The page features a decorative design with three overlapping blue circles of varying sizes and shades, arranged in a descending diagonal line from the top right towards the bottom right. Thin blue lines also extend from the top left towards the bottom right, creating a sense of movement and structure.

**Α.Τ.Ε.Ι ΜΕΣΟΛΛΟΓΙΟΥ**  
**ΠΡΟΓΡΑΜΜΑ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ**  
**ΤΜΗΜΑΤΟΣ ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΔΙΚΤΥΩΝ**

## **Πτυχιακή εργασία**

Κατασκευή εφαρμογής σε λειτουργικό  
Android με διασύνδεση σε  
απομακρυσμένη βάση δεδομένων

Τιμολή Γιάννης 0912  
Νικολουδάκη Δήμητρα 0469

4/4/2014

Κατασκευή εφαρμογής σε λειτουργικό Android με  
διασύνδεση σε απομακρυσμένη βάση δεδομένων

Τιμολή Γιάννης  
Νικολουδάκη Δήμητρα

ΠΡΟΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ  
ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ  
ΔΙΚΤΥΩΝ

Επιβλέπων καθηγητής  
Ασημακόπουλος Γεώργιος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την  
04/04/2014

Ονοματεπώνυμο 1 Ονοματεπώνυμο 2 Ονοματεπώνυμο 3

.....

.....

Τιμολή Γιάννης

.....

Νικολουδάκη Δήμητρα

.....

## Περίληψη

Το παρόν κείμενο χωρίζεται σε δύο κύριες ενότητες. Στην πρώτη ενότητα αναφέρονται κάποια στοιχεία και πληροφορίες σχετικά με το πώς δημιουργήσαμε τη βάση δεδομένων μας και το site που θα έχουμε την δυνατότητα να κάνουμε ηλεκτρονική κράτηση ειτηρίων υπεραστικών λεωφορείων για τον νομό Αχαΐας.

Στην δεύτερη ενότητα αναφέρονται κάποιες εισαγωγικές πληροφορίες και ορισμοί σχετικά με το Android αλλά και ο τρόπος με τον οποίο κάποιος μπορεί να χρησιμοποιήσει για πρώτη φορά το πρόγραμμα eclipse για να κάνει μια εφαρμογή.

Στη συνέχεια φτιάξαμε μια εφαρμογή από όπου επίσης μπορούμε να κάνουμε ηλεκτρονική κράτηση εισητηρίων όπως και στο site μας.

# Περιεχόμενα:

## Μέρος Α

### Κεφάλαιο 1ο

1. Εισαγωγή στην γλώσσα Mysql.....	6
1.1. Βασικές εντολές διαχείρισης μιας βάσης .....	7
1.2. Τύποι δεδομένων.....	8
1.3. Ιδιότητες επί των στηλών.....	11
1.4. Δημιουργία πίνακα και ακεραιότητα αναφορών.....	12
1.5. Αρχικοποίηση πίνακα.....	14
1.6. Ερωτήματα (Queries).....	15
1.7. Διαγραφές πινάκων και εγγραφών.....	17
1.8. Ενημέρωση δεδομένων πινάκων.....	18
1.9. Τροποποίηση σχήματος πίνακα.....	18

### Κεφάλαιο 2ο

2. Εισαγωγή στην γλώσσα PHP .....	20
2.1. Λειτουργία της PHP.....	22
2.2. Συντακτικό της PHP.....	22
2.3. Μεταβλητές , πίνακες και αλφαριθμητικά.....	24
2.4. Δομές επιλογής και επανάληψης.....	26
2.5. Πίνακες.....	27
2.6. Διασύνδεση της PHP με την Mysql .....	29
2.6.1 Σύνδεση ενός προγράμματος PHP με τον MYSQL SERVER.....	32

### Κεφάλαιο 3ο

3. Δημιουργία Βάσης Δεδομένων Εφαρμογής.....	35
--	----

### Κεφάλαιο 4ο

4. Ιστότοπος για πρόσβαση στα δεδομένα της βάσης.....	40
4.1. Δομή – Οργάνωση – Λειτουργία του ιστότοπο .....	40
4.2. Ενημέρωση κοινού για δρομολόγια.....	41
4.3. Ηλεκτρονική Κράτηση Εισιτηρίου.....	45
4.4. Λειτουργία ηλεκτρονικής κράτησης.....	55

## Μέρος Β

### Κεφάλαιο 5<sup>ο</sup>

5. Τι είναι το Android.....	59
5.1. Η εξέλιξη του Android.....	60
5.2. Εισαγωγή στο Περιβάλλον Ανάπτυξης.....	64
5.3. Δημιουργία ενός Android Project με το Eclipse.....	65
5.4. Η αρχιτεκτονική δομή ενός android project.....	70
5.5. Το αρχείο Android Manifest (xml).....	72

### Κεφάλαιο 6<sup>ο</sup>

6. Κώδικας Android.....	74
6.1 Βιβλιοθήκες-Κατηγορίες βιβλιοθηκών.....	74
6.2 MainActivity.....	78
6.3 LazyAdapter.....	90
6.4 RSSItem .....	91
6.5 RSSFeed.....	92
6.6 RSSHandler.....	92
6.7 Show Details .....	93

### Κεφάλαιο 7<sup>ο</sup>

7. Αρχεία php .....	97
7.1 Routes.php.....	97
7.2 Reservation.php.....	100

## Μέρος Γ

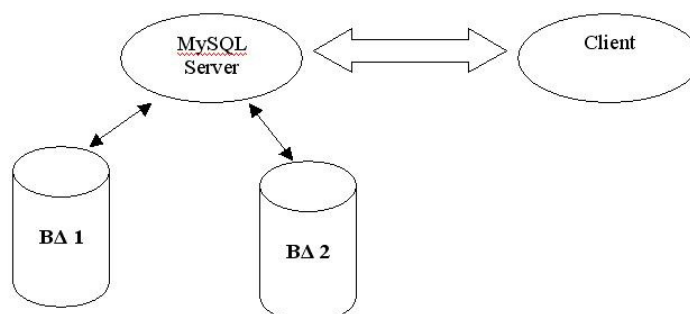
Βιβλιογραφία-Πηγές.....	103
-------------------------	-----

# Μέρος Α

## Κεφάλαιο 1<sup>ο</sup>

### 1. Εισαγωγή στην γλώσσα Mysql

Η Mysql είναι ένα ΣΔΒΔ όπου γίνεται διαχείριση των δεδομένων μιας βάσης που υπάρχει στα μέσα αποθήκευσης ενός υπολογιστή host . Υποστηρίζει τις βασικές λειτουργίες διαχείρισης μιας ΒΔ, όπως ο χειρισμός πινάκων (δημιουργία, τροποποίηση διαγραφή), τη διαχείριση εγγραφών (εισαγωγή, τροποποίηση, διαγραφή) και την ανάκτηση δεδομένων τα οποία ικανοποιούν συγκεκριμένα κριτήρια. Η επικοινωνία διαφόρων χρηστών που θέλουν να διαχειριστούν την βάση (που είναι αποθηκευμένη στον υπολογιστή host) γίνεται με βάση το μοντέλο client – server. Οι χρήστες (clients) , από τον προσωπικό τους υπολογιστή, που υπάρχει εγκατεστημένο το λογισμικό ΣΔΒΔ Mysql, μέσω αυτού του λογισμικού υποβάλλουν ερωτήματα διαχείρισης της βάσης που υπάρχει αποθηκευμένη στον υπολογιστή host (server) και εκείνος με τη σειρά του, αφού τα επεξεργαστεί, επιστρέφει αποτελέσματα της εντολής που του στάλθηκε, όπως φαίνεται από το παρακάτω σχήμα :



Τα αποτελέσματα αυτά μπορεί να είναι ανάκτηση δεδομένων που ζητήθηκαν, γνωστοποίηση επιτυχούς ενημέρωσης της βάσης ή αποτυχία εκτέλεσης ερωτήματος λόγω λάθους σύνταξης του. Η Mysql είναι πολυνηματική , που σημαίνει ότι μπορεί να δεχθεί και να επεξεργαστεί ταυτόχρονα περισσότερες από μια κλήσεις – ερωτημάτων χρηστών.

Επιπλέον δεν είναι απαραίτητο ο υπολογιστής host της βάσης να είναι συμβατός ως προς υλικό ή λογισμικό με τον υπολογιστή ενός χρήστη, που σημαίνει ότι η Mysql δεν θέτει περιορισμούς στα λειτουργικά συστήματα των υπολογιστών που επικοινωνούν μέσω αυτής. Για παράδειγμα ο υπολογιστής host μπορεί να έχει λειτουργικό σύστημα Unix και ένας χρήστης που θέλει να προσπελάσει την βάση του host, μπορεί να έχει Windows.

Από μία γραμμή εντολών του λειτουργικού μας συστήματος ( και δεδομένου ότι είμαστε στον κατάλληλο φάκελο εργασίας), μπορούμε να συνδεθούμε στο MySQL server, εκτελώντας την εντολή:

**mysql -h *host* -u *user* -p,** (1)

όπου στη θέση του *host* πρέπει να τοποθετήσουμε το όνομα ή την IP διεύθυνση του υπολογιστή στον οποίο εκτελείται ο MySQL server. Αν εκτελούμε το πρόγραμμα πελάτη στον ίδιο υπολογιστή με τον MySQL server, μπορούμε να παραλείψουμε το τμήμα *-h host*. Το *user* πρέπει να είναι το όνομα του χρήστη που έχει οριστεί για αυτόν στη MySQL.

Παρακάτω θα παρουσιάσουμε ορισμένες βασικές εντολές της Mysql, ταξινομημένες σε κατηγορίες. Δεδομένου ότι η εντολή (1) ενεργοποίησης της Mysql έχει εκτελεστεί επιτυχώς, τότε το σύμβολο προτροπής του λειτουργικού συστήματος αλλάζει σε mysql> . Κάθε εντολή Mysql θα την παρουσιάζουμε μετά το σύμβολο προτροπής σε πλάγια γραφή. Όλοι οι προαιρετικοί όροι μιας εντολής θα περικλείονται σε αγκύλες ([ ]).

## 1.1 Βασικές εντολές διαχείρισης μιας βάσης

Η δημιουργία μιας ΒΔ γίνεται με την παρακάτω εντολή Mysql :

```
mysql> CREATE DATABASE [IF NOT EXISTS] dbname;
```

όπου ο προαιρετικός όρος *IF NOT EXISTS*, εξασφαλίζει την αποτροπή εμφάνισης σφάλματος από το σύστημα αν υπάρχει μια βάση με το ίδιο όνομα. Αν υπάρχει αυτή η βάση, χωρίς αυτό τον όρο το σύστημα εμφανίζει σφάλμα, διαφορετικά μια προειδοποίηση. *Dbname* είναι το όνομα της βάσης που δίνουμε.

Μια βάση γίνεται τρέχουσα, δηλαδή οι πίνακες που δημιουργούμε να αποθηκεύονται εκεί καθώς και οι προς χρήση πίνακες για

τροποποίηση και ανάκτηση πληροφοριών να είναι αποθηκευμένοι εντός αυτής, με την παρακάτω εντολή :

```
Mysql>USE dbname;
```

Η διαγραφή μιας ΒΔ γίνεται με την παρακάτω εντολή :

```
Mysql>DROP DATABASE [IF EXISTS] dbname;
```

Ο προαιρετικός όρος *IF EXISTS* χρησιμεύει στην αποτροπή εμφάνισης σφαλμάτων από το σύστημα σε περίπτωση απόπειρας διαγραφής ανύπαρκτης ΒΔ.

Η παρακάτω εντολή εμφανίζει του πίνακες της τρέχουσας βάσης ..

```
Mysql>SHOW TABLES;
```

Αφού δημιουργήσουμε μια βάση και την επιλέξουμε ως τρέχουσα θα πρέπει, δεδομένου ότι έχουν ολοκληρωθεί τα στάδια της λήψης απαιτήσεων και της δημιουργίας του ΣΜ, εντός αυτής να εισάγουμε τους πίνακες (σχέσεις ). Κάθε πίνακας αναπαριστά μια οντότητα του μικρόκοσμου που θα εφαρμοστεί η βάση μας και οι στήλες του αναπαριστούν τα χαρακτηριστικά της οντότητας. Οι γραμμές του πίνακα (πλειάδες) αναπαριστούν ένα στιγμιότυπα της οντότητας με συγκεκριμένες τιμές στα χαρακτηριστικά του.

Για να ορίσουμε ένα πίνακα σε Mysql θα πρέπει αφού του δώσουμε ένα όνομα, στη συνέχεια να ορίσουμε τις στήλες του. Αυτό γίνεται δίνοντας σε κάθε στήλη ένα όνομα και καθορίζοντας ακολούθως τον τύπο δεδομένων του. Ο τελευταίος προσδιορίζει το είδος και το εύρος των τιμών μιας στήλης. Κατά τον ορισμό μιας στήλης πίνακα μπορούμε να καθορίσουμε επιπλέον και άλλους περιορισμούς όπως αν επιτρέπονται οι NULL τιμές, το εύρος πλάτους προβολής, και προκαθορισμένες τιμές σε περίπτωση που ο χρήστης δεν ορίσει ρητά μια τιμή για συγκεκριμένη στήλη.

## 1.2 Τύποι δεδομένων

Θα παρουσιάσουμε τους κυριότερους τύπους δεδομένων των στηλών της Mysql σε συνδυασμό με πληροφορίες σχετικά με το εύρος τιμών τους, την ποσότητα αποθηκευτικού χώρου που καταλαμβάνουν . Θα παρουσιάσουμε τους τύπους δεδομένων χωρισμένους σε κατηγορίες



ανάλογα με το είδος των δεδομένων που περιέχουν (αριθμητικά δεδομένα, χαρακτήρες, καθορισμένες τιμές).

#### Αριθμητικοί τύποι δεδομένων

Τύπος	Ποσότητα αποθήκευσης (byte)	Εύρος με πρόσημο	Εύρος χωρίς πρόσημο	Παρατηρήσεις
TINYINT	1	-128 - 127	0 - 255	Για ακέραιες τιμές ανάλογα με το εύρος τιμών των δεδομένων. Η μη προσημασμένη μορφή του ακεραίου διπλασιάζει το εύρος τιμών.
SMALLINT	2	-32768 - 32767	0 - 65535	
MEDIUMINT	3	-8.388.608 – 8.388.607	0 – 16.777.215	
INT	4	-2.147.683.648 – 2.147.483.647	0 – 16.777.215	
BIGINT	8	9.223.372.036.8 54.775.808 – 9.223.372.036.8 54.775.807	0 – 18.446.744.073.709.551.651	
FLOAT(M,D)	4	Όλοι οι δεκαδικοί <u>κινητής υποδιαστολής</u> με εύρος τιμών όλους τους δεκαδικούς αριθμούς M (ακρίβεια) ψηφίων συνολικά από τα οποία τα D (κλίμακα) θα είναι μετά την υποδιαστολή		Αναπαριστούν τους αριθμούς κινητής υποδιαστολής απλής(M<24) και διπλής ακρίβειας(M<53) αντίστοιχα. Δέχονται δύο παραμέτρους που αντιπροσωπεύουν την ακρίβεια και κλίμακα αντίστοιχα. Χρησιμοποιούνται σε εφαρμογές που απαιτείται ταχύτητα υπολογισμών σε βάρος της ακρίβειας αποτελέσματος.
DOUBLE(M,D)	8			
DECIMAL(M,D)	8	Δεκαδικοί αριθμοί <u>σταθερής υποδιαστολής</u> με εύρος τιμών όλους τους δεκαδικούς αριθμούς M (ακρίβεια) ψηφίων συνολικά από τα οποία τα D (κλίμακα) θα είναι μετά την υποδιαστολή		Αναπαριστούν τους αριθμούς σταθερής υποδιαστολής. Δέχονται δύο παραμέτρους που αντιπροσωπεύουν στην ακρίβεια και κλίμακα αντίστοιχα. Χρησιμοποιούνται σε εφαρμογές που απαιτείται

			ακρίβεια υπολογισμών (οικονομικές εφαρμογές).
BIT(n)	(n+7)/8		0 – (2 <sup>n</sup> -1) Όταν θέλουμε να εισάγουμε ακολουθίες από bits. π.χ '001111' μια τέτοια τιμή. Η παράμετρος n αναπαριστά το μέγιστο μήκος της δυαδικής ακολουθίας του πεδίου.

### Τύποι δεδομένων χαρακτήρων

Τύπος	Ποσότητα αποθήκευσης (byte)	Μέγιστο μήκος χαρακτηροσειράς	Παρατηρήσεις
CHAR(M)	M	255	Μπορούν να αποθηκευτούν το πολύ M χαρακτήρες. Αν δοθεί μικρότερου μήκους χαρακτηροσειρά οι υπολειπόμενες θέσεις θα γεμίσουν με μηδενικά.
VARCHAR(M)	L + (1-2) bytes	65535	Μπορούν να αποθηκευτούν το πολύ M χαρακτήρες. Αν η προς αποθήκευση χαρακτηροσειρά έχει μήκος L<M τότε αποθηκεύονται L χαρακτήρες συν 1(αν L<=256) ή 2 (αν L>256) byte. Δεν γίνεται προσθήκη μηδενικών στο τέλος της χαρακτηροσειράς.
TINYTEXT(M)	L + 1 byte	255	Λειτουργούν όπως και οι προηγούμενος τύπο δεδομένων και ανάλογα με το πλήθος των χαρακτήρων που θέλουμε να εισάγουμε επιλέγουμε τον καταλληλότερο από αυτούς.
TEXT	L + 2 bytes	65535	
MEDIUMTEXT(M)	L + 3 bytes	16.777.215	
LONGTEXT (M)	L + 4 bytes	4.294.967.295	

### Τύποι δεδομένων συνόλου

- **Enum** : Τύπος απαρίθμησης, δηλαδή μία λίστα στοιχείων. Χρησιμοποιούμε τον τύπο ENUM σε μια στήλη ,όταν θα πρέπει να πάρει ένα οποιαδήποτε στοιχείο από μια λίστα τιμών. Μετά την δεσμευμένη λέξη enum ακολουθεί μία λίστα στοιχείων από τα οποία μπορεί να επιλεγθεί μία τιμή ( ή μπορεί να είναι και NULL). Πχ αν θέλουμε η στήλη να περιέχει τιμές A,B ή C τότε ορίζουμε τον τύπο του ως: ENUM('A','B','C').

- **Set** : Παρόμοιος με τον προηγούμενο τύπο απαρίθμησης enum, μόνο που από μια λίστα διαθέσιμων τιμών ανάλογα με τα δεδομένα του προβλήματος μας μπορούμε να επιλέξουμε περισσότερες από μια τιμές για μια στήλη. πχ για SET('A','B','C') κάποιο πεδίο μπορεί να πάρει μεταξύ άλλων την τιμή A,B ή την τιμή A,B,C για παράδειγμα, δηλαδή περισσότερα από ένα στοιχεία από την λίστα που ακολουθεί την δεσμευμένη λέξη Set.

Χρονικοί τύποι δεδομένων

- **DATE**: Τιμή ημερομηνίας με μορφή: EEEE-MM-HH
- **DATETIME**: Ημερομηνία και ώρα με μορφή: EEEE-MM-HH ΩΩ:ΛΛ:ΔΔ
- **TIME**: Ωρα με μορφή: ΩΩ:ΛΛ:ΔΔ
- **YEAR(M)**: Αποθηκεύει έτος σε διψήφια ή τετραψήφια μορφή. Στη διψήφια μορφή περιλαμβάνονται τα έτη από 1970-2069. Στην τετραψήφια από 1901-2155.
- **TIMESTAMP**: Χρονική ένδειξη με μορφές: 14(EEEEMMHHΩΩΛΛΔΔ) 12(EEMMHHΩΩΛΛΔΔ) 8(EEEEMMHH) 6(EEMMHH). Χρησιμοποιείται σε αντίθεση με τον τύπο DATETIME μόνο σε περιπτώσεις όπου θέλουμε να αποθηκεύσουμε την ημερομηνία – ώρα μιας λειτουργίας εισαγωγής ή τροποποίησης μιας εγγραφής σε ένα πίνακα.

### 1.3 Ιδιότητες επί των στηλών

Αφού δώσουμε ένα κατάλληλο όνομα σε μια στήλη και ορίσουμε τον τύπο δεδομένων της, στη συνέχεια μπορούμε να ορίσουμε και επιπλέον ιδιότητες . Οι ιδιότητες αυτές θα παρουσιαστούν σε κατηγορίες ανάλογα με το είδος των δεδομένων που υποστηρίζει κάθε τύπος .

Γενικές ιδιότητες στηλών

- Οι ιδιότητες *NULL* και *NOT NULL* επιτρέπουν και απαγορεύουν αντίστοιχα την χρήση *NULL* τιμών σε μια στήλη ενός πίνακα. Όταν δεν υπάρχει καμιά από τις δύο αυτές ιδιότητες τότε εξορισμού επιτρέπεται η χρήση *NULL* τιμών για αυτή τη στήλη.

- Η ιδιότητα *DEFAULT* ακολουθείται από μια τιμή που είναι η προκαθορισμένη τιμή για αυτή την στήλη. Αν ο χρήστης κατά την διαδικασία εισαγωγής μιας πλειάδας, παραλείψει να εισάγει μια τιμή για αυτό το πεδίο, τότε εκχωρείται αυτόματα η προκαθορισμένη τιμή για αυτήν την πλειάδα.

- Η ιδιότητα *PRIMARY* ορίζει μια στήλη σαν πρωτεύον κλειδί μιας σχέσης.

- Η ιδιότητα *UNIQUE* ορίζει μια στήλη σαν χαρακτηριστικό μοναδικής τιμής, που σημαίνει ότι δεν μπορεί να υπάρχουν δύο εγγραφές του πίνακα που να έχουν την ίδια τιμή στην συγκεκριμένη στήλη. Η χρήση αυτής της ιδιότητας δεν αποκλείει της *NULL* τιμές στην στήλη.

Αριθμητικές ιδιότητες στηλών :

- Η *UNSIGNED* απαγορεύει αρνητικές τιμές.

- Η *AUTO\_INCREMENT* εφαρμόζεται μόνο σε ακεραίους τύπους δεδομένων και χρησιμοποιείται για την παραγωγή ακολουθιών συνεχόμενων μοναδικών τιμών. Επιπλέον δεν επιτρέπονται οι *NULL* τιμές σε πεδίο με ιδιότητα *AUTO\_INCREMENT*. Σε έναν πίνακα επιτρέπεται μόνο μια στήλη να έχει αυτή την ιδιότητα και η στήλη αυτή ορίζεται ως πρωτεύον κλειδί.

Ιδιότητες στηλών συμβολοσειράς

- Η *CHARACTER SET* καθορίζει το σύνολο χαρακτήρων που θα χρησιμοποιηθούν για την στήλη.

- Η *COLLATE* προσδιορίζει την αντιπαραβολή του συνόλου χαρακτήρων (π.χ. αν θα κάνει διάκριση πεζών ή κεφαλαίων χαρακτήρων, τονισμένων ή μη χαρακτήρων).

## 1.4 Δημιουργία πίνακα και ακεραιότητα αναφορών

Η δημιουργία ενός πίνακα γίνεται με την παρακάτω εντολή :

```
CREATE TABLE [IF NOT EXISTS] όνομα_πίνακα(  
όνομα_στήλης τύπος_στήλης [options],
```

*όνομα\_στήλης2* *τύπος\_στήλης2* [*options2*], κλπ..  
**PRIMARY KEY** (*όνομα\_στήλης\_κλειδιού1*,...)  
**UNIQUE** (*index\_col\_name*,...)  
 [**CONSTRAINT** *symbol*] **FOREIGN KEY** (*index\_col\_name*,...)  
**REFERENCES** *tbl\_name* [(*index\_col\_name*,...)] [**ON DELETE**  
*reference\_option*] [**ON UPDATE** *reference\_option*]

Όπου **options**=[**NOT NULL** | **NULL**] [**DEFAULT** *default\_value*]  
 [**AUTO\_INCREMENT**] και **reference option**=[**RESTRICT** | **CASCADE**  
 | **SET NULL** | **NO ACTION** | **SET DEFAULT** ]

Αρχικά δίνουμε το όνομα πίνακα και στη συνέχεια ορίζουμε κάθε στήλη του δίνοντας ένα όνομα ακολουθούμενο από τον τύπο δεδομένων του, με τυχόν ιδιότητες. Οι ιδιότητες **PRIMARY KEY** και **UNIQUE** μπορούν να γραφούν ξεχωριστά από την δήλωση μιας στήλης όπως φαίνεται από την παραπάνω δομή της εντολής δημιουργίας πίνακα. Επιπλέον με την ιδιότητα **FOREIGN KEY** μπορούμε να δηλώσουμε ένα πεδίο σε ένα πίνακα σαν ξένο κλειδί και ακολούθως μετά την δεσμευμένη λέξη **REFERENCES** δηλώνουμε το πίνακα και το πρωτεύον κλειδί του από το οποίο το ξένο κλειδί παίρνει τιμές. Επιπλέον με τους προαιρετικούς όρους **ON DELETE** και **ON UPDATE** δηλώνουμε τον τρόπο με τον οποίο εξασφαλίζουμε **ακεραιότητα αναφορών** (*referential integrity*) σε περίπτωση διαγραφής μιας εγγραφής από τον πίνακα που είναι στην πλευρά ένα και ενημέρωσης ενός πρωτεύοντος κλειδιού του αντίστοιχα.

Ας δούμε μερικές από τις επιλογές αυτές που εξασφαλίζουν ακεραιότητα αναφορών σε περίπτωση διαγραφής( με όμοιο τρόπο και σε περίπτωση ενημέρωσης πρωτεύοντος κλειδιού) :

- **CASCADE** : Όταν γίνει διαγραφή μιας εγγραφής ενός πίνακα στην πλευρά ένα, αυτόματα διαγράφονται όλες οι συσχετιζόμενες εγγραφές από τους συνδεδεμένους με αυτόν πίνακες.
- **SET NULL** : Σε περίπτωση διαγραφής μιας εγγραφής ενός πίνακα στην πλευρά ένα μιας συσχέτισης, στις σχετιζόμενες εγγραφές τοποθετείται αυτόματα από το σύστημα στα πεδία ξένου κλειδιού η τιμή **NULL**.
- **NO ACTION** : Η διαγραφή μιας εγγραφής αποτρέπεται από το σύστημα (εμφάνιση σφάλματος) όταν υπάρχουν σχετιζόμενες εγγραφές σε άλλον πίνακα.

## 1.5 Αρχικοποίηση πίνακα

Η καταχώρηση συγκεκριμένων δεδομένων σε ένα πίνακα γίνεται με την εντολή *INSERT*, που είναι της εξής μορφής :

```
INSERT [INTO] table [(column1,column2...)]  
VALUES (value1,value2,...);
```

Όπου καταχωρούμε στις στήλες που εμφανίζονται μετά την λέξη *table* (*column1*, *column2*) τις τιμές *value1,value2*. Αυτή η μορφή χρησιμοποιείται όταν θέλουμε να εισάγουμε σε συγκεκριμένες στήλες τιμές. Για να εισάγουμε τιμές σε κάθε στήλη του πίνακα μπορούμε να παραλείψουμε τον όρο *[(column1,column2...)]*. Σε κάθε περίπτωση θα πρέπει να υπάρχει ένα προς ένα αντιστοιχία μεταξύ των στηλών και των τιμών ως προς το πλήθος και τον τύπο δεδομένων. Αν θέλω να εισάγω πολλές πλειάδες τότε η μορφή της εντολής είναι :

```
INSERT [INTO] table VALUES  
(value11,value12,..., value1n),  
(value21,value22,..., value2n),  
...  
(valuen1,valuen2,..., valuenn);
```

Άλλη μορφή εντολής εισαγωγής μιας μόνο πλειάδας είναι η :

```
INSERT [INTO] table set  
column1="value1", column2="value2",...;
```

Σε κάθε περίπτωση όταν δεν αναφέρεται ρητά μια στήλη σε μια εντολή *INSERT*, τότε μπορεί να αρχικοποιηθεί με έναν από τους παρακάτω τύπους

- Αν κατά τον ορισμό του πίνακα έχει ανατεθεί σε μια στήλη η ιδιότητα *DEFAULT* τότε εκχωρείται αυτόματα στην στήλη η τιμή που ακολουθεί την ιδιότητα *DEFAULT*.
- Αν δεν έχει η στήλη ούτε την ιδιότητα *DEFAULT* ούτε την *NOT NULL*, τότε ανατίθεται έμμεσα στην στήλη η ιδιότητα *DEFAULT NULL*. Στην περίπτωση ελλιπούς αναφοράς στην μεταβλητή κατά την εντολή *INSERT*, εκχωρείται σε αυτή αυτόματα η τιμή *NULL*.

- Αν δεν υπάρχει η ιδιότητα *DEFAULT* αλλά υπάρχει η *NOT NULL*, τότε εκχωρείται στην στήλη μια καθορισμένη τιμή ανάλογα με τον τύπο δεδομένων του. Για παράδειγμα αν η στήλη είναι ακέραιων αριθμών τότε της εκχωρείται η τιμή 0, αν είναι χαρακτηροσειρά η τιμή ‘ ‘ ενώ αν είναι ENUM της εκχωρείται το πρώτο στοιχείο της λίστας. Για να γίνει αυτό θα πρέπει να είναι ενεργοποιημένος ο χαλαρός έλεγχος τύπων της Mysql (χρήση μεταβλητής *sqlmode*). Περισσότερες λεπτομέρειες σχετικά με προκαθορισμένες τιμές και την εσωτερική μεταβλητή *sqlmode* υπάρχουν στο εγχειρίδιο τεκμηρίωσης της γλώσσας.

- Αν η στήλη έχει ιδιότητα *AUTO\_INCREMENT* τότε αυτόματα παίρνει τον επόμενο αύξων αριθμό, ενώ αν είναι τύπου *TIMESTAMP* παίρνει την ημερομηνία και ώρα που γίνεται η εντολή *INSERT*.

## 1.6 Ερωτήματα (Queries)

Μπορούμε να λαμβάνουμε πληροφορίες από μια ΒΔ χρησιμοποιώντας κατάλληλα ερωτήματα, που έχουν γενική μορφή :

Στήλες

**SELECT** *items*

**FROM** *tables* Πίνακες

[**WHERE** *condition*] Συνθήκη

[**GROUP BY** *group\_type*]

[**HAVING** *where\_definition*]

[**ORDER BY** *order\_type*]

[**LIMIT** *limit\_criteria*]

Ας δούμε τους όρους του ερωτήματος αναλυτικά με παραδείγματα :

- **SELECT** : Δέχεται μια ή περισσότερες στήλες για τις οποίες θέλουμε να λάβουμε πληροφορίες για τα περιεχόμενα τους. Τα αποτελέσματα εξόδου εμφανίζουν μόνο τις στήλες αυτές. Μπορούμε να αλλάξουμε το όνομα μιας στήλης στα αποτελέσματα εξόδου χρησιμοποιώντας τον όρο *AS*. Σε περίπτωση που θέλουμε να λάβουμε πληροφορίες για όλες τις στήλες του πίνακα τοποθετούμε το σύμβολο \*.

- **FROM** : Προσδιορίζουμε τον πίνακα, που περιέχει τις στήλες για τις οποίες θέλουμε να λάβουμε δεδομένα. Μπορούμε στην εντολή SELECT να τοποθετήσουμε στήλες από συσχετιζόμενους πίνακες. Σε αυτή την περίπτωση, στον όρο from θα πρέπει να τοποθετήσουμε τους συσχετιζόμενους πίνακες χωρισμένους με κόμμα.

- **WHERE** : Τοποθετούμε την συνθήκη του ερωτήματος. Περιλαμβάνει πεδία πινάκων , συγκριτικούς τελεστές, λογικούς τελεστές (and or) καθώς και σταθερές. Οι πλειάδες των πινάκων που οι τιμές των στηλών τους , που εμφανίζονται στην συνθήκη την ικανοποιούν, είναι το αποτέλεσμα εξόδου του ερωτήματος.

I. =,<>,>,>=,<,<= : χρησιμοποιούνται για έλεγχο αριθμητικών δεδομένων και χαρακτηριστικών (με καθορισμένη αντιπαράβολή)

II. like(με % και \_) : ελέγχει αν μια χαρακτηριστική ικανοποιεί ένα συγκεκριμένο πρότυπο. Το σύμβολο % αναπαριστά οποιοδήποτε συνδυασμό 0 ή περισσότερων χαρακτήρων ενώ το \_ ακριβώς ένα οποιοδήποτε χαρακτήρα.

III. IS null, is not null, In, not in, between : έλεγχος αν ένα πεδίο περιέχει null τιμές, αν περιέχει μια τιμή από ένα σύνολο τιμών (τελεστής in) ή παίρνει μια τιμή από ένα εύρος τιμών ( τελεστής between).

- **GROUP BY** : Στις πλειάδες που ανακτήθηκαν από τους πίνακες που εμφανίζονται στον όρο from, μπορούμε να τις ομαδοποιήσουμε με βάση κάποια τιμή ενός πεδίου (θα πρέπει αυτό να αναφέρεται αμέσως μετά το **GROUP BY**). Στην συνέχεια υπολογίσουμε μια στατιστική τιμή (π.χ μέσος όρος, μέγιστο, ελάχιστο) ενός πεδίου για κάθε ομάδα πλειάδων. Ας δούμε μερικές διαθέσιμες συναρτήσεις :

- **SUM(column):** Άθροισμα
- **COUNT(items):** Μέτρηση Πλήθους
  - μη κενών εγγραφών
  - Με Distinct column προσμετρώνται μόνο οι διαφορετικές τιμές
  - Με count(\*) προσμετρώνται και οι Null εγγραφές.
- **MAX(column):** Μέγιστο
- **MIN(column):** Ελάχιστο
- **AVG(column):** Μέσος όρος

- **HAVING** : Ο όρος αυτός υπάρχει μόνον μετά από τον GROUP BY, και φιλτράρει επιπλέον τις πλειάδες που προέκυψαν από την συνθήκη του όρου WHERE. Όταν υπάρχει ο όρος αυτός τότε εμφανίζονται σαν αποτέλεσμα μόνο εκείνες οι εγγραφές που οι τιμές των πεδίων του ικανοποιούν την συνθήκη *where\_definition*. Συνήθως η



συνθήκη αυτή περιέχει μια ή περισσότερες από τις διαθέσιμες συναρτήσεις.

- **ORDER BY** : Ταξινομεί τις πλειάδες εξόδου με βάση την τιμή μιας στήλης που ακολουθεί. Επιπλέον προαιρετικά ακολουθεί και ο προσδιοριστής αύξουσας (ASC – προκαθορισμένος) ή φθίνουσας (DESC) ταξινόμησης.

- **LIMIT** : Ακολουθείται από έναν ή δύο ακεραίους αριθμούς. Στην πρώτη περίπτωση αν το αποτέλεσμα είναι πολλές εγγραφές εξόδου εμφανίζει μόνον τόσες όσες ο αριθμός μετά το LIMIT. Αν ακολουθούν δύο ακέραια ορίσματα τότε αφού αγνοήσουμε αρχικά τις πρώτες εγγραφές εξόδου (1<sup>ο</sup> όρισμα) στην συνέχεια στο αποτέλεσμα εμφανίζονται τόσες όσες το δεύτερο όρισμα.

- 

## 1.7 Διαγραφές πινάκων και εγγραφών

Η εντολή με την οποία διαγράφουμε ένα πίνακα είναι της μορφής :

*DROP TABLE [IF EXISTS] table\_name ;*

Η εντολή αυτή διαγράφει από την τρέχουσα ΒΔ τον πίνακα, που το όνομα του είναι στην θέση table\_name της παραπάνω εντολής. Ο προαιρετικός όρος *IF EXISTS* αποτρέπει την εμφάνιση σφάλματος στην περίπτωση που ο πίνακας δεν υπάρχει.

Η διαγραφή όλων των εγγραφών ενός πίνακα, χωρίς να διαγραφεί αυτός, γίνεται με μια από τις εντολές :

*DELETE FROM table\_name;*

*TRUNCATE TABLE table\_name;*

Έχουν το ίδιο αποτέλεσμα (την διαγραφή όλων των εγγραφών ενός πίνακα) με δύο βασικές διαφορές :

- Η 2<sup>η</sup> εκτελείτε πιο γρήγορα από την πρώτη
- Η 1<sup>η</sup> εμφανίζει αποτελέσματα εκτέλεσης της διαγραφής., σε αντίθεση με την 2<sup>η</sup> .

Αν θέλω να διαγράψω συγκεκριμένες εγγραφές από ένα πίνακα τότε θα πρέπει να χρησιμοποιήσω από τις προηγούμενες μορφές διαγραφής μόνο την πρώτη στην εξής μορφή :

*DELETE FROM table\_name where condition [ORDER BY field];  
[LIMIT number];*

Μετά τον όρο *where* τοποθετούμε την συνθήκη όπου πρέπει να πληρούν οι προς διαγραφή εγγραφές. Ο προαιρετικός όρος *LIMIT* καθορίζει πόσες εγγραφές θα διαγραφούν σε περίπτωση που είναι περισσότερες από μια. Επιπλέον ο προαιρετικός όρος *ORDER BY* καθορίζει την σειρά με την οποία θα διαγραφούν οι εγγραφές.

Παράδειγμα :

*Delete from book where ISBN="0-672-11297-2"*

## **1.8 Ενημέρωση δεδομένων πινάκων**

Αφού γίνει η καταχώρηση των στοιχείων σε ένα πίνακα με εντολές *INSERT*, αν θέλουμε να τροποποιήσουμε κάποια από αυτά τα δεδομένα θα πρέπει να χρησιμοποιήσουμε την εντολή *UPDATE* που είναι της παρακάτω μορφής :

*UPDATE όνομα\_πίνακα  
SET στήλη=τιμή,...  
[WHERE συνθήκη] ;*

Μετά τον όρο *UPDATE* προσδιορίζω το όνομα του πίνακα στον οποίο θα γίνει ενημέρωση δεδομένων. Ο όρος *SET* περιλαμβάνει μια εντολή εκχώρησης που καθορίζει πως θα ενημερωθεί μια συγκεκριμένη στήλη του πίνακα (αριστερό μέλος της εντολής εκχώρησης). Η συνθήκη μετά το *WHERE* καθορίζει τις πλειάδες που θα γίνει ενημέρωση στα δεδομένα της συγκεκριμένης στήλης.

## **1.9 Τροποποίηση σχήματος πίνακα**

Αφού έχουμε δημιουργήσει έναν πίνακα, ανεξάρτητα αν έχουμε εισάγει δεδομένα ή όχι μπορούμε με την εντολή τροποποίησης, της παρακάτω μορφής

*ALTER TABLE όνομα-πίνακα αλλαγή [,αλλαγή...];*  
Μπορούμε να κάνουμε ταυτόχρονα με την ίδια εντολή μια ή περισσότερες αλλαγές στο σχήμα του .

Οι αλλαγές είναι των παρακάτω τύπων :

- Προσθήκη στήλης σε πίνακα :

Είναι της μορφής :

*ALTER TABLE όνομα-πίνακα ADD νέα-στήλη [FIRST] [AFTER υπάρχουσα στήλη ] ;*

Μετά τον όρο ADD τοποθετούμε το όνομα της στήλης που θέλουμε να εισάγουμε στον πίνακα. Αν δεν ακολουθεί τίποτε μετά το όνομα της στήλης τότε αυτή τοποθετείται σαν τελευταία στον πίνακα. Αν αντίθετα χρησιμοποιήσουμε τον προαιρετικό όρο FIRST, τότε τοποθετείται σαν πρώτη στήλη του πίνακα. Αντιθέτως αν χρησιμοποιήσω τον προαιρετικό όρο AFTER ακολουθούμενος από ένα όνομα υπάρχουσας στήλης του πίνακα, η νέα στήλη τοποθετείται αμέσως μετά από αυτή.

- Διαγραφή στήλης από πίνακα

Είναι της μορφής :

*ALTER TABLE όνομα-πίνακα DROP στήλη ;*

Διαγράφει από έναν πίνακα μια υπάρχουσα στήλη που το όνομα της ακολουθεί τον όρο DROP .

- Αλλαγή τύπου δεδομένων μιας στήλης

Είναι της μορφής :

*ALTER TABLE όνομα-πίνακα MODIFY στήλη νέος τύπος;*

- Μετονομασία μιας στήλης ενός πίνακα

Είναι της μορφής :

*ALTER TABLE όνομα-πίνακα CHANGE στήλη νέο- όνομα;*

ενώ η μετονομασία πίνακα είναι της παρακάτω μορφής :

*ALTER TABLE όνομα-πίνακα RENAME TO νέο- όνομα;*  
*book drop price; ; Διαγραφεί την στήλη price από τον πίνακα book.*

# Κεφάλαιο 2<sup>ο</sup>

## 2. Εισαγωγή στην γλώσσα PHP

Στις αρχές της δεκαετίας του '90 το Web άρχισε να συναντά ευρεία απήχηση. Οι χρήστες χρησιμοποιώντας το πρωτόκολλο HTTP μπορούσαν να επισκεφτούν ένα δικτυακό τόπο και να δουν τα περιεχόμενα του απλώς γράφοντας την url διεύθυνση στη γραμμή διευθύνσεων ενός web browser. Η δημιουργία των περιεχομένων ενός ιστότοπου γίνεται χρησιμοποιώντας την γλώσσα HTML. Ενώ οι τελευταίες εκδόσεις της HTML προσφέρουν σαφώς περισσότερες δυνατότητες σε σύγκριση με τις αρχικές εκδόσεις, δεν παύει να είναι ένα σύστημα με περιορισμένες δυνατότητες και έλλειψη ευελιξίας, αποκριτικότητας και δυναμικής συμπεριφοράς.

Οι επισκέπτες ενός δικτυακού τόπου, δημιουργημένου αποκλειστικά με HTML, μπορούν να δουν ιστοσελίδες με κατάλληλη μορφοποίηση και σχεδίαση, που περιέχουν εικόνες, μορφοποιημένο κείμενο, ηχητικά εφέ, γραφικά και animation . Δυστυχώς όμως δεν μας παρέχουν αλληλεπίδραση με τον χρήστη. Για παράδειγμα δεν υποστηρίζουν δυνατότητα σε ένα χρήστη να γράψει και να στείλει σχόλια, ούτε να προσαρμόζονται τα περιεχόμενα τις σελίδας (χρώματα, διαμόρφωση σελίδας) ανάλογα με τις προτιμήσεις του. Για να υπάρχει αλληλεπίδραση ενός χρήστη με έναν δικτυακό τόπο, θα πρέπει να γραφεί κώδικας σε κάποια κατάλληλη γλώσσα προγραμματισμού και να ενσωματωθεί στον html κώδικα του ιστότοπου, ώστε να έχουμε την επιθυμητή λειτουργικότητα.

Υπάρχουν διάφορες γλώσσες προγραμματισμού που μας παρέχουν αυτή την επιπλέον δυνατότητα διαλογικής επικοινωνίας του χρήστη με μια ιστοσελίδα. Τέτοιες γλώσσες είναι η PHP, Java, Javascript, Perl καθώς και γλώσσες CGI (π.χ. C++) . Οι τρεις πρώτες γλώσσες είναι οι πιο ευρέως χρησιμοποιούμενες. Τα τελευταία χρόνια η PHP κερδίζει έδαφος και τείνει να επικρατήσει στον χώρο των Web εφαρμογών, αφού έχει κατασκευαστεί αποκλειστικά και μόνο για τέτοιου είδους εφαρμογές. Στα πλαίσια της παρούσας εργασίας θα χρησιμοποιήσουμε την γλώσσα PHP . Θεωρούμε δεδομένο ότι ο χρήστης έχει μια βασική γνώση της γλώσσας HTML. Επιπλέον θα πρέπει να γνωρίζει τις δομές επιλογής και επανάληψης μιας οποιασδήποτε διαδικαστικής γλώσσας προγραμματισμού.

Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού ανοικτού λογισμικού ειδικά σχεδιασμένη για δημιουργία σελίδων Web με δυναμικό περιεχόμενο. Ο κώδικας PHP ενσωματώνεται μέσα σε HTML σελίδες και εκτελείται από τον Server (που είναι αποθηκευμένες) κάθε φορά που ο χρήστης επισκέπτεται τις σελίδες αυτές. Η εκτέλεση ενός κώδικα PHP έχει ως αποτέλεσμα τη παραγωγή κώδικα HTML ο οποίος στέλνεται τελικά στο φυλλομετρητή του χρήστη. Η PHP είναι μια γλώσσα σεναρίου (script), αφού ανάλογα με κάποια επιλογή του χρήστη (πάτημα κουμπιού, επιλογή από λίστα) εκτελείται το τμήμα προγράμματος που είναι συνδεδεμένο με αυτή την ενέργεια (σενάριο) . Τα προγράμματα , γραμμένα σε PHP, εκτελούνται στον διακομιστή (server-side γλώσσα) και όχι στον υπολογιστή του χρήστη (όπως π.χ. η Javascript).

Η ευρεία διάδοση της γλώσσας είναι απόρροια των παρακάτω πλεονεκτημάτων :

- **Υψηλή απόδοση** : Συγκριτικά tests (πηγή <http://www.zend.com>) έδειξαν ότι εφαρμογές php, μπορούν να εξυπηρετήσουν πιο αποτελεσματικά τον ίδιο αριθμό χρηστών ενός ιστότοπου, σε σχέση με ανταγωνίστριες γλώσσες (π.χ. Java, Javascript).

- **Διασυνδέσεις με πολλά διαφορετικά ΣΔΒΔ** : Εκτός από την Mysql υποστηρίζει επικοινωνία με όλα σχεδόν τα εμπορικά περιβάλλοντα ΒΔ. Έτσι γίνεται εφικτή η εμφάνιση δεδομένων από πολλά περιβάλλοντα στο διαδίκτυο.

- **Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες** : Η τελευταίες εκδόσεις της γλώσσας περιέχουν έτοιμες συναρτήσεις με ιδιαίτερα χρήσιμες λειτουργίες όπως ηλεκτρονικού ταχυδρομείου, υπηρεσίες δικτύων, δημιουργία pdf εγγράφων, που διευκολύνουν αφάνταστα το έργο των προγραμματιστών εφαρμογών.

- **Χαμηλό κόστος** : Παρέχεται δωρεάν το πακέτο Wamp (περιέχει php,mysql, apache server) για την ανάπτυξη Web εφαρμογών.

- **Ευκολία μάθησης και χρήσης** : Για ένα γνώστη της C, Java είναι εύκολη η εκμάθηση της PHP, αφού οι βασικές δομές και ο τρόπος ανάπτυξης εφαρμογών έχουν αρκετά κοινά.

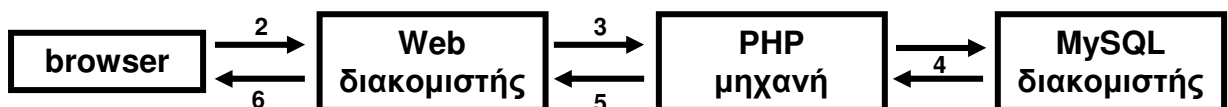
- **Μεταφερσιμότητα** : Ένα πρόγραμμα σε Php μπορεί να παραμετροποιηθεί εύκολα, έτσι ώστε να ανιχνεύει τις μεταβλητές λειτουργικού περιβάλλοντος και να εκτελείται χωρίς τροποποιήσεις σχεδόν σε όλα τα λειτουργικά συστήματα.

- **Διαθεσιμότητα του πηγαίου κώδικα** : Όλοι μπορούμε να έχουμε πρόσβαση στον κώδικα προέλευσης ( σε αντίθεση με προϊόντα πατέντας – κλειστού κώδικα) και να κάνουμε επεκτάσεις διαθέσιμες σε

όλους του φίλους της γλώσσας. Δεν είναι απαραίτητο μόνο ο κατασκευαστής (<http://www.zend.com>) να κάνει διορθώσεις, που σημαίνει ότι το προϊόν αυτό θα συνεχίσει να υπάρχει και να βελτιώνεται ακόμα και αν ο κατασκευαστής πάψει να το υποστηρίζει.

## 2.1 Λειτουργία της PHP:

1. Η PHP τρέχει σε ένα διακομιστή.
2. Ο **Web browser** ενός χρήστη κάνει μία `http://` αίτηση για μια συγκεκριμένη php Web σελίδα. (πχ. `http://www.example.com/info.php`)
3. Ο **Web διακομιστής** λαμβάνει την αίτηση για την σελίδα php, ανακαλεί το αρχείο και το περνά στην μηχανή php για επεξεργασία.
4. Η **μηχανή php** αρχίζει την ανάλυση του php κώδικα επικοινωνώντας αν υπάρχει ανάγκη με τον **διακομιστή της βάση δεδομένων**.
5. Η **μηχανή php** σταματά την εκτέλεση του προγράμματος επιστρέφοντας στον Web διακομιστή την τελική HTML σελίδα.
6. Ο **Web διακομιστής** περνά την html σελίδα ξανά στο **Web browser** όπου ο χρήστης μπορεί να διαβάσει το αποτέλεσμα της php σελίδας.



## 2.2 Συντακτικό της PHP

Για ν' αναγνωρίσει η PHP ότι ένα κομμάτι κώδικα μέσα σε μία σελίδα HTML είναι PHP πρέπει να τον περικλείετε ανάμεσα σε εισαγωγικά της μορφής `<?php ... ?>` ή της μορφής `<? ... ?>`.

```
<html>
<body>

  <?php
  echo 'Hello, World!';
  ?>

</body>
</html>
```

```
<html>
<body>
  <strong>
    <?php
    echo "Hello," . " <i>" . " World!" . "</i>";
    ?>
  </strong>
</body>
</html>
```

Η συνάρτηση **echo** τυπώνει στον φυλλομετρητή μια χαρακτηρισ σειρά που μπορεί να έχει HTML μορφοποίηση. Με την επιλογή του φυλλομετρητή View Source για να δω το κώδικα προέλευσης των ιστοσελίδων που παράγονται από τα παραπάνω αρχεία πηγαίου κώδικα php, παίρνω αντίστοιχα :

```
<html>
<body>

  Hello, World!

</body>
</html>
```

και

```
<html>
<body>

  <strong><i> Hello, World! </i></strong>

</body>
</html>
```

Αυτό συμβαίνει επειδή ο κώδικας που υπάρχει στο αρχείο php, εκτελείται τοπικά από το περιβάλλον php που υπάρχει στον διακομιστή

και επομένως ο χρήστης βλέπει μόνο τον αντίστοιχο html κώδικα που προκύπτει από το αντίστοιχο script σε php και στέλνεται στον φυλλομετρητή του χρήστη. Αντίθετα σε γλώσσες δημιουργίας ιστοσελίδων , που εκτελούνται τοπικά στον υπολογιστή του χρήστη (client side), όπως η Javascript , με την επιλογή View Source του φυλλομετρητή μπορούμε να δούμε τις εντολές δημιουργίας της ιστοσελίδας, με χρήση της συγκεκριμένης γλώσσας . Όταν ένας χρήστης επιλέγει ιστοσελίδα, που έχει ενσωματωμένο κάποιο από τα παραπάνω script, εμφανίζονται στον φυλλομετρητή του τα παρακάτω αντίστοιχα αποτελέσματα :

Hello, World!

**Hello, World!**

## 2.3 Μεταβλητές, πίνακες και αλφαριθμητικά

Όπως σε όλες τις γλώσσες προγραμματισμού, έτσι και στην php, μπορούμε να χρησιμοποιήσουμε μεταβλητές για να αποθηκεύσουμε μια τιμή ενός δεδομένου κάποιου τύπου. Για την ονοματολογία μεταβλητών ισχύουν κανόνες που είναι κοινοί στις περισσότερες γλώσσες προγραμματισμού :

- Χρησιμοποιούμε συνδυασμούς κεφαλαίων και πεζών γραμμάτων του αγγλικού αλφαβήτου, τα ψηφία 0 – 9 καθώς και τον χαρακτήρα \_ .
- Το όνομα μιας μεταβλητής δεν μπορεί να αρχίζει με ψηφίο.
- Επιπλέον πάντα το όνομα μιας μεταβλητής ξεκινά πάντα με τον χαρακτήρα \$.
- Δεν είναι ευαίσθητα στην διάκριση πεζών κεφαλαίων γραμμάτων.

Ας δούμε ένα απλό παράδειγμα :

Ανοίγουμε το πρόγραμμα *Σημειωματάριο* , γράφουμε σε ένα αρχείο με όνομα test1.php τον παρακάτω κώδικα και αποθηκεύω το αρχείο αυτό στον φάκελο www του apache server που υπάρχει στο πακέτο Wamp :

- |   |
|---|
| <ol style="list-style-type: none"><li>1. &lt;html&gt;</li><li>2. &lt;body&gt;</li></ol> |
|---|



```

3. <?php
4. $var_name_number = 143;
5. $var_name_txt = "Hello World ";
6. $var_name_array = array("Peter","Quagmire","Joe");
7. $new_var = $var_name_number+8;
8. echo "Day: " . $new_var . "<br>";
9. echo $var_name_txt . " " . $var_name_array[1];
10. ?>
11. </body>
12. </html>

```

*Γραμμές 3 – 10 :* Εδώ περιέχεται ο php κώδικας (περικλείεται σε <?php και ?>) της ιστοσελίδας. Είναι ένας απλός κώδικας που εμφανίζει το περιεχόμενο κάποιων μεταβλητών.

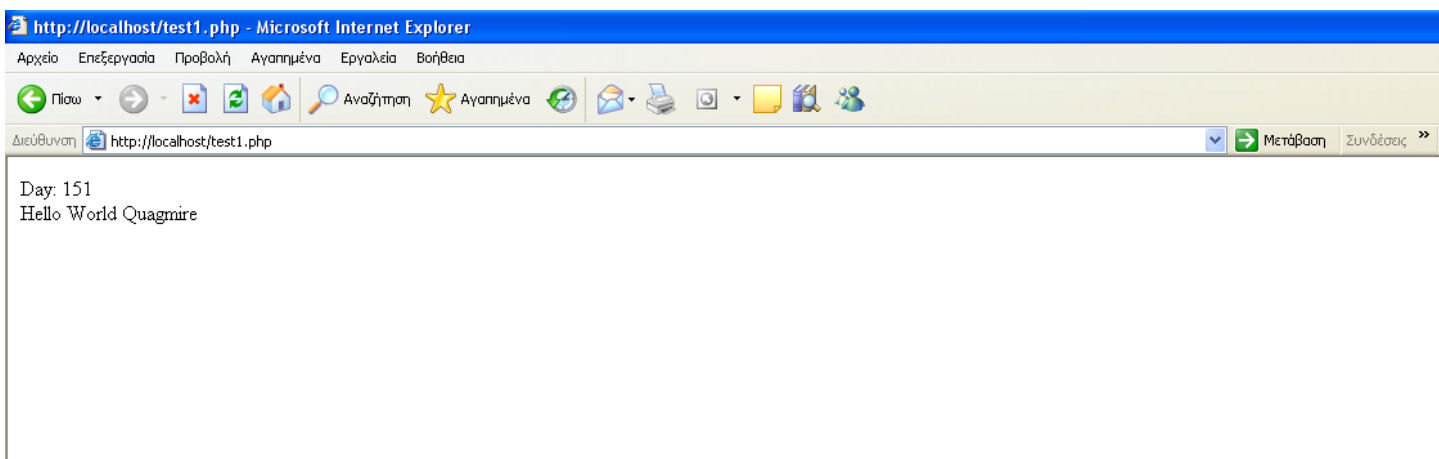
*Γραμμές 4 – 5 :* Στην php δεν δηλώνουμε μεταβλητές, απλώς μετά από μια εντολή εκχώρησης ο τύπος δεδομένων της σταθεράς ή της παράστασης που εκχωρείται στην μεταβλητή θα είναι και ο τύπος δεδομένων της μεταβλητής. Κατά την διάρκεια εκτέλεσης του προγράμματος ο τύπος δεδομένων μιας μεταβλητής μπορεί να αλλάξει δυναμικά, αφού μπορεί να τις εκχωρηθεί σταθερά διαφορετικού τύπου. Έτσι η μεταβλητή \$var\_name\_number είναι τύπου ακεραίου και η \$var\_name\_txt χαρακτηριστικής.

*Γραμμή 6 :* Η μεταβλητή \$var\_name\_array είναι ένας πίνακας (λόγω του προσδιοριστή array) χαρακτηριστικών τριών στοιχείων. Οι δεικτοδότηση των στοιχείων του πίνακα ξεκινάει από το 0.

*Γραμμή 7 :* Η μεταβλητή \$new\_var λόγω της παράστασης που τις εκχωρείται είναι ακεραίου τύπου και παίρνει την τιμή 151.

*Γραμμές 8 – 9 :* Η συνάρτηση echo δέχεται σαν όρισμα μια χαρακτηριστική και την τυπώνει στον φυλλομετρητή. Για να συνενώσουμε δύο χαρακτηριστικές ή μια χαρακτηριστική και μια αριθμητική τιμή χρησιμοποιούμε τον τελεστή τελεία ( . ).

Για να δούμε τα αποτελέσματα εκτέλεσης του παραπάνω προγράμματος γράφουμε στην γραμμή url διεύθυνσεων <http://localhost/test1.php>



## 2.4 Δομές επιλογής και επανάληψης

Η PHP υποστηρίζει την δομή επιλογής if , καθώς και τις δομές επανάληψης for και while , με τον ίδιο τρόπο όπως σχεδόν όλες οι άλλες γλώσσες προγραμματισμού. Παραθέτουμε ένα απλό παράδειγμα που συνδυάζει τις παραπάνω δομές ελέγχου :

```
1. <html>
2. <body>
3. <?php
4. $k = 143;
5. if (fmod($k,2)==0)
6.     echo $k." is even".<br>;
7. else
8.     echo $k." is odd".<br>;
9. $i=0;
10.while($i<=5)
11.{
12.    echo "Number: ".$i.<br>;
13.    $i++;
14.}
15.for ($i=1; $i<=5; $i++)
16.{
17.    echo "Hello World!<br>;
18.}
19.??>
20.</body>
21.</html>
```

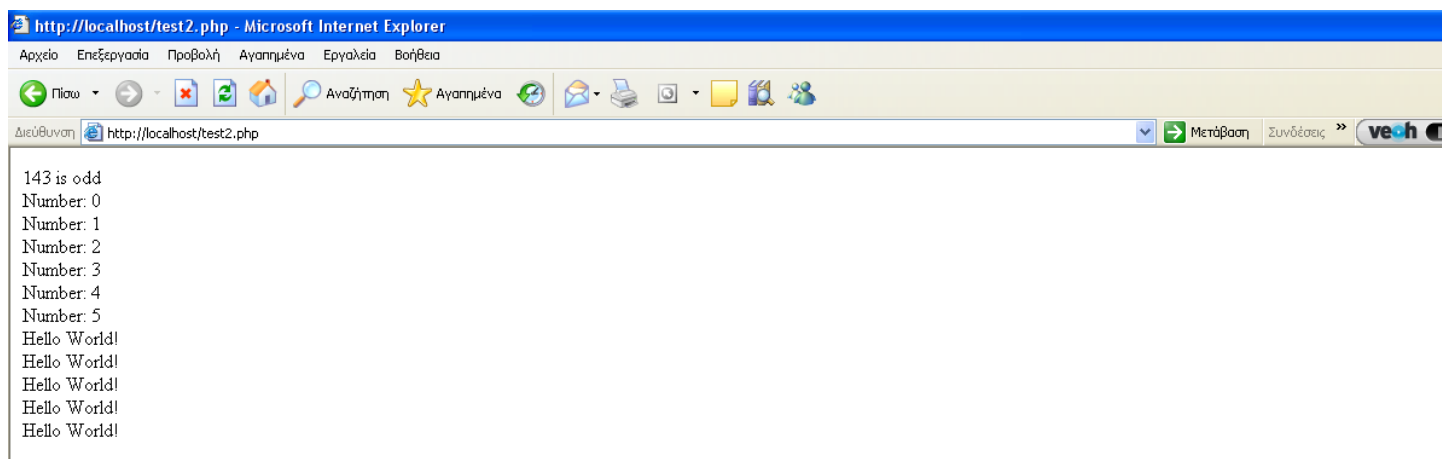
*Γραμμές 4 – 8 :* Ελέγχει αν η τιμή της μεταβλητής \$k είναι άρτια ή περιττή , χρησιμοποιώντας την συνάρτηση υπολογισμού υπολοίπου διαίρεσης mod. Στη συνέχεια τυπώνει κατάλληλο διαγνωστικό μήνυμα.

*Γραμμές 9 – 14 :* Χρησιμοποιώντας την δομή ελέγχου while , παίρνω τους αριθμούς από 0 έως 5 και τυπώνω έξι γραμμές . Η κάθε γραμμή θα περιέχει το κείμενο Number , ακολουθούμενο κάθε φορά από έναν από αυτούς τους αριθμούς.

*Γραμμές 15 – 18 :* Χρησιμοποιώντας την δομή επανάληψης for τυπώνω πέντε φορές στην οθόνη την φράση "Hello World!". Ο HTML χαρακτήρας ελέγχου για κενή γραμμή είναι ο <br>.

Αφού τοποθετήσουμε τον παραπάνω κώδικα στο αρχείο test2.php και το αποθηκεύσουμε στον φάκελο www του server, μπορούμε να δούμε τα αποτελέσματα εκτέλεσης του, απλώς γράφοντας στην γραμμή διευθύνσεων <http://localhost/test2.php>

Χρησιμοποιούμε τον προσδιοριστή localhost για να προσπελάσουμε το περιεχόμενο ενός δικτυακού τόπου, όταν ο server του δικτυακού τόπου και ο browser ενός χρήστη μοιράζονται από κοινού τον ίδιο υπολογιστή.



Αρχικά η δομή επιλογής if έχει σαν αποτέλεσμα να τυπωθεί η 1<sup>η</sup> γραμμή. Ακολούθως η δομή while εμφανίζει τις επόμενες 6 γραμμές. Τέλος η δομή for τυπώνει 5 φορές την φράση Hello World! .

## 2.5 Πίνακες

Είδαμε νωρίτερα ένα παράδειγμα ορισμού πίνακα, με όνομα \$var\_name\_array, τριών στοιχείων , τύπου δεδομένων χαρακτηριστικής :

```
$var_name_array = array("Peter", "Quagmire", "Joe");
```

Η αρίθμηση των δεικτών των στοιχείων ξεκινά από 0. Έτσι για να πάρω τις τιμές των δύο πρώτων στοιχείων του πίνακα \$var\_name\_array, έχω

```
echo $var_name_array[0]. " " . $var_name_array[1];
```

Το μέγεθος των πινάκων μεταβάλλεται δυναμικά, γεγονός που σημαίνει ότι μπορώ να εισάγω ή να διαγράψω στοιχεία κατά τη διάρκεια

εκτέλεσης του προγράμματος. Για παράδειγμα για να εισάγω ένα ακόμα στοιχείο στον πίνακα `$var_name_array` γράφω :

```
$var_name_array[3]="Nick";
```

Ενώ διαγράφω το στοιχείο που μόλις εισήγαγα καθώς κι οποιαδήποτε άλλη μεταβλητή ενός προγράμματος :

```
unset($var_name_array[3]);
```

Τέλος μπορώ να εκκενώσω ένα πίνακα από τα περιεχόμενα του, χωρίς όμως να τον διαγράψω με την παρακάτω εντολή :

```
$var_name_array=array();
```

Πολλές φορές είναι βολικό να αριθμούμε τα στοιχεία ενός πίνακα ξεκινώντας όχι από το 0, αλλά από άλλον αριθμό π.χ. το 1. Αυτό καθορίζεται κατά τον ορισμό του πίνακα. Για παράδειγμα για να ξεκινήσω την αρίθμηση των στοιχείων του πίνακα `$var_name_array` από το 1, τον δηλώνω ως εξής :

```
$var_name_array = array(1=>"Peter" , "Quagmire",  
"Joe");
```

Η `php` εκτός από αριθμητικούς δείκτες , επιτρέπει τα δεδομένα ενός πίνακα να αντιστοιχίζονται σε δείκτες από άλλους τύπους δεδομένων. Αυτό είναι χρήσιμο και βολικό σε εφαρμογές όπου έχουμε συσχετιζόμενες λίστες στοιχείων. Για παράδειγμα αν θέλω να υλοποιήσω μια εφαρμογή τιμολογίου μπορώ να ορίσω έναν πίνακα που να περιέχει αντιστοιχίες προϊόντος – τιμής ως εξής :

```
$prices=array('Tires'=>100, 'Oil'=>10);
```

Ο πίνακας `prices` περιέχει σαν δεδομένα αριθμητικές τιμές, που αντί για αριθμητικούς δείκτες έχουν δείκτες χαρακτηρισειράς (που αναπαριστούν τα προϊόντα). Έχει επικρατήσει στην βιβλιογραφία σε κάθε τέτοιο ζευγάρι δεδομένων, ο πρώτος όρος να λέγεται κλειδί (key) και ο δεύτερος τιμή (value). Για να προσπελάσουμε ταυτόχρονα τα κλειδιά και τις τιμές ενός πίνακα χρησιμοποιούμε την παρακάτω δομή `foreach` :

```
foreach($prices as $key=>$value)  
echo $key."----".$value;
```

Η δομή επανάληψης foreach, είναι μια ειδική μορφή for για προσπέλαση στοιχείων πινάκων . Ξεκινά από το πρώτο ζεύγος κλειδιού - τιμής ενός πίνακα και προσπελαύνει τα περιεχόμενα τους με την χρήση των βοηθητικών μεταβλητών \$key και \$value. Ο παραπάνω κώδικας τυπώνει τις αντιστοιχίες του πίνακα \$prices :

```
Tires----100  
Oil----10
```

Ο βρόγχος foreach ολοκληρώνεται μόλις εξεταστεί και το τελευταίο ζεύγος κλειδιού - τιμής ενός πίνακα.

Μπορώ να προσπελάσω ή να τροποποιήσω εύκολα το περιεχόμενο μιας τιμής ενός πίνακα, αρκεί να χρησιμοποιήσω το όνομα ενός πίνακα και σαν δείκτη την τιμή του κλειδιού. Στο παράδειγμα μας αν θέλω να κάνω ανατίμηση στην τιμή των λάστιχων αυτοκινήτου , χρησιμοποιώ την εντολή :

```
Prices['Tires']=105;
```

## 2.6 Διασύνδεση της PHP με την Mysql

Ένας από τους λόγους που συντελούν στην ευρεία διάδοση της php είναι η δυνατότητα διασύνδεσης με το επίσης δημοφιλή ανοιχτού κώδικα ΣΔΒΔ Mysql. Έτσι είναι εφικτή η δημοσίευση δεδομένων και πληροφοριών μιας βάσης στο Web καθώς και η ενημέρωση της μέσω Web. Θα εξετάσουμε βηματικά πως γίνεται η διασύνδεση της php με την mysql χρησιμοποιώντας ένα απλό παράδειγμα . Δημιουργούμε μια απλή φόρμα εισαγωγής στοιχείων ενός προσώπου. Θέλουμε σε τρία πεδία κειμένου να εισάγουμε το όνομα, επώνυμο και ηλικία αντίστοιχα. Στην συνέχεια ένα σενάριο php θα λαμβάνει αυτά τα δεδομένα και θα δημιουργεί μια νέα εγγραφή στον πίνακα

### PIN

<u>id</u>	fname	lname	age
-----------	-------	-------	-----

που θα περιέχεται σε μια βάση δεδομένων. Πρωτεύον κλειδί θα είναι το πεδίο id , που θα παίρνει μια ακέραια τιμή αύξουσας αρίθμησης όταν προσθέτουμε μια νέα εγγραφή στον πίνακα. Τα επόμενα τρία πεδία του πίνακα, θα αρχικοποιούνται με τις τιμές που θα εισάγει ο χρήστης στα αντίστοιχα πεδία κειμένου της html φόρμας.

```

1. <html>
2. <head>
3. <title>TESTING</title>
4. </head>
5. <body>
6. <br>
7. <form method="post" action="test3.php">
8. <table cellspacing=5 align="center">
9. <tr><th align="left"> FIRST NAME </th>
10.<td><input type="text" name="fname" maxlength="20"
size="20"></td></tr>
11.<tr><th align="left"> LAST NAME</th>
12.<td><input type="text" name="lname" maxlength="20"
size="20"></td></tr>
13.<tr><th align="left"> AGE</th>
14.<td><input type="text" name="age" maxlength="20"
size="20"></td></tr>
15.<tr>
16.<tr><td align="left"> <input type="submit"
value="ΑΠΟΣΤΟΛΗ"> </td>
17.<td align="right"> <input type="reset"
value="ΔΙΑΓΡΑΦΗ"></td></tr>
18.</table>
19.</form>
20.</body>
21.</html>

```

Αρχείο test3.php

Αν πληκτρολογήσουμε <http://localhost/test3.php> στην γραμμή διευθύνσεων του φυλλομετρητή, θα εμφανιστεί σε αυτόν :

The screenshot shows a web browser window titled "TESTING - Microsoft Internet Explorer". The address bar contains "http://localhost/test3.html". The page content displays a form with three text input fields labeled "FIRST NAME", "LAST NAME", and "AGE". Below these fields are two buttons: "ΑΠΟΣΤΟΛΗ" (Submit) and "ΔΙΑΓΡΑΦΗ" (Reset).

*Γραμμή 7.* Η μεταβλητή της φόρμας *action* περιέχει το πρόγραμμα *php* (*test3.php*) που εκτελείται αυτόματα μόλις ο χρήστης πατήσει το κουμπί (τύπου *Submit*) με επιγραφή *ΑΠΟΣΤΟΛΗ*. Το πρόγραμμα αυτό συνήθως επεξεργάζεται τα δεδομένα των συστατικών της φόρμας που το εμπεριέχει. : Η μεταβλητή *method* της φόρμας μπορεί να πάρει δύο δυνατές τιμές : *get* και *post*. Στην πρώτη περίπτωση τα δεδομένα που εισήγαγε ο χρήστης στα στοιχεία της φόρμας (στο παράδειγμα μας τρία πεδία κειμένου) στέλνονται στο πρόγραμμα διαχείρισης ( παράμετρος της *action*) και παράλληλα εμφανίζονται στην γραμμή διευθύνσεων του φυλλομετρητή με ειδική κωδικοποίηση. Έτσι είναι εκτεθειμένα σε αδιάκριτα βλέμματα καθώς και στο ιστορικό διευθύνσεων του φυλλομετρητή . Αυτήν η όχι επιθυμητή ιδιότητα δεν υπάρχει στην μέθοδο *post*, που χρησιμοποιούμε στο παράδειγμα μας.

Τα στοιχεία της φόρμας (επιγραφές, κουμπιά και πεδία κειμένου) τα έχουμε τοποθετήσει κατάλληλα σαν στοιχεία ενός πίνακα 4x2 .

*Γραμμή 10,12,15 :* Ιδιαίτερη προσοχή θα πρέπει να δοθεί να δοθεί στον προσδιοριστή *name* καθενός από τα τρία πεδία κειμένου. Το πρόγραμμα *test3.php* αναφέρεται σε καθένα από αυτά τα στοιχεία, προκειμένου να προσπελάσει την τιμή που έχει εισάγει σε αυτά ο χρήστης, με βάση την τιμή που εκχωρήσαμε στην αντίστοιχη μεταβλητή *name*, όπως θα δούμε παρακάτω.

```

1. <html>
2. <body>

3. <?php

4. $con = mysql_connect("localhost","root","");
5. if (!$con)
6.     die('Could not connect: ' . mysql_error());
7. if ( @mysql_query ("CREATE DATABASE IF NOT EXISTS
    my_db",$con ) )

8.     echo "Database created";
9. else
10.     echo "Error creating database: ".mysql_error();
11.if(!@mysql_select_db('my_db'))
12.die('could not select the database because'.mysql_error());
13.$query='CREATE TABLE IF NOT EXISTS pin (
14.id INT UNSIGNED NOT NULL AUTO_INCREMENT
    PRIMARY KEY,
15.
```

```

16.fname VARCHAR(20) NOT NULL,
17.lname VARCHAR(20) NOT NULL,
18.age INT NOT NULL
19.);
20.if(!@mysql_query($query))
21.die('could not create the table because'.mysql_error());
22.$query1="INSERT INTO pin(id, fname, lname, age) VALUES (
    0, '{$_POST['fname']}', '{ $_POST['lname'] }', '{$_POST['age']}'
    )";
23.if(!@mysql_query($query1))
24.die('could not insert into table because'.mysql_error());
25.??>

26.</body>
27.</html>

```

Ας εξετάσουμε τον κώδικα του παραπάνω παραδείγματος και ας δούμε την γενική μορφή των εντολών διασύνδεσης της php με την mysql.

### 2.6.1 Σύνδεση ενός προγράμματος PHP με τον MYSQL SERVER

```
mysql_connect(servername,username,password);
```

Αυτό επιτυγχάνεται στο πρόγραμμα μας με την *Γραμμή4* όπου

*Servername* : localhost (ο κοινός υπολογιστής που 'στεγάζει' τον server του php σεναρίου, καθώς και το ΣΔΒΔ της Mysql).

*Username* : Επιλέξαμε τον χρήστη ρίζα (root), αφού χρειαζόμαστε δικαιώματα δημιουργίας ΒΔ και πινάκων.

*Password* : Αρχικά δεν έχουμε εκχωρήσει κάποιο password στον χρήστη ρίζα.

Η μεταβλητή \$con, επειδή δεν χρησιμοποιείται για αποθήκευση τιμής αλλά για να υποδηλώσει επίτευξη σύνδεσης του προγράμματος μας με τον Mysql server θεωρείται τύπου *πόρος (resource)*.

είναι κενή , ικανοποιείται η συνθήκη (!\$con) και τυπώνεται κατάλληλο διαγνωστικό μήνυμα. Διαφορετικά δεν τυπώνεται τίποτε και το πρόγραμμα συνεχίζει στην επόμενη εντολή. *Γραμμές 5 – 6* : Αν για κάποιο λόγο δεν επιτεύχθηκε σύνδεση τότε η μεταβλητή \$con

*Γραμμές 7 – 10* : Χρησιμοποιούμε για πρώτη φορά την συνάρτηση *mysql\_query*, που παίρνει σαν όρισμα ένα ερώτημα χαρακτηρισειράς και το μεταβιβάζει προς εκτέλεση στον server της Mysql. Εδώ παίρνει



σαν είσοδο το ερώτημα (query) δημιουργίας μιας ΒΔ για τις ανάγκες του παραδείγματος μας. Η βάση αυτή θα λέγεται *my\_db*. Επιπλέον σε ερωτήματα δημιουργίας ΒΔ, η συνάρτηση θα δέχεται ένα δεύτερο όρισμα, τον πόρο σύνδεσης με την Mysql. Σε περίπτωση επιτυχούς ή μη εκτέλεσης της *mysql\_query* (εξαρτάται από την σύνταξη του ερωτήματος – ορίσματος) τυπώνεται κατάλληλο διαγνωστικό μήνυμα. Να αναφέρουμε την συνάρτηση *mysql\_error* , που σε περίπτωση μη επιτυχούς εκτέλεσης της *mysql\_query* , επιστρέφει το κείμενο αιτιολόγησης αποτυχίας (failure message).

*Γραμμές 11 – 12* : Χρησιμοποιούμε την συνάρτηση *mysql\_select\_db* , που το όρισμα ΒΔ, το καθιστά τρέχουσα ΒΔ.

*Γραμμές 13 – 19* : εκχωρώ στην μεταβλητή *query* , την χαρακτηρισειρά που περιέχει το Mysql ερώτημα δημιουργίας του πίνακα του παραδείγματος μας.

*Γραμμές 20 – 21* : Εκτέλεση του ερωτήματος δημιουργίας πίνακα.

Σε ένα πρόγραμμα PHP ορίζεται αυτόματα ο πίνακας *\$\_POST*, που περιέχει τις τιμές δεδομένων που τυχόν εστάλησαν στο πρόγραμμα αυτό από κάποιο άλλο, χρησιμοποιώντας την μέθοδο *post*. Για προσπέλαση δεδομένων, που εστάλησαν, με την μέθοδο *get*, χρησιμοποιείται με τον ίδιο τρόπο ο πίνακας *\$\_GET*. Στην φόρμα του αρχείου *test3.html* , υπάρχουν τρία πεδία κειμένου για καταχώρηση ονόματος, επωνύμου και ηλικίας. Τα δεδομένα που καταχωρεί ο χρήστης αποστέλλονται με την μέθοδο *post*, στο πρόγραμμα *test3.php* . Το πρόγραμμα αυτό μπορεί να προσπελάσει τις τιμές που εισήγαγε ο χρήστης σε αυτά τα πεδία κειμένου, απλώς παίρνοντας τις τιμές του πίνακα *\$\_POST*, χρησιμοποιώντας σαν δείκτη προσπέλασης την τιμή της μεταβλητής *name*, του κάθε πεδίου κειμένου, κατά τον ορισμό φόρμας στο *test3.html* . Έτσι οι τιμές που εισήγαγε ο χρήστης στα πεδία κειμένου της φόρμας του αρχείου *test3.html*, είναι *\$\_POST['fname']*, *\$\_POST['lname']* και *\$\_POST['age']*.

*Γραμμές 22 – 24* : Ορισμός μεταβλητής *\$query1*, που περιέχει την χαρακτηρισειρά ερωτήματος εισαγωγής των δεδομένων της φόρμας στα αντίστοιχα πεδία του πίνακα *pin*. Εκτέλεση του ερωτήματος εισαγωγής.



## Κεφάλαιο 3<sup>ο</sup>

### 3. Δημιουργία Βάσης Δεδομένων Εφαρμογής

**Καθορισμός απαιτήσεων προβλήματος :** Θέλουμε να κατασκευάσουμε δικτυακό τόπο όπου ένας χρήστης θα πρέπει να έχει την δυνατότητα να ενημερώνεται για τα διαθέσιμα δρομολόγια ενός σταθμού υπεραστικών δρομολογίων. Επιπλέον θα πρέπει να του παρέχεται η δυνατότητα να κάνει μέσω του ιστότοπου ηλεκτρονική κράτηση εισιτηρίων για συγκεκριμένους προορισμούς. Στην ηλεκτρονική κράτηση θα πρέπει να επιλέγει την πόλη αφετηρίας και την πόλη προορισμού, την επιθυμητή ημερομηνία ταξιδιού καθώς και τον αριθμό θέσεων. Σε περίπτωση διαθεσιμότητας της επιλεγμένης γραμμής το σύστημα θα καταχωρεί επιτυχώς τα στοιχεία καταχώρησης κράτησης του επιβάτη. Υπάρχει ο περιορισμός ότι ηλεκτρονικά μπορούν να γίνουν το πολύ σαράντα(40) κρατήσεις θέσεων για ένα δρομολόγιο και ότι ένας επιβάτης μπορεί να κάνει κράτηση για 1 – 4 θέσεις. Οι πληροφορίες σχετικά με τα διαθέσιμα δρομολόγια καθώς και τις πληροφορίες κράτησης επιβατών θα πρέπει να καταχωρούνται σε μια βάση δεδομένων . Ο ζητούμενος δικτυακός τόπος θα είναι η διαπροσωπία προσπέλασης της βάσης για αναζήτηση πληροφοριών δρομολογίων καθώς και καταχώρηση κρατήσεων δρομολογίων.

Στα πλαίσια της παρούσας εργασίας μελετώντας διάφορους δικτυακούς τόπους σταθμών υπεραστικών λεωφορείων επιλέξαμε σαν υπόδειγμα τον ιστότοπο των ΚΤΕΛ Πατρών, λόγω της ευχρηστίας του και της λειτουργικότητας του, αφού παρέχει δυνατότητα για ενημέρωση δρομολογίων καθώς και ηλεκτρονικής κράτησης. Η τελευταία δυνατότητα εντυπωσιάζει με την πληρότητα λειτουργίας της καθώς και την ευκολία χρήσης της. Τα ΚΤΕΛ Πατρών παρέχουν αυτή την δυνατότητα , που σπανίζει από τους περισσότερους ιστότοπους ΚΤΕΛ . Στα πλαίσια της παρούσας εργασίας χρησιμοποιήσαμε σαν υπόδειγμα τον ιστότοπο του υπεραστικού ΚΤΕΛ Πατρών για τους παραπάνω λόγους. Τα δεδομένα στην βάση καθώς και στον ιστότοπο επιλέξαμε να είναι στην Αγγλική γλώσσα αφού δεν υπάρχει αντίστοιχη υλοποίηση για τον ιστότοπο του συγκεκριμένου ΚΤΕΛ.

Η υλοποίηση της ΒΔ θα γίνει σε Mysql. Τα δεδομένα, που θα πρέπει να καταχωρηθούν στη ΒΔ, με βάση τις απαιτήσεις, είναι τα στοιχεία δρομολογίων καθώς και τα αντίστοιχα στοιχεία των πελατών , που κάνουν ηλεκτρονική κράτηση σε κάποια από τα παραπάνω

δρομολόγια. Για τις πληροφορίες σχετικά με τα δρομολόγια από και προς Πάτρα , συμβουλευμένοι τον πίνακα δρομολογίων από τον ιστότοπο, καθώς και την έντυπη μορφή του που παρέχετε από τον ΚΤΕΛ , σχεδιάσαμε τον παρακάτω πίνακα για την αποθήκευση τους :

### **timetable**

<u>source</u>	<u>dep_time</u>	days	<u>dest</u>	remarks
---------------	-----------------	------	-------------	---------

και κάθε καταχώρηση στον πίνακα θα αναπαριστά ένα δρομολόγιο από μια πόλη σε μια άλλη για συγκεκριμένη ώρα. Με αυτό το σκεπτικό έχουμε επιλέξει σαν σύνθετο πρωτεύον κλειδί του πίνακα τα πεδία (source , dep\_time, dest) και έχει τα παρακάτω πεδία :

- source : περιέχει την πόλη αφετηρίας κάθε δρομολογίου. Τα δεδομένα του πεδίου αυτού θα είναι τύπου χαρακτήρα μήκους το πολύ 17 .
- dep\_time : η ώρα αναχώρησης του δρομολογίου
- days : Οι ημέρες που το δρομολόγιο της καταχώρησης ( για καθορισμένες πόλεις άφιξης – προορισμού καθώς και ώρας αναχώρησης ) είναι διαθέσιμο. Θα είναι τύπου δεδομένων συνόλου με περιεχόμενα τις ημέρες της εβδομάδας στα αγγλικά.
- dest: : περιέχει την πόλη προορισμού του δρομολογίου. Τα δεδομένα του πεδίου αυτού θα είναι τύπου χαρακτήρα μήκους το πολύ 17
- remarks : Επιτρέπεται να έχει NULL τιμές. Θα περιέχει τυχόν παρατηρήσεις για το δρομολόγιο (π.χ. αν είναι express από ή προς Αθήνα, αν σταματάει να ισχύει για συγκεκριμένη χρονική περίοδο κ.α.) .Τύπος δεδομένων σύντομο κείμενο.

Ο πίνακας δημιουργείται με την παρακάτω εντολή mysql :

```
CREATE TABLE IF NOT EXISTS timetable
```

```
(  
  source char(17) NOT NULL,  
  dep_time TIME NOT NULL,  
  days
```

```
SET('Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday')  
  NOT NULL,  
  dest CHAR(17) NOT NULL,  
  remarks TINYTEXT NULL,  
  PRIMARY KEY(source,dep_time,dest)
```

);

Για την καταχώρηση δεδομένων σχετικά με τις κρατήσεις δρομολογίων αρχικά σκεφτήκαμε να δημιουργήσουμε ένα πίνακα που σε κάθε καταχώρηση του θα περιέχει τα στοιχεία του κάθε επιβάτη μαζί με το δρομολόγιο που επέλεξε και θα είναι της παρακάτω μορφής :

ΠΡΟΤΗΡΗΣΗ	ΠΟΛΗ ΠΡΟΟΡΙΣΜΟΥ	ΗΜΕΡΟΜΗΝΙΑ ΤΑΞΙΔΙΟΥ	ΩΡΑ ΤΑΞΙΔΙΟΥ	ΟΝΟΜΑ ΕΠΙΒΑΤΗ	ΕΠΩΝΥΜΟ ΕΠΙΒΑΤΗ	ΑΡΙΘΜΟ ΘΕΣΕΩΝ	ΤΥΧΟΝ ΕΚΠΙΣΤΕΥΣΗ
-----------	-----------------	---------------------	--------------	---------------	-----------------	---------------	------------------

ΣΤΟΙΧΕΙΑ ΕΠΙΛΕΓΜΕΝΟΥ ΔΡΟΜΟΛΟΓΙΟΥ

ΣΤΟΙΧΕΙΑ ΕΠΙΒΑΤΗ

Το παραπάνω σχήμα μειονεκτεί στο ότι για κάθε δρομολόγιο που επιλέγεται από πολλούς επιβάτες, σε κάθε καταχώρηση του θα επαναλαμβάνονται τα στοιχεία του. Αυτός ο πλεονασμός μπορεί να αποφευχθεί αν διασπάσουμε τον παραπάνω πίνακα σε δύο άλλους όπου ο ένας θα περιέχει τα στοιχεία του δρομολογίου και ο άλλος τα στοιχεία των επιβατών που το επέλεξαν. Με άλλα λόγια δημιουργείται μια σχέση ένα προς πολλά μεταξύ του πίνακα δρομολογίων και του πίνακα επιβατών, που το επέλεξαν . Τα στοιχεία θα καταχωρούνται από τον χρήστη μέσω Web διαπροσωπείας.

Έτσι προκύπτει μια βελτιωμένη έκδοση του παραπάνω σχήματος :

#### reservation

<u>id</u>	source	dest	dep_date	dep_time
-----------	--------	------	----------	----------

#### psger

id_card	discount	name	num_pos	route (ξκ)
---------	----------	------	---------	------------

Ο πίνακας reservation περιέχει τις πληροφορίες σχετικά με το δρομολόγιο στο οποίο έχει γίνει κράτηση. Έχει τα παρακάτω πεδία :

- id : κωδικός καταχωρημένου δρομολογίου. Στον πίνακα αυτό θα είναι πρωτεύων κλειδί και επιλέξαμε να είναι τύπου δεδομένων αυτόματης αρίθμησης.

- source : Η πόλη αφετηρίας του επιλεγμένου δρομολογίου. Θα παίρνει δεδομένα από την αντίστοιχη στήλη του πίνακα timetable.
- dest : Ομοίως η πόλη προορισμού του επιλεγμένου δρομολογίου.
- dep\_time : Η ώρα αναχώρησης του δρομολογίου από την αντίστοιχη στήλη του πίνακα timetable.
- dep\_date : Θα είναι η ημερομηνία που επιλέγει ο χρήστης να ταξιδέψει. Θα εισάγεται μέσω της Web διαπροσωπίας . Μέσω αυτής θα ελέγχεται αν υπάρχει διαθεσιμότητα για το συγκεκριμένο δρομολόγιο διαβάζονταςτα περιεχόμενα της στήλης days του πίνακα timetable.

Αντίστοιχα ο πίνακας psger , που θα καταχωρούνται τα στοιχεία των επιβατών, που έκαναν κράτηση , θα περιέχει τις παρακάτω στήλες :

- id\_card : ο κωδικός της εκπτωτικής κάρτας που τυχόν έχει ο επιβάτης. Είναι τύπου χαρακτήρα και επιτρέπονται NULL τιμές.
- discount : ο επιβάτης μέσω της web διαπροσωπείας επιλέγει έκπτωση σε περίπτωση που διαθέτει εκπτωτική κάρτα.
- name : το όνομα επιβάτη στο οποίο θα γίνει η κράτηση
- num\_pos : ο αριθμός θέσεων που θέλει να κρατήσει ο επιβάτης. Με βάση τις προδιαγραφές του προβλήματος πρέπει να είναι 1 – 4. Ο έλεγχος αυτός θα γίνεται από την web διαπροσωπεία.
- route : ξένο κλειδί που αντιστοιχεί στο πρωτεύων κλειδί του πίνακα reservation και αναπαριστά το δρομολόγιο που επέλεξε ο χρήστης.

Οι εντολές mysql που δημιουργούν τους παραπάνω πίνακες είναι :

```
CREATE TABLE IF NOT EXISTS reservation
(
  id INT UNSIGNED NOT NULL AUTO_INCREMENT,
  source char(17) NOT NULL,
  dest CHAR(17) NOT NULL,
  dep_date DATE NOT NULL,
  dep_time TIME NOT NULL,
  PRIMARY KEY(id)
);
```

```
CREATE TABLE IF NOT EXISTS psger
(
  id_card CHAR(17) NULL,
```

```
discount CHAR(17) NOT NULL ,
name CHAR(17) NOT NULL,
num_pos INT NOT NULL,
route INT UNSIGNED NOT NULL,
CONSTRAINT CHOOSE
FOREIGN KEY (route)
REFERENCES reservation(id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

Έχουμε επιλέξει ακεραιότητα αναφορών μεταξύ των σχετιζόμενων εγγραφών των δύο πινάκων ως προς ενημέρωση και διαγραφή. Αυτό σημαίνει ότι όταν αλλάξει το πρωτεύον κλειδί σε μια εγγραφή του πίνακα της πλευράς 1 , θα τροποποιηθεί αυτόματα το ξένο κλειδί των σχετιζόμενων εγγραφών. Ομοίως αν διαγραφεί μια εγγραφή από τον πίνακα reservation , θα διαγραφούν αυτόματα από τον πίνακα psger οι σχετιζόμενες εγγραφές.

Επειδή είναι χρονοβόρα η εκτέλεση εντολών mysql δημιουργίας πινάκων και καταχώρησης δεδομένων σε αυτούς , δεν συνιστάται η εκτέλεση τους από τον χρήστη ακολουθιακά μέσω της γραμμής εντολών αλλά η ενσωμάτωση τους σε ένα αρχείο και η εκτέλεση του αρχείου αυτού σε κατάσταση ομαδικής επεξεργασίας. Οι εντομές δημιουργίας πινάκων καθώς και εισαγωγής δεδομένων σε αυτούς γίνεται με την εκτέλεση του αρχείου *dbktel.txt* μέσω της γραμμής εντολών :

`\. dbktel.txt` ή `SOURCE dbktel.txt`

Το συνολικό σχήμα της βάσης δεδομένων του προβλήματος μας είναι :

### timetable

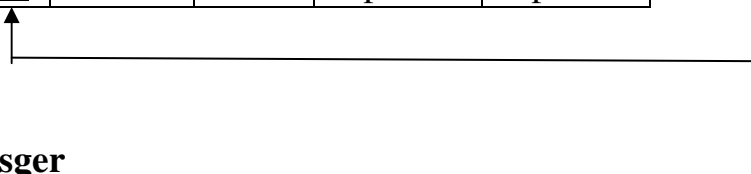
<u>source</u>	<u>dep_time</u>	days	<u>dest</u>	remarks
---------------	-----------------	------	-------------	---------

### reservation

<u>id</u>	source	dest	dep_date	dep_time
-----------	--------	------	----------	----------

### psger

id_card	discount	name	num_pos	route (ξκ)
---------	----------	------	---------	------------



# Κεφάλαιο 4<sup>ο</sup>

## 4. Ιστότοπος για πρόσβαση στα δεδομένα της βάσης

### 4.1 Δομή – Οργάνωση – Λειτουργία του ιστότοπου

Λογισμικά που χρησιμοποιήθηκαν για την δημιουργία του ιστότοπου :

- Windows XP
- Wamp server (<http://www.wampserver.com>)
  - Apache Web Server v. 2.2.11
  - MySQL 5.1.32
  - PHP 5.2.9-1
- Notepad
- Internet Explorer
- Mozilla Firefox

Για κάθε ιστοσελίδα έχουμε επιλέξει ομοιόμορφη εμφάνιση ως προς το χρώμα φόντου, τον τίτλο, την εικόνα λεωφορείου και το μενού επιλογών χρήστη.

Η κεντρική σελίδα επιλέγεται φορτώνοντας το αρχείο *index.html* :

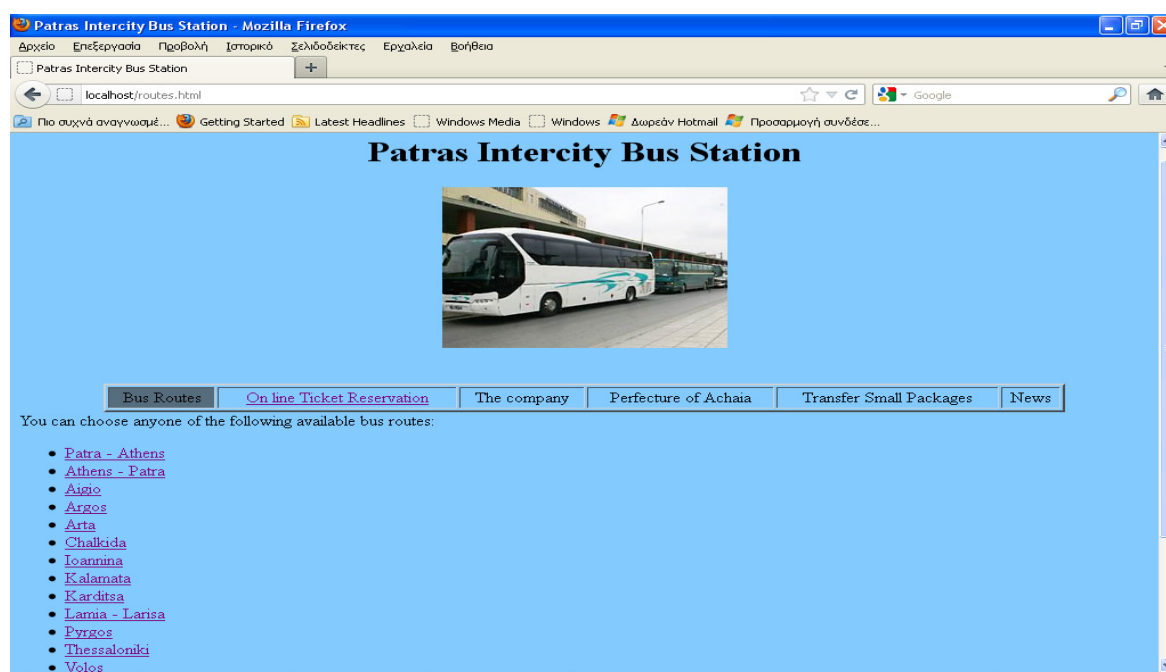




Εμφανίζεται ένα αρχικό μενού επιλογών, από το οποίο ο χρήστης μπορεί να επιλέξει τον σύνδεσμο Bus Routes ώστε να ενημερωθεί για τις ώρες και ημέρες των δρομολογίων από και προς Πάτρα , καθώς και τον σύνδεσμο On line Ticket Reservation, που του παρέχει δυνατότητα για ηλεκτρονική κράτηση εισητηρίου. .

## 4.2 Ενημέρωση κοινού για δρομολόγια

Αφού ο χρήστης επιλέξει από το αρχικό μενού τον πρώτο σύνδεσμο, εκτελείται ο κώδικας *routes.html*.



Στον παρακάτω πίνακα παρουσιάζουμε τι περιέχει ο κάθε σύνδεσμος και το αρχείο κώδικα php που τον υλοποιεί.

Σύνδεσμος	Δρομολόγια	Αρχείο
Patra - Athens	Πάτρα - Αθήνα	patra_athina.php
Athens - Patra	Αθήνα - Πάτρα	athina_patra.php
Aigio	Από και προς Αίγιο	aigio.php
Argos	Από και προς Άργος	argos.php
Arta	Από και προς Άρτα	arta.php
Chalkidi	Από και προς Χαλκίδα	chalkida.php
Ioannina	Από και προς Ιωάννινα	ioannina.php
Kalamata	Από και προς Καλαμάτα	kalamata.php
Karditsa	Από και προς	karditsa.php

	Καρδίτσα	
Lamia - Larisa	Από και προς Λαμία - Λάρισα	lamia.php
Pyrgos	Από και προς Πύργο	pyrgos.php
Thessaloniki	Από και προς Θεσσαλονίκη	thessaloniki.php
Volos	Από και προς Βόλο	volos.php

Καθένα από τα παραπάνω αρχεία κώδικα εκτελείται αυτόματα , μόλις επιλέγεται ο αντίστοιχος σύνδεσμος, και περιέχει κώδικα που κάνει αναζήτηση του συγκεκριμένου δρομολογίου στην βάση. Εμφανίζει τα αποτελέσματα αναζήτησης σε πίνακες , όπου για κάθε ημέρα της εβδομάδας εμφανίζονται οι αντίστοιχες ώρες δρομολογίου. Ενδεικτικά ας δούμε τον κώδικα του αρχείου *patra\_adina.php* , που εμφανίζει τις ώρες και ημέρες δρομολογίων από Αθήνα προς Πάτρα και όχι αντίστροφα :

```

1. <html>
2. <head>
3. <title>
4. Patras Intercity Bus Station - Patra - Athina
5. </title>
6. </head>
7. <body bgcolor=#82CAFF>
8. <center>
9. <h1>Patras Intercity Bus Station</h1>
10.
11.</center>
12.<br><br>
13.<table align=center width=85% border=2>
14.<tr>
15.<td align=middle bgcolor=#566D7E><a href="routes.html">Bus Routes</a></td>
16.<td align=middle><a href="reserve1.php">On line Ticket Reservation</a></td>
17.<td align=middle>The company</td>
18.<td align=middle>Perfecture of Achaia</td>
19.<td align=middle>Transfer Small Packages</td>
20.<td align=middle>News</td>
21.</tr>
22.</table>
23.<?php
24.print'<font size=+2> Patra - Athens</font>';
25.print'<p>';
26.print'<table width=80% border=1>';
27.print'<tr>';

```

```

28.print'<td bgcolor=#00FFFF align=middle> Time</td>';
29.print'<td bgcolor=#00FFFF align=middle> Monday</td>';
30.print'<td bgcolor=#00FFFF align=middle> Thursday</td>';
31.print'<td bgcolor=#00FFFF align=middle> Wensday</td>';
32.print'<td bgcolor=#00FFFF align=middle> Thursday</td>';
33.print'<td bgcolor=#00FFFF align=middle> Friday</td>';
34.print'<td bgcolor=#00FFFF align=middle> Saterdag</td>';
35.print'<td bgcolor=#00FFFF align=middle> Sunday</td>';
36.print'<td bgcolor=#00FFFF align=middle> Remarks</td>';
37.print'</tr>';
38.$hostname='localhost';
39.$user='root';
40.$passwd="";
41.$dbc=mysql_connect($hostname,$user,$passwd);
42.mysql_select_db('ktel');
43.$query="SELECT * FROM timetable WHERE source='Patra' AND dest='Athens'";
44.$r=mysql_query($query);
45.while($row=mysql_fetch_array($r))
46.{
47.print'<tr>';
48.$t=substr($row[dep_time],0,5);
49.print"<td align=middle>$t</td>";
50.$days=array("Monday", "Thursday", "Wednesday", "Thursday", "Friday", "Saturday", "Sun
51.foreach($days as $key=>$value)
52.{
53.if(strstr($row[days],$value))
54.print"<td align=middle>X</td>";
55.else
56.print"<td> &nbsp; </td>";
57.}
58.$str=(strlen($row[remarks])>0?$row[remarks]:"&nbsp;");
59.print"<td align=middle>$str</td>";
60.print'</tr>';
61.}
62.mysql_close();
63.?>
64.<p>Every Friday and Sunday may be made for additional routes serving the traveling p
65.</body>
66.</html>

```

Γραμμές 1 – 6 : Δημιουργία γραμμής τίτλου της τρέχουσας σελίδας αποτελεσμάτων.

Γραμμές 7 – 12 : Καθορισμός χρώματος φόντου καθώς και τίτλου περιεχομένου ιστοσελίδας.

Γραμμές 13 – 22 : Εμφάνιση με χρήση πίνακα ενός αρχικού μενού επιλογών στον χρήστη της ιστοσελίδας.

Γραμμές 23 – 37 : Έναρξη `php` κώδικα. Εμφάνιση της επικεφαλίδας του πίνακα των δρομολογίων.

Γραμμές 38 – 41 : Σύνδεση του προγράμματος `php`, με την `mysql`. Είναι εγκατεστημένα και τα δύο περιβάλλοντα στον ίδιο υπολογιστή. Συνδεόμαστε ως χρήστης `root`, ο οποίος δημιουργείται αυτόματα με την εγκατάσταση του `wamp server` χωρίς κωδικό πρόσβασης. Δεν συνίσταται για λόγους ασφάλειας. Έχουμε δημιουργήσει κατά την εκτέλεση του αρχείου `kteldb.txt` στο τέλος έναν χρήστη με όνομα `e_system1` και συνθηματικό `ktel`, με κατάλληλα δικαιώματα στους πίνακες `reservation`, `psger`.

```
CREATE USER 'e_system1'@'%' IDENTIFIED BY 'ktel';
GRANT SELECT ON ktel.timetable TO 'e_system1'@'%';
GRANT INSERT,UPDATE ON ktel.reservation TO 'e_system1'@'%';
GRANT INSERT,UPDATE ON ktel.psger TO 'e_system1'@'%';
```

Γραμμή 42 : Η βάση δεδομένων `ktel` γίνεται τρέχουσα.

Γραμμές 43 – 44 : Δημιουργούμε το ερώτημα για την εύρεση δρομολογίων από Πάτρα προς Αθήνα και υποβάλλεται στον εξυπηρετητή της `mysql`.

Γραμμές 45 – 58 : Σαρώνεται ο πόρος με τα αποτελέσματα και κάθε γραμμή, που αντιστοιχεί σε μια συγκεκριμένη ώρα δρομολογίου, σημειώνουμε με `X` την αντίστοιχη ημέρα στήλη.

Στην παρακάτω εικόνα εμφανίζεται ένα μέρος των δρομολογίων :

localhost/patra\_a9ina.php

Πιο συχνά αναγνωμέ... Getting Started Latest Headlines Windows Media Windows Δωρεάν Hotmail Προσαρμογή συνδέσε...

### Patra - Athens

Every Friday and Sunday may be made for additional routes serving the traveling public.

Time	Monday	Thursday	Wednesday	Thursday	Friday	Saturday	Sunday	Remarks
02:30	X		X	X	X	X	X	
05:00	X							
05:15			X	X	X			
05:45	X					X		
06:00	X		X	X	X		X	Express
06:30	X		X	X	X	X		
07:00							X	
07:15	X		X	X	X	X		
07:45	X		X	X	X	X	X	Express
08:15	X		X	X	X	X	X	
09:00	X		X	X	X		X	
09:15						X		
09:30	X		X	X	X		X	
10:00	X		X	X	X	X	X	
10:30	X						X	
10:45			X	X	X	X		
11:00	X						X	
11:30	X		X	X	X	X	X	Express
12:00	X		X	X	X	X	X	

### 4.3 Ηλεκτρονική Κράτηση Εισιτηρίου

Αν από το μενού της αρχικής σελίδας επιλέξω On line ticket reservation, λαμβάνει χώρα η διαδικασία ηλεκτρονικής κράτησης εισιτηρίου που περιλαμβάνει τέσσερα στάδια. Κάθε στάδιο υλοποιείται αντίστοιχα από τα αρχεία κώδικα *reserve1.php*, *reserve2.php*, *reserve3.php* και *reserve4.php*. Αρχικά εκτελείται το αρχείο *reserve1.php* και εμφανίζεται στον χρήστη η παρακάτω διαπροσωπεία

Patras Intercity Bus Station - Ticket Reservation - Mozilla Firefox

Αρχείο Επεξεργασία Προβολή Ιστορικό Σελιδοδείκτες Εργαλεία Βοήθεια

localhost/reserve1.php

Πιο συχνά αναγνωμέ... Getting Started Latest Headlines Windows Media Windows Δωρεάν Hotmail Προσαρμογή συνδέσε...

## Patras Intercity Bus Station



[Bus Routes](#)
[On line Ticket Reservation](#)
[The company](#)
[Perfucture of Achaia](#)
[Transfer Small Packages](#)
[News](#)

From:

To:

Date:

Seats:

[Ticket issuance instruction](#) | [Term of use](#)

Από όπου μπορεί να επιλέξει την πόλη αφετηρίας, προορισμού , την ημερομηνία ταξιδιού και των αριθμό θέσεων (από 1 – 4 ). Το σύστημα υποστηρίζει δυνατότητα κράτησης μόνο από και προς συγκεκριμένες πόλεις και όχι για όλα τα διαθέσιμα δρομολόγια. Μόλις ο χρήστης επιλέξει το κουμπί με τίτλο Continue και κάνει υποβολή φόρμας εκτελείται αυτόματα το αρχείο reserve2.php . Παρουσιάζουμε μόνο το τμήμα του αρχείου που περιέχει εντολές php και όχι html στοιχεία :

```
1. <?php
2.   $from=$_POST['from'];
3.   $tt=$_POST['to'];
4.   $day=$_POST['day'];
5.   $month=$_POST['month'];
6.   $year=$_POST['year'];
7.   $seats=$_POST['seats'];
8.   $date=mktime(0,0,0,$month,$day,$year);
9.   if(!checkdate($month,$day,$year))
10.  {
11.    print'<br><br><font size="5" color="red">you should select a start date. Please
    try again!</font><br>';
12.    for($i=1;$i<=8;$i++)
13.      print'<br>';
14.  }
15.  else
16.  {
17.    $hostname='localhost';
18.    $user='root';
19.    $passwd="";
20.    $dbc=mysql_connect($hostname,$user,$passwd);
21.    mysql_select_db('ktel');
22.    $query="SELECT * FROM timetable WHERE source='{$_POST['from']}' AND
    dest='{$_POST['to']}'" ;
23. $row=mysql_fetch_array($r);
24. if(!$row)
25.  {
26.
27.
28.    print'<br><br><font size="5" color="red">The selected bus route is not
    available. Please try again!</font><br>';
29.    for($i=1;$i<=8;$i++)
```

```

30.         print'<br>';
31.     }
32.     else
33.     {
34.         print'<center>';
35.         $r=mysql_query($query);
36.         $d=date('l',$date);
37.     print"<br><font size=+2 > $d $day $month $year</font> <br>";
38.         print"<br><font size=+2 > $from - $tt </font> <br>";
39.
40.         print'<table width=50% border=0>';
41.         print'<tr>';
42.         print'<td bgcolor=#00FFFF align=middle> Time </td>';
43.         print'<td bgcolor=#00FFFF align=middle> Remarks</td>';
44.         print'<td bgcolor=#00FFFF align=middle> </td>';
45.         print'</tr>';
46.     while($row=mysql_fetch_array($r))
47.     {
48.         print'<tr>';
49.         $t=substr($row[dep_time],0,5);
50.         $t1=$row[dep_time];
51.         if(strpos($row[days],$d))
52.         {
53.             print'<td align=middle>$t</td>';
54.             $str=(strlen($row[remarks])>0?$row[remarks]:"&nbsp;");
55.             print'<td align=middle>$str</td>';
56.             print'<td align=middle><input type="radio" name="tm" value=$t1>';
57.         }
58.         print'</tr>';
59.     }
60.     print'<input type="hidden" name=from value=$from />';
61.     print'<input type="hidden" name=to value=$tt />';
62.     print'<input type="hidden" name=dep_d value=$date />';
63.     print'<input type="hidden" name=seats value=$seats />';
64.     print'<td colspan=2 align="middle" ><input type="submit" name="submit"
value="Continue"></td>';
65.     print'</table>';
66.     print'</form>';
67. }
68. }
69. ?>

```

Γραμμές 2 – 7 : Λαμβάνουμε σε μεταβλητές τα στοιχεία που μας στάλθηκαν από το σενάριο reserve1.php, μέσω της μεθόδου post.

Γραμμή 8 : Δημιουργούμε μια μεταβλητή τύπου ημερομηνίας, από τα προηγούμενα στοιχεία, που θα είναι η ημερομηνία ταξιδιού.

Γραμμές 9 – 14 : Αρχικά γίνεται έλεγχος αν η ημερομηνία κράτησης είναι έγκυρη. Σε αυτήν την περίπτωση τυπώνεται κατάλληλο διαγνωστικό μήνυμα και διακόπτεται η διαδικασία κράτησης.

Γραμμές 17 – 23 : Διαφορετικά γίνεται σύνδεση του προγράμματος με το την mysql και υποβάλλεται το ερώτημα για εύρεση δρομολογίου με πόλεις αφετηρίας και προορισμού εκείνες που έδωσε ο χρήστης.

Γραμμές 24 – 31 : Σε περίπτωση που δεν υπάρχει διαθέσιμο δρομολόγιο με τα στοιχεία που έδωσε ο χρήστης, τυπώνεται κατάλληλο διαγνωστικό μήνυμα και διακόπτεται η διαδικασία κράτησης.

Γραμμές 32 – 69 : Διαφορετικά αρχικά βρίσκουμε την ημέρα της εβδομάδας που αντιστοιχεί στην ημερομηνία ταξιδιού (Γραμμή 36) . Ακολουθεί η δημιουργία της επικεφαλίδας του πίνακα που θα περιέχει τις ώρες ταξιδιού και το αντίστοιχο κουμπί επιλογής (Γραμμές 38 – 45 ) . Έπειτα αφού γίνει αναζήτηση των ωρών του επιλεγμένου δρομολογίου συμπληρώνεται ο πίνακας φόρμα με τις ώρες ταξιδιού και το αντίστοιχο κουμπί επιλογής.


Γραμμές 60 – 63 : Τα δεδομένα του χρήστη που μας στάλθηκαν από το σενάριο reserve1.php για να είναι διαθέσιμα και στα επόμενα σενάρια, τα ενσωματώνουμε στην φόρμα σαν κρυφά πεδία.

Στην παρακάτω εικόνα φαίνονται τα αποτελέσματα εκτέλεσης του σεναρίου reserve2.php , για τα δεδομένα εισόδου χρήστη που φαίνονται στην αντίστοιχη εικόνα του σεναρίου reserve1.php :

Patras Intercity Bus Station - Mozilla Firefox

localhost/reserve2.php

### Patras Intercity Bus Station



[Bus Routes](#) | [On line Ticket Reservation](#) | [The company](#) | [Perfecture of Achaia](#) | [Transfer Small Packages](#) | [News](#)

Sunday 17 2012

Patra - Thessaloniki

Time	Remarks
08:30	<input type="radio"/>
12:15	<input type="radio"/>
15:15	<input type="radio"/>
21:00	<input type="radio"/>

[Ticket issuance instruction](#) | [Term of use](#)



Μόλις ο χρήστης επιλέξει το κουμπί υποβολής με ετικέτα Continue, εκτελείται το σενάριο reserve3.php που περιέχει τον ακόλουθο php κώδικα :

```
1. <?php
2. $from=$_POST['from']
3. $to=$_POST['to']
4. $date=$_POST['dep_d']
5. $seats=$_POST['seats']
6. $tm=$_POST['tm']
7. $date=date("Y-m-d",$date)
8. print'<br><br>';
9. if(!isset($tm))
10.  print '<br><br><font size="5" color="red">You should choose a desirable travel
    time!</font><br>';
11. else
12. {
13.  $hostname='localhost';
14.  $user='root';
15.  $passwd="";
16.  $dbc=mysql_connect($hostname,$user,$passwd);
17.  mysql_select_db('ktel');
18.  print'<center>';
19.
20.  print'<form method="post" action="reserve4.php">';
21.  print'<table width=50% border=0>';
22.  print'<tr>';
23.  print'<td bgcolor=#00FFFF align=middle height="20"> Discount </td>'
24.  print'<td>'
25.  print'<select name="discount">'
26.  print'<option value=" ">None</option>'
27.  print'<option value="25%">25%</option>'
28.  print'<option value="50%">50%</option>'
29.  print'</select>'
30.  print'</td>'
31.  print'</tr>'
32.  print'<tr>'
33.  print'<td bgcolor=#00FFFF align=middle height="20"> Name </td>'
34.  print'<td><input type="text" name="name" size="25" value="Enter your name
    here!"/></td>'
35.  print'</tr>'
```

```

36. print'<td bgcolor=#00FFFF align=middle height="20"> Card code (in case it is
exists) </td>'
37. print'<td><input type="text" name="card" size="25" /></td>'
38. print'</tr>'
39. print'</tr>'
40. print'<td bgcolor=#00FFFF align=middle height="20"> email </td>'
41. print'<td><input type="text" name="mail" size="25" /></td>'
42. print'</tr>'
43. print"<input type=\"hidden\" name=from value=$from />"
44. print"<input type=\"hidden\" name=to value=$to />"
45. print"<input type=\"hidden\" name=seats value=$seats />"
46. print"<input type=\"hidden\" name=dep_d value=$date />"
47. print"<input type=\"hidden\" name=dep_t value=$tm />"
48. print'<tr>'
49. print'<td size="15"> &nbsp;
50. print'<td> &nbsp;
51. print'</tr>'
52. print'<tr>';
53. print'<td colspan=2 align="middle" ><input type="submit" name="submit"
value="Continue"></td>';
54. print'</tr>';
55. print'</table>';
56. print'</form>';
57. mysql_close();
58. }
59.
60. ?>

```

Γραμμές 2 – 7 : Αρχικά λαμβάνουμε τα δεδομένα εισόδου του χρήστη από τα κρυφά πεδία της φόρμας του reserve2.php .

Γραμμές 9 – 10 : Γίνεται έλεγχος αν ο χρήστης έχει επιλέξει κάποια ώρα ταξιδιού και σε περίπτωση που δεν έχει επιλέξει τυπώνεται κατάλληλο διαγνωστικό μήνυμα.

Γραμμές 11 – 18 : Διαφορετικά αρχικά θα γίνει σύνδεση με την βάση ktelddb, στον εξυπηρετητή της mysql, ως χρήστης root.

Γραμμές 24 – 31 : Δημιουργώ μια φόρμα, όπου το πρώτο στοιχείο της θα είναι ένα μενού επιλογής , από όπου ο χρήστης θα επιλέγει μια έκπτωση που τυχόν δικαιούται ( καθόλου, 25% ή 50%).

Γραμμές 32 – 36 : Το επόμενο στοιχείο της φόρμας θα είναι ένα πεδίο κείμενου , όπου ο χρήστης θα συμπληρώνει το όνομα στο οποίο θα έχει γίνει η κράτηση.

Γραμμές 36 – 37 : Εδώ θα υπάρχει ένα πεδίο κειμένου, όπου ο χρήστης θα συμπληρώνει τον κωδικό της εκπτωτικής κάρτας, που τυχόν διαθέτει.

Γραμμές 38 – 39 : Το τελευταίο ορατό πεδίο, όπου ο χρήστης θα συμπληρώνει το email του.

Γραμμές 43 – 48 : Κρυφά πεδία με είσοδο του χρήστη από τα προηγούμενα δύο σενάριο, που θα πρέπει να μεταβιβαστούν στο επόμενο σενάριο reserve4.php.

Αφού επιλέξουμε το κουμπί υποβολής εκτελείται το σενάριο reserve4.php, όπου ολοκληρώνεται η διαδικασία ηλεκτρονικής κράτησης :

```
1. <?php
2.   $from=$_POST['from'];
3.   $to=$_POST['to'];
4.   $date=$_POST['dep_d'];
5.   $seats=$_POST['seats'];
6.   $tm=$_POST['dep_t'];
7.   $discount=$_POST['discount'];
8.   $name=$_POST['name'];
9.   $card=$_POST['card'];
10.  $email=$_POST['mail'];
11.  $email_pattern='/^([[:alnum:]]_ - \.){1}+(@{1}[[:alnum:]]+)(\.{1}[a-z]{2,3})$/';
12.  if(!preg_match($email_pattern,$email))
13.    print '<br><br><font size="5" color="red">You should enter a valid email
    address!</font><br>';
14.  else
15.    {
16.    $hostname='localhost';
17.    $user='root';
18.    $passwd="";
19.    $dbc=mysql_connect($hostname,$user,$passwd);
20.    mysql_select_db('ktel');
21.    $query="SELECT * FROM reservation WHERE (source='$from' AND dest='$to'
    AND
22.    dep_date='$date' AND dep_time='$tm')" ;
23.    $r=mysql_query($query);
24.    while($row=mysql_fetch_array($r))
25.    {
26.      ;
27.    }
28.    if(mysql_num_rows($r)==1)
29.    {
30.      $r=mysql_query($query);
31.      $row=mysql_fetch_array($r);
```

```

32. $id=$row['id'];
33. $query="SELECT * FROM reservation,psger WHERE (source='$from' AND
dest='$to' AND dep_date='$date' AND dep_time='$tm' AND
reservation.id=psger.route)" ;
34. $r=mysql_query($query);
35. if($seats>20 - mysql_num_rows($r))
36.     print '<br><br><font size="5" color="red"> There is not sufficient room for the
requested number of seats !</font><br>';
37. else
38. {
39.     $query="insert into psger values('$card','$discount','$name',$seats,$id,'$email)";
40.     mysql_query($query);
41.     $t1=strtotime( $date );
42.     $d = date( 'l j-m-Y', $t1);
43.     $tm=substr($tm,0,5);
44.     print'<center>';
45.     print'Succesful ticket reservation!<br><br>';
46.     print'<table width=50% border=1>';
47.     print'<tr>';
48.     print'<td align=left> From </td>'
49.     print"<td align=middle> $from</td>"
50.     print'</tr>'
51.     print'<tr>'
52.     print'<td align=left> To </td>'
53.     print"<td align=middle> $to</td>"
54.     print'</tr>'
55.     print'<tr>'
56.     print'<td align=left> Date </td>'
57.     print"<td align=middle>$d</td>"
58.     print'<tr>';
59.     print'<td align=left> Time </td>';
60.     print"<td align=middle> $tm</td>";
61.     print'</tr>';
62.     print'<tr>';
63.     print'<td align=left> Name</td>';
64.     print"<td align=middle>$name</td>";
65.     print'</tr>';
66.     print'<tr>';
67.     print'<td align=left> Email</td>';
68.     print"<td align=middle>$email</td>";
69.     print'</tr>';
70.     print'</table>';
71.     print'<p> You should be at the ticket office at least 30 minutes before the departures

```

```

of the selected bus route.</p>';
72.     print'</center>';
73.   }
74. }
75. else
76. {
77.   $query1="insert into reservation values(0,'$from','$to','$date','$tm)";
78.   $id=mysql_insert_id();
79.   $query2="insert into psger values('$card','$discount','$name','$seats',$id,'$email)";
      mysql_query($query2);
80.   $t1=strtotime( $date );
81.     $d = date( 'l j-m-Y', $t1);
82.     $tm=substr($tm,0,5);
83.     print'<center>';
84.     print'Succesful ticket reservation!<br><br>';
85.     print'<table width=50% border=1>';
86.   print'<tr>';
87.     print'<td align=left> From </td>';
88.     print"<td align=middle> $from</td>";
89.     print'</tr>';
90.     print'<tr>';
91.     print'<td align=left> To </td>';
92.     print"<td align=middle> $to</td>";
93.     print'</tr>';
94.     print'<tr>';
95.     print'<td align=left> Date </td>';
96.     print"<td align=middle>$d</td>";
97.     print'</tr>';
98.     print'<tr>';
99.     print'<td align=left> Time </td>';
100    print"<td align=middle> $tm</td>";
101    print'</tr>';
102    print'<tr>';
103    print'<td align=left> Name</td>';
104    print"<td align=middle>$name</td>";
105    print'</tr>';
106    print'<tr>';
107    print'<td align=left> Email</td>';
108    print"<td align=middle>$email</td>";
109    print'</tr>';
      print'</table>';
      print'<p> You should be at the ticket office at least 30 minutes before the
departures of the selected bus route.</p>';

```

```

        print'</center>';
110 }
    mysql_close();
}
?>

```

Γραμμές 2 – 10 : Λαμβάνουμε τα δεδομένα της φόρμας από το προηγούμενο σενάριο με αποστολή POST.

Γραμμές 12 – 13 : Με χρήση κανονικών εκφράσεων ελέγχω αν το email του χρήστη είναι σε αποδεκτή μορφή , αν δεν είναι τυπώνεται κατάλληλο διαγνωστικό μήνυμα.

Γραμμές 14 – 20 : Διαφορετικά συνεχίζεται κανονικά η διαδικασία της ηλεκτρονικής κράτησης με σύνδεση με την βάση δεδομένων.

Γραμμές 21 – 23 : Υποβάλλω ερώτημα για να δω αν υπάρχει καταχώρηση στον πίνακα reservation με το ίδιο δρομολόγιο, δηλαδή αν κάποιος άλλος χρήστης έχει κάνει κράτηση για το ίδιο δρομολόγιο.

Γραμμή 28 : Ελέγχω την περίπτωση, όπου υπάρχει ήδη μια κράτηση για το δρομολόγιο.

Γραμμές 30 – 32 : Βρίσκω τον κωδικό του ζητούμενου δρομολογίου όπως είναι καταχωρημένο στον πίνακα reservation.

Γραμμές 33 – 34 : Βρίσκω τον αριθμό των επιβατών που έχουν κάνει κράτηση για το συγκεκριμένο δρομολόγιο, σαν ερώτημα εύρεσης συσχετιζόμενων εγγραφών της εγγραφής στον πίνακα reservation, με τις αντίστοιχες του πίνακα psger.

Γραμμές 34 – 36 : Επειδή υπάρχει ο περιορισμός ότι μέσω ηλεκτρονικής κράτησης θα γίνονται το πολύ 20 κρατήσεις θέσεων, αρχικά γίνεται έλεγχος αν ο ζητούμενος αριθμός θέσεων από τον χρήστη συν των ήδη υπάρχον αριθμό κρατήσεων για το συγκεκριμένο δρομολόγιο υπερβαίνει τις 20. Σε μια τέτοια περίπτωση η διαδικασία της κράτησης σταματά με εμφάνιση κατάλληλου μηνύματος.

Γραμμή 37 : Αν υπάρχουν ακόμα διαθέσιμες θέσεις καταχωρούμε στον πίνακα psger τα στοιχεία της κράτησης του χρήστη.

Γραμμές 41 – 72 : Εμφάνιση σε κατάλληλα μορφοποιημένο πίνακα των στοιχείων με την κράτηση εισιτηρίου. Μπορούν να εκτυπωθούν και να τα προσκομίσει ο επιβάτης στα εκδοτήρια , προκειμένου να εκδώσει το εισιτήριό του.

Γραμμές 75 – 76 : Από εδώ και κάτω εξετάζουμε την περίπτωση, που για το ζητούμενο δρομολόγιο δεν έχει γίνει κάποια κράτηση από άλλον ενδιαφερόμενο επιβάτη.

Γραμμές 77 – 79 : Καταχώρηση στον πίνακα reservation των στοιχείων του δρομολογίου. Ο κωδικός δρομολογίου παράγεται αυτόματα. Στην συνέχεια ακολουθεί η αντίστοιχη καταχώρηση με τα στοιχεία του

επιβάτη στον πίνακα psger, όπου θα έχει σαν πεδίο ξένου κλειδιού το πρωτεύον κλειδί κωδικού δρομολογίου του πίνακα psger.  
Γραμμές 80 – 110 : Ομοίως όπως στις γραμμές 41 – 72

#### 4.4 Λειτουργία ηλεκτρονικής κράτησης

Επιλέγουμε από την αρχική σελίδα, όπως εμφανίζεται στην σελίδα 29 τον σύνδεσμο on line ticket reservation και εκτελείται το σενάριο reserve1.php όπου εμφανίζεται στον χρήστη η παρακάτω φόρμα, από όπου επιλέγουμε να κάνουμε κράτηση για το δρομολόγιο της γραμμής Πάτρας – Θεσσαλονίκης την 1<sup>η</sup> Ιουλίου 2012 για μια θέση :

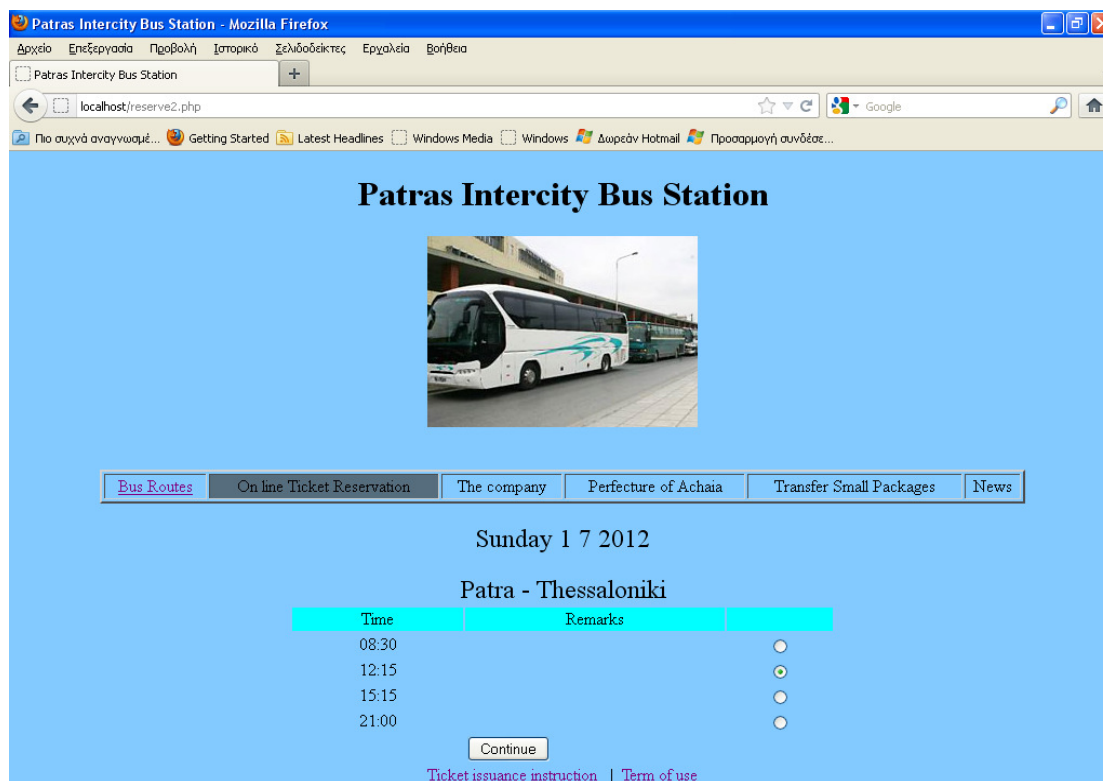


The screenshot shows a web browser window titled "Patras Intercity Bus Station - Ticket Reservation - Mozilla Firefox". The address bar shows "localhost/reserve1.php". The page content includes the title "Patras Intercity Bus Station" and a photograph of a white bus with green accents. Below the photo is a navigation menu with the following items: "Bus Routes", "On line Ticket Reservation", "The company", "Perfection of Achaia", "Transfer Small Packages", and "News". The main form has the following fields:

- From: Patra (dropdown menu)
- To: Thessaloniki (dropdown menu)
- Date: 1 (dropdown), July (dropdown), 2012 (dropdown)
- Seats: 1 (dropdown)
- Continue (button)

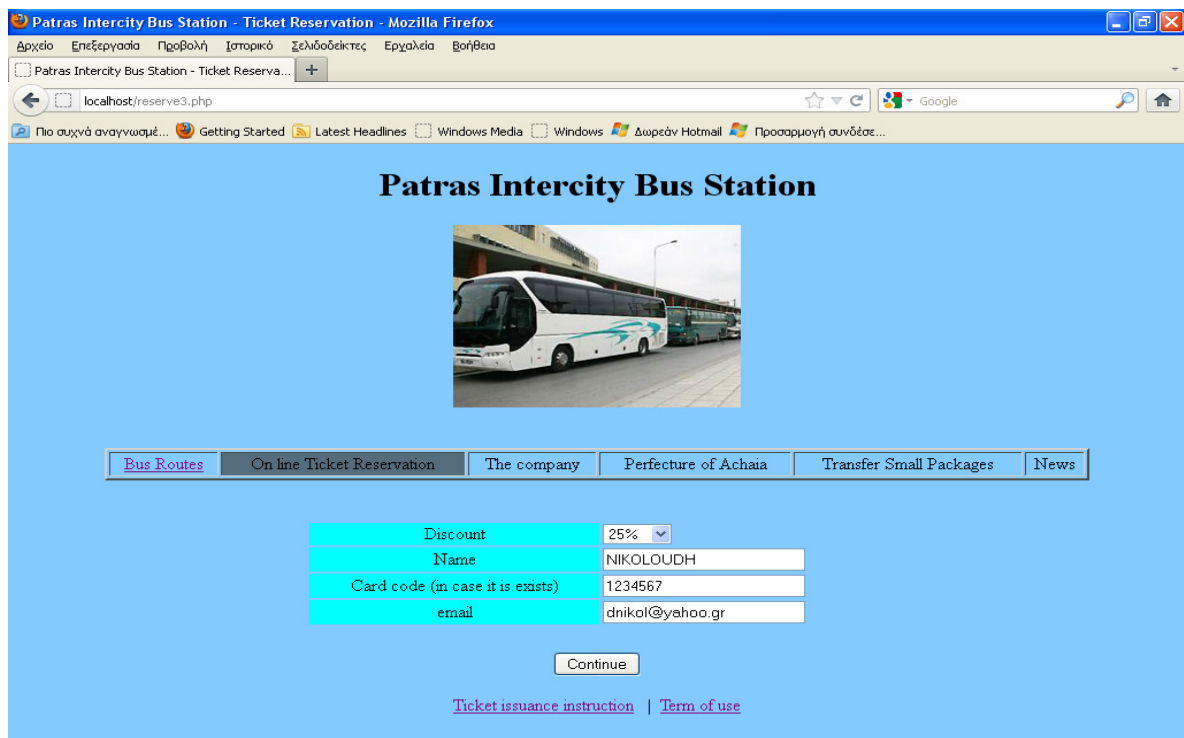
At the bottom of the page, there are two links: "Ticket issuance instruction" and "Term of use".

Στην συνέχεια πατώντας το Continue, μεταβαίνουμε στο σενάριο reserve2.php, που υλοποιεί την δεύτερη φάση της κράτησης :



Αρχικά το σύστημα ελέγχει αν η ημερομηνία ταξιδιού είναι έγκυρη. Στην συνέχεια ελέγχει αν υπάρχει στην βάση δεδομένων το ζητούμενο δρομολόγιο. Στην περίπτωση μας και οι δύο έλεγχοι είναι επιτυχείς. Έπειτα το σύστημα βρίσκει την ημέρα της εβδομάδας της ζητούμενης ημερομηνίας και τυπώνει με έντονα γράμματα την ημερομηνία στην οθόνη. Στην συνέχεια για το ζητούμενο δρομολόγιο την συγκεκριμένη ημέρα εμφανίζουμε σε πίνακα – φόρμα τις ώρες που είναι διαθέσιμο και ο χρήστης μπορεί να επιλέξει μια από αυτές. Επιλέξαμε τις 12:15 και μετά το κουμπί υποβολής εκτελείται το σενάριο reserve3.php :





Αφού ελέγξει ότι ο χρήστης έχει επιλέξει μια ώρα ταξιδιού στην συνέχεια εμφανίζει μια φόρμα όπου επιλέγουμε έκπτωση από μια λίστα (αν δικαιούμαστε), ένα όνομα στο οποίο θα γίνει η κράτηση, ο αριθμός πάσο αν υπάρχει και το email μας. Σε επίπεδο διαχειριστή συστήματος βλέπουμε ότι οι δύο πίνακες με τις κρατήσεις είναι κενοί :

```

c:\ C:\WINDOWS\system32\cmd.exe - mysql -u root
Microsoft Windows XP [Έκδοση 5.1.2600]
(C) Πνευματικά δικαιώματα 1985-2001 Microsoft Corporation

C:\Documents and Settings\giannis>cd C:\wamp\bin\mysql\mysql15.0.51a\bin
C:\wamp\bin\mysql\mysql15.0.51a\bin>mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.51a-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select * from reservation;
ERROR 1046 (3D000): No database selected
mysql> use ktel;
Database changed
mysql> select * from reservation;
Empty set (0.02 sec)

mysql> select * from psger;
Empty set (0.00 sec)

mysql>

```

Αφού πατήσουμε το κουμπί υποβολής εκτελείται το σενάριο reserve4.php, το οποίο αρχικά ελέγχει αν το email είναι έγκυρο, με χρήση κανονικών εκφράσεων όπως προείπαμε. Στην συνέχεια διαπιστώνει ότι για το ζητούμενο δρομολόγιο δεν υπάρχει ακόμα καταχώρηση στον

πίνακα reservation και το καταχωρεί εκεί, ενώ στον πίνακα psg καταχωρεί μια σχετιζόμενη εγγραφή με τα στοιχεία του επιβάτη, που έκανε την κράτηση.

Patras Intercity Bus Station - Mozilla Firefox

localhost/reserve4.php

## Patras Intercity Bus Station

[Bus Routes](#) | 
 [On line Ticket Reservation](#) | 
 [The company](#) | 
 [Perfecture of Achaia](#) | 
 [Transfer Small Packages](#) | 
 [News](#)

Successful ticket reservation!

From	Patra
To	Thessaloniki
Date	Sunday 1-07-2012
Time	12:15
Name	NIKOLOUDH
Email	dnikol@yahoo.gr

You should be at the ticket office at least 30 minutes before the departures of the selected bus route.

Εδώ ολοκληρώνεται επιτυχώς η κράτηση και με μια εκτύπωση, ο χρήστης μπορεί να παραλάβει στα εκδοτήρια το εισιτήριό του. Σε επίπεδο διαχειριστή συστήματος βλέπουμε τις παρακάτω μεταβολές στους πίνακες κρατήσεων :

```
mysql> select * from reservation;
+----+-----+-----+-----+-----+
| id | source | dest       | dep_date | dep_time |
+----+-----+-----+-----+-----+
| 1  | Patra  | Thessaloniki | 2012-07-01 | 12:15:00 |
+----+-----+-----+-----+-----+
1 row in set (0.08 sec)

mysql> select * from psger;
+-----+-----+-----+-----+-----+-----+
| id_card | discount | name       | num_pos | route | email |
+-----+-----+-----+-----+-----+-----+
| 1234567 |          | NIKOLOUDH | 1       | 1     | dnikol@yahoo.gr |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)

mysql>
```

# Μέρος Β

## Κεφάλαιο 5<sup>ο</sup>

### 5 . Τι είναι το Android

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο βασίζεται στον πυρήνα του λειτουργικού συστήματος Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance, η οποία είναι μια κοινοπραξία εταιριών λογισμικού καθώς και κατασκευής υλικού (hardware), οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google.

Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού.

Τον Ιούλιο του 2005, η Google εξαγόρασε την Android Inc, μια μικρή εταιρεία με έδρα το Palo Alto στην California των ΗΠΑ. Εκείνη την εποχή ελάχιστα ήταν γνωστά για τις λειτουργίες της Android Inc, εκτός του ότι ανέπτυσαν λογισμικό για κινητά τηλέφωνα. Αυτή ήταν η αρχή της φημολογίας περί σχεδίων της Google για να διεισδύσει στην αγορά κινητής τηλεφωνίας.

Η πλατφόρμα Android επιτρέπει στους προγραμματιστές τη δημιουργία νέων και καινοτόμων εφαρμογών που αξιοποιούν πλήρως όλες τις δυνατότητες μιας συσκευής. Βασίζεται σε ένα ισχυρό λειτουργικό σύστημα, ένα πλήρες σύνολο βιβλιοθηκών, με πολλαπλές δυνατότητες χρήσης πολυμέσων και ένα πλήρες σύνολο εφαρμογών τηλεφωνίας. Αξιοποιεί βέβαια πλήρως τα νέας γενιάς τσιπ που έχουν σχεδιαστεί συγκεκριμένα για αυτή τη πλατφόρμα.

## 5.1 Η εξέλιξη του Android

Όπως αναφέραμε παραπάνω, το Android είναι ένα λειτουργικό σύστημα ανοιχτού κώδικα. Η εξέλιξη του λόγω της open source φύσης του είναι ραγδαία και αυτό αντικατοπτρίζεται στο γεγονός ότι οι 7 κύριες εκδόσεις του έχουν κυκλοφορήσει σε διάστημα 2.5 ετών, από τον Απρίλη του 2009 μέχρι τον Νοέμβριο του 2011.

Στην πληροφορική συνηθίζεται τα προϊόντα υλικού (hardware) και λογισμικού (software) να κυκλοφορούν εκτός από τον αριθμό έκδοσης τους, και με μία κωδική ονομασία. Η κωδική ονομασία μπορεί να είναι πχ ονόματα πόλεων (Windows Viena, Chicago), ονόματα ζώων (OSX Leopard, Lion), στην περίπτωση όμως του Android τα κώδικα ονόματα έρχονται στη μορφή επιδότησης!

### Android Cupcake

Η έκδοση “Cupcake”, βασισμένη στο Linux Kernel 2.6.27, παρουσιάστηκε στις 30 Απριλίου του 2009. Υποστηρίζει νέες λειτουργίες για την κάμερα τις συσκευής, όπως η καταγραφή και παρακολούθηση βίντεο από την λειτουργία της κάμερας και η άμεση μεταφόρτωση του βίντεο αλλά και των φωτογραφιών στο Youtube και το Picasa αντίστοιχα απευθείας από το τηλέφωνο. Έχει νέο έξυπνο πληκτρολόγιο με πρόβλεψη κειμένου. Υποστηρίζει πρότυπο Bluetooth A2DP και AVRCP ενώ έχει και την ικανότητα να συνδέεται αυτόματα σε μικροσυσκευές Bluetooth από μια συγκεκριμένη απόσταση. Ακόμα στην έκδοση αυτή έχει νέο γραφικό περιβάλλον με κινούμενες μεταβάσεις οθόνης.



## Android Donut

Η έκδοση “Donut”, βασισμένη στο Linux Kernel 2.6.29, παρουσιάστηκε στις 15 Σεπτεμβρίου του 2009. Έχει ταχύτερη απόκριση σε σχέση με την προηγούμενη έκδοση. Υποστηρίζεται πλέον η επιλογή πολλαπλών αρχείων ταυτόχρονα, έχει ανανεωμένο γκαλερί και φωτογραφική μηχανή, καθώς και βελτιωμένο Android Market. Έχει ανανεωμένη φωνητική αναζήτηση, με ταχύτερη απόκριση και βαθύτερη ολοκλήρωση με εγγενείς (native) εφαρμογές, συμπεριλαμβανομένης της δυνατότητας κλήσης επαφών. Δυνατότητα αναζήτησης σελιδοδεικτών, ιστορικού, επαφών αλλά και στο διαδίκτυο από την αρχική οθόνη. Υποστήριξη για ανάλυση οθονών WVGA. Ανανεωμένη υποστήριξη τεχνολογιών για CDMA/EVDO, 802.1x, VPNs και με μηχανή μετατροπής κειμένου σε ομιλία (text-to-speech). 11



## Android Éclair

Η έκδοση “Eclair”, βασισμένη και αυτή στον Linux Kernel 2.6.29, παρουσιάστηκε στις 26 Οκτωβρίου του 2009, ενώ τον Ιανουάριο του 2010 επανεκδόθηκε σε Android 2.1 Eclair (MR1). Σε αυτή την έκδοση υπάρχει ακόμα ταχύτερη απόκριση του υλικού σε σχέση με τις δυο προηγούμενες και πλέον υποστηρίζονται περισσότερες οθόνες και αναλύσεις. Υπάρχει νέος browser ο οποίος υποστηρίζει το πρότυπο HTML5, νέο User Interface, και βελτιωμένοι χάρτες Google (Google Maps 3.1.2). Έχει ενσωματωθεί η υποστήριξη φλας για την κάμερα η οποία έχει πλέον και ψηφιακό zoom. Επίσης έχει βελτιωθεί η κλάση MotionEvent ώστε να υπάρχει η δυνατότητα για γεγονότα πολλαπλής αφής (multitouch events). Υποστηρίζεται Bluetooth 2.1 και έχει βελτιωθεί και το πληκτρολόγιο.



## Android Froyo

Η έκδοση “Froyo”, βασισμένη στο Linux Kernel 2.6.32, παρουσιάστηκε στις 20 Μαΐου του 2010. Υπάρχουν βελτιστοποιήσεις στην ταχύτητα γενικά του λειτουργικού συστήματος, στην μνήμη και στην απόδοση. Έχει ενσωματωθεί ο μηχανισμός JavaScript του Chrome V8 στον browser, υπάρχει πλέον Adobe Flash 10.1, ενώ υποστηρίζεται καλύτερα πλέον το Microsoft Exchange. Έχει γίνει ανανέωση του Android Market. Ο χρήστης μπορεί πλέον να ελέγχει αν θα γίνεται ή όχι κίνηση πακέτων δεδομένων από το δίκτυο κινητής τηλεφωνίας. Υπάρχει η δυνατότητα εγκατάστασης εφαρμογών στην κάρτα μνήμης και η μεταφορά τους εκεί από τη μνήμη του τηλεφώνου. Επίσης το τηλέφωνο πλέον μπορεί να μετατραπεί σε WiFi hotspot.



## Android Gingerbread

Η έκδοση “Gingerbread”, βασισμένη στο Linux Kernel 2.6.35.7, παρουσιάστηκε στις 6 Δεκεμβρίου του 2010, ενώ τον Φεβρουάριο του 2011 επανεκδόθηκε σε Android 2.3.3. Στην έκδοση αυτή υπάρχουν αλλαγές στο User Interface το οποίο έχει γίνει πιο απλό και ταχύ, ενώ υποστηρίζονται πλέον οθόνες μεγάλων μεγεθών και αναλύσεων. Υπάρχει πλέον το πρωτόκολλο SIP για κλήσεις μέσω VoIP, υποστηρίζεται ο τύπος βίντεο WebM/VP8 και ο κωδικοποιητής AAC, έχει βελτιωθεί ο ήχος καθώς και οι λειτουργίες απεικόνισης για την ανάπτυξη παιχνιδιών. Υπάρχει η δυνατότητα για Copy-Paste σε όλο το σύστημα και όχι μόνο στην ίδια εφαρμογή. Υποστηρίζεται το NFC (Near Field Communication) και η ύπαρξη πολλαπλών καμερών. Επίσης, έχει βελτιωθεί η ενεργειακή υποστήριξη και έχει γίνει μετάβαση από το σύστημα αρχείων YAFFS στο ext4 στις νέες συσκευές.





## Android Honeycomb

Η έκδοση “Honeycomb”, βασισμένη στο Linux Kernel 2.6.36, παρουσιάστηκε στις 9 Μαΐου του 2011, με την ιδιαιτερότητα ότι προοριζόταν αποκλειστικά για tablets. Οι αλλαγές που έγιναν στην έκδοση αυτή έχουν να κάνουν κυρίως με τη βελτίωση της υποστήριξης των tablets. Υπάρχει ένα νέο, εντελώς διαφορετικό, User Interface και υποστηρίζονται διπύρνηνοι και τετραπύρνηνοι επεξεργαστές. Ακόμα, έχει απλοποιηθεί το multitasking έτσι ώστε ο χρήστης να μπορεί με τη χρήση ενός πλήκτρου (recent apps) να περνάει από μια εφαρμογή σε άλλη. Υπάρχει η δυνατότητα για Video Chat μέσω της εφαρμογής Google Talk καθώς η ανάγνωση βιβλίων μέσω του Google eBooks. Επιπλέον, μπορούν να κρυπτογραφηθούν όλα τα δεδομένα χρήστη.



## Android Ice Cream Sandwich

Η έκδοση “Ice Cream Sandwich”, βασισμένη Ice Cream Sandwich στο Linux Kernel 3.0.1, παρουσιάστηκε στις 19 Οκτωβρίου του 2011. Για άλλη μια φορά έχει βελτιωθεί η ταχύτητα και η απόδοση του συστήματος. Πλέον στο User Interface, το οποίο είναι και παλι διαφορετικό, υπάρχουν εικονικά πλήκτρα τα οποία παίρνουν τη θέση των φυσικών ή αφής που υπήρχαν στις συσκευές. Βελτίωση της ασφάλεια του συστήματος με την προσθήκη αναγνώρισης προσώπου για να ξεκλειδώσει η συσκευή. Ο browser μπορεί να ανοίξει ταυτόχρονα μέχρι και 16 καρτέλες. Υπάρχει η δυνατότητα ο χρήστης να τερματίσει εφαρμογές οι οποίες τρέχουν στο background, ενώ μπορεί να θέσει και όρια στην κίνηση πακέτων δεδομένων. Η εφαρμογή Android Beam αξιοποιεί πλέον το NFC αφού επιτρέπει την αποστολή δεδομένων από τη συσκευή σε όσες βρίσκονται εντός μιας μικρής ακτίνας εμβέλειας. Ακόμα με την ύπαρξη του Wi-Fi Direct συσκευές μπορούν να συνδεθούν μεταξύ τους ασύρματα χωρίς την μεσολάβηση κάποιου access point. Τέλος, υποστηρίζεται η εγγραφή βίντεο σε 1080p.



## 5.2 Εισαγωγή στο Περιβάλλον Ανάπτυξης

Ξεκινώντας από μηδενική βάση, το πρώτο βήμα που έχουμε να πραγματοποιήσουμε είναι να εγκαταστήσουμε στον υπολογιστή μας όλα τα απαραίτητα προγράμματα που θα χρειαστούμε για να αναπτύξουμε τις Android εφαρμογές. Το περιβάλλον ανάπτυξης Android εφαρμογών στηρίζεται σε τρία βασικά εργαλεία, τα οποία και τα τρία είναι δωρεάν: Java, IDE και SDK. Τα συγκεκριμένα εργαλεία πρέπει να εγκατασταθούν με την σειρά με την οποία τα αναφέραμε γιατί το κάθε ένα έχει σαν προϋπόθεση ότι το προηγούμενο ήδη έχει εγκατασταθεί στο σύστημα. Σαν προϋπόθεση για την καλύτερη κατανόηση ανάπτυξης εφαρμογών σε πλατφόρμα Android, θεωρούνται κάποιες βασικές γνώσεις στην γλώσσα προγραμματισμού Java.

Για να εκτελεστεί η εφαρμογή, το Android χρησιμοποιεί έναν εξομοιωτή (emulator) ο οποίος προσομοιώνει μια Android κινητή συσκευή. Από την πλευρά του προγραμματιστή, αυτό είναι ένα πολύ καλό και θετικό στοιχείο γιατί μπορεί να αναπτύσσει Android εφαρμογές χωρίς να διαθέτει Android συσκευή αλλά ταυτόχρονα να δοκιμάζει τις εφαρμογές αν τρέχουν σωστά σε Android περιβάλλον.

Ο Android εξομοιωτής μπορεί να μιμηθεί αρκετές Android συσκευές, διαφόρων εκδόσεων, και ονομάζεται Dalvik Virtual Machine (DVM). Το DVM είναι μια εικονική μηχανή που επιτρέπει την μέγιστη δυνατή απόδοση της εφαρμογής με την σωστή κατανομή πόρων στην συσκευή αφού είναι ο ίδιος που χρησιμοποιείται και εσωτερικά στην αληθινή συσκευή. Η κάθε εφαρμογή εκτελείται μέσω τις δικής της εικονικής μηχανής στη δικιά της διεργασία και για αυτό το λόγο καμία εφαρμογή δεν έχει επαφή με την άλλη, ενώ εκτελούνται ταυτόχρονα.

Κάθε φορά που μια καινούργια εφαρμογή ξεκινάει, το DVM δημιουργεί ένα αντίγραφο του εαυτού του για να αντιπροσωπεύσει την συγκεκριμένη εφαρμογή με όλους τους διαθέσιμους πόρους του συστήματος. Από εκεί και πέρα το DVM παίρνει όλες τις Java εντολές της εφαρμογής μαζί με τον XML κώδικα και οτιδήποτε άλλα αρχεία εικόνας και ήχου που ίσως να υπάρχουν, και τα μεταφράζει σε δυαδικό κώδικα μηχανής (bytecode) για τον καταλάβει η Android πλατφόρμα και να το εκτελέσει.

Πριν ξεκινήσουμε να βλέπουμε πώς να δημιουργήσουμε ένα καινούργιο project πρέπει πρώτα να έχουμε εγκαταστήσει φυσικά το Android SDK. Απαραίτητες προϋποθέσεις είναι η ύπαρξη του πιο



πρόσφατου JDK, το οποίο μπορεί κανείς να κατεβάσει από την εξής ιστοσελίδα:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>  
καθώς και το IDE το οποίο μπορεί να βρεθεί στον σύνδεσμο <http://www.eclipse.org/downloads/>. Μετά την εγκατάσταση του JDK και IDE επόμενο και κύριο βήμα είναι η εγκατάσταση του Android SDK. Από τον σύνδεσμο <http://developer.android.com/sdk/index.html> μπορεί κανείς να κατεβάσει την κατάλληλη για τον υπολογιστή του έκδοση και στη συνέχεια να ξεκινήσει την εγκατάσταση των εργαλείων που περιλαμβάνει. Τέλος, χρήσιμο εργαλείο για την εύκολη δημιουργία εφαρμογών είναι το ADT Plugin. Για την εγκατάσταση αυτού του εργαλείου χρειάζεται ο χρήστης μέσα από το περιβάλλον του Eclipse να κάνει την εξής διαδικασία: να επιλέξει από το μενού το Help και Install New Software. Από το παράθυρο που θα εμφανιστεί να επιλέξει το κουμπί Add και να προσθέσει το ADT plugin τοποθετώντας στο πεδίο Location την διεύθυνση <http://dl-ssl.google.com/android/eclipse/> και να ολοκληρώσει την εγκατάσταση αποδέχοντας τους όρους.

### 5.3 Δημιουργία ενός Android Project με το Eclipse

Για να δημιουργήσουμε λοιπόν ένα νέο project ακολουθούμε τα ακόλουθα βήματα από το Eclipse:

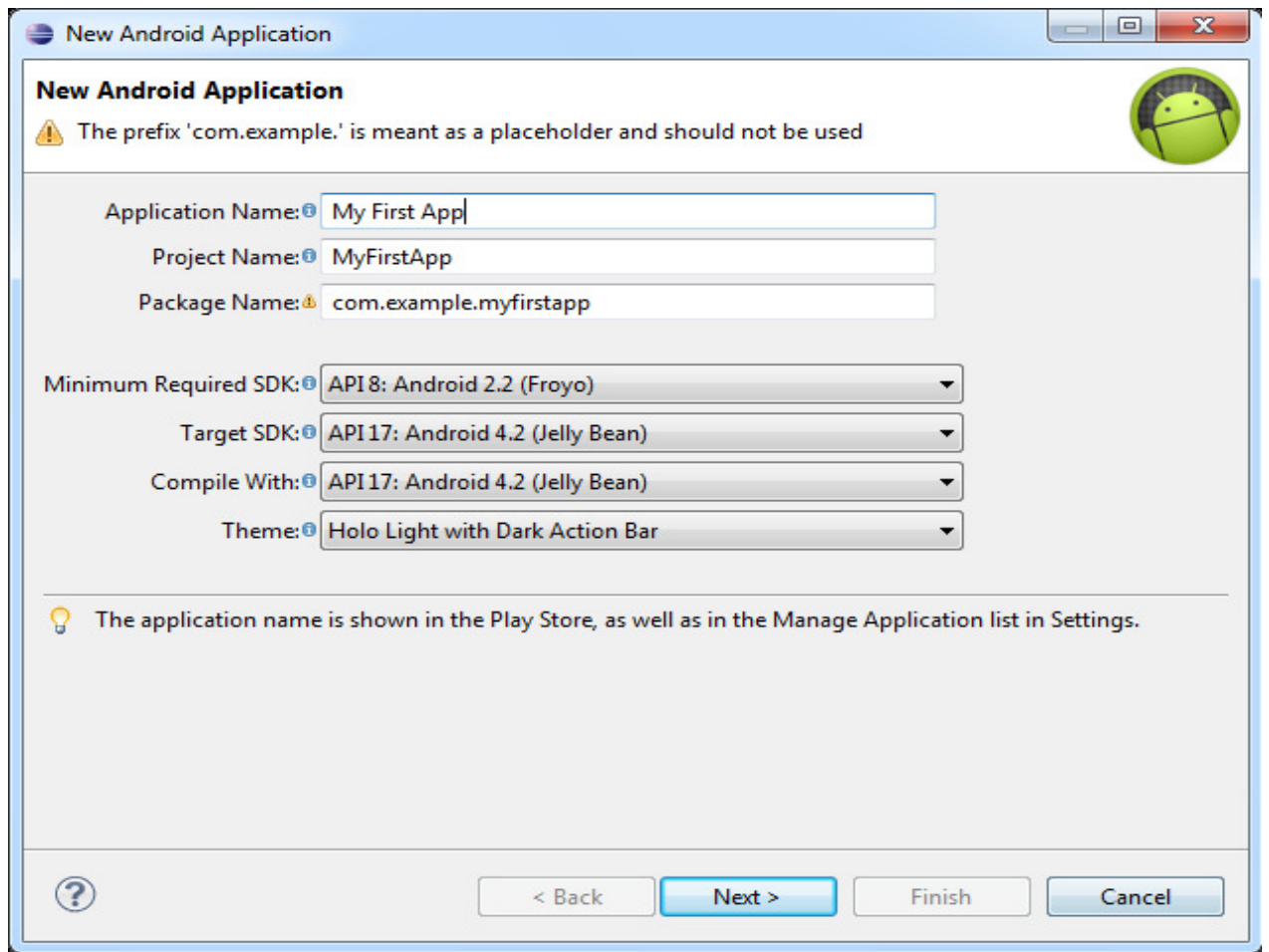
1. Επιλέγουμε File, New, Android Project ή πατάμε το εικονίδιο δημιουργίας νέων project στην μπάρα εργαλείων του Eclipse, ανοίγουμε το φάκελο Android και επιλέγουμε Android Application.

2. Συμπληρώνουμε τη φόρμα που εμφανίζεται:

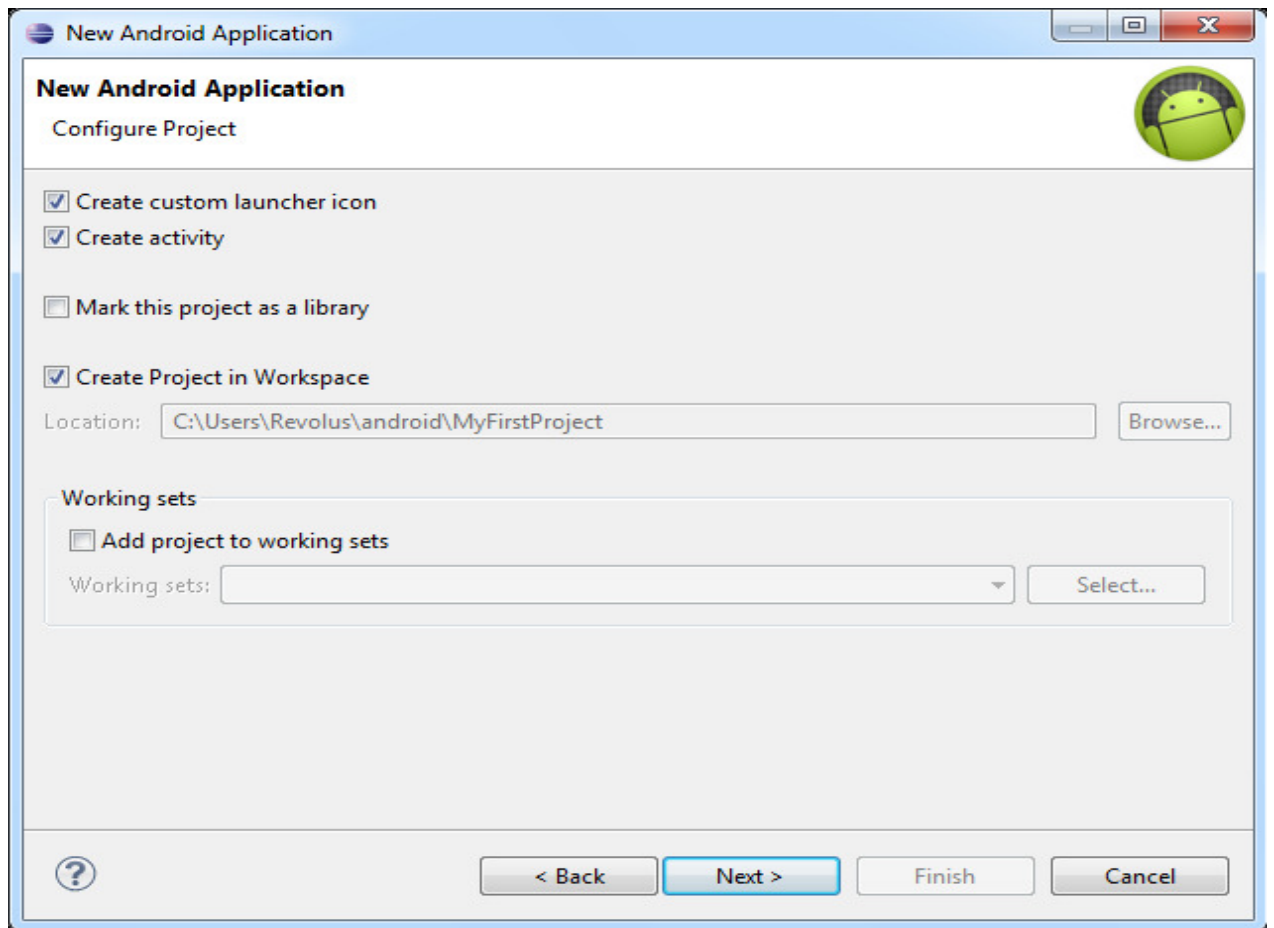
- ✓ Το πεδίο Application Name είναι το όνομα της εφαρμογής που θα εμφανίζεται στους χρήστες. Για το παράδειγμα αυτό, θα χρησιμοποιήσουμε το “My First App”.
- ✓ Το πεδίο Project Name είναι το όνομα για το directory του project και το όνομα που είναι ορατό στο Eclipse.
- ✓ Το πεδίο Package Name είναι ο ονοματοχώρος των κλάσεων της εφαρμογής. Ακολουθεί τους ίδιους κανόνες με τα πακέτα στη Java. Το όνομα του πακέτου πρέπει να είναι μοναδικό ανάμεσα σε όλα τα πακέτα που είναι εγκατεστημένα στο σύστημα. Για το λόγο αυτό είναι προτιμότερο να χρησιμοποιούμε ένα όνομα που ξεκινά με το αντίστροφο domain name του οργανισμού ή της ιστοσελίδας μας. Για το project αυτό μπορούμε να χρησιμοποιήσουμε το “com.example.myfirstapp” που συμπλήρωσε αυτόματα το Eclipse,

ωστόσο αν θέλουμε να δημοσιεύσουμε την εφαρμογή αυτή στο Google Play θα πρέπει να δώσουμε στο πακέτο ένα πραγματικό όνομα.

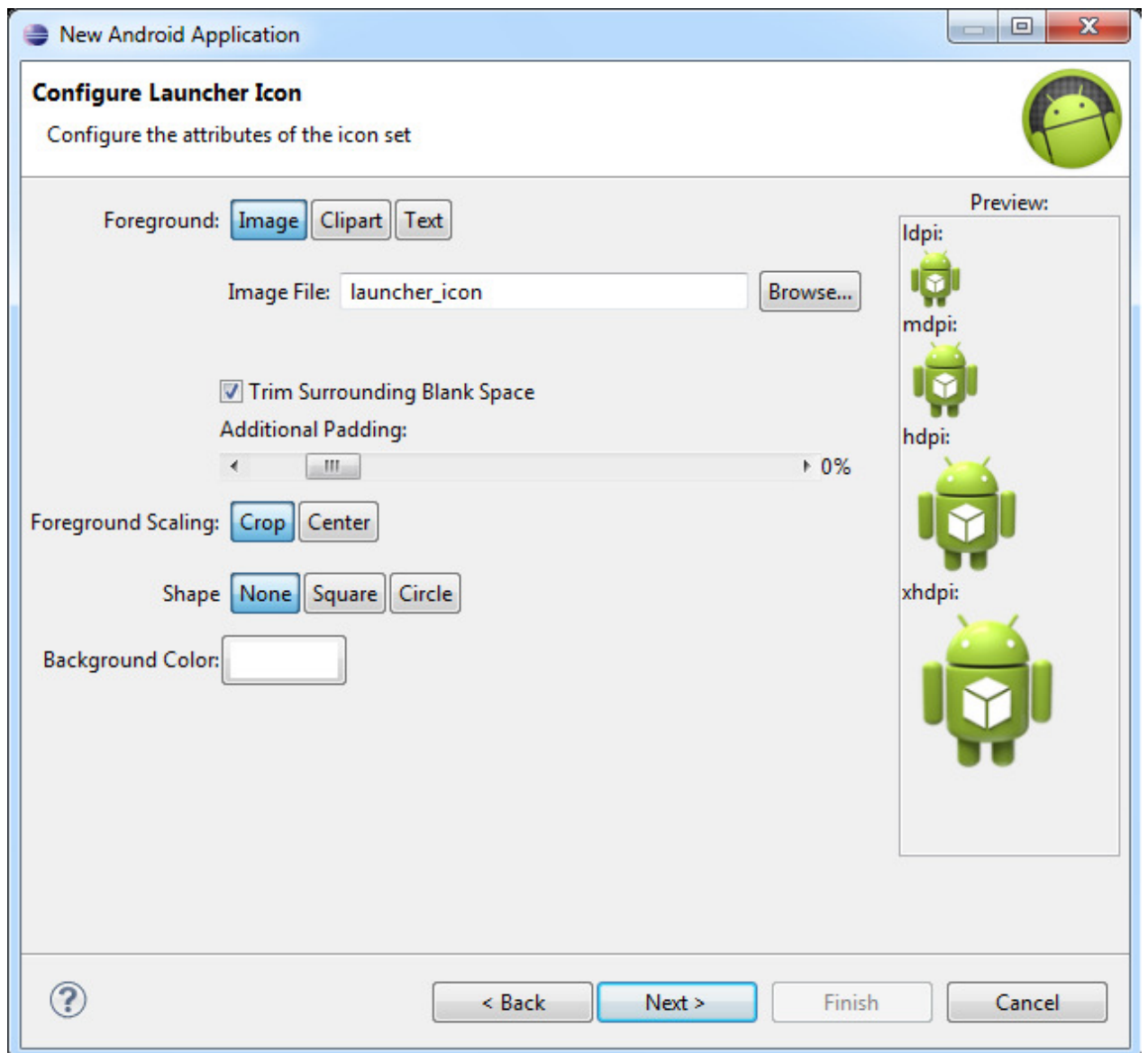
- ✓ Το `minimum Required SDK` είναι η χαμηλότερη έκδοση του Android που υποστηρίζει η εφαρμογή μας και μπορεί να λάβει σαν τιμή κάποιο από τα `API Levels`. Για να υποστηρίξουμε όσο το δυνατό περισσότερες συσκευές, θα πρέπει να ορίσουμε το μικρότερο `API Level` που υποστηρίζει τις δυνατότητες της εφαρμογής μας. Αν οποιοδήποτε χαρακτηριστικό της εφαρμογής μας υποστηρίζεται μόνο σε κάποιο από τα πιο πρόσφατα `levels` και δεν είναι σημαντικό για την όλη εφαρμογή, μπορούμε να το ενεργοποιούμε μόνο στις εκδόσεις που το υποστηρίζουν.
- ✓ Το πεδίο `Target SDK` ορίζει το υψηλότερο επίπεδο με το οποίο δοκιμάσαμε την εφαρμογή μας. Καθώς διατίθενται οι νέες εκδόσεις του Android, θα πρέπει να δοκιμάζουμε την εφαρμογή στη νέα έκδοση και να ενημερώσουμε αυτή την τιμή για να ταυτίζεται με την τελευταία έκδοση ή να εκμεταλλεύεται τα νέα χαρακτηριστικά της πλατφόρμας.
- ✓ Το πεδίο `Compile With` είναι η έκδοση της πλατφόρμας με βάση την οποία πρόκειται η εφαρμογή μας να μεταγλωττιστεί της οποίας μεταγλωττίζουμε την εφαρμογή. Προεπιλεγμένα τίθεται στην τελευταία διαθέσιμη έκδοση του SDK. Αν δεν υπάρχει κάποια έκδοση κατεβασμένη, χρειάζεται να κατεβάσουμε.
- ✓ Το πεδίο `Theme` ορίζει το στυλ του `user interface` του Android που θα εφαρμοστεί στο app.



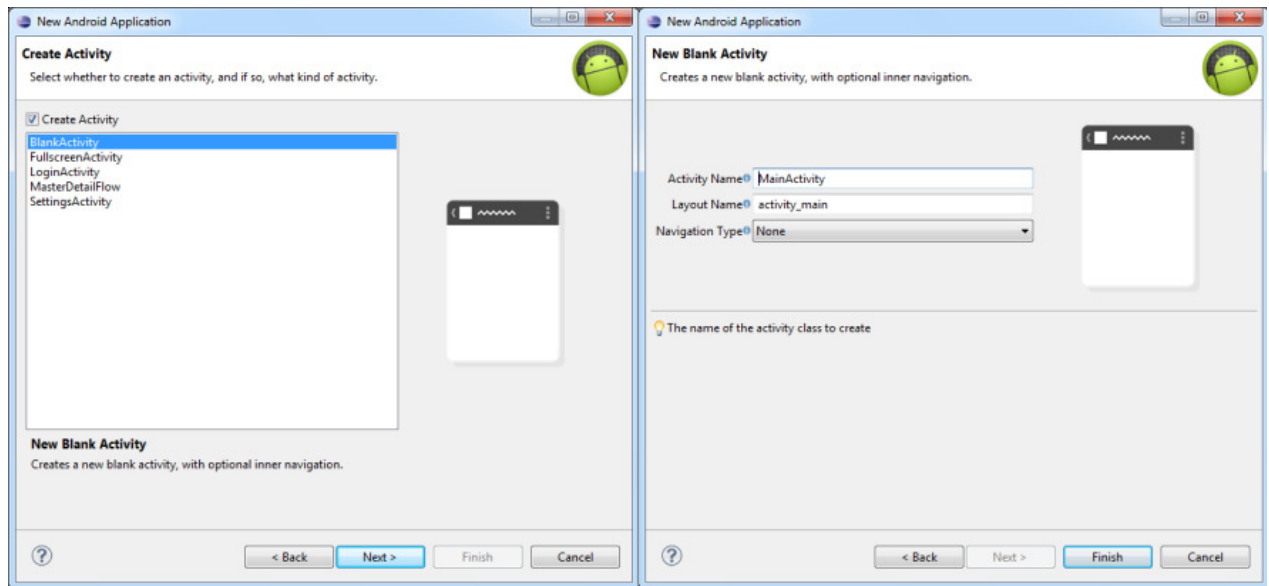
3. Στο επόμενο παράθυρο με τις ρυθμίσεις, αφήνουμε τις προεπιλεγμένες τιμές και πατάμε Next.



4. Το επόμενο παράθυρο μας βοηθάει να δημιουργήσουμε ένα εικονίδιο εκκίνησης της εφαρμογής μας. Μπορούμε να προσαρμόσουμε το εικονίδιο ανάλογα με τις απαιτήσεις μας, ενώ το εργαλείο παράγει ένα εικονίδιο για διάφορες πυκνότητες οθονών. Πριν όμως δημοσιεύσουμε την εφαρμογή μας θα πρέπει να βεβαιωθούμε ότι το εικονίδιο τηρεί τους κανόνες που ορίζονται στον οδηγό σχεδίασης εικονιδίων<sup>1</sup>. Στη συνέχεια πατάμε Next.



5. Τώρα μπορούμε να επιλέξουμε ένα activity template από το οποίο μπορούμε να ξεκινήσουμε την ανάπτυξη της εφαρμογής μας.
6. Αφήνουμε όλες τις λεπτομέρειες του Activity στις προεπιλεγμένες τιμές τους και πατάμε Finish.

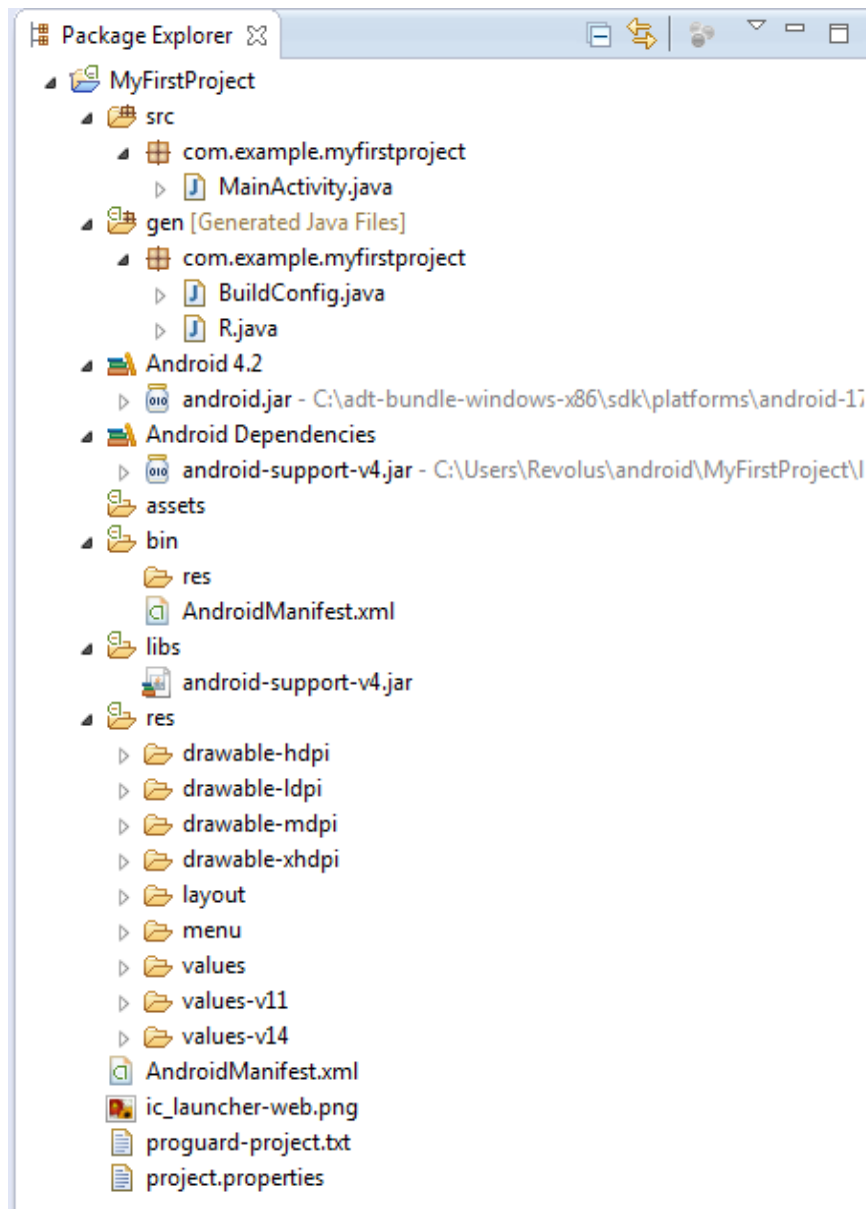


Το project είναι έτοιμο με τα προεπιλεγμένα αρχεία δημιουργημένα. Είμαστε σε θέση να ξεκινήσουμε την ανάπτυξη της εφαρμογής.

## 5.4 Η αρχειακή δομή ενός android project

Παρόλο που το μέγεθος και η πολυπλοκότητα ενός android project ποικίλλει, η δομή τους είναι κοινή. Στην εικόνα 4-1 μπορούμε να δούμε τη δομή των αρχείων του project που μόλις δημιουργήσαμε. Οι android εφαρμογές έχουν μερικά απαραίτητα και μερικά προαιρετικά στοιχεία.

Τα στοιχεία αυτά φαίνονται στον πίνακα 4-1. Όπως μπορούμε να δούμε στον πίνακα αυτόν, μια εφαρμογή αποτελείται κυρίως από τρία κομμάτια: α) το κομμάτι που περιγράφει την εφαρμογή, β) μια συλλογή από διάφορους πόρους και γ) τον πηγαίο κώδικα της εφαρμογής. Αν εξαιρέσουμε το αρχείο `AndroidManifest.xml` μπορούμε να δούμε μια εφαρμογή Android πολύ απλά σαν την λογική της εφαρμογής μας (υλοποιημένη μέσα στον κώδικα) και τους πόρους. Η βασική δομή μιας android εφαρμογής μοιάζει λίγο με τη βασική δομή μιας J2EE εφαρμογής, όπου οι πόροι αντιστοιχεί στις JSPs, η λογική της εφαρμογής αντιστοιχεί στα servlets και το `AndroidManifest.xml` με το αρχείο `web.xml`.



Εικόνα : Η αρχειακή δομή της εφαρμογής MyFirstProject

Μερικές παρατηρήσεις που πρέπει να κάνουμε για την αρχειακή δομή ενός project είναι οι εξής: Το Android υποστηρίζει μόνο μια γραμμική λίστα αρχείων μέσα στους προκαθορισμένους φακέλους κάτω από το φάκελο res. Για παράδειγμα, δεν υποστηρίζει φωλιασμένους φάκελος κάτω από το φάκελο layout (ή άλλους φακέλους κάτω από τον res).

Επίσης υπάρχουν διάφορες ομοιότητες μεταξύ του φακέλου assets και του φακέλου raw κάτω από το res. Και οι δυο φάκελοι περιέχουν 'raw' αρχεία, ωστόσο τα αρχεία στο φάκελο raw θεωρούνται πόροι σε αντίθεση με τα αρχεία στο φάκελο assets που δεν θεωρούνται. Κατά συνέπεια τα αρχεία στο raw μπορούν να προσπελαστούν με IDs πόρων κλπ. Αλλά τα περιεχόμενα του φακέλου assets θεωρούνται γενικού σκοπού και χρησιμοποιούνται χωρίς περιορισμούς και μπορούμε να εισάγουμε μια οποιαδήποτε άλλη αρχειακή δομή επιθυμούμε.

## 5.5 Το αρχείο Android Manifest

Κάθε Android project περιλαμβάνει ένα αρχείο manifest, το AndroidManifest.xml, που αποθηκεύεται στην κορυφή της ιεραρχίας των αρχείων στο project. Το manifest ορίζει τη δομή και τα μεταδεδομένα της εφαρμογής, τα συστατικά της και τις απαιτήσεις της.

Περιλαμβάνει κόμβους για κάθε μια από τις Activities, Services, Content Providers και Broadcast Receivers που αποτελούν μια εφαρμογή και χρησιμοποιώντας τα Intent Filters και τα Permissions καθορίζει πως θα αλληλεπιδρούν μεταξύ τους και με άλλες εφαρμογές.

Το manifest μπορεί επίσης να καθορίσει τα μεταδεδομένα της εφαρμογής (όπως τα icons, τον αριθμό της έκδοσης ή το theme) και επιπρόσθετους κόμβους στο top-level που ορίζουν τα απαιτούμενα permissions, τα unit tests ενώ ορίζονται και τις απαιτήσεις ως προς το υλικό, την οθόνη και την πλατφόρμα.

Το manifest αποτελείται από το root tag manifest το οποίο περιέχει ένα package attribute που ορίζεται στο πακέτο του project. Περιλαμβάνει επίσης και ένα xmlns:android attribute που παρέχει διάφορα attributes συστήματος που χρησιμοποιούνται μέσα στο αρχείο.

Χρησιμοποιούμε το attribute versionCode για να ορίσουμε την τρέχουσα έκδοση της εφαρμογής σαν έναν ακέραιο που αυξάνει με κάθε σημαντική αλλαγή στην εφαρμογή και χρησιμοποιούμε το versionName attribute για να καθορίσουμε το όνομα της έκδοσης που θα είναι ορατή στους χρήστες.

Μπορούμε επίσης να ορίσουμε αν θα επιτρέπεται ή αν είναι επιθυμητό να εγκαθίσταται η εφαρμογή σε εξωτερικό αποθηκευτικό μέσο (συνήθως μια κάρτα SD) αντί μιας εσωτερικής πηγής χρησιμοποιώντας το attribute installLocation, θέτοντας την τιμή preferExternal ή auto, όπου η πρώτη εγκαθιστά την εφαρμογή σε εξωτερικό μέσο όποτε είναι δυνατό και η δεύτερη αφήνει την απόφαση στο σύστημα. Αν δεν ορίσουμε το attribute αυτό, η εφαρμογή θα εγκατασταθεί στην εσωτερική μνήμη και οι χρήστες δεν θα μπορούν να τη μεταφέρουν σε κάποια εξωτερική. Επειδή η εσωτερική μνήμη είναι περιορισμένη, είναι καλό – όποτε αυτό είναι δυνατό – να εγκαθιστούμε την εφαρμογή στην εξωτερική μνήμη.

Ένα από τα πιο σημαντικά elements του manifest είναι το <uses-sdk>, το οποίο ορίζει τη συμβατότητα της εφαρμογής με διάφορες εκδόσεις του Android χρησιμοποιώντας τα attributes android:minSdkVersion και android:targetSdkVersion. Για την εφαρμογή μας αυτά μοιάζουν με τον ακόλουθο κώδικα:



```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="tei.project.bt1"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="14"
        android:targetSdkVersion="16" />

    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="tei.project.bt1.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"
            />
            </intent-filter>
        </activity>
        <activity android:name=".ShowDetails"></activity>
    </application>

</manifest>

```

# Κεφάλαιο 6<sup>ο</sup>

## 6. Κώδικας android

### 6.1 Βιβλιοθήκες-Κατηγορίες βιβλιοθηκών

Σε αυτήν την ενότητα θα μελετήσουμε τις κατηγορίες βιβλιοθηκών που χρησιμοποιήθηκαν στην εργασία καθώς και σημαντικές επεκτάσεις αυτών που αξίζει να σημειωθούν.

Συνολικά οι κατηγορίες βιβλιοθηκών που χρησιμοποιήθηκαν στην παρούσα εργασία

είναι οι εξής:

- **android.app** – περιέχει κλάσεις υψηλού επιπέδου ενσωματώνοντας το ολικό μοντέλο της εφαρμογής
- **android.os** – παρέχει βασικές υπηρεσίες του λειτουργικού συστήματος
- **android.view** – περιέχει κλάσεις διαχείρισης της οθόνης και αλληλεπίδρασης με τον χρήστη
- **android.widget** – περιέχει UI στοιχεία που χρησιμοποιούνται στην οθόνη της εφαρμογής
- **android.content** – περιέχει κλάσεις για την πρόσβαση και δημοσιοποίηση δεδομένων στην συσκευή
- **java.net**
- **java.util**
- **javax.xml.parsers** – περιέχει κλάσεις που επιτρέπουν την διαχείριση XML δεδομένων
- **org.xml.sax** – παρέχει τον πυρήνα των SAX APIs

Σημειώνουμε, ότι οι τέσσερις πρώτες βιβλιοθήκες είναι οι βασικές βιβλιοθήκες οι

οποίες χρησιμοποιούνται σε κάθε εφαρμογή Android.

Ενώ, οι υπόλοιπες βιβλιοθήκες οι οποίες χρησιμοποιήθηκαν

για τον συγκεκριμένο σκοπό της εργασίας, το rss, αναλύονται παρακάτω:  
android.content

- **android.app και android.os**

Ανοίγοντας την πρώτη κλάση MainActivity την οποία έχουμε ορίσει κατά την δημιουργία του project αντικρίζουμε τις παρακάτω γραμμές κώδικα:

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //requestWindowFeature(Window.FEATURE_NO_TITLE);
        setContentView(R.layout.activity_main_drawer);
```

Στις γραμμές αυτές βλέπουμε την κλάση MainActivity στην οποία γίνεται extend η κλάση Activity (ή ListActivity σε περίπτωση που θέλουμε να συμπεριλάβουμε πίνακα). Η διαδικασία αυτή γίνεται επιτρεπτή μόνο με την χρήση της βιβλιοθήκης android.app. Στην συνέχεια βλέπουμε την μέθοδο onCreate η οποία για την συγγραφή της απαιτεί την ύπαρξη της βιβλιοθήκης android.os. Στην μέθοδο αυτή ορίζεται η διάταξη που θα ακολουθεί η κλάση σύμφωνα με τα αρχεία xml που δημιουργούμε στον φάκελο layout.

- **android.widget**

Η βιβλιοθήκη αυτή δεν αποτελεί μία από τις βιβλιοθήκες που συμπεριλαμβάνονται στην εφαρμογή κατά την εκκίνηση της αλλά αποτελεί εξίσου σημαντική και απαραίτητη βιβλιοθήκη για οποιοδήποτε κανούργια εφαρμογή.

Η βιβλιοθήκη android.widget περιέχει UI στοιχεία που χρησιμοποιούνται στην οθόνη της εφαρμογής. Τα στοιχεία UI, είναι στοιχεία που χρησιμοποιούνται για τη γραφική διεπαφή της εφαρμογής, όπως κείμενα, κουμπιά, λίστες, ετικέτες κ.ά. Άξια αναφοράς είναι η δυνατότητα που μας δίνει να δημιουργήσουμε και δικά μας widget κάνοντας κάποιες επιπρόσθετες λειτουργίες και αναφορές.

- **android.content**

Η βιβλιοθήκη android.content περιέχει κλάσεις για την πρόσβαση και δημοσιοποίηση δεδομένων στην συσκευή.

Στην παρούσα εφαρμογή χρησιμοποιήθηκε κυρίως η βιβλιοθήκη android.content.Intent, υπεύθυνη για λειτουργίες/δραστηριότητες που πρέπει να εκτελεστούν. Στην εργασία χρησιμοποιήθηκε σε πολλά σημεία στα οποία χρειάστηκε να μεταβούμε από μια κλάση στην άλλη ή να μεταφέρουμε κάποια δεδομένα μεταξύ των κλάσεων.

Για παράδειγμα, μπορεί σε μία εφαρμογή να θέλουμε να ορίσουμε ότι με το πάτημα ενός κουμπιού θέλουμε από την παρούσα κλάση να μεταβούμε σε μία άλλη η οποία θα αναλαμβάνει την διαδικασία που θα εκτελείται από το πάτημα του 41 κουμπιού. Η μετάβαση από μια κλάση FirstClass σε μια κλάση SecondClass γίνεται με την χρήση των εξής εντολών:

```
Intent intent = new Intent(FirstClass.this, SecondClass.class);  
startActivity(intent);
```

Ορίζουμε δηλαδή μία μεταβλητή τύπου Intent στην οποία καθορίζουμε ότι θέλουμε να μεταβούμε από την παρούσα κλάση (FirstClass.this) στην οποία βρισκόμαστε στην κλάση που ακολουθεί (SecondClass.class). Και στην συνέχεια ορίζουμε την έναρξη αυτής της ενέργειας.

Στην δεύτερη περίπτωση, η μεταφορά δεδομένων μεταξύ δύο κλάσεων γίνεται λίγο πιο περίπλοκη καθώς οι σειρές κώδικα που απαιτούνται είναι περισσότερες όπως επίσης και η προσοχή που χρειάζεται και για συγγραφή τους. Όπως βλέπουμε και στο ακόλουθο παράδειγμα απαραίτητη για την μεταφορά δεδομένων μεταξύ κλάσεων είναι και η χρήση της βιβλιοθήκης android.os.Bundle. Στο παράδειγμα αυτό θέλουμε να μεταφέρουμε κάποια δεδομένα (data) από την κλάση στην οποία βρισκόμαστε (FirstClass.this) στην κλάση που ακολουθεί (SecondClass.class). Για τον λόγο αυτό χρησιμοποιούμε Bundle. Το Bundle μπορούμε να το παρομοιάσουμε για χάριν ευκολίας με ένα δοχείο στο οποίο τοποθετούμε τα δεδομένα που θέλουμε να στείλουμε στη κλάση. Για να μπορούν όμως τα δεδομένα αυτά να ανιχνευτούν από την κλάση ορίζουμε ένα κλειδί όπου και πάλι για λόγους ευκολίας το ονομάζουμε “key” (δεύτερη σειρά κώδικα). Στην συνέχεια ορίζουμε το Intent με τον ίδιο τρόπο που είδαμε πιο πάνω και αμέσως μετά τοποθετούμε σε αυτό το δοχείο το οποίο θέλουμε να μεταφέρει. Και τέλος ξεκινά το γεγονός. Αφού, λοιπόν, έχουμε στείλει τα δεδομένα πρέπει να κάνουμε τις απαραίτητες λειτουργίες ώστε η δεύτερη κλάση να μπορεί να τα δεχτεί.

Έτσι ορίζουμε και στην δεύτερη κλάση ένα Intent το οποίο όπως βλέπουμε δέχεται πληροφορίες (getIntent()) και ένα Bundle το οποίο θα μπορεί να δεχτεί το “δοχείο” το οποίο στείλαμε. Και τέλος ορίζουμε την μεταβλητή στην οποία θα αποθηκεύσουμε τα δεδομένα που στείλαμε (που στην περίπτωση αυτή είναι τύπου String) και την ενημερώνουμε να δεχτεί τα δεδομένα τα οποία έχουμε ορίσει με το κλειδί “key”.

```
Αποστολή δεδομένων από την πρώτη κλάση  
Bundle b = new Bundle();
```

```
b.putString("key", data);
ourIntent = new Intent(FirstClass.this, SecondClass.class);
ourIntent.putExtras(b);
startActivity(ourIntent);
```

Λήψη δεδομένων από την δεύτερη κλάση

```
Intent i = getIntent();
Bundle b = i.getExtras();
String a = b.getString("key");
java.net
```

- **java.net**

Η βιβλιοθήκη `java.net` είναι και αυτή μία από τις βιβλιοθήκες που μας βοηθούν να διαχειριστούμε στοιχεία του Διαδικτύου. Δεδομένα τα οποία χρειάζεται να αποθηκευτούν, να σταλούν, να ληφθούν κ.ά. Εμείς χρησιμοποιήσαμε αυτή την βιβλιοθήκη για να ορίσουμε ότι τα δεδομένα τα οποία επρόκειτο να λάβουμε είναι αυτού του τύπου

- **java.util**

Η βιβλιοθήκη αυτή προσφέρει κλάσεις οι οποίες μας βοηθούν να διαχειριστούμε πίνακες, λίστες, συνδεδεμένες λίστες, χάρτες, ουρές, στοίβες, χρονικά περιθώρια ή ακόμα και ημερομηνίες και ημερολόγια. Στην παρούσα εργασία χρησιμοποιήθηκε για την διαχείριση και επεξεργασία λιστών δικού μας τύπου.

- **javax.xml.parsers και org.xml.sax**

Η πρώτη βιβλιοθήκη περιέχει κλάσεις που επιτρέπουν την διαχείριση XML δεδομένων, ενώ η δεύτερη παρέχει τον πυρήνα των SAX APIs. Στην εργασία αυτή χρειαστήκαμε τις κλάσεις αυτών των κατηγοριών για την κύρια λειτουργία της εργασίας, την λήψη δεδομένων από το Διαδίκτυο. Ο κώδικας ο οποίος πραγματοποιεί τη λειτουργία αυτή είναι ο ακόλουθος:

```
URL url = new URL(urlToRssFeed);
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParser parser = factory.newSAXParser();
XMLReader xmlreader = parser.getXMLReader();
RSSParser theRssHandler = new RSSParser(list);
xmlreader.setContentHandler(theRssHandler);
```

```
InputSource is = new InputSource(url.openStream());  
xmlreader.parse(is);
```

## **6.2 MainActivity**

```
1 package tei.project.bt1;  
2 import java.net.URL;  
3 import java.util.Vector;  
4 import javax.xml.parsers.SAXParser;  
5 import javax.xml.parsers.SAXParserFactory;  
6 import org.xml.sax.InputSource;  
7 import org.xml.sax.XMLReader;  
10 import android.opengl.Visibility;  
11 import android.os.AsyncTask;  
12 import android.os.Bundle;  
13 import android.app.Activity;  
14 import android.app.SearchManager.OnDismissListener;  
15 import android.content.Intent;  
16 import android.content.pm.ActivityInfo;  
17 import android.content.res.Configuration;  
18 import android.support.v4.app.ActionBarDrawerToggle;  
19 import android.support.v4.view.GravityCompat;  
20 import android.support.v4.widget.DrawerLayout;  
21 import android.view.Menu;  
22 import android.view.MenuInflater;  
23 import android.view.MenuItem;  
24 import android.view.View;  
25 import android.view.View.OnClickListener;  
26 import android.widget.AdapterView;  
27 import android.widget.AdapterView.OnItemClickListener;  
28 import android.widget.ListView;  
29 import android.widget.RadioButton;  
30 import android.widget.Toast;  
  
31 public class MainActivity extends Activity implements  
OnItemClickListener {  
32 public static final String URL_BASE = "http://www.actuspro.gr";
```

```

33 public static final String RSSFEED_BASE = URL_BASE +
34 "/_ktel/routes.php?routemenu=";
35 public static final String IMGURL_BASE = URL_BASE + "/logos/";
36 public static final String DETAIL_BASE = URL_BASE +
37 "/?page=newsdetail&DocID=";
38 ListView list;
39 LazyAdapter adapter;
40 Vector<RSSItem> rssitems;
41 private RSSFeed feed = null;
42 private DownloadNewsTask downloadNewsTask;
43 private int feedNum;
44 private RadioButton rbToday;
45 private RadioButton rbTomorrow;
46 private int routemenu;
// NAVIGATION DRAWER MENU
47 private ListView mDrawerList;
48 private DrawerLayout mDrawer;
49 private CustomActionBarDrawerToggle mDrawerToggle;
50 private String[] menuItems;
51 private String[] menuItems2;
52 static int itemPos = 0;
53 @Override
54 protected void onCreate(Bundle savedInstanceState) {
55 super.onCreate(savedInstanceState);
56 setContentView(R.layout.activity_main_drawer);
57 rbToday = (RadioButton) findViewById(R.id.rbToday);
58 rbToday.setOnClickListener(new OnClickListener() {
59 @Override
60 public void onClick(View v) {
61 Toast.makeText(MainActivity.this, "Έπισημότητα επίσημοτητας",
62 Toast.LENGTH_LONG).show();
63 updateFeed(feedNum);
64 });
65 rbTomorrow = (RadioButton) findViewById(R.id.rbTomorrow);
66 rbTomorrow.setOnClickListener(new OnClickListener() {
67 @Override
68 public void onClick(View v) {
69 Toast.makeText(MainActivity.this, "Έπισημότητα επίσημοτητας",
70 Toast.LENGTH_LONG).show();
71 updateFeed(feedNum);
72 });
ActionBar εικονίδιο ενεργοποιεί τον drawer
// NAVIGATION DRAWER MENU PART

```

```

// enable ActionBar app icon to behave as action to toggle nav drawer
73 getActionBar().setDisplayHomeAsUpEnabled(true);
74 getActionBar().setHomeButtonEnabled(true);
75 mDrawer = (DrawerLayout) findViewById(R.id.drawer_layout);
επικαλύπτει με σκιά το κύριο περιεχόμενο
// set a custom shadow that overlays the main content when the drawer
// opens
76 mDrawer.setDrawerShadow(R.drawable.drawer_shadow,
77 GravityCompat.START);_initMenu();
78 mDrawerToggle = new CustomActionBarDrawerToggle(this,
mDrawer);
79 mDrawer.setDrawerListener(mDrawerToggle);
80 list = (ListView) findViewById(R.id.lvRoutes);
81 if ((getLastNonConfigurationInstance() != null)){
82 RotateInfo ri = (RotateInfo) getLastNonConfigurationInstance();
83 feed = ri.feed;
84 rssitems = ri.rssItems;
85 adapter = ri.adapter;
86 list.setAdapter(adapter);
87 list.setOnItemClickListener(this);
88 if (feed==null){
89 list.setAdapter(null);
90 try{
91 downloadNewsTask.cancel(true);
92 }catch(Exception e){
// if not executing or not exist yet will cause exception. Just ignore it! ;)
93 }}}
94 feedNum = 4; // the default feed to first load (Patra - Athens)
95 updateFeed(feedNum);
96}
97 public class RotateInfo {
98 LazyAdapter adapter;
99 Vector<RSSItem> rssItems;
100 String displayingTitle;
101 RSSFeed feed;
102 }
103 @Override
104 public RotateInfo onRetainNonConfigurationInstance()
105 {
106 RotateInfo ri = new RotateInfo();
107 ri.adapter = adapter;
108 ri.rssItems = rssitems;

```



```

109 ri.displayingTitle = "some title";//((TextView)
110 findViewById(R.id.feedtitle)).getText().toString();
111 ri.feed = feed;
112 return ri;
113 }
114 @Override
115 public boolean onCreateOptionsMenu(Menu menu) {
116     MenuInflater inflater = getMenuInflater();
117     inflater.inflate(R.menu.main, menu);
118     return super.onCreateOptionsMenu(menu);
119 }

```



Μπορούμε να προσθέσουμε και να αφαιρέσουμε αναζήτηση η ρυθμίσεις στο action bar.

```

120 private void updateFeed(int feedNum) {
121     try{
122         downloadNewsTask.cancel(true);
123     }catch(Exception e){
124     }
125 }
//pbhorizontal.setMax(rssMenuButtonProp[feedNum].getItemCount());
downloadNewsTask = new DownloadNewsTask(this);
127 Integer maincategory = feedNum;
128 downloadNewsTask.execute(maincategory);
129 }

130 private void UpdateDisplay() {
//TextView feedtitle = (TextView) findViewById(R.id.feedtitle);
131 if (feed == null) {
132     list.setAdapter(null);
//feedtitle.setText("Ότιδήποτε...");
133 Toast.makeText(MainActivity.this, "Αν έχετε  

server,\n\nΑν έχετε ός 134 ότιδήποτε P όαδ P äiêiÛóôâ íáíÛ" ,
Toast.LENGTH_LONG).show();
135 return;
136 }
//feedtitle.setText(strTitle2);
137 rssitems = (Vector<RSSItem>) feed.getAllItems();
138 adapter = new LazyAdapter(this, rssitems);
139 list.setAdapter(adapter);

```

```

140 list.setOnItemClickListener(this);
//list.setSelection(0);
141 }
142 private class DownloadNewsTask extends AsyncTask<Integer, Void,
143RSSFeed> implements OnDismissListener{
144 MainActivity _owner;
145 DownloadNewsTask(MainActivity owner)
146{
147 _owner = owner;
148 }

149 XMLReader xmlreader;
150 @Override
151 protected RSSFeed doInBackground(Integer... params) {
152 String urlToRssFeed = RSSFEED_BASE + params[0].intValue();
// if the tomorrow radio button is checked add the tomorrow parameter for
the php 153 file so it returns tomorrow's routes
154 if (rbTomorrow.isChecked())
155 urlToRssFeed += "&tomorrow=1";
156 try {
// setup the url
157 URL url = new URL(urlToRssFeed);
// create the factory
158 SAXParserFactory factory = SAXParserFactory.newInstance();
// create a parser
159 SAXParser parser = factory.newSAXParser();
// create the reader (scanner)
160 xmlreader = parser.getXMLReader();
// instantiate our handler
161 RSSHandler theRssHandler = new RSSHandler(_owner);
// assign our handler
162 xmlreader.setContentHandler(theRssHandler);
// get our data via the url class
163 if (isCancelled()) return null;
164InputSource is = new InputSource(url.openStream());
// perform the synchronous parse
//pbhorizontal.setIndeterminate(false);
165 publishProgress((Void)null);
166 xmlreader.parse(is);
// get the results - should be a fully populated RSSFeed instance,
// or null on error
167 if (isCancelled()) return null;
168 return theRssHandler.getFeed();

```

```

169 } catch (Exception ee) {
// if we have a problem, simply return null
170 return null;
171 }}
172 @Override
173 protected void onCancelled() {
174 xmlreader = null;
175 }
176 @Override
177 protected void onPreExecute() {
// if cancelled before starting parsing, simply exit
178 if (isCancelled())
179 this.cancel(true);
180 if (getResources().getConfiguration().orientation ==
181 Configuration.ORIENTATION_PORTRAIT)
182 setRequestedOrientation
183 (ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
184 else
185 setRequestedOrientation
186 (ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
187 }
188 @Override
189 protected void onProgressUpdate(Void... progress) {
//setTitleMessage("Έπιθç ÁέäÞóâùí:\n[ "+_owner.strTitle2+" ]");
190 super.onProgressUpdate(progress);
191 }
Εμφανίζει σε ποιο σημείο είναι στο κατέβασμα. (Super καλεί την asynk
task)
192 @Override
193 protected void onPostExecute(RSSFeed result) {
194 feed = result;
195 ListView itemlist = (ListView) findViewById(R.id.lvRoutes);
196 itemlist.setVisibility(ListView.VISIBLE);
197 UpdateDisplay();
198 setRequestedOrientation
199 (ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
200 }
201 public void onDismiss() {
202 }
203 @Override
204 public void onItemClick(AdapterView<?> parent, View v, int
position, long id) 205 {
206 itemPos = position;

```

```

207 Intent itemintent = new Intent(this, ShowDetails.class);
208 Bundle b = new Bundle();
209 String day = "ÓÐiãñá";
210 if (rbTomorrow.isChecked())
211 day = "Áýñéi";
212 b.putString("title", feed.getItem(position).getTitle());
213 b.putString("source", feed.getItem(position).getSource());
214 b.putString("dest", feed.getItem(position).getDest());
215 b.putString("depDateTime", day+"
216 "+feed.getItem(position).getDepTime());
217 b.putString("depDate", day);
218 b.putString("depTime", feed.getItem(position).getDepTime());
219 b.putInt("availableSeats",
220 Integer.parseInt(feed.getItem(position).getAvailableSeats()));
221 b.putString("availableSeatsStr",
feed.getItem(position).getAvailableSeats());

222 itemintent.putExtra("android.intent.extra.INTENT", b);
223 startActivityForResult(itemintent, 2);
224 }

225 @Override
226 protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
227 super.onActivityResult(requestCode, resultCode, data);
228 if (resultCode == RESULT_OK) {
229 int reservedSeats = data.getExtras().getInt("reservedSeats");
230 int theSeats =
Integer.parseInt(feed.getItem(itemPos).getAvailableSeats());
231 theSeats -= reservedSeats;
232 feed.getItem(itemPos).setAvailableSeats(Integer.toString(theSeats));
// force repaint of the adapter by notifying the list's adapter
233 adapter.notifyDataSetChanged();
234 }}
// NAVIGATION DRAWER MENU
235 private void _initMenu() {
236 NsMenuAdapter mAdapter = new NsMenuAdapter(this);
// Add Header
237 mAdapter.setHeader(R.string.ns_menu_main_header);
// Add first block
238 menuItems = getResources().getStringArray(
239 R.array.ns_menu_items);
240 String[] menuItemsIcon = getResources().getStringArray(

```

```

241 R.array.ns_menu_items_icon);
242 int res = 0;
243 for (String item : menuItems) {
244 int id_title = getResources().getIdentifier(item, "string",
245 this.getPackageName());
246 int id_icon = getResources().getIdentifier(menuItemsIcon[res],
247 "drawable", this.getPackageName());
248 NsMenuItemModel mItem = new NsMenuItemModel(id_title,
id_icon);
//if (res==1) mItem.counter=12; //it is just an example...
//if (res==3) mItem.counter=3; //it is just an example...
249 mAdapter.addItem(mItem);
250 res++;
251 }
252 mAdapter.addHeader(R.string.ns_menu_main_header2);
253 menuItems2 = getResources().getStringArray(
254 R.array.ns_menu_items2);
255 String[] menuItemsIcon2 = getResources().getStringArray(
256 R.array.ns_menu_items_icon2);
257 res = 0;
258 for (String item : menuItems2) {

259 int id_title = getResources().getIdentifier(item, "string",
260 this.getPackageName());
261 int id_icon = getResources().getIdentifier(menuItemsIcon2[res],
262 "drawable", this.getPackageName());
263 NsMenuItemModel mItem = new NsMenuItemModel(id_title,
id_icon);
//if (res==1) mItem.counter=12; //it is just an example...
//if (res==3) mItem.counter=3; //it is just an example...
264 mAdapter.addItem(mItem);
265 res++;
266 }
267 mDrawerList = (ListView) findViewById(R.id.drawer);
268 if (mDrawerList != null)
269 mDrawerList.setAdapter(mAdapter);
270 mDrawerList.setOnItemClickListener(new
DrawerItemClickListener());
271 }
272 @Override
273 protected void onCreate(Bundle savedInstanceState) {
274 super.onCreate(savedInstanceState);
// Sync the toggle state after onRestoreInstanceState has occurred.

```

```

275 mDrawerToggle.syncState();
276 }
277 @Override
278 public void onConfigurationChanged(Configuration newConfig) {
279     super.onConfigurationChanged(newConfig);
280     mDrawerToggle.onConfigurationChanged(newConfig);
281 }
/* Called whenever we call invalidateOptionsMenu() */
282 @Override
283 public boolean onPrepareOptionsMenu(Menu menu) {
// If the nav drawer is open, hide action items related to the content view
284     boolean drawerOpen = mDrawer.isDrawerOpen(mDrawerList);
285     menu.findItem(R.id.action_save).setVisible(!drawerOpen);
286     return super.onPrepareOptionsMenu(menu);
287 }
288 @Override
289 public boolean onOptionsItemSelected(MenuItem item) {
/*
* The action bar home/up should open or close the drawer.
* ActionBarDrawerToggle will take care of this.
*/
290     if (mDrawerToggle.onOptionsItemSelected(item)) {
291         return true;
292     }
// Handle your other action bar items...
293     return super.onOptionsItemSelected(item);
294 }
295 private class CustomActionBarDrawerToggle extends
ActionBarDrawerToggle 296 {
297     public CustomActionBarDrawerToggle(Activity
mActivity, DrawerLayout 298 mDrawerLayout){
299         super(
300             mActivity,
301             mDrawerLayout,
302             R.drawable.ic_drawer,
303             R.string.ns_menu_open,
304             R.string.ns_menu_close);
305     }
306     @Override
307     public void onDrawerClosed(View view) {
308         getActionBar().setTitle(getString(R.string.ns_menu_close));
309         invalidateOptionsMenu(); // creates call to onPrepareOptionsMenu()
310     }

```

```

311 @Override
312 public void onDrawerOpened(View drawerView) {
313     getActionBar().setTitle(getString(R.string.ns_menu_open));
314     invalidateOptionsMenu(); // creates call to onPrepareOptionsMenu()
315 }
316 private class DrawerItemClickListener implements
317     ListView.OnItemClickListener {
318     @Override
319     public void onItemClick(AdapterView<?> parent, View view, int
    position,
    long id) {
    // Highlight the selected item, update the title, and close the drawer
    // update selected item and title, then close the drawer
320     mDrawerList.setItemChecked(position, true);
    //String text= "menu click... position "+position;
    //Toast.makeText(MainActivity.this, text ,
    Toast.LENGTH_LONG).show();
321     feedNum = position;
322     updateFeed(feedNum);
    //You should reset item counter
323     mDrawer.closeDrawer(mDrawerList);
324 }
}

```

Γραμμές 1 – 30: Βιβλιοθήκες Android

Γραμμές 31 – 72: Δημιουργείτε ένα bundle που θα εμφανιστεί στο layout.activity\_main\_drawer. Με τη βοήθεια του findViewById (int) ανακτούμε τα widgets (rbtoday & rbtomorrow ) στο UI που θα πρέπει να αλληλεπιδρούν, έτσι ώστε να μας εμφανίσει τα δρομολόγια σύμφωνα με το κουμπί που επιλέχθηκε. Τέλος γίνεται η λήψη των δρομολογίων και ενημερώνει τον χρήστη με ανάλογο μήνυμα(toast)

Γραμμές 72 – 96 : Έχουμε ορίσει με το άνοιγμα της εφαρμογής να εμφανίζει το δρομολόγιο 4 (Πάτρα-Αθήνα)

Γραμμές 97 – 129 : Η μέθοδος αυτή μας βοηθάει έτσι ώστε αν η εφαρμογή κατεβάζει και πατήσουμε κάτι άλλο κάνει cancel και ανανεώνει τα feeds.

Γραμμές 130 – 136 : Ανοίγοντας την εφαρμογή αν δεν υπάρχει πρόσβαση στο internet μας εμφανίζει (δεν βρέθηκε ο σερβερ)

Γραμμές 137 – 148 : Με τη βοήθεια της κλάσης DownloadNewsTask μπορούμε να κατεβάζουμε και να διαχειριζόμαστε την εφαρμογή χωρίς να παγώνει. Πχ. Αν είναι αργό το δίκτυο.

Γραμμές 149 – 171 : Αυτό που κάνει αυτός ο κώδικας είναι να προσδιορίσει το factory API το οποίο δίνει τη δυνατότητα στην εφαρμογή να ρυθμίσει και να αποκτήσει ένα SAX parser για την λήψη των XML δεδομένων.

Γραμμές 172 – 187 : Πριν ξεκινήσει ο parser αν γίνει διακοπή σταματά και καθαρίζει ο xml reader.

Γραμμές 188 – 200 : Κάνει ορατή την λίστα δρομολογίων και ανανεώνει την οθόνη της εφαρμογής.

Γραμμές 201 – 224 : Με την επιλογή κάποιου δρομολογίου δημιουργείτε ένα intent το οποίο περιέχει το activity και το bundle b τα οποία περνούν πληροφορίες στη δεύτερη οθόνη (extras). Αν ο χρήστης επιστρέψει στη πρώτη οθόνη ανανεώνει τις αλλαγές π.χ. θέσεις

Γραμμές 225 – 234: Αν η κατάσταση στο τέλος της κράτησης είναι result\_ok τότε μεταφέρει τις reservedSeats στην πρώτη οθόνη και στην onActivityResult αφαιρεί από τις διαθέσιμες θέσεις όσες κρατήθηκαν.

Γραμμές 235 – 324: Δημιουργία drawer

Γραμμές 316 – 324: Ξεκινάει ο έλεγχος για το ποιο feed num επιλέχθηκε και κάνει update το feed και κλείνει ο drawer.

## Arrays.xml

Για την υλοποίηση του Drawer χρησιμοποιήσαμε τη βιβλιοθήκη android-support-v4.jar. Προσθέσαμε στο arrays.xml 24 snippets για όλα τα διαθέσιμα δρομολόγια που είναι μέσα στη λίστα του ns menu.

```
<string-array name="ns_menu_items">
    <item>ns_menu_snippet1</item>
    <item>ns_menu_snippet2</item>
    <item>ns_menu_snippet3</item>
    <item>ns_menu_snippet4</item>
    <item>ns_menu_snippet5</item>
    <item>ns_menu_snippet6</item>
    <item>ns_menu_snippet7</item>
    <item>ns_menu_snippet8</item>
    <item>ns_menu_snippet9</item>
    <item>ns_menu_snippet10</item>
    <item>ns_menu_snippet11</item>
    <item>ns_menu_snippet12</item>
</string-array>
<string-array name="ns_menu_items2">
```



```
<item>ns_menu_snippet14</item>
<item>ns_menu_snippet15</item>
<item>ns_menu_snippet16</item>
<item>ns_menu_snippet17</item>
<item>ns_menu_snippet18</item>
<item>ns_menu_snippet19</item>
<item>ns_menu_snippet20</item>
<item>ns_menu_snippet21</item>
<item>ns_menu_snippet22</item>
<item>ns_menu_snippet23</item>
<item>ns_menu_snippet24</item>
</string-array>
```

## Strings.xml

Σε αυτό το αρχείο δώσαμε τίτλο για όλα τα δρομολόγια και την αντιστοιχία του κάθε snippet. Επίσης αλλάξαμε τα main header «ΑΠΟ ΠΑΤΡΑ» και «ΠΡΟΣ ΠΑΤΡΑ»

```
<string name="app_name">Δρομολόγια</string>
```

```
<string name="menu_settings">Settings</string>
```

```
<string name="ebook_title">ΤΙΤΛΟΣ</string>
```

```
<string name="ebook_preview">Κείμενο Preview</string>
```

```
<string name="ebook_shortinfo">Σύντομο Κατατοπιστικό
Κείμενο</string>
```

```
<string name="action_settings">Settings</string>
```

```
<string name="action_save">Save</string>
```

```
<string name="hello_world">Hello world!</string>
```

```
<string name="drawer_text">Navigation Drawer example.</string>
```

```
<string name="ns_menu_main_header">ΠΡΟΣ ΠΑΤΡΑ</string>
```

```
<string name="ns_menu_main_header2">ΑΠΟ ΠΑΤΡΑ</string>
```

```
<string name="ns_menu_open">Σημερινά Δρομολόγια</string>
```

```
<string name="ns_menu_close">Δρομολόγια</string>
```

```
<string name="ns_menu_snippet1">Αίγιο - Πάτρα</string>
```

```
<string name="ns_menu_snippet2">Άργος - Πάτρα</string>
```

```

<string name="ns_menu_snippet3">Άρτα - Πάτρα</string>
<string name="ns_menu_snippet4">Αθήνα - Πάτρα</string>
<string name="ns_menu_snippet5">Χαλκίδα - Πάτρα</string>
<string name="ns_menu_snippet6">Ιωάννινα - Πάτρα</string>
<string name="ns_menu_snippet7">Καλαμάτα - Πάτρα</string>
<string name="ns_menu_snippet8">Καρδίτσα - Πάτρα</string>
<string name="ns_menu_snippet9">Πύργος - Πάτρα</string>
<string name="ns_menu_snippet10">Θεσ/νίκη - Πάτρα</string>
<string name="ns_menu_snippet11">Τίρανα - Πάτρα</string>
<string name="ns_menu_snippet12">Βόλος - Πάτρα</string>

```

```

<string name="ns_menu_snippet14">Πάτρα - Αίγιο</string>
<string name="ns_menu_snippet15">Πάτρα - Άργος</string>
<string name="ns_menu_snippet16">Πάτρα - Άρτα</string>
<string name="ns_menu_snippet17">Πάτρα - Αθήνα</string>
<string name="ns_menu_snippet18">Πάτρα - Χαλκίδα</string>
<string name="ns_menu_snippet19">Πάτρα - Ιωάννινα</string>
<string name="ns_menu_snippet20">Πάτρα - Καλαμάτα</string>
<string name="ns_menu_snippet21">Πάτρα - Καρδίτσα</string>
<string name="ns_menu_snippet22">Πάτρα - Πύργος</string>
<string name="ns_menu_snippet23">Πάτρα - Θεσ/νίκη</string>
<string name="ns_menu_snippet24">Πάτρα - Τίρανα</string>
<string name="ns_menu_snippet25">Πάτρα - Βόλος</string>

```

Στις κλάσεις NsmenuAdapter.java και NsmenuItemModel.java δεν έγιναν κάποιες αλλαγές καθώς ήταν ενσωματωμένες με το tutorial του drawer που χρησιμοποιήθηκε.

### **6.3 LazyAdapter**

Η κλάση LazyAdapter μας επιτρέπει να αναγνωρίζουμε μέσα στην κλάση τα στοιχεία UI που έχουμε δημιουργήσει στο item.xml αρχεία ώστε να μπορούμε να τα επεξεργαστούμε.

Για παράδειγμα, αφού έχουμε ορίσει ένα Text View στο αντίστοιχο xml αρχείο όπως φαίνεται παρακάτω

```

<TextView
    android:id="@+id/text"
    android:layout_width="fill_parent"
    android:layout_height="22dp"

```

```
android:minHeight="35dp"  
android:text="@string/ebook_title"  
android:textColor="#000000"  
android:textSize="18sp"  
android:textStyle="bold" />
```

```
<TextView  
    android:id="@+id/preview"  
    android:layout_width="match_parent"  
    android:layout_height="24dp"  
    android:minHeight="45dp"  
    android:paddingBottom="7dp"  
    android:text="@string/ebook_preview"  
    android:textColor="#000000"  
    android:textSize="14sp" />
```

```
<TextView  
    android:id="@+id/detail"  
    android:layout_width="wrap_content"  
    android:textColor="#777777"  
    android:textSize="14sp"  
    android:layout_height="fill_parent"  
android:text="@string/ebook_shortinfo" />
```

πληκτρολογούμε στην κλάση την εξής εντολή:

```
TextView text=(TextView)vi.findViewById(R.id.text);
```

ώστε να ορίσουμε ότι η μεταβλητή title τύπου TextView αναφέρεται στο UI στοιχείο TextView το οποίο ορίσαμε με κωδικό όνομα id text .

```
TextView detail = (TextView)vi.findViewById(R.id.detail);
```

ορίζουμε ότι η μεταβλητή detail τύπου TextView αναφέρεται στο UI στοιχείο TextView το οποίο ορίσαμε με κωδικό όνομα id detail .

```
TextView preview=(TextView)vi.findViewById(R.id.preview);
```

ορίζουμε ότι η μεταβλητή preview τύπου TextView αναφέρεται στο UI στοιχείο TextView το οποίο ορίσαμε με κωδικό όνομα id preview.

Τα στοιχεία της λίστας στην main γίνονται style από την κλάση Lazyadapter.

## **6.4 RSSItem**

Περιγραφή: η κλάση αυτή δεν φαίνεται πουθενά στον προσκήνιο της εφαρμογής αφού δεν έχει εισόδους και εξόδους αλλά ούτε καλεί ή καλείται από κάποια. Η κλάση RssItem δεν έχει κάποιο κομμάτι κώδικα παρά μόνο την δήλωση κάποιων μεταβλητών. Ο λόγος, λοιπόν, ύπαρξης

της κλάσης αυτής είναι ότι με την δήλωση αυτών των μεταβλητών βοηθά άλλες κλάσεις και κυρίως την RssActivity να ορίσει μεταβλητές (για την ακρίβεια ArrayList) οι οποίες θα είναι τύπου RssItem, δηλαδή θα έχουν αυτά τα χαρακτηριστικά της κλάσης RssItem. Πιο συγκεκριμένα, οι μεταβλητές οι οποίες δηλώνονται σε αυτή την κλάση είναι τα στοιχεία τα οποία θέλουμε να κάνουμε parse από το feed της ιστοσελίδας και είναι τα παρακάτω:

```
private String _title = null;
private String _description = null;

private String _dep_time = null;
private String _reserved_seats = null;
private String _available_seats = null;
private String _source = null;
private String _dest = null;
```

## 6.5 RSSFeed

Περιγραφή: όπως και η κλάση RssItem έτσι και η RssFeed δεν φαίνεται πουθενά στον προσκήνιο της εφαρμογής αφού δεν έχει εισόδους και εξόδους αλλά ούτε καλεί ή καλείται από κάποια. Η κλάση RssFeed δημιουργήθηκε για να αρχικοποιεί τον item counter και να δημιουργεί μια λίστα από items. Σε περίπτωση που δεν υπάρχουν δρομολόγια ο counter θα μείνει 0.

## 6.6 RSSHandler

Περιγραφή: Η δουλειά που κάνει είναι να μεταφέρει δεδομένα από μία δραστηριότητα σε μία άλλη, είτε τα δεδομένα είναι ένα μήνυμα είτε αντικείμενα. Ο Handler μας φάνηκε χρήσιμος για την αποστολή μηνυμάτων από και προς την MainActivity που ασχολείται με την σύνδεση και ανταλλαγή feeds και items. Ουσιαστικά, τα feeds και items φιλοξενούνται σε μία λίστα, και στην συνέχεια όταν η MainActivity μπορεί να λάβει κάποιο item για επεξεργασία, μεταφέρεται με την βοήθεια του Handler.

Εν ολίγοις δέχεται τα feeds που έχει διαβάζει ο sax parser. Στη συνέχεια ψάχνει για elements έτσι ώστε να δημιουργήσει items. Όταν αναγνωρίσει ένα κλειστό element π.χ. (</item>) το προσθέτει στη λίστα με τα items.

```
public void startElement(String namespaceURI, String localName,String
qName, Attributes atts) throws SAXException
{
    chars = new StringBuffer();
    if (localName.equals("item"))
    {
```

```

        // create a new item
        _item = new RSSItem();
        return;
    }
}

public void endElement(String namespaceURI, String localName, String
qName) throws SAXException
{
    if (localName.equalsIgnoreCase("title"))
        _item.setTitle(chars.toString().trim());
    else if (localName.equalsIgnoreCase("description"))
        _item.setDescription(chars.toString().trim());

    else if (localName.equalsIgnoreCase("dep_time"))
        _item.setDepTime(chars.toString().trim());
    else if (localName.equalsIgnoreCase("reserved_seats"))
        _item.setReservedSeats(chars.toString());
    else if (localName.equalsIgnoreCase("available_seats"))
        _item.setAvailableSeats(chars.toString());
    else if (localName.equalsIgnoreCase("source"))
        _item.setSource(chars.toString());
    else if (localName.equalsIgnoreCase("dest"))
        _item.setDest(chars.toString());

    else if (localName.equals("item"))
    {
        // add our item to the list!
        _feed.addItem(_item);

        return;
    }
}
}

```

## **6.7 Show Details**

```

1 tvTitle = (TextView) findViewById(R.id.tvDetailsTitle);
2 tvTitle.setText(b.getString("title"));

3 tvDate = (TextView) findViewById(R.id.tvDetailsDateTime);
4 tvDate.setText(b.getString("depDate"));

5 tvSeats = (TextView)
6 findViewById(R.id.tvDetailsAvailableSeats);
7 tvSeats.setText("Àěáýèañò ÈÝóáéò:
8 "+b.getString("availableSeatsStr"));

```



```

46 response.getEntity().getContent().close();
47 throw new IOException(statusLine.getReasonPhrase());
48 }
49 } catch (ClientProtocolException e) {
50 } catch (IOException e) {
51 }
52 return responseString;
53 }
54 @Override
55 protected void onPostExecute(String result) {
56 super.onPostExecute(result);
57 Toast.makeText(ShowDetails.this, responseString ,
58 Toast.LENGTH_LONG).show();
59 if (responseString.contains("Άδέοδ÷Ðò êñÛôçóç")){
60 Intent resultIntent = new Intent(ShowDetails.this,
61 MainActivity.class);
62 resultIntent.putExtra("reservedSeats",
63 Integer.parseInt(etSeats.getText().toString()));
64 setResult(RESULT_OK, resultIntent);
65 finish();
66 class RequestTask extends AsyncTask<String, String, String>{
67 String responseString = null;
68 @Override
69 protected String doInBackground(String... uri) {
70 HttpClient httpclient = new DefaultHttpClient();
71 HttpResponse response;
72 try {
73 response = httpclient.execute(new HttpGet(uri[0]));
74 StatusLine statusLine = response.getStatusLine();
75 if(statusLine.getStatusCode() == HttpStatus.SC_OK){
76 ByteArrayOutputStream out = new ByteArrayOutputStream();
77 response.getEntity().writeTo(out);
78 out.close();
79 responseString = out.toString();
80 } else{
81 response.getEntity().getContent().close();
82 throw new IOException(statusLine.getReasonPhrase());
83 }
84 } catch (ClientProtocolException e) {
85 } catch (IOException e) {
86 }
87 return responseString;
88 }

```

```

87 @Override
88 protected void onPostExecute(String result) {
89     super.onPostExecute(result);
90     Toast.makeText(ShowDetails.this, responseString ,
91     Toast.LENGTH_LONG).show();
92     if (responseString.contains("Άδέδω÷Pð êñÛôçóç")){
93     Intent resultIntent = new Intent(ShowDetails.this,
94     MainActivity.class);
95     resultIntent.putExtra("reservedSeats",
96     Integer.parseInt(etSeats.getText().toString()));
97     setResult(RESULT_OK, resultIntent);
98     finish();

```

Γραμμές 1 - 8 : Η κλάση αυτή ανοίγει την δεύτερη οθόνη της εφαρμογής μας όταν γίνει επιλογή κάποιου δρομολογίου. Στον κώδικα που ακολουθεί λαμβάνει το bundle που έρχεται από την main activity και περιέχει όλα τα χαρακτηριστικά του.

Γραμμές 9 -26 : Σε αυτό το κομμάτι κώδικα εισάγουμε το όνομα, e-mail και θέσεις έτσι ώστε πατώντας το κουμπί submit να γίνουν οι απαραίτητοι έλεγχοι για την διαθεσιμότητα θέσεων. Αν ο αριθμός των θέσεων που επιθυμούμε είναι μεγαλύτερος των διαθέσιμων τότε δεν γίνεται κράτηση και μας ενημερώνει με ανάλογο μήνυμα (toast).

Γραμμές 27 -30: Με την βοήθεια της παρακάτω εντολής περνάμε τις παραμέτρους που μεταφέρονται μέσω του url έτσι ώστε να γίνει ένα νέο αίτημα στο reservation.php και να τα καταχωρήσει στη βάση δεδομένων.

Γραμμές 31 -53 : Στην παρακάτω κλάση γίνεται ένα http request προς το server.

Γραμμές 54 -97 : Στο σημείο αυτό παίρνει ανάλογο μήνυμα αν είναι επιτυχής η κράτηση. Τότε το bundle μεταφέρετε από την showDetails στην MainActivity και αλλάζει τις διαθέσιμες θέσεις. Αν δεν είναι επιτυχής ενημερώνει με ανάλογο μήνυμα.



# Κεφάλαιο 7<sup>ο</sup>

## Αρχεία Php

### **Routes.php**

```
1 <?php include('_includes/config.php');?>
2 <?php include('_includes/inc_mysql_con.php');?>
3 <?php
4 $extraSQL="";
5 $mySource="";
6 if(isset($_GET['source'])) {
7 $mySource=$_GET['source'];
8 }
9 $myDest="";
10 if(isset($_GET['dest'])) {
11 $myDest=$_GET['dest'];
12 }
13 if (trim($mySource)!="")
14 $extraSQL = " and t.source='".$mySource.'" ";
15 if (trim($myDest)!=""){
16 if (trim($extraSQL)=="")
17 $extraSQL = " and t.dest='".$myDest.'" ";
18 else
19 $extraSQL = $extraSQL . " and t.dest='".$myDest.'" ";
20 }
21 $date = ' current_date ';
22 $time = ' and t.dep_time>=current_time ';
23 if(isset($_GET['tomorrow'])) {
24 $date = ' date_add(current_date, INTERVAL 1 Day) ';
25 $time = ";
26 }
27 if(isset($_GET['routemenu'])) {
28 $routemenu = $_GET['routemenu'];
29 if (is_numeric($routemenu)){
30 switch ($routemenu){
30 case 1:      $extraSQL = " and t.source='Aigio' and t.dest='Patra' ";break;
31 case 2:      $extraSQL = " and t.source='Argos' and t.dest='Patra' ";break;
32 case 3:      $extraSQL = " and t.source='Arta' and t.dest='Patra' ";break;
33 case 4:      $extraSQL = " and t.source='Athens' and t.dest='Patra' ";break;
34 case 5:      $extraSQL = " and t.source='Chalkida' and t.dest='Patra' ";break;
35 case 6:      $extraSQL = " and t.source='Ioannina' and t.dest='Patra' ";break;
36 case 7:      $extraSQL = " and t.source='Kalamata' and t.dest='Patra' ";break;
37 case 8:      $extraSQL = " and t.source='Karditsa' and t.dest='Patra' ";break;
38 case 9:      $extraSQL = " and t.source='Pyrgos' and t.dest='Patra' ";break;
```

```

39 case 10:$extraSQL = " and t.source='Thessaloniki' and t.dest='Patra' ";break;
40case 11:      $extraSQL = " and t.source='Tirana Albania' and t.dest='Patra' ";break;
41 case 12:$extraSQL = " and t.source='Volos' and t.dest='Patra' ";break;
42 case 14:      $extraSQL = " and t.source='Patra' and t.dest='Aigio' "; break;
43 case 15:      $extraSQL = " and t.source='Patra' and t.dest='Argos' ";break;
44 case 16:      $extraSQL = " and t.source='Patra' and t.dest='Arta' ";break;
45 case 17:      $extraSQL = " and t.source='Patra' and t.dest='Athens' ";break;
46 case 18:$extraSQL = " and t.source='Patra' and t.dest='Chalkida' ";break;
47 case 19:$extraSQL = " and t.source='Patra' and t.dest='Ioannina' ";break;
48 case 20:$extraSQL = " and t.source='Patra' and t.dest='Kalamata' ";break;
49 case 21:$extraSQL = " and t.source='Patra' and t.dest='Karditsa' ";break;
50 case 22:$extraSQL = " and t.source='Patra' and t.dest='Pyrgos' ";break;
51 case 23:$extraSQL = " and t.source='Patra' and t.dest='Thessaloniki' ";break;
52 case 24:$extraSQL = " and t.source='Patra' and t.dest='Tirana Albania' ";break;
53 case 25:$extraSQL = " and t.source='Patra' and t.dest='Volos' ";break;
54 }
55 }
56 else die ('route must be between 1..12 or 14..25');
57 }
58 echo
59 '<?xml version="1.0" encoding="utf-8"?>
60 <rss version="2.0" xmlns:TeiKtel="".ROOT_URL."/">
61 <channel>
62 <title>'.SERVICE_TITLE.'</title>
63 <language>el-GR</language>
64 <image>
65<url>'.ROOT_URL.'/_includes/logo.png</url>
66 <title>'.SERVICE_TITLE.'</title>
67<link>'.ROOT_URL.'</link>
68</image>
69';
70 $query =
71 "select t.idtimetable, t.source, date_format(t.dep_time, '%H:%i')
dep_time, t.days, t.dest, t.remarks,
72 case when sum(p.num_pos) is null then 0 else sum(p.num_pos) end
reserved,
73 case when sum(p.num_pos) is null then 40 else 40-sum(p.num_pos)
end available
74 from timetable t left join reservation r on t.source=r.source and
t.dest=r.dest and 75 r.dep_time=t.dep_time
75left join psger p on r.id=p.route and r.dep_date={ $date }
76 where t.days like concat('%', dayname({ $date }),'%') { $time }
77 { $extraSQL }
78 group by t.idtimetable
79 order by t.dep_time"
80 $result = mysql_query($query, $con);
81 while($row = mysql_fetch_array($result)){
82 echo '

```

```

83 <item>
84<title><![CDATA['.$row['source'].'.$row['dest'].('.$row['dep_time'].')'.
]]></title>
85 <description><![CDATA['.$row['remarks'].']]></description>
86 <TeiKtel:dep_time>'.$row['dep_time'].</TeiKtel:dep_time>
87 <TeiKtel:reserved_seats>'.$row['reserved'].</TeiKtel:reserved_seats>
88
<TeiKtel:available_seats>'.$row['available'].</TeiKtel:available_seats>
89 <TeiKtel:source><![CDATA['.$row['source'].']]></TeiKtel:source>
90 <TeiKtel:dest><![CDATA['.$row['dest'].']]></TeiKtel:dest>
91 </item>
92 ';
93 }
94 echo '
95 </channel>
96 </rss>';?>
97 <?php include('_includes/inc_mysql_dc.php');?>

```

Γραμμές 1 - 3 : Σύνδεση με τη βάση δεδομένων

Γραμμές 4 - 26 : Καθορισμός επιλογών source και destination μέσω των cases 1 έως 12 14 έως 25. Ορισμός σημερινής ημερομηνίας και ώρας.(αν έχει οριστεί η μεταβλητή tomorrow κοιτάζει στην ΒΔ στην επόμενη μέρα)

Γραμμές 27 - 56 : Επιλογή δρομολογίου από τα διαθέσιμα cases

Γραμμές 58 – 69: Δήλωση χαρακτηριστικών xml

Γραμμές 70 – 97: Το βασικό query της PHP για όλα τα δρομολόγια (στο παράδειγμά μας Πάτρα - Αθήνα).

Παίρνει όλα τα δρομολόγια της σημερινής ημέρας όπου διαθεσιμότητα βάσει μέρας (π.χ. να περιέχεται η λέξη Wednesday στο πεδίο days αν η σημερινή ημέρα είναι Τετάρτη, αλλά και με τις κρατήσεις που έχουν γίνει για τη σημερινή ημέρα). Επιπλέον δείχνει τα δρομολόγια της σημερινής ημέρας από αυτή τη στιγμή και πέρα μόνο.

## Reservation.php

```
1 <?php include('_includes/config.php');?>
2 <?php include('_includes/inc_mysql_con.php');?>
3 <?php
4 $to = "";
5 if(isset($_GET['to'])) {
6   $to=$_GET['to'];
7 }
8 if ($to=="") die ('Δε δώσατε email');
9 $name = "";
10 if(isset($_GET['name'])) {
11   $name=$_GET['name'];
12 }
13 if ($name=="") die ('Δε δώσατε όνομα');
14 $seats = 0;
15 if(isset($_GET['seats'])) {
16   $seats=$_GET['seats'];
17 }
18 if ($seats<=0) die ('Δε δώσατε θέσεις');
19 $source = "";
20 if(isset($_GET['source'])) {
21   $source=$_GET['source'];
22 }
23 if ($source=="") die ('Δε δώσατε αφετηρία');
24 $dest = "";
25 if(isset($_GET['dest'])) {
26   $dest=$_GET['dest'];
27 }
28 if ($dest=="") die ('Δε δώσατε προορισμό');
29 $deptime = "";
30 if(isset($_GET['deptime'])) {
31   $deptime=$_GET['deptime'];
32 }
33 if ($deptime=="") die ('Δε δώσατε ώρα αναχώρισης');
34 $date = 'current_date';
35 $isToday = "";
36 if(isset($_GET['today'])) {
37   $isToday=$_GET['today'];
38 // today = date 1
```

```

// tomorrow = date 0
38 if ($isToday==0)
39 $date = ' date_add(current_date, INTERVAL 1 Day) ';
40 }
// check if route on specific date exists on the reservation table
41 $routeid = 0;
42 $query =
43 "select id, count(*) entries, date_format({$date}, '%d/%m/%Y')
44friendly_date
45 from reservation
46 where source='{ $source}'
47 and dest='{ $dest}'
48 and dep_date={ $date}
49 and dep_time='{ $deptime}''";
50 $result = mysql_query($query, $con) or die('could not check
reservation status');
51 $entries = 0;
52 while($row = mysql_fetch_array($result)){
53 $routeid = $row['id'];
54 $entries = $row['entries'];
55 $friendly_date = $row['friendly_date'];
56 }
// if no entry exists already, insert it
57 if ($entries==0){
58 $query =
59 "insert into reservation (source, dest, dep_date, dep_time) values
60 ('{ $source}', '{ $dest}', { $date}, '{ $deptime}')";
61 $result = mysql_query($query, $con) or die('Could not insert into
reservation table');
// get last inserted id to use it on our next insert statement
62 $query = 'SELECT LAST_INSERT_ID() AS insert_id';
63 $result = mysql_query($query, $con) or die('Could not obtain last
inserted id');
64 while($row = mysql_fetch_array($result)){
65 $routeid = $row['insert_id'];
66 }
67 }
68 $query =
69 "insert into psger (name, num_pos, route)
70 values
71 ('{ $name}', { $seats}, { $routeid})";
72 $result = mysql_query($query, $con) or die('could not insert passenger
data');

```

```

// get last inserted id to use it as reservation number for the email sent to
the client
73 $query = 'SELECT LAST_INSERT_ID() AS insert_id';
74 $result = mysql_query($query, $con) or die('Could not obtain last
inserted id');
75 while($row = mysql_fetch_array($result)){
76 $reservation_number = $row['insert_id'];
77 }
78 $subject = "ΚΤΕΛ ΠΑΤΡΑΣ - Κράτηση";
79 $message =
80 "Η Κράτησή σας ήταν επιτυχής
81 Δρομολόγιο: {$source} - {$dest}
82 Ώρα: {$deptime} - {$friendly_date}
83 Όνομα Κράτησης: {$name}
84 Θέσεις: {$seats}
85 Αριθμός Κράτησης: {$reservation_number}";
86 $from = "ktel@giannis-dimitra-tours.gr";
87 $headers = "From:" . $from;
88 mail($to,$subject,$message,$headers);
89 echo "Επιτυχής κράτηση, ελέγξτε το email σας";
90 ?>
91 <?php include('_includes/inc_mysql_dc.php');?>

```

Γραμμές 1 - 3 : Σύνδεση με τη βάση δεδομένων

Γραμμές 4 - 33 : Γίνετε ο έλεγχος για το αν έχουν περαστεί e-mail, όνομα, θέσεις, αφετηρία, προορισμό και ώρα αναχώρησης. Σε περίπτωση μη κατοχύρωσης έστω και ενός από τα παραπάνω μας εμφανίζει ένα μήνυμα λάθους.

Γραμμές 34 - 40 : Έλεγχος για το αν είναι σημερινό ή αυριανό δρομολόγιο

Γραμμές 41 - 55 : Ελέγχει αν υπάρχει ήδη κράτηση στο δρομολόγιο που έχουμε επιλέξει σε συγκεκριμένη ημερομηνία και ώρα.

Γραμμές 56 - 67 : Καταχώρηση στον πίνακα reservation μια νέα εγγραφή στην ημερομηνία και ώρα που έχουμε δώσει και τοποθετεί τα id κατά αύξοντα αριθμό

Γραμμές 68- 77 : Καταχώρηση των στοιχείων στον πίνακα psger με το όνομα, τις θέσεις που έχει κρατήσει και το route id της κράτησης που έχει κάνει. Επίσης κρατάμε το τελευταίο id κράτησης για να το σταλεί με e-mail στον πελάτη

Γραμμές 78- 89 : Αποστολή e-mail με τα στοιχεία που της κράτησης

# Μέρος Γ

## Βιβλιογραφία-Πηγές

<http://stackoverflow.com/questions/3505930/make-an-http-request-with-android>

[http://www.w3schools.com/php/php\\_mail.asp](http://www.w3schools.com/php/php_mail.asp)

<http://gmariotti.blogspot.gr/2013/05/creating-navigation-drawer.html>

<http://developer.android.com/tools/support-library/features.html>

<http://stackoverflow.com/questions/3505930/make-an-http-request-with-android>

<http://jqueryui.com/datepicker/>

<http://theopentutorials.com/tutorials/android/xml/android-simple-xml-sax-parser-tutorial/>

<http://www.codeproject.com/Articles/334859/Parsing-XML-in-Android-with-SAX>

<http://www.androidsnippets.com/simple-xml-parsing>

<http://www.w3schools.com/php/default.asp>

[http://www.w3schools.com/php/php\\_mysql\\_create.asp](http://www.w3schools.com/php/php_mysql_create.asp)

[http://www.w3schools.com/php/php\\_if\\_else.asp](http://www.w3schools.com/php/php_if_else.asp)

[http://www.w3schools.com/php/php\\_switch.asp](http://www.w3schools.com/php/php_switch.asp)