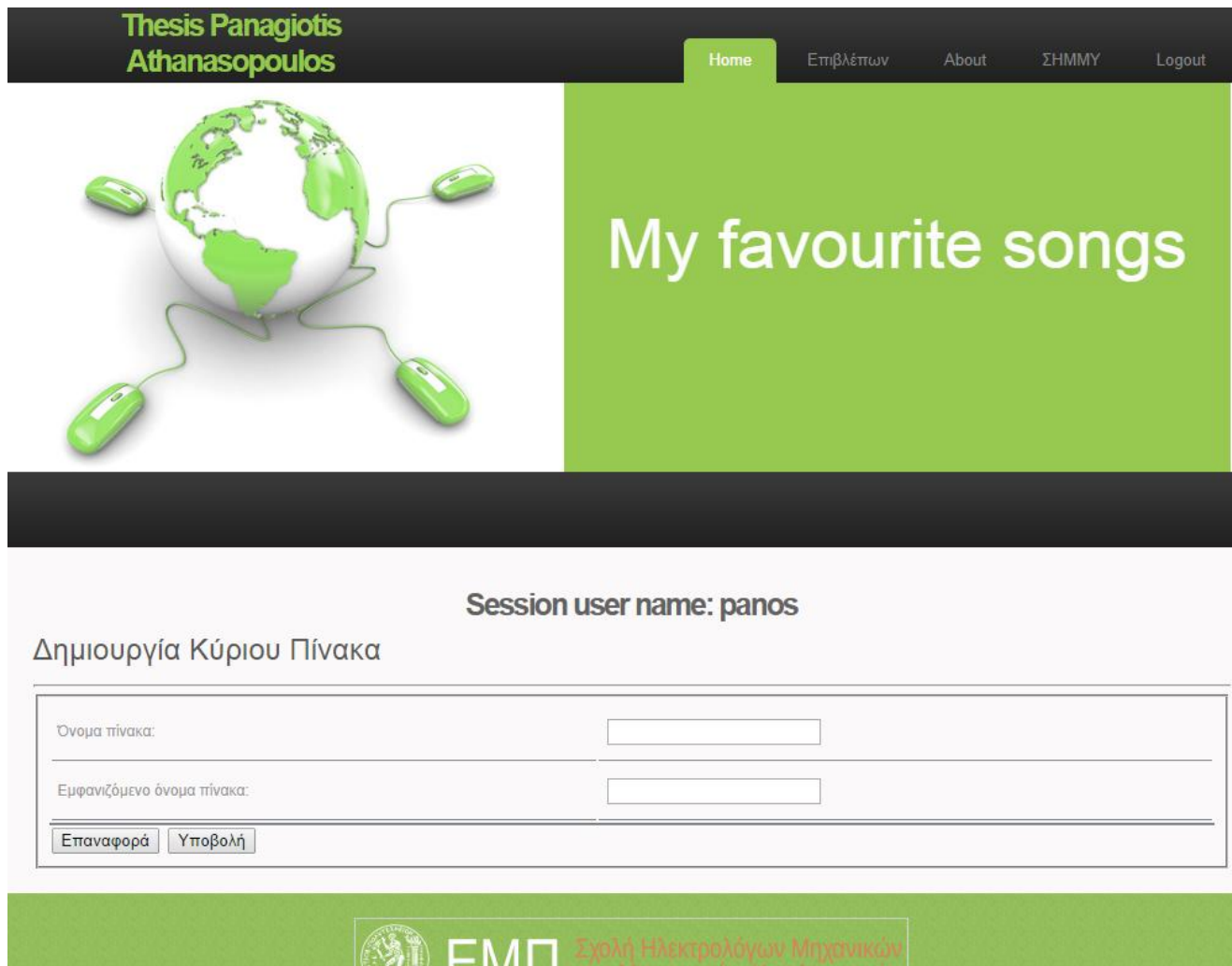
```

private class RegisterTask extends AsyncTask<String, Integer, Long> {
    .....
    protected Long doInBackground(String... progress) {
        try {
            Map<String,String> parametersMap = new HashMap<String,
String>();
            parametersMap.put("external", "true");
            parametersMap.put("username", username);
            parametersMap.put("password", password);
            parametersMap.put("name", name);
            parametersMap.put("surname", surname);
            parametersMap.put("email", email);
            registerResult =
NetworkUtil.createPost(ThesisConstants.REGISTER_ACTION_URL,
parametersMap);
        } catch (Exception e) {
            .....
        }
        return 0L;
    }
    .....
    protected void onPostExecute(Long result) {
        progressDialog.dismiss();
        checkRegister();
    }
}

```

#### 4.2.5 Αρχική οθόνη

Αφού ο χρήστης κάνει login οδηγείται στην αρχική σελίδα. Η σελίδα αυτή έχει δυο μορφές. Η πρώτη μορφή εμφανίζεται όταν η εφαρμογή χρησιμοποιείται για πρώτη φορά και δεν έχει δημιουργηθεί ξανά αρχικός πίνακας.



Thesis Panagiotis Athanasopoulos

Home Επιβλέπων About ΣΗΜΜΥ Logout

My favourite songs

Session user name: panos

Δημιουργία Κύριου Πίνακα

Όνομα πίνακα:

Εμφανιζόμενο όνομα πίνακα:

Επαναφορά Υποβολή

EMΠ Σχολή Ηλεκτρολόγων Μηχανικών

Από το χρήστη ζητείται να δώσει ένα όνομα για τον πίνακα που θέλει να δημιουργήσει και ένα εμφανιζόμενο όνομα για αυτόν. Το όνομα πίνακα αντιστοιχεί στο όνομα που θα έχει ο πίνακας στη βάση δεδομένων, για αυτό και πρέπει να είναι σε λατινικούς χαρακτήρες χωρίς κενά και special χαρακτήρες. Το εμφανιζόμενο όνομα μπορεί να είναι κείμενο και αντιστοιχεί με το όνομα που φαίνεται μέσα στην εφαρμογή όταν εμφανίζεται ο πίνακας.

Όταν ο χρήστης δώσει τα δύο πεδία που απαιτούνται και πατήσει το πλήκτρο submit, τότε εκτελείται η action "createTable.php".

```
$external = false;
if (isset($_POST["external"])) {
    $external = MiscUtil::test_input($_POST["external"]);
}

$isMaster = 0;
```

```

    if (isset($_POST["isMaster"])) {
        $isMaster = MiscUtil::test_input($_POST["isMaster"]);
    }

    if (isset($_POST["tableName"])) {
        $tableName = MiscUtil::test_input($_POST["tableName"]);
    }

    if (isset($_POST["tableDisplayName"])) {
        $tableDisplayName =
MiscUtil::test_input($_POST["tableDisplayName"]);
    }

    if (isset($_POST["parentId"])) {
        $parentId = MiscUtil::test_input($_POST["parentId"]);
    }

    //-----VALIDATION START-----//
    if (empty($_POST["tableName"])) {
        $tableErr .= "|Όνομα Πίνακα - Υποχρεωτικό πεδίο";
        $validationError = true;
    } else {
        $tableName = MiscUtil::test_input($_POST["tableName"]);

        $ReferenceDAO = new ReferenceDAO();
        $tableExists = $ReferenceDAO->checkIfTableExists ( $tableName
);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-z_]{4,10}$/", $tableName)) {
            $tableErr .= "|Όνομα Πίνακα - Επιτρεπτή είσοδος μόνο a-
z, ";
            $validationError = true;
        } else if ( $tableExists == 'true' ) {
            $tableErr .= "|Όνομα Πίνακα - Υπάρχει ήδη";
            $validationError = true;
        }
    }

    if (empty($_POST["tableDisplayName"])) {
        $tableErr .= "|Εμφανιζόμενο Όνομα Πίνακα - Υποχρεωτικό
πεδίο";
        $validationError = true;
    } else {
        $tableDisplayName =
MiscUtil::test_input($_POST["tableDisplayName"]);

        $ReferenceDAO = new ReferenceDAO();
        $tableDisplayNameExists = $ReferenceDAO-
>checkIfTableDisplayNameExists ( $tableDisplayName );
        // check if name only contains letters and whitespace
        if (!preg_match("/^[A-Za-z0-9_ -]*$/", $tableDisplayName)) {
            $tableErr .= "|Εμφανιζόμενο Όνομα Πίνακα - Επιτρεπτή
είσοδος μόνο γράμματα και αριθμοί";
            $validationError = true;
        } else if ( $tableDisplayNameExists == 'true' ) {
            $tableErr .= "|Εμφανιζόμενο Όνομα Πίνακα - Υπάρχει ήδη";
            $validationError = true;
        }
    }
}

```

Η δημιουργία του κεντρικού πίνακα με τη δημιουργία οποιουδήποτε πίνακα είναι δυο ίδιες ενέργειες με δυο μικρές διαφορές. Στον κεντρικό πίνακα ορίζεται ως true της κολόνας “is\_master” στη βάση δεδομένων, ενώ στους υπόλοιπους πίνακες ορίζεται το πεδίο “parent\_id” ώστε να φαίνεται με ποιον πίνακα έχουν σχέση. Για το λόγο αυτό και η δημιουργία κεντρικού πίνακα και η δημιουργία απλού πίνακα γίνονται μέσω του ίδιου rph αρχείου.

Παρατηρούμε το διάβασμα των παραμέτρων. Ο έλεγχος αν τα ονόματα των πεδίων του πίνακα τηρούν τους κανόνες που έχουν τεθεί γίνεται με regular expressions. Σημαντικό σημείο ο έλεγχος αν ο πίνακας που ορίζει ο χρήστης υπάρχει ήδη και στην περίπτωση αυτή ενημερώνεται με μήνυμα λάθους ώστε να αλλάξει το όνομα.

Στη συνέχεια καλείται η μέθοδος “createTable” από το αρχείο service “DBAdminService”.

```
public function createTable ( $tableName, $tableDisplayName,
$parentId, $isMaster ) {
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $DBAdminDAO = new DBAdminDAO();
    $ReferenceDAO = new ReferenceDAO();

    //create the new table
    $result = $DBAdminDAO->createTable ( $con, $tableName );

    //add new table in reference
    $newTableId = null;
    if ( $result != 'false' ) {
        $newTableId = $ReferenceDAO->createTable ( $con,
$tableName, $tableDisplayName, $parentId, $isMaster );
        if ( $newTableId == 'false' ) {
            $result = 'false';
        }
    }

    //create main column
    if ( $result != 'false' ) {
        $result = $this->addColumn ( $con, $tableName, 'value',
$tableDisplayName, 'TEXT', 1, 'NULL' );
    }

    if ( $result != 'false' &&
StringUtil::notEmptyOrNull($parentId) && $parentId > 0 ) {
        $parentTable = $ReferenceDAO->findTableById ( $parentId
);
        $result = $this->addColumn ( $con,
$parentTable['table_name'] , $tableName . "_id", $tableDisplayName ,
'REFERENCE', 0, $newTableId );
    }

    //DDL CUSTOM ROLLBACK
    if ( $result == 'false' ) {
        $DBAdminDAO->dropTable ( $con, $tableName );
        $ReferenceDAO->dropColumn ( $con, $tableName, 'value' );
        $ReferenceDAO->dropTable ( $con, $tableName );
        mysqli_commit($con);
    }
}
```

```

        //commit or rollback in case of error
        $transClose = DBHelper::endTransaction( $con , $result );
        if ( $transClose == 'false' ) {
            $result = 'false';
        }

        if ( $result != 'false' ) {
            $result = $newTableId;
        }

        return $result;
    }
}

```

Παρατηρούμε εδώ ότι δημιουργείται ο πίνακας στη βάση και στη συνέχεια τα στοιχεία του δηλώνονται μέσω της ReferenceDAO στον πίνακα “tables\_reference”.

Στη συνέχεια προστίθεται μια κολόνα στον πίνακα με όνομα “value”, η οποία αποτελεί την κεντρική κολόνα του πίνακα. Επίσης στην περίπτωση που ο πίνακας που δημιουργείται είναι ένας referenced πίνακας, τότε δημιουργείται και μια κολόνα στον πίνακα πατέρα με όνομα το όνομα του νέου πίνακα με επίθεμα το “\_id” και χρησιμοποιείται για τη σύνδεση των δεδομένων μεταξύ των πινάκων.

Παρατηρούμε τη μέθοδο “addColumn”

```

public function addColumn ( $con, $tableName, $columnName,
    $columnDisplayName, $columnType, $isMaster, $referenceTableId ) {
    $result = 'true';

    $DBAdminDAO = new DBAdminDAO();
    $ReferenceDAO = new ReferenceDAO();

    $tableId = null;
    $tableReference = $ReferenceDAO->findTableByName ( $con,
    $tableName );
    if ( $tableReference == 'false' || $tableReference == null )
    {
        $result = 'false';
    } else {
        $tableId = $tableReference['id'];
    }

    if ( $result == 'true' ) {
        //add a column to the new table
        if ( $result != 'false' ) {
            $columnDataType = TableUtil::getColumnDataType(
    $columnType );
            $result = $DBAdminDAO->addColumn ( $con, $tableName,
    $columnName, $columnDataType );
        }

        //add a column to the new table reference
        if ( $result != 'false' ) {
            $columnId = $ReferenceDAO->addColumn ( $con,
    $tableId, $columnName, $columnDisplayName, $columnType, $isMaster,
    $referenceTableId );
            if ( $columnId == 'false' ) {
                $result = 'false';
            }
        }
    }
}

```



```
return $result;
}
```

η οποία προσθέτει στον αντίστοιχο πίνακα μια κολόνα και στη συνέχεια σώζει τις σχετικές πληροφορίες με αυτή στον πίνακα "columns\_reference".

Η δεύτερη μορφή της αρχικής οθόνης εμφανίζεται όταν έχει δημιουργηθεί κεντρικός πίνακας και οι υπόλοιποι referenced πίνακες.

Στο επόμενο κομμάτι κώδικα βλέπουμε τον τρόπο με τον οποίο φέρνουμε από τη βάση τον κεντρικό πίνακα και τους referenced πίνακες και τυπώνουμε την HTML για να τα εμφανίσει.

```
$masterTable = $ReferenceDAO->findMasterTable();
echo "<h1><a href=\"\" . $contextRoot . \"table/viewTable.php?tableId=\" .
$masterTable['id'] . \"\" >Προβολή \" .
$masterTable['table_display_name'] . \"</a></h1>\";
echo "<h1 class=\"bold\"><a class=\"tableData\" href=\"\"
.$contextRoot . \"view/viewTableData.php?tableId=\" . $masterTable['id']
. \"\" >Προβολή δεδομένων \" . $masterTable['table_display_name'] .
\"</a></h1>\";
```

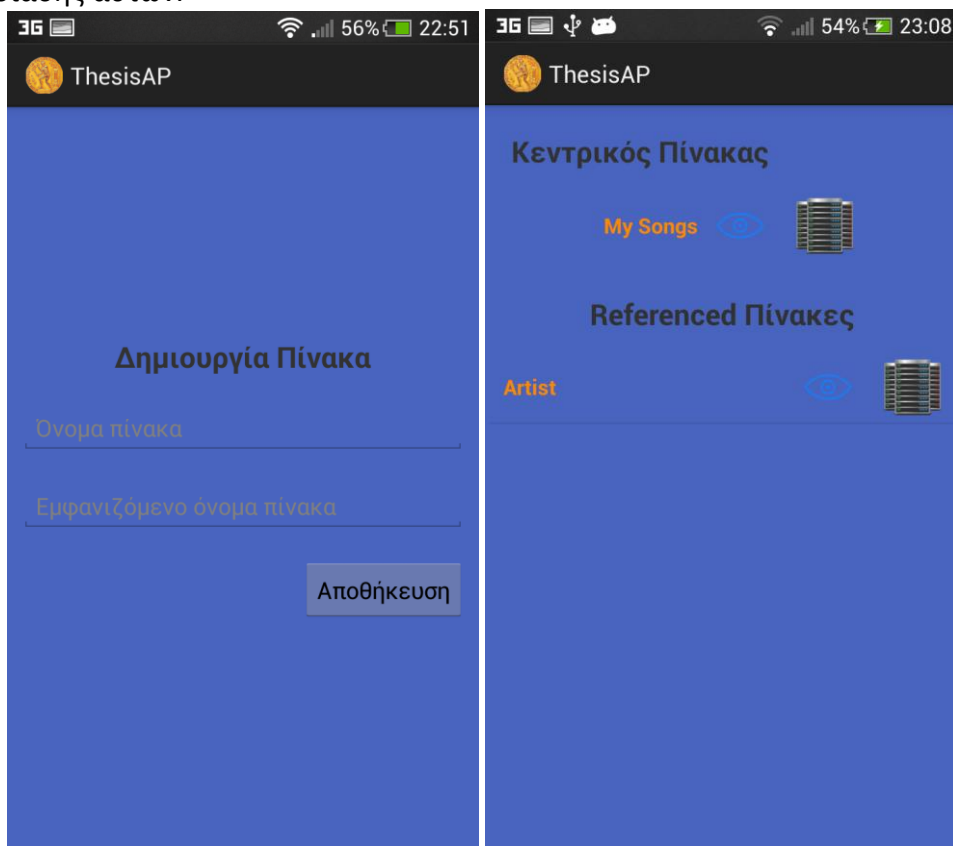
```

echo "<h2>Referenced Πίνακες</h2>";

$referencedTables = $ReferenceDAO->findAllReferencedTables();
while ( $referencedTable = mysqli_fetch_array($referencedTables) ) {
    echo "<h4><a href=\"\" . $contextRoot
    . "table/viewTable.php?tableId=" . $referencedTable['id'] . "\"
    >Προβολή \" . $referencedTable['table display name'] . "</a></h4>";
    echo "<h4 class=\"bold\"><a class=\"tableData\" href=\"\"
    . $contextRoot . "view/viewTableData.php?tableId=" .
    $referencedTable['id'] . "\" >Προβολή δεδομένων \" .
    $referencedTable['table_display_name'] . "</a></h4>";
}

```

Αντίστοιχα στο Android όταν ο χρήστης οδηγηθεί στη Home Page και δεν υπάρχει κεντρικός πίνακας, τότε εμφανίζεται η φόρμα δημιουργίας αυτού. Στην περίπτωση που υπάρχουν πίνακες στο σύστημά μας τότε εμφανίζεται η οθόνη παρουσίασης αυτών.



Παρακάτω θα δούμε τα κομμάτια κώδικα τα οποία πρώτον αποφασίζουν αν θα εμφανιστεί η οθόνη δημιουργίας κεντρικού πίνακα ή η οθόνη παρουσίασης των πινάκων αυτών αντίστοιχα.

```

private void prepareViews() {
    if ( masterTable == null ) {
        setContentView(R.layout.add_table_layout);
    } else {
        setContentView(R.layout.home_activity_layout);
        setTextInTextViewWithId(masterTable.getTableDisplayName(),
R.id.masterTableName);
        progDailog = ProgressDialog.show(this, "", "", true);
        PrepareRefTablesTask task = new PrepareRefTablesTask();
        task.execute("prepareRefTablesTask");
    }
}

```

#### 4.2.6 Προβολή πεδίων πίνακα

Πατώντας την επιλογή προβολής του κάθε πίνακα στην αρχική οθόνη οδηγούμαστε στην οθόνη προβολής λεπτομερειών του αντίστοιχου πίνακα. Εκεί βλέπουμε τη λίστα των πεδίων που έχει ο κάθε πίνακας με τις λεπτομέρειές του.

Session user name: panos

### Τραγούδια

Εμφανιζόμενο όνομα πεδίου	Όνομα πεδίου	Τύπος	
Τραγούδια	value	TEXT	
Music Type	musictype	TEXT	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

### Προσθήκη πεδίου πίνακα

Όνομα πεδίου:	<input type="text"/>
Εμφανιζόμενο όνομα πεδίου:	<input type="text"/>
Τύπος πεδίου:	<input type="text" value="TEXT"/>
<input type="button" value="Επαναφορά"/> <input type="button" value="Υποβολή"/>	

### Δημιουργία Referenced Πίνακα

Όνομα πίνακα:	<input type="text"/>
Εμφανιζόμενο όνομα πίνακα:	<input type="text"/>
<input type="button" value="Επαναφορά"/> <input type="button" value="Υποβολή"/>	

Παρατηρούμε τον πίνακα που παρουσιάζει το όνομα πεδίου, το εμφανιζόμενο όνομα πεδίου, τον τύπο του καθώς και μια κολόνα με ενέργειες που μπορεί να κάνει ο χρήστης πάνω σε αυτά. Στη συνέχεια βλέπουμε το κομμάτι του κώδικα που παρουσιάζει αυτά τα πεδία.

```
<table>
  <thead>
    <th>Εμφανιζόμενο όνομα πεδίου</th>
    <th>Όνομα πεδίου</th>
    <th>Τύπος</th>
    <th></th>
  </thead>
  <tbody>

  <?php
```

```

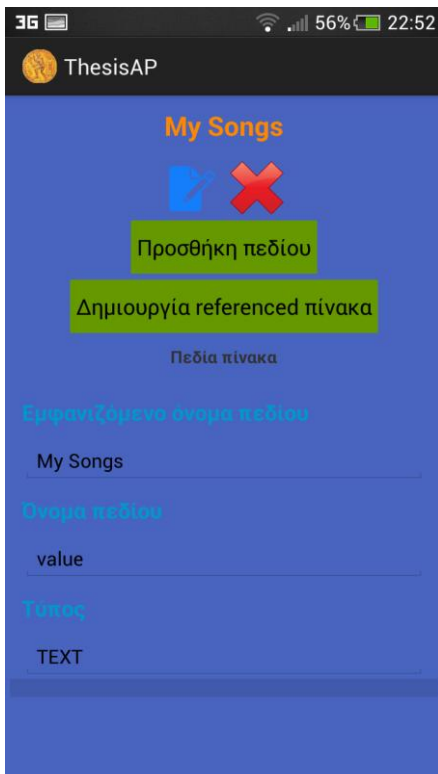
        $tableColumns = $ReferenceDAO->findAllColumnsForTableId (
$tableId );
        while ( $column = mysqli_fetch_array($tableColumns) ) {
            echo "<tr>";
            echo "<td>" . $column['column_display_name'] .
"</td>";
            echo "<td>" . $column['column_name'] . "</td>";
            echo "<td>" . $column['column_type'] . "</td>";
            echo "<td>";
            if ( $column['column_name'] != 'value' &&
$column['column type'] != 'REFERENCE' ) {
                echo " </div><div><a
href=\"editColumn.php?tableId=" . $tableId . "&columnId=" .
$column['id'] . "\">Επεξεργασία</a></div><div><a
href=\"../action/deleteColumn.php?tableId=" . $tableId . "&columnId="
. $column['id'] . "\">Διαγραφή</a></div>";
            }
            echo "</td>";
            echo "</tr>";
        }
    }
?>

</tbody>
</table>

```

Παρατηρούμε ότι από το αρχείο ReferenceDAO ζητάμε όλες τις κολόνες του πίνακα με Id αυτού που έχει ζητήσει ο χρήστης. Τυπώνουμε έναν HTML πίνακα στον οποίο κάθε γραμμή αντιστοιχεί σε ένα πεδίο του πίνακά μας.

Αντίστοιχα στο Android όταν ο χρήστης πατήσει το πλήκτρο  οδηγείται στην οθόν



η παρουσίαση των πεδίων του πίνακα όπως φαίνεται στην παρακάτω εικόνα.

Καθώς τα δεδομένα αυτά είναι δυναμικά και άγνωστος ο αριθμός τους στον προγραμματιστή, εμφανίζονται μέσα σε μια ListView που είναι στο Android το αντίστοιχο του HTML table.

Για να λειτουργήσει μια ListView χρειάζεται έναν “Adapter”, ο οποίος κώδικας αναλαμβάνει να εμφανίσει στο κάθε “κελί” της ListView τα δεδομένα που αντιστοιχούν στη θέση που εμφανίζεται.

Αφού εμφανιστούν τα δεδομένα ο χρήστης θα μπορεί με την κάθετη κίνηση του δακτύλου του να οδηγηθεί σε επόμενα ή προηγούμενα πεδία του πίνακα.

Στη συνέχεια ακολουθεί το πιο σημαντικό κομμάτι του κώδικα του adapter που αναλαμβάνει να εμφανίσει τα πεδία στην οθόνη.

```
List<Column> dataList = null;
```

```
@Override
public View getView (int position , View convertView, ViewGroup
parent ) {

    View row = convertView;
    RowHolder holder = null;

    if ( row == null ) {
        //LayoutInflater inflater = ((Activity)
context).getLayoutInflater();
        LayoutInflater inflater = (LayoutInflater)
getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        row = inflater.inflate(R.layout.column_row_layout, parent,
false);

        holder = new RowHolder();
        holder.cell_columnDisplayName = (EditText)
row.findViewById(R.id.columnDisplayName);
        holder.cell_columnName = (EditText)
row.findViewById(R.id.columnName);
        holder.cell_columnType = (EditText)
row.findViewById(R.id.columnType);
        holder.cell_deleteColumnButton = (Button)
row.findViewById(R.id.deleteColumnButton);
        holder.cell_editColumnButton = (Button)
row.findViewById(R.id.editColumnButton);

        row.setTag(holder);
    } else {
        holder = (RowHolder) row.getTag();
    }

    Column column = dataList.get(position); //data [position];


    holder.cell_columnDisplayName.setText (
column.getColumnDisplayName() );
    holder.cell_columnName.setText ( column.getColumnName() );
    holder.cell_columnType.setText ( column.getColumnType() );

    if ( column.getColumnType().equals("REFERENCE") ||
column.getColumnName().equals("value") ) {
        holder.cell_deleteColumnButton.setVisibility(View.GONE);
        holder.cell_editColumnButton.setVisibility(View.GONE);
    } else {
        holder.cell_deleteColumnButton.setVisibility(View.VISIBLE);
        holder.cell_editColumnButton.setVisibility(View.VISIBLE);
    }

    return row;
}
```

Παρατηρούμε ότι από τη λίστα με τις κολόνες (η οποία έχει γεμίσει με τα δεδομένα από τον πίνακα “tables\_reference”) κάθε φορά παίρνουμε τα δεδομένα από αυτή που αντιστοιχεί στο πεδίο που πρόκειται να εμφανιστεί στην οθόνη και γεμίζουμε τα αντίστοιχα widgets.

#### 4.2.7 Επεξεργασία ονόματος πίνακα

Ο χρήστης έχει τη δυνατότητα να αλλάξει το εμφανιζόμενο όνομα ενός πίνακα οποιαδήποτε στιγμή επιθυμεί. Πατώντας το πλήκτρο  οδηγείται στη σελίδα όπου μπορεί να δώσει ένα καινούριο όνομα.



The screenshot shows a web application interface. At the top, there is a dark navigation bar with the text "Thesis Panagiotis Athanasopoulos" on the left and menu items "Home", "Επιβλέπων", "About", "ΣΗΜΜΥ", and "Logout" on the right. Below the navigation bar is a large green banner with the text "My favourite songs" in white. To the left of the banner is an illustration of a globe with four computer mice connected to it by green wires. Below the banner is a white section titled "Επεξεργασία Ονόματος πίνακα". This section contains a form with a label "Εμφανιζόμενο όνομα πίνακα:" and a text input field containing the word "Τραγούδια". Below the input field are two buttons: "Επαναφορά" and "Υποβολή". At the bottom of the page is a green footer bar containing the logo of the National Technical University of Athens (NTUA) and the text "ΕΜΠ Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών".

Η μετονομασία ενός πίνακα έχει μια ιδιαίτερη λογική καθώς δεν μετονομάζεται μόνο ο πίνακας. Όπως έχει σχεδιαστεί η εφαρμογή μας, πρώτον ο κάθε πίνακας όταν δημιουργείται αποκτά μία κολόνα με όνομα "value" και εμφανιζόμενο όνομα το όνομα του πίνακα, δεύτερον όταν δημιουργείται ένας referenced πίνακας δημιουργείται αυτόματα και μια κολόνα στον πίνακα "πατέρα" με εμφανιζόμενο όνομα αυτό του νέου πίνακα. Οπότε στην περίπτωση που αλλάξουμε το όνομα ενός πίνακα πρέπει αυτόματα να αλλάξει και το εμφανιζόμενο όνομα της κεντρικής του κολόνας καθώς και το όνομα της κολόνας στον πίνακα "πατέρα" (στην περίπτωση που πρόκειται για referenced πίνακα). Στη συνέχεια βλέπουμε το πιο σημαντικό κομμάτι κώδικα αυτής της λειτουργίας.

```

public function editTableColumnName ( $tableId, $columnId,
$columnDisplayName ) {
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $ReferenceDAO = new ReferenceDAO();

    $result = $ReferenceDAO->changeColumnName ( $con,
$columnId, $columnDisplayName );

    //commit or rollback in case of error
    $result = DBHelper::endTransaction( $con , $result );

    return $result;
}

public function editTableDisplayName ( $tableId, $tableDisplayName )
{
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $ReferenceDAO = new ReferenceDAO();

    $ReferenceDAO->changeTableDisplayName ( $con, $tableId,
$tableDisplayName );

    //change master column display name
    if ( $result == 'true' ) {
        $column = $ReferenceDAO->findTableColumnByName ( $tableId,
'value' );
        $result = $this->editTableColumnName ( $tableId,
$column['id'], $tableDisplayName );
    }


    //Change parent table reference column name
    if ( $result == 'true' ) {
        $table = $ReferenceDAO->findTableById ( $tableId );
        if ( ($table['is_master'] != '1') && ($table['parent_id'] >
0) ) {
            $column = $ReferenceDAO->findTableColumnByName (
$table['parent_id'], $table['table_name'] . "_id" );
            $result = $this->editTableColumnName ( $tableId,
$column['id'], $tableDisplayName );
        }
    }

    //commit or rollback in case of error
    $result = $result = DBHelper::endTransaction( $con , $result );

    return $result;
}

```

Αρχικά αλλάζει το εμφανιζόμενο όνομα του πίνακα, στη συνέχεια αλλάζει το όνομα της κεντρικής κολόνας. Τέλος αν ο πίνακας είναι “referenced” και όχι ο κεντρικός, τότε γίνεται η αλλαγή του ονόματος της κολόνας του πίνακα “πατέρα” που κάνει τη σύνδεση μεταξύ αυτών των πινάκων.

Αντίστοιχα στο Android όταν ο χρήστης πατήσει το πλήκτρο  οδηγείται στην οθόνη επεξεργασίας του ονόματος ενός πίνακα.



Όπως και στην περίπτωση της web εφαρμογής ο χρήστης διορθώνει το όνομα για τον πίνακα που έχει επιλέξει και πατάει το πλήκτρο αποθήκευση.


Με το πάτημα του πλήκτρου αυτού γίνεται μία κλήση στην αντίστοιχη action rph στο server που αναφέρθηκε προηγουμένως και γίνεται η αλλαγή του εμφανιζόμενου ονόματος.

Στον ακόλουθο κομμάτι κώδικα βλέπουμε την κλήση με το νέο όνομα που γίνεται προς το server. Η διαδικασία αυτή έχει πολλά κοινά με την αλλαγή ονόματος πεδίου του πίνακα, οπότε φαίνεται ο διαχωρισμός της εκάστοτε κλήσης στον κοινό κώδικα.

```
private class EditDisplayNameTask extends AsyncTask<String, Integer,
Long> {
...
    protected Long doInBackground(String... progress) {
        try {
            String url = ThesisConstants.EDIT_TABLE_NAME_ACTION;
            String displayNameParamName = "tableDisplayName";
            Map<String,String> parametersMap = new HashMap<String,
String>();
            parametersMap.put("external", "true");
            parametersMap.put("tableId", tableId.toString());
            if ( columnId > 0 ) {
                parametersMap.put("columnId", columnId.toString());
                url = ThesisConstants.ADD_EDIT_COLUMN_ACTION_URL;
                displayNameParamName = "columnDisplayName";
            }
            parametersMap.put(displayNameParamName,
getTextFromEditText(R.id.displayNameET));
            editDisplayNameActionResutl = NetworkUtil.createPost(url,
parametersMap);
        } catch (Exception e) {
            Log.e(ThesisConstants.THEISIS_TAG, "ERROR", e);
        }
        return 0L;
    }
...
}
```



#### 4.2.8 Διαγραφή πίνακα

Δίνεται η δυνατότητα στο χρήστη να διαγράψει έναν πίνακα. Ο περιορισμός που υπάρχει είναι ότι για να εμφανιστεί το κουμπί διαγραφής πρέπει ο πίνακας αυτός να μην έχει δεμένους πάνω του referenced πίνακες καθώς η διαγραφή του θα άφηνε “ορφανούς” πίνακες που δε θα είχαν καμία χρησιμότητα. Ο χρήστης πατώντας το πλήκτρο  δίνει την εντολή διαγραφής.

Session user name: panos

### Τραγούδια

Εμφανιζόμενο όνομα πεδίου	Όνομα πεδίου	Τύπος	
Τραγούδια	value	TEXT	
Music Type	musictype	TEXT	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

### Προσθήκη πεδίου πίνακα

Όνομα πεδίου:

Εμφανιζόμενο όνομα πεδίου:

Τύπος πεδίου:

### Δημιουργία Referenced Πίνακα

Όνομα πίνακα:

Εμφανιζόμενο όνομα πίνακα:

Στη συνέχεια βλέπουμε το κομμάτι του κώδικα που διαχειρίζεται τη διαγραφή πίνακα.

```
public function deleteTable ( $tableId ) {  
    $result = 'true';  
    $con = DBHelper::beginTransaction();  
  
    $DBAdminDAO = new DBAdminDAO();  
    $ReferenceDAO = new ReferenceDAO();
```

```

    $table = $ReferenceDAO->findTableById ( $tableId );
    $tableName = $table['table_name'];

    $parentId = $table['parent_id'];
    $parentColumnName = $table['table_name'] . "_id";
    $parentTableColumn = $ReferenceDAO->findTableColumnByName (
$parentId, $parentColumnName );
    if ( $parentTableColumn != null ) {
        $result = $this->deleteTableColumn ( $parentId,
$parentTableColumn['id'] );
    }

    if ( $result == 'true' ) {
        $result = $ReferenceDAO->dropAllTableColumns ( $con, $tableId
);
    }

    if ( $result == 'true' ) {
        $result = $ReferenceDAO->dropTable ( $con, $tableName );
    }

    if ( $result == 'true' ) {
        $result = $DBAdminDAO->dropTable ( $con, $tableName );
    }

    //commit or rollback in case of error
    $result = DBHelper::endTransaction( $con , $result );

    return $result;
}

public function deleteTableColumn ( $tableId, $columnId ) {
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $DBAdminDAO = new DBAdminDAO();
    $ReferenceDAO = new ReferenceDAO();

    $table = $ReferenceDAO->findTableById ( $tableId );
    $tableName = $table['table_name'];
    $column = $ReferenceDAO->findColumnById( $columnId );
    $columnName = $column['column_name'];

    $result = $ReferenceDAO->dropColumn ( $con, $tableId, $columnId
);

    if ( $result == 'true' ) {
        $result = $DBAdminDAO->dropColumn ( $con, $tableName,
$columnName );
    }


    //commit or rollback in case of error
    $result = DBHelper::endTransaction( $con , $result );

    return $result;
}

```

Στον παραπάνω κώδικα παρατηρούμε ότι για να διαγραφεί ένας πίνακας αρχικά ελέγχουμε αν αυτός ο πίνακας είναι referenced κάποιου άλλου, οπότε σβήνουμε την κολόνα σύνδεσης από τον πίνακα “πατέρα”. Στη συνέχεια διαγράφονται από



τον πίνακα “columns\_reference” όλες οι γραμμές που έχουν σχέση με πεδία του υπο-διαγραφή πίνακα, μετά διαγράφεται από τον πίνακα “tables\_reference” η εγγραφή του πίνακα αυτού και τέλος αφαιρείται από το σχήμα της βάσης μας ο εν λόγω πίνακας.

Αντίστοιχα στο Android ο χρήστης πατώντας το ίδιο πλήκτρο  στέλνει την εντολή στο server για τη διαγραφή του πίνακα. Ομοίως με τη web εφαρμογή το πλήκτρο αυτό εμφανίζεται μόνο όταν ο πίνακας δεν έχει άλλους referenced πίνακες.

#### 4.2.9 Προσθήκη πεδίου πίνακα

Όταν ο χρήστης βρίσκεται στη σελίδα της προβολής ενός πίνακα έχει τη δυνατότητα να προσθέσει ένα νέο πεδίο σε αυτόν. Βάζοντας το όνομα του πεδίου, το εμφανιζόμενο όνομα αυτού καθώς και τον τύπο ( μεταξύ των προετοιμασμένων τύπων (κείμενο, ακέραιος αριθμός, double αριθμός, συντεταγμένες και ημερομηνία).

**Session user name: panos**

**Τραγούδια**  

Εμφανιζόμενο όνομα πεδίου	Όνομα πεδίου	Τύπος	
Τραγούδια	value	TEXT	
Music Type	musictype	TEXT	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

---

**Προσθήκη πεδίου πίνακα**

Όνομα πεδίου:	<input type="text"/>
Εμφανιζόμενο όνομα πεδίου:	<input type="text"/>
Τύπος πεδίου:	<input type="text" value="TEXT"/>

---

**Δημιουργία Referenced Πίνακα**

Όνομα πίνακα:	<input type="text"/>
Εμφανιζόμενο όνομα πίνακα:	<input type="text"/>

Στη συνέχεια ακολουθεί το κομμάτι κώδικα που διαχειρίζεται την προσθήκη πεδίου σε έναν πίνακα.

```

public function addColumn ( $con, $tableName, $columnName,
    $columnDisplayName, $columnType, $isMaster, $referenceTableId ) {
    $result = 'true';

    $DBAdminDAO = new DBAdminDAO ();
    $ReferenceDAO = new ReferenceDAO ();

    $tableId = null;
    $tableReference = $ReferenceDAO->findTableByName ( $con,
    $tableName );
    if ( $tableReference == 'false' || $tableReference == null )
    {
        $result = 'false';
    } else {
        $tableId = $tableReference['id'];
    }

    if ( $result == 'true' ) {
        //add a column to the new table
        if ( $result != 'false' ) {
            $columnDataType = TableUtil::getColumnDataType (
    $columnType );
            $result = $DBAdminDAO->addColumn ( $con, $tableName,
    $columnName, $columnDataType );
        }

        //add a column to the new table reference
        if ( $result != 'false' ) {
            $columnId = $ReferenceDAO->addColumn ( $con,
    $tableId, $columnName, $columnDisplayName, $columnType, $isMaster,
    $referenceTableId );
            if ( $columnId == 'false' ) {
                $result = 'false';
            }
        }
    }

    return $result;
}

```

Παρατηρούμε ότι αρχικά γράφεται το νέο πεδίο στον πίνακα “columns\_reference” με τις πληροφορίες του και στη συνέχεια προστίθεται η αντίστοιχη κολόνα στον πίνακα. Σημαντικό μέρος είναι η γραμμή:

```
$columnDataType = TableUtil::getColumnDataType ( $columnType );
```

όπου ανάλογα με τον τύπο πεδίου που έχει διαλέξει ο χρήστης να προσθέσει, τότε η βοηθητική μέθοδος από το αρχείο TableUtil φέρνει την αντιστοίχιση σε σχήμα βάσης. Ακολουθεί η βοηθητική μέθοδος.

```

public static function getColumnTypes () {
    $types['TEXT'] = 'varchar (255)';
    $types['INTEGER'] = 'int (11)';
    $types['DOUBLE'] = 'double';
    $types['LOCATION'] = 'varchar (100)';
    $types['DATE'] = 'datetime';
    $types['REFERENCE'] = 'int (11)';
}

```

```

        //$types['timestamp'] = 'timestamp';

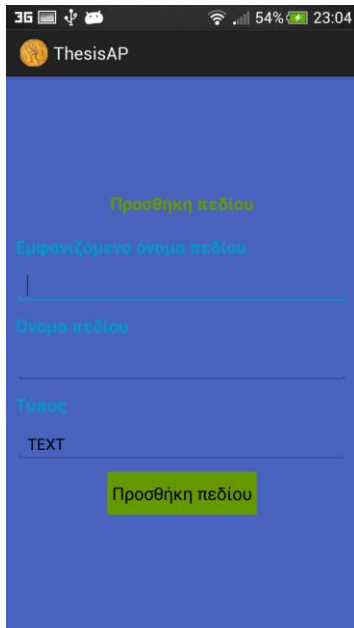
        return $types;
    }

    public static function getColumnDataType ( $columnType ) {
        $result = TableUtil::getColumnTypes () [$columnType];
        return $result;
    }

```

Αντίστοιχα στο Android ο χρήστης στην οθόνη όπου βλέπει την οθόνη με τα πεδία πίνακα, μπορεί να πατήσει το πλήκτρο “Προσθήκη πεδίου” και να οδηγηθεί στην οθόνη όπου εμφανίζεται η φόρμα με τα πεδία του πίνακα. Εκεί ο χρήστης δίνει τα στοιχεία της εγγραφής που θέλει να προσθέσει και πατώντας το πλήκτρο “Αποθήκευση πεδίου” τα στοιχεία αυτά στέλνονται στο server.

Ακολουθεί τα κομμάτι του κώδικα της αποστολής των στοιχείων στο server.



```

private class AddColumnTask extends AsyncTask<String, Integer, Long> {
    .....
    try {
        Map<String,String> parametersMap = new HashMap<String, String>();
        parametersMap.put("external", "true");
        parametersMap.put("tableId", tableId.toString());
        parametersMap.put("columnType", getTextFromEditText(R.id.columnType));
        parametersMap.put("columnName", getTextFromEditText(R.id.columnName));
        parametersMap.put("columnDisplayName", getTextFromEditText(R.id.columnDisplayName));
        actionResult = NetworkUtil.createPost(ThesisConstants.ADD_EDIT_COLUMN_ACTION_URL,
parametersMap);
    } catch (Exception e) { }
    return 0L;
}

protected void onProgressUpdate(Integer... progress) {
}

protected void onPostExecute(Long result) {
    progDailog.dismiss();
    checkAddColumn();
}
}

```

#### 4.2.10 Επεξεργασία ονόματος πεδίου πίνακα

Στο κάθε πεδίο πίνακα, πλυν του κεντρικού και όσων προέρχονται από referenced πίνακες, μπορεί ο χρήστης να αλλάξει το εμφανιζόμενο όνομα οποιαδήποτε στιγμή το επιθυμεί. Πηγαίνοντας στην οθόνη παρουσίασης πεδίων πίνακα, δίπλα στο κάθε πεδίο υπάρχει η επιλογή επεξεργασίας η οποία οδηγεί στην αντίστοιχη οθόνη.

Thesis Panagiotis  
Athanasopoulos

Home Επιβλέπων About ΣΗΜΜΥ Logout

My favourite songs

Επεξεργασία πεδίου πίνακα

Εμφανιζόμενο όνομα πεδίου:

EMΠ Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

© Copyright Athanasopoulos Panagiotis. Layout by | Website Templates

Ακολουθεί το κομμάτι κώδικα στο οποίο γίνεται η μετονομασία. Στην πραγματικότητα το μόνο που χρειάζεται να αλλάξει κατά τη μετονομασία ενός πεδίου είναι απλά το όνομά του στον πίνακα "columns\_reference".

```
public function changeColumnName ( $con, $columnId,
    $columnDisplayName ) {

    $result = 'false';

    $query = " UPDATE columns_reference cr SET
cr.column_display_name = '" . $columnDisplayName . "' WHERE cr.id =
'" . $columnId . "'";
    //die($query);
    if ( false === ($execute = mysqli_query($con,$query)) ) {
        echo mysql_error();
    }
}
```

```

        $result = 'false';
    } else {
        $result = 'true';
    }

    return $result;
}

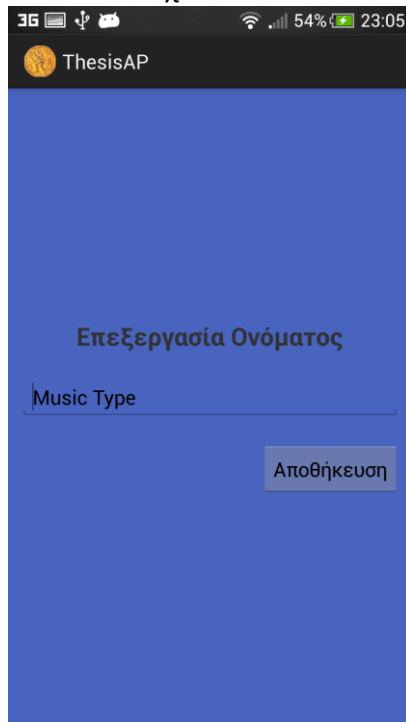
```

Τονίζουμε ότι σε ένα υπάρχων πεδίο πίνακα **δεν** επιτρέπουμε την αλλαγή τύπου καθώς αυτό θα δημιουργούσε πρόβλημα στις υπάρχοντες τιμές οι οποίες, πιθανότατα, με την αλλαγή δεν θα ταίριαζαν στους κανόνες του νέου τύπου.

Αντίστοιχα στο Android ο χρήστης έχει τη δυνατότητα να διορθώσει το όνομα ενός πεδίου το πίνακα οποιαδήποτε στιγμή θελήσει (εκτός αν, όπως και στη web εφαρμογή, πρόκειται για την κεντρική κολόνα του πίνακα ή για κολόνα σύνδεσης με referenced πίνακα).

Ακολουθεί το κομμάτι του κώδικα στο οποίο αποστέλεται στο server το όνομα του νέου πεδίου καθώς και το id αυτού το οποίο αλλάζει.

Όπως έχει αναφερθεί ξανά η διαδικασία αυτή έχει πολλές ομοιότητες με την αλλαγή ονόματος πίνακα, έτσι στο κοινό κομμάτι κώδικα ορίζοντας το id της προς αλλαγή κολόνας, αποφασίζεται ότι θα χρησιμοποιηθεί το αρχείο του server που αλλάζει το όνομα πεδίου.



```



private class EditDisplayNameTask extends AsyncTask<String, Integer, Long> {
...
    protected Long doInBackground(String... progress) {
        try {
            String url = ThesisConstants.EDIT_TABLE_NAME_ACTION;
            String displayNameParamName = "tableDisplayName";
            Map<String,String> parametersMap = new HashMap<String, String>();
            parametersMap.put("external", "true");
            parametersMap.put("tableId", tableId.toString());
            if (columnId > 0) {
                parametersMap.put("columnId", columnId.toString());
                url = ThesisConstants.ADD_EDIT_COLUMN_ACTION_URL;
                displayNameParamName = "columnDisplayName";
            }
            parametersMap.put(displayNameParamName, getTextFromEditText(R.id.displayNameET));
            editDisplayNameActionResult = NetworkUtil.createPost(url, parametersMap);
        } catch (Exception e) {
            Log.e(ThesisConstants.THESIS_TAG, "ERROR", e);
        }
        return 0L;
    }
}

```

#### 4.2.11 Διαγραφή πεδίου πίνακα

Στην οθόνη εμφάνισης πεδίων πίνακα ο χρήστης έχει τη δυνατότητα να διαγράψει κάποια από αυτά (εκτός αν πρόκειται για το κεντρικό πεδίο ή για κολόνα σύνδεσης με referenced πίνακα όπου θα πρέπει να κάνει διαγραφή πίνακα). Δίπλα από το κάθε πεδίο που μπορεί να διαγραφεί υπάρχει το κουμπί “Διαγραφή” και πατώντας το ο χρήστης εκτελεί τη διαδικασία αυτή.

**Session user name: panos**

**Τραγούδια**  

Εμφανιζόμενο όνομα πεδίου	Όνομα πεδίου	Τύπος	
Τραγούδια	value	TEXT	
Music Type	musictype	TEXT	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

---

**Προσθήκη πεδίου πίνακα**

Όνομα πεδίου:	<input type="text"/>
Εμφανιζόμενο όνομα πεδίου:	<input type="text"/>
Τύπος πεδίου:	<input type="text" value="TEXT"/>

---

**Δημιουργία Referenced Πίνακα**

Όνομα πίνακα:	<input type="text"/>
Εμφανιζόμενο όνομα πίνακα:	<input type="text"/>

Η διαδικασία διαγραφής πεδίου φαίνεται στο ακόλουθο κομμάτι κώδικα:

```
public function deleteTableColumn ( $tableId, $columnId ) {  
    $result = 'true';  
    $con = DBHelper::beginTransaction();  
  
    $DBAdminDAO = new DBAdminDAO();  
    $ReferenceDAO = new ReferenceDAO();  
  
    $table = $ReferenceDAO->findTableById ( $tableId );  
    $tableName = $table['table_name'];  
    $column = $ReferenceDAO->findColumnById ( $columnId );  
    $columnName = $column['column_name'];
```



```

$result = $ReferenceDAO->dropColumn ( $con, $tableId, $columnId
);

if ( $result == 'true' ) {
    $result = $DBAdminDAO->dropColumn ( $con, $tableName,
    $columnName );;
}

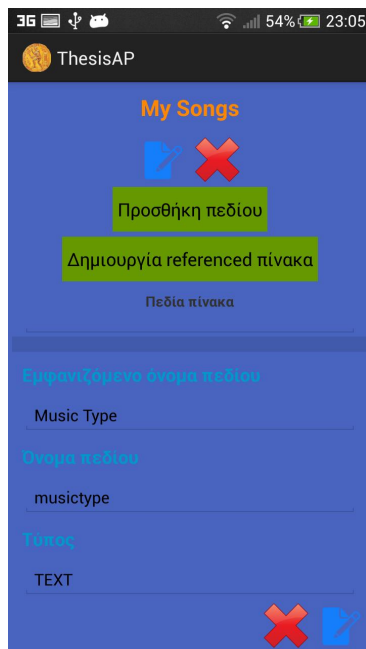
//commit or rollback in case of error
$result = DBHelper::endTransaction( $con , $result );

return $result;
}


```

Παρατηρούμε ότι αρχικά αφαιρείται η εγγραφή του πεδίου στον πίνακα “columns\_reference” και στη συνέχεια διαγράφεται η κολόνα από τον ίδιο τον πίνακα στον οποίο ανήκει.

Όμοιος με τη web εφαρμογή, ο χρήστης στην εφαρμογή του Android μπορεί να



διαγράψει μια κολόνα, αρκεί αυτή να μην είναι η κεντρική καθώς και να μην πρόκειται για κολόνα σύνδεσης με referenced πίνακα.

Πατώντας πάνω στο πλήκτρο  πάνω στο αντίστοιχο πεδίο, στέλνεται η εντολή στο server για τη διαγραφή του πεδίου που έχει επιλεγεί.

Ακολουθεί ο κώδικας που στέλνει την εντολή στο server για τη διαγραφή.


```

private class DeleteColumnTask extends AsyncTask<String, Integer, Long> {
...
protected Long doInBackground(String... progress) {
    try {
        Map<String,String> parametersMap = new HashMap<String, String>();
        parametersMap.put("external", "true");
        parametersMap.put("tableId", tableId.toString());
        parametersMap.put("columnId", selectedColumnId.toString());
        String url = NetworkUtil.createUrlWithParameters(ThesisConstants.DELETE_COLUMN_ACTION,
        parametersMap);
        deleteColumnActionResult = NetworkUtil.createGet(url);
    } catch (Exception e) { }
    return 0L;
}
...
}

```

#### 4.2.12 Προσθήκη Referenced πίνακα

Εκτός από την προσθήκη πεδίου πίνακα, δίνεται η δυνατότητα στο χρήστη να προσθέσει και άλλων συνδεδεμένων πινάκων για την καλύτερη οργάνωση των δεδομένων του. Στην οθόνη παρουσίασης πεδίων πίνακα υπάρχει η φόρμα δημιουργίας referenced πίνακα

Τραγούδια 

Εμφανιζόμενο όνομα πεδίου	Όνομα πεδίου	Τύπος	
Τραγούδια	value	TEXT	
Music Type	musictype	TEXT	<a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Artist	artist_id	REFERENCE	

#### Προσθήκη πεδίου πίνακα

Όνομα πεδίου:	<input type="text"/>
Εμφανιζόμενο όνομα πεδίου:	<input type="text"/>
Τύπος πεδίου:	TEXT ▼
<input type="button" value="Επαναφορά"/> <input type="button" value="Υποβολή"/>	

#### Δημιουργία Referenced Πίνακα

Όνομα πίνακα:	<input type="text"/>
Εμφανιζόμενο όνομα πίνακα:	<input type="text"/>
<input type="button" value="Επαναφορά"/> <input type="button" value="Υποβολή"/>	



ΕΜΠ

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών

Παρατηρούμε ότι με τη δημιουργία ενός referenced πίνακα, δημιουργείται μια κολόνα στον πίνακα “πατέρα” με όνομα το όνομα του πίνακα που προστίθεται μαζί με το επίθεμα “\_id”. Αυτή η κολόνα είναι foreign key προς το νέο πίνακα έτσι ώστε να φαίνεται σε ποια εγγραφή του νέου πίνακα αντιστοιχεί μια εγγραφή του πίνακα “πατέρα”. Στην περίπτωση μας η νέα κολόνα έχει όνομα “artist\_id” εκ του πίνακα “artist”. Επίσης όπως φαίνεται και έχει αναφερθεί ανωτέρω, ο χρήστης δεν έχει τη δυνατότητα να διαγράψει απευθείας αυτό το πεδίο, θα πρέπει να διαγράψει τον πίνακα στον οποίο αναφέρεται.

Οι κανόνες που ακολουθούνται κατά τη δημιουργία ενός referenced πίνακα μοιάζουν σε μεγάλο βαθμό με τους κανόνες της δημιουργίας κύριου πίνακα. Η ουσιαστική διαφορά τους είναι ότι σε αυτή την περίπτωση η κολόνα “is\_master”

στον πίνακα “tables\_reference” παίρνει τιμή 0 και ότι δημιουργείται κολόνα σύνδεσης στον πίνακα “πατέρα”. Στο ακόλουθο κομμάτι κώδικα

```
public function createTable ( $tableName, $tableDisplayName,
$parentId, $isMaster ) {
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $DBAdminDAO = new DBAdminDAO();
    $ReferenceDAO = new ReferenceDAO();

    //create the new table
    $result = $DBAdminDAO->createTable ( $con, $tableName );

    //add new table in reference
    $newTableId = null;
    if ( $result != 'false' ) {
        $newTableId = $ReferenceDAO->createTable ( $con, $tableName,
$tableDisplayName, $parentId, $isMaster );
        if ( $newTableId == 'false' ) {
            $result = 'false';
        }
    }

    //create main column
    if ( $result != 'false' ) {
        $result = $this->addColumn ( $con, $tableName, 'value',
$tableDisplayName, 'TEXT', 1, 'NULL' );
    }

    if ( $result != 'false' && StringUtil::notEmptyOrNull($parentId)
&& $parentId > 0 ) {
        $parentTable = $ReferenceDAO->findTableById ( $parentId );
        $result = $this->addColumn ( $con, $parentTable['table_name']
, $tableName . "_id", $tableDisplayName , 'REFERENCE', 0, $newTableId
);
    }

    //DDL CUSTOM ROLLBACK
    if ( $result == 'false' ) {
        $DBAdminDAO->dropTable ( $con, $tableName );
        $ReferenceDAO->dropColumn ( $con, $tableName, 'value' );
        $ReferenceDAO->dropTable ( $con, $tableName );
        mysqli_commit($con);
    }

    //commit or rollback in case of error
    $transClose = DBHelper::endTransaction( $con , $result );
    if ( $transClose == 'false' ) {
        $result = 'false';
    }

    if ( $result != 'false' ) {
        $result = $newTableId;
    }

    return $result;
}
```

παρατηρούμε τη δημιουργία του νέου πίνακα, την εγγραφή του στον πίνακα “tables\_reference”, την προσθήκη κύριου πεδίου στον πίνακα αυτό και στην περίπτωση μας που πρόκειται για referenced πίνακα τη δημιουργία της κολόνας σύνδεσης στον πίνακα “πατέρα”.

Σε όλες τις περιπτώσεις δημιουργίας κολόνας καλείται η μέθοδος service που αναλαμβάνει να δημιουργήσει μια κολόνα καθώς αυτή περιέχει όλη τη λογική αυτής της πράξης, όπως την εγγραφή της στον πίνακα “columns\_reference”.

Επίσης μετά τη δημιουργία του referenced πίνακα παρατηρούμε στην αρχική οθόνη την εμφάνισή του στη λίστα με τους referenced πίνακες.

Thesis Panagiotis  
Athanasopoulos

Home Επιβλέπων About ΣΗΜΜΥ Logout

My favourite songs

Session user name: panos

Προβολή Τραγούδια

Προβολή δεδομένων Τραγούδια

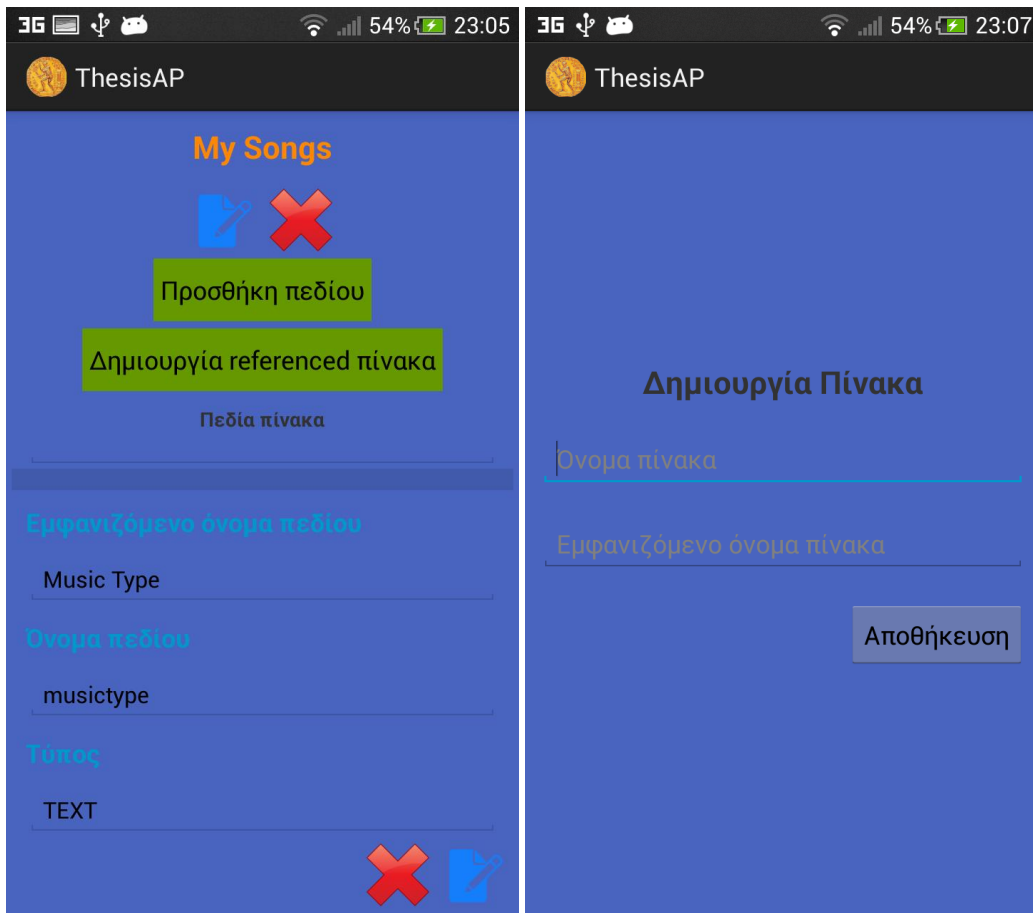
Referenced Πίνακες

[Προβολή Artist](#)

[Προβολή δεδομένων Artist](#)

ΕΜΠ Σχολή Ηλεκτρολόγων Μηχανικών

Αντίστοιχα με τη λειτουργία της web εφαρμογής ο χρήστης μέσα από την εφαρμογή του Android μπορεί να δημιουργήσει έναν referenced πίνακα. Πατώντας το πλήκτρο “Δημιουργία referenced πίνακα” στην οθόνη προβολής πεδίων πίνακα οδηγείται στην οθόνη προσθήκης νέου πίνακα όπου καλείται να δώσει το όνομα και το εμφανιζόμενο όνομα πίνακα. Πατώντας το πλήκτρο “Αποθήκευση” τα στοιχεία αποστέλλονται στο server για τη δημιουργία του νέου πίνακα. Ακολουθεί το κομμάτι του κώδικα που στέλνει τα στοιχεία στο server.



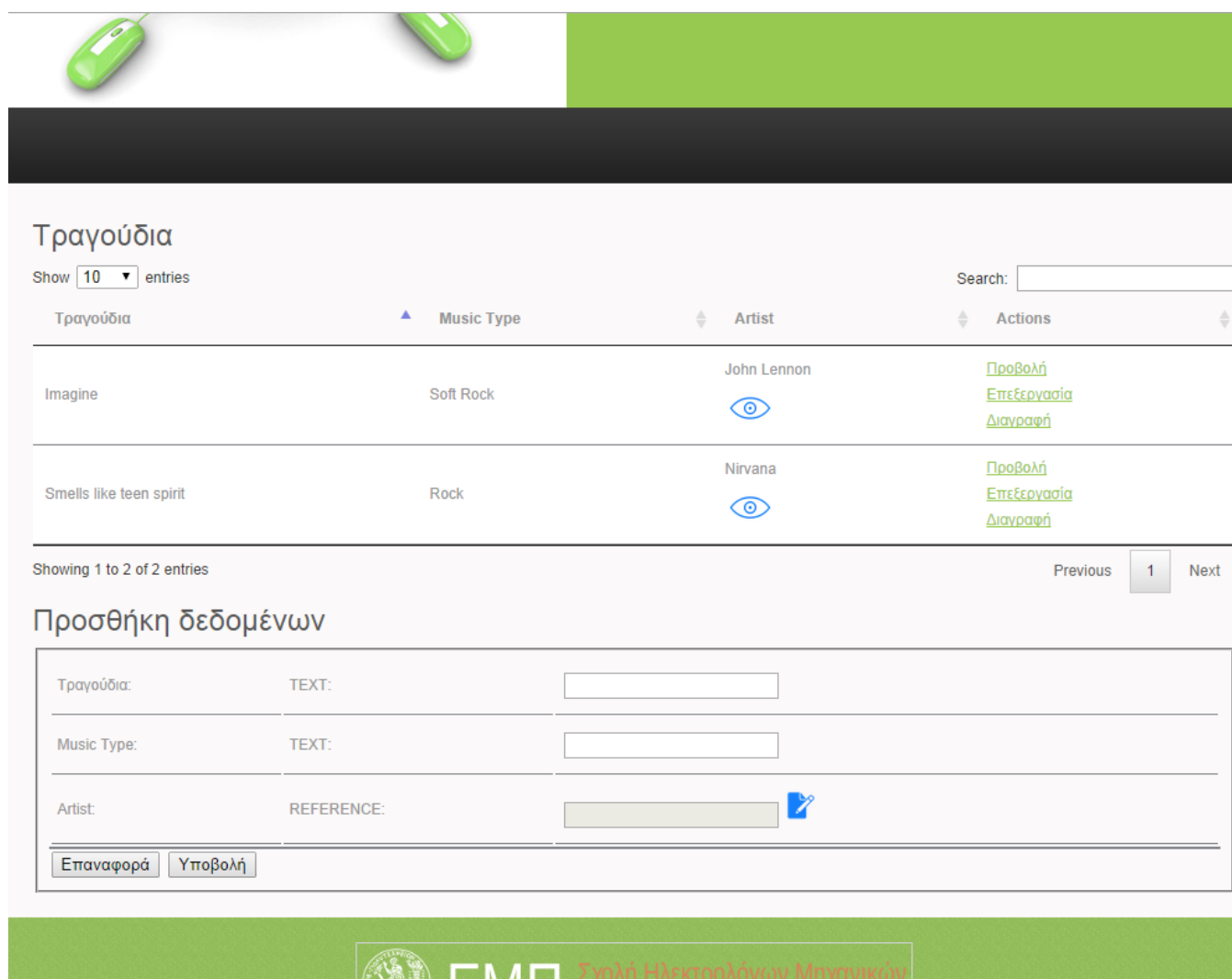
```

protected Integer newTableParentId = null;
protected String newTableName = null;
protected String newTableDisplayName = null;
protected String newTableIsMaster = "0";
protected String newTableRequestResult = "";
protected class CreateTableTask extends AsyncTask<String, Integer, Long> {
...
    protected Long doInBackground(String... progress) {
        try {
            Map<String,String> parametersMap = new HashMap<String, String>();
            parametersMap.put("external", "true");
            if ( newTableParentId != null ) {
                parametersMap.put("parentId", newTableParentId.toString());
            }
            parametersMap.put("isMaster", newTableIsMaster.toString());
            parametersMap.put("tableName", newTableName);
            parametersMap.put("tableDisplayName", newTableDisplayName);
            newTableRequestResult =
NetworkUtil.createPost(ThesisConstants.CREATE_TABLE_ACTION_URL, parametersMap);
        } catch (Exception e) {}
        return 0L;
    }
...
    protected void onPostExecute(Long result) {
        progDailog.dismiss();
        checkNewTableResult();
    }
}

```

#### 4.2.13 Προβολή δεδομένων πίνακα

Μία από τις πιο σημαντικές λειτουργίες των εφαρμογών μας είναι η εμφάνιση των δεδομένων που έχει εισάγει ο χρήστης στους πίνακες που έχει δημιουργήσει.




The screenshot displays a web application interface with a green header and a dark grey navigation bar. Below the navigation bar, the page title is "Τραγούδια". There is a "Show 10 entries" dropdown and a search box. The main content is a table with columns: "Τραγούδια", "Music Type", "Artist", and "Actions".

Τραγούδια	Music Type	Artist	Actions
Imagine	Soft Rock	John Lennon	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Smells like teen spirit	Rock	Nirvana	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

Below the table, it says "Showing 1 to 2 of 2 entries" and has "Previous", "1", and "Next" navigation buttons.

Below the table is a section titled "Προσθήκη δεδομένων" with a form:

Τραγούδια:	TEXT:	<input type="text"/>
Music Type:	TEXT:	<input type="text"/>
Artist:	REFERENCE:	<input type="text"/> 

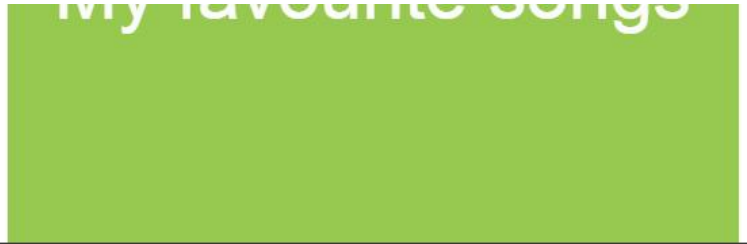
At the bottom of the form are buttons for "Επαναφορά" and "Υποβολή".

At the bottom of the page is a green footer with the logo of the "Σχολή Ηλεκτρονικών Μηχανικών" (School of Electronic Engineering) and the acronym "ΕΜΠ".

Όπως βλέπουμε στην ανωτέρω εικόνα στον πίνακα τραγούδια έχουν προστεθεί δύο εγγραφές από τραγούδια. Βλέπουμε τις τιμές των πεδίων από όλα τα πεδία καθώς και αυτά που προέρχονται από referenced πίνακα. Στο τέλος εμφανίζονται μια σειρά εντολών που μπορεί να εκτελέσει ο χρήστης όπως λεπτομερή προβολή, επεξεργασία ή διαγραφή μιας εγγραφής στον πίνακα.

Παρατηρούμε την εμφάνιση της κολόνας "Artist" η της οποίας τα δεδομένα προέρχονται από referenced πίνακα. Κατά την εμφάνιση των δεδομένων όταν τυπώνονται οι τιμές των referenced πινάκων, η εφαρμογή ζητάει από τον referenced πίνακα την τιμή της κεντρικής κολόνας ("value") που αντιστοιχεί στην εγγραφή στην οποία δείχνει το foreign key του παρόντος πίνακα στην κολόνα σύνδεσης.

Στην παρακάτω εικόνα βλέπουμε τις τιμές του referenced πίνακα "Artist" με τις εγγραφές των καλλιτεχνών οι οποίες μπορούν να χρησιμοποιηθούν για τη σύνδεση με τα τραγούδια.



## Artist

Show  entries Search:

Artist	Actions
John Lennon	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Nirvana	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

Showing 1 to 2 of 2 entries Previous  Next

### Προσθήκη δεδομένων

Artist:  TEXT:

Στο ακόλουθο κομμάτι κώδικα βλέπουμε τον τρόπο με τον οποίο εμφανίζονται τα δεδομένα του πίνακα.

```
$table = $ReferenceDAO->findTableById ( $tableId );
if ( $table != null ) {
    echo "<h2>" . $table['table_display_name'] . "</h2>";

    echo "<table id=\"tableData\">";
    echo " <thead>";
    if ( $selectItem == 'true' ) {
        echo "<th>Επιλογή</th>";
    }
    $tableColumns = $ReferenceDAO->findAllColumnsForTableId ( $tableId );
    while ( $column = mysqli_fetch_array($tableColumns) ) {
        echo "<th>" . $column['column_display_name'] . "</th>";
    }

    if ( $selectItem != 'true' ) {
        echo " <th>Actions</th>";
    }

    echo " </thead>";
    echo " <tbody>";

    $tableValues = $GeneralDAO->findAllValuesOfTable (
    $table['table_name'] );
```

```

while ( $row = mysqli_fetch_array($tableValues) ) {
    echo "<tr>";

    if ( $selectItem == 'true' ) {
        echo "<td><input type=\"radio\" name=\"selectedItemId\"
value=\"\" . $row['id'] . \"|\" . $row['value'] . \"\"></td>";
    }

    $tableColumns = $ReferenceDAO->findAllColumnsForTableId (
$tableId );
    while ( $column = mysqli_fetch_array($tableColumns) ) {

        if ( $column['column_type'] == 'REFERENCE' ) {
            echo "<td>";
            if ( $row[$column['column_name']] != null &&
$row[$column['column_name']] > 0 ) {
                $referencedTable = $ReferenceDAO->findTableById (
$column['reference_table_id'] );
                $columnValueRow = $GeneralDAO->getValueOfTableRow(
$referencedTable['table name'], $row[$column['column name']] );
                echo $columnValueRow['value'] . "<div><a
href=\"viewTableRow.php?tableId=\" . $column['reference_table_id'] .
\"&id=\" . $row[$column['column_name']] . \"\"><img width=\"40\"
height=\"40\" src=\"\" . $contextRoot .
\"template/images/eye_icon.png\" /></a></div>";
            }
            echo "</td>";
        } else if ( $column['column_type'] == 'LOCATION' &&
StringUtil::notEmptyOrNull($row[$column['column_name']]) &&
TableUtil::validateInput ( 'LOCATION', $row[$column['column_name']]
)) {
            echo "<td>" . $row[$column['column_name']] . "<div><a
href=\"http://maps.google.com/?ie=UTF8&hq=&ll=\" .
$row[$column['column_name']] . "&z=13\"><img width=\"40\"
height=\"40\" src=\"\" . $contextRoot .
\"template/images/google_maps.png\" /></a></div></td>";
        } else {
            echo "<td>" . $row[$column['column_name']] . "</td>";
        }
    }
    if ( $selectItem != 'true' ) {
        echo " <td><div><a href=\"viewTableRow.php?tableId=\" .
$tableId . "&id=\" . $row['id'] . \"\">Προβολή</a></div><div><a
href=\"editTableData.php?tableId=\" . $tableId . "&id=\" . $row['id'] .
\"\">Επεξεργασία</a></div><div><a
href=\"../action/deleteTableRow.php?tableId=\" . $tableId . "&id=\" .
$row['id'] . \"\">Διαγραφή</a></div></td>";
    }
    echo "</tr>";
}

echo " </tbody>";
echo "</table>";

```

Αρχικά ζητείται από τον πίνακα “columns\_reference” να έρθουν όλα τα πεδία που ανήκουν στον ζητούμενο πίνακα. Με αυτό τον τρόπο γνωρίζουμε ποιες κολόνες να δείξουμε στον πίνακα HTML που θα τυπωθεί στη σελίδα μας.

Τρέχοντας όλη τη λίστα με τα πεδία αυτά τυπώνεται το “head” του HTML πίνακα όπου για κάθε κολόνα δείχνεται ως κεφαλίδα το εμφανιζόμενο όνομα του



κάθε πεδίου. Στη συνέχεια ζητούνται όλα τα δεδομένα του πίνακα από τη βάση δεδομένων. Για κάθε γραμμή του πίνακα, τρέχοντας τη λίστα με τα διαθέσιμα πεδία πίνακα, αντιστοιχίζεται η κάθε τιμή με την κολόνα στην οποία ανήκει τυπώνοντας έτσι το “body” του πίνακα. Στην τελευταία κολόνα μπαίνουν τα links των εντολών που μπορεί να εκτελέσει ο χρήστης.

Παρατηρούμε ότι στην περίπτωση που μία κολόνα αντιστοιχεί σε reference πίνακα τότε ζητείται η τιμή της κεντρικής κολόνας για την εγγραφή που δείχνει η κολόνα σύνδεσης (αν υπάρχει, αλλιώς εμφανίζεται κενή), καθώς και το πλήκτρο



το οποίο κάνοντας click πάνω του οδηγούμαστε στη λεπτομερή προβολή της αντίστοιχης εγγραφής του referenced πίνακα.

Στην εφαρμογή του Android έχουμε την οθόνη όπου ο χρήστης βλέπει τις



εγγραφές του πίνακα. Λόγω του μικρού μεγέθους των οθονών των κινητών συσκευών εδώ αντί να εμφανίσουμε τις τιμές όλων των πεδίων, αντίθετα εμφανίζουμε την τιμή της κεντρικής μόνο κολόνας. Στη συνέχεια ο χρήστης επιλέγοντας κάποια εγγραφή οδηγείται στη λεπτομερή εμφάνιση των δεδομένων όπου φαίνονται όλες οι πληροφορίες.

Ακολουθεί το πιο σημαντικό κομμάτι του κώδικα του adapter που αναλαμβάνει την εμφάνιση των εγγραφών του πίνακα.

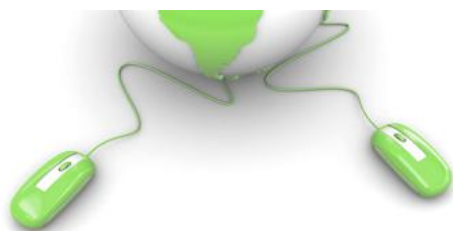
```
public View getView (int position , View convertView, ViewGroup parent ) {
    View row = convertView;
    RowHolder holder = null;
    if ( row == null ) {
        LayoutInflater inflater = (LayoutInflater)
        getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        row = inflater.inflate(R.layout.table_data_list_row, parent, false);
        holder = new RowHolder();
        holder.cell_rowId = (TextView) row.findViewById(R.id.tableRowId);
        holder.cell_rowValue = (TextView) row.findViewById(R.id.tableRowValue);
        row.setTag(holder);
    } else {
        holder = (RowHolder) row.getTag();
    }
    Map<String,String> tableRow = dataList.get(position); //data [position];
    holder.cell_rowId.setText( tableRow.get("id" ) );
    holder.cell_rowValue.setText( tableRow.get("value" ) );
    return row; }
```

#### 4.2.14 Αναζήτηση δεδομένων πίνακα

Στην οθόνη εμφάνισης δεδομένων ενός πίνακα δίνεται η δυνατότητα αναζήτησης μέσα στον εμφανιζόμενο πίνακα. Όταν ο αριθμός των εγγραφών γίνει πολύ μεγάλος τότε εμφανίζεται η ανάγκη ευκολότερης εύρεσης και εμφάνισης των δεδομένων. Εδώ είναι που χρησιμοποιείται η βιβλιοθήκη DataTables που έχει αναφερθεί προηγουμένως. Ενεργοποιώντας τη στον HTML πίνακα που μας ενδιαφέρει με την ακόλουθη εντολή

```
<script>
    $(document).ready(function() {
        $('#tableData').DataTable();
    });
</script>
```

ο πίνακας αυτομάτως εμφανίζει τα 10 πρώτα αποτελέσματα και στο πάνω δεξιά σημείο του εμφανίζεται ένα πεδίο αναζήτησης. Γράφοντας σε αυτό το πεδίο, η κάθε γραμμή το πίνακα φιλτράρεται σύμφωνα με την είσοδο του χρήστη και αυτόματα εμφανίζονται σε αυτόν μόνο οι εγγραφές οι οποίες περιλαμβάνουν το κείμενο της εισόδου. Οι υπόλοιπες γραμμές κρύβονται.



### Artist

Show  entries Search:

Artist	Actions
John Lennon	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Nirvana	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

Showing 1 to 2 of 2 entries Previous  Next

### Προσθήκη δεδομένων

Artist:  TEXT:

Άλλα δύο σημαντικά χαρακτηριστικά που προσφέρει η βιβλιοθήκη αυτή είναι η “σελιδοποίηση” και η “ταξινόμηση” των δεδομένων. Αν οι εγγραφές ξεπεράσουν τις 10 τότε τα δεδομένα που είναι μετά τη 10<sup>η</sup> θέση κρύβονται και στο κάτω μέρος του πίνακα εμφανίζονται αριθμοί σελίδων έτσι πατώντας στον καθένα από αυτούς οδηγούμαστε στη Χ δεκάδα εγγραφών. Κάνοντας click στα βελάκια στην κεφαλίδα του πίνακα γίνεται ταξινόμηση των εγγραφών σύμφωνα με τα δεδομένα που βρίσκονται στην αντίστοιχη κολόνα την οποία επιλέξαμε για ταξινόμηση.

#### 4.2.15 Λεπτομερής εμφάνιση εγγραφής πίνακα


Στην οθόνη εμφάνισης δεδομένων πίνακα, δίπλα από κάθε εγγραφή υπάρχει το πλήκτρο “Προβολή”. Πατώντας το ο χρήστης οδηγείται στην οθόνη λεπτομερούς προβολής μιας εγγραφής πίνακα.



## My favourite songs

### Τραγούδια

#### Προβολή δεδομένων


Τραγούδια:	<input type="text" value="Imagine"/>
Music Type:	<input type="text" value="Soft Rock"/>
Artist:	<input type="text" value="John Lennon"/> 



EMΠ

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών

Βλέπουμε εδώ την εμφάνιση όλων των πεδίων του πίνακα καθώς και την τιμή της κεντρικής κολόνας της εγγραφής που αντιστοιχεί στον referenced πίνακα “Artist”. Δίπλα στο πεδίο που αντιστοιχεί σε referenced πίνακα φαίνεται και το

πλήκτρο  το οποίο κάνοντας click πάνω του οδηγούμαστε στη λεπτομερή προβολή της αντίστοιχης εγγραφής του referenced πίνακα.

Στο ακόλουθο κομμάτι κώδικα βλέπουμε τον τρόπο με τον οποίο ανακτώνται και εμφανίζονται οι πληροφορίες της λεπτομερούς προβολής μιας εγγραφής πίνακα.

```

<table>
  <tbody>

    <?php
      $tableColumns = $ReferenceDAO->findAllColumnsForTableId (
      $tableId );
      while ( $column = mysqli_fetch_array($tableColumns) ) {
        echo "<tr>";

          echo "<td><span>" . $column['column_display_name'] .
          " :</span></td><td>";

          if ( $column['column_type'] == 'REFERENCE' ) {

            if ( $object[$column['column_name']] != null &&
            $object[$column['column_name']] > 0 ) {
              $referencedTable = $ReferenceDAO->findTableById (
              $column['reference_table_id'] );
              $columnValueRow = $GeneralDAO->
              >getValueOfTableRow( $referencedTable['table_name'],
              $object[$column['column_name']] );
              echo "<input type=\"text\" name=\"" .
              $column['column_name'] . "\" value=\"" . $columnValueRow['value'] .
              "\" disabled=\"disabled\" /><a href=\"viewTableRow.php?tableId=" .
              $column['reference_table_id'] . "&id=" .
              $object[$column['column_name']] . "\"><img width=\"40\" height=\"40\"
              src=\"" . $contextRoot . "template/images/eye_icon.png\" /></a>";
            }

            } else if ( $column['column_type'] == 'LOCATION' &&
            StringUtil::notEmptyOrNull($object[$column['column_name']]) &&
            TableUtil::validateInput ( 'LOCATION',
            $object[$column['column_name']] )) {
              echo "<input type=\"text\" name=\"" .
              $column['column_name'] . "\" value=\"" .
              $object[$column['column_name']] . "\" disabled=\"disabled\" /> <a
              href=\"http://maps.google.com/?ie=UTF8&hq=&ll=" .
              $object[$column['column_name']] . "&z=13\"><img width=\"40\"
              height=\"40\" src=\"" . $contextRoot .
              "template/images/google_maps.png\" /></a>";
            } else {
              echo "<input type=\"text\" name=\"" .
              $column['column_name'] . "\" value=\"" .
              $object[$column['column_name']] . "\" disabled=\"disabled\" />";
            }
          echo "</td></tr>";
        }
      }
    <?>

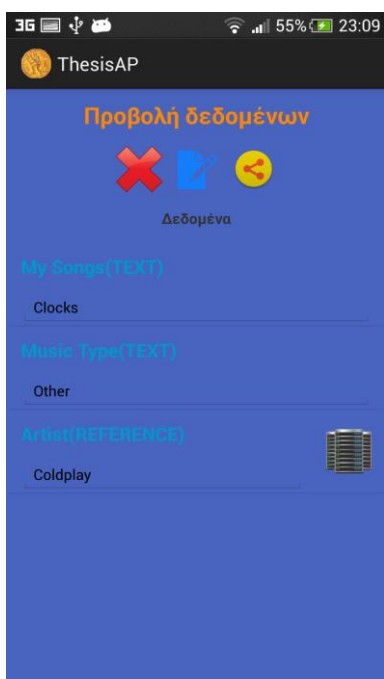
  </tbody>
</table>


```

Αρχικά ζητούνται τα διαθέσιμα “πεδία” πίνακα από τα οποία εμφανίζεται το εμφανιζόμενο όνομα ως τίτλος του κάθε πεδίου. Στη συνέχεια έχοντας φέρει τα

δεδομένα της εγγραφής από τον πίνακα, δίπλα σε κάθε πεδίο τυπώνεται η τιμή που του αντιστοιχεί. Ιδιαίτερη προσοχή χρειάζεται στην περίπτωση πεδίων που αντιστοιχούν σε referenced πίνακα καθώς εκεί τυπώνεται η τιμή της κεντρικής κολόνας της εγγραφής που αντιστοιχεί στον συνδεδεμένο πίνακα. Επίσης στην περίπτωση που πρόκειται για πεδίο τύπου "LOCATION" και έχει τιμή, τότε εμφανίζεται πλήκτρο που οδηγεί στους χάρτες της Google (αυτή η περίπτωση θα αναλυθεί περαιτέρω στη συνέχεια).

Έχοντας επιλέξει μια εγγραφή από την οθόνη προβολής δεδομένων ενός πίνακα στην εφαρμογή του Android, ο χρήστης οδηγείται στην οθόνη λεπτομερούς προβολής δεδομένων. Εκεί ομοίως με τη web εφαρμογή, έρχεται από το server η λίστα με τα δυναμικά πεδία του πίνακα και για κάθε πεδίο τυπώνεται η τιμή που αντιστοιχεί από την εγγραφή. Στην περίπτωση που το πεδίο αντιστοιχεί σε referenced πίνακα εμφανίζεται η τιμή της κεντρικής κολόνας της του referenced πίνακα για την εγγραφή που αντιστοιχεί.



Σε αυτή την οθόνη παρατηρούμε και το πλήκτρο "share"  η λειτουργία του οποίου θα αναλυθεί στη συνέχεια.

Ακολουθεί το κομμάτι του adapter που εμφανίζει τα πεδία με τις τιμές τους.

```
Context context;
int layoutResourceId;
Map<String,String> data[] = null;
List<Column> columnList = null;
Map<String,String> tableRow = null;
Map<String, Map<String,String>> referencedTablesRows = null;
Boolean editRow = false;

public View getView (int position , View convertView, ViewGroup parent ) {

    View row = convertView;
    RowHolder holder = null;

    if ( row == null ) {
        //LayoutInflater inflater = ((Activity) context).getLayoutInflater();
        LayoutInflater inflater = (LayoutInflater)
getContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        row = inflater.inflate(R.layout.title_value_row_layout, parent, false);

        holder = new RowHolder();
        holder.cell_title = (TextView) row.findViewById(R.id.rowTitle);
        holder.cell_value = (EditText) row.findViewById(R.id.rowValue);
        holder.cell_viewReferencedTableRowButton = (Button)
row.findViewById(R.id.viewReferencedTableRowButton);
        holder.cell_goToMapsButton = (Button) row.findViewById(R.id.goToMapsButton);
```

```

        row.setTag(holder);
    } else {
        holder = (RowHolder) row.getTag();
    }

    Column column = columnList.get(position);

    holder.cell_title.setText( column.getColumnDisplayName() + "(" + column.getColumnType() + ")" );
    holder.cell_viewReferencedTableRowButton.setVisibility(View.GONE);
    holder.cell_goToMapsButton.setVisibility(View.GONE);

    String rowValue = "";
    if ( column.getColumnType().equals("REFERENCE") ) {
        Map<String,String> referencedRow = referencedTablesRows.get(column.getColumnName());
        if ( referencedRow != null ) {
            rowValue = referencedRow.get("value");
        }
        holder.cell_viewReferencedTableRowButton.setVisibility(View.VISIBLE);
    } else {
        rowValue = tableRow.get(column.getColumnName());
    }


    if ( column.getColumnType().equals("LOCATION") ) {
        holder.cell_goToMapsButton.setVisibility(View.VISIBLE);
    }
    holder.cell_value.setText( rowValue );
    holder.cell_value.setId(position);
    viewsColumns.put(holder.cell_value.getId(), column.getColumnName());

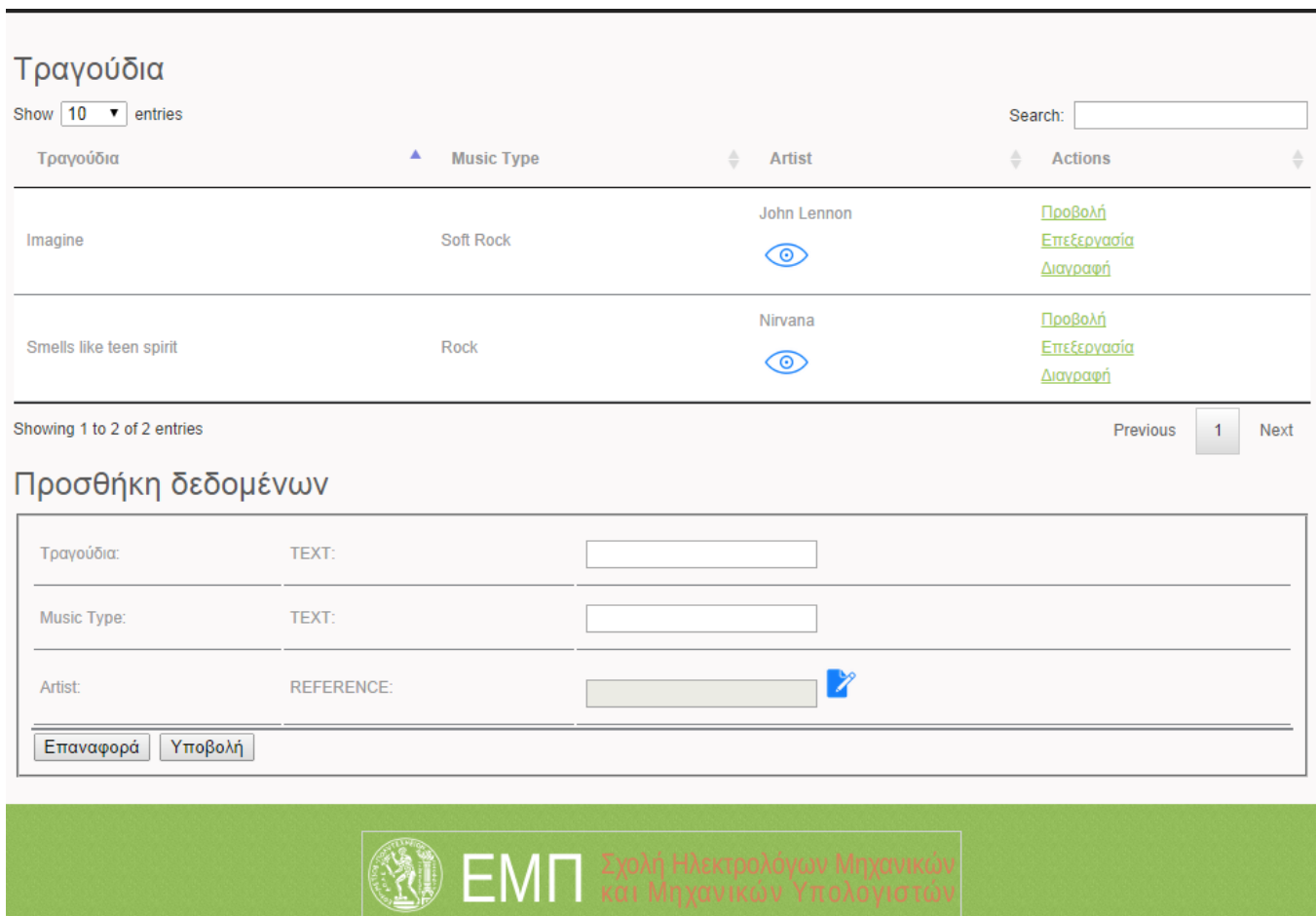
    if ( editRow && !column.getColumnType().equals("REFERENCE") ) {
        holder.cell_value.setEnabled(true);
        holder.cell_value.setOnFocusChangeListener(new OnFocusChangeListener() {
            public void onFocusChange(View v, boolean hasFocus) {
                if (!hasFocus) {
                    final int viewId = v.getId();
                    final EditText editText = (EditText) v;
                    String columnName = viewsColumns.get(viewId);
                    String editedText = editText.getEditableText().toString();
                    tableRow.put(columnName, editedText);
                }
            }
        });
    } else {
        holder.cell_value.setEnabled(false);
    }

    return row;
}

```

#### 4.2.16 Προσθήκη δεδομένων

Στην οθόνη εμφάνισης δεδομένων, κάτω από τον πίνακα που παρουσιάζει τις εγγραφές του πίνακα, εμφανίζεται μια φόρμα προσθήκης νέων δεδομένων. Σε αυτή τη φόρμα εμφανίζονται inputs για όλα τα διαθέσιμα πεδία του πίνακα. Προσοχή στα πεδία τα οποία αντιστοιχούν σε referenced πίνακα, καθώς αυτά είναι κλειδωμένα και ο χρήστης δε μπορεί να εισάγει τιμή με το χέρι, αλλά να πατήσει το πλήκτρο επιλογής (  ) δεδομένων όπου μέσω ενός popup καλείται να επιλέξει μια τιμή από τις υπάρχουσες του συνδεόμενου πίνακα, είτε να προσθέσει μια καινούρια εγγραφή σε αυτόν και να την επιλέξει (η διαδικασία αυτή θα αναλυθεί στη συνέχεια).



The screenshot displays a web interface for managing a music database. At the top, there's a section titled "Τραγούδια" (Songs) with a search bar and a "Show 10 entries" dropdown. Below this is a table with columns: "Τραγούδια", "Music Type", "Artist", and "Actions". Two rows are visible: "Imagine" by John Lennon (Soft Rock) and "Smells like teen spirit" by Nirvana (Rock). Each row has an eye icon and three action links: "Προβολή", "Επεξεργασία", and "Διαγραφή".

Below the table, there's a "Προσθήκη δεδομένων" (Add data) form. It has three rows of input fields: "Τραγούδια:" (TEXT), "Music Type:" (TEXT), and "Artist:" (REFERENCE). The "Artist:" field is locked, indicated by a lock icon. At the bottom of the form are two buttons: "Επαναφορά" (Reset) and "Υποβολή" (Submit).

At the bottom of the page, there's a green banner with the logo of the Department of Electrical and Computer Engineering (EMΠ) at the University of Patras.

Ο χρήστης βάζει τις τιμές που θέλει και πατάει το πλήκτρο submit. Τα δεδομένα, ανάλογα με τον τύπο του κάθε πεδίου, περνούν από validation με τη χρήση regular expressions όπως φαίνεται στο παρακάτω κομμάτι κώδικα:

```
public static function validateInput ( $columnType, $input ) {  
  
    $result = 'true';  
  
    $regex = "/.*"/;  
    $allowEmpty = true;
```

```

if ( $columnType == 'INTEGER' ) {
    $regex = "/^[1-9][0-9]*$/";
} else if ( $columnType == 'DOUBLE' ) {
    $regex = "/^-?(?:\d+|\d*\.\d+)$/";
} else if ( $columnType == 'LOCATION' ) {
    $regex = "/-?\d*\.\d*,-?\d*\.\d*/"; //24.12345,19.7654
} else if ( $columnType == 'DATE' ) {
    $regex = "/^[0-9]{4}-(0[1-9]|1[0-2])-(0[1-9]|1[1-2][0-9]|3[0-
1])$/"; //2012-09-12
}

//validate
if ( $allowEmpty == false && empty($input) ) {
    $result = 'false';
} else if ( $allowEmpty == true && empty($input) ) {
    $result = 'true';
} else {
    $input = TableUtil::test_input($input);
    if ( !preg_match($regex, $input) ) {
        $result = 'false';
    }
}

return $result;
}

```

Αν όλες οι τιμές περάσουν επιτυχώς το validation τότε η εγγραφή προστίθεται στον πίνακα.

Ενδιαφέρον έχει ο τρόπος με τον οποίο δημιουργείται η φόρμα υποβολής τιμών για εισαγωγή δεδομένων στον πίνακα, όπου παίρνοντας τα διαθέσιμα πεδία του πίνακα από τον πίνακα "columns\_reference", δημιουργείται από ένα input για κάθε πεδίο όπως φαίνεται στο ακόλουθο κομμάτι κώδικα.

```

<form id="dataForm" class="form" method="get" action="<?php echo
$contextRoot; ?>action/addEditTableRow.php">
    <input type="hidden" name="tableId" value="<?php echo $tableId;
?>" />
    <input type="hidden" name="selectItem" value="<?php echo
$selectItem; ?>" />

    <fieldset>
        <table>
            <tbody>

                <?php
                $tableColumns = $ReferenceDAO->findAllColumnsForTableId (
                $tableId );
                while ( $column = mysqli_fetch_array($tableColumns) ) {
                    echo "<tr>";

                    echo "<td><span>" . $column['column_display_name'] .
                    ":</span></td>";
                    echo "<td><span>" . $column['column_type'] .
                    ":</span></td>";
                    if ( $column['column_type'] == 'REFERENCE' ) {
                        echo "<td>";
                        echo "<input id=\"tableText\".
                        $column['reference_table_id'] .\" type=\"text\"
                        disabled=\"disabled\" />";

```



```

        echo " <a href=\"\#dataForm\"
onClick=\"selectItemFromReferencedTable(\".
$column['reference_table_id'] . "\")\" ><img width=\"25\" height=\"25\"
src=\"\" . $contextRoot . "template/images/edit icon.png\" /></a>";
        echo "<input type=\"hidden\" id=\"tableValue\".
$column['reference_table_id'] . \"\" name=\"\" . $column['column_name']
. \"\" value=\"\" />";
        echo "</td>";
    } else {
        echo "<td><input type=\"text\" name=\"\" .
$column['column_name'] . \"\" value=\"\" /></td>";
    }

    echo "</tr>";
}

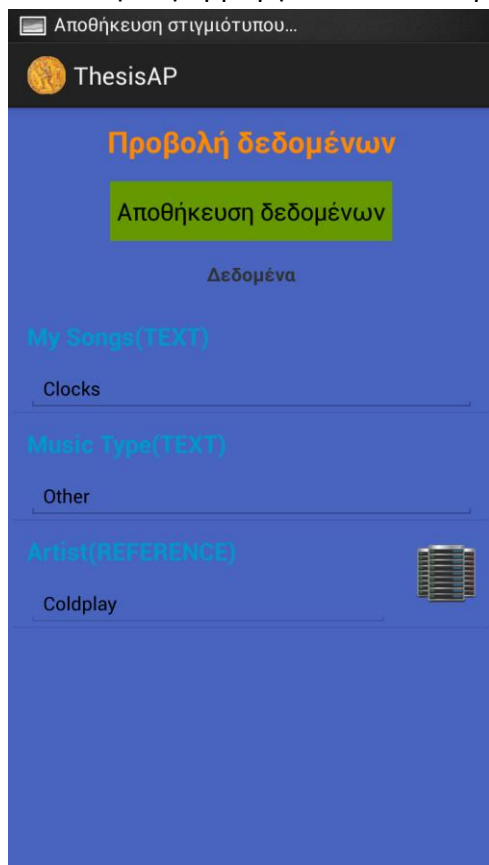
?>
</tbody>
</table>

<span class="btn">
    <input type="reset" data-type="submit" ></a>
    <input type="submit" data-type="submit" ></a>
</span>

<span class="clear"></span>
</fieldset>
</form>

```

Στην εφαρμογή του Android μέσα σε μια ListView εμφανίζονται τα πεδία του



πίνακα και για το καθένα υπάρχει ένα widget “EditText” μέσα στο οποίο ο χρήστης γράφει τις τιμές της εγγραφής που θέλει να προσθέσει.

Στη δίπλα εικόνα βλέπουμε τα πεδία με τις τιμές που έχουμε επιλέξει για να αποθηκεύσουμε. Πατώντας το πλήκτρο “Αποθήκευση” τα δεδομένα αποστέλλονται στο server για έλεγχο και αποθήκευση.


Χρησιμοποιείται ο ίδιος Adapter με την οθόνη λεπτομερούς εμφάνισης δεδομένων, με τη διαφορά ότι τώρα τα πεδία δεν έχουν τιμές.

Παρατηρούμε στον κώδικα που δόθηκε προηγουμένως τον “OnFocusChangeListener” όπου όταν ο χρήστης τελειώσει με την πληκτρολόγηση των δεδομένων και αλλάξει πεδίο ή πατήσει το πλήκτρο της αποθήκευσης, τότε αποθηκεύονται στη List τα δεδομένα που πρόσθεσε ο χρήστης.

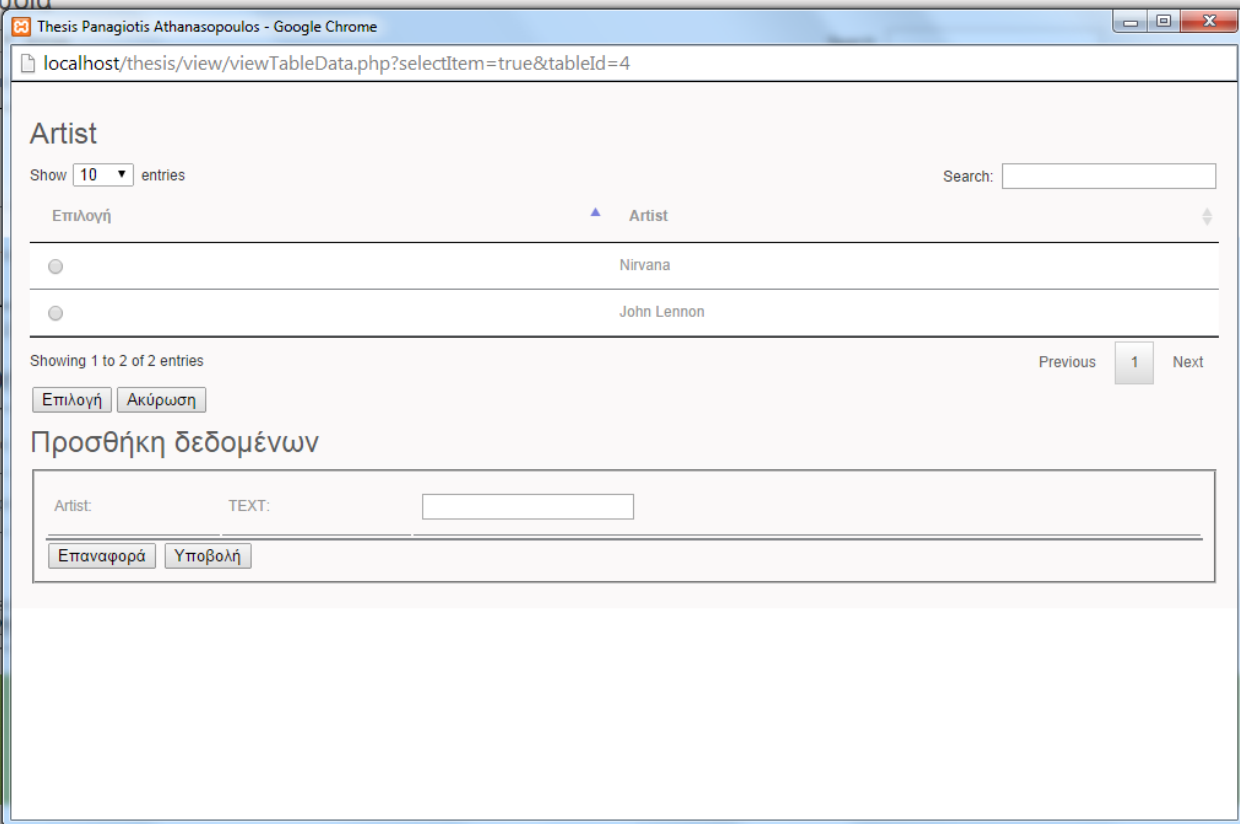
Η περίπτωση δεδομένων από referenced πίνακα θα αναλυθεί στη συνέχεια.

#### 4.2.17 Επιλογή δεδομένων από referenced πίνακα

Κατά την προσθήκη και επεξεργασία μιας εγγραφής ενός πίνακα, μια από τις σημαντικότερες και συνάμα δυσκολότερες προγραμματιστικά λειτουργίες είναι η επιλογή δεδομένων από referenced πίνακα για σύνδεση της εγγραφής μας με κάποια στον συνδεδεμένο πίνακα. Όταν πρόκειται για επιλογή τιμής τότε το input που στη reference κολόνα είναι απενεργοποιημένο έτσι ώστε ο χρήστης να μη μπορεί να βάλει τιμή με το χέρι και δίπλα σε αυτό εμφανίζεται το πλήκτρο επιλογής

(  ) δεδομένων.

Αν σκεφτούμε τη λειτουργία που χρειάζεται για να γίνει αυτή η πράξη παρατηρούμε ότι στην ουσία πρόκειται για τη σελίδα εμφάνισης δεδομένων πίνακα με τη δυνατότητα επιλογής μιας εγγραφής. Έτσι και σχεδιάστηκε. Όταν ο χρήστης πατήσει το παραπάνω πλήκτρο ανοίγει ένα popup το οποίο ανοίγει τη σελίδα εμφάνισης δεδομένων πίνακα με παράμετρο την “selectItem=true”. Η εν λόγω σελίδα έχει σχεδιαστεί έτσι ώστε όταν διαβάσει την παράμετρο αυτή τότε να κρύψει την κεφαλίδα και το υποσέλιδο της εφαρμογής και να εμφανίσει μέσα στο popup μόνο τον HTML πίνακα με τις εγγραφές και τη φόρμα προσθήκης νέας εγγραφής. Μπροστά από κάθε εγγραφή εμφανίζεται ένα “radio button” και τέλος το πλήκτρο επιλογής.



The screenshot shows a web browser window with a popup overlay. The popup is titled "Artist" and contains a table with two entries: Nirvana and John Lennon. Each entry has a radio button to its left. Below the table, there are buttons for "Επιλογή" (Select) and "Ακύρωση" (Cancel). Below the buttons, there is a section titled "Προσθήκη δεδομένων" (Add data) with a form containing a text input field for "Artist" and buttons for "Επαναφορά" (Reset) and "Υποβολή" (Submit).

Αφού ο χρήστης επιλέξει μια εγγραφή (ή προσθέσει μια καινούρια και την επιλέξει στη συνέχεια), πατώντας το πλήκτρο “Επιλογή” το popup κλείνει και στέλνει στην αρχική μας σελίδα την τιμή της κεντρικής κολόνας της εγγραφής που

επιλέχθηκε και το ID αυτής. Με αυτό τον τρόπο εμφανίζεται στο χρήστη η τιμή που επέλεξε και σε ένα κρυφό πεδίο ορίζεται το ID για να χρησιμοποιηθεί για την αποθήκευση της σύνδεσης στην reference κολόνα με τον αντίστοιχο πίνακα.

Στο ακόλουθο κομμάτι κώδικα βλέπουμε τις javascript μεθόδους που αναλαμβάνουν το άνοιγμα του popup και στη συνέχεια τον ορισμό των επιλεγμένων τιμών στα αντίστοιχα πεδία.

```
<script>
    $(document).ready(function() {
        $('#tableData').DataTable();
    });

    function returnSelectedItem() {

        var selectedItem = $('input[name=selectedItemId]:checked',
        '#tableData').val();
        var tableId = <?php echo $tableId; ?>;

        if( selectedItem != null && selectedItem != undefined ) {
            window.opener.setItemFromReferencedTable(tableId,
            selectedItem);
            window.close();
        }else{
            alert("Επιλέξτε πεδίο.");
        }
    }

    function selectItemFromReferencedTable(tableId) {
        window.open("viewTableData.php?selectItem=true&tableId=" +
        tableId,'Select','toolbar=no,resizable=yes,scrollbars=yes,width=1000,
        height=600,top=50,left=50');
    }

    function setItemFromReferencedTable(tableId, selectedItem) {

        var frm =
        document.getElementById("templateLibraryDocumentForm");

        //for our case we have one result
        var itemId = selectedItem.split("|")[0];
        var itemName = selectedItem.split("|")[1];

        if( itemId != null && itemId != undefined ) {
            $("#tableText"+tableId).val(itemName);
            $("#tableValue"+tableId).val(itemId);
        }
    }
</script>
```

Η function “selectItemFromReferencedTable” αναλαμβάνει να ανοίγει το popup με παράμετρο το Id του πίνακα (που αντιστοιχεί στην εγγραφή του πίνακα “tables\_reference”).

Αντίστοιχα η function “setItemFromReferencedTable” αναλαμβάνει την ανάθεση των επιλεγμένων τιμών στα αντίστοιχα πεδία.


Τέλος η function “returnSelectedItem” η οποία εκτελείται όταν ο χρήστης πατήσει το πλήκτρο “Επιλογή”, αναλαμβάνει να πάρει τις τιμές από την επιλεγμένη

εγγραφή, να τις μετατρέψει στη μορφή που μπορούν να χρησιμοποιηθούν στη συνέχεια και εκτελεί τη function “setItemFromReferencedTable” στο παράθυρο το οποίο άνοιξε το popup.

Το radio button επιλογής εγγραφής εμφανίζεται με το ακόλουθο κομμάτι κώδικα:

```
if ( $selectItem == 'true' ) {
    echo "<td><input type=\"radio\" name=\"selectedItemId\" value=\"\"
    . $row['id'] .\"|\" . $row['value'] . \"\></td>\";
}
```

Στην εφαρμογή του Android ο χρήστης όταν πρόκειται να προσθέσει ή να επεξεργαστεί δεδομένα πεδίου που έχουν σχέση με referenced πίνακα τότε

πατώντας το πλήκτρο  ανοίγει την οθόνη προβολής δεδομένων του referenced πίνακα. Όπως και με τη web εφαρμογή και εδώ η οθόνη αυτή έχει τα χαρακτηριστικά που απαιτούνται για τη λειτουργία αυτή, οπότε χρησιμοποιείται η ίδια για ευκολία σχεδιασμού και συντήρησης. Η διαφορά είναι αυτή τη φορά καλείται με χρήση της μεθόδου: **startActivityResult** ώστε η activity του Android να αναμένει ένα αποτέλεσμα από την οθόνη που ανοίγει. Στην περίπτωσή μας αυτό το αποτέλεσμα είναι το id και η τιμή της κεντρικής κολονας του πίνακα για την εγγραφή που επέλεξε ο χρήστης. Ακολουθεί το κομμάτι του κώδικα που διαχειρίζεται το αποτέλεσμα καθώς και αυτό το οποίο επιστρέφει τα δεδομένα της επιλεγμένης εγγραφής αντίστοιχα.

```
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == ThesisConstants.REQUEST_TABLE_ROW) {
        if (resultCode == RESULT_OK) {
            HashMap<String,String> selectedRow = (HashMap<String, String>)
data.getSerializableExtra(ThesisConstants.TABLE_ROW_INTENT);
            row.put(requestedColumn, selectedRow.get("id"));
            referencedTablesRows.put(requestedColumn, selectedRow);
            tableRowAdapter.setReferencedTablesRows(referencedTablesRows);
            tableRowAdapter.notifyDataSetChanged();
            tableRowAdapter.notifyDataSetInvalidated();
        }
    }
}
```

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    if ( !returnRowResult ) {
        goToViewTableRow(position);
    } else {
        Intent intent = new Intent();
        LinkedHashMap<String,String> row = (LinkedHashMap<String, String>) tableData.get(position);

        intent.putExtra(ThesisConstants.TABLE_ROW_INTENT, row);
        setResult(RESULT_OK, intent);
        finish();
    }
}
```

#### 4.2.18 Επεξεργασία εγγραφής πίνακα

Στην οθόνη εμφάνισης δεδομένων πίνακα, στην κολόνα με τις ενέργειες που μπορεί να εκτελέσει ο χρήστης βρίσκεται το πλήκτρο “Επεξεργασία”. Πατώντας αυτό ο χρήστης οδηγείται στη σελίδα στην οποία μπορεί να διορθώσει τις τιμές που έχουν τεθεί σε μια εγγραφή.

Όπως και με τη φόρμα προσθήκης εγγραφής σε έναν πίνακα, έτσι και εδώ εμφανίζεται μια φόρμα με όλα τα πεδία του πίνακα και τα αντίστοιχα inputs (έχοντας λάβει υπόψιν τα πεδία που αφορούν referenced πίνακες κτλ όπως προηγουμένως). Η διαφορά που έχουμε εδώ είναι ότι αντί στη φόρμα να εμφανίζονται κενά τα πεδία τιμών, αντίθετα προετοιμάζονται με τις τιμές που ισχύουν αυτή τη στιγμή για την αντίστοιχη εγγραφή.



Η οθόνη επεξεργασίας δεδομένων φαίνεται στην ακόλουθη εικόνα:



## My favourite songs

### Τραγούδια

#### Επεξεργασία δεδομένων

Τραγούδια:	<input type="text" value="Imagine"/>
Music Type:	<input type="text" value="Soft Rock"/>
Artist:	<input type="text" value="John Lennon"/>  
<input type="button" value="Επανάφορα"/> <input type="button" value="Υποβολή"/>	



EMΠ

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών

Ενδιαφέρον αποτελεί το κομμάτι του κώδικα όπου παίρνουμε τα δεδομένα της εγγραφής από τη βάση δεδομένων καθώς και τα διαθέσιμα πεδία του πίνακα και δημιουργούμε τη φόρμα γεμίζοντας όλα τα inputs με τις τιμές της εγγραφής.

```
$object = $GeneralDAO->findTableRow ( $table['table_name'], $id );  
?>
```

```
<div>  
  <h2>Επεξεργασία δεδομένων</h2>
```

```

<form id="dataForm" class="form" method="get" action="<?php echo
$contextRoot; ?>action/addEditTableRow.php">
    <input type="hidden" name="tableId" value="<?php echo
$tableId; ?>" />
    <input type="hidden" name="id" value="<?php echo $id; ?>" />

    <fieldset>
        <table>
            <tbody>

                <?php
                    $tableColumns = $ReferenceDAO-
>findAllColumnsForTableId ( $tableId );
                    while ( $column = mysqli_fetch_array($tableColumns) )
                {

                    echo "<tr>";

                    echo "<td><span>" .
                    $column['column_display_name'] . " :</span></td><td>";

                                if ( $column['column_type'] == 'REFERENCE' )
                {
                    if ( $object[$column['column_name']] !=
null && $object[$column['column_name']] > 0 ) {
                        $referencedTable = $ReferenceDAO-
>findTableById ( $column['reference_table_id'] );
                        $columnValueRow = $GeneralDAO-
>getValueOfTableRow( $referencedTable['table_name'],
$object[$column['column_name']] );
                        echo "<input id=\"tableText\".
                    $column['reference_table_id'] . \"\" type=\"text\" value=\"\" .
                    $columnValueRow['value'] . \"\" disabled=\"disabled\" /><a
                    href=\"viewTableRow.php?tableId=" . $column['reference_table_id'] .
                    "&id=" . $object[$column['column_name']] . "\"><img width=\"25\"
                    height=\"25\" src=\"\" . $contextRoot .
                    "template/images/eye icon.png\" /></a> ";
                        } else {
                            echo "<input id=\"tableText\".
                    $column['reference_table_id'] . \"\" type=\"text\"
                    disabled=\"disabled\" />";
                        }
                            echo " <a href=\"#\"
                    onClick=\"selectItemFromReferencedTable(\".
                    $column['reference_table_id'] . ")\" ><img width=\"25\" height=\"25\"
                    src=\"\" . $contextRoot . "template/images/edit icon.png\" /></a>";
                            echo "<input type=\"hidden\"
                    id=\"tableValue\". $column['reference_table_id'] . \"\" name=\"\" .
                    $column['column_name'] . \"\" value=\"\" .
                    $object[$column['column_name']] . \"\" />";
                            } else {
                                echo "<input type=\"text\" name=\"\" .
                    $column['column_name'] . \"\" value=\"\" .
                    $object[$column['column_name']] . \"\" />";
                            }
                        }
                    echo "</td></tr>";
                }
            }
        ?>

```

```

        </tbody>
    </table>

    <span class="btn">
        <input type="reset" data-type="submit" ></a>
        <input type="submit" data-type="submit" ></a>
    </span>

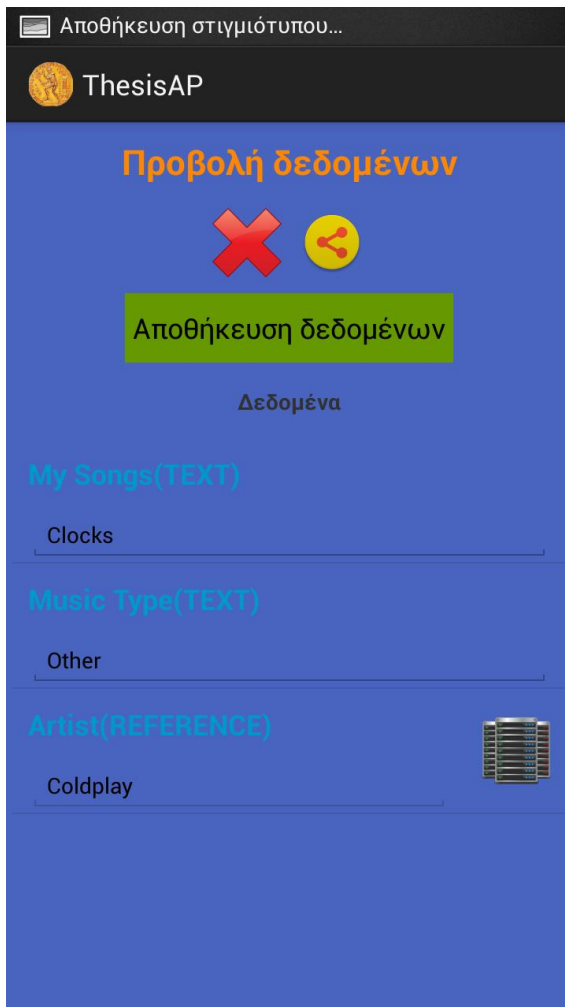
    <span class="clear"></span>
</fieldset>
</form>

</div>

```

Τέλος με το πάτημα του πλήκτρου submit από το χρήστη, τα νέα δεδομένα υπόκεινται στο ίδιο validation που γίνεται και κατά την προσθήκη δεδομένων και στην περίπτωση που περάσουν επιτυχώς αποθηκεύονται στη βάση δεδομένων κάνοντας ‘UPDATE’ στην αντίστοιχη γραμμή του πίνακα.

Στην εφαρμογή του Android για την επεξεργασία μιας εγγραφής



χρησιμοποιείται η ίδια οθόνη με την οθόνη λεπτομερούς εμφάνισης δεδομένων. Τα widgets ‘EditText’ προετοιμάζονται με τις υπάρχουσες τιμές, με τη διαφορά ότι τώρα είναι ‘ενεργοποιημένα’ όπως και στην προσθήκη δεδομένων, οπότε ο χρήστης μπορεί να τα επεξεργαστεί.

Ο κώδικας που αναλαμβάνει την αποστολή των δεδομένων στο server είναι ο ίδιος και με την περίπτωση της εισαγωγής νέας εγγραφής μιας και οι δυο περιπτώσεις έχουν πολλά κοινά χαρακτηριστικά.

Για τα πεδία που αντιστοιχούν σε referenced πίνακες, ο τρόπος λειτουργίας είναι ο ίδιος με την επιλογή δεδομένων από referenced πίνακα, όπως αναλύθηκε προηγουμένως.



Πατώντας το πλήκτρο ‘Αποθήκευση’ τα νέα δεδομένα αποστέλλονται στο server για έλεγχο και αποθήκευση.

#### 4.2.19 Διαγραφή εγγραφής πίνακα

Ανα πάσα στιγμή, από την οθόνη εμφάνισης δεδομένων πίνακα ο χρήστης μπορεί από τις ενέργειες στην τελευταία κολόνα του HTML πίνακα να πατήσει το πλήκτρο “Διαγραφή” και να διαγράψει την εγγραφή της επιλογής του.


### Τραγούδια


Show  entries Search:

Τραγούδια	Music Type	Artist	Actions
Imagine	Soft Rock	John Lennon 	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Smells like teen spirit	Rock	Nirvana 	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

Showing 1 to 2 of 2 entries Previous  Next

### Προσθήκη δεδομένων

Τραγούδια:	TEXT:	<input type="text"/>
Music Type:	TEXT:	<input type="text"/>
Artist:	REFERENCE:	<input type="text"/> 

 **EMΠ** Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Πατώντας το πλήκτρο διαγραφή εγγραφής εκτελείται το ακόλουθο τμήμα κώδικα το οποίο αναλαμβάνει την εκτέλεση της ενέργειας αυτής.

```
public function deleteTableRow ( $tableId, $id ) {
    $result = 'true';
    $con = DBHelper::beginTransaction();

    $ReferenceDAO = new ReferenceDAO();
    $table = $ReferenceDAO->findTableById ( $tableId );

    if ( $table != null ) {
        $GeneralDAO = new GeneralDAO();
        $result = $GeneralDAO->deleteTableRow ( $con,
        $table['table_name'], $id );

        if ( $result == 'true' && $table['parent_id'] != null &&
        $table['parent_id'] > 0 ) {
            $parentTable = $ReferenceDAO->findTableById (
            $table['parent_id'] );
        }
    }
}
```



```

        $result = $GeneralDAO->nullValueOfTable ( $con,
$parentTable['table_name'], $table['table_name'] . "_id" , $id );
    }


    } else {
        $result = 'false';
    }

    $result = DBHelper::endTransaction( $con , $result );

    return $result;
}

```

Παρατηρούμε την ιδιαιτερότητα αυτής της ενέργειας όπου όταν διαγράφεται μια εγγραφή από έναν πίνακα, αν αυτός ο πίνακας είναι referenced τότε μηδενίζονται όλες εγγραφές του πίνακα “πατέρα” έδειχναν στην υπο-διαγραφή εγγραφή. Αν δε γινόταν αυτό τότε εγγραφές του πίνακα “πατέρα” θα έδειχναν σε εγγραφές που δεν υπάρχουν πια και θα δημιουργούσε πρόβλημα κατά την εμφάνιση αυτών των δεδομένων.

Ομοίως στην εφαρμογή του Android ο χρήστης πατώντας το πλήκτρο  στην οθόνη λεπτομερούς προβολής εγγραφής πίνακα μπορεί να στείλει την εντολή διαγραφή στο server. Όταν η εντολή είναι επιτυχής η οθόνη κλείνει και η εφαρμογή επιστρέφει στην αμέσως προηγούμενη. Παρακάτω βλέπουμε το κομμάτι του κώδικα που στέλνει αυτή την εντολή στο server.

```

private class DeleteTableRowTask extends AsyncTask<String, Integer, Long> {

    protected void onPreExecute() {
    }

    protected Long doInBackground(String... progress) {

        try {
            delteTableRowResult = TableService.deleteTableRow(tableId, rowId);
        } catch (Exception e) {
            Log.e(ThesisConstants.THESIS_TAG, "ERROR", e);
        }

        return 0L;
    }

    protected void onProgressUpdate(Integer... progress) {
        // setProgressPercent(progress[0]);
    }


    protected void onPostExecute(Long result) {
        progDailog.dismiss();
        if ( delteTableRowResult ) {
            finish();
        } else {
        }
    }
}

```

#### 4.2.20 Εμφάνιση δεδομένων τύπου LOCATION στα Google Maps



Στις περιπτώσεις που έχουμε πεδίο πίνακα τύπου LOCATION ο χρήστης καλείται να εισάγει ως δεδομένα ένα ζεύγος συντεταγμένων στη μορφή XX.YYYY,ZZ.WWWW, όπως δηλαδή τις δέχονται τα Google Maps. Στην περίπτωση που έχουμε έγκυρα



δεδομένα τότε εμφανίζεται το πλήκτρο  το οποίο πατώντας το ο χρήστης οδηγείται στο website των Google Maps για να εμφανίσουν το σημείο στο χάρτη. [18]

Artist

Show 10 entries Search:

Artist	Singer Address	Actions
John Lennon	53.377222,-2.881389 	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>
Nirvana	47.619144129874456,-122.28172590470257 	<a href="#">Προβολή</a> <a href="#">Επεξεργασία</a> <a href="#">Διαγραφή</a>

Showing 1 to 2 of 2 entries Previous 1 Next

### Προσθήκη δεδομένων


Artist:	TEXT:	<input type="text"/>
Singer Address:	LOCATION:	<input type="text"/>

Το κομμάτι του κώδικα που δημιουργεί το link για τα Google Maps είναι το ακόλουθο.

```
} else if ( $column['column_type'] == 'LOCATION' &&
StringUtil::notEmptyOrNull($row[$column['column_name']]) &&
TableUtil::validateInput ( 'LOCATION', $row[$column['column_name']]
)) {
    echo "<td>" . $row[$column['column_name']] . "<div><a
href=\"http://maps.google.com/?ie=UTF8&hq=&ll=\" .
$row[$column['column_name']] . "&z=13\"><img width=\"40\"
height=\"40\" src=\"\" . $contextRoot .
\"template/images/google_maps.png\" /></a></div></td>";
}
```

Ομοίως και στο Android αν ένα πεδίο είναι τύπου LOCATION και έχει έγγυρες



τιμές τότε εμφανίζεται το πλήκτρο , το οποίο όταν πατήσει ο χρήστης οδηγείται στους χάρτες. Ακολουθεί το κομμάτι του κώδικα που ανοίγει την εξωτερική εφαρμογή χαρτών.

```
protected void openCoordinatesInExternalMap ( String coordinates ) {
    String uri = "http://maps.google.com/maps?f=d&z=13&daddr=" + coordinates;
    Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(uri));
    this.startActivity(intent);
}
```

#### 4.2.21 Android-Webserver data

Για να λειτουργήσει η εφαρμογή του Android και να εμφανίσει δεδομένα σε όλες τις παραπάνω περιπτώσεις χρειάζεται με κάποιο τρόπο να λάβει αυτή την πληροφορία από το web-server σε μορφή που μπορεί να χειριστεί.

Γι αυτό το λόγο, σε όλες τις περιπτώσεις που χρειάζεται να ανακτηθούν δεδομένα, η εφαρμογή του android τα ζητά από το server μέσω των αρχείων που βρίσκονται στο path "view\external". Ανάλογα με τα δεδομένα που χρειάζεται

- ✓ checkTableHasReferencedTables.php
- ✓ getMasterTable.php
- ✓ getReferencedTables.php
- ✓ getTable.php
- ✓ getTableColumns.php
- ✓ getTableRow.php
- ✓ getTableRows.php

κάθε φορά η εφαρμογή, ζητάει από το αντίστοιχο αρχείο περνώντας τις απαιτούμενες κάθε φορά παραμέτρους την πληροφορία που έχει ανάγκη. Σε όλες τις περιπτώσεις τα δεδομένα, με τη βοήθεια της php function "json\_encode(\$object)", μετατρέπονται σε μορφή JSON και τυπώνονται στην έξοδο των αντίστοιχων αρχείων php. Το παράδειγμα μιας εξόδου από αυτές είναι η ακόλουθη:

```
{"referencedTables":  
[{"0":"4","id":"4","1":"artist","table_name":"artist","2":"3","parent  
_id":"3","3":"Artist","table_display_name":"Artist","4":"0","is_maste  
r":"0"} ]}
```

στο οποίο επιστρέφονται οι τιμές των referenced πινάκων του συστήματος. Η εφαρμογή του Android δέχεται αυτά τα δεδομένα και τα μετατρέπει σε objects τα οποία μπορεί να χρησιμοποιήσει για να τα εμφανίσει.

Για τη μετατροπή του JSON σε αντικείμενα που μπορούν να χρησιμοποιηθούν από την εφαρμογή μας, έχουν κατασκευαστεί διάφορες μέθοδοι μέσα στην class TableService.java στην εφαρμογή του Android. Εκεί με τη βοήθεια του ενσωματωμένου μηχανισμού "JSONObject" του Android καθώς και της βιβλιοθήκης GSON που αναφέρθηκε προηγουμένως έχουμε την επιθυμητή μετατροπή. Ένα παράδειγμα μετατροπής είναι το ακόλουθο:

```
public static List<Table> getReferencedTables () {  
    List<Table> tables = null;  
  
    try {  
        String serverResult =  
NetworkUtil.createGet(ThesisConstants.GET_REFERENCED_TABLES_URL);  
        if ( isValidResult(serverResult) ){  
  
            JSONObject mainObject = new JSONObject(serverResult);  
  
            JSONArray referencedTables =  
mainObject.getJSONArray("referencedTables");  
            if ( referencedTables != null &&  
StringUtil.notEmptyOrNull(referencedTables.toString()) ) {  
                Gson gson = new Gson();  
                Type listType = new TypeToken<ArrayList<Table>>()  
{}.getType();  
                tables = gson.fromJson(referencedTables.toString(),  
listType);  
            }  
        }  
    }  
}
```

```

    }
    } catch (Exception e) {
        Log.e(ThesisConstants.THESIS_TAG, "ERROR DESERIALIZE JSON",
e);
    }

    return tables;
}

```

Όπου βλέπουμε πως με τη χρήση των προαναφερθέντων εργαλείων, το JSON που παρουσιάσαμε νωρίτερα μετατρέπεται σε μία `List<Table>`, δηλαδή μία λίστα από objects τύπου `Table`. Ομοίως γίνονται οι μετατροπές και των υπολοίπων δεδομένων.

#### 4.2.22 Android Social media Share

Μία άλλη δυνατότητα που υπάρχει στην εφαρμογή του Android είναι η κοινοποίηση κάποιας εγγραφής πίνακα σε διάφορα social media. Ο χρήστης στην οθόνη όπου βλέπει τις λεπτομέρειες μιας εγγραφής μπορεί να πατήσει το πλήκτρο



οπότε και το σύστημα δημιουργεί ένα κείμενο που παρουσιάζει την επιλεγμένη εγγραφή και εμφανίζει στον χρήστη μια λίστα από διαθέσιμες εφαρμογές στις οποίες μπορεί να κοινοποιήσει τα δεδομένα αυτά.

Στο ακόλουθο κομμάτι κώδικα βλέπουμε τη μέθοδο η οποία προετοιμάζει το κείμενο της εγγραφής προς κοινοποίηση και τη μέθοδο που εκτελεί το "Intent" του Android για κοινοποίηση κειμένου.

```

public void shareRowButton (View v) {
    String shareText = ThesisConstants.APPLICATION_NAME + " : " +
table.getTableDisplayName() + " -> ";
    String comma = "";
    for ( Column column : columns ) {
        String rowText = comma + column.getColumnDisplayName() + " :
";

        Boolean addColumn = false;
        if ( !column.getColumnType().equals("REFERENCE") ) {
            String columnValue = row.get(column.getColumnName());
            if ( StringUtil.isEmptyOrNull(columnValue) ) {
                rowText += row.get(column.getColumnName());
                addColumn = true;
            }
        } else {
            Map<String,String> referencedRow =
referencedTablesRows.get(column.getColumnName());
            if ( referencedRow != null ) {
                String columnValue = row.get(column.getColumnName());
                if ( StringUtil.isEmptyOrNull(columnValue) ) {
                    rowText += referencedRow.get("value");
                    addColumn = true;
                }
            }
        }
        if ( addColumn ) {
            shareText += rowText;
            comma = " , ";
        }
    }
}

```

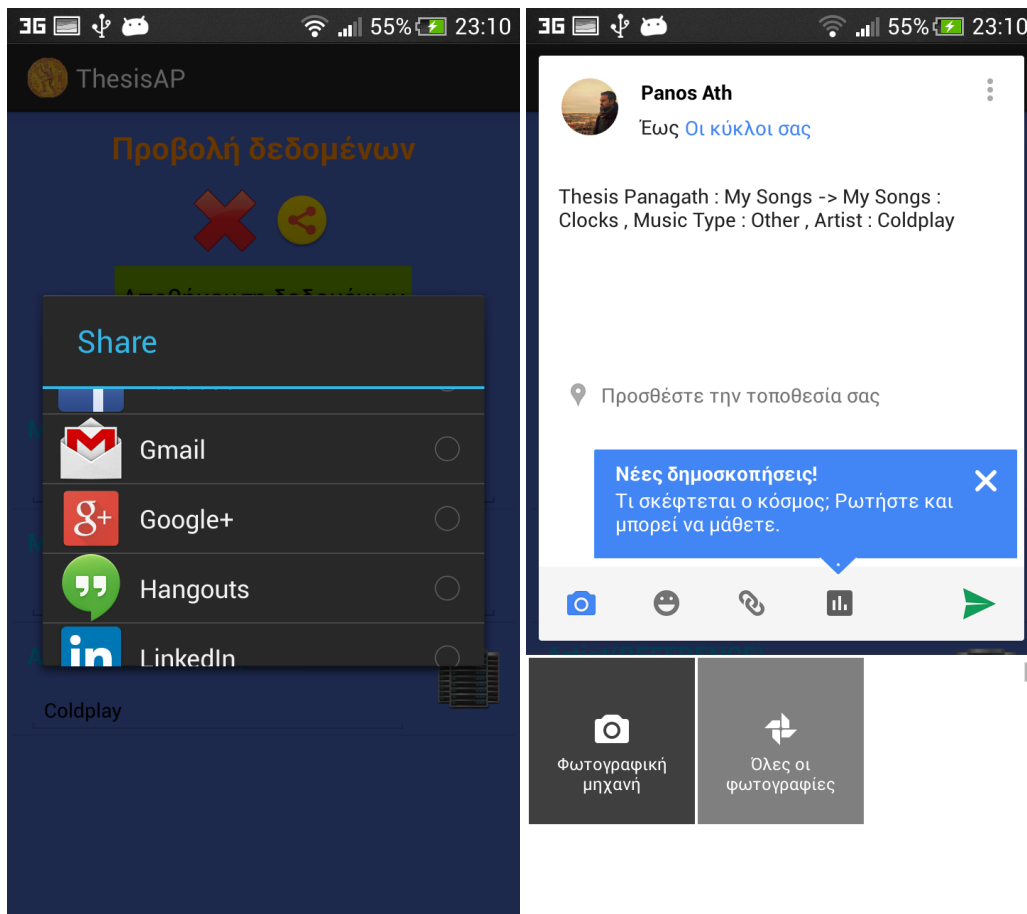
```

    }
    startActivity(Intent.ACTION_SEND);
}

protected void shareIntent( String message, String dialogTitle) {
    Intent intent = new Intent(Intent.ACTION_SEND );
    intent.setType("text/plain");
    intent.putExtra(Intent.EXTRA_TEXT, message);
    startActivity(Intent.createChooser(intent, dialogTitle));
}

```

Στις παρακάτω εικόνες βλέπουμε τη λειτουργία αυτή καθώς και την εμφάνιση μέσα στην εφαρμογή Google+.<sup>[19]</sup>



## Βιβλιογραφία

1. Mark Watson: Java Programming
2. Wikipedia – [www.wikipedia.org](http://www.wikipedia.org)
3. PHP Official website - <http://php.net>
4. Paul J. Deitel, Harvey M. Deitel, Abbey Deitel: Android for Programmers An App-Driven Approach, 2<sup>nd</sup> edition
5. David, Sklar, Adam, Trachtenberg: PHP Cookbook
6. Gray B., Shelly, Denise, Woods: HTML, XHTML AND CSS, 6<sup>th</sup> edition
7. Apache Official website - <http://httpd.apache.org>
8. Paul N., Weinberg, James R., Groff : SQL
9. XAMPP Official website - <https://www.apachefriends.org>
10. Lynn Beighley, Cody Lindley – jQuery for Dummies
11. PHPMyAdmin Official website - <http://www.phpmyadmin.net>
12. Gson Official website - <https://code.google.com/p/google-gson/>
13. Datatables Official website - <http://datatables.net/>
14. Kathy Sierra, Bert Bates: Head First Java
15. Software Licenses in Plain English - <https://tldrlegal.com/>
16. all-free-download Free website templates - <http://all-free-download.com/free-website-templates/>
17. IconFinder free icons - <https://www.iconfinder.com/>
18. Google Maps - <https://maps.google.com/>
19. Google+ - <https://plus.google.com>
20. Android Official website - <http://www.android.com/>
21. w3schools the world's largest web development site - <http://www.w3schools.com/>
22. Java Official Website - <http://java.com/>

\*<sup>1</sup> – Όλες οι παραπομπές σε ιστοσελίδες στηρίζονται στα δεδομένα που παρουσιάζονταν από αυτές κατά τη διάρκεια του Οκτωβρίου 2014.

\*<sup>2</sup> – Τα πνευματικά δικαιώματα των λογοτύπων και των κατοχυρωμένων ονομασιών ανήκουν στους αντίστοιχους ιδιοκτήτες τους.