Hotel Booking System

Description

1. The Hotel Booking System is a Java-based application designed to manage hotel room reservations. It allows users to search for available rooms, make bookings, view reservation details, and manage customer information. The system provides functionalities for both hotel staff and guests, including checking in, checking out, and updating booking details. It uses object-oriented principles to organize data into classes such as `Room`, `Booking`, ensuring a modular and efficient structure. With a user-friendly interface and database integration, the system streamlines hotel operations and enhances customer experience.

Project Requirements

1. **Add Guest:** Add new guest records to the system with basic information such as name, contact details, and ID proof.
2. **Edit Guest:** Modify existing guest details like name, contact information, or ID proof.
3. **Delete Guest:** Remove a guest's record from the system and store it in a guest archive for future reference.
4. **Add Booking:** Book a room for a guest by specifying details such as check-in date, check-out date, room type, and special requests.
5. **Edit Booking:** Update booking details, such as dates, room type, or special requests.
6. **View Available Rooms:** Display a list of available rooms filtered by date, room type, and capacity.
7. **Edit Room:** Modify room details such as price, capacity, or status (e.g., available, under maintenance).
8. **Price calculation:** Will automatically calculate the price for the chosen room, considering room type and number of days staying.
9. **Clear Input Fields:** Clear and understandable input fields for adding or editing records.

   10. **Database:** Develop a working database for easy and efficient
   .   work of the system.

Team Members List:

- **Ten Artem:** Interface designer, Project manager.
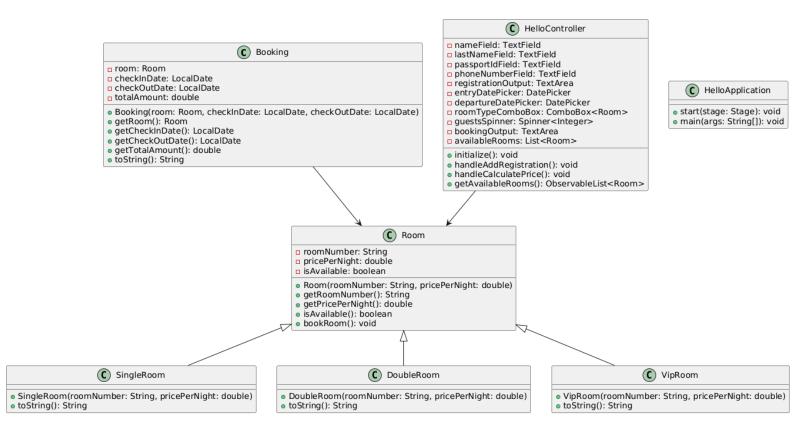- **Eremin David:** Class developer, Tester.

Role of group members:

Ten Artem: Designed interface and wrote code for HelloController, integrated database functions.

Eremin David: Wrote abstract class and subclasses, made presentation and worked on the README file.

UML Class Diagram

The UML class diagram provides a visual representation of the system's structure, detailing the classes, attributes, methods, and their relationships.

Booking

- room: Room
- checkInDate: LocalDate
- checkOutDate: LocalDate
- totalAmount: double

- Booking(room: Room, checkInDate: LocalDate, checkOutDate: LocalDate)
- getRoom(): Room
- getCheckInDate(): LocalDate
- getCheckOutDate(): LocalDate
- getTotalAmount(): double
- toString(): String

HelloController

- nameField: TextField
- lastNameField: TextField
- passportIdField: TextField
- phoneNumberField: TextField
- registrationOutput: TextArea
- entryDatePicker: DatePicker
- departureDatePicker: DatePicker
- roomTypeComboBox: ComboBox<Room>
- guestsSpinner: Spinner<Integer>
- bookingOutput: TextArea
- availableRooms: List<Room>

- initialize(): void
- handleAddRegistration(): void
- handleCalculatePrice(): void
- getAvailableRooms(): ObservableList<Room>

HelloApplication

- start(stage: Stage): void
- main(args: String[]): void

Room

- roomNumber: String
- pricePerNight: double
- isAvailable: boolean

- Room(roomNumber: String, pricePerNight: double)
- getRoomNumber(): String
- getPricePerNight(): double
- isAvailable(): boolean
- bookRoom(): void

SingleRoom

- SingleRoom(roomNumber: String, pricePerNight: double)
- toString(): String

DoubleRoom

- DoubleRoom(roomNumber: String, pricePerNight: double)
- toString(): String

VipRoom

- VipRoom(roomNumber: String, pricePerNight: double)
- toString(): String

Weekly Meeting Documentation

[Weekly Meetings Documentation](#)