**BITP 3113: OBJECT ORIENTED PROGRAMMING**

**Mini Project Deliverable**

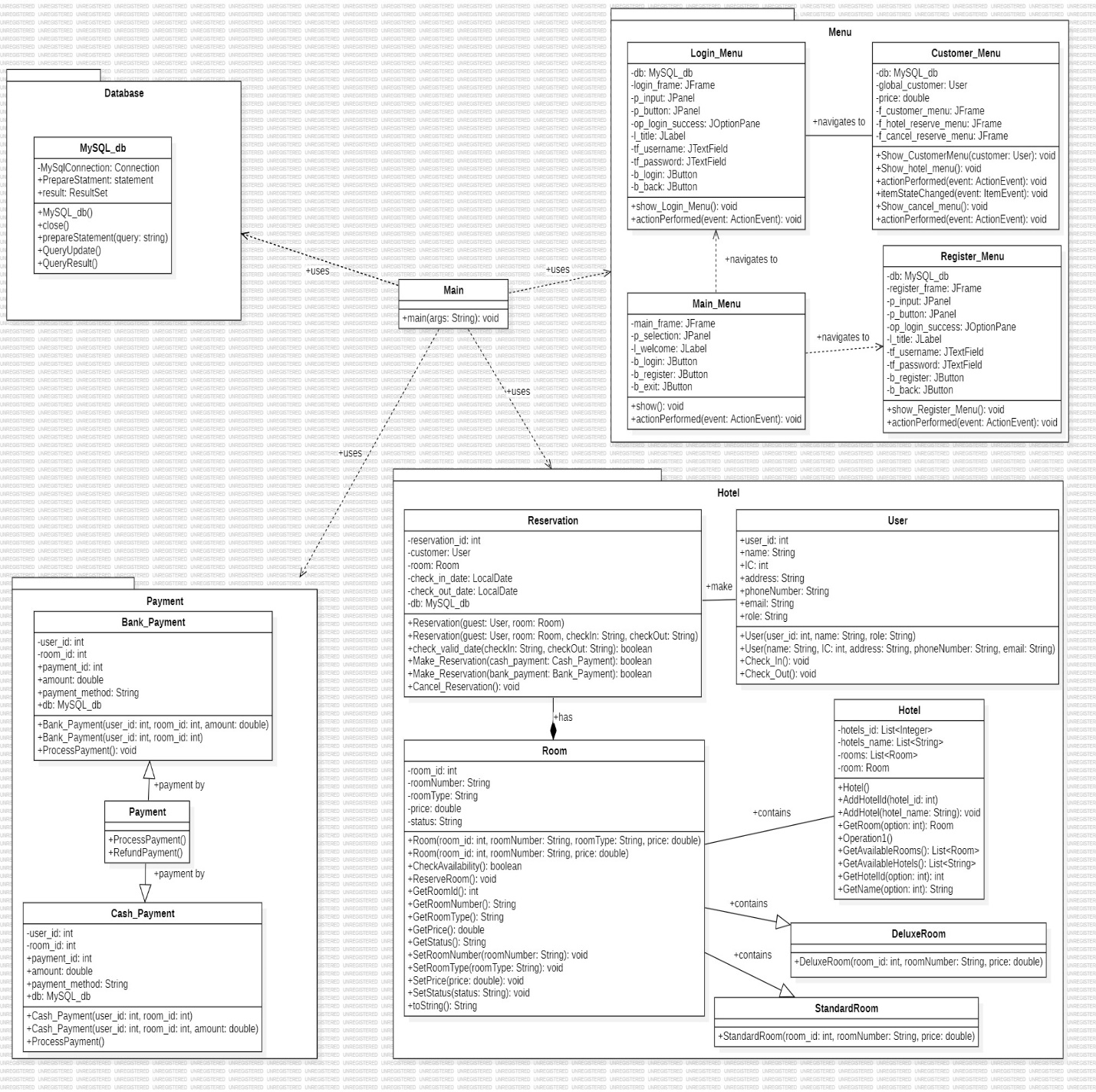
**Project Title:**  Hotel Reservation System

**Team members:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Matric** | **Section/ Group** |
| DING JIA JUN | B032310366 | S1G1 |
| CHEN JIN HAN | B032310333 | S1G1 |
| ELDHON CHEONG JIE QI | B032310606 | S1G1 |
| CHOONG WILLIAM | B032310371 | S1G1 |
| TAN YEE | B032310306 | S1G2 |

**Diagram:**

1. **UML diagrams**

****

1. **GUI**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer register

Description automatically generated**

**A screenshot of a login screen

Description automatically generated**

**A screenshot of a customer menu

Description automatically generated**

**A screenshot of a hotel reservation

Description automatically generated**

**A screenshot of a hotel reservation

Description automatically generated**

**A screenshot of a reservation

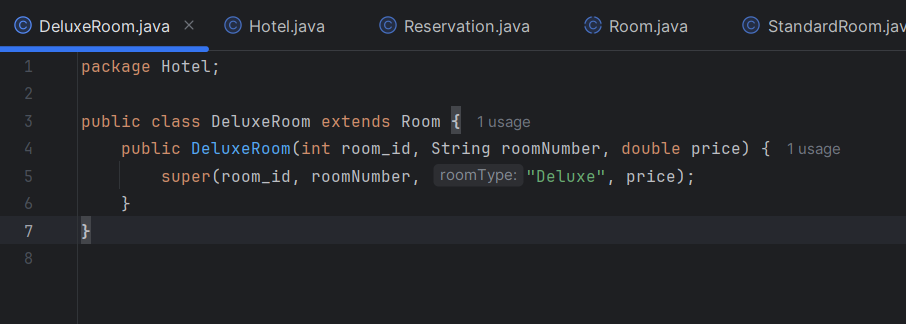
Description automatically generated**

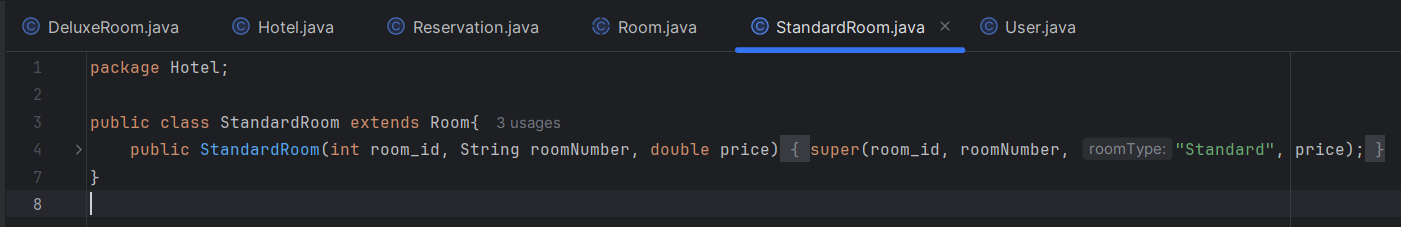
**A screenshot of a reservation

Description automatically generated**

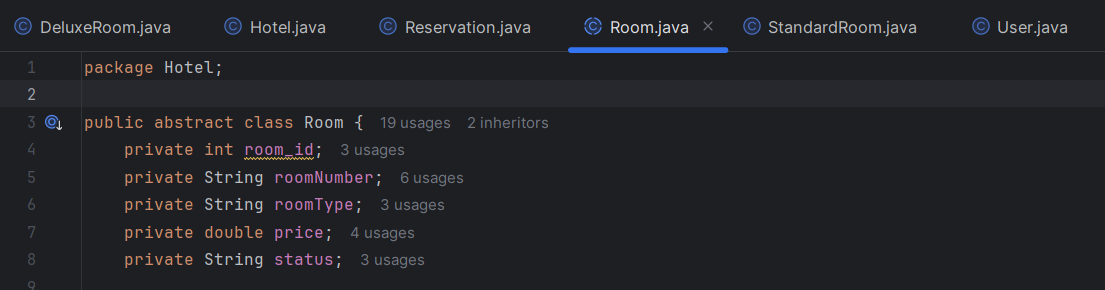
**Code snapshot:**

1. **Inheritance / Abstract**

****

****

- Inheritance is a concept where a class (child class) gets properties and methods from another class (parent class), making it easier to reuse code. DeluxeRoom and StandardRoom extends it, providing a concrete implementation for creating specific room types.



- An abstract class is a class that can't be created directly and is meant to be extended by other classes. This is an example of abstract where Room is an abstract class that cannot be instantiated directly

1. **Polymorphism**

**A screen shot of a computer program

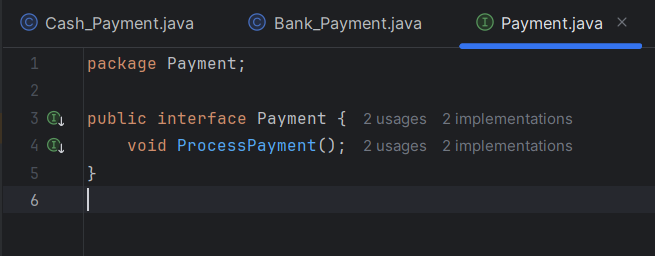
Description automatically generated**

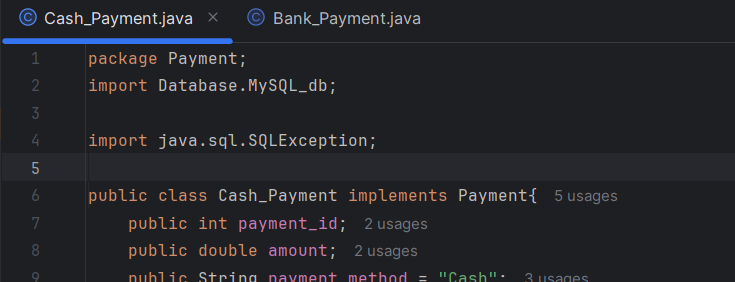
**A screen shot of a computer program

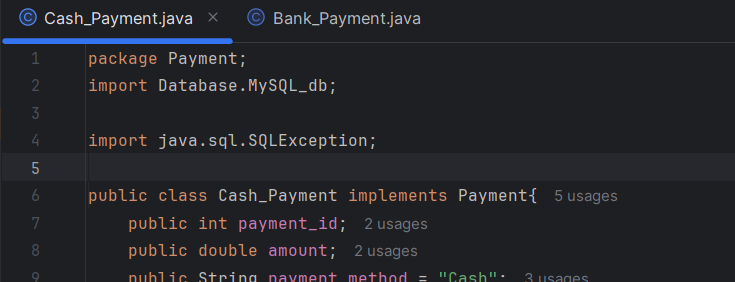
Description automatically generated**

* Methods make reservation in the same class have the same name but different parameters, allowing them to perform similar tasks on different types of inputs.

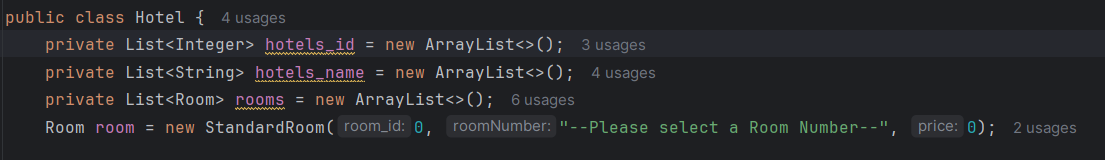
1. **Interface**

****

****

****

1. **Java Collection: List / Set / Map**

****

-The List interface in Java is part of the java.util package and provides an ordered collection. It allows duplicates and maintains insertion order.

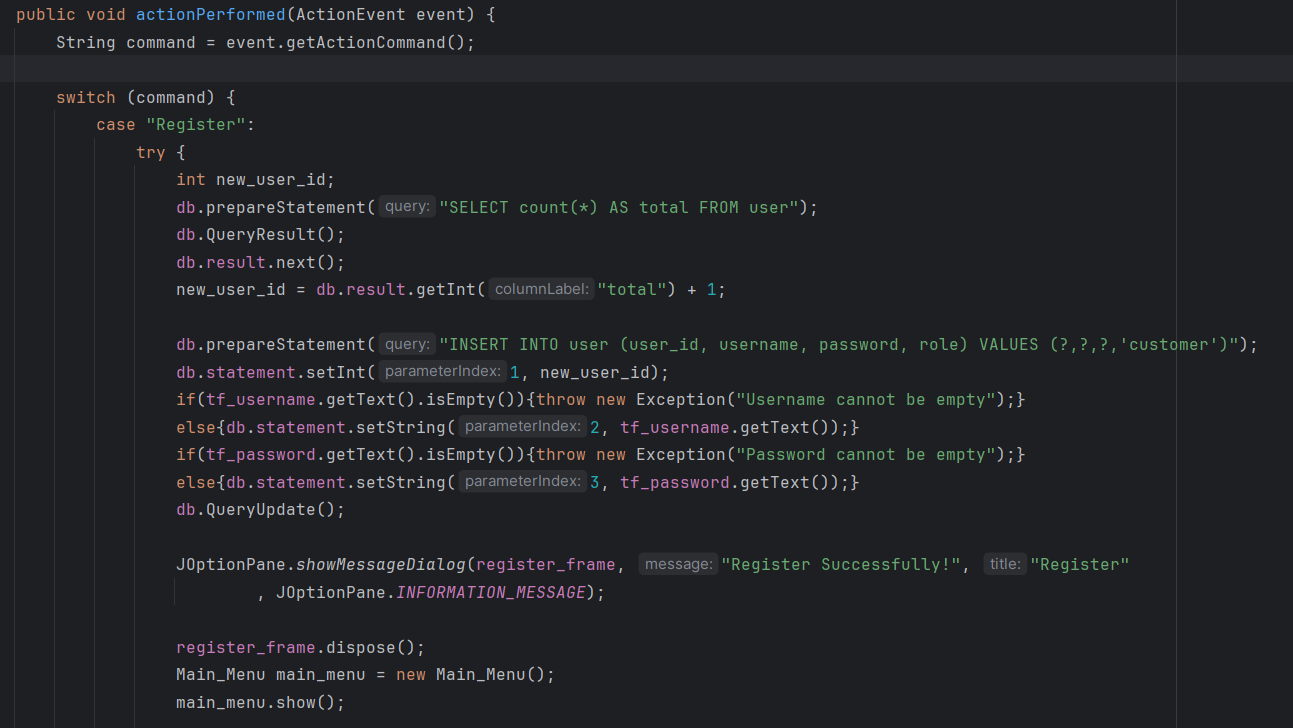
1. **Exception Handling (Create your own)**

**A computer screen shot of a code

Description automatically generated**

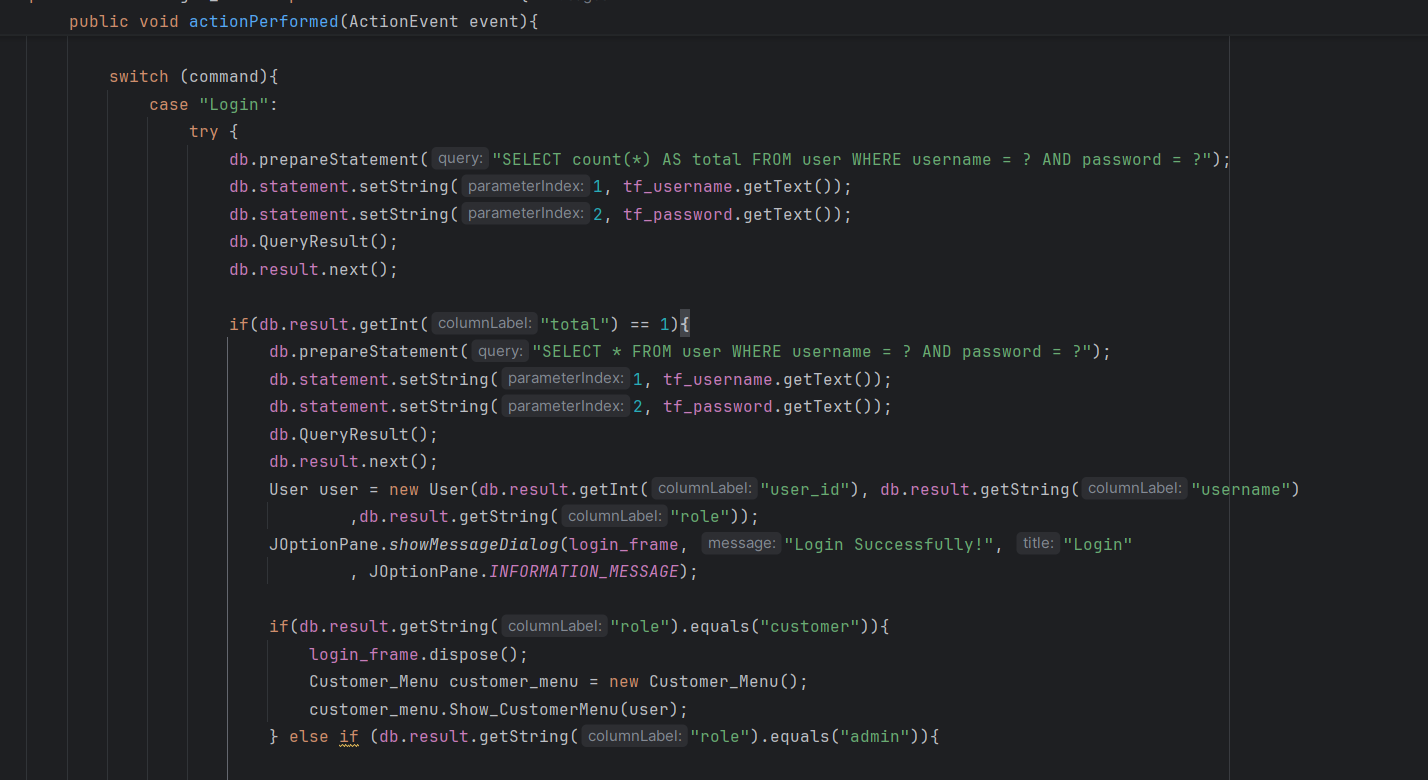
* The DateTimeParseException occurs when a date-time string being parsed does not adhere to the expected format or contains invalid date-time information.

1. **Connecting to database (Insert, view, search, update, delete)**
2. **Insert**

****

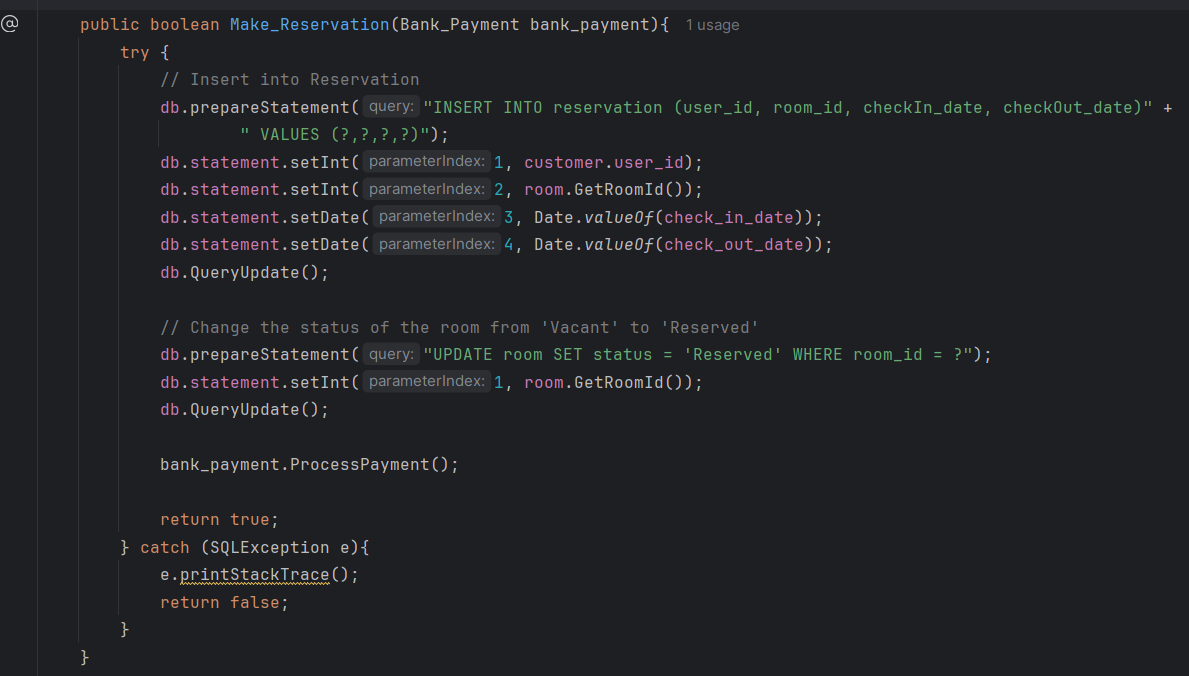
* This code snippet performs an INSERT operation to add a new user into a database table named "user," handling potential empty username/password inputs with exceptions.

1. **Read**

****

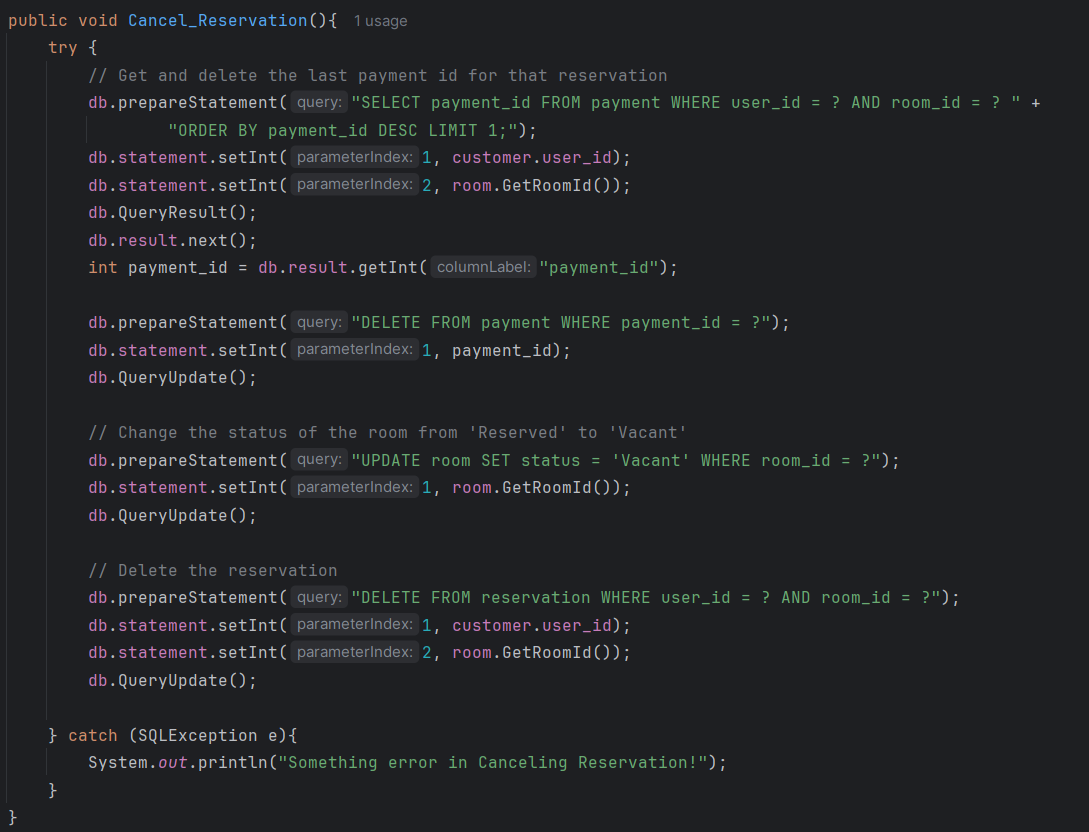
* This code performs VIEW operation, one counting matching users and the other retrieving the user's details, to verify login credentials and fetch user information for subsequent actions.

1. **Update**

****

* This code performs an UPDATE operation on the "room" table, changing the "status" of a specific room to "Reserved" after a reservation is made.

1. **Delete**

****

* This code performs DELETE operations to cancel a reservation by first deleting the associated payment record, then updating the room status to "Vacant", and finally deleting the reservation record itself from the database.