



Team Caribou

Ty Bergman
Phoebe Evans
Lauren Glaser
Vince Gnodle
Zoe Martinez

Project Overview



This project will implement a Hotel Room Registration System.

The problem:

Currently, our hotel creates room reservations by phone or in-person only. Other hotels in our nearby area have the option to reserve rooms online. We believe the ease of reserving online, rather than by phone or in-person, has led to our hotel losing business.

To ensure the success of our hotel, we will develop a hotel room registration system which will allow customers and hotel staff to reserve and manage rooms online.



Key Architectural Drivers

Essential system requirements:

- The system must have a dynamic user interface, that visually updates the rooms as they become available/pending/unavailable.
- Multiple customers can reserve different rooms at the same time.
- The system gives customers the ability to view their account information and view past/current room reservations.
- The system gives admin the ability to update room reservations, view past/upcoming reservations, and view reports.

Architectural Style Choices



Option 1: Model View Controller (MVC)

Why?

1. The Model would save all records for the users, admin, and room reservations.
2. The View would create the user interface, handling the display of the hotel rooms and the data.
3. The Controller would support the communication between the Model and the View, ensuring the information on the screen is consistently accurate.

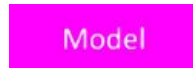
Option 2: Client-Server

Why?

1. The Database Server would store all data and information for the system.
2. The Application Server would contain the business logic for all system actions. For instance, reserving and managing the rooms.
3. The Client Presentation would create the user interface, handling the visual display of the hotel rooms and the data.

We chose the MVC architectural style.

The Architecture



Handles data storage.

Classes:
User
Admin
Rooms
Reservations

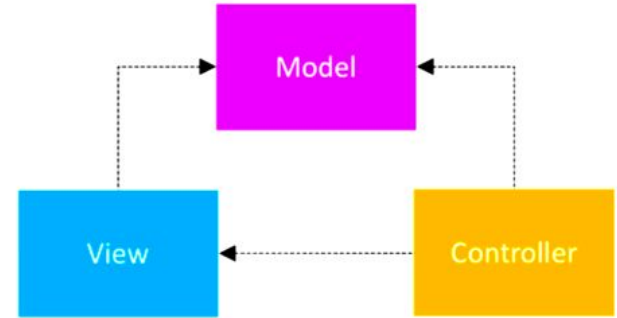


Handles the visual display.

Classes:
Log In Screen
Room Viewer
Reservation Process
Cart
View Bookings
User Profile
Admin Profile
Reports



Connects the Model and View Classes.



Folder Structure:

```
src
---main
    ---app
        ---Program.java
    ---features
        ---LogIn.java
        ---SearchRooms.java
        ---CheckRoomAvailability.java
        ---ReserveRoom.java
        ---CompleteTransaction.java
        ---UpdateUserAccount.java
        ---UpdateAdminAccount.java
    ---mvc
        ---Model.java
        ---View.java
        ---Controller.java
```



Conclusion

Our system will use MVC design architecture because of its organization between the data storage and visual components. Utilizing the standard practices with MVC architecture will help complete our online Hotel Room Reservation System while keeping our project systematic and efficient.

Would you agree with our choice to use MVC architecture, or would you advise to use Client Server architecture?

Do you foresee any gaps in our architecture?