



# Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Μηχανικών Πληροφορικής



## Πτυχιακή Εργασία

**Τίτλος: Υλοποίηση Εφαρμογής Διαχείρισης Ξενοδοχείου Πόλης**

**Νάσος Τηλέμαχος Γιαννούσης (ΑΜ:1459)**

**Επιβλέπων Καθηγητής : Αϊβαλής Κωστής**

**ΗΡΑΚΛΕΙΟ  
2017**

## **Πίνακας Περιεχομένων**

## Πίνακας Εικόνων

- 1 Τύποι δεδομένων Sql.
- 2-3 Στήλες πίνακα guest
- 4 Πίνακας guest παράδειγμα.
- 5 Μεταγλώττιση κοινής γλώσσας.
- 6 Μεταγλώττιση Java
- 7 Κώδικας hello world
- 8 Εκτέλεση κώδικα Hello World
- 9 Ανάλυση κώδικα Hello world
- 10 Αλληλεπίδραση JDBC-ODBC
- 11-12 Σύνδεση Workbench με MySQL Server
- 13 Περιβάλλον σχεδίασης της βάσης δεδομένων
- 14 Σχεδίαση νέας βάσης
- 15 Διαδικασία εγκατάστασης NetBeans IDE
- 22 Δημιουργία project γραμμένο σε κώδικα Java με την βοήθεια του NetBeans
- 29 Αλληλεπίδραση δεδομένων ανάμεσα στο (ΣΔΒΔ) και την βάση δεδομένων.
- 31 Σχεδιασμός Υλοποίησης
- 32 Συσχετίσεις μεταξύ πινάκων
- 33 Ανάπτυξη του μοντέλου της βάσης δεδομένων
- 35 Ολοκληρωμένη βάση δεδομένων
- 36 mysql-connector-java-5.1.42-bin.jar
- 37 Κώδικας κεντρικού μενού pellahotel
- 51 Φωτογραφία κεντρικού μενού

## Σύνοψη

Είναι γνωστό ότι ο καλύτερος τρόπος να μάθει κάποιος κάτι είναι να το εφαρμόσει σε πραγματικό σενάριο. Προφανώς, το ίδιο ισχύει για την μοντελοποίηση αλλά και την διαχείριση οποιασδήποτε βάσης δεδομένων. Το συγκεκριμένο μοντέλο δεδομένων σχεδιάστηκε για ένα σύστημα κράτησης δωματίων το οποίο αφορά ένα μικρό ξενοδοχείο πόλης. Η εφαρμογή δεν απαιτεί ιδιαίτερες γνώσεις απο τον χρήστη καθώς το περιβάλλον εργασίας σχεδιάστηκε με όσο το δυνατόν λιγότερες επιλογές για να είναι ευκολότερη η κατανόηση και η διαχείριση του.

It is well known that the best way to learn something is to apply it to real scenario. Obviously, the same applies to the modeling and management of any base data. This data model was designed for a room reservation system which concerns a small city hotel. The application does not require special knowledge from the user as the interface was designed with as few options as possible easier to understand and manage.

# 1 Εισαγωγή

Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν δύο εφαρμογές (MySQL Workbench, NetBeans IDE) οι οποίες προσφέρουν ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού για την δημιουργία προγραμμάτων. Ο προγραμματιστής έχει την δυνατότητα με τα κατάλληλα οπτικά εργαλεία που του παρέχονται να σχεδιάσει και να υλοποιήσει ευκολότερα αλλά και πιο γρήγορα την βάση δεδομένων και το γραφικό περιβάλλον εργασίας του χρήστη. Οι τρόποι με τους οποίους σχεδιάστηκαν για να αλληλεπιδρούν μεταξύ τους οι εφαρμογές αναλύονται εκτενέστερα στα παρακάτω καιφάλα.

## 1.1 Περίληψη

Όλα τα σύγχρονα ξενοδοχεία μικρής ή μέσης κατηγορίας στις μέρες μας διαθέτουν ψηφιακή βάση δεδομένων για έλεγχο ή προσθήκη νέων στοιχείων των πελατών τους. Για την διαχείριση της βάσης δεδομένων δημιουργείται ένα γραφικό περιβάλλον από γλώσσες προγραμματισμού οι οποίες ορίζονται από ένα σύνολο συντακτικών και εννοιολογικών κανόνων, που τις καθιστούν ικανές να αλληλεπιδρούν μεταξύ τους. Για την συγκεκριμένη εφαρμογή διαχείρισης ξενοδοχείου πόλης χρησιμοποιήθηκαν οι γλώσσες προγραμματισμού SQL και Java.

Η SQL παρέχει την δυνατότητα δημιουργίας σχεσιακού μοντέλου από πίνακες οι οποίοι αποτελούνται από πεδία στα οποία μπορούμε να κάνουμε ανάκτηση και ενημέρωση δεδομένων. Επίσης ένα σημαντικό χαρακτηριστικό της SQL είναι ότι προσφέρει έλεγχο πρόσβασης στα δεδομένα.

Η Java από την άλλη μεριά είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού. Στον αντικειμενοστραφή προγραμματισμό ο προγραμματιστής μπορεί να καλέσει ή ακόμα και να ορίσει νέους τύπους δεδομένων που ονομάζονται κλάσεις.

## 1.2 Κίνητρο για τη διεξαγωγή της εργασίας

Το σημαντικότερο κίνητρο για την υλοποίηση εφαρμογής διαχείρισης ξενοδοχείου πόλης είναι ο τουρισμός που αποτελεί την κύρια βιομηχανία εσόδων της χώρας μας. Ο τουρισμός δημιουργεί θέσεις εργασίας οι οποίες σχετίζονται με αρκετά επαγγέλματα που απορροφούν εργαζόμενους από διάφορους κλάδους. Πολλοί από αυτούς τους κλάδους δεν έχουν άμεση σχέση με την πληροφορική η οποία όμως αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας μας απλουστεύοντας χρονοβόρες διαδικασίες.

Μια από αυτές τις διαδικασίες είναι η διαδικασία κράτησης δωματίου σε ένα ξενοδοχείο. Οι υπάλληλοι των ξενοδοχείων με τα εργαλεία που τους προσφέρει η πληροφορική όπως για παράδειγμα οι υπολογιστές και το διαδίκτυο έχουν πλέον την δυνατότητα να συλλέξουν τα στοιχεία του πελάτη και να τα καταχωρήσουν στην βάση δεδομένων πολύ πιο γρήγορα από το παρελθόν. Οι πελάτες με την σειρά τους μπορούν ακόμα και άμεσα να ενημερωθούν για την διαθεσιμότητα του ξενοδοχείου.

## 1.3 Σκοπός και στόχοι της εργασίας

Ένας από τους στόχους της εργασίας είναι η διαχείριση της εφαρμογής από τον χρήστη να προσεγγίζει όσο το δυνατόν περισσότερο ένα πραγματικό σενάριο κράτησης δωματίου. Αυτό δεν μπορεί να γίνει απόλυτα εφικτό καθώς μελλοντικά πάντα θα προκύπτουν προβλήματα τα οποία δεν ήταν δυνατόν να προβλεφθούν κατά την δημιουργία της εφαρμογής. Ένας άλλος πολύ σημαντικός στόχος είναι το γραφικό περιβάλλον της εφαρμογής να γίνεται εύκολα κατανοητό από τον χρήστη χωρίς να απαιτούνται ιδιέστερες γνώσεις πληροφορικής από αυτόν. Οι δύο αυτοί στόχοι αποσκοπούν στην δημιουργία βάσης δεδομένων από την οποία ο διαχειριστής του προγράμματος θα μπορεί εύκολα και γρήγορα να επεξεργαστεί τα στοιχεία των πελατών.

## 1.4 Δομή Εργασίας

Στο δεύτερο κεφάλαιο γίνεται αναλυτική περιγραφή στις μεθόδους που χρησιμοποιήθηκαν για την υλοποίηση της εργασίας αλλά και την αλληλεπίδραση δύο διαφορετικών γλωσσών προγραμματισμού όπως είναι η SQL και η Java. Θα γίνει επίσης αναφορά σε σημαντικές θεωρίες και αλγόριθμους που επηρεάζουν την λειτουργικότητα της εφαρμογής.

Στο τρίτο κεφάλαιο θα γίνει αναφορά στις ψηφιακές πλατφόρμες ανάπτυξης λογισμικού (MySQL Workbench, NetBeans IDE) που χρησιμοποιήθηκαν για τον σχεδιασμό και την υλοποίηση της βάσης δεδομένων και του γραφικού περιβάλλοντος στο οποίο θα εργάζεται ο χρήστης.

Στο τέταρτο κεφάλαιο θα αναλυθεί ο τρόπος με τον οποίο σχεδιάστηκε το συγκεκριμένο μοντέλο δεδομένων για ένα σύστημα κρατήσεων ξενοδοχείου.

## 2 Μεθοδολογία Υλοποίησης

Στο συγκεκριμένο κεφάλαιο γίνεται περιγραφή στις γλώσσες προγραμματισμού SQL και Java αλλά και στις μεθόδους που χρησιμοποιούν για να αλληλεπιδρούν μεταξύ τους.

### 2.1 Περιγραφή SQL

Η SQL (Structured Query Language) είναι μια γλώσσα προγραμματισμού για την κατασκευή βάσεων δεδομένων η οποία βασίζεται στην σχεσιακή άλγεβρα. Με την SQL ο διαχειριστής της βάσης έχει την δυνατότητα να δημιουργεί πίνακες και να τροποποιεί τις σχέσεις που έχουν μεταξύ τους καθώς επίσης να επεξεργάζεται τα δεδομένα που περιέχουν οι πίνακες. Αρχικά αναπτύχθηκε από την IBM στις αρχές της δεκαετίας του 1970 και είχε σκοπό να επεξεργάζεται τα δεδομένα του πρώτου συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων. Στις μέρες μας τα περισσότερα συστήματα βάσεων δεδομένων χρησιμοποιούν την SQL γιατί η διατύπωση σχεσιακών πράξεων είναι ευκολότερη από ότι στην σχεσιακή άλγεβρα στην οποία πρέπει να ορίσουμε ακριβώς την σειρά που θα εκτελεστούν οι πράξεις. Ουσιαστικά με την SQL ορίζουμε τι θέλουμε να κάνουμε και όχι το πώς αποδεσμευοντας τον χρήστη από λεπτομέρειες υλοποίησης. Μερικοί από τους όρους που χρησιμοποιεί η SQL είναι:

- Πίνακας
- Γραμμή
- Στήλη

Οι έννοιες αυτών των όρων αντίστοιχα είναι

- Σχέση
- Πλειάδα
- Χαρακτηριστικό

Υποδιαιρείται σε δύο ξεχωριστές υπογλώσσες:

- Τη γλώσσα ορισμού δεδομένων data definition language (DDL)
- Τη γλώσσα χειρισμού δεδομένων data manipulation language (DML)

Τύποι Δεδομένων	Δηλώσεις
λογικός	BOOLEAN
χαρακτήρας	CHAR, VARCHAR
δυαδικό ψηφίο	BIT, BIT VARYING
ακριβής αριθμητικός	NUMERIC, DECIMAL, INTEGER, SMALLINT
προσεγγιστικός αριθμητικός	FLOAT, REAL, DOUBLE PRECISION
ημερομηνία και ώρα	DATE, TIME, TIMESTAMP
διάστημα	INTERVAL

Εικόνα 1

Καθώς εισάγουμε δεδομένα στην βάση χρειάζεται προσοχή γιατί ο τύπος δεδομένων πρέπει να είναι συμβατός με αυτόν της στήλης.

Βασικές εντολές της SQL.

- Διαχείριση Σχήματος  
CREATE ALTER DROP
- Διαχείριση Δεδομένων  
INSERT UPDATE DELETE
- Ερωτήματα  
SELECT

Στο παρακάτω παράδειγμα θα δημιουργήσουμε ένα πίνακα με το όνομα **guest** και θα του προσθέσουμε τέσσερις στήλες *id*, *first\_name*, *last\_name*, *member\_since*. Κάθε στήλη του πίνακα έχει δικό της τύπο δεδομένων που επιβάλλει διάφορους περιορισμούς.

```
CREATE TABLE guest (
id int,
first_name varchar(45),
last_name varchar(45),
member_since TIMESTAMP
);
```

	id	first_name	last_name	member_since
--	----	------------	-----------	--------------

Εικόνα 2

Η στήλη με το όνομα **id** δέχεται μόνο ακέραιους, οι στήλες **first\_name** και **last\_name** μόνο χαρακτήρες. Ο τύπος δεδομένων της στήλης *member\_since* είναι ημερομηνία **TIMESTAMP** και συντάσσεται YYYY/MM/DD HH/MM/SS.

Για να εισάγουμε νέα δεδομένα στον πίνακα **guest** θα χρησιμοποιήσουμε την εντολή **INSERT INTO**.

```
INSERT INTO guest (first_name, last_name)
VALUES('Nasos', 'Gianusis')
```

Με την εισαγωγή των εγγραφών στις στήλες **first\_name** και **last\_name** δημιουργείται το **id** και η ημερομηνία εγγραφής. Το **id** είναι μια τιμή για κάθε γραμμή του πίνακα η οποία την κάνει μοναδική αποτρέποντας με αυτόν τον τρόπο την επανάληψη εγγραφών στην βάση.

Θα διατυπώσουμε ένα ερώτημα με την εντολή **SELECT** για να παρουσιάσουμε την εγγραφή που εισάγαμε.

```
SELECT id, first_name, last_name, member_since FROM guest
WHERE id=1;
```

	id	first_name	last_name	member_since
	1	Nasos	Gianusis	2017-11-05 15:14:40

Εικόνα 3

Για να παρουσιάσουμε όλες τις γραμμές του πίνακα **guest** χρησιμοποιούμε \* με την εντολή **SELECT**. **SELECT \* FROM guest;**

	id	first_name	last_name	member_since
	1	Nasos	Gianusis	2017-09-05 15:14:40
	2	Clint	Eastwood	2017-11-05 16:14:50
	3	Michael	Jordan	2017-10-27 02:04:37
	4	Tom	John	2017-07-07 11:07:45

Εικόνα 4

Όπως θα δούμε στο επόμενο κεφάλαιο που θα περιγράψουμε την γλώσσα αντικειμενοστραφούς προγραμματισμού Java επιτρέπεται η εμφύτευση εντολών SQL. Αυτό δεν ισχύει μόνο για την Java αλλά και για άλλες γλώσσες προγραμματισμού όπως η C/C++, Visual Basic, PHP και άλλες κάτι το οποίο καθιστά την SQL εξαιρετικά διαδεδομένη.



## 2.2 Περιγραφή γλώσσας αντικειμενοστραφούς προγραμματισμού Java

### 2.2.1 Ιστορία της Java

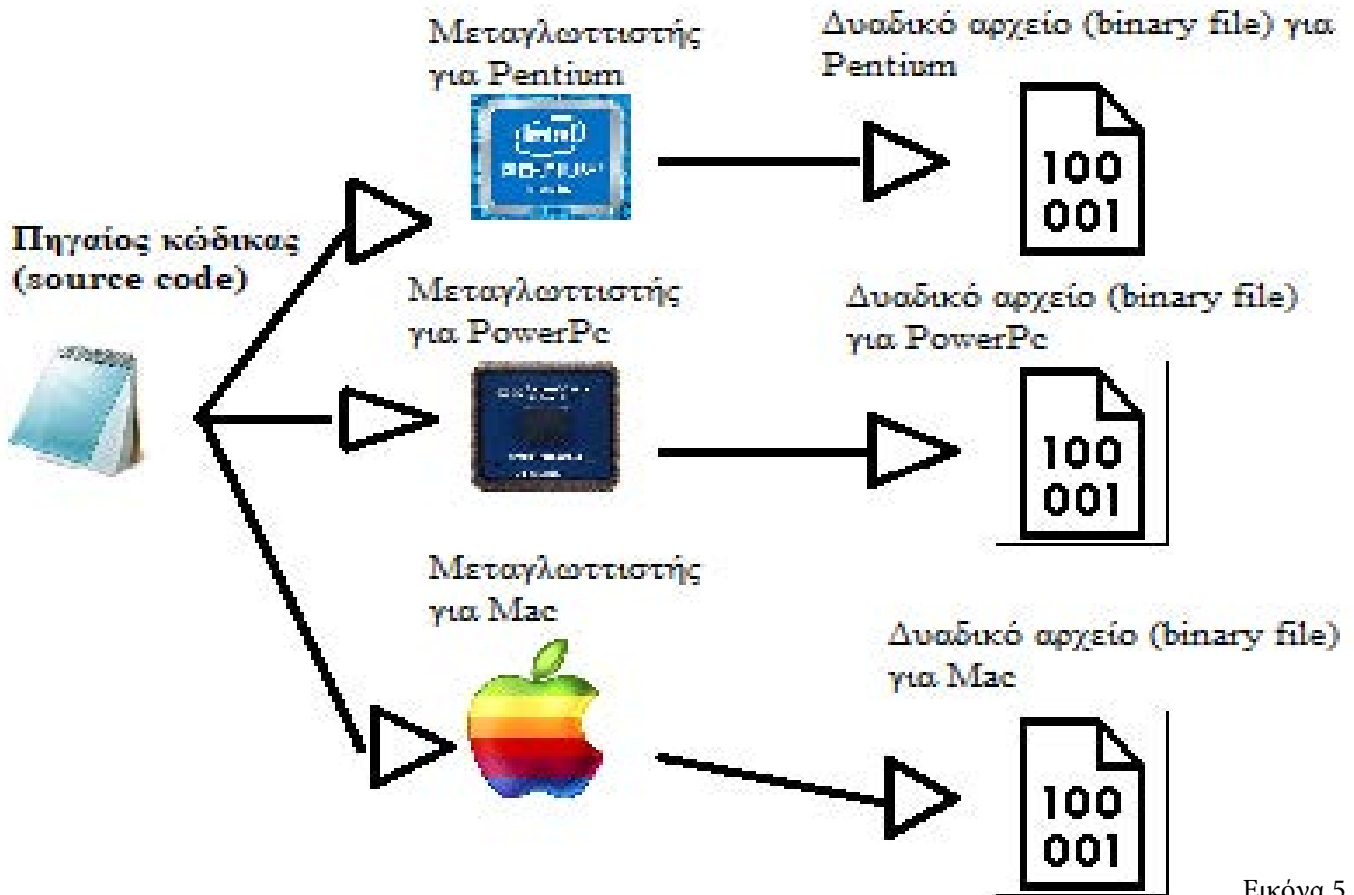
Η ιστορία της Java αρχίζει το 1990, όταν μια ομάδα ερευνητών της εταιρείας Sun Microsystems ξεκινάει ένα πρότζεκτ για την δημιουργία λογισμικού οικιακών ηλεκτρικών συσκευών (βίντεο, τηλεοράσεις, στερεοφωνικά, συστήματα συναγερμού). Το 1995 η εταιρία Sun παρουσιάζει την Java ως εργαλείο ανάπτυξης εφαρμογών για το Internet, τυποποιεί την γλώσσα και την διανέμει δωρεάν μέσω του παγκόσμιου ιστού (World Wide Web) δημιουργώντας το πρώτο πρόγραμμα περιήγησης αποκλειστικά γραμμένο σε Java HotJava. Οι εταιρίες Netscape Communications και Microsoft κυκλοφορούν τα προγράμματα περιήγησης Netscape Navigator και Internet Explorer αντίστοιχα με υποστήριξη Java.

### 2.2.2 Χαρακτηριστικά της Java

- Ένα από τα πράγματα που κάνουν την Java να ξεχωρίζει από τις άλλες γλώσσες προγραμματισμού, είναι ότι δίνει την δυνατότητα στον προγραμματιστή να δημιουργήσει μικροεφαρμογές οι οποίες ονομάζονται applets. Ένας χρήστης ο οποίος είναι συνδεδεμένος στο Internet μέσω ενός προγράμματος περιήγησης μπορεί να "κατεβάσει" και να "εκτελέσει" αυτές τις μικροεφαρμογές. Το πρόβλημα που προκύπτει όταν ένας χρήστης κατεβάζει αρχεία από το Internet με τον κίνδυνο να "εισπράξει" κάποιον ιό η Java το λύνει σε μεγάλο βαθμό περιορίζοντας αυστηρά το τι μπορεί να κάνει ένα applet στον σκληρό μας δίσκο ή στον υπολογιστή μας γενικότερα.
- Είναι σχετικά απλή. Οι δημιουργοί της Java άφησαν έξω από την γλώσσα δύσκολα χαρακτηριστικά που συναντάμε σε άλλες γλώσσες, όπως π.χ. οι δείκτες ή οι δομές.
- Είναι αντικειμενοστραφής (Object Oriented). Η Java όπως και η C χρησιμοποιεί τις κλάσεις (Classes) για να οργανώσει τον κώδικα σε λογικές ενότητες. Την στιγμή που εκτελείται ο κώδικας του προγράμματος οι κλάσεις δημιουργούν αντικείμενα. Τα αντικείμενα αυτά έχουν δύο συνιστώσες: τα πεδία και τις μεθόδους. Τα πεδία περιγράφουν τι είναι το αντικείμενο οι μέθοδοι περιγράφουν τι κάνει το αντικείμενο. Οι κλάσεις μπορούν να κληρονομήσουν ιδιότητες από άλλες κλάσεις. Εκείνο που δεν επιτρέπεται είναι η πολλαπλή κληρονομικότητα (multiple inheritance).
- Η μεταγλώττιση του πηγαίου προγράμματος έχει σαν αποτέλεσμα την παραγωγή ενός ειδικού κώδικα ο οποίος ονομάζεται κώδικας byte (bytecode). Αυτός ο κώδικας μοιάζει πολύ με τον κώδικα σε γλώσσα μηχανής, μπορεί να εκτελεστεί από οποιοδήποτε λειτουργικό σύστημα (Windows, Linux, Mac Os κτλ) που διαθέτει διερμηνευτή της Java. Η διαφορά του κώδικα byte από τον κώδικα σε γλώσσα μηχανής είναι ότι ο κώδικας byte είναι ίδιος σε όλες τις πλατφόρμες δίνοντας λύση στην Java από το πρόβλημα της ανεξαρτησίας πλατφόρμας (platform independence).
- Η Java σχεδιάστηκε από την αρχή με τέτοιο τρόπο, ώστε να παρέχει ασφάλεια εκτέλεσης του κώδικα στο δίκτυο. Αυτό έγινε περιορίζοντας χαρακτηριστικά όπως είναι για παράδειγμα οι δείκτες ή η αυθαίρετη προσπέλαση διευθύνσεων της μνήμης. Επίσης χρησιμοποιεί ισχυρό μηχανισμό για τον έλεγχο αναμενόμενων και μη αναμενόμενων σφαλμάτων (exception handling).
- Ένα πρόγραμμα Java μπορεί να περιλαμβάνει πολλές ξεχωριστές διαδικασίες, οι οποίες να εκτελούνται συνεχώς και ανεξάρτητα η μια από την άλλη. Η συγκεκριμένη διαδικασία ονομάζεται πολυνημάτωση (multithreading) και δίνει στον χρήστη την δυνατότητα λ.χ. να παρακολουθεί εικόνα από ένα πρόγραμμα και συγχρόνως να εισάγει στοιχεία από το πληκτρολόγιο.
- Λόγο της εξάπλωσης που έχει πάρει η Java πολλές εταιρίες έχουν αναπτύξει ειδικούς μεταγλωττιστές οι οποίοι μετατρέπουν τον κώδικα byte της Java σε γλώσσα μηχανής και μπορούν να ανταγωνιστούν σε ταχύτητα την C++ ή ακόμα και άλλες γλώσσες προγραμματισμού.
- Παρέχει βιβλιοθήκες με κώδικα για διάφορες χρήσεις, όπως δημιουργία γραφικών, χειρισμό αλφαριθμητικών, μαθηματικές πράξεις, προσπέλαση αρχείων, χειρισμό σχεσιακών βάσεων δεδομένων κτλ.

### 2.2.3 Το περιβάλλον της γλώσσας

Στο σημείο αυτό πρέπει να τονίσουμε ότι η Java είναι μια γενική γλώσσα προγραμματισμού και χρησιμοποιείτε σε ένα ευρύ φάσμα εργασιών. Η δημιουργία προγραμμάτων (applet) που εκτελούνται στον παγκόσμιο ιστό είναι ένας μόνο τομέας όπου η γλώσσα αυτή χρησιμοποιείται. Στις μέρες μας ένας προγραμματιστής μπορεί να χρησιμοποιήσει την Java για την δημιουργία εφαρμογής η οποία στην ορολογία της Sun χαρακτηρίζεται "application". Όπως αναφέραμε η Java μπορεί να δημιουργήσει προγράμματα, τα οποία τρέχουν σε μια ποικιλία υπολογιστών και λειτουργικών συστημάτων. Εάν παρατηρήσουμε τα παρακάτω σχήματα καταλαβαίνουμε πώς λειτουργεί ο μεταγλωττιστής μιας παραδοσιακής γλώσσας και πώς ο μεταγλωττιστής της Java.

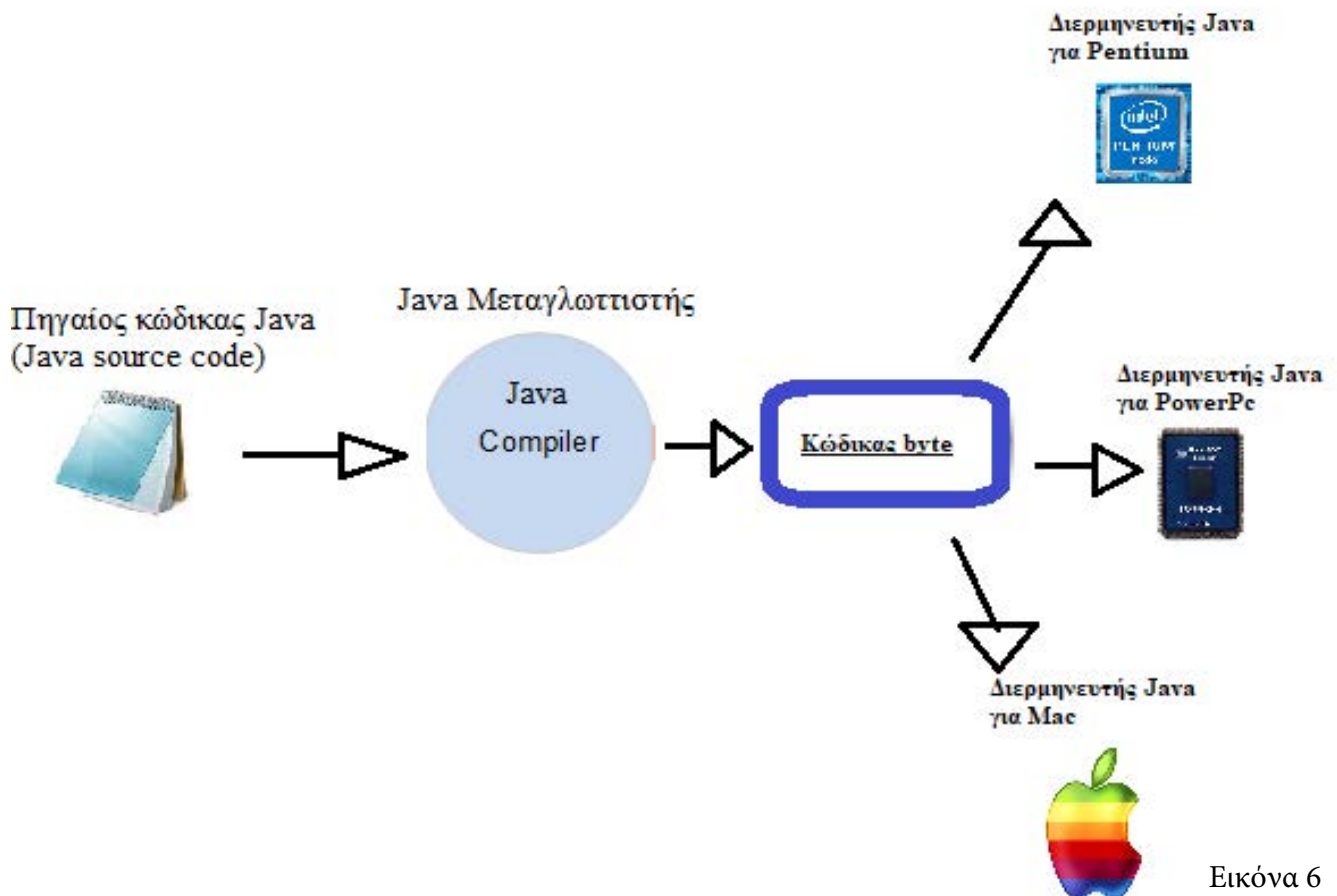


Εικόνα 5

Σε μια παραδοσιακή γλώσσα (Pascal, C, κτλ) ο μεταγλωττιστής μετατρέπει το πρόγραμμα που έχουμε γράψει (source code) σε δυαδικό αρχείο (binary file), δηλαδή σε κώδικα γλώσσας μηχανής κατάλληλο για τον συγκεκριμένο επεξεργαστή.

Η Java από την άλλη μεριά αποτελείται από δύο μέρη στο περιβάλλον ανάπτυξης: α) το μεταγλωττιστή, ο οποίος αντί για δυαδικό αρχείο δημιουργεί ένα είδος κώδικα που είναι ανεξάρτητος από πλατφόρμα και λέγεται κώδικας byte, β) το διερμηνευτή κώδικα byte ο οποίος προσαρμόζεται ανάλογα με την πλατφόρμα και ονομάζεται Εικονική Μηχανή Java (Java Virtual Machine, JVM). Ο διερμηνευτής κώδικα byte διαβάζει τον κώδικα και εκτελεί τις κατάλληλες εντολές γλώσσας μηχανής που αντιστοιχούν στο συγκεκριμένο επεξεργαστή.

Στο παρακάτω σχήμα θα δούμε πώς λειτουργεί ο μεταγλωττιστής της Java.



Εικόνα 6

## 2.2.4 Δημιουργία απλής εφαρμογής Java

Σε αυτή τη παράγραφο θα δημιουργήσουμε ένα απλό πρόγραμμα Java το οποίο θα τρέχει αυτόνομα χωρίς προγράμμα πλοήγησης ή άλλο ειδικό πρόγραμμα.

### Βήμα 1 Πηγαίος κώδικας

Για να γράψουμε τον πηγαίο κώδικα αρκεί ένας απλός συντάκτης κειμένου π.χ.(TextPad, Notepad++). Μέσα στον συντάκτη γράφουμε τον παρακάτω κώδικα. Πρέπει να δώσουμε προσοχή στην πληκτρολόγηση, γιατί στην Java έχουν διαφορά τα κεφαλαία απο τα πεζά γράμματα.

```

hello.java x
1 public class hello
2 {
3     public static void main (String args[])
4     {
5         System.out.println("Hello World!");
6     }
7 }
    
```

Εικόνα 7

Αποθηκεύουμε το αρχείο με το όνομα hello γιατί το όνομα του αρχείου πρέπει να είναι το ίδιο με αυτό που υπάρχει στον ορισμό της κλάσης. Επίσης το αρχείο μας πρέπει να έχει κατάληξη .java.

## Βήμα 2 Μεταγλώττιση

Για την μεταγλώττιση του αρχείου μας χρειαζόμαστε ένα παράθυρο εντολών. Στην περίπτωση που το λειτουργικό μας σύστημα είναι Windows ανοίγουμε το παράθυρο της γραμμής εντολών. Στην δική μας περίπτωση έχουμε αποθηκεύσει το αρχείο του πηγαίου κώδικα στον υποκατάλογο

```
C:\Users\obi wan>
```

## Βήμα 3 Εκτέλεση

Για να μεταγλωττίσουμε το αρχείο του πηγαίου κώδικα hello.java γράφουμε στην γραμμή εντολών : **javac hello.java** **javac** ονομάζεται ο μεταγλωττιστής της Java (Java compiler) ο οποίος δημιουργεί ένα αρχείο κώδικα byte με προέκταση .class και ίδιο όνομα με το αρχείο του πηγαίου κώδικα hello.class. Αν θέλουμε μπορούμε να δούμε το αρχείο με την εντολή **dir hello.\***

Για να εκτελέσουμε το αρχείο του κώδικα byte **hello.class** καλούμε τον διερμηνευτή **Java** με το αρχείο κώδικα byte **hello** γράφοντας την εντολή **java hello**.

Στην εικόνα 8 φαίνεται η διαδικασία που ακολουθήσαμε.

```

C:\Users\obi wan>set path=C:\Program Files\Java\jdk1.8.0_111\bin;
C:\Users\obi wan>javac hello.java
C:\Users\obi wan>dir hello.*
Volume in drive C has no label.
Volume Serial Number is E8A2-DA87

Directory of C:\Users\obi wan

    02:55 PM                416 hello.class
    12:47 PM                115 hello.java
           2 File(s)                531 bytes
           0 Dir(s) 123,767,250,944 bytes free
C:\Users\obi wan>java hello
Hello World!
C:\Users\obi wan>
  
```

Εικόνα 8

Στην συνέχεια θα εξετάσουμε λίγο πιο αναλυτικά τον κώδικα που γράψαμε για να κατανοήσουμε καλύτερα την έννοια της μεθόδου αλλά και πώς ορίζουμε μια κλάση.

Κάθε εφαρμογή της Java ξεκινά πάντα με τον ορισμό μιάς κλάσης, η οποία περιέχει μια μέθοδο που λέγεται **main()**. Η κλάση μπορεί να έχει οποιοδήποτε όνομα, η μέθοδος όμως που εκτελείται πρώτη πρέπει να έχει πάντα το όνομα **main()**. Μπορούμε να πούμε ότι μέθοδος είναι μια συνάρτηση η οποία εμπεριέχεται σε ένα αντικείμενο, και καθορίζει μια λειτουργία του ή μια ιδιότητα του. Η γραμμή 5 βλέπουμε οτι τελειώνει με ";" αυτό συμβαίνει γιατί στην γραμμή 5 υπάρχει η εντολή **System.out.println("Hello World!")** και όχι ορισμός όπως για παράδειγμα στην γραμμή 3. **System** είναι το όνομα μιας βασικής κλάσης, η οποία περιλαμβάνει αντικείμενα και μεταβλητές για υποστήριξη εισαγωγής δεδομένων απο το πληκτρολόγιο, καθώς και για την έξοδο χαρακτήρων στην οθόνη. Το αντικείμενο **out** είναι μέλος της κλάσης **System** και δηλώνει μια τυπική έξοδο προς την οθόνη. Η μέθοδος **println** τυπώνει στην οθόνη αυτό που βρίσκεται μέσα στην παρένθεση ("Hello World") και είναι μέθοδος του αντικειμένου **out**.

**hello.java**

```

1 public class hello
2 {
3     public static void main (String args[])
4     {
5         System.out.println("Hello World!");
6     }
7 }
    
```

**Ορισμός της κλάσης hello.** Ο ορισμός περιλαμβάνει μια μόνο μέθοδο, την main()

**Οι αγκύλες πηγαίνουν ανά ζεύγη και περικλείουν αυτόνομα κομμάτια κώδικα.**

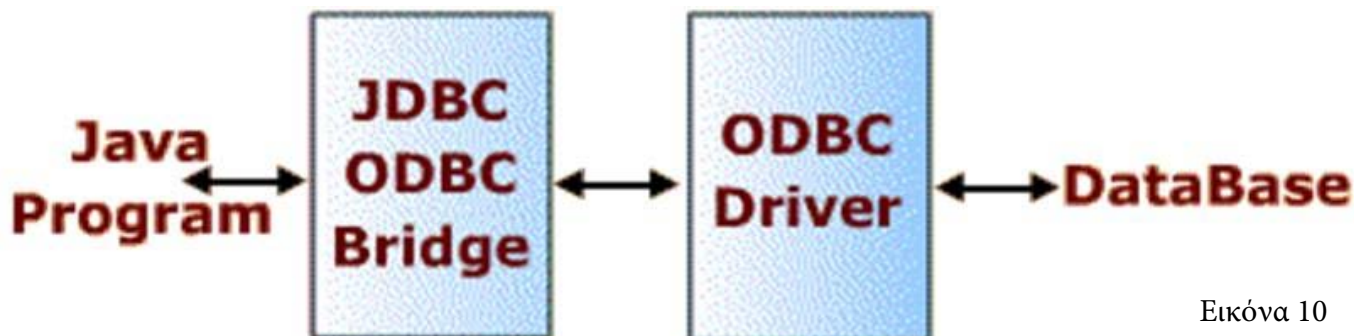
**Ορισμός της μεθόδου main().** Η λέξη κλειδί **public** δηλώνει ότι η μέθοδος είναι προσπελάσιμη απο παντού. Η λέξη κλειδί **static** δηλώνει ότι η μέθοδος είναι προσπελάσιμη, ακόμα κι αν δεν υπάρχουν αντικείμενα της κλάσης. Η λέξη κλειδί **void** σημαίνει ότι δεν υπάρχει επιστροφή τιμής

Εικόνα 9

## 2.3 Μέθοδος αλληλεπίδρασης Java και SQL

Η αλληλεπίδραση μεταξύ Java και SQL γίνεται μέσω μιάς διεπαφής η οποία ονομάζεται JDBC (Java, Database, Connectivity). Το JDBC είναι ένα πακέτο απο συναρτήσεις οι οποίες επιτρέπουν στον χρήστη να έχει πρόσβαση σε μία βάση με την ιδιότητα πρόσθεσης, εξαγωγής, ανανέωσης ή διαγραφής δεδομένων. Ο λόγος που δημιουργήθηκε το JDBC ήταν γιατί υπήρξε ανάγκη γεφύρωσης απο το περιβάλλον της Java στο περιβάλλον οποιασδήποτε βάσης δεδομένων χρησιμοποιεί την διεπαφή ODBC (Open, Database, Connectivity) η οποία όμως είναι γραμμένη σε γλώσσα C.

Στο σχήμα φαίνεται η διαδρομή που ακολουθούν τα δεδομένα απο μία εφαρμογή Java στην βάση δεδομένων και αντίστροφα.



Εικόνα 10



### 3 Παρουσίαση MySQL Workbench και Java Netbeans

Στο καιφάλαιο 3 γίνεται παρουσίαση και ανάλυση στις ψηφιακές πλατφόρμες MySQL Workbench και Java Netbeans οι οποίες θα χρησιμοποιηθούν ως εργαλεία για την γρηγορότερη σχεδίαση της βάσης δεδομένων αλλά και της εφαρμογής με την οποία ο χρήστης θα διαχειρίζεται την βάση.

#### 3.1 MySQL Workbench

Το MySQL Workbench είναι ένα ενιαίο οπτικό εργαλείο για αρχιτέκτονες και διαχειριστές βάσεων δεδομένων. Ο προγραμματιστής ή ο διαχειριστής μπορούν με την βοήθεια της συγκεκριμένης πλατφόρμας να απλοποιούν χρονοβόρες και επιρρεπείς σε σφάλματα διαδικασίες όπως για παράδειγμα μοντελοποίηση δεδομένων, διαχείριση χρηστών, δημιουργία αντιγράφων ασφαλείας και πολλά άλλα.

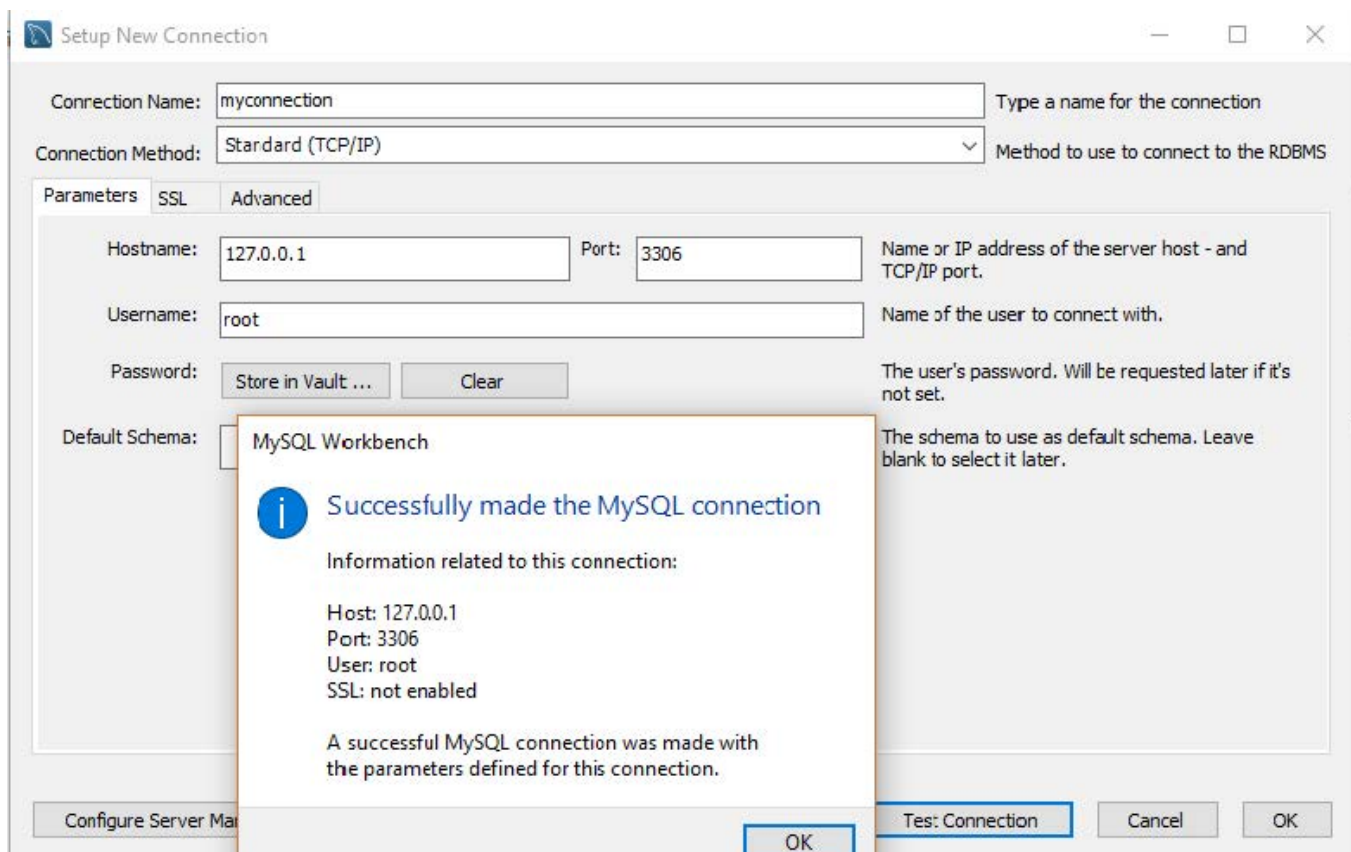
##### 3.1.2 Διαδικασία εγκατάστασης MySQL Workbench

Για να μπορέσουμε να εγκαταστήσουμε και να εκτελέσουμε το MySQL Workbench στα Windows χρειάζονται ορισμένες προϋποθέσεις όπως για παράδειγμα το Microsoft .NET Framework και το Visual C++ Redistributable for Visual Studio 2015. Σε αυτό το σημείο πρέπει να αναφέρουμε ότι το MySQL Workbench είναι ικανό να "τρέξει" και σε άλλα λειτουργικά συστήματα όπως είναι τα Linux και τα MacOS.

Αρχικά κάνουμε λήψη της έκδοσης που μας εξυπηρετεί από την σελίδα <https://dev.mysql.com/downloads/workbench>. Στην συνέχεια βρίσκουμε και εκτελούμε το πακέτο που περιλαμβάνει όλα τα απαραίτητα αρχεία για την εγκατάσταση της εφαρμογής. Κατά την διάρκεια της εγκατάστασης ένας οδηγός ρυθμίσεων θα μας δώσει την δυνατότητα να επιλέξουμε μονάχα τις εφαρμογές που μας εξυπηρετούν κερδίζοντας χρόνο αλλά και χώρο από την αποθήκευση αρχείων τα οποία δεν θα χρησιμοποιήσουμε.

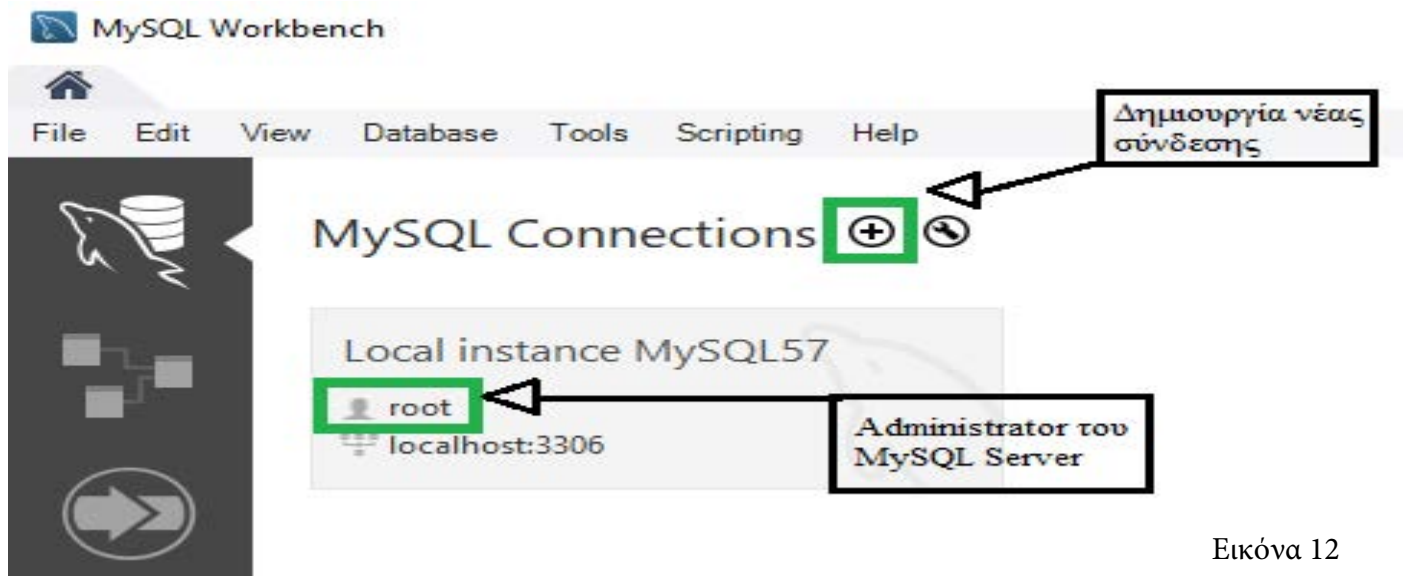
##### 3.1.3 Σύνδεση Workbench με MySQL Server

Ξεκινώντας το MySQL Workbench βλέπουμε στο αρχικό μενού την επιλογή MySQL Connections. Επιλέγουμε το σύμβολο [+] για την δημιουργία νέας σύνδεσης με τον SQL Server η οποία θα μας επιτρέψει στην συνέχεια να έχουμε πρόσβαση στην βάση δεδομένων. Δίνουμε ένα όνομα στην νέα μας σύνδεση πχ **Connection name: myconnection** και πατάμε **Test:Connection** το αποτέλεσμα φαίνεται στην εικόνα 11.



Εικόνα 11

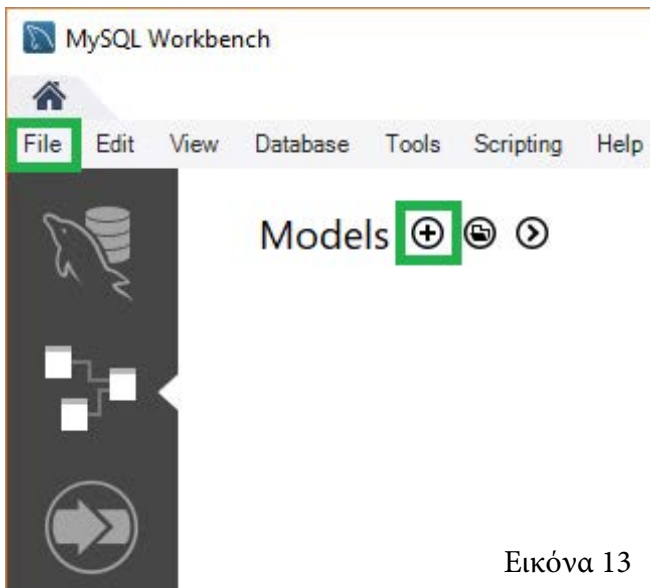
Στο παράδειγμα μας αφήσαμε για ευκολία Username το όνομα root χωρίς να δώσουμε password. Αν δεν ορίσουμε password και αλλάξουμε το Username της σύνδεσης σε πραγματικές συνθήκες αφήνουμε την βάση δεδομένων τρομερά εκτεθειμένη.



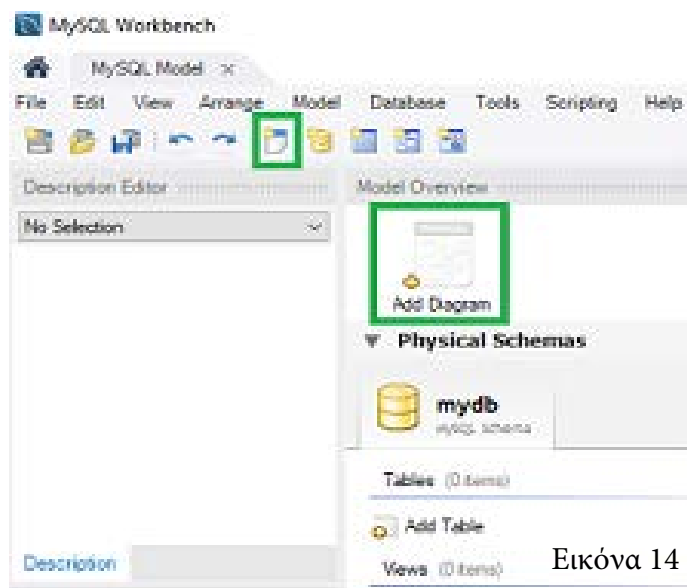
Εικόνα 12

### 3.1.4 Περιβάλλον σχεδίασης της βάσης δεδομένων (Data Modeling)

Σε αυτή την παράγραφο θα παρουσιάσουμε το περιβάλλον στο οποίο ο χρήστης του Workbench μπορεί εύκολα αλλά και γρήγορα να σχεδιάσει και να υλοποιήσει την βάση δεδομένων. Για να προβάλλουμε το περιβάλλον που θα σχεδιάσουμε την βάση όπως βλέπουμε στην εικόνα 13 και 14 επιλέγουμε File->New Model ή το σύμβολο [+] και στην συνέχεια Add Diagram.

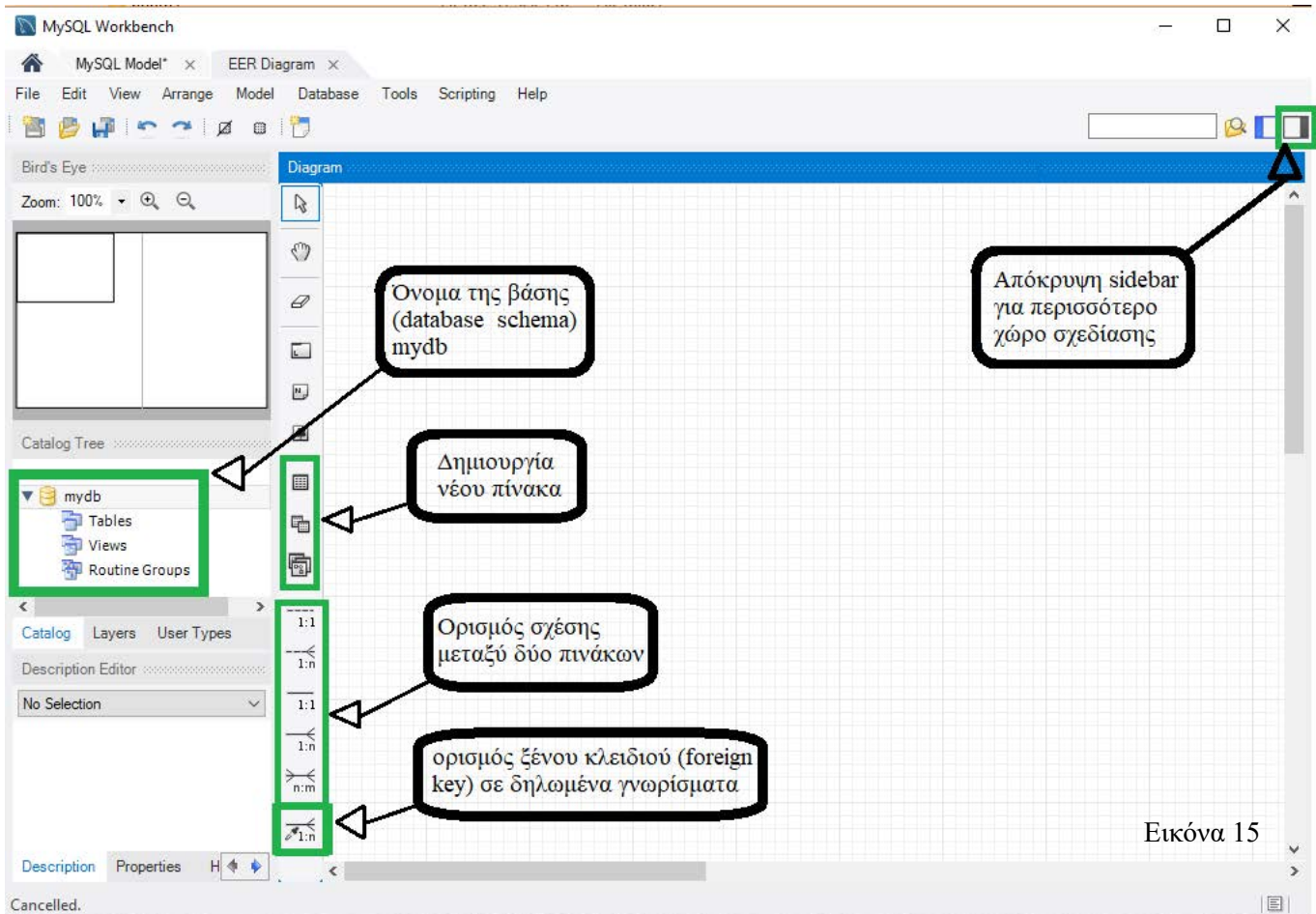


Εικόνα 13



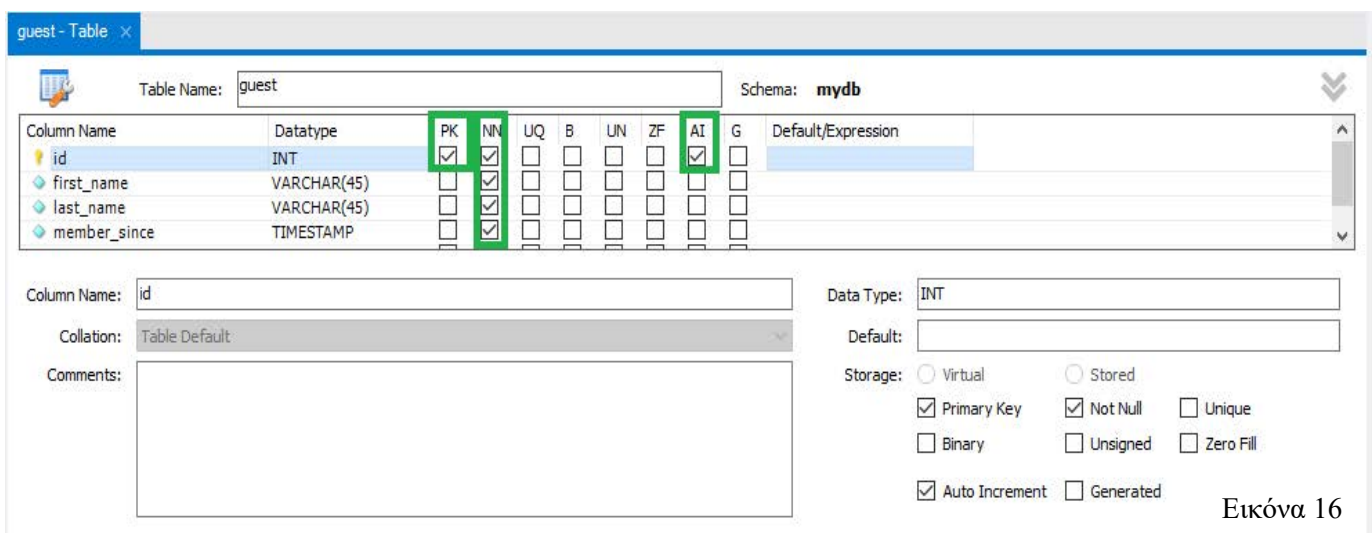
Εικόνα 14

Το αποτέλεσμα φαίνεται στην εικόνα 15 όπου αναφέρουμε μερικές από τις κύριες επιλογές στην περιοχή σχεδίασης Diagram.



Εικόνα 15

Στο παρακάτω παράδειγμα δημιουργήσαμε ένα απλό μοντέλο βάσης δεδομένων με δύο πίνακες τους οποίους ονομάσαμε guest και reservation. Στην συνέχεια προσθέσαμε τέσσερις στήλες (Columns ή γνωρίσματα/ attributes) στον πίνακα guest (id, first\_name, last\_name, member\_since) και πέντε στον πίνακα reservation (id, date\_in, date\_out, status, made\_by). Στην εικόνα 16 βλέπουμε τις βασικές ρυθμίσεις του πίνακα guest.



Εικόνα 16

Παρατηρούμε ότι ο τύπος δεδομένων (Datatype) της στήλης id είναι INT με αυτό τον τρόπο εισάγονται μονάχα ακέραιοι. Οι στήλες first\_name και last\_name είναι δηλωμένες VARCHAR(45) δηλαδή αλφαριθμητικό το οποίο δέχεται έως 45 χαρακτήρες τέλος ο τύπος TIMESTAMP της στήλης member\_since δηλώνει ότι θα καταχωρείτε η ημερομηνία του συστήματος (Server) στο οποίο είναι συνδεδεμένη η βάση δεδομένων. Ως πρωτεύων κλειδί του πίνακα ορίσαμε την στήλη id στην οποία επιλέξαμε τρία checkboxes δηλώνοντας με αυτόν τον τρόπο τα εξής:



**PK:** (Primary Key) Είναι το πρωτεύων κλειδί του πίνακα. Στις περισσότερες περιπτώσεις όλοι οι πίνακες μιας βάσης έχουν μια στήλη με την οποία δηλώνουν την μοναδικότητα κάθε εγγραφής (πλειάδας) που εκχωρείται στον πίνακα. Με αυτόν τον τρόπο όταν επιλέξουμε μια στήλη ως πρωτεύων κλειδί αυτόματα επιλέγεται και το checkbox NN (NotNull).

**NN:**(NotNull) Δηλώνουμε ότι κάθε πεδίο της στήλης πρέπει να έχει τιμή.

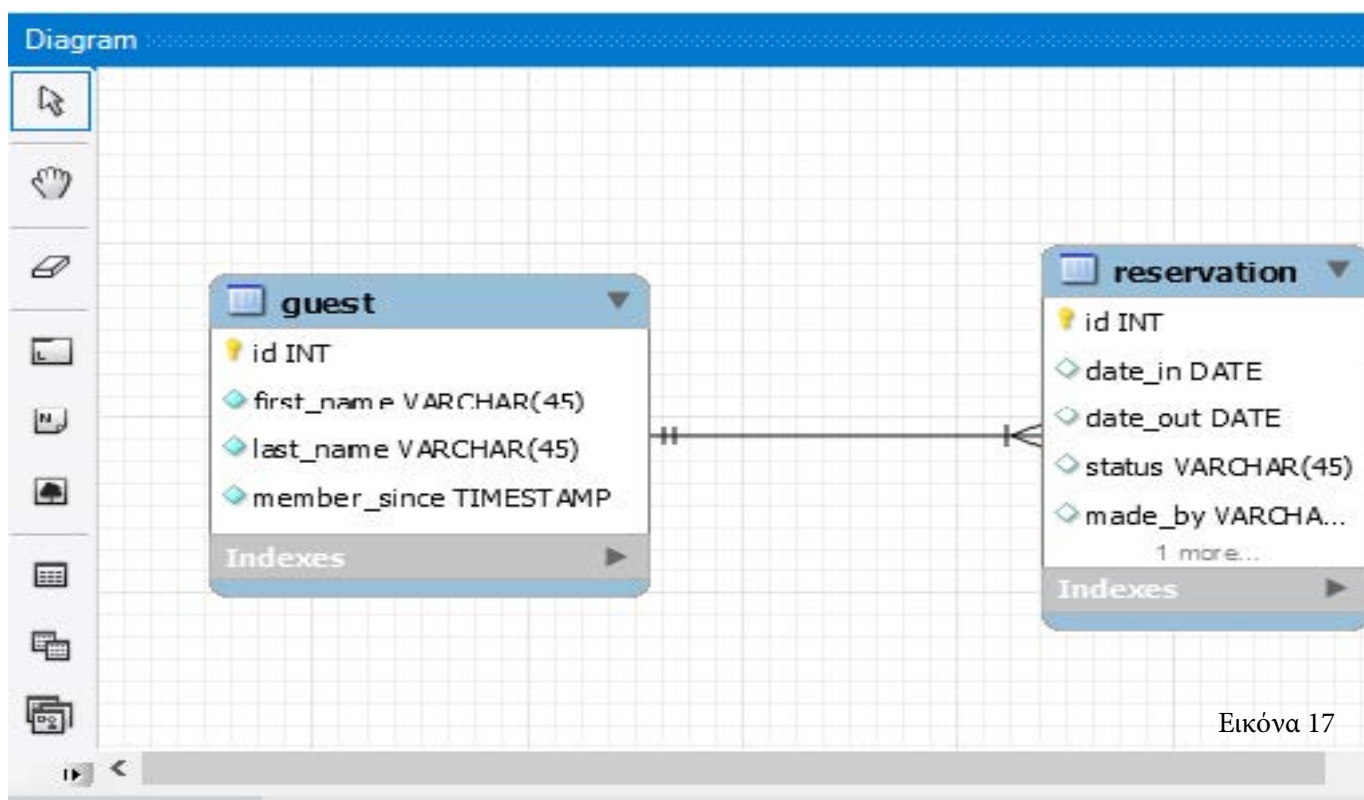
**UQ:** (Unique Index) Δεν επιτρέπει να υπάρχουν δύο ή περισσότερα πεδία με την ίδια τιμή σε μια στήλη. Το πρωτεύων κλειδί έχει εξορισμού αυτή την ιδιότητα και δεν απαιτείται η συγκεκριμένη ρύθμιση.

**B:** Σε πεδία τύπου CHAR, VARCHAR, TEXT τα πεζά από τα κεφαλαία γράμματα θα διαφοροποιούνται.

**UN:** (Unsigned) Αφορά τύπους δεδομένων INT. Ζητάμε μόνο θετικούς επιρεάζοντας το επιτρεπτό εύρος τιμών π.χ. αντί για εύρος -127 ως 128 έχουμε 0 ως 255.

**AI:**(Auto Incremental) Αφορά τυπους δεδομένων INT με αυτόν τον τρόπο δηλώνουμε ότι οι εγγραφές που εκχωρούνται στην στήλη θα παίρνουν άυξοντα αριθμό.

Στην εικόνα 17 βλέπουμε τους πίνακες guest και reservation τους οποίους ενώνει μια σχέση 1 προς πολλά. Πρακτικά αυτό σημαίνει ότι ένας πελάτης μπορεί να κάνει πολλές κρατήσεις.



Εικόνα 17

### Σχέσεις μεταξύ πινάκων.



Ένα προς ένα



Ένα προς πολλά



Πολλά προς πολλά

Για την μετατροπή του οπτικού μοντέλου σε πραγματική βάση δεδομένων (schema) η οποία είναι ικανή να συνδεθεί με τον MySQL Server επιλέγουμε από το κύριο μενού **Database->Forward Engineer**. Με αυτόν τον τρόπο επιλέγουμε τις ρυθμίσεις της σύνδεσης και στην συνέχεια ο κώδικας SQL της βάσης δεδομένων δημιουργείται αυτόματα.

## 3.2 Java Netbeans

Το Netbeans είναι ένα δωρεάν οπτικό περιβάλλον ανοιχτού κώδικα για την ολοκληρωμένη ανάπτυξη λογισμικού. Παρέχει όλα τα εργαλεία που χρειάζεται ένας προγραμματιστής για να δημιουργήσει μια επαγγελματική επιφάνεια εργασίας όπως είναι για παράδειγμα προγράμματα για επιχειρήσεις, εφαρμογές web και κινητών συσκευών. Υποστηρίζει γλώσσες προγραμματισμού java, javascript, C/C++, php, xml, HTML κτλ

### 3.2.1 Διαδικασία εγκατάστασης NetBeans IDE

Αρχικά κατεβάζουμε την τελευταία έκδοση του NetBeans IDE 8.2 από την ηλεκτρονική διεύθυνση <http://download.netbeans.org/netbeans/8.0.2/final/>.

The screenshot shows the NetBeans IDE 8.0.2 Download page. The page has a navigation bar with 'NetBeans IDE' selected. Below the navigation bar, there is a search bar and a 'HOME / Download' link. The main content area is titled 'NetBeans IDE 8.0.2 Download' and includes a form for email subscription and language/platform selection. The 'All' download bundle is highlighted with a green box.

Supported technologies *	Java SE	Java EE	C/C++	HTML5 & PHP	All
NetBeans Platform SDK	•	•			•
Java SE	•	•			•
Java FX	•	•			•
Java EE		•			•
Java ME					•
HTML5		•		•	•
Java Card™ 3 Connected					•
C/C++			•		•
Groovy					•
PHP				•	•
Bundled servers					
GlassFish Server Open Source Edition 4.1		•			•
Apache Tomcat 8.0.15		•			•

Download buttons: Java SE (Free, 90 MB), Java EE (Free, 186 MB), C/C++ (Free, 63 MB), HTML5 & PHP (Free, 63 MB), All (Free, 205 MB).

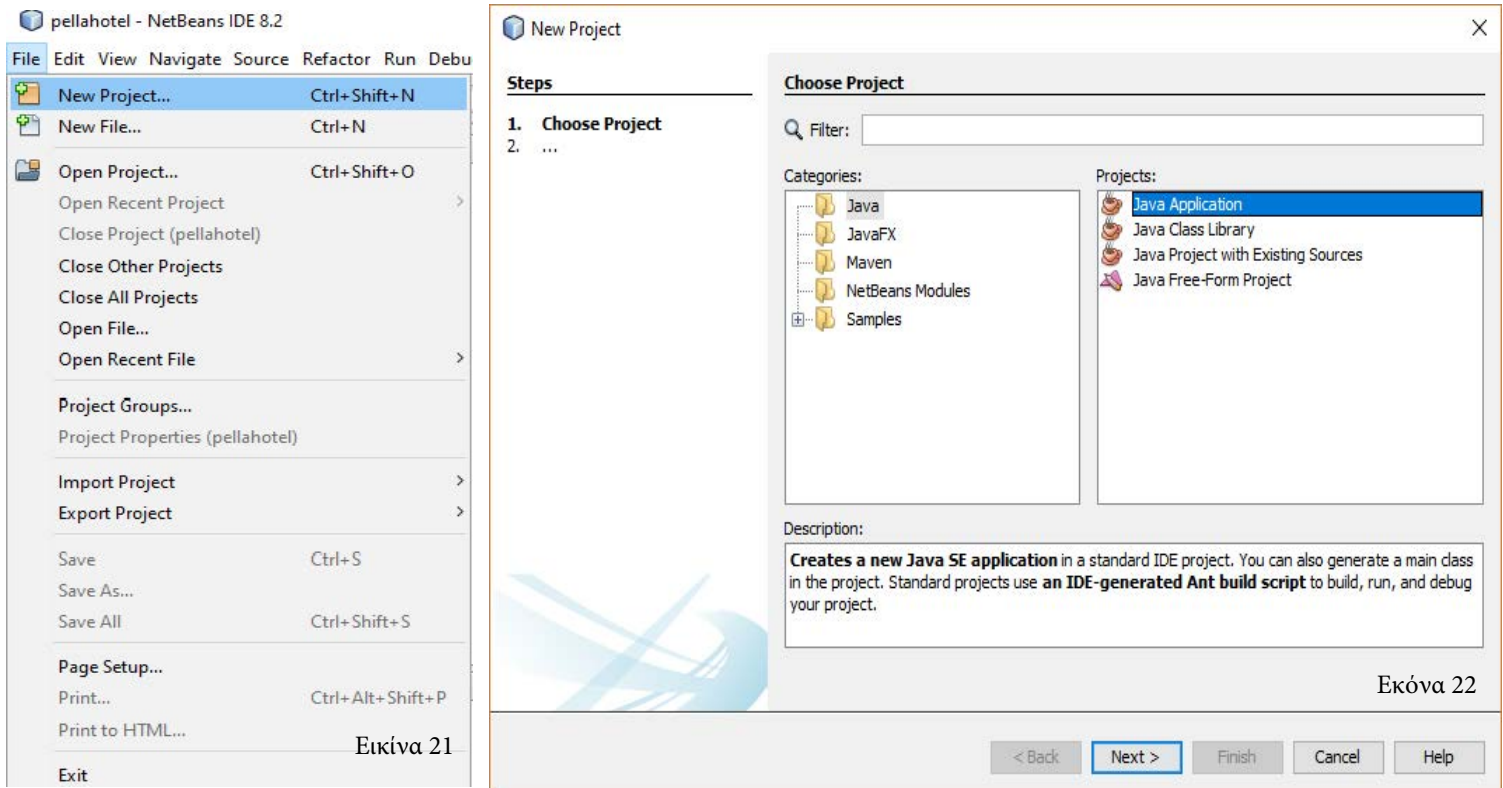
Εικόνα 18

Στην συνέχεια επιλέγουμε το αρχείο που κατεβάσαμε και το κάνουμε εγκατάσταση install στον υπολογιστή μας. Ο οδηγός εγκατάστασης μας δίνει την δυνατότητα να επιλέξουμε μονάχα τις γλώσσες προγραμματισμού που μας εξυπηρετούν.

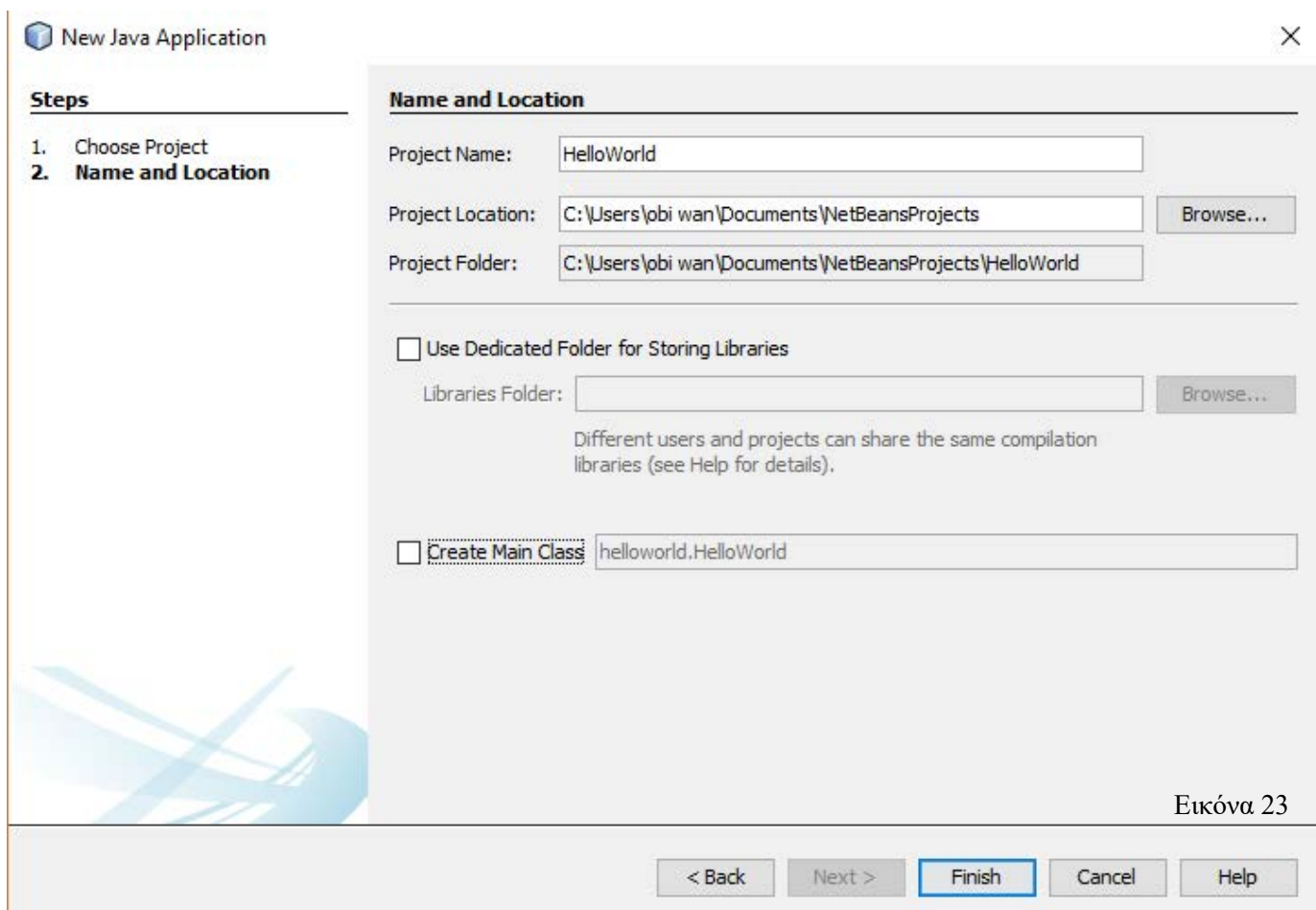
### 3.2.2 Δημιουργία project γραμμένο σε κώδικα Java με την βοήθεια του NetBeans IDE

Σε αυτή την παράγραφο θα δημιουργήσουμε ένα project και θα δείξουμε με αυτόν τον τρόπο την ευκολία που προσφέρει το NetBeans σε έναν προγραμματιστή κατά την σχεδίαση και υλοποίηση μιας εφαρμογής. Στην συγκεκριμένη εφαρμογή θα πατάμε ένα κπουμπί το οποίο θα γράφει επάνω του την κλασική φράση "Hello World" και στην συνέχεια η φράση θα τυπώνεται σε ένα πεδίο κειμένου. Θα γίνει αναφορά στις κυριότερες κλάσεις αλλά και στην βιβλιοθήκη Swing η οποία περιλαμβάνει πολλά συστατικά για την δημιουργία γραφικού περιβάλλοντος επικοινωνίας (GUI, Graphical, User, Interface) όπως είναι για παράδειγμα κουμπιά, πλαίσια ελέγχου, ετικέτες, περιγράμματα, τμήματα παραθύρων με δυνατότητες κύλισης, μενού, γραμμές εργαλείων και πολλά άλλα. Τα συστατικά του Swing έχουν όνομα που ξεκινά με **J** π.χ. JFrame, JPanel, JTextField, JButton.

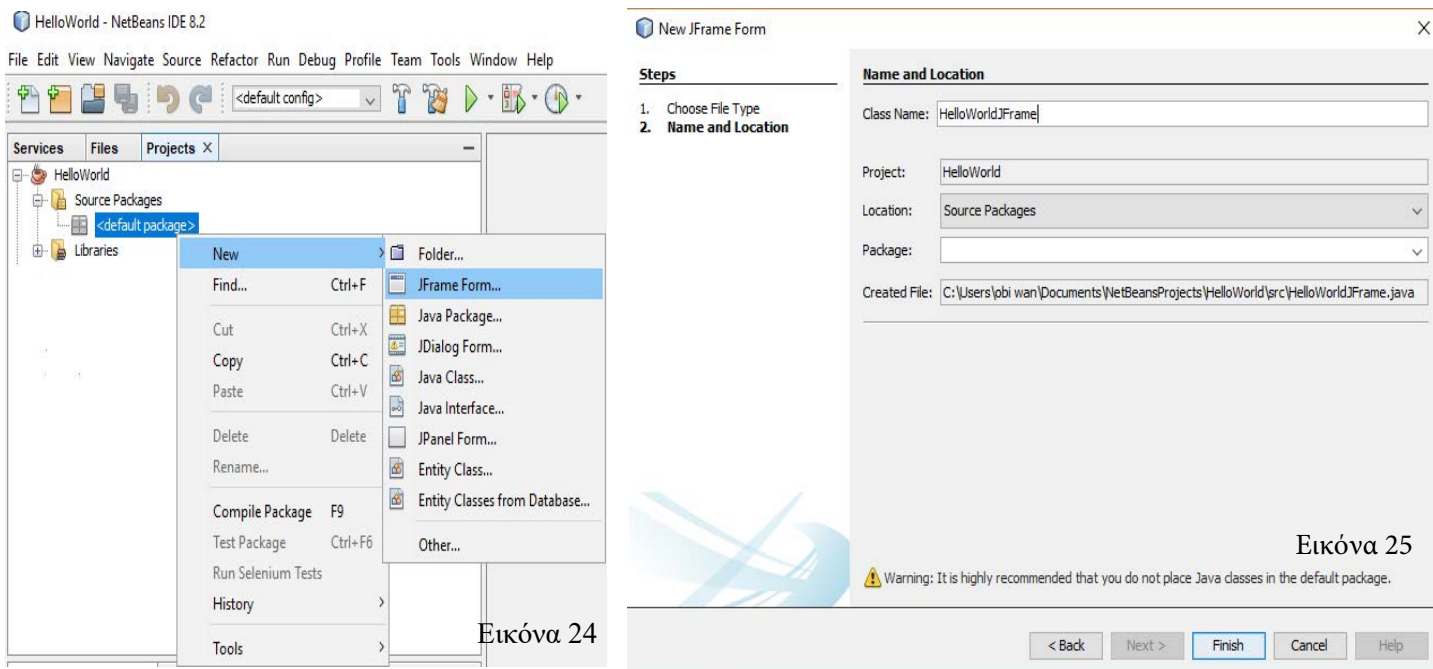
Το πρώτο βήμα είναι η δημιουργία του project. Επιλέγουμε απο το κεντρικό μενού του NetBeans **File->New Project** εικόνα 21 και στο παράθυρο που θα εμφανιστεί εικόνα 22 Java Application



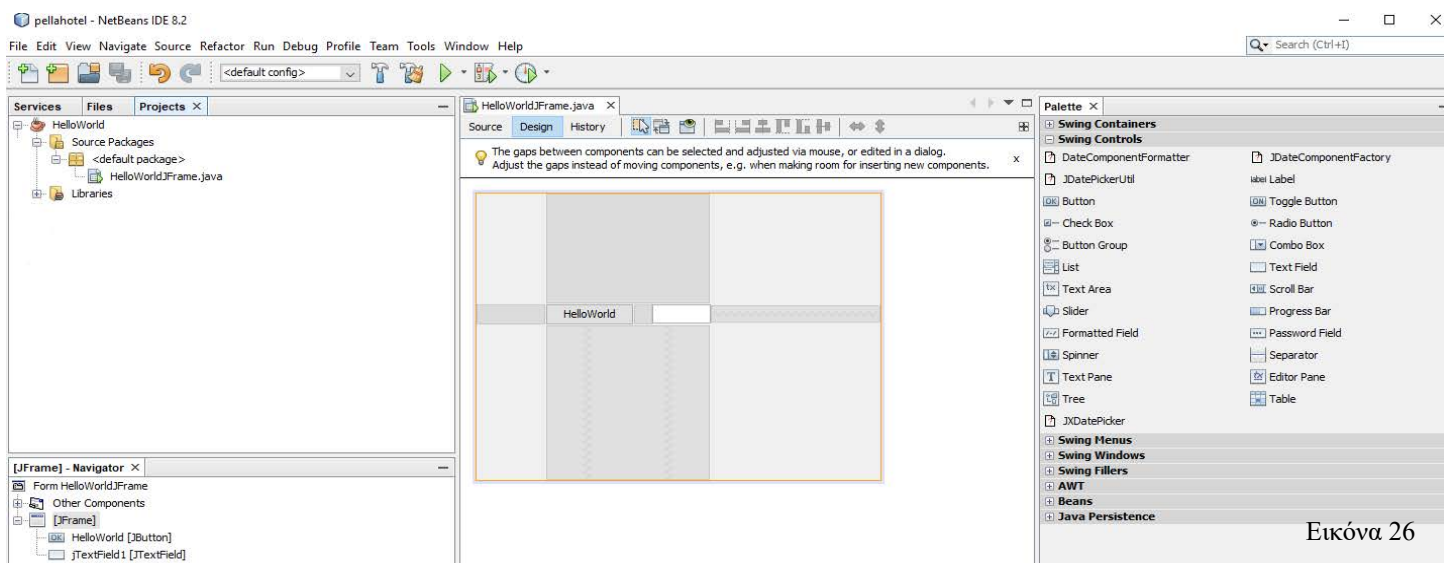
Στην συνέχεια όπως βλέπουμε στην εικόνα 23 γράφουμε το όνομα και επιλέγουμε τον φάκελο που θα αποθηκεύονται τα δεδομένα του project. Στην συγκεκριμένη περίπτωση το επιλεγμένο checkbox "Create Main Class" το ξαναεπιλέγουμε για να είναι κενό.



Το δεύτερο βήμα είναι να δημιουργήσουμε ένα πλαίσιο Swing με την χρήση ενός JFrame και στην συνέχεια να προσθέσουμε ένα κουμπί JButton και ένα πεδίο κειμένου JTextField. Στην εικόνα 24 βλέπουμε στο παράθυρο **Projects** την νέα μας εφαρμογή "HelloWorld", κάνουμε δεξί κλικ στον υποφάκελο **<default package>** και επιλέγουμε **New-> JFrame Form**. Το JFrame (κλάση) που θα προσθέσουμε στην εφαρμογή μας θα έχει το όνομα "HelloWorldJFrame" εικόνα 25.



Στην συνέχεια απο το παράθυρο **Palette** επιλέγουμε **Swing Controls** και τοποθετούμε στην περιοχή **Design** του HelloWorldJFrame ένα JButton και ένα JTextField εικόνα 26. Αρχικά το κουμπί και το πεδίο κειμένου θα έχουν όνομα JButton1, JTextField1 αντίστοιχα. Για να αλλάξουμε το όνομα του κουμπιού στην περιοχή **Design** κάνουμε δεξί κλικ στο JButton1 και επιλέγουμε **Change Variable Name**. Αφού δώσουμε νέο όνομα "HelloWorld" στο κουμπί κάνουμε διπλό κλικ και με αυτόν τον τρόπο πολύ απλά και χωρίς λάθη ο κώδικας της κλάσης JButton η οποία κληρονομεί απο την κλάση AbstractButton δημιουργείται αυτόματα. Για να δούμε τον κώδικα της εφαρμογής μας επιλέγουμε απο το παράθυρο HelloWorldJFrame Source.



Το τρίτο βήμα είναι ο χειρισμός του **Ctrl+G** που θα δημιουργεί ο χρήστης μόλις πατήσει το κουμπί ώστε να εμφανίζεται στο πεδίο κειμένου η φράση HelloWorld. Στην εικόνα 27 βλέπουμε τον κώδικα της εφαρμογής μας.



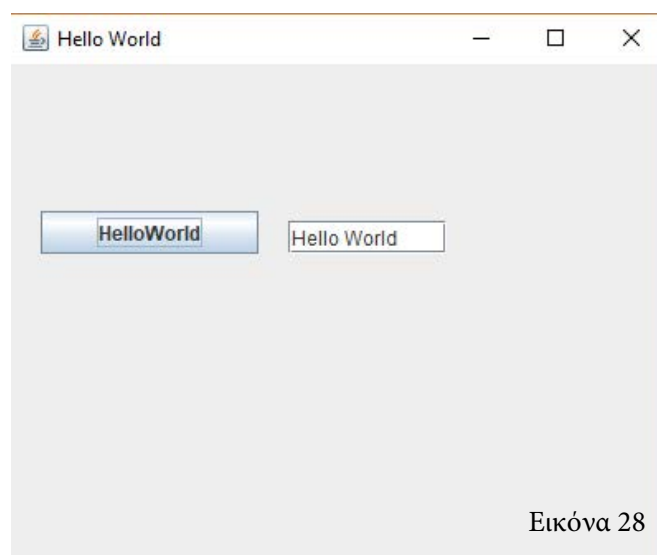
```

1  public class HelloWorldJFrame extends javax.swing.JFrame {
2
3
4
5  public HelloWorldJFrame() {
6      initComponents();
7  }
8
9
10 @SuppressWarnings("unchecked")
11 Generated Code
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56 private void HelloWorldActionPerformed(java.awt.event.ActionEvent evt) {
57     jTextField1.setText("Hello World");
58 }
59 private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
60     // TODO add your handling code here:
61 }
62
63
64 public static void main(String args[]) {
65
66     java.awt.EventQueue.invokeLater(new Runnable() {
67         public void run() {
68             new HelloWorldJFrame().setVisible(true);
69         }
70     });
71 }
72
73 // Variables declaration - do not modify
74 private javax.swing.JButton HelloWorld;
75 private javax.swing.JTextField jTextField1;
76 // End of variables declaration
77 }

```

Εικόνα 27

Αρχικά να πούμε ότι ο κώδικας από την γραμμή 2 έως 77 με εξαίρεση την γραμμή 57 δημιουργήθηκε από το NetBeans. Στην γραμμή 2 η κλάση HelloWorldJFrame κληρονομεί από το πακέτο javax.swing.JFrame. Πρακτικά αυτό σημαίνει ότι η κλάση HelloWorld JFrame μπορεί να χρησιμοποιεί κλάσεις υποκλάσεις και συστατικά του Swing με την χρήση της μεθόδου getContentPane() η οποία βρίσκεται μέσα στην μέθοδο initComponents() στις γραμμές 12 έως 54. Η μέθοδος initComponents() περιέχει όλες τις απαραίτητες κλάσεις και μεθόδους με τις οποίες εξασφαλίζεται η λειτουργικότητα των συστατικών JButton, JTextField που προσθέσαμε στην περιοχή **Design**. Στην γραμμή 56 έως 57 βρίσκεται η μέθοδος διασύνδεσης HelloWorldActionPerformed για τον χειρισμό του γεγονότος που θα δημιουργηθεί όταν ο χρήστης πατήσει το κουμπί. Στην γραμμή 57 κάνουμε χρήση της μεθόδου setText() με αυτόν τον τρόπο η φράση "Hello World" θα τυπώνεται στο πεδίο κειμένου JTextField1. Τέλος στις γραμμές 64 έως 71 βρίσκεται η κύρια μέθοδος του προγράμματος main() η οποία θα εκτελεστεί πρώτη και με την χρήση της run() θα δημιουργήσει ένα νέο JFrame το οποίο εμφανίσει με την μέθοδο setVisible(True). Το αποτέλεσμα φαίνεται στην εικόνα 28.



Εικόνα 28

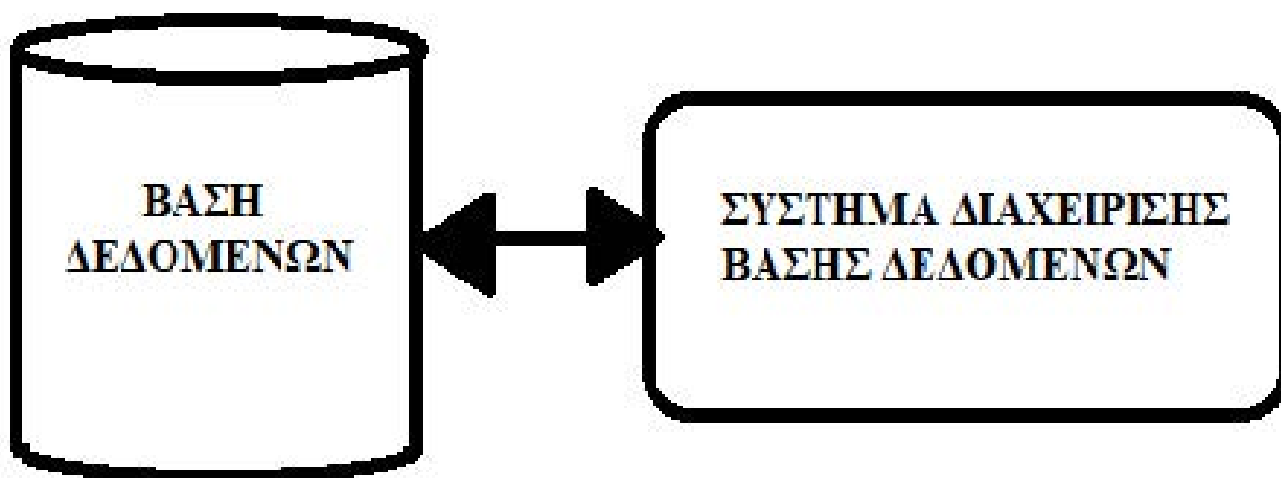
## 4 Ανάλυση σχεδίαση και υλοποίηση Εφαρμογής Διαχείρισης Ξενοδοχείου Πόλης

Σε αυτό το κεφάλαιο θα σχεδιάσουμε και θα υλοποιήσουμε ένα σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΔ) για την διαχείριση κρατήσεων ενός ξενοδοχείου πόλης. Ψάχνουμε για ένα μοντέλο δεδομένων όπου μπορούμε να αναπαριστούμε πληροφορίες σχετικά με τα δωμάτια, τους επισκέπτες και τις κρατήσεις. Αφού σχεδιάσουμε το μοντέλο θα υλοποιήσουμε την βάση δεδομένων σε SQL με την βοήθεια του WorkBench. Στην συνέχεια κάνοντας χρήση του Java NetBeans θα δημιουργήσουμε μια εφαρμογή για να μπορεί ο χρήστης να εισάγει και να εξάγει πληροφορίες από την βάση.

### 4.1 Ανάλυση Προβλήματος

Ένα (ΣΔΒΔ) ουσιαστικά παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και την βάση δεδομένων. Κύριος στόχος του είναι η εξυπηρέτηση και των δύο όσο το δυνατόν γρηγορότερα.

Στην εικόνα 29 βλέπουμε πώς αλληλεπιδρούν τα δεδομένα ανάμεσα στο (ΣΔΒΔ) και την βάση δεδομένων.



Εικόνα 29

Κατά την σχεδίαση και υλοποίηση ενός συστήματος διαχείρισης βάσης δεδομένων μπορεί να προκύψουν τα εξής προβλήματα:

#### 1. Πλεονασμός δεδομένων

Υπάρχει περίπτωση να έχουμε επανάληψη των ίδιων δεδομένων σε διαφορετικούς πίνακες (π.χ. έχουμε ένα πίνακα πελατών και ένα πίνακα κρατήσεων είναι σχεδόν σίγουρο πως θα υπάρχουν στοιχεία πελατών που είναι ίδια και στους δύο πίνακες).

#### 2. Ασυνέπεια δεδομένων

Όταν έχουμε πλεονασμό δεδομένων και χρειασθεί κάποια αλλαγή (π.χ αλλαγή στο email ενός πελάτη) τότε είναι πολύ πιθανό να γίνει διόρθωση στο ένα μόνο αρχείο και όχι στο άλλο.

#### 3. Αδυναμία μερισμού δεδομένων

Μερισμός δεδομένων πρακτικά σημαίνει να έχουμε την δυνατότητα για κοινή χρήση των δεδομένων συλλέγοντας τα από διαφορετικούς πίνακες(π.χ. κοιτάζοντας την κράτηση ενός πελάτη να μπορούμε να δούμε και το υπόλοιπό του). Η αδυναμία μερισμού δεδομένων δημιουργεί καθυστέρηση στην λήψη αποφάσεων.

#### 4. Αδυναμία προτυποποίησης

Έχει να κάνει με την ανομοιομορφία και την διαφορετική οργάνωση δεδομένων σε αρχεία. Η συγκεκριμένη αδυναμία δημιουργεί προβλήματα κυρίως στην ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.

Όλα τα παραπάνω προβλήματα πρέπει να ληφθούν σοβαρά υπόψη κατά την ανάλυση απαιτήσεων ενός (ΣΔΒΔ). Οι προδιαγραφές πρέπει να καθοριστούν όσο το δυνατόν σαφέστερα και πληρέστερα σε αντίθετη περίπτωση οι αλλαγές κατά την διάρκεια της υλοποίησης καταναλώνουν χρόνο και πόρους.

## 4.1.2 Ανάλυση απαιτήσεων

Σε αυτή την παράγραφο θα δούμε και θα αναλύσουμε τα βασικά χαρακτηριστικά της εφαρμογής μας. Σκοπός της εφαρμογής είναι να πληρεί τις εξής προϋποθέσεις:

1. Η εφαρμογή πρέπει να είναι ευέλικτη ώστε να μπορεί να τρέχει σε διαφορετικά υπολογιστικά συστήματα. Αυτός είναι ο κύριος λόγος για τον οποίο επιλέξαμε για γλώσσα προγραμματισμού την Java και την Sql. Ένα από τα βασικά πλεονεκτήματα της Java και της Sql έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και της πλατφόρμας. Οι εφαρμογές που είναι γραμμένες σε Java και Sql τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh χωρίς να χρειαστεί μεταγλώττιση (Compiling) ξανά ή να αλλάξει ο πηγαίος κώδικας.
2. Ένα μεγάλο πρόβλημα της οργάνωσης ενός μοντέλου του πραγματικού κόσμου χωρίς τη χρήση Βάσης Δεδομένων είναι η αδυναμία αποφυγής αρκετής εργασίας από τον άνθρωπο για να αντληθεί η απαιτούμενη πληροφορία, προκειμένου να εξαχθούν απαντήσεις σε απλές ερωτήσεις. Αντίθετα, με τη χρήση Βάσης Δεδομένων τέτοιου είδους πληροφορίες μπορούν να είναι διαθέσιμες ταχύτατα στο χρήστη. Ο τρόπος οργάνωσης των δεδομένων δίνει τη δυνατότητα στις Βάσεις Δεδομένων να μπορούν να δίνουν απαντήσεις αυτόματα και ταχύτατα τόσο σε πολύπλοκες ερωτήσεις, όσο και σε απλές ερωτήσεις οι οποίες όμως θα ήταν πολύ δύσκολο να απαντηθούν εάν είχε ακολουθηθεί οργάνωση βασισμένη σε ένα παραδοσιακό σύστημα (είτε με χειρόγραφα, είτε με μη οργανωμένα σε Βάση Δεδομένων αρχεία στο ηλεκτρονικό υπολογιστή).
3. Πολλές φορές το μοντέλο του πραγματικού κόσμου αλλάζει. Αυτό κατά κανόνα απαιτεί αλλαγές που τις περισσότερες φορές είναι πολύ δύσκολο να υλοποιηθούν σε ένα παραδοσιακό μοντέλο. Αντίθετα, ο τρόπος οργάνωσης των δεδομένων σε μία Βάση Δεδομένων, κατά κανόνα, επιδέχεται αλλαγές και μπορεί πολύ πιο εύκολα να ακολουθεί το μοντέλο του πραγματικού κόσμου. Αυτό σημαίνει ότι οι χρήστες της Βάσης Δεδομένων μπορούν να προσαρμόσουν την μορφή των δεδομένων ευκολότερα και οι αλλαγές αυτές να εφαρμοστούν σε όλα τα δεδομένα, σε αντίθεση με το παραδοσιακό μοντέλο.
4. Η Βάση Δεδομένων χρησιμοποιεί τις δυνατότητες που παρέχει το Σύστημα Διαχείρισης Βάσεων Δεδομένων και ελέγχει για πλεονασμούς και παραβιάσεις της ορθότητας των δεδομένων, καθώς και για ταυτόχρονες προσπελάσεις και μη εξουσιοδοτημένες εγγραφές. Έτσι τα δεδομένα προστατεύονται από ένα μηχανισμό που δεν θα μπορούσε να υπάρξει σε ένα παραδοσιακό σύστημα οργάνωσης. Με αυτό τον τρόπο διασφαλίζεται η ποιότητα των δεδομένων, δηλαδή ο χρήστης έχει μεγαλύτερη ασφάλεια ότι τα δεδομένα που παίρνει ως απάντηση στις ερωτήσεις του είναι σωστά.
5. Ο χρήστης με μια απλή επιλογή από το κύριο μενού θα έχει την ικανότητα να παρουσιάσει οπτικοποίηση των στοιχείων της βάσης δεδομένων με την χρήση ενός Report Generator θα παρουσιάζονται τα στοιχεία του πελάτη π.χ. Όνομα Επίθετο ημερομηνία εγγραφής στο ξενοδοχείο κτλ.

Σε αυτό το σημείο πρέπει να αναφέρουμε ότι πρόκειται για μια πρωτότυπη εφαρμογή η οποία μπορεί να δεχθεί περαιτέρω ανάπτυξη ανάλογα με τις ανάγκες του χρήστη ή της επιχείρησης που θα την χρησιμοποιεί. Όπως αναφέραμε πολλές φορές το μοντέλο του πραγματικού κόσμου μπορεί να αλλάξει επηρεάζοντας το σχεσιακό μοντέλο που σχεδιάσαμε και γενικότερα την βάση δεδομένων. Για αυτό το λόγο θα δούμε στην επόμενη παράγραφο ότι η εφαρμογή μας θα είναι ικανή να δεχτεί τροποποιήσεις ανάλογα με τις ανάγκες του χρήστη. Οι τροποποιήσεις σε μια βάση δεδομένων δεν συνιστώνται μετά το στάδιο υλοποίησης γιατί όπως αναφέραμε μπορεί να καταναλώσουν πολύ χρόνο και πόρους από το σύστημα.

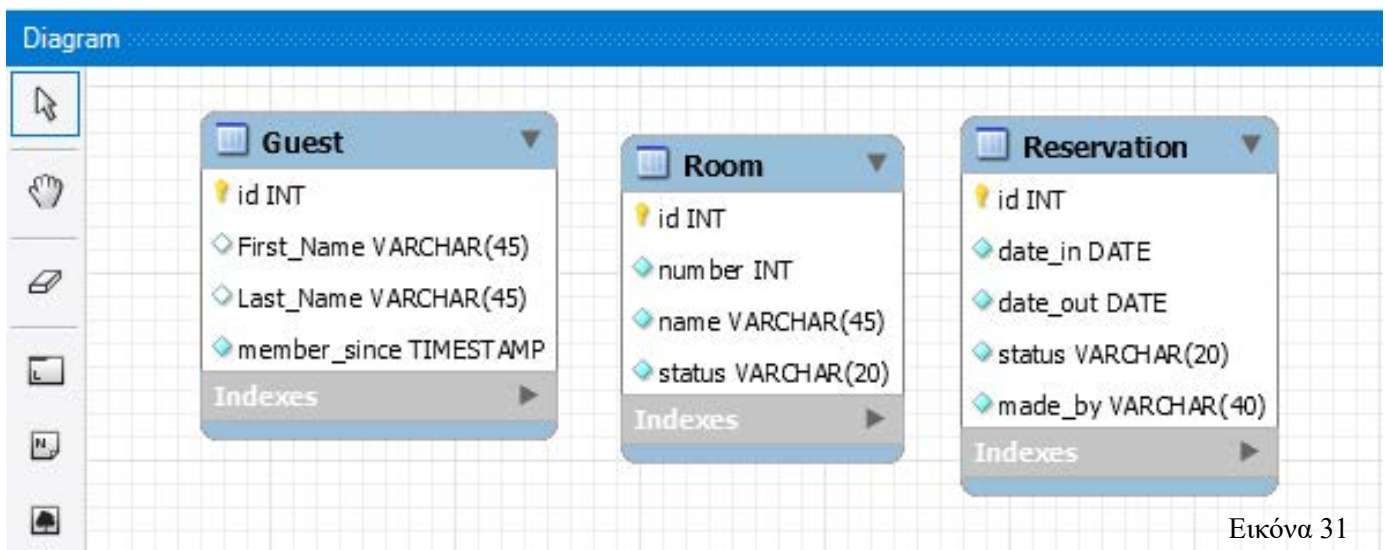
Στο παρακάτω σχήμα φαίνεται ο κύκλος ζωής σχεδίασης και υλοποίησης της Βάσης Δεδομένων εικόνα 30.



Εικόνα 30

#### 4.2.1 Σχεδιασμός Υλοποίησης

Σε αυτή την παράγραφο θα αναλύσουμε βήμα προς βήμα το σχεσιακό μοντέλο δεδομένων στο οποίο βασιστήκαμε για τον σχεδιασμό της βάσης δεδομένων. Το συγκεκριμένο μοντέλο δεδομένων θα ορίζει τον τρόπο που περιγράφονται τα δεδομένα, οι μεταξύ τους σχέσεις η σημασία τους και οι περιορισμοί που επιβάλλονται σε αυτά. Το Σχεσιακό μοντέλο αποθηκεύει την πληροφορία σε πίνακες οι οποίοι ονομάζονται (σχέσεις). Κάθε πίνακας (table) αποτελείται από στήλες (Columns) με διαφορετικά ονόματα. Μια γραμμή (row) του πίνακα αναπαριστά μια σχέση (relationship) μεταξύ ενός συνόλου τιμών. Η μοντελοποίηση βάσεων δεδομένων είναι μια διαδικασία κυκλικής ανακάλυψης. Αναγνωρίζουμε πρώτα τους κύριους πίνακες και τα γνωρίσματα τους εικόνα 31.



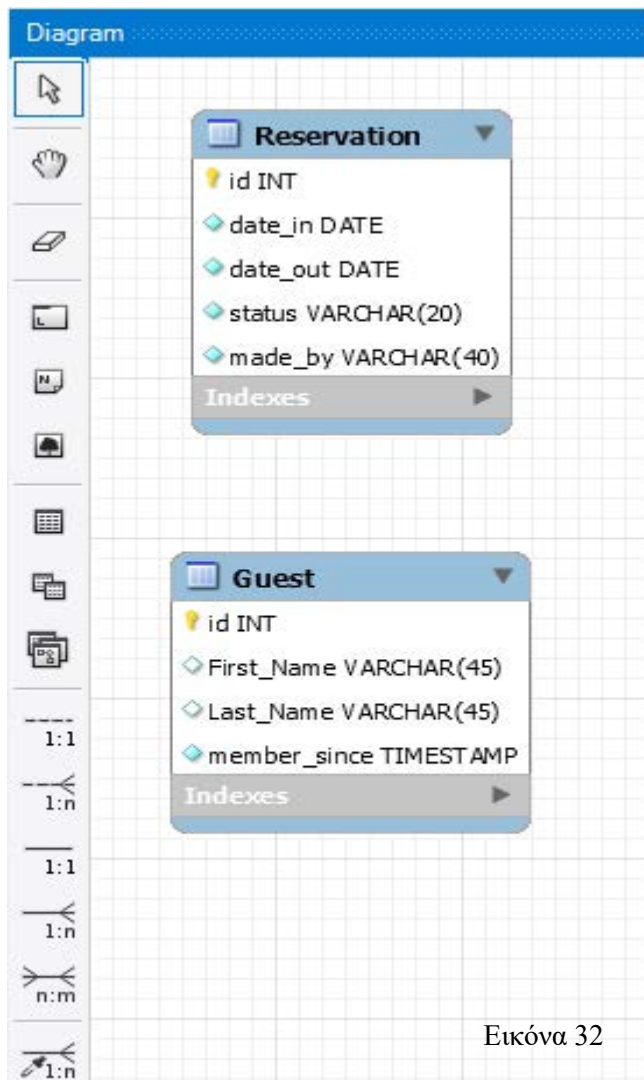
Εικόνα 31



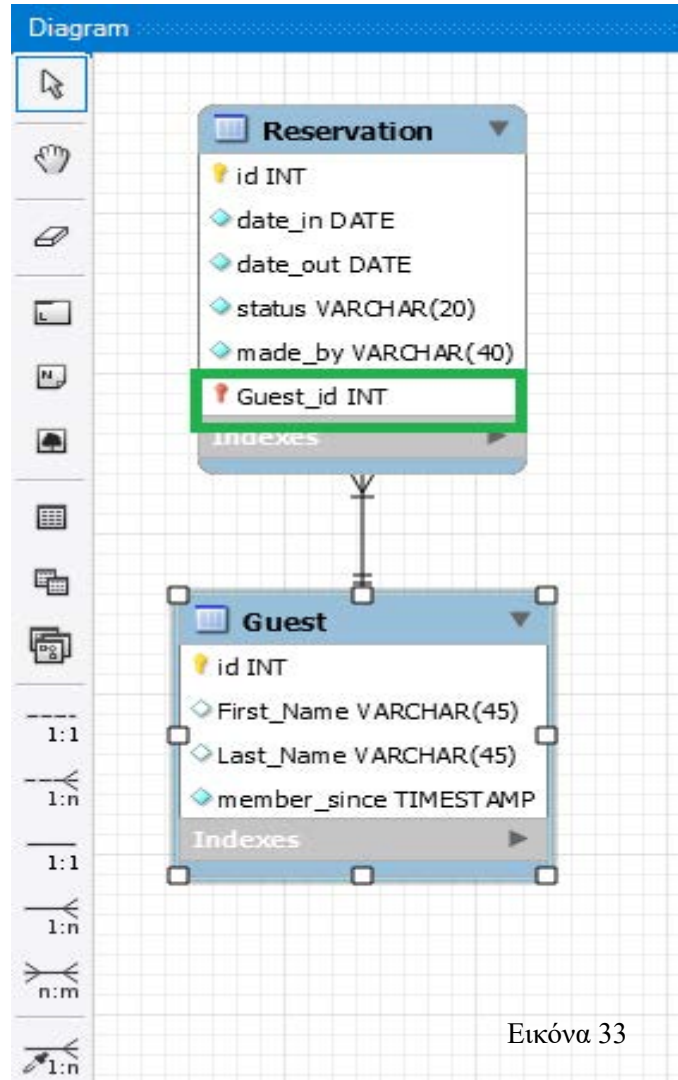
Παρατηρώντας την εικόνα 31 βλέπουμε ότι κάθε στήλη έχει και από ένα τύπο δεδομένων (VARCHAR, INT, TIMESTAMP, DATE) για να υποδείξει τι είδους δεδομένα θα καταχωρούνται στις στήλες. Για παράδειγμα κάθε στήλη status από τούς πίνακες Reservation και Room μπορεί να δεχτεί μόνο μέχρι 20 χαρακτήρες. Κάθε πίνακας πρέπει να έχει μια ή περισσότερες στήλες που να λειτουργεί ως αναγνωριστικό για κάθε εγγραφή του πίνακα. Αυτή η στήλη ονομάζεται πρωτεύον κλειδί (Primary Key) και οι βέλτιστες τεχνικές στον σχεδιασμό της βάσης δεδομένων υποδεικνύουν ότι κάθε πίνακας πρέπει να έχει ένα (PK). Μια σημαντική έννοια, ίσως προφανής για πολύ κόσμο, είναι ότι μια στήλη PK δεν μπορεί να έχει διπλές τιμές. Πρακτικά αυτό σημαίνει ότι κάθε στήλη PK έχει έναν μοναδικό περιορισμό και κάθε προσπάθεια δημιουργίας νέας εγγραφής με διπλή τιμή θα παρουσιάζει σφάλμα.

## 4.2.2 Συσχετίσεις μεταξύ πινάκων

Μια συσχέτιση είναι ένα πολύ σημαντικό στοιχείο σε ένα μοντέλο δεδομένων. Μια συσχέτιση περιγράφει τον τρόπο με τον οποίο ένας πίνακας σχετίζεται με έναν άλλο πίνακα. Στο μοντέλο μας οι συσχέτισεις θα αναπαριστώνται γραφικά ως γραμμές που συνδέουν και τους δύο πίνακες. Όταν δημιουργούμε μια συσχέτιση μεταξύ δύο πινάκων, μια νέα στήλη προστίθεται σε έναν από τους δύο πίνακες. Αυτή η νέα στήλη που μόλις προστέθηκε ονομάζεται ξένο κλειδί (Foreign Key) και λειτουργεί ως δείκτης στον άλλο πίνακα παράδειγμα εικόνα 32, εικόνα 33.



Εικόνα 32

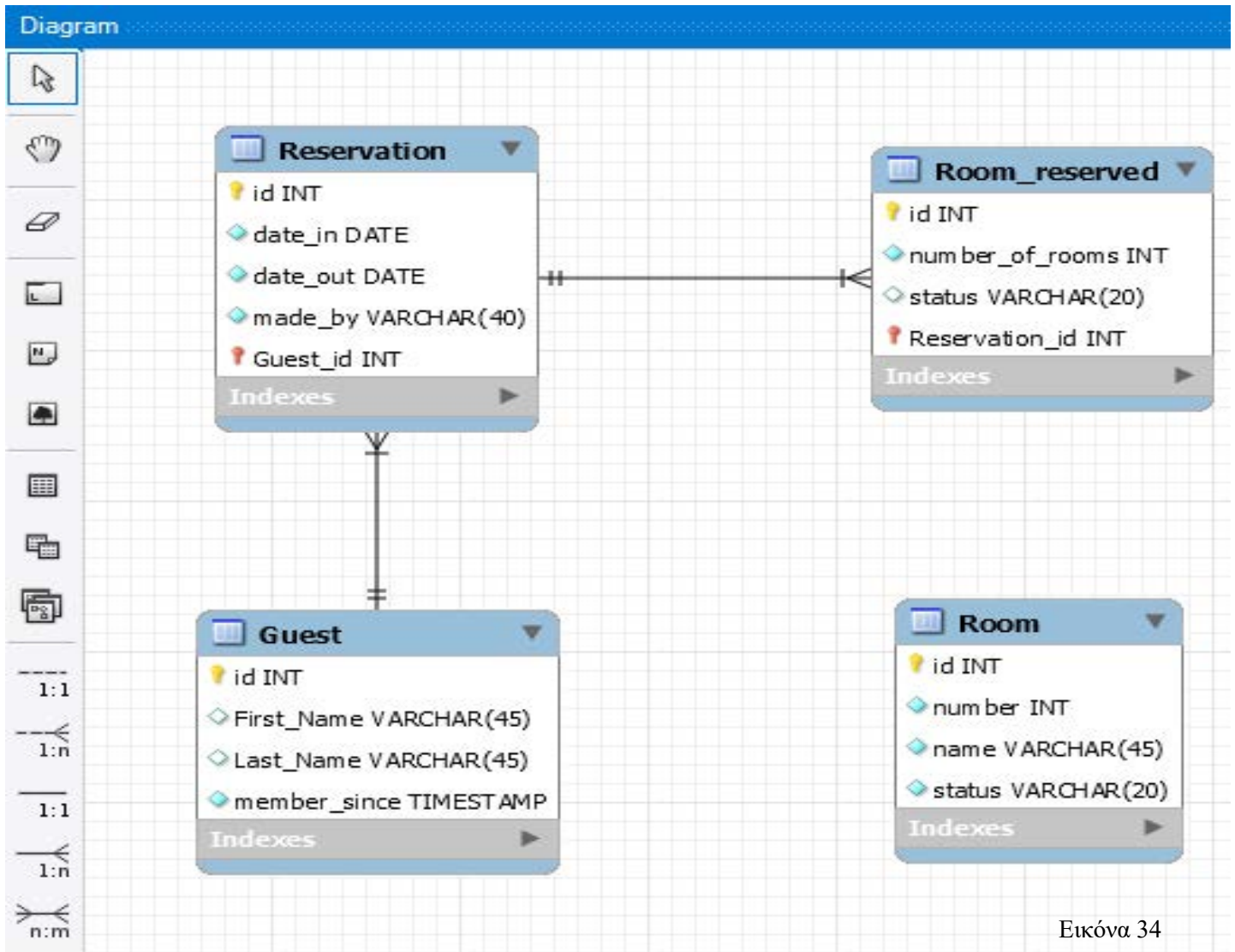


Εικόνα 33

Η σχέση μεταξύ του πίνακα Guest και Reservation είναι ένα προς πολλά 1:n αυτό σημαίνει ότι ένας πελάτης μπορεί να κάνει πολλές κρατήσεις.

### 4.2.3 Ανάπτυξη του μοντέλου της βάσης δεδομένων

Ένας πελάτης μπορεί να θέλει κράτηση για ένα δίκλινο δωμάτιο και ένα ξεχωριστό δωμάτιο με τρία κρεβάτια για τα παιδιά του. Η συγκεκριμένη απαίτηση προσθέτει νέο πίνακα στο μοντέλο μας. Σε αυτό τον νέο πίνακα θα αποθηκεύουμε όλα τα δωμάτια που ανήκουν σε μία κράτηση εικόνα 34.



Εικόνα 34

Έχοντας ως δεδομένο ότι μπορούμε να διαθέσουμε πολλά δωμάτια σε μια κράτηση, πρέπει να επιτρέψουμε την ακύρωση ανά ξεχωριστό δωμάτιο. Παρατηρώντας την εικόνα 34 βλέπουμε ότι το χαρακτηριστικό **status** του πίνακα **Reservation** μεταφέρθηκε στον νέο πίνακα **Room\_reserved**.

Ένα άλλο σημείο που εκκρεμεί στο μοντέλο μας είναι ότι τα δωμάτια μπορούν να χρησιμοποιηθούν από ορισμένους πελάτες για μια χρονική περίοδο. Η συγκεκριμένη επιχειρηματική απαίτηση προσθέτει ακόμα δύο νέους πίνακες: **Hosted\_at** και **Occupied\_room**. Με αυτό τον τρόπο κάθε πελάτης που έμεινε στο ξενοδοχείο θα έχει μια εγγραφή στον πίνακα **Hosted\_at**. Αυτή η εγγραφή θα έχει αναφορά στο δωμάτιο που κρατήθηκε και στον πελάτη. Όπως θα δούμε και στην εικόνα 35 αυτός είναι και ο λόγος που ο πίνακας **Hosted\_at** σχετίζεται με τους πίνακες **Guest** και **Occupied\_room**.



## 4.2.4 Ανάπτυξη διεπαφής για την διαχείριση της βάσης δεδομένων

Στην συγκεκριμένη παράγραφο θα παρουσιάσουμε τον τρόπο με τον οποίο αναπτύξαμε την διεπαφή για την διαχείριση της βάσης δεδομένων. Το κύριο μέρος του κώδικα έχει γραφτεί σε Java με την βοήθεια της ψηφιακής πλατφόρμας NetBeans. Όπως αναφέραμε και στο κεφάλαιο 2.2.2 στην Java επιτρέπονται οι εμφυτεύσεις και άλλων γλωσσών προγραμματισμού όπως η SQL που θα χρησιμοποιήσουμε στην δική μας περίπτωση. Για να είναι ικανή η εφαρμογή μας να εκτελεί εντολές SQL θα κάνουμε χρήση της βιβλιοθήκης JDBC. Οι κυριότερες κλάσεις και διασυνδέσεις του JDBC είναι οι εξής:

**DriverManager:** Φορτώνει τους κατάλληλους οδηγούς JDBC και δημιουργεί σύνδεση με την βάση δεδομένων.

**Connection:** Αντιπροσωπεύει μια σύνδεση με μια συγκεκριμένη βάση δεδομένων. Η επικοινωνία με τη βάση δεδομένων γίνεται μέσω των μεθόδων Connection.

**Statement:** Αντιπροσωπεύει μια εντολή SQL. Οι κλήσεις για την εκτέλεση εντολών SQL πραγματοποιούνται με τις μεθόδους Statement.

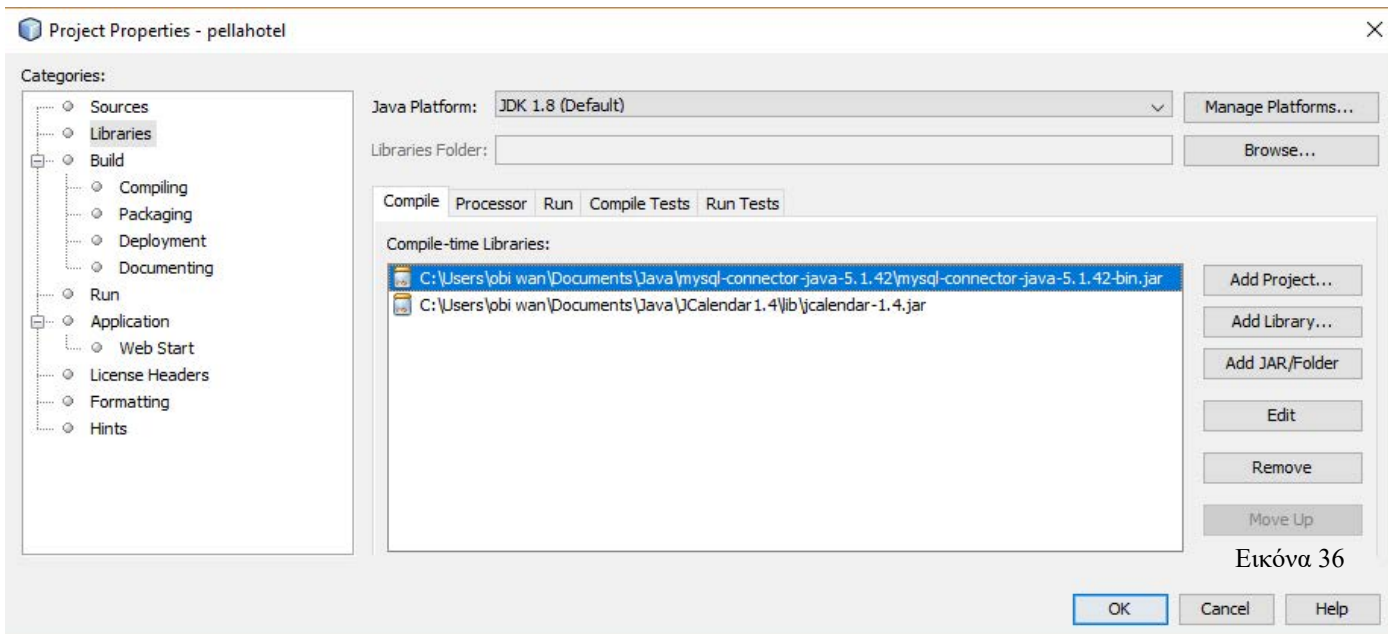
**ResultSet:** Αντιπροσωπεύει τα αποτελέσματα που λαμβάνονται από την εκτέλεση των εντολών SQL. Με τις μεθόδους ResultSet μπορούμε να διαχειριστούμε τα δεδομένα.

**Date, Time, Timestamp:** Βρίσκονται στο JDBC και περιλαμβάνουν τους τύπους για τα αντίστοιχα δεδομένα.

Ένα πρόγραμμα που κάνει χρήση του JDBC, για να ανοίξει και να χρησιμοποιήσει μια βάση δεδομένων ακολουθεί τα παρακάτω βήματα:

- Εισάγει τις απαραίτητες κλάσεις
- Φορτώνει τον κατάλληλο οδηγό JDBC
- Προσδιορίζει την πηγή των δεδομένων
- Δημιουργεί ένα αντικείμενο Connection
- Δημιουργεί ένα αντικείμενο Statement
- Κάνει ερώτηση (query) στη βάση δεδομένων με την βοήθεια του αντικειμένου
- Παίρνει δεδομένα από το αντικείμενο ResultSet που επιστρέφεται
- Κλείνει το αντικείμενο ResultSet
- Κλείνει το αντικείμενο Statement
- Κλείνει το αντικείμενο Connection

Όλες τις κλάσεις και τις μεθόδους που αναφέραμε θα τις αναλύσουμε καλύτερα στη συνέχεια κατά την παρουσίαση του κώδικα. Αρχικά πρέπει να πούμε ότι για να είναι λειτουργική η βιβλιοθήκη JDBC πρέπει να εισαχθεί μαζί με τα υπόλοιπα αρχεία του project στο Netbeans. Για αυτόν τον λόγο επιλέγουμε τον φάκελο Libraries->Properties βρίσκουμε στην συνέχεια τον φάκελο που είναι αποθηκευμένο το αρχείο mysql-connector-java-5.1.42-bin.jar και το προσθέτουμε στην εφαρμογή μας εικόνα 36.



Εικόνα 36



Τα αρχεία JAR είναι η μέθοδος που χρησιμοποιούμε για τη συσκευασία και τη συμπίεση μιας ομάδας αρχείων προκειμένου να διανεμηθούν στους χρήστες. Τα αρχεία αυτά συμπιέζονται με αλγόριθμους παρόμοιους με τους αλγόριθμους συμπίεσης αρχείων ZIP. Αυτό είναι πολύ εξυπηρετικό και για την περίπτωση που κάποιος θέλει να κατεβάσει ένα αρχείο από το internet. Το jar βρίσκεται στον ίδιο φάκελο με το μεταγλωττιστή και άλλα εργαλεία της java.

Στην συνέχεια θα παρουσιάσουμε και θα αναλύσουμε τον κώδικα του κεντρικού μενού για την σύνδεση με την βάση δεδομένων δεδομένων . Αρχικά εισάγουμε τις απαραίτητες κλάσεις και διασυνδέσεις. Αυτές οι κλάσεις βρίσκονται στο πακέτο java.sql. Στην συνέχεια φορτώνουμε τους κατάλληλους οδηγούς με τη μέθοδο forName() γραμμή 219. Στην συνέχεια δημιουργούμε ένα αλφαριθμητικό το οποίο θα περιέχει την κατάλληλη URL ώστε να προσδιορίζει μονοσήμαντα την πηγή των δεδομένων. Το επόμενο βήμα είναι να δημιουργήσουμε ένα αντικείμενο Connection. Το συγκεκριμένο αντικείμενο θα πραγματοποιεί τη σύνδεση με την πηγή των δεδομένων και θα κατασκευάζει ένα περιβάλλον μέσα στο οποίο θα μπορούμε να δημιουργούμε και να εκτελούμε εντολές. Το αντικείμενο Connection δημιουργείται με τη μέθοδο getConnection() της κλάσης DriverManager γραμμή 229 εικόνα 37.

```

214
215
216 private void doConnectToDB() {
217     // check for the driver
218     try {
219         Class.forName("com.mysql.jdbc.Driver");
220     } catch (ClassNotFoundException e) {
221         String msg = "The com.mysql.jdbc.Driver is missing\n"
222             + "install and rerun the application";
223         JOptionPane.showMessageDialog(this, msg, this.getTitle(), JOptionPane.ERROR_MESSAGE);
224         System.exit(1);
225     }
226
227     // connect to db
228     try {
229         con = DriverManager.getConnection("jdbc:mysql://localhost:3306/pellahotel?useUnicode=true&characterEncoding=UTF-8&useSSL=false",
230             "root", "1234");
231         //doRefreshViews();
232     } catch (SQLException e) {
233         String msg = "Error Connecting to Database:\n"
234             + e.getMessage() + "\n\nChange Settings: Tools > Options";
235         JOptionPane.showMessageDialog(this, msg, this.getTitle(), JOptionPane.ERROR_MESSAGE);
236         //tblSessions.setModel(new DefaultTableModel());
237         //tblSessionStudents.setModel(new DefaultTableModel());
238         try {
239             if (isDBConnected(con)) {
240                 con.close();
241             }
242         } catch (SQLException se) {
243             System.out.println("SQLException: " + se.getMessage());
244         }
245     }
246 }
247
  
```

Εικόνα 37

Όπως θα δούμε παρακάτω στην εικόνα 38 στις γραμμές 285-287 δημιουργούμε ένα αντικείμενο με την μέθοδο getConnection(). Με αυτό τον τρόπο εκτός από την δημιουργία κώδικα για πρόσβαση στην βάση δεδομένων θα έχουμε την δυνατότητα να γράψουμε μεθόδους οι οποίες θα περιέχουν ελέγχους για την πρόσβαση στην βάση δεδομένων. Αυτές οι μέθοδοι βρίσκονται στις γραμμές 248-257 και 270-283

```

247
248 private boolean isDBConnected(Connection c) {
249     try {
250         if (c != null && !c.isClosed()) {
251             return true;
252         }
253     } catch (SQLException e) {
254         System.out.println("SQLException: " + e.getMessage());
255     }
256     return false;
257 }
258
259 /**
260  * @param args the command line arguments
261  */
262 public static void main(String args[]) {
263     java.awt.EventQueue.invokeLater(new Runnable() {
264         public void run() {
265             new PellaHotelMenu().setVisible(true);
266         }
267     });
268 }
269
270 public void closeApplication() {
271
272     try {
273         if (con != null) {
274             con.close();
275         }
276         if (pstmt != null) {
277             pstmt.close();
278         }
279     } catch (SQLException sqle) {
280         JOptionPane.showMessageDialog(this, "Closing Connection problem " + sqle.getMessage());
281     }
282 }
283
284
285 public Connection getConnection() {
286     return con;
287 }

```

Εικόνα 38

Η κύρια μέθοδος **main()** η οποία είναι και η μέθοδος που θα εκτελεστεί πρώτη γραμμή 262-268 θα εμφανίσει στην οθόνη ένα JFrame με το όνομα PellaHotelMenu. Ο κώδικας με τις διαστάσεις που θα έχει το παράθυρο του κεντρικού μενού φαίνεται στην εικόνα 39.

```

31
32 public PellaHotelMenu() {
33     Toolkit thekit = getToolkit();
34     this.setTitle("Pella Hotel v1.00");
35
36     initComponents();
37     pack();
38     // center to screen
39     Dimension scrSize = thekit.getScreenSize();
40     Dimension winSize = new Dimension();
41     winSize.width = 512;
42     winSize.height = 256;
43     int newX = (scrSize.width - this.getWidth()) / 2;
44     int newY = (scrSize.height - this.getHeight()) / 2;
45     this.setBounds(newX, newY, this.getWidth(), this.getHeight());
46     //this.setBounds(newX, newY, winSize.width, winSize.height);
47 }

```

Εικόνα 39

Για να μπορεί η εφαρμογή μας να είναι λειτουργική με ένα μηχανισμό αναφοράς, όπως είναι για παράδειγμα η εκτύπωση του ιστορικού των κρατήσεων σε ένα αρχείο pdf χρησιμοποιήσαμε το JasperReports. Το JasperReports είναι μια ανοικτή πηγή από βιβλιοθήκες της Java και όχι ένας αυτόματος μηχανισμός. Αυτό πρακτικά σημαίνει ότι ο κώδικας που περιέχουν οι βιβλιοθήκες δεν μπορεί να εκτελεστεί από μόνος του αλλά πρέπει να ενσωματωθεί με τον υπόλοιπο κώδικα του προγράμματος. Αρχικά κάνουμε εξαγωγή τις βιβλιοθήκες από το αρχείο JasperReport.ZIP στους υποφάκελους του project μας. Στην συνέχεια ενσωματώνουμε τον κώδικα που μας εξυπηρετεί στην εφαρμογή μας. Με αυτόν τον τρόπο όταν ο χρήστης πατήσει το αντικείμενο JMenuItem11 το οποίο στο κεντρικό παράθυρο θα έχει το όνομα "Print All Reservations" θα ενεργοποιεί την μέθοδο **doReport()** η οποία με την σειρά της θα τυπώνει την αναφορά για τις κρατήσεις από τον πίνακα Reservations της βάσης δεδομένων.

```

168 private void jMenuItem11ActionPerformed(java.awt.event.ActionEvent evt) {
169     doReport();
170 }

```

Εικόνα 40

```

181 private void doReport() {
182     doConnectToDB();
183
184     try {
185
186         JasperDesign jasperDesign = JRXmlLoader.load(getClass().getResourceAsStream("/reports/ReservationsNasos.jrxml"));
187         JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);
188         JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, con);
189
190
191         JasperViewer jv = new JasperViewer(jasperPrint);
192
193
194         JDialog viewer = new JDialog(this, "Batch Report", true);
195         viewer.setBounds(jv.getBounds());
196         viewer.getContentPane().add(jv.getContentPane());
197         viewer.setResizable(true);
198         viewer.setIconImage(jv.getIconImage());
199         viewer.setVisible(true);
200
201     } catch (JRException ex) {
202         System.out.println("print: " + ex.getMessage());
203         String connectMsg = "Could not create the report " + ex.getMessage() + " " + ex.getLocalizedMessage();
204         System.out.println(connectMsg);
205     }
206
207     closeApplication();
208 }
209
210
211
212
213
214
215

```

Εικόνα 41

Με τον ίδιο τρόπο που ενεργοποιούμε την μέθοδο για την εκτύπωση της αναφοράς κρατήσεων θα ενεργοποιούμε και τα υπόλοιπα παράθυρα. Πρακτικά αυτό σημαίνει ότι όταν πατήσουμε για παράδειγμα από το κεντρικό μενού το JMenuItem13 με την ονομασία "Guests" θα ενεργοποιούμε ένα JDialog, ένα παράθυρο δηλαδή το οποίο θα διαθέτει όλα τα απαραίτητα JButton, JTextField, JRadioButtons κτλ. για τον έλεγχο την εισαγωγή και διαγραφή ενός πελάτη από την βάση δεδομένων του ξενοδοχείου όπως επίσης και για την εμφάνιση ενός JDialog παραθύρου που θα μας επιτρέπει να έχουμε πρόσβαση στον πίνακα κρατήσεων της βάσης. Στην συνέχεια στις εικόνες 42,43 θα δούμε περιληπτικά τον κώδικα που "ανοίγει" τα παράθυρα Guest, About αλλά και τον κώδικα που κλείνει το παράθυρο του κεντρικού μενού εικόνα 44.



Κώδικας για να εμφανιστεί το jDialog Guest.

```

172 private void jMenuItem13ActionPerformed(java.awt.event.ActionEvent evt) {
173     if (dlgGuest == null) {
174         dlgGuest = new GuestDlg(this, true);
175     }
176     dlgGuest.setVisible(true);
177 }

```

Εικόνα 42

Κώδικας για να εμφανιστεί το jDialog About.

```

164 private void jMenuItem9ActionPerformed(java.awt.event.ActionEvent evt) {
165     new About(this).setVisible(true);
166 }

```

Εικόνα 43

Κώδικας για να κλείσουμε την εφαρμογή.

```

144 private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
145     // File->Exit
146
147     int n = JOptionPane.showConfirmDialog(this,
148         "Do you really want to exit?",
149         "Exit", JOptionPane.YES_NO_OPTION);
150     if (n == JOptionPane.YES_OPTION) {
151         closeApplication();
152         System.exit(0);
153     }
154 }

```

Εικόνα 44

## 4.2.5 Οι διασυνδέσεις ResultSet, Statement και η εκτέλεση εντολών SQL

Σε αυτή τη παράγραφο θα αναλύσουμε τον τρόπο με τον οποίο θα εισάγουμε και θα εξάγουμε δεδομένα από τους πίνακες της βάσης καθώς επίσης και τον τρόπο με τον οποίο εκτελούνται τα ερωτήματα προς την βάση. Μόλις εκτελείται ένα ερώτημα (query) προς την βάση δεδομένων, τα αποτελέσματα επιστρέφονται ως ένα αντικείμενο της διασύνδεσης **ResultSet**. Το αντικείμενο αυτό παρουσιάζει τις τιμές υπό μορφή πίνακα με γραμμές και στήλες. Η προσπέλαση στα στοιχεία του πίνακα γίνεται ανά μια γραμμή κάθε φορά. Το αντικείμενο **ResultSet** περιέχει ένα δείκτη προς την αναφερόμενη γραμμή. Στην αρχή ο δείκτης δείχνει μια θέση πριν την πρώτη γραμμή του πίνακα. Για να μετακινηθεί ο δείκτης προς την επόμενη γραμμή χρησιμοποιείται η μέθοδος **next()** της κλάσης **ResultSet**. Εάν υπάρχει γραμμή τότε η μέθοδος **next()** επιστρέφει τιμή true, ενώ αν ο πίνακας έχει φτάσει στο τέλος του επιστρέφει τιμή false. Η διασύνδεση **ResultSet** παρέχει μια σειρά από μεθόδους **get()** για την επιστροφή των διαφόρων τύπων των δεδομένων κάθε γραμμής. Μερικές από τις μεθόδους που θα χρησιμοποιήσουμε είναι:

- **getInt()**
- **getString()**
- **getTimestamp()**
- **getTime()**
- **getDate()**

Ο κώδικας της εικόνας 45 αναφέρεται στο Button NEXT του Guest JDialog. Όταν ο χρήστης του προγράμματος πατήσει το κουμπί NEXT η συνάρτηση if ενεργοποιεί την μέθοδο **next()** και ο δείκτης μετακινείται μια θέση ώστε να δείχνει την πρώτη θέση του πίνακα Guest. Στην περίπτωση που ο δείκτης φτάσει στο τέλος του πίνακα και δεν υπάρχουν άλλες εγγραφές χρησιμοποιούμε την μέθοδο **previous()** με αυτόν τον τρόπο ο δείκτης επιστρέφει στην τελευταία θέση του πίνακα.



Μετά την χρήση της μεθόδου **getInt()** για να "πάρουμε" το ID του πελάτη στην γραμμή 287 παρατηρούμε την μετατροπή δεδομένων από ακέραιο που είναι αρχικά το ID σε αλφαριθμητικό "String". Στην συγκεκριμένη περίπτωση η μετατροπή γίνεται γιατί στις κλάσεις `TextField` μπορούμε να τοποθετήσουμε κείμενο στο πεδίο ή να παρουσιάσουμε κείμενο από αυτό με τις γνωστές μεθόδους `getText()`, `setText()`. Παρόμοια περίπτωση είναι και αυτή στην γραμμή 291 όπου μετατρέπουμε τα δεδομένα τύπου `Date` από την στήλη `member_since` του πίνακα `Guest`.

```

283 private void jButtonnextActionPerformed(java.awt.event.ActionEvent evt) {
284     try {
285         if (rg.next()) {
286             int ID = rg.getInt("id");
287             String numberAsString = Integer.toString(ID);
288
289             Date date = rg.getDate("member_since");
290             DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
291             String text = df.format(date);
292
293             jTextFieldid.setText(numberAsString);
294             jTextFieldmembersince.setText(text);
295             jTextFieldfirstname.setText(rg.getString("first_name"));
296             jTextFieldlastname.setText(rg.getString("last_name"));
297         } else {
298             rg.previous();
299             JOptionPane.showMessageDialog(GuestDlg.this, "End of File");
300         }
301
302     } catch (SQLException ex) {
303         Logger.getLogger(PellaHotelMenu.class.getName()).log(Level.SEVERE, null, ex);
304     }
305 }

```

Εικόνα 45

Στην εικόνα 46 βλέπουμε ότι αντί για την μέθοδο **next()** χρησιμοποιούμε την μέθοδο **first()** με αυτόν τον τρόπο ο δείκτης μετακινείται κατευθείαν στην πρώτη θέση του πίνακα.

```

353 private void jButtonfirstActionPerformed(java.awt.event.ActionEvent evt) {
354     try {
355         if (rg.first()) {
356             int ID = rg.getInt("id");
357             String numberAsString = Integer.toString(ID);
358
359             Date date = rg.getDate("member_since");
360             DateFormat df = new SimpleDateFormat("dd/MM/yyyy");
361             String text = df.format(date);
362
363             jTextFieldid.setText(numberAsString);
364             jTextFieldmembersince.setText(text);
365             jTextFieldfirstname.setText(rg.getString("first_name"));
366             jTextFieldlastname.setText(rg.getString("last_name"));
367         } else {
368             rg.next();
369             JOptionPane.showMessageDialog(GuestDlg.this, "End of File");
370         }
371     } catch (SQLException ex) {
372         Logger.getLogger(PellaHotelMenu.class.getName()).log(Level.SEVERE, null, ex);
373     }
374 }

```

Εικόνα 46



Με την μέθοδο **Fillcomboroomnumber()** εξάγουμε τους αριθμούς και τους τύπους των δωματίων από τους πίνακες room και room\_type της βάσης δεδομένων. Στην συνέχεια χρησιμοποιούμε την κλάση JComboBoxroom για να εισαγουμε στο πλαίσιο του JComboBox τον αριθμό και τον τύπο του δωματίου. Για να παρουσιάζονται μόνο τα δωμάτια που δεν έχουν επιλεγεί στο ερώτημα (query) που θέτουμε προς την βάση πεοσθέτουμε τον κώδικα SQL "Where room.status is NULL".

```

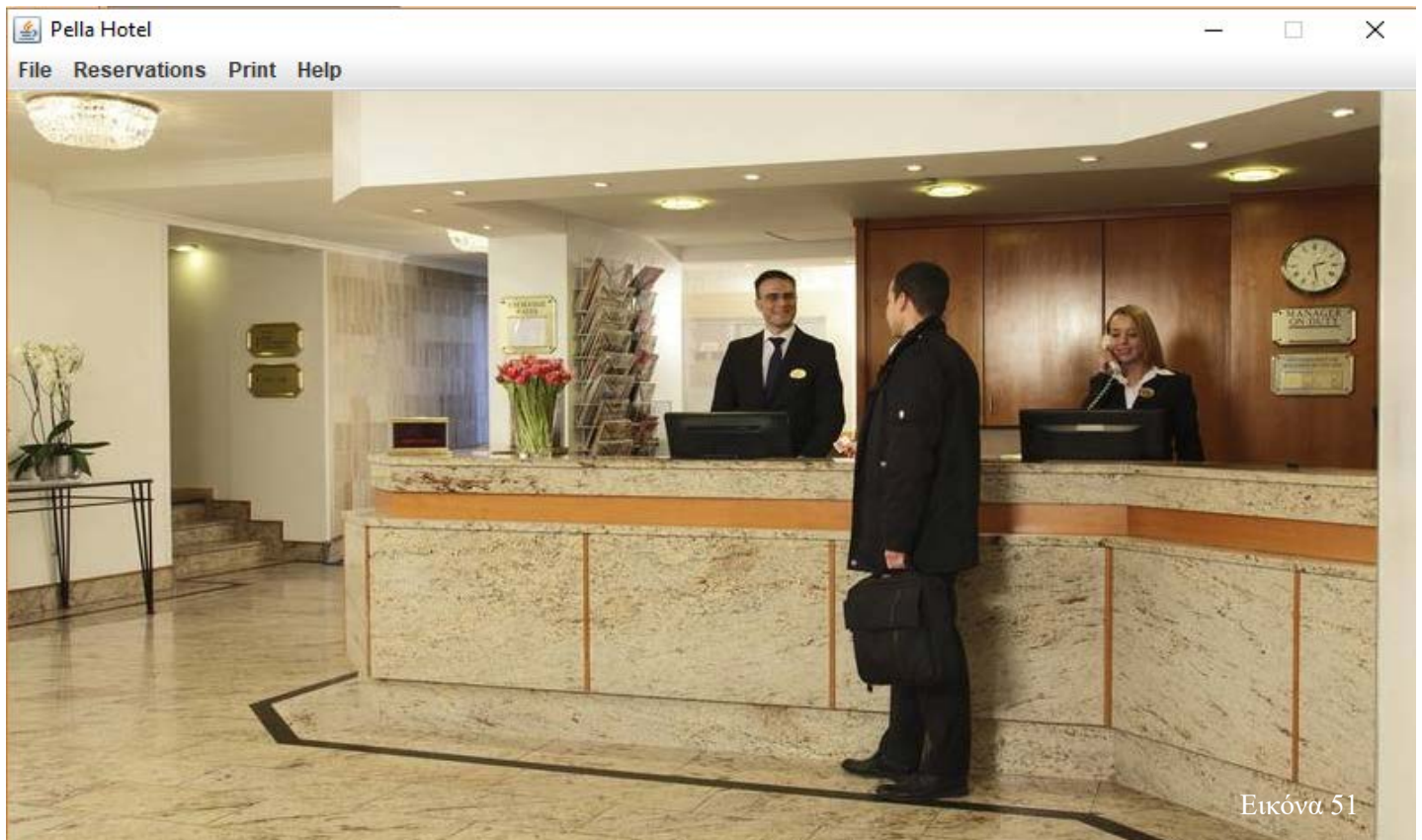
285 private void Fillcomboroomnumber() {
286
287     try {
288
289         while (ro.next()) {
290             int num = ro.getInt("number");
291             String typ = ro.getString("description");
292             String snum = Integer.toString(num);
293             jComboBoxroom.addItem(snum + " " + typ);
294
295         }
296     } catch (SQLException e) {
297     }
298 }

```

Εικόνα 50

### 4.3 Υλοποίηση

Σε αυτή την ενότητα θα παρουσιάσουμε την Εφαρμογή Διαχείρισης Ξενοδοχείου Πόλης με ένα πλήρες εγχειρίδιο χρήσης. Το πρώτο πράγμα που θα δούμε είναι το αρχικό μενού της εφαρμογής. Στην συνέχεια θα δημιουργήσουμε ένα σενάριο κράτησης και θα παρουσιάσουμε ποία είναι η χρήση του προγράμματος στην συγκεκριμένη περίπτωση.



Εικόνα 51



Στην περίπτωση που έρθει ένας πελάτης στο ξενοδοχείο το πρώτο πράγμα που θα κάνει ο χρήστης του προγράμματος είναι να ελέγξει εάν ο πελάτης έχει ξαναμείνει στο ξενοδοχείο. Για να το κάνει αυτό επιλέγει από το κεντρικό μενού Reservations και στην συνέχεια Guest. Στο νέο παράθυρο (Guest Maintenance) που θα παρουσιαστεί ο χρήστης θα γράψει το όνομα και το επίθετο του πελάτη στα πεδία "First Name:" και "Last Name:" και στην συνέχεια θα πατήσει το κουμπί Search εικόνα 52. Ο χρήστης μπορεί ακόμα να αναζητήσει το ονοματεπώνυμο του πελάτη εάν γνωρίζει το ID.

The screenshot shows a window titled "Guest Maintenance" with a close button (X) in the top right corner. The window contains the following elements:

- ID:** An empty text input field.
- First Name:** A text input field containing the name "Michael".
- Last Name:** A text input field containing the name "Jordan".
- Member since:** An empty text input field.
- Search:** A blue button located to the right of the "Member since" field.
- Reservation:** A blue button located below the "Search" button.
- Clear:** A blue button located below the "Reservation" button.
- ADD:** A blue button located at the bottom center of the window.
- Delete:** A blue button located to the right of the "ADD" button.
- Previous:** A blue button located at the bottom left of the window.
- Next:** A blue button located at the bottom right of the window.
- First:** A blue button located at the bottom left, below the "Previous" button.
- Last:** A blue button located at the bottom right, below the "Next" button.

Εικόνα 52

Με αυτόν τον τρόπο γίνεται έλεγχος στον πίνακα Guest της βάσης δεδομένων. Εάν το όνομα και το επίθετο του πελάτη δεν βρεθούν στην ίδια πλειάδα του πίνακα Guest τότε τα πεδία "ID" και "Member since" παραμένουν κενά. Πρακτικά αυτό σημαίνει ότι ο πελάτης επισκέπτεται για πρώτη φορά το ξενοδοχείο. Το επόμενο πράγμα που πρέπει να κάνει ο χρήστης είναι να πατήσει το κουμπί ADD για να καταχωρηθεί στην βάση το όνομα και το επίθετο του πελάτη. Στην αντίθετη περίπτωση που το όνομα και το επίθετο βρεθούν στην βάση, παρουσιάζεται το ID και η ημερομηνία "Member since" η οποία δηλώνει πότε εκχωρήθηκαν τα στοιχεία του πελάτη για πρώτη φορά στην βάση.

Εφόσον τα στοιχεία του πελάτη έχουν καταχωρηθεί και ο πελάτης θέλει να κάνει κράτηση στο ξενοδοχείο το κουμπί Reservations στην εικόνα 53 θα ανοίξει το παράθυρο κρατήσεων εικόνα 54.

The screenshot shows a window titled "Guest Maintenance" with a close button (X) in the top right corner. The window contains the following elements:

- ID:** Input field containing "3".
- First Name:** Input field containing "Michael".
- Last Name:** Input field containing "Jordan".
- Member since:** Input field containing "27/10/2017".
- Buttons:** "Search", "Reservation", "Clear", "ADD", "Delete", "Previous", "Next", "First", and "Last".

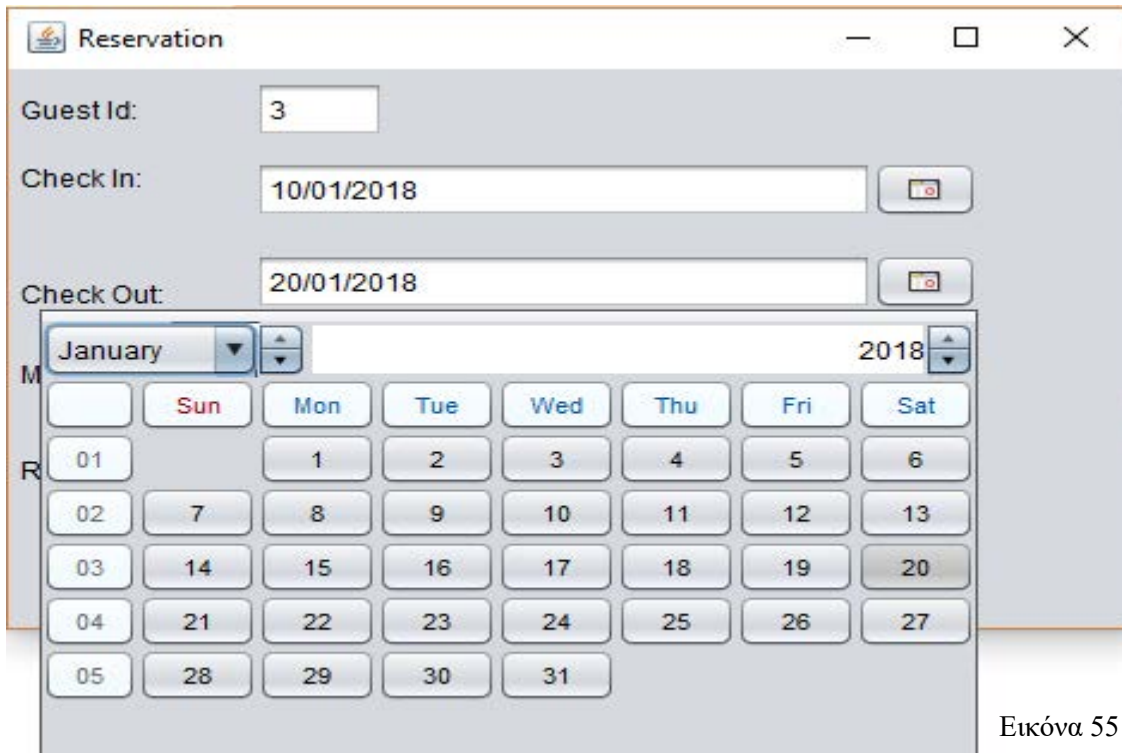
Εικόνα 53

Το πεδίο "Guest Id" στην εικόνα 54 αντιπροσωπεύει τα στοιχεία του πελάτη που θέλει να κάνει κράτηση και παρουσιάζεται αυτόματα στην περίπτωση που ακολουθήσουμε την παραπάνω διαδικασία. Τα πεδία "Check In" και "Check Out" αντιπροσωπεύουν την ημερομηνία της κράτησης. Η επιλογή της ημερομηνίας γίνεται πατώντας τα κουμπιά δεξιά από τα πεδία "Check In" και "Check Out".

The screenshot shows a window titled "Reservation" with standard window controls (minimize, maximize, close) in the top right corner. The window contains the following elements:

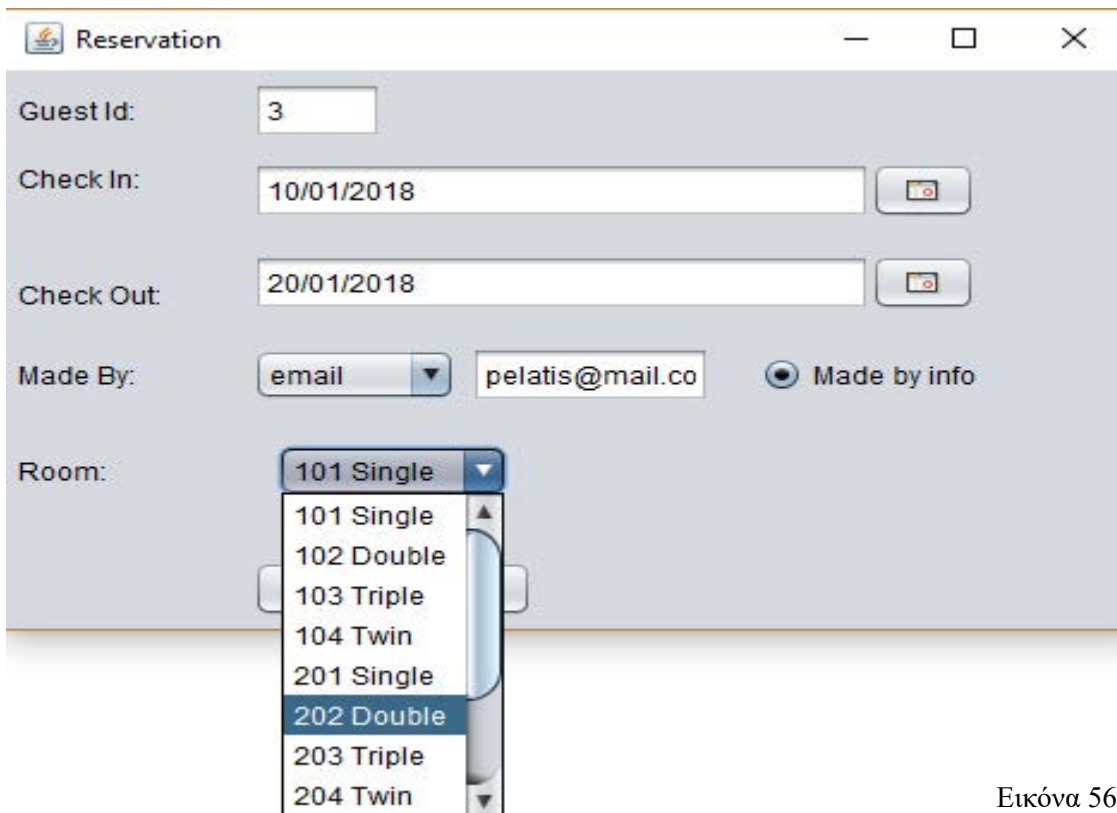
- Guest Id:** Input field containing "3".
- Check In:** Input field with a calendar icon button to its right.
- Check Out:** Input field with a calendar icon button to its right.
- Made By:** A dropdown menu showing "In person" and an empty input field, followed by a radio button labeled "Made by info".
- Room:** A dropdown menu showing "101 Single".
- Buttons:** "ADD" and "Clear".

Εικόνα 54



Εικόνα 55

Το **jComboBox** made by στην εικόνα 56 αντιπροσωπεύει τον τρόπο που έγινε η κράτηση, εάν έγινε απευθείας "In Person" ή μέσω τηλεφώνου "phone" ή "email". Στην περίπτωση που το RadioButton "Made by info" δεν επιλεγεί τότε καταχωρούνται απευθείας το όνομα και το επίθετο του πελάτη στην στήλη **made\_by** του πίνακα **reservation** της βάσης δεδομένων. Η λίστα του **jComboBoxroom** δίνει στον χρήστη την δυνατότητα να επιλέξει τον αριθμό και τον τύπο του δωματίου. Τέλος με το πλίκτρο ADD όλα τα στοιχεία που έχουν επιλεγεί και καταγραφεί στο παράθυρο Reservation καταχωρούνται στην βάση δεδομένων.



Εικόνα 56



