Objectives: -

Hotel Management System Web Application is widely used by ADMIN to store, retrieve and modified their hotel room availability information. This project is the implementing of a backend web application where we can create, Read, update, and delete Hotel information. This application will be built using Spring boot and MySQL (for managing RDBMS).

Project Architecture: -

The project Architecture can be divided into 4 stages as follows:

1.Design the front-end of the web app using **JSP, CSS, html, java Script, jQuery**.

2.implement controller to manage requests and response for all the CURD operations using **java Spring Boot**.

3.implement hibernate JPA Queries to fetch& post Data.

4.intregrate the web application with the **MYSQL** database to fetch data from the backend.

Functional Specification:

**Requirement:**

Hotel Management different actors:

**Admin User:**

Room Listing Page:

* Display a list of all available rooms using a jQuery Database.
* Include columns for Room Type, Price, and Action (Edit/Delete)

Add/Update Room:

* Provide a pop page to give add/update room details.
* Provide each field as below,

- Room Type (Dynamic Drop-Down bring data from DB table)

- Price

- Room Amenities

- AC/Non-AC (Radio Button)

- Smoking/Non-Smoking Room (Radio Button)

- WI-FI/TV/Geyser (Check Box)

Delete Room:

* + Verify if the room has any outgoing or future bookings. If so, display an error message. Otherwise, delete the room.

**Customer:** Enquiry about room availability.

NOTE: -

1.Navigation menu: -Manage Room, manage booking

2. Manage Room: -

1.remove from & todate filter

2.remove from & todate from Data table

3.remove from & todate from Add/Update View

Add/Update View:-

1.need to add additional field "totalRoomsAvaliable"

3.Update DB Schema accordingly

manage booking: -----

1.use current view for listing but also add two more columns,"totalroomavaliable" & "totalRoomBooked"

2.update action menu

if totalroomavaliable = = totalRoomBooked

then disable book room option.

else

then enable book room option.

also, onclick of bookroom show alert message like “it will be implementing in phase 2"

Technical Specification:

Spring Boot: this Framework used for creating backend application, handling dependency, remove boiler plate code and providing essential features.

JSP: generating dynamic HTML content and integrating java code with web page.

JavaScript: for Client-Side interactivity and enhancing the user experience.

jQuery: help to easily interact with HTML elements.

MySQL: Database Management system to Store and Retrieve the Data.

**Development Environment:**

IDE: Spring Boot Suite

Version Control: GitHub

Build System: Maven

Class Diagram: -

**Entity classes**

RoomType

- roomTypeId : int

- roomType : String

Rooms

- roomId : int

- price : int

- roomTypeId : int

- totalRoomAvaliable : int

- totalRoomBooked : int

- isACAvailable: String

- isSmokingAvailable :String

- amenties: text

**Controller**

HomePageController

+ editRenderUrl1()

+ bookRoomUrl()

+ homeCall()

+ ManageRoomUrl()

+ ManageBookingUrl()

RoomController

+ addRoomData()

+ addRoomDataForUpdate()

+ fetchBookingDetails()

+ fetchBookingFilterDetails()

+ deleteRoomData()

+ ErrorPage()

RoomsTypeController

+ fetchRoomType()

+ getRoomTypeAllData()

**Repository**

RoomTypeRepository

+ save (roomType : RoomType)

+ findById(roomId: RoomType)

+findByroomType(roomType: RoomType)

RoomRepository

+ save (room : Room)

+ findById (roomId: room)

**Service**

RoomTypeService

+ getRoomDeatils();

RoomsService

+ fetchAllRoomData();

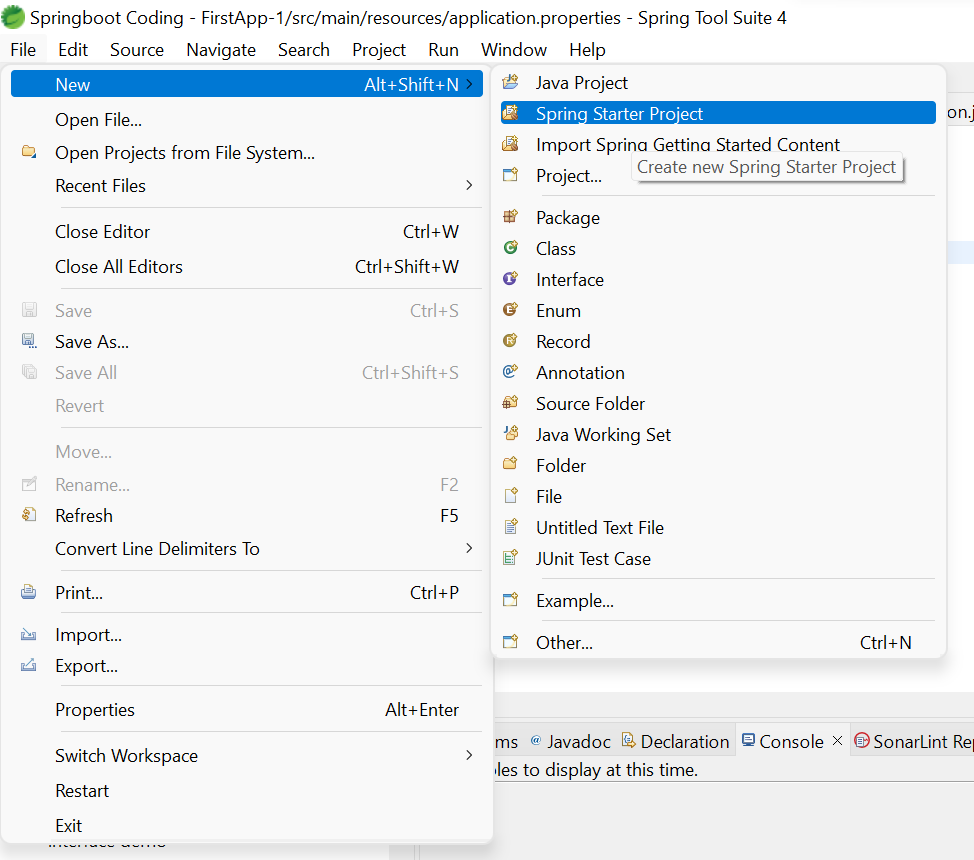
+addRoomDetails(amenitiesjsonData :JSON, formData : JSON)

+deleteRoomDetails(roomid : int);

+fetchAllRoomDataFilterData(roomtype :String);

**Backend Development process:**

**1.Create a new Spring Starter project using Spring tool Suite.**



A screenshot of a computer program

Description automatically generated with low confidence

**Project Structure: -**

**A screenshot of a computer

Description automatically generated with medium confidence**

**2.add below Dependency in pom.xml file.**

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

<!-- https://mvnrepository.com/artifact/com.fasterxml.jackson.core/jackson-databind -->

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

<version>10.1.8</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

**3.add below properties in application. Properties file.**

**application. Properties**

spring.mvc.view.prefix=/views/

spring.mvc.view.suffix=.jsp

server.servlet.context-path=/infy

logging.level.root=info

logging.file.name=errorlog.log

#logging.file.path=errorlog.log

logging.file.path=logs

server.port=8084

spring.datasource.driverClassName=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/hotelmanagment

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**3.. Design the Database Tables for hotel management project Using MYSQL DB.** Shown on below: -

**DB Details:**

|  |  |
| --- | --- |
| Schema Name | hotelmanagement |
| Tables | Room Type, Rooms, BookingDetails |

**Tables Structure:**

|  |  |
| --- | --- |
| **RoomType** | |
| Field Name | Data Type |
| roomTypeId (Primary Key) | int, pk |
| Room Type | VARCHAR (255) |

|  |  |
| --- | --- |
| **Rooms** | |
| Field Name | Data Type |
| roomId (Primary Key) | int, pk |
| isACAvailable | VARCHAR (255) |
| roomTypeId | int, fk |
| isSmokingAvailable | VARCHAR (255) |
| amenties | VARCHAR (255) |

|  |  |
| --- | --- |
| BookingDetails | |
| Field Name | Data Type |
| bookingDetailsId (Primary Key) | INT,pk |
| roomId | Int,fk |
| bookingDates | Date |
| customerName | VARCHAR (255) |
| customerEmail | VARCHAR (255) |
| customerMobile | VARCHAR (255) |
| desc | VARCHAR (255) |

3.once the Database table design is ready, write the controller, service, Dao layers to execute CURD operations.

4. build the front end of the web app using **JSP** to accept input and render the data correspondingly.

5.integrate the controller with the database to successfully build the hotel management system web application.

**UI Model:**

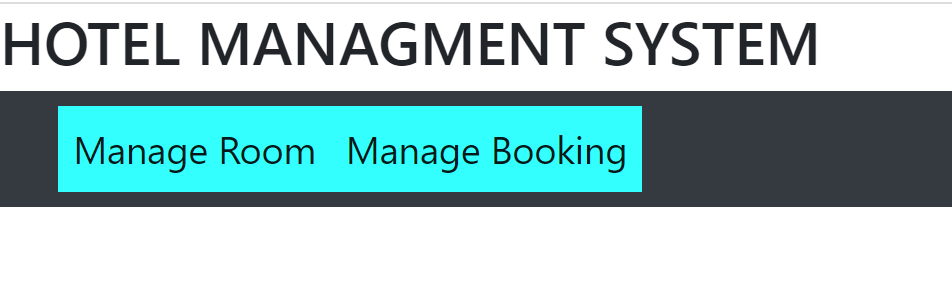
Different actors:

|  |  |
| --- | --- |
| ADMIN | Can add/update/book /delete Rooms |
| CUSTOMER | Enquiry / views Rooms |

**jQuery Data table Field list: -**

|  |  |  |
| --- | --- | --- |
| jQuery Table Field | Visible for Admin | Visible for Customer |
| Room Type | true | true |
| price | true | true |
| AC/Non-AC | true | true |
| Smoking/Non-Smoking Room | true | true |
| WI-FI/TV/Geyser | true | true |
| Action (Edit/delete bottom) | true | false |

URL: <http://localhost:8084/infy/homepage>



Manage Room: -This Page hold Information of rooms availability, Admin can view, add, edit, delete Room Details.

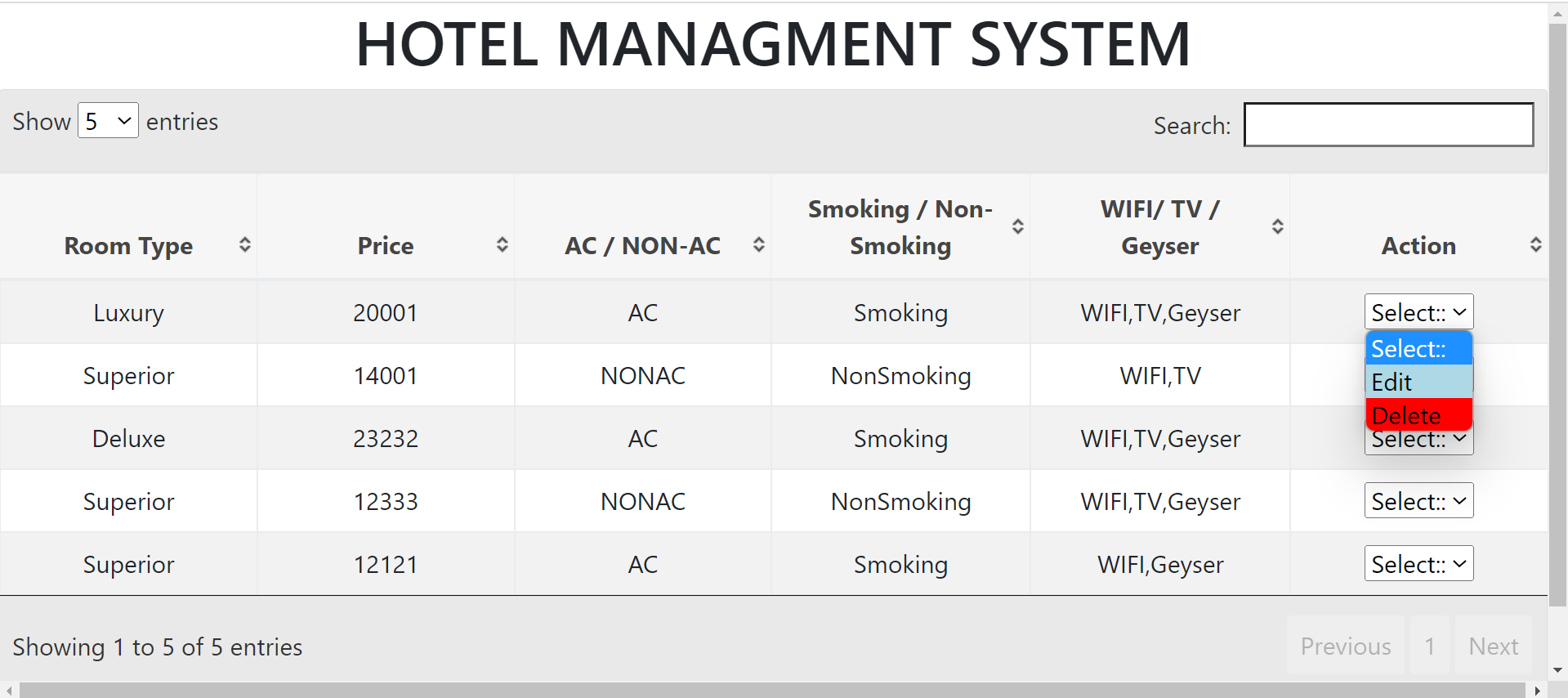
Manage Booking: - Costumer can view all rooms and they can Book rooms based on their requirement.

1. Manage Room: -

URL: http://localhost:8084/infy/ManageRoom

A screenshot of a computer

Description automatically generated with medium confidence



Manage Room: -

Manage room hold above fields in jQuery table refer above SS.

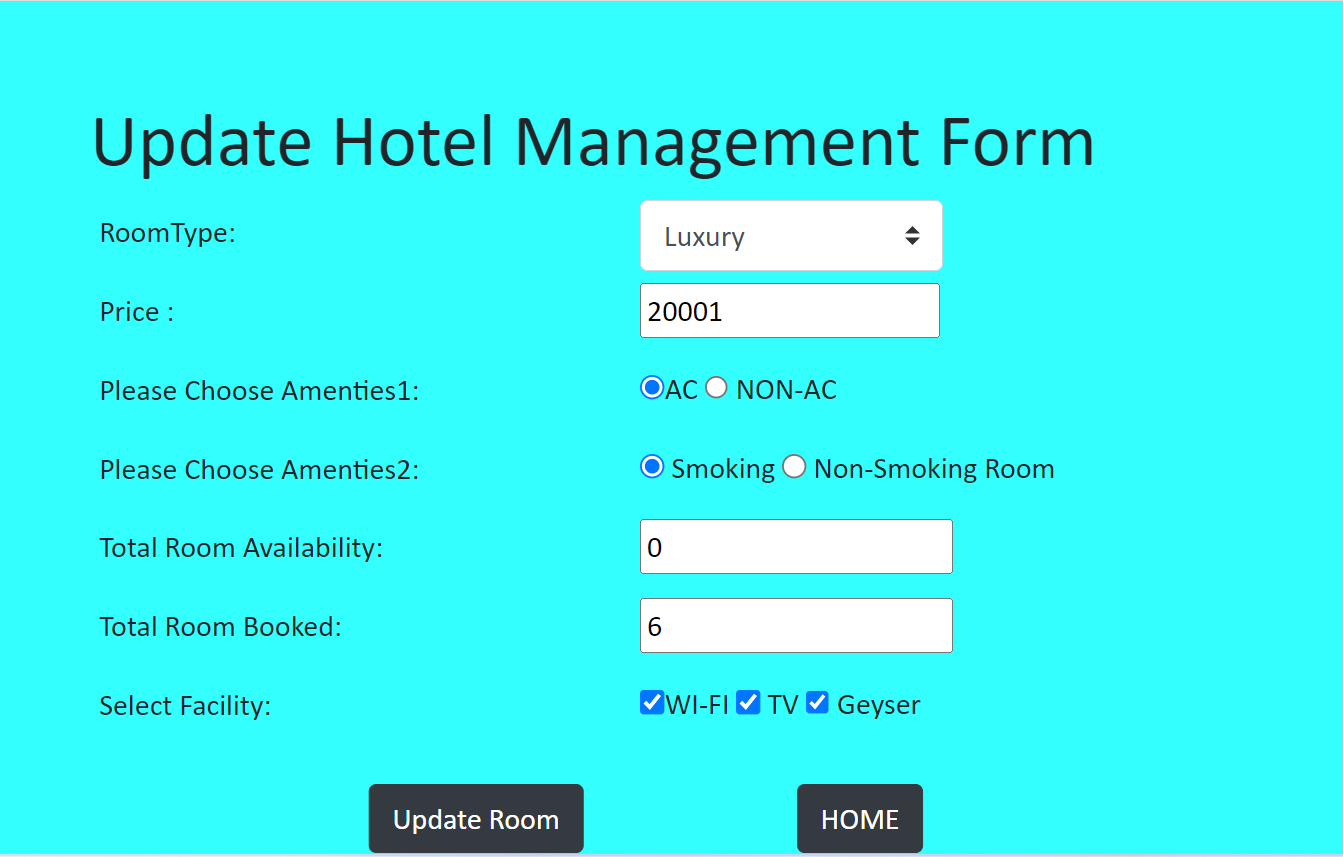
Admin can render Edit, Delete page using Action Dropdown.

Admin can also render Add page using Add Room button.

Admin can Show entries based on their preferences [5,10,15,20], default entry showing as “5”.

Edit Room: -

URL: http://localhost:8084/infy/editroomdetails



when Admin user click on Action dropdown edit button, it renders Edit page refer above SS.

Based on Click, edit room button, same row all Room’s records are forwarded to edit page for updating.

Java Script level validation present here if input is missing or incorrect it shows error pop-up when Admin click “Update Room” button. Refer below SS.

A screenshot of a computer

Description automatically generated with low confidence

Edit page contain “HOME” button it renders back to home page.

Add Room: -

URL: http://localhost:8084/infy/addCostumer

A picture containing text, screenshot, font, number

Description automatically generated

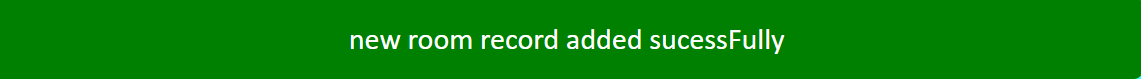
when Admin user click on Add button, it renders “Add page form” refer above SS.

Java Script level validation present here if input is missing or incorrect it shows error pop-up when Admin click “ADD ROOM” button. Refer above SS.

Edit page contain “HOME” button it renders back to home page.

Room Type dropdown data by default fetch from DB.

If record Added Successfully, showing below message.



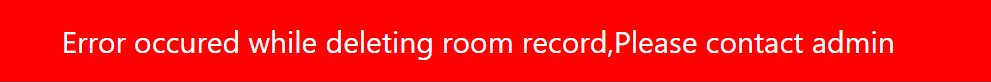
else 

Delete Room: -

When Admin click delete room button, same row all Room’s record data Deleted from tables, showing below pop-up message.



If any Error Occurs,



2. Booking Room: -

A screenshot of a computer

Description automatically generated with medium confidence

Booking Room: -

Booking room hold above fields in jQuery table refer above SS.

User can render “Book Room”, “Delete Room” page using Action Dropdown.

User can view entries based on their preferences [5,10,15,20], default entry showing as “5”.

if Room Availability equals Room Booked, “Book Room” option Disabled for User.

Search Filter based on “Room type” available for User. refer below SS.

A screenshot of a computer

Description automatically generated with medium confidence

**If Based on Room Type Data not available Showing below pop-up massage.**

A screenshot of a computer

Description automatically generated with medium confidence