

NIT Jamshedpur



Project Members:

1. Prajwal Kumar - 59
2. Ayush Gupta - 72
3. Ritik Raj - 83
4. Ajay Gupta - 68
5. Indu Preethi Toyaka -87

Mentored by:

Dr. Koushlendra Kumar Singh

Abstract

TOPIC: Hotel Management System

OBJECTIVES: To develop a single platform where all the operations of a hotel can perform e.g. - booking, reservation, housekeeping, service management, customer records, staff records, food for customers, billing, etc. The main motive of this project is to provide the user with a hotel management platform with all the necessary functionalities.

APPROACH: For handling such a huge database, we needed a reliable and relational database. So, we used MySQL to store all information in the form of 6 Tables in the database which includes -staff info table, room info table, customer info table, room booking info table, restaurant menu table and food booking table. These tables can be used to extract data as well as manipulate data. Tables can also be altered, if required. All the bookings information is unique because of a Booking ID which is the primary key of the table. This table is connected to other tables in respect to other columns such as aadhaar no. of customer and room no.

RESULTS: A login portal authorises a user (staff) to use the platform for different purposes. The user gets switched to the main screen where there are 5 navigations. Now, the user can book a room (details get added to the booking info table as well as customer info table), take food order from a room (details get added to the food booking table and total bill), see customer information, other staffs information, availability of rooms and their types (from the room table) and change the menu of the restaurant according to the availability of the items (from rest menu table).

TOOLS USED:

- Netbeans IDE
- GUI ToolKit of JAVA
- Java Development Kit
- Xampp (local host)
- Standard Query Language

Table of Content

S. no.	Title	Page no.
1	Introduction	4
2	Feature Implemented	5
3	ER Diagram	6
4	Implementation and Coding	7-12
5	Testing	13
6	Technologies Used	14
7	Conclusion and Future Enhancements	15-16
8	References	17

Introduction

A hotel is a hive of numerous operations such as front office, booking, and reservation, banquet, finance, HR, inventory, material management, quality management, security, energy management, housekeeping, CRM and more. The hotel has some rooms, and these rooms are of different categories. By room category, each room has the different price. A hotel has some employees to manage the services provided to customers. It is quite important to store the customer record in hotel database which contains customer identity, his address, check in time, check out time, etc., for future and security purposes. Hotel provides food and beverages to their customers and generates the bill for this at the time of their check out.

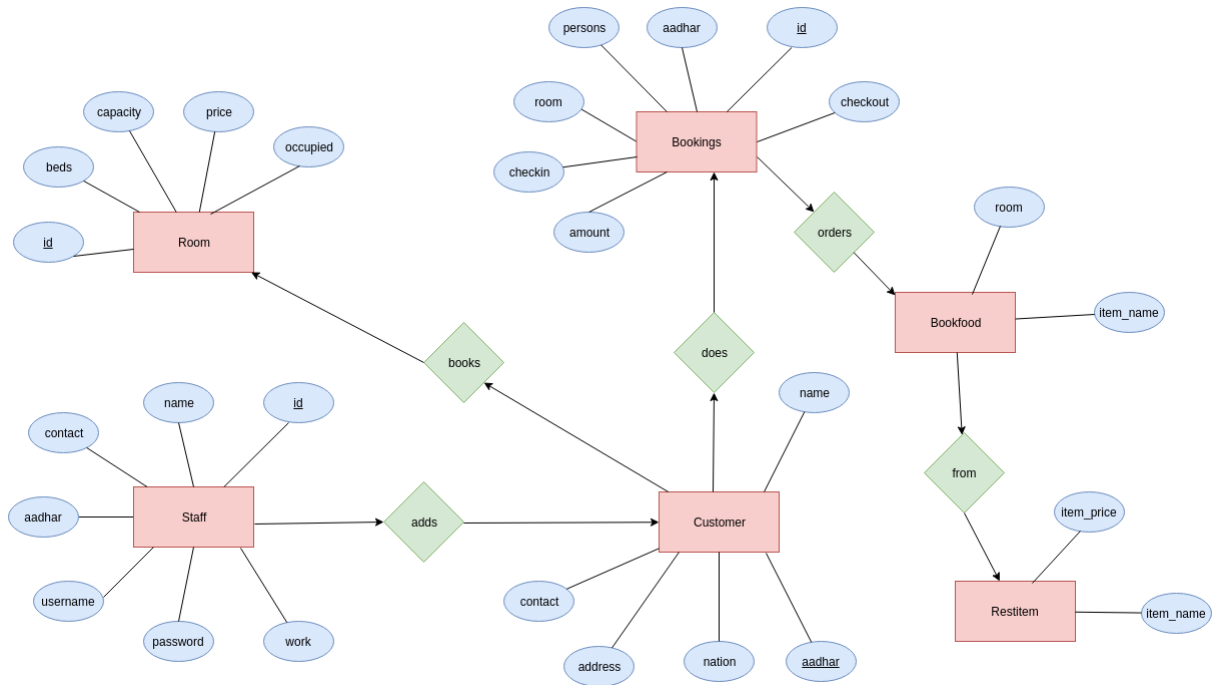
Currently, there are many hotels which lack a common platform for all these functionalities to work together. Databases will be required in huge amount to store all this information and to be retrieved as and when required. Handling of such a huge database will also require a reliable and efficient software (or application).

So, we present here a Java Swing Application built on the JAVA RUNTIME ENVIRONMENT on the platform of NetBeans. It is connected to a MySQL Database for hotel management. This database contains information of staffs, rooms, previous and current bookings, customer details, restaurant menu, staff authentication, etc. It is capable of hosting almost all the features that can be required by a hotel.

Features Implemented

- Login feature with password. Username and password are assigned to each staff in the hotel.
- Once logged in, user can access staff information, customer list and room information.
- Booking of Rooms. Check in time will be set according to computer date. Same applies for checkout time.
- Feature to add new staffs.
- Ordering of Food. Food can be ordered and amount is added in final bill.
- Option to Add and Remove items in Restaurant Menu.

ER Diagram



Implementation and Coding

This project is a simple JAVA Swing Application for Hotel Management.

About the Framework:

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs. Swing API is set of extensible GUI Components to ease developer's life to create JAVA based Front End/ GUI Applications. It is built upon top of AWT API and acts as replacement of AWT API as it has almost every control corresponding to AWT controls.

Database Used:

This project uses MySQL as its database. MySQL is an open source relational database management system (RDBMS). MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

Connecting with the Database:

```
try{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con;
    con=DriverManager.getConnection("JDBC:mysql://localhost:3306/mysql","root",Credentials.
    sqlPassword);
    Statement stmt;
    stmt = con.createStatement();
    stmt.executeUpdate("USE hotelsystem");
    stmt.executeUpdate("insert into staff(name,contact,aadhar,username,password,work)
    values('"+name+"','"+contact+"','"+aadhar+"','"+usr+"','"+pass+"','"+work+"');");
    JOptionPane.showMessageDialog(frame, "Staff Added");
    new MainScreen().setVisible(true);
    this.setVisible(false);
}
catch( Exception e){
    System.out.println("Exception: "+e);
}
```

Here, JDBC is used to connect the program with the database. The JDBC standard defines an application program interface (API) that Java programs can use to connect to database servers. The word JDBC stands for Java Database Connectivity.

Each database product that supports JDBC provides a JDBC driver that must be dynamically loaded in order to access the database from Java. This is done by invoking `Class.forName` with one argument specifying a concrete class implementing the `java.sql.Driver` interface. This interface provides for the translation of product-independent JDBC calls into the product-specific calls needed by the specific database management system being used. The driver is available in a `.jar` file at vendor Web sites and should be placed within the classpath so that the Java compiler can access it.

The Java program must import `java.sql.*`, which contains the interface definitions for the functionality provided by JDBC. The first step in accessing a database from a Java program is to open a connection to the database. This step is required to select which database to use, here `mysql`. Only after opening a connection can a Java program execute SQL statements. A connection is opened using the `getConnection` method of the `DriverManager` class (within `java.sql`). This method takes three parameters.

- The first parameter to the `getConnection` call is a string that specifies the URL, or machine name, where the server runs (here, `mysql://localhost:3306/mysql`), along with possibly some other information such as the protocol to be used to communicate with the database, the port number the database system uses for communication, and the specific database on the server to be used.
- The second parameter to `getConnection` is a database user identifier, which is a string.
- The third parameter is a password, which is also a string and is stored in `Credentials.java` file.

Once a database connection is open, the program can use it to send SQL statements to the database system for execution. This is done via an instance of the class `Statement`. To execute a statement, we invoke either the `executeQuery` method or the `executeUpdate` method, depending on whether the SQL statement is a query (and, thus, returns a result set) or non-query statement such as `update`, `insert`, `delete`, `create table`, etc.

Frames in the project:

Login

This frame implements the login feature with password. On Entering an invalid id or password, a popup appears saying so.

MainScreen

The Main screen is the next frame. It has six different options-

- Checking the available rooms
- Checking the Customer List
- Ordering food via Restaurant
- Checking Staff Details
- Adding Staff
- LogOut

RoomList:

This shows the list of vacant room in a table. It has an option to book rooms. On clicking on any room in the table, we are redirected to Booking room frame.

```
private void tableMouseClicked(java.awt.event.MouseEvent evt) {
    int row = table.rowAtPoint(evt.getPoint());
    String room = (String) table.getValueAt(row, 0);
    RoomBooking rb = new RoomBooking();
    rb.setTextField(room);
    rb.setVisible(true);
    this.setVisible(false);
}
```

In RoomBooking class the function setTextField:

```
void setTextField(String room){
    roomf.setText(room);
}
```

RoomBooking:

This frame is used for booking rooms. On entering aadhaar number the system checks if the customer data is present in database or not. If it is present it automatically fills the data from database. On booking the check-in time is automatically entered from system time.

```
Date date = new Date();  
long time = date.getTime();
```

If the details like contact or aadhaar are incorrect, the system sends a popup saying so. On booking the occupied column is set 1 in room table.

CustomerList:

This frame shows the customers residing in the hotel. On clicking on the customer, a popup is generated with the bill and an option to checkout. Checkout time is generated from the system. After checkout the room would be now available in the roomlist. On Checking out the occupied column is set 0 in room table.

Calculation of Price:

```
long div = 1000*60*60*24;  
long days = (time-chkin)/div+1; //time is current time i.e. checkout time  
bill+=price*days; //bill contains existing bill i.e. restaurant bill and price is the rate of the  
room
```

RestMenu:

This displays the Restaurant menu. There is also an option to add and remove items from the menu as well as to order food.

BookFood:

This books food and adds the amount to the bill in database. It takes the room number and the item name to order food.

StaffInfo:

It displays the information of the staffs working in the Restaurant.

AddStaff:

This frame is used for adding new staffs. New staffs are assigned with username and password.

Tables in the project:

Customer:

Field	Type	Null	Key
aadhar	varchar	NO	PRIMARY KEY
name	varchar	NO	
contact	varchar	NO	
address	varchar	YES	
nation	varchar	YES	

Room:

Field	Type	Null	Key
id	int	NO	PRIMARY KEY
beds	int	NO	
capacity	int	NO	
price	int	NO	
occupied	int	NO	

Bookings:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRIMARY KEY	NULL	AUTO_INCREMENT
aadhar	varchar	NO		NULL	
number_of_persons	int	NO		NULL	
room	int	NO		NULL	
checkin	varchar	NO		NULL	
checkout	varchar	YES		NULL	
amount	int	YES		0	

Staff:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRIMARY KEY	NULL	AUTO_INCREMENT
name	varchar	NO		NULL	
contact	varchar	NO		NULL	
aadhar	varchar	NO		NULL	
username	varchar	NO	UNIQUE	NULL	
password	varchar	NO		NULL	
work	varchar	NO		NULL	

Restitem:

Field	Type	Null	Key
Item_name	varchar	NO	
Item_price	varchar	NO	

Bookfood:

Field	Type	Null	Key
Room	int	NO	
Item_name	varchar	NO	

Testing

Checks Implemented:

Room Booking:

- Aadhaar number should be 12 digits.
- Contact number should be 10 digits.
- Aadhar and contact should only contain numbers.
- On Entering Aadhaar, the system automatically checks if data of the customer is present in database or not.
- None of the field should be left blank.
- Number of people shouldn't exceed the capacity of the room.
- Booked Room is not already occupied.

Add Staff:

- Aadhaar number should be 12 digits.
- Contact number should be 10 digits.
- Aadhar and contact should only contain numbers.
- None of the field should be left blank.

Book Food:

- Item name should exist in menu.
- None of the field should be left blank.
- Room shouldn't be vacant.

Add Menu Item:

- None of the field should be left blank.
- Price should only contain numbers.

Delete Menu Item:

- None of the field should be left blank.
- Item should exist in Menu.

Technologies Used

JAVA

Java is a set of computer software and specifications developed by James Gosling at Sun Microsystems, which was later acquired by the Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers.

NetBeans Editor

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, macOS, Linux and Solaris. In addition to Java development, it has extensions for other languages like PHP, C, C++, HTML5 and JavaScript.

MySQL

MySQL is an open source relational database management system (RDBMS). MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing.

Draw.io

Draw.io is an open source technology stack for building diagramming applications, and the world's most widely used browser-based end-user diagramming application. Draw.io is used to draw Entity Relationship diagram.

GitHub

GitHub is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Conclusion and Future Enhancement

Our objective was to provide the user with a hotel management platform with all the necessary functionalities and to develop a single platform where all the operations of a hotel can perform e.g. - booking, reservation, housekeeping, service management, customer records, staff records, food for customers, billing, etc. For this, we used MySQL to store all information in the form of 6 Tables in the database which includes -staff info table, room info table, customer info table, room booking info table, restaurant menu table and food booking table. These tables can be used to extract data as well as manipulate data. Tables can also be altered, if required.

Now, the user can book a room (details get added to the booking info table as well as customer info table), take food order from a room (details get added to the food booking table and total bill), see customer information, other staffs information, availability of rooms and their types (from the room table) and change the menu of the restaurant according to the availability of the items(from rest menu table).

This application can now be used by any hotel and so functionalities can also be customized according to the corresponding needs. It is capable of hosting almost all the features that can be required by a hotel. But, still there are operations which may be required by some users, not by others. So, there will always be a chance of improvement and enhancement for it. e.g.- online payment, online booking, reservation cancellation, allotting housekeeping and other staffs their work through this, etc. These functions can be added to the system, if required.

References

- Database System Concepts, SIXTH EDITION by *Abraham Silberschatz, Henry F. Korth, S. Sudarshan*
- Java Platform Standard Edition 8 Documentation - Oracle Docs
<https://docs.oracle.com/javase/8/docs/>
- MySQL Documentation <https://dev.mysql.com/doc/>
- Wikipedia <https://wikipedia.com>