



PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering
Session: Jan - May 2022

**UE19CS353 – Object Oriented Analysis and Design with Java
Theory ISA (Mini Project)**

Report on
Hotel Management System
Using Java Swing, MySQL

By:

Minal H R – PES2UG19CS234

Nidhi Gupta – PES2UG19CS256

Nikila S – PES2UG19CS260

6th Semester D

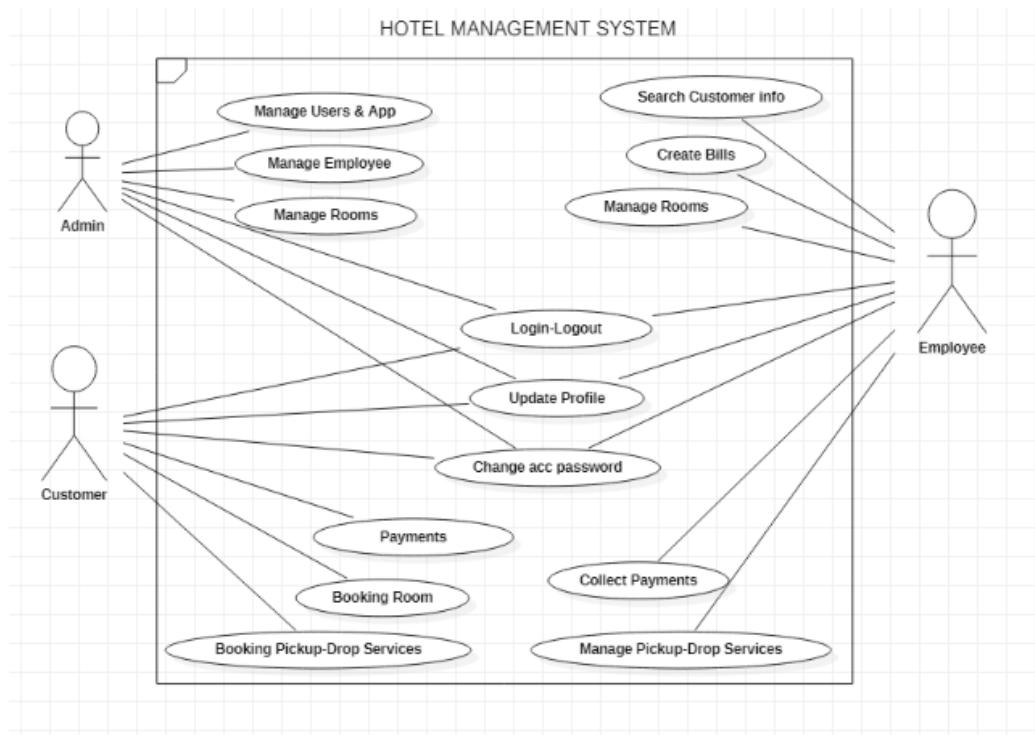
1. Project Description

Link to Github Repository

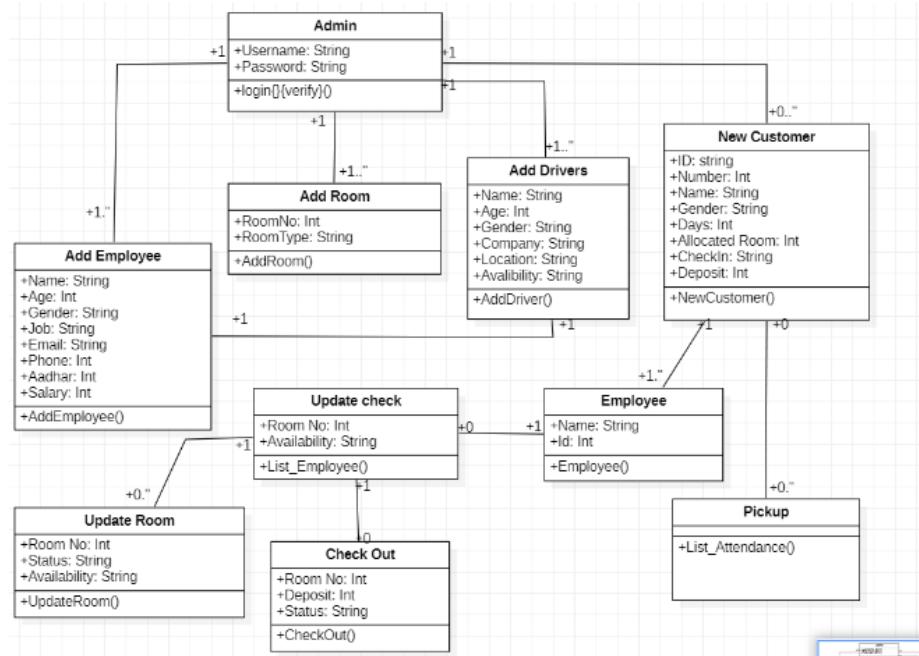
Hotel Management System is a project that manages a typical hotel environment which includes room availability operations, customer friendly service, and maintains employee status. To make it more user friendly, we provide a simple UI which has features like allocation of rooms as per the customer need and interest, records such as checkout time and details, releasing or empty of room and to record the process in a computer system for future.

2. Analysis and Design Models

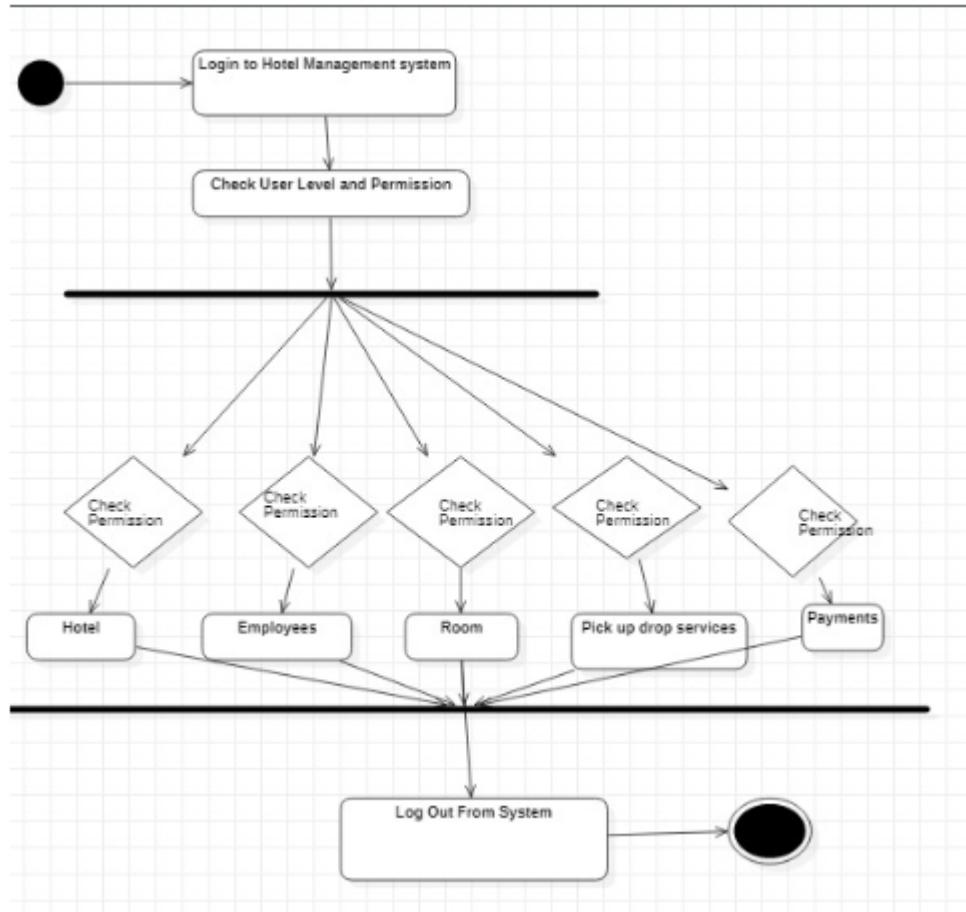
> Use Case Diagram



> Class Diagram

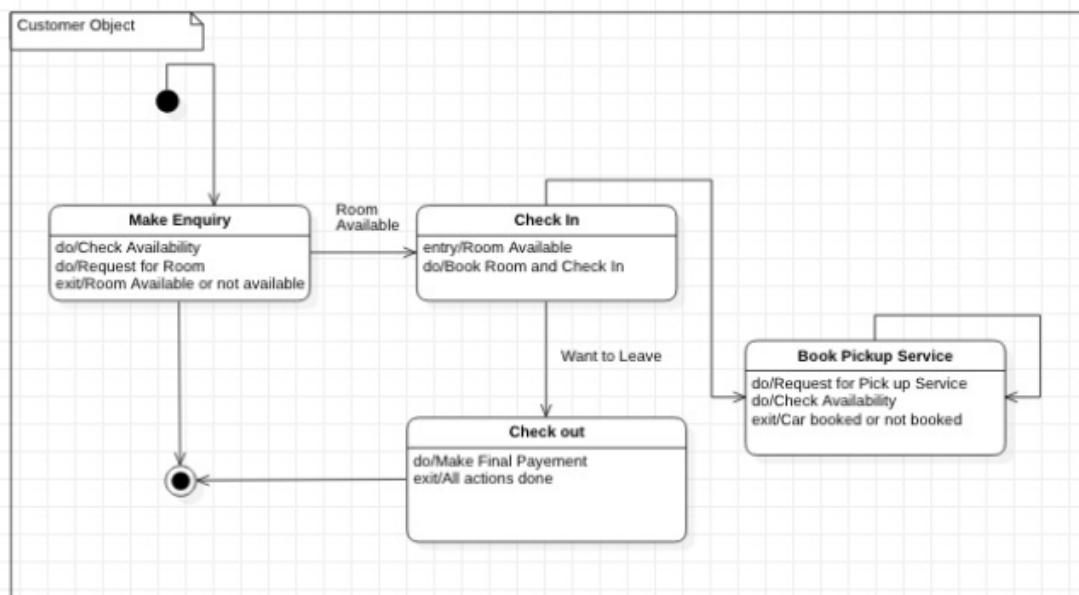


> Activity Diagram

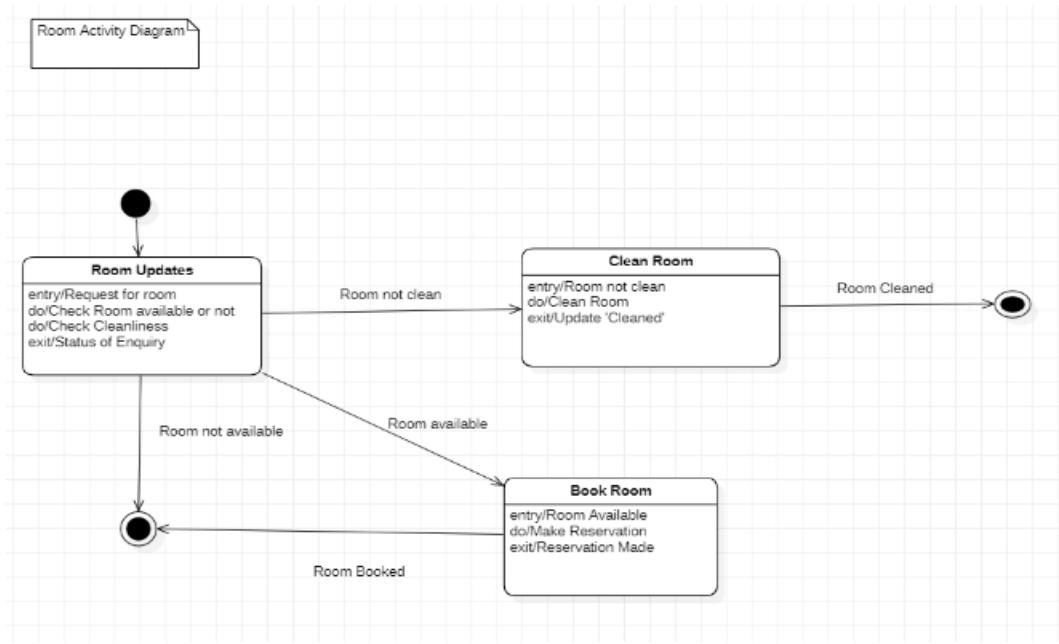


> State Diagrams

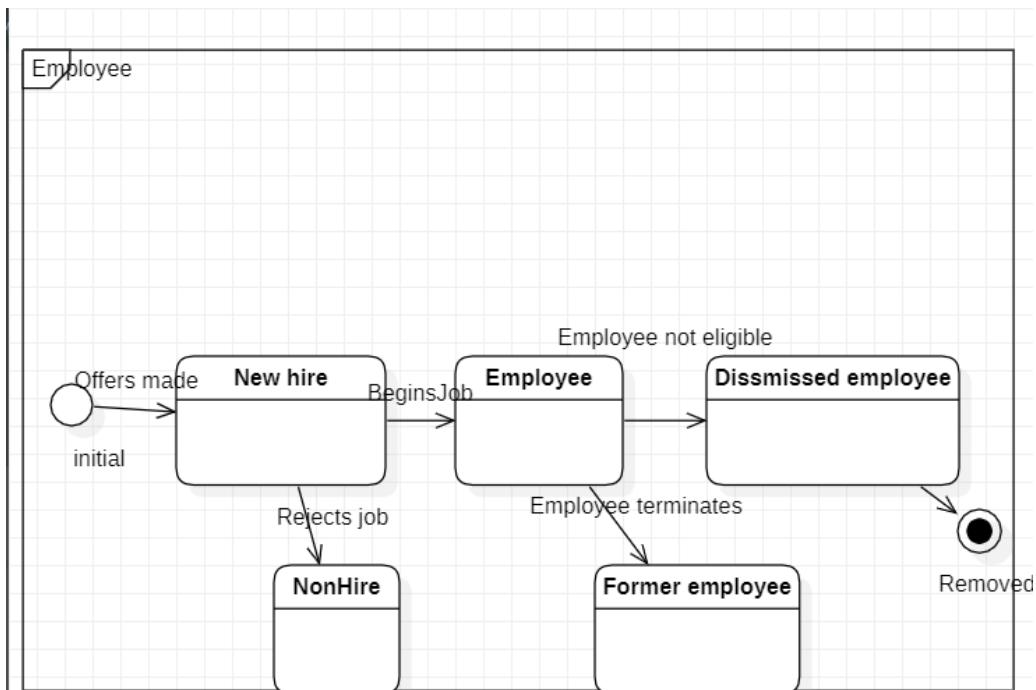
Customer



Room



Employee



3. Tools and Frameworks Used

- MySQL for DataBase

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

- Java Swing Framework

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes – an API for providing a graphical user interface for Java programs. Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit.

- Java AWT

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java.

- Netbeans IDE

NetBeans is an integrated development environment for Java. NetBeans allows applications to be developed from a set of modular software components called modules.

- Github

GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git, plus its own features

4. Design Principles and Design Patterns Applied

Facade Pattern

Facade pattern hides the complexities of the system and provides an interface to the client using which the client can access the system. This type of design pattern comes under structural pattern as this pattern adds an interface to existing systems to hide its complexities.

This pattern involves a single class which provides simplified methods required by the client and delegates calls to methods of existing system classes.

Here, Dashboard.java uses a Facade design pattern where the internal implementation details are hidden from user view. Employees, Rooms and Drivers can be added by the click of a few buttons from user pov.

```
HotelDetailshello1.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        try{
            new AddEmployee().setVisible(true);
        }catch(Exception e){}
    }
});
```

```
HotelDetailshello2.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        try{
            new AddRoom().setVisible(true);
        }catch(Exception e){}
    }
});
```

```
HotelDetailshello3.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent ae){
        try{
            new AddDrivers().setVisible(true);
        }catch(Exception e){}
    }
});
```

Single Responsibility Principle

The single responsibility principle states that every Java class must perform a single functionality. Implementation of multiple functionalities in a single class mashup the code and if any modification is required may affect the whole class. It is precise and the code can be easily maintained.

Here,room.java has only one responsibility, that is, to create rooms. It calls only that function.

```

        lblNewLabel_1 = new JLabel("Bed Type");
        lblNewLabel_1.setBounds(417, 15, 76, 14);
        contentPane.add(lblNewLabel_1);

        lblId = new JLabel("Room Number");
        lblId.setBounds(12, 15, 90, 14);
        contentPane.add(lblId);

    }

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    Room frame = new Room();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

Open/Closed Principle

The application or module entities the methods, functions, variables, etc. The open-closed principle states that according to new requirements the module should be open for extension but closed for modification. The extension allows us to implement new functionality to the module.

We can add from Jframes without affecting the existing application.

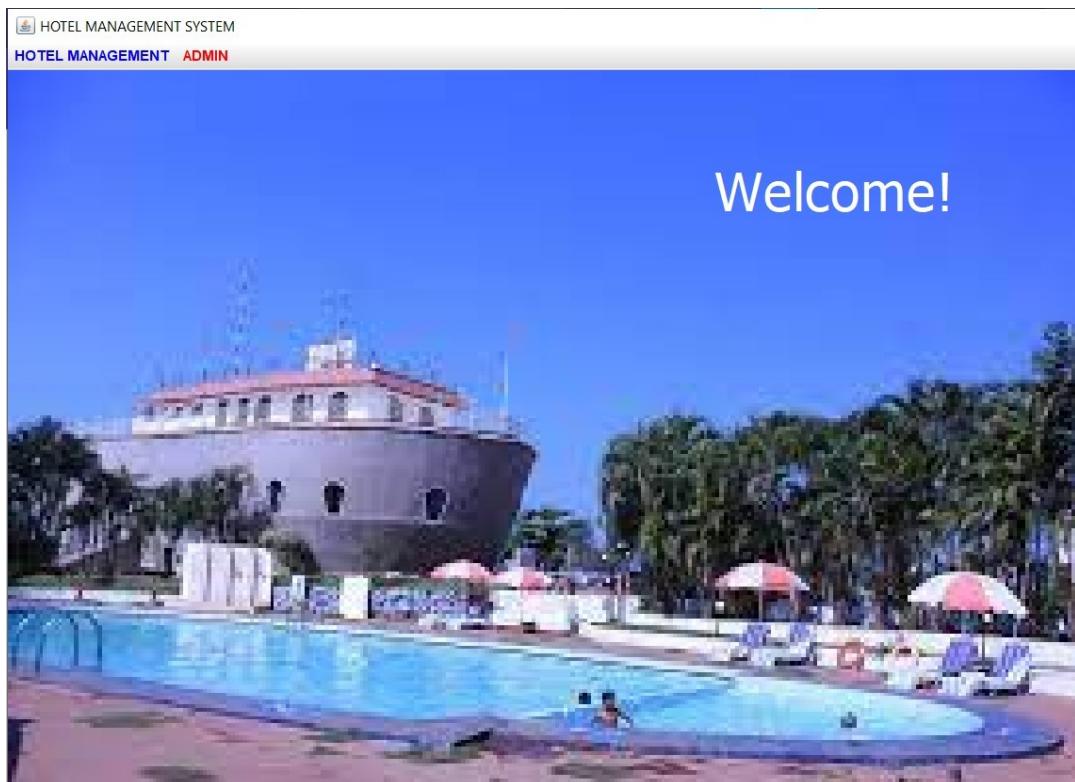
Jframe is open for extension but closed for modification.

```
public class Dashboard extends JFrame{  
  
    public static void main(String[] args) {  
        new Dashboard().setVisible(true);  
    }  
  
    public Dashboard() {  
        super("HOTEL MANAGEMENT SYSTEM");  
  
        setForeground(Color.CYAN);  
        setLayout(null);  
    }  
}
```

5. Application Screenshots (3-4 important pages)

Main Application Page



A login interface for a hotel management system. At the top left is a "Login" button with a key icon. On the right are standard window control buttons for minimize, maximize, and close. Below the controls, there are two input fields: one for "Username" containing "admin" and another for "Password" containing ".....". To the right of the password field is a black silhouette of a person wearing a suit and tie. At the bottom are two buttons: "Login" on the left and "Cancel" on the right.

Employee

 ADD EMPLOYEE DETAILS

NAME	Nidhi
AGE	23
GENDER	<input type="radio"/> MALE <input checked="" type="radio"/> FEMA...
JOB	Accountant
SALARY	67000
PHONE	9876543212
AADHAR	4567890
EMAIL	qwertuio@gmail.com
SAVE	



 ADD EMPLOYEE DETAILS

Name	Age	Gender	Job	Salary	Phone	Aadhar	Gmail
ftbthbf	32	male	Front Desk Clerks	34567	1345678653	3456789-0987654	vgfbfh@gmail.com
minl	20	female	Manager	78900	345678	1gjhfj	ngjktf
karishma	21	female	Housekeeping	13000	98765432	345676543	qwertuui@hotmail.cc
nikila	21	female	Manager	60000	9876543210	1234567890	nikila@gmail.com
Nidhi	23	female	Accountant	67000	9876543212	4567890	qwertuio@gmail.co
nikila . s	21	female	Manager	56000	9876543221	123456789	nikila@gmail.com

Room



Room Num...

Availability

Cleaning St...

Price

Bed Type

Add

Back



Customer



NEW CUSTOMER FORM

ID :	Aadhar Card
Number :	7778998922315647
Name :	Pranathi
Gender :	<input type="radio"/> Male <input checked="" type="radio"/> Female
Country :	india
Room Number :	916
Checked-In :	9-4-22
Deposit :	5000



Add

Back



ID	Number	Name	Gender	Country	Room	Check-in Status	Deposit
Passport			null				
Passport	2	min	Female	in	2	23-2-22	5000
Passport	1	minal	Female	In	1	1-1-22	3000
Passport	2	nidhi	Female	India	4	23-3-22	6000
Aadhar Card	7778998922315647	Pranathi	Female	india	916	9-4-22	5000

6. Team member contributions

NAME	CONTRIBUTIONS
Minal H R	Room
Nidhi Gupta	Customer
Nikila S	Employee