Hotel Reservation System

Krish Sarkar (E055)

Mithil Save (E056)

Manan Shah (E062)

**Introduction**

This Java application is a Hotel Reservation System, allowing users to view available rooms, make bookings, and cancel reservations.

1. **Room Types**: The hotel has two room types: Suite Room and Standard Room. Each room has attributes like room number, price, and availability status.
2. **Reservation Options**:
   * **View Available Rooms**: Displays rooms that are currently free to book, showing their room numbers and prices.
   * **Book a Room**: Users input the room number and provide customer details (name, email, phone) to book a specific room. If available, a Reservation object is created, linking the customer and room.
   * **Cancel Reservation**: Users can cancel a reservation by entering the room number. The room's availability is updated accordingly.
3. **Exit**: Exits the system.

The main loop offers these options continuously, allowing users to manage reservations efficiently. The application ensures that only available rooms can be booked and only reserved rooms can be cancelled.

**Software & Hardware requirements**

1. **IDE:**

**IntelliJ IDEA** or **Eclipse**: Integrated Development Environments (IDEs) for Java development, offering features like code completion, debugging, and project management, which make writing and testing Java applications efficient.

1. **Java Version:**

**Java 8 or later**: Java SE 8 has features like lambdas and the Stream API, which are widely used in Java applications. The code is compatible with Java 8, but using Java 11 or Java 17 (LTS versions) is recommended for long-term support and modern enhancements in Java’s runtime performance and security.

1. **Libraries:**

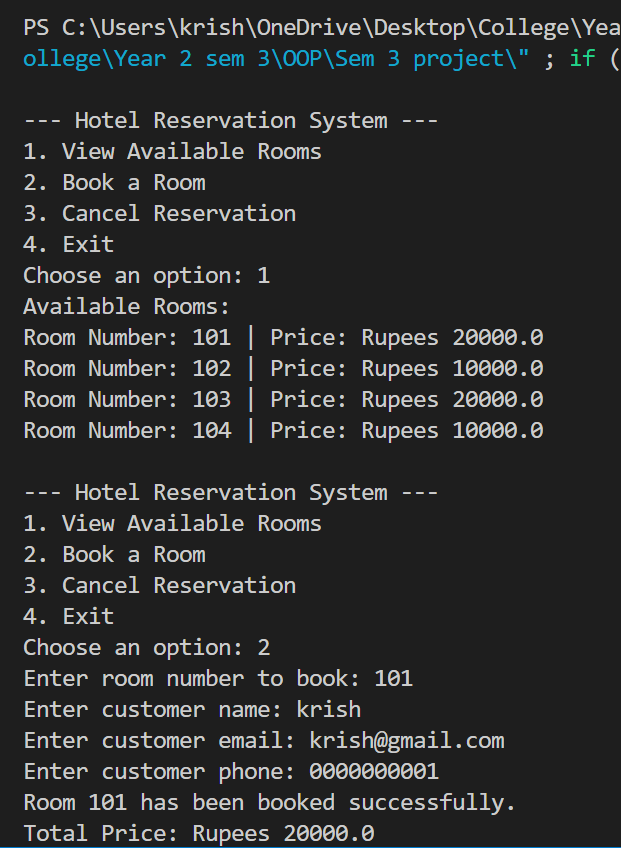
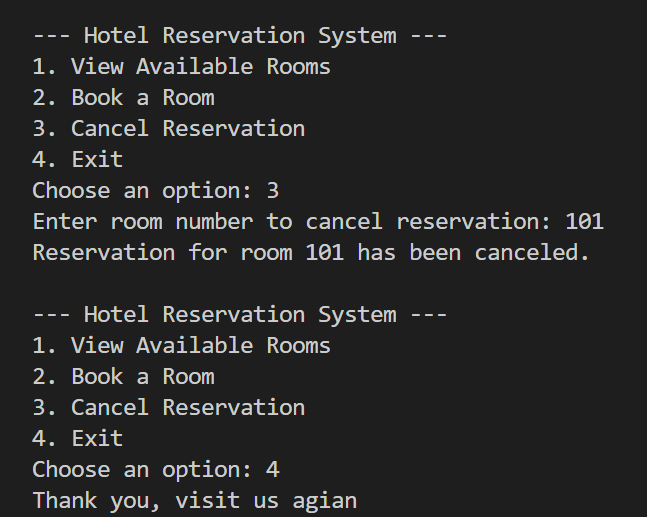
This application relies on standard Java libraries (java.util.Scanner) and does not require any external dependencies, making it straightforward and easy to run on any Java platform.

1. **Execution Platform:**

* **Java Virtual Machine (JVM)**: The application runs on the JVM, allowing it to be executed on any OS (Windows, macOS, Linux) with the correct Java version installed.
* **Command-Line Interface (CLI)**: The program can be compiled and run directly from the command line, especially for testing purposes without an IDE.

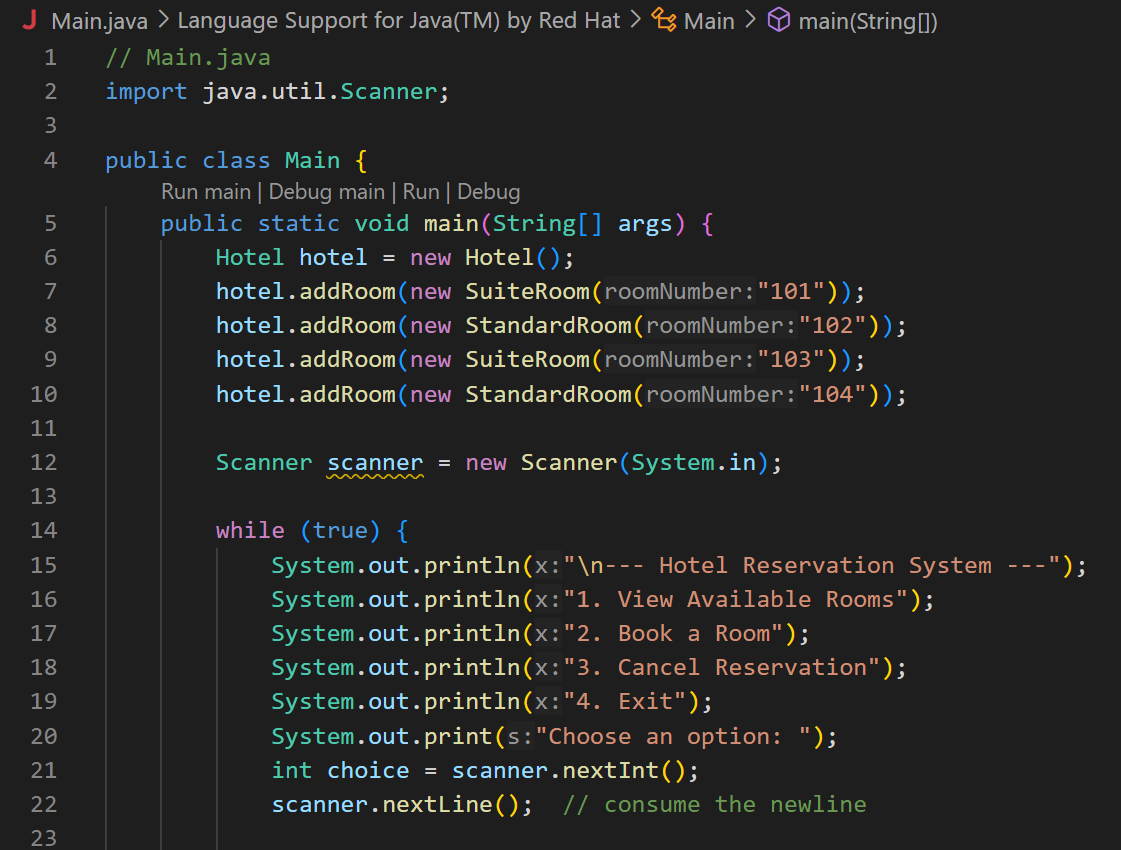
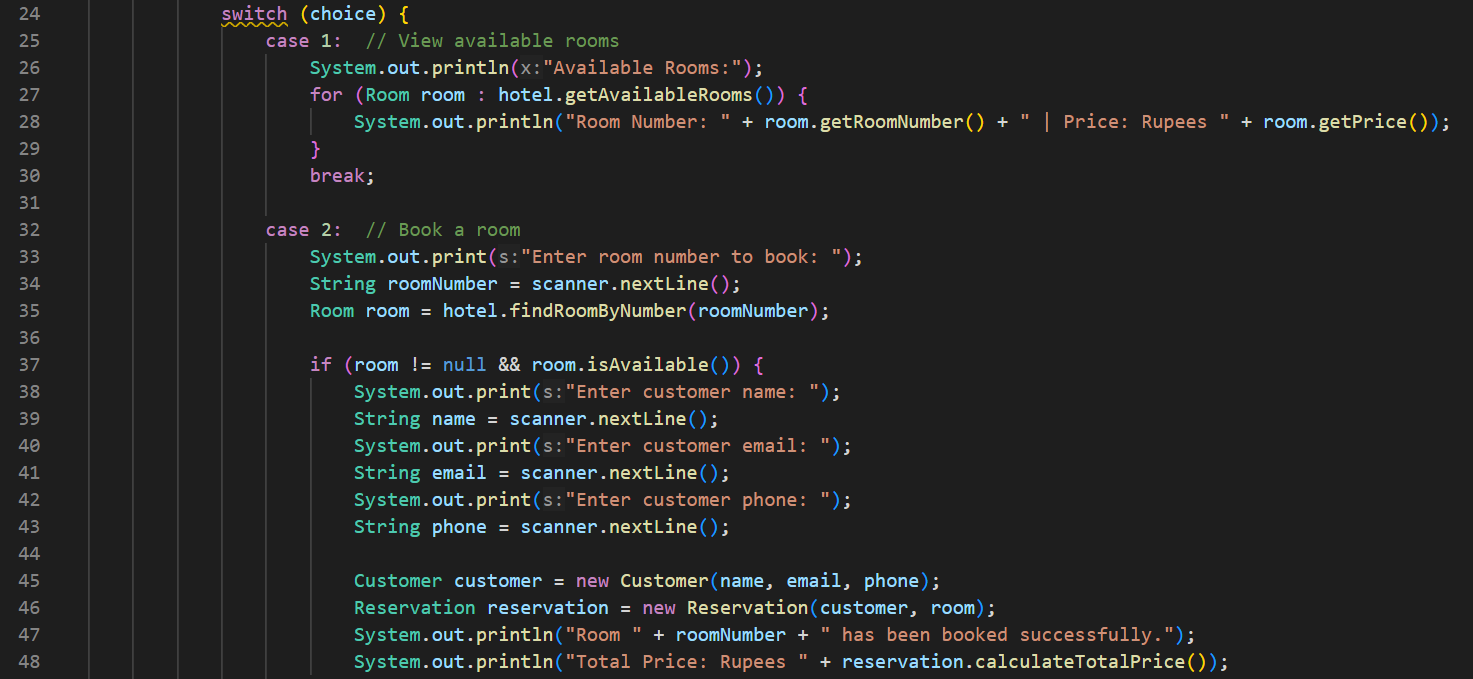
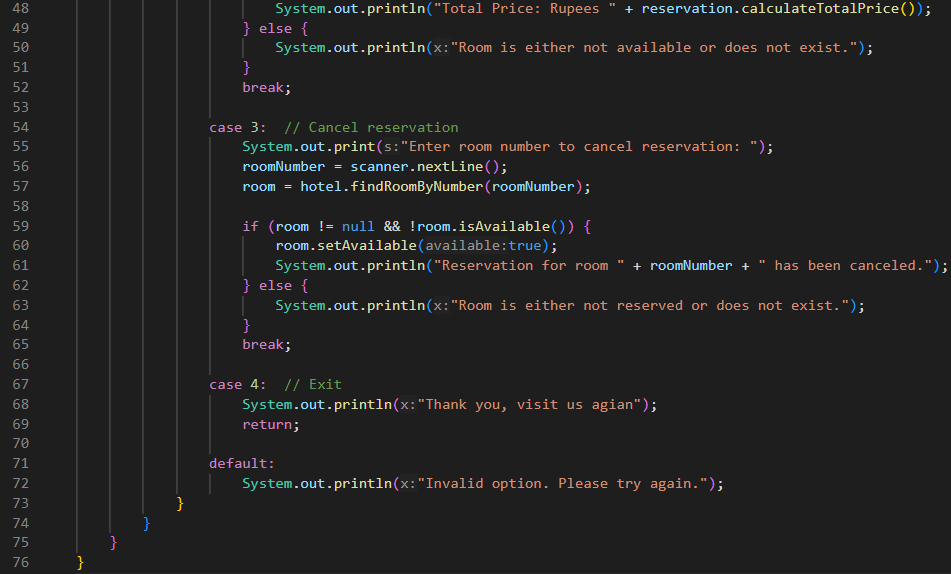
**UML Diagram**

**Screenshots**

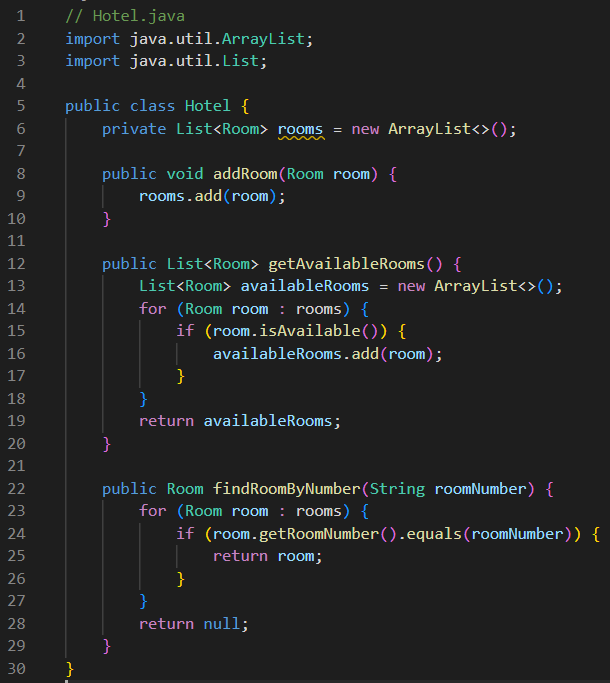
  


Code Snippets

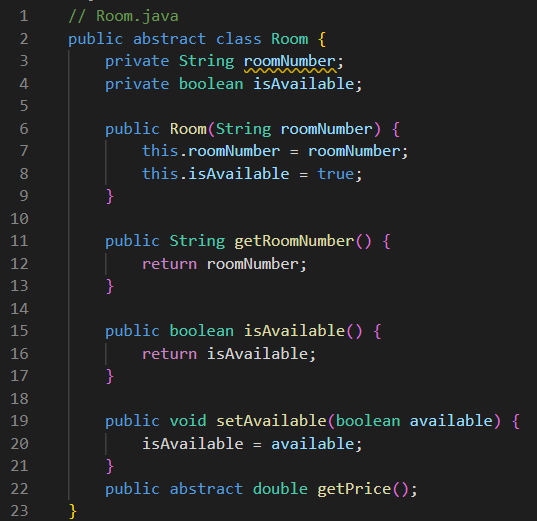
**Main class:**

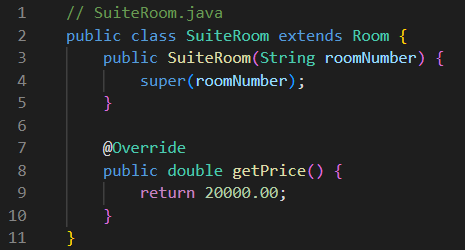
**Hotel class:**



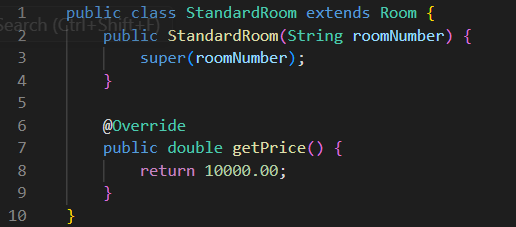
**Room** (Abstract Class):



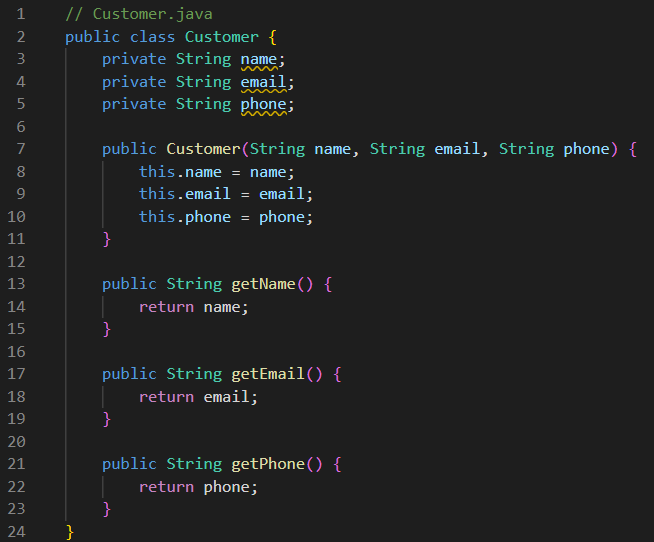
**SuiteRoom** (Subclass of Room):



**StandardRoom** (Subclass of Room):



**Customer class:**



**Reservation class:**



**Conclusion:**

The Hotel Reservation System is a simple Java-based application that allows users to manage room reservations. Through a console-based interface, users can view available rooms, make bookings, and cancel reservations. The program uses object-oriented principles to separate concerns across various classes, making it easy to maintain and extend. This design ensures the application's core functionality remains straightforward and effective for a basic hotel booking system.

**Future Scope:**

1. **Graphical User Interface (GUI)**: Adding a GUI would enhance user experience, making it more intuitive and visually appealing, and potentially improving accessibility.
2. **Database Integration**: Storing room and reservation data in a database (e.g., MySQL, SQLite) would allow persistent storage, making data management more robust.
3. **Payment Processing**: Integrating a payment gateway for booking confirmations could add real-world functionality, making it suitable for commercial use.
4. **Room Categorization and Dynamic Pricing**: Adding more room types and allowing dynamic pricing (e.g., seasonal rates) would create a more flexible booking system.
5. **Enhanced Reservation Features**: Features like booking history, guest check-in/check-out, and automated reminders could provide a comprehensive system for hotel management.
6. **Web-Based Application**: Transforming it into a web application would allow users to access it remotely, expanding its usability beyond a local setup.