Project title : Hotel management system project
Date :20-06-2023

Table of Contents

Background and Overview:

The Hotel Management System is a Java-based application designed to streamline and automate various hotel management operations. It aims to provide an efficient and user-friendly solution for managing hotels, rooms, guests, and reservations. By automating these processes, the system simplifies the tasks of hotel staff and enhances the overall guest experience.

Purpose:

The purpose of the Hotel Management System is to facilitate the smooth functioning of hotel operations and improve productivity. It eliminates the need for manual paperwork and reduces the chances of errors in managing hotel-related tasks. The system enables hotel administrators to efficiently handle reservations, allocate rooms, manage guest information,

Objectives:

The Hotel Management System is designed to facilitate the management of hotels, their rooms, guests, and reservations. It provides a comprehensive solution for hotel administrators and staff to efficiently handle various aspects of hotel operations

Reservation Management:
Easy creation, modification, and cancellation of reservations, ensuring accurate room availability and guest preferences.
Room Allocation:

Optimize room assignments based on preferences, availability, and criteria, minimizing overbooking and mismanagement.

Guest Management:

Centralize guest information for efficient check-ins, check-outs

System Architecture:

The Hotel Management System follows a modular and scalable architecture. It consists of different modules or components, including reservation management, room management, guest management, front desk operations, billing and invoicing, and reporting. These modules interact with a central database that stores hotel, room, guest, and reservation data.
System architecture

Package Structure :

The project is organized into the following packages:

1. com.tpe.HotelManagementSystem.main

   - Main class and entry point of the application.
   - Contains the main method and the displayMenuHotelManagementSystem() method.

2. com.tpe.HotelManagementSystem.service

   - Contains service interfaces and their implementations for managing hotels, rooms, guests, and reservations.

3. com.tpe.HotelManagementSystem.exception

- Custom exceptions used in the application.

4. com.tpe.HotelManagementSystem.model

- Contains the entity classes representing hotels, rooms, guests, and reservations.

5. com.tpe.HotelManagementSystem.repository

- Contains repository interfaces and their implementations for data access.

6. com.tpe.HotelManagementSystem.repository.hibernate

- Contains Hibernate-based implementations of the repository interfaces.

9. com.tpe.HotelManagementSystem.config

- Configuration files and classes for setting up dependencies and database connections.

10. com.tpe.HotelManagementSystem.resources

- Additional resources such as SQL scripts or property files.

11. com.tpe.HotelManagementSystem.tests

- Contains unit tests and integration tests for the application.

Dependencies

- Java 8 or higher

- Hibernate (version 5.6.9.Final)
- JDBC (version 42.3.6)
- Other dependencies...

In the com.tpe.HotelManagementSystem.service package, you'll find service interfaces and their implementations for managing hotels, rooms, guests, and reservations.

Service Interfaces and Implementations:

HotelService:

Provides methods for managing hotels, such as adding, finding, deleting, updating, and retrieving hotels.

RoomService:

Provides methods for managing rooms, such as adding, finding, deleting, updating, and retrieving rooms.

GuestService:

Provides methods for managing guests, such as adding, finding, deleting, updating, and retrieving guests.

ReservationService:

Provides methods for managing reservations, such as adding, finding, deleting, updating, and retrieving reservations.

In the com.tpe.HotelManagementSystem.repository package, you'll find repository interfaces and their implementations for data access related to hotels, rooms, guests, and reservations.

Repository Interfaces and Implementations:

HotelRepository:

Provides methods for accessing and manipulating hotel data, such as saving a hotel, finding a hotel by ID, deleting a hotel, updating a hotel, and retrieving all hotels.

RoomRepository:

Provides methods for accessing and manipulating room data, such as saving a room, finding a room by ID, deleting a room, updating a room, and retrieving all rooms.

GuestRepository:

Provides methods for accessing and manipulating guest data, such as saving a guest, finding a guest by ID, deleting a guest, updating a guest, and retrieving all guests.

ReservationRepository:

Provides methods for accessing and manipulating reservation data, such as saving a reservation, finding a reservation by ID, deleting a reservation, updating a reservation, and retrieving all reservations

In the com.tpe.HotelManagementSystem.exception package, you'll find custom exception classes that are used in the Hotel Management System.

Custom Exception Classes:

HotelNotFoundException:

This exception is thrown when a hotel is not found in the system.

RoomNotFoundException:

This exception is thrown when a room is not found in the system.

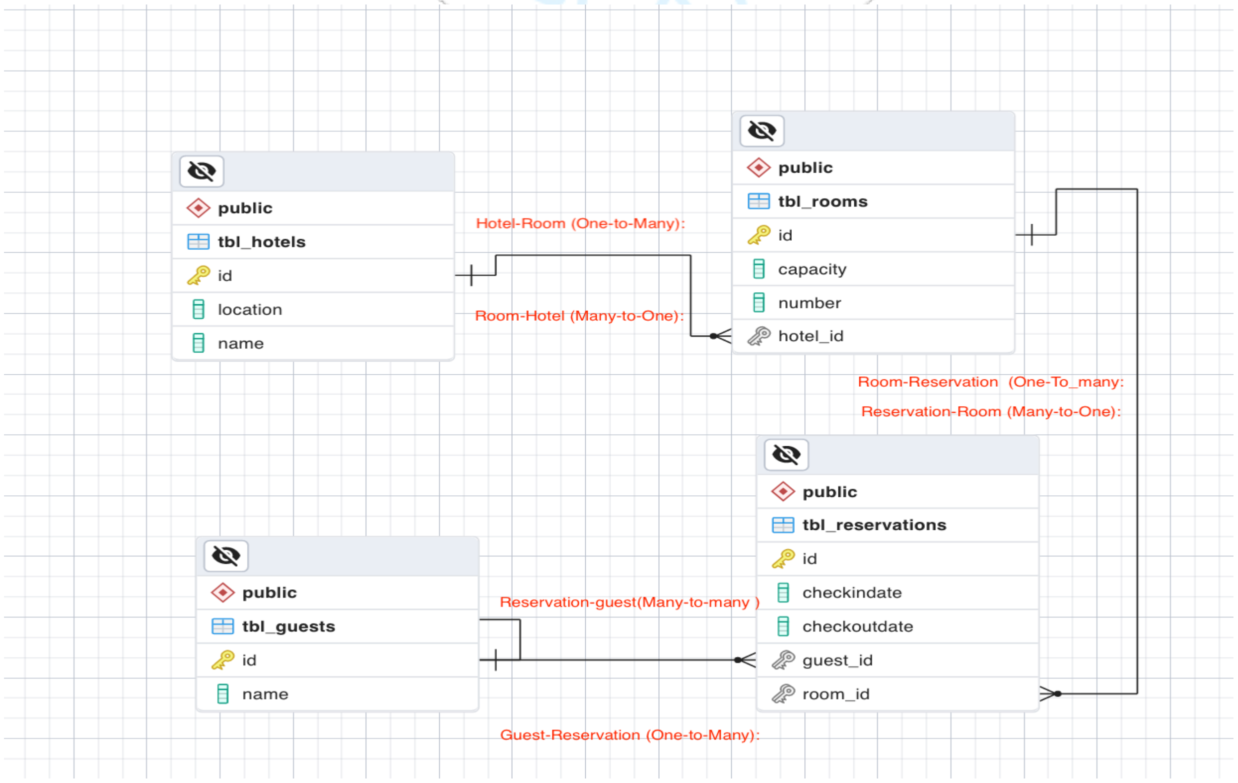
GuestNotFoundException:

This exception is thrown when a guest is not found in the system.

ReservationNotFoundException:

This exception is thrown when a reservation is not found in the system.

Entity diagram :


public
tbl_hotels
id
location
name
Hotel-Room (One-to-Many):
Room-Hotel (Many-to-One):
public
tbl_rooms
id
capacity
number
hotel_id
Room-Reservation (One-To_many:
Reservation-Room (Many-to-One):
public
tbl_reservations
id
checkindate
checkoutdate
guest_id
room_id
public
tbl_guests
id
name
Reservation-guest(Many-to-many)
Guest-Reservation (One-to-Many):

Entities

The entities in the Hotel Management System are typically located in the com.tpe.HotelManagementSystem.model package. This package contains the entity classes representing hotels, rooms, guests, and reservations. Here is a breakdown of the entities and their properties:

Hotel:

Properties: ID, name, location
Represents a hotel entity with a unique ID, name, and location.

Room:

Properties: ID, number, type
Represents a room entity with a unique ID, room number, and type.
Each room belongs to a specific hotel and can have multiple reservations.

Guest:

Properties: ID, name, contact information
Represents a guest entity with a unique ID, name, and contact information.
Each guest can make multiple reservations.

Reservation:

Properties: ID, guest, room, dates
Represents a reservation entity with a unique ID, associated guest, room, and reservation dates.
Each reservation is linked to a specific guest and room.

Hotel Management System relationships

Hotel-Room (One-to-Many):

A hotel can have multiple rooms, but each room belongs to only one hotel.Each room has a reference to the hotel it belongs to.

Guest-Reservation (One-to-Many):

A guest can make multiple reservations, but each reservation is associated with only one guest. Each reservation has a reference to the guest it is associated with.

Reservation-Room (Many-to-One):

A reservation is made for a specific room, and each room can have multiple reservations. Each reservation has a reference to the room it is reserved for.

Room-Hotel (Many-to-One):

Each room belongs to a specific hotel, and each hotel can have multiple rooms. Each room has a reference to the hotel it belongs to.

CRUD OPERATION

common CRUD (Create, Read, Update, Delete) operations typically performed on each repository in the Hotel Management System:

Hotel Repository:

createHotel(Hotel hotel):	Creates a new hotel and saves it to the database.
findHotelById(int id):	Retrieves a hotel from the database based on its ID.
updateHotel(Hotel hotel):	Updates an existing hotel in the database.
deleteHotel(int id):	Deletes a hotel from the database based on its ID.

findAllHotels():	Retrieves all hotels from the database.

Room Repository:

createRoom(Room room):	Creates a new room and saves it to the database.
findRoomById(int id):	Retrieves a room from the database based on its ID.
updateRoom(Room room):	Updates an existing room in the database.
deleteRoom(int id):	Deletes a room from the database based on its ID.
findAllRooms():	Retrieves all rooms from the database.

Guest Repository:

createGuest(Guest guest):	Creates a new guest and saves it to the database.
findGuestById(int id):	Retrieves a guest from the database based on their ID.
updateGuest(Guest guest):	Updates an existing guest in the database.
deleteGuest(int id):	Deletes a guest from the database based on their ID.
findAllGuests():	Retrieves all guests from the database.

Reservation Repository:

createReservation(Reservation reservation):	Creates a new reservation and saves it to the database.
findReservationById(int id):	Retrieves a reservation from the database based on its ID.
updateReservation(Reservation reservation):	Updates an existing reservation in the database.
deleteReservation(int id):	Deletes a reservation from the database based on its ID.
findAllReservations():	Retrieves all reservations from the database

==== Hotel Management System Menu ====

Please select an option:

1. Hotel Operations
2. Room Operations
3. Guest Operations
4. Reservation Operations
5. Exit

Enter the number corresponding to your choice: [1-5]

Option 1: Hotel Operations

1. Add a new hotel
2. Find a hotel by ID
3. Update a hotel
4. Delete a hotel
5. View all hotels

Option 2: Room Operations

1. Add a new room
2. Find a room by ID
3. Update a room
4. Delete a room
5. View all rooms

Option 3: Guest Operations

1. Add a new guest
2. Find a guest by ID
3. Update a guest
4. Delete a guest
5. View all guests

Option 4: Reservation Operations

1. Add a new reservation
2. Find a reservation by ID
3. Update a reservation
4. Delete a reservation
5. View all reservations
6. Option 5: Exit

Exit the Hotel Management System

Conclusion

The Hotel Management System is a Java-based application designed to facilitate the management of hotels, rooms, guests, and reservations. It provides various functionalities for adding, updating, deleting, and retrieving information related to hotels and their operations