
Design Document

for

Homestay Booking Website Project

Version 1.0

Prepared by

Group 6

Vu Nguyet Hang
Duong Quoc Khanh
Nguyen Thi Thu Trang
Nguyen Van Son
Nguyen Truong Giang

22028079
22028090
22028254
22028020
19021259

22028079@vnu.edu.vn
22028090@vnu.edu.vn
22028254@vnu.edu.vn
22028020@vnu.edu.vn
19021259@vnu.edu.vn

Instructor: Assoc. Prof. Dr. Dang Duc Hanh

Course: INT2208E_23

Teaching Assistant: Kieu Van Tuyen

Date: 04/05/2024

CONTENTS

TABLE OF FIGURES	3
1. Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	4
1.4 References	4
2. Architectural Design	5
2.1 Overview	5
2.2 Architectural Details	5
2.3 Use Case Diagrams	7
3. User Interface Design	17
3.1 Login	17
3.2 Register	17
3.3 Landing Page	18
3.4 Search Page	18
3.5 View Bookings	19
3.6 Booking Confirmation	19
3.7 Hotel Detail Page	20
3.8 User Profile	21
3.9 Change Password	21
3.10 View Host's Rooms	22
3.11 View Guests' Bookings	22
3.12 Add Room	23
4. Data Structures and Algorithms	23
5. External Interfaces	24

REVISIONS

Version	Primary Author(s)	Description of Version	Date Completed
#1	Group 6	Initial Design Document	04/05/24
#2	Group 6	Change Sequence & Class Diagram Details Add Database Schema Diagram	10/05/24

TABLE OF FIGURES

Figure 1. Architecture overview	5
Figure 2. Architectural model	6
Figure 3. Login class diagram	8
Figure 4. Login sequence diagram	8
Figure 5. Register class diagram	9
Figure 6. Register sequence diagram	9
Figure 7. Edit profile class diagram	10
Figure 8. Edit profile sequence diagram	10
Figure 9. Search room class diagram	11
Figure 10. Search room sequence diagram	11
Figure 11. Book room class diagram	12
Figure 12. Book room sequence diagram	13
Figure 13. Manage room class diagram	14
Figure 14. View my room sequence diagram	14
Figure 15. Add room sequence diagram	15
Figure 16. Edit room sequence diagram	15
Figure 17. Delete room sequence diagram	16
Figure 18. Login with an existing account	17
Figure 19. Create an account	17
Figure 20. Landing Page	18
Figure 21. Search results	18
Figure 22. Booking confirmation page	19
Figure 23. View current account's bookings	19
Figure 24. View homestay detail	20
Figure 25. View personal details	21
Figure 26. Change password	21
Figure 27. View list of host's rooms	22
Figure 28. View guests' bookings	22
Figure 29. Add room page	23
Figure 30. Database schema diagram	24

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed architecture design of the Homestay Booking System by focusing on four key quality attributes: usability, availability, maintainability, and testability. This document will help the development team to determine how the system will be structured at the highest level. It is also intended for the project manager to sign off on the high-level structure before the team shifts into detailed design, and to validate that the development team is meeting the agreed-upon requirements during evaluation process.

1.2 Scope

This document will address the architecturally significant functional requirements as well as a prototype of the user interface design. Additionally, a class diagram and sequence diagram are provided for each use case. By addressing these aspects comprehensively, this document aims to provide stakeholders, including developers, designers, and project managers, with a clear understanding of the homestay booking website's design and functionality. It serves as a reference point throughout the development lifecycle, guiding decision-making and facilitating effective communication among project team members.

1.3 Definitions, Acronyms and Abbreviations

User refers to any individual who interacts with the homestay booking website.

Guest a user of the homestay booking website who seeks temporary accommodation in various locations.

Host a user who offers accommodation on the homestay booking website.

UI User Interface

1.4 References

Application Architecture Guide 2.0 patterns and practices. © 2008 Microsoft Corporation.

Retrieved from: http://fizyka.umk.pl/~jacek/docs/net/Application_Architecture_Guide_v2.pdf

Textbook: Software Engineering PEARSON Tenth Edition – Ian Sommerville.

Architecture model, use case diagram templates provided by teaching assistant.

2. Architectural Design

2.1 Overview

This figure shows a high-level overview of the system's architecture. Further details on the system components and their interactions will be explained in detail in the following sections.

Overall, the sections of the application design can be thought of as four basic sets of services:

- **Presentation services:** These are the user-oriented services responsible for managing user interaction with the system, and generally consist of components located within the presentation layer. They provide a common bridge into the core business logic encapsulated in the business services.
- **Application services:** These services handle the presentation layer requests, transformation of disparate data for presentation and serve as the controller component.
- **Business services:** These services implement the core functionality of the system and encapsulate the relevant business logic. They generally consist of components located within the business layer, which may expose service interfaces that other callers can use.
- **Data services:** These services provide access to data that is hosted within the boundaries of the system, and data exposed by other back-end systems; perhaps accessed through services. The data layer exposes data to the business layer through generic interfaces designed to be convenient for use by business services. This layer implements the communication with the data source (in this case, a database).

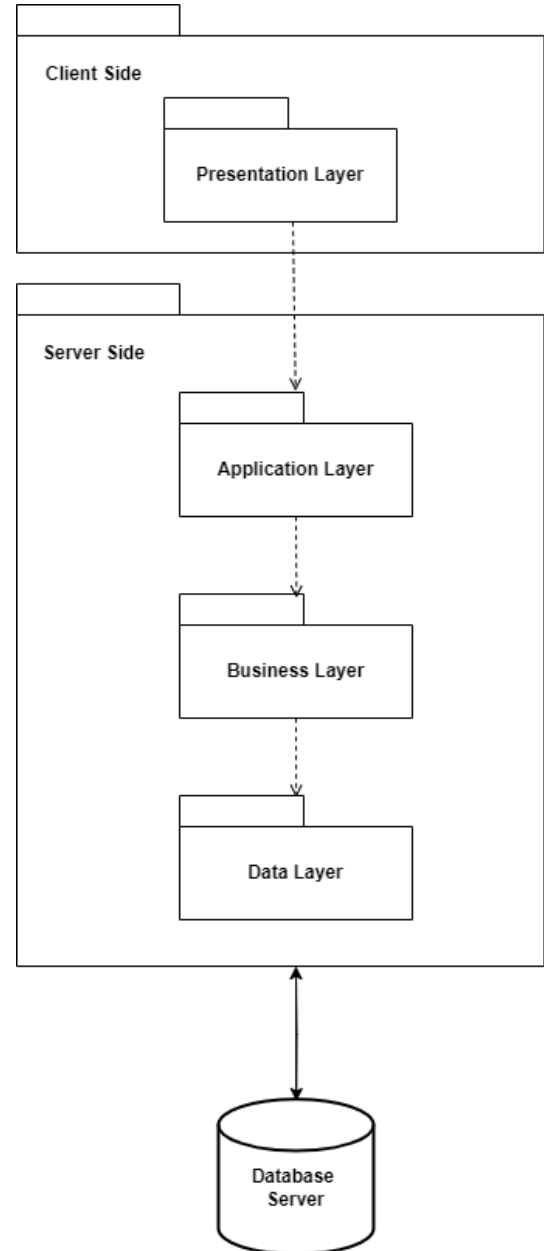


Figure 1. Architecture overview

2.2 Architectural Details

2.2.1 Presentation Layer Components

Presentation layer components implement the functionality required to allow users to interact with the application. In our Homestay Booking System, these components are separated into two types:

- **User interface (UI) components.** These components provide the mechanism for users to interact with the application. They format data and render it for display, acquire and validate data entered by user. Specifically, they are React components such as header, search bar, pop-up messages, etc.

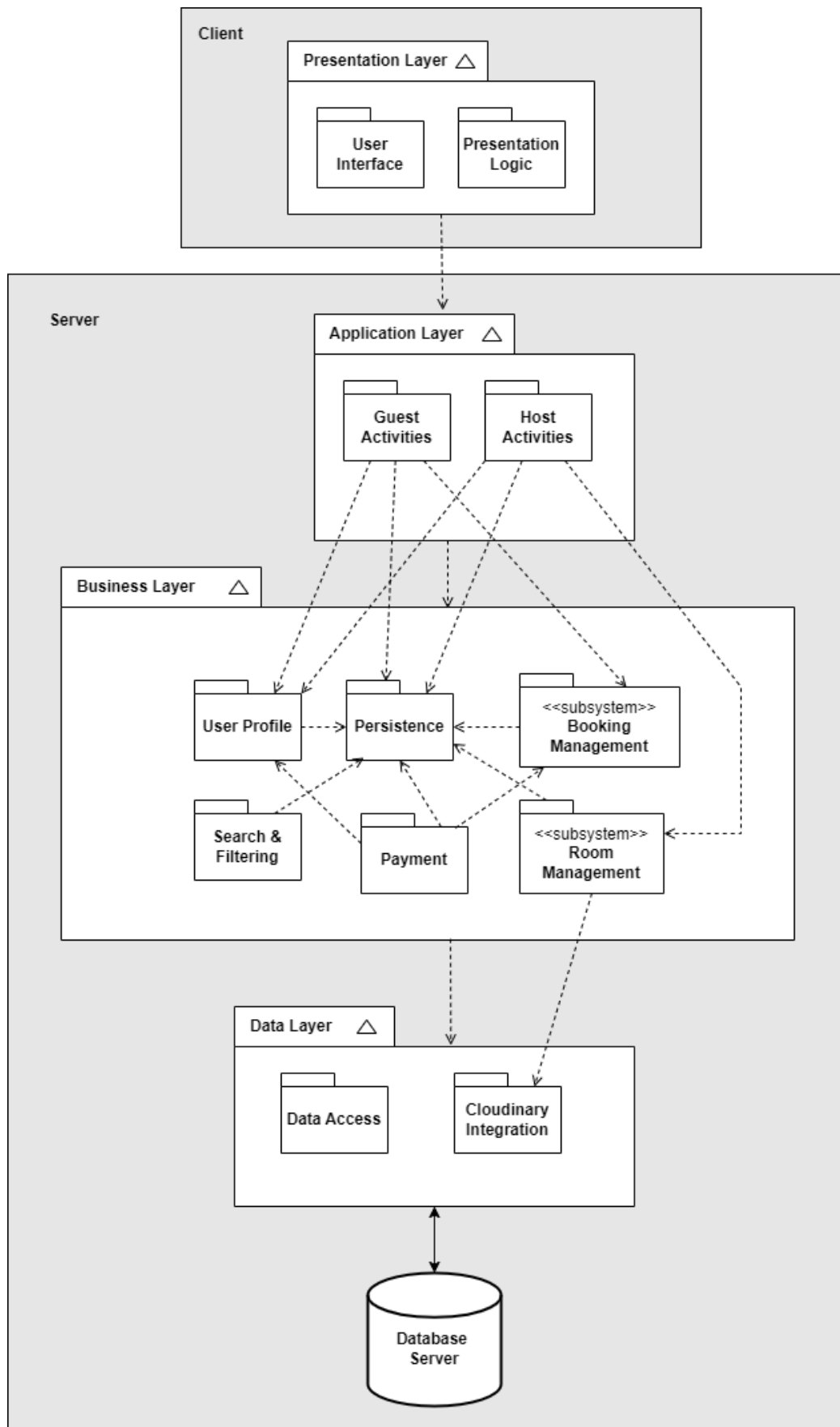


Figure 2. Architectural model

- **Presentation Logic components.** They are responsible for managing the user interface and controlling how data is presented to the user. These components handle tasks such as managing the state of UI elements and responding to user interactions. Within the homestay booking system, we also leverage caching mechanisms to optimize room lookups and avoid network round trips.

2.2.2 Application Layer Components

By placing guest and host activities in this layer, we separate the concerns of managing user interactions and business rules from other layers as they represent different sets of functionalities available to users acting as guests and hosts within the Homestay Booking System.

- Both guest and host activities rely on **User Profile** in the Business Layer for managing user information, preferences, and authentication. They have common user management functionalities, such as registration, login, profile editing.
- Both guest and host activities depend on **Persistence** component in the Business Layer for storing and retrieving data from the database.
- Guest activities rely on the **Booking Management** subsystem for functionalities such as searching for available rooms, making reservations, and managing bookings.
- Host activities depend on the **Room Management** subsystem for functionalities such as listing properties, managing room availability, editing room, viewing guests' bookings.

2.2.3 Business Layer Components

- The **Payment** component handles payment processing for homestay booking (in our project, this is done through Stripe). It interacts with both **User Profile** component to retrieve user payment information and the **Booking Management** component to associate payments with specific bookings.
- The **Search & Filtering** component enables users to search for rooms based on criteria such as location, dates, number of guests.
- All components in this layer interacts with **Persistence** component to ensure that each component can access and manipulate data stored in the database efficiently.
- **Room Management** subsystem is responsible for managing room inventory, including descriptions, amenities, and images. **Integrating with Cloudinary** in the Data Layer allows system to store room images and media assets in the cloud, providing a centralized and scalable solution for managing and serving media content.

2.2.4 Data Layer Components

- The **Data Access** component is responsible for handling interactions with the database, providing a set of interfaces, methods, or classes that abstract the details of data storage and retrieval operations, allowing other components in higher layers to interact with the database without needing to know the underlying implementation details. It includes functionalities such as: connecting to the database, executing queries, inserting – updating – deleting data.
- **Cloudinary Integration**, as mentioned before, is a cloud-based media management platform that manages host's images, directly involves in the process of adding room which belongs to **Room Management** subsystem.

Figure 2 illustrates components of each layer and the dependencies between them.

2.3 Use Case Diagrams

2.3.1 Login

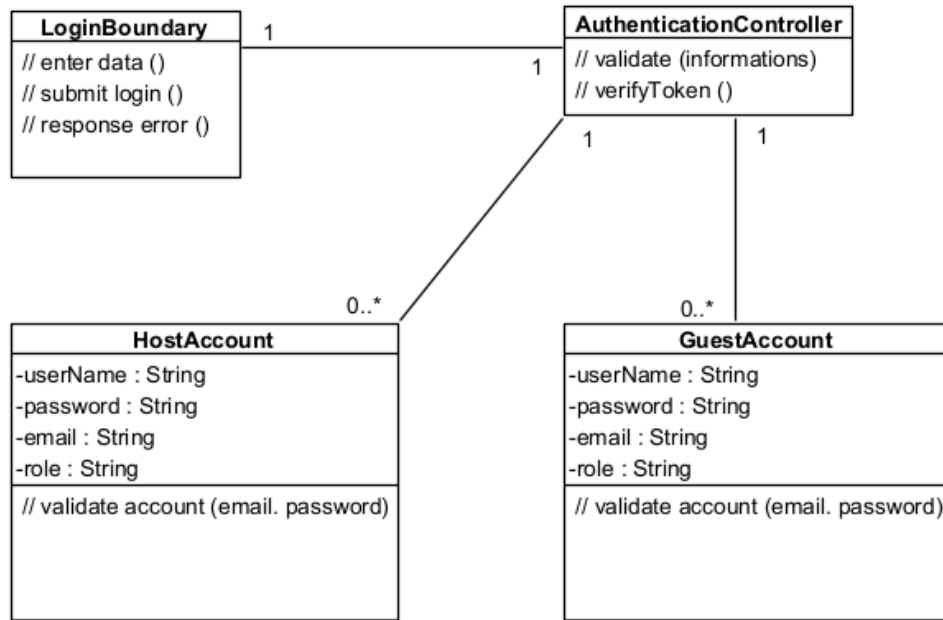


Figure 3. Login class diagram

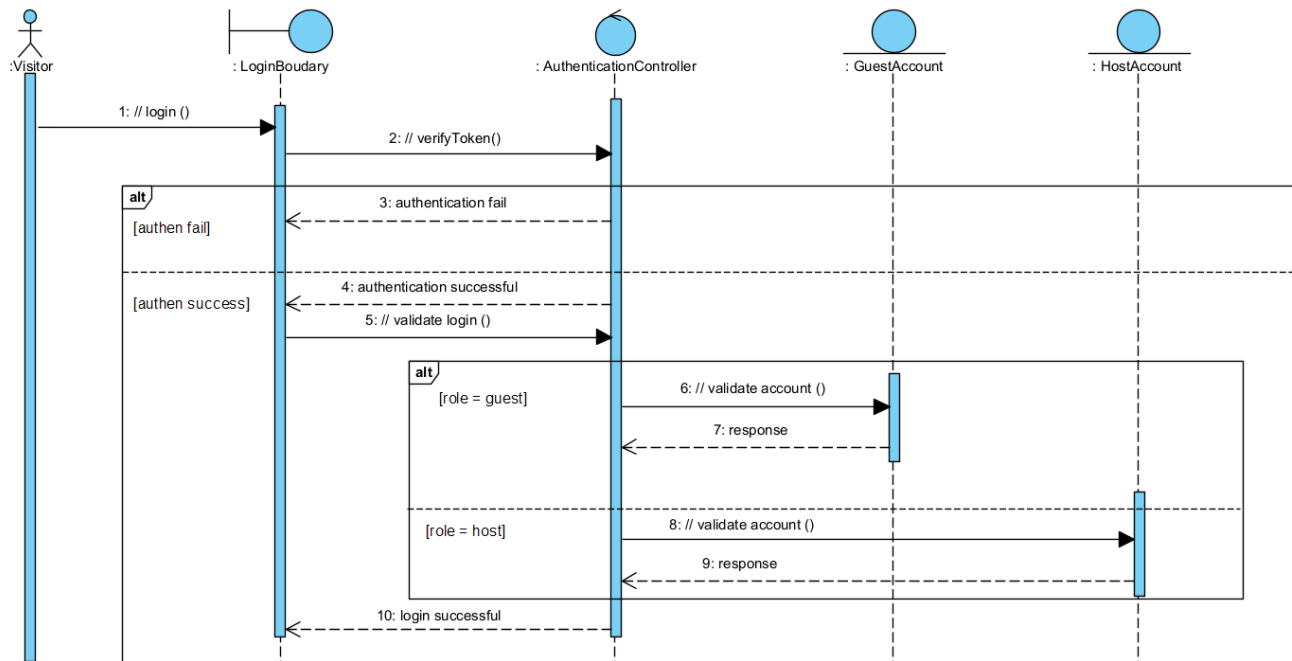


Figure 4. Login sequence diagram

2.3.2 Register

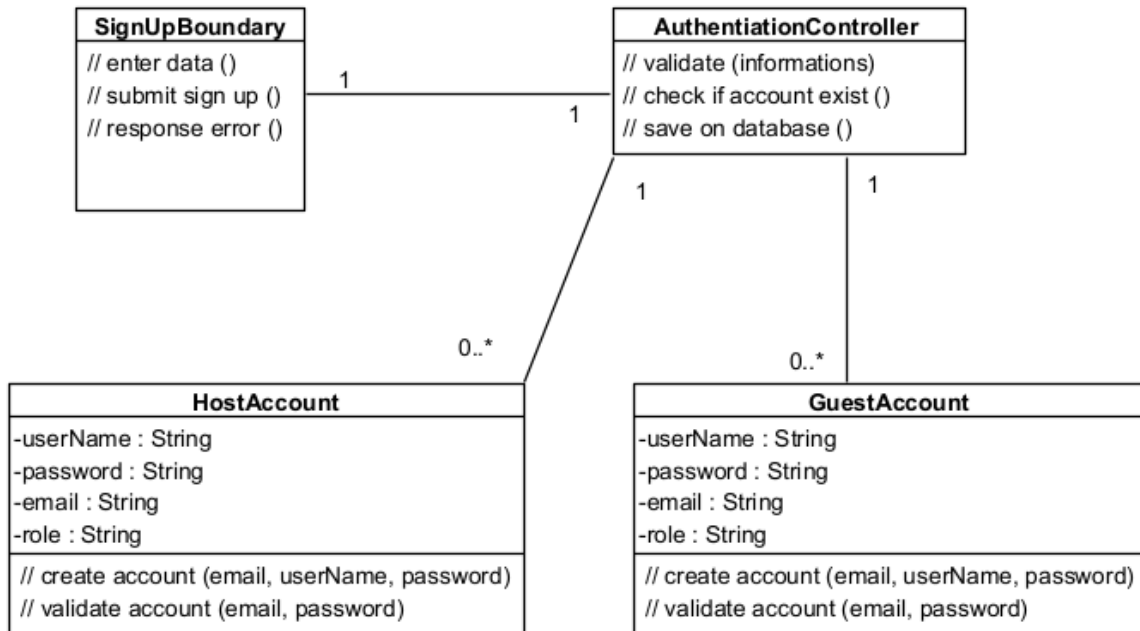


Figure 5. Register class diagram

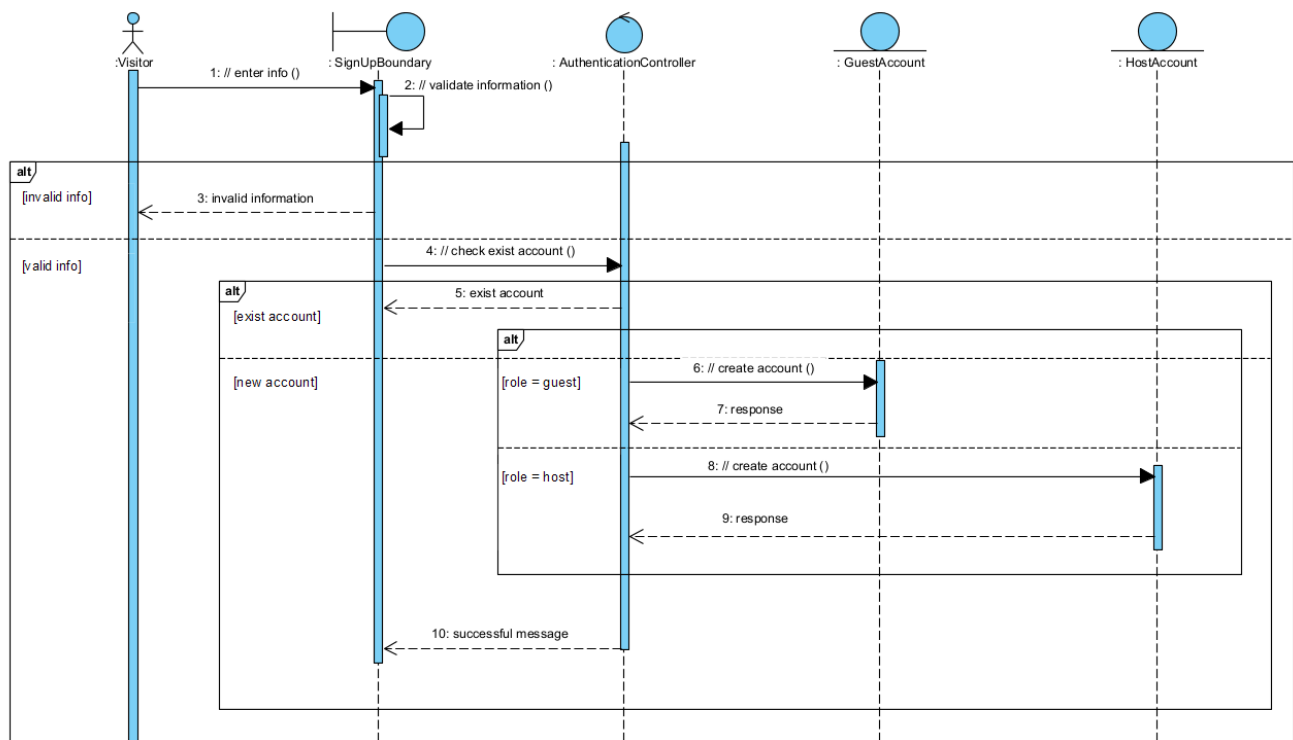


Figure 6. Register sequence diagram

2.3.3 Edit Profile

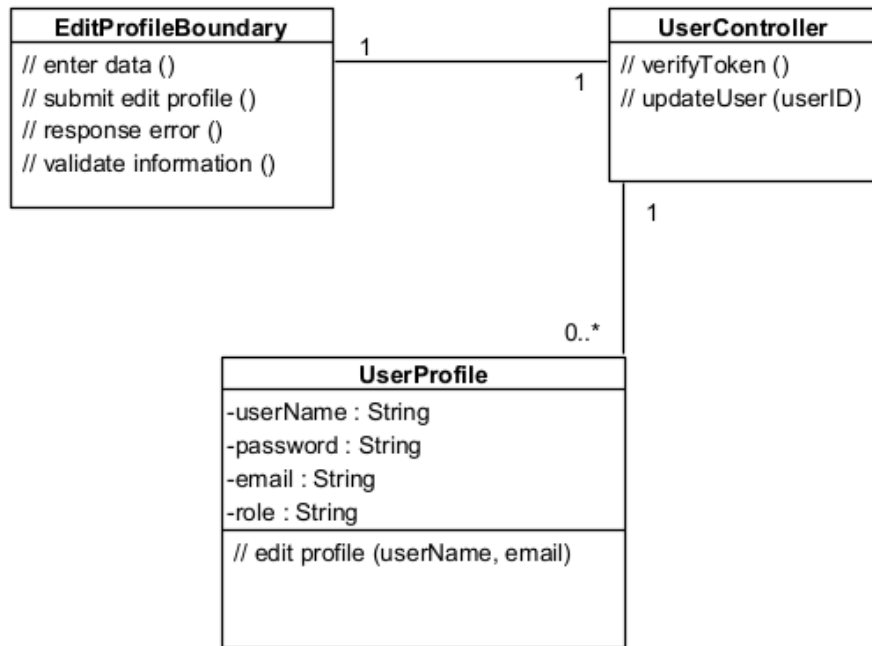


Figure 7. Edit profile class diagram

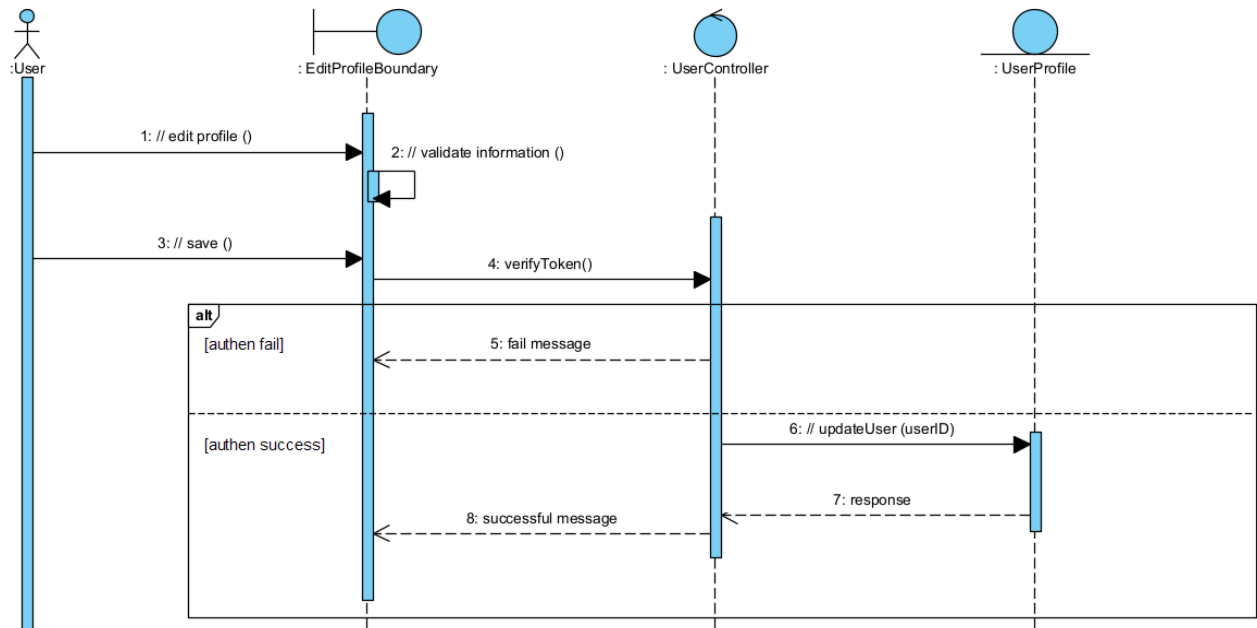


Figure 8. Edit profile sequence diagram

2.3.4 Search Room

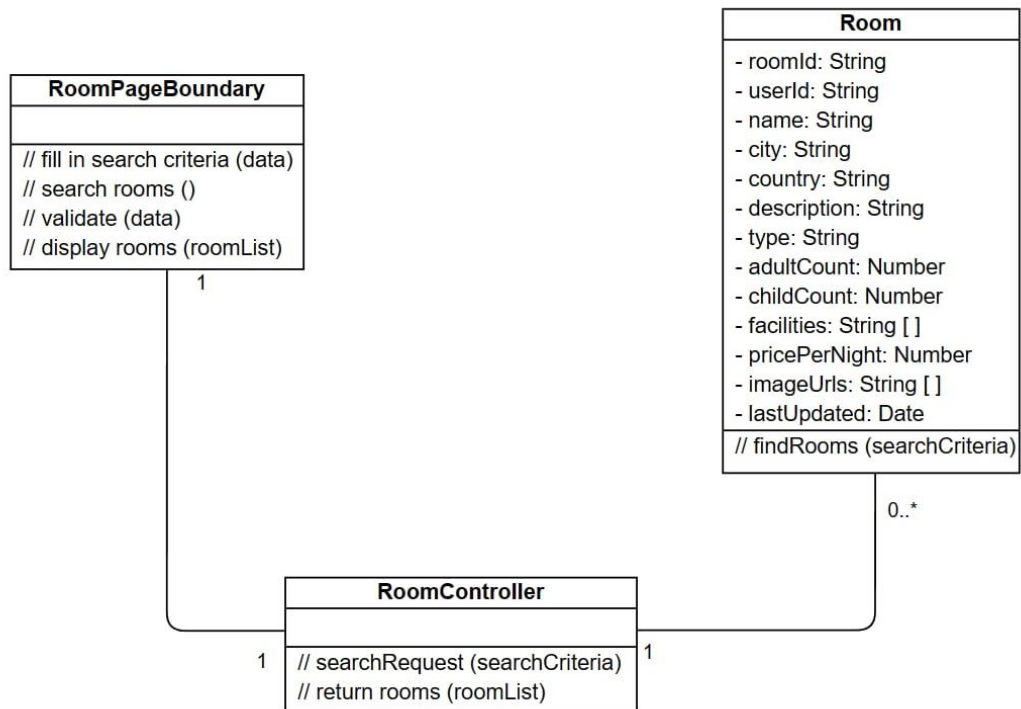


Figure 9. Search room class diagram

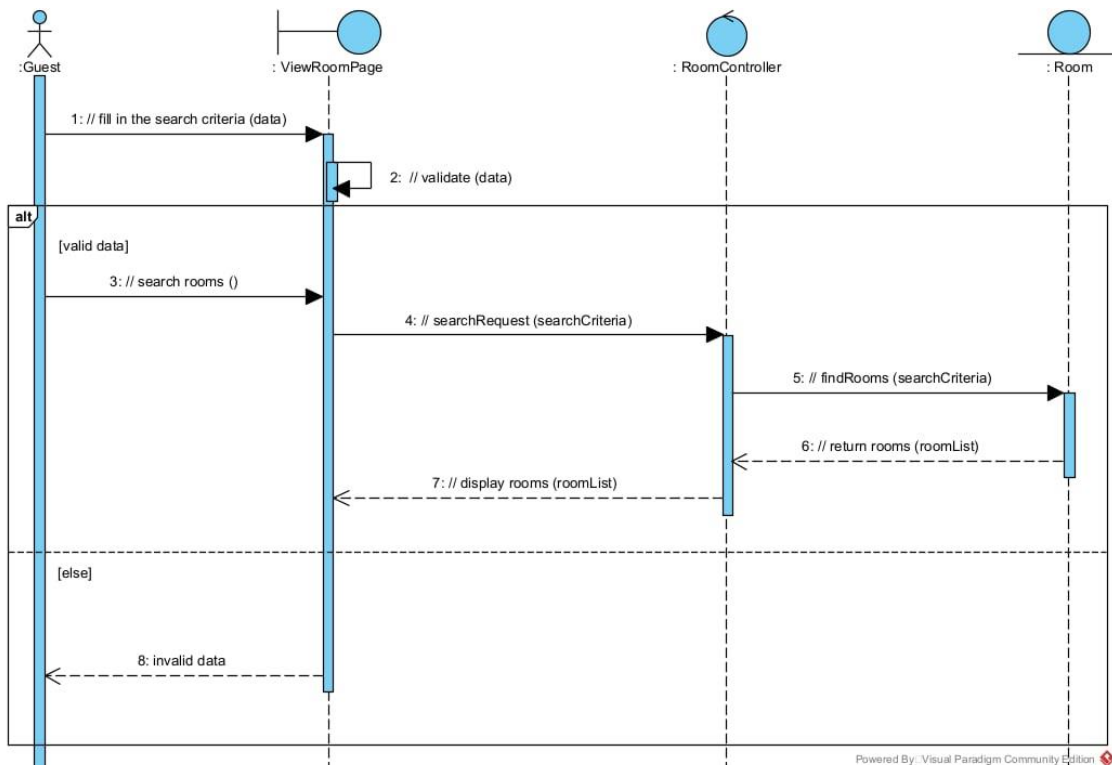


Figure 10. Search room sequence diagram

2.3.4 Book Room

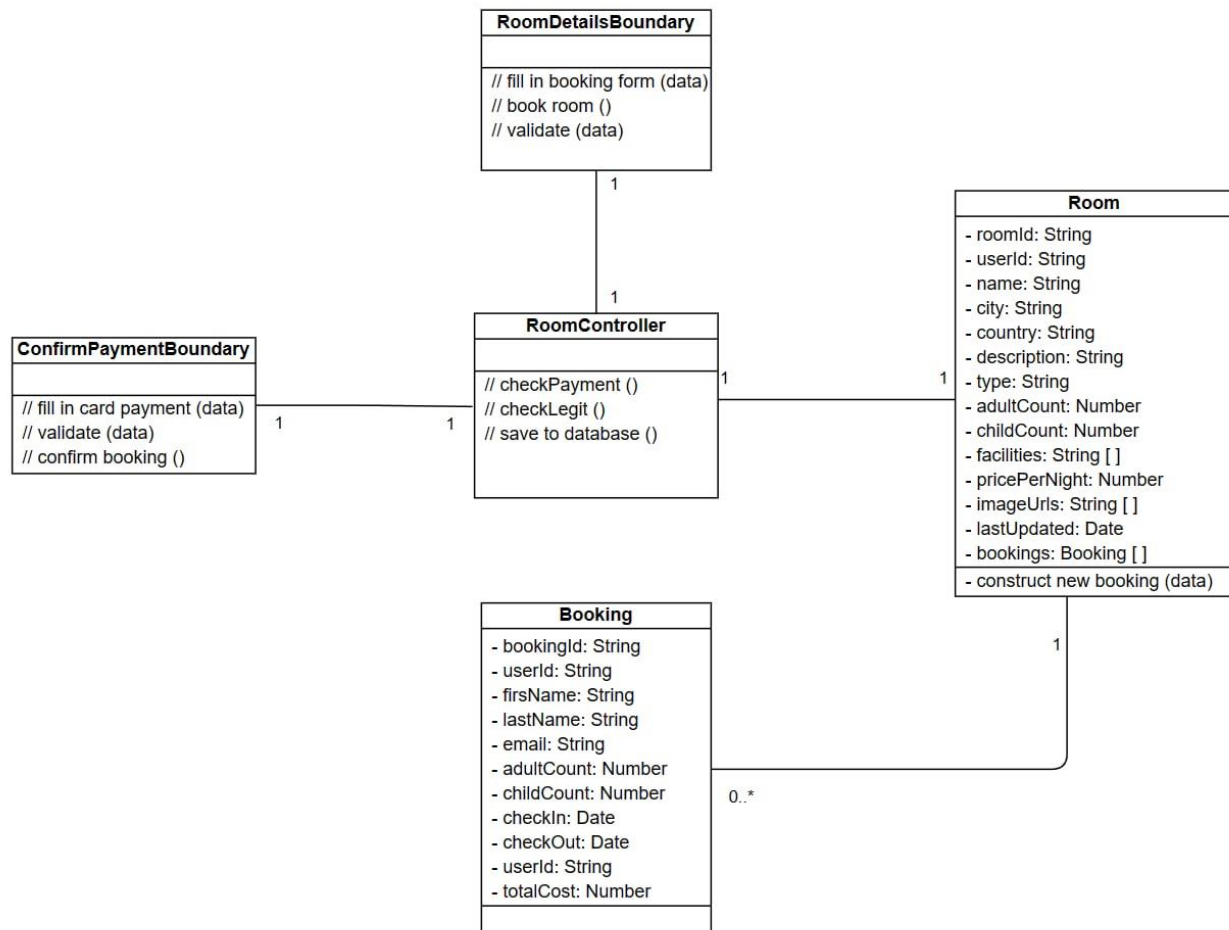


Figure 11. Book room class diagram

DESIGN DOCUMENT FOR

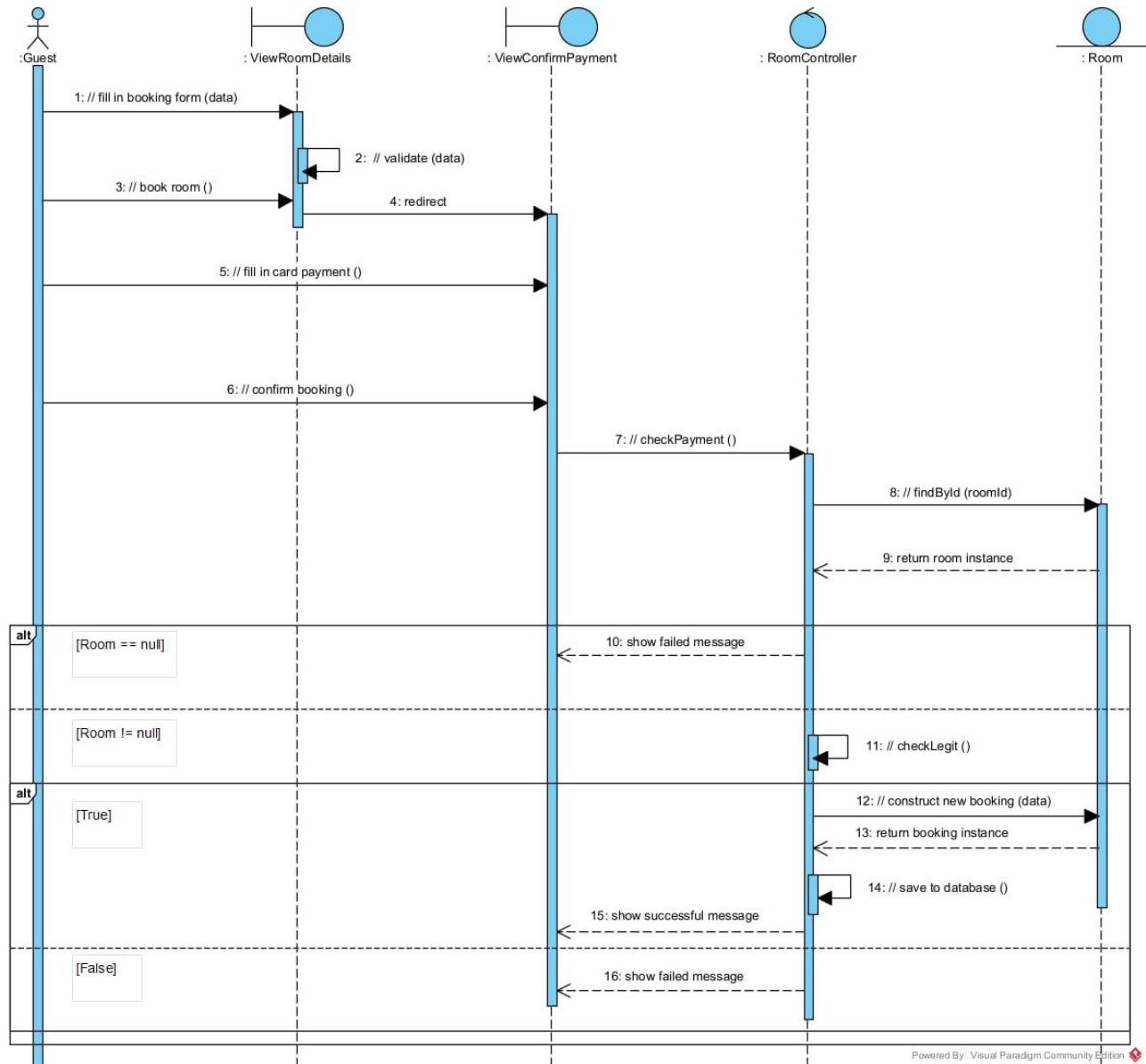


Figure 12. Book room sequence diagram

2.3.5 Manage Room

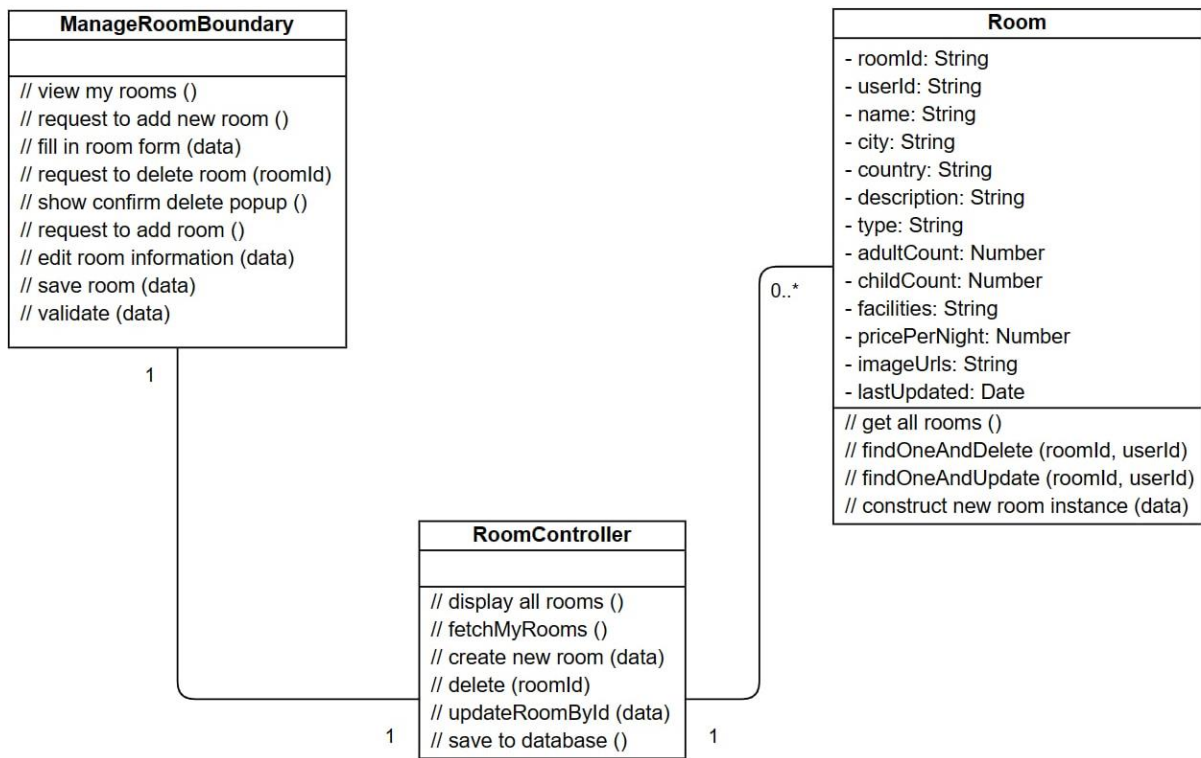


Figure 13. Manage room class diagram

2.3.5.1 View my room

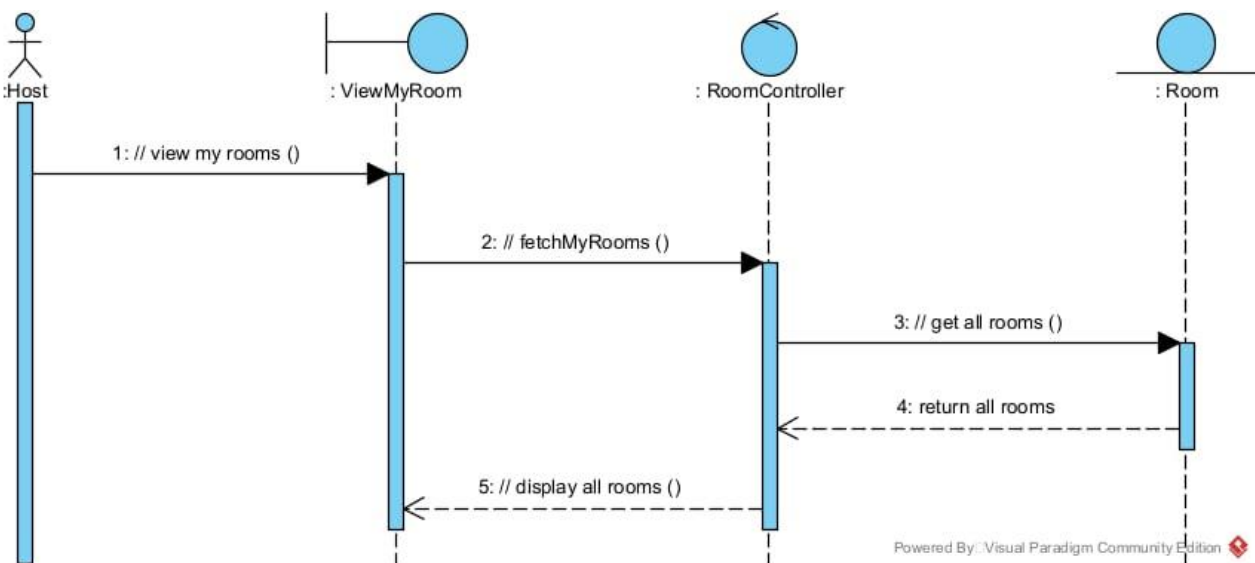


Figure 14. View my room sequence diagram

2.3.5.2 Add room

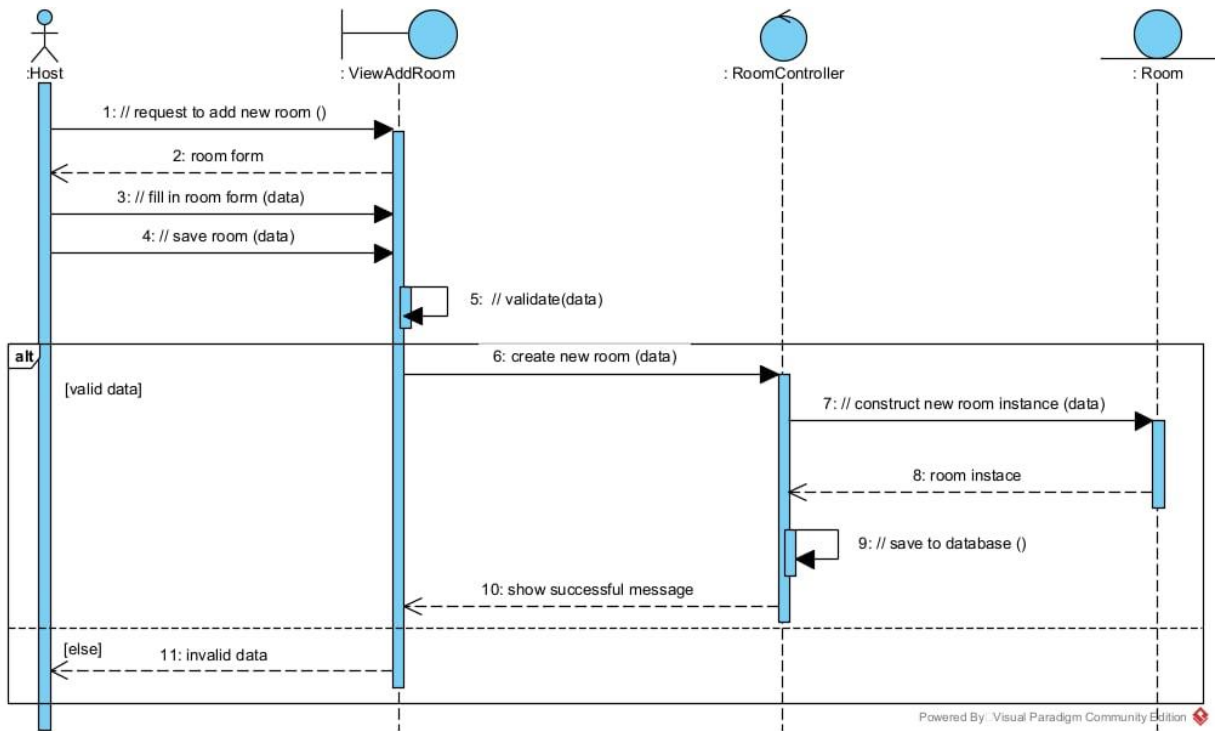


Figure 15. Add room sequence diagram

2.3.5.3 Edit room

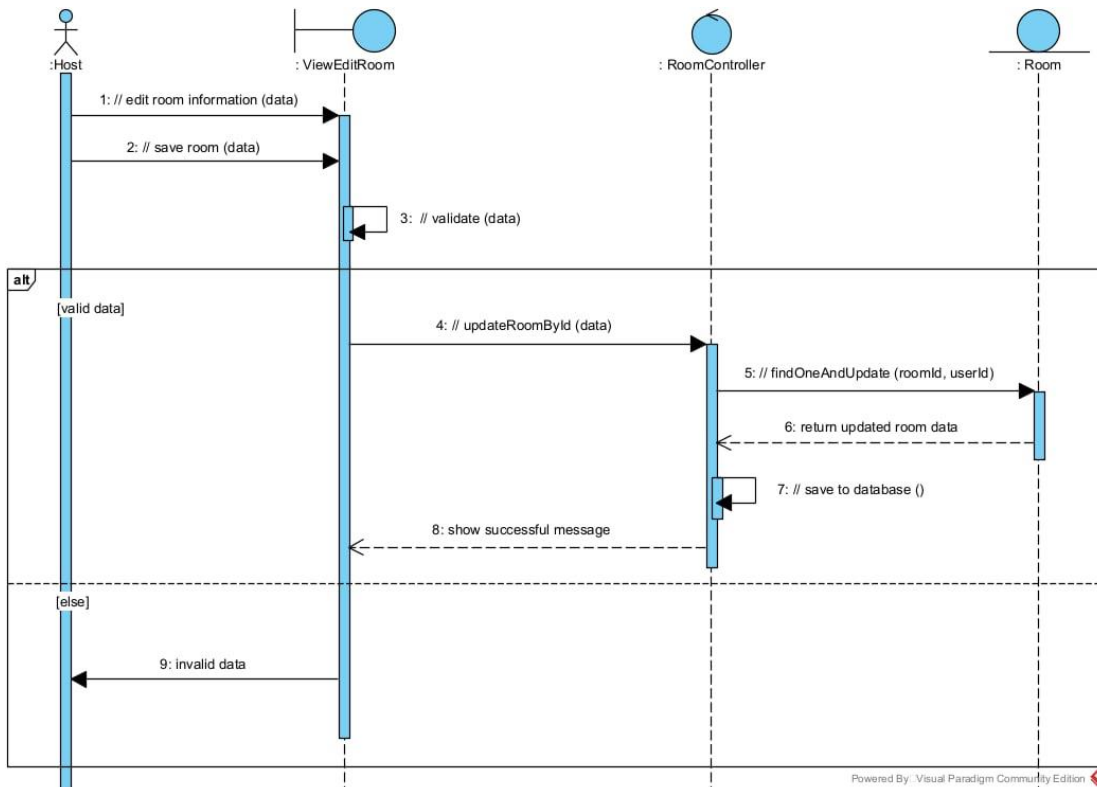


Figure 16. Edit room sequence diagram

2.3.5.4 Delete room

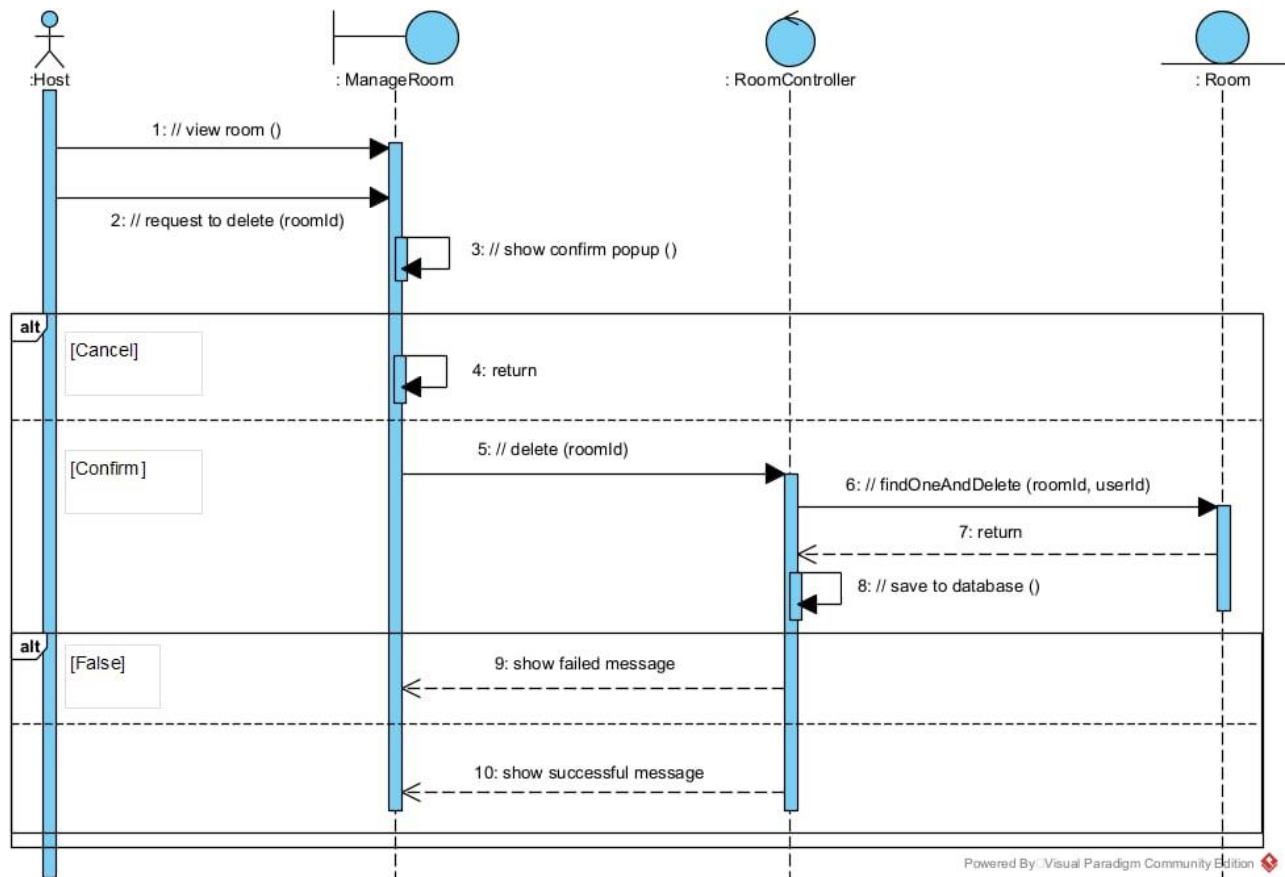
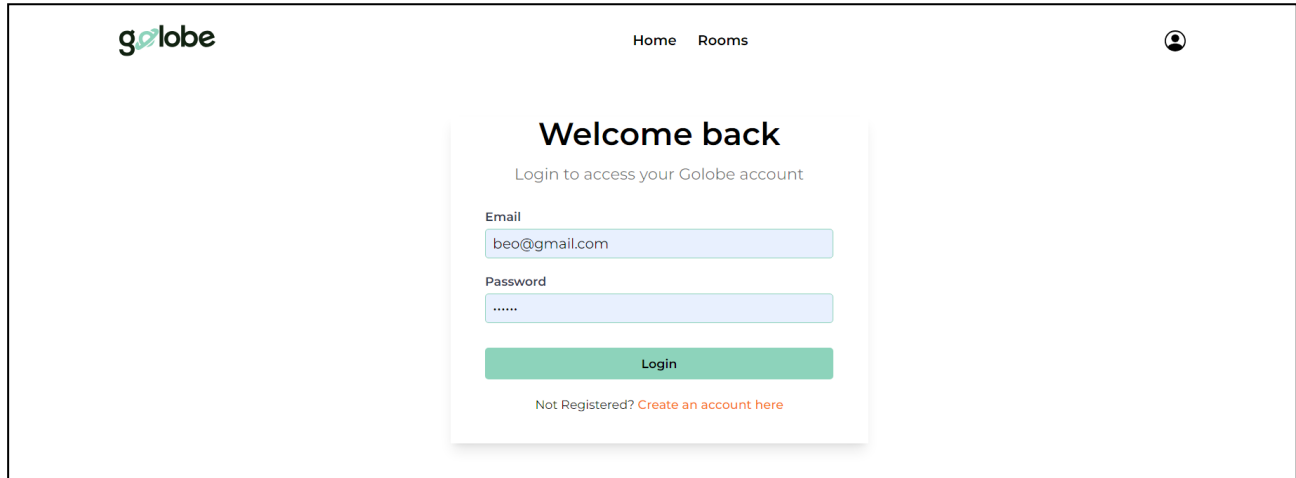


Figure 17. Delete room sequence diagram

3. User Interface Design

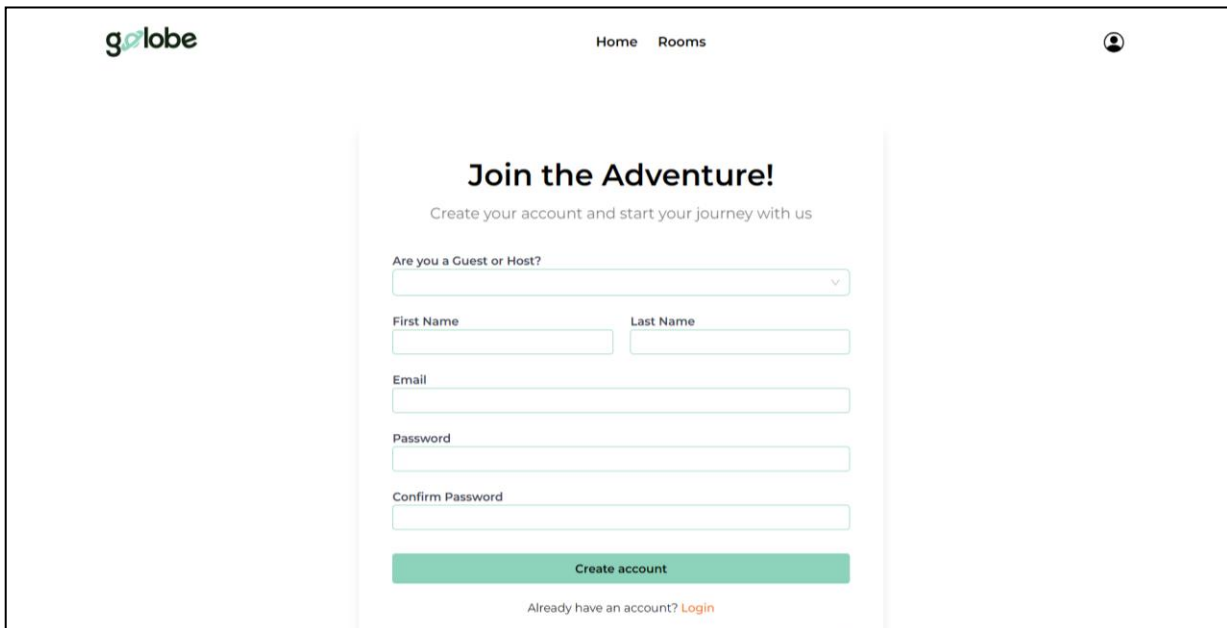
3.1 Login



The login form is centered on the page. It features the 'golobe' logo in the top left, 'Home' and 'Rooms' links in the top center, and a user icon in the top right. The main heading is 'Welcome back', followed by the subtext 'Login to access your Golobe account'. The form includes an 'Email' field with the placeholder 'beo@gmail.com', a 'Password' field with masked characters '.....', and a green 'Login' button. At the bottom, there is a link: 'Not Registered? [Create an account here](#)'.

Figure 18. Login with an existing account

3.2 Register



The registration form is centered on the page. It features the 'golobe' logo in the top left, 'Home' and 'Rooms' links in the top center, and a user icon in the top right. The main heading is 'Join the Adventure!', followed by the subtext 'Create your account and start your journey with us'. The form includes a dropdown menu for 'Are you a Guest or Host?', 'First Name' and 'Last Name' fields, an 'Email' field, a 'Password' field, and a 'Confirm Password' field. A green 'Create account' button is at the bottom. At the very bottom, there is a link: 'Already have an account? [Login](#)'.

Figure 19. Create an account

3.3 Landing Page

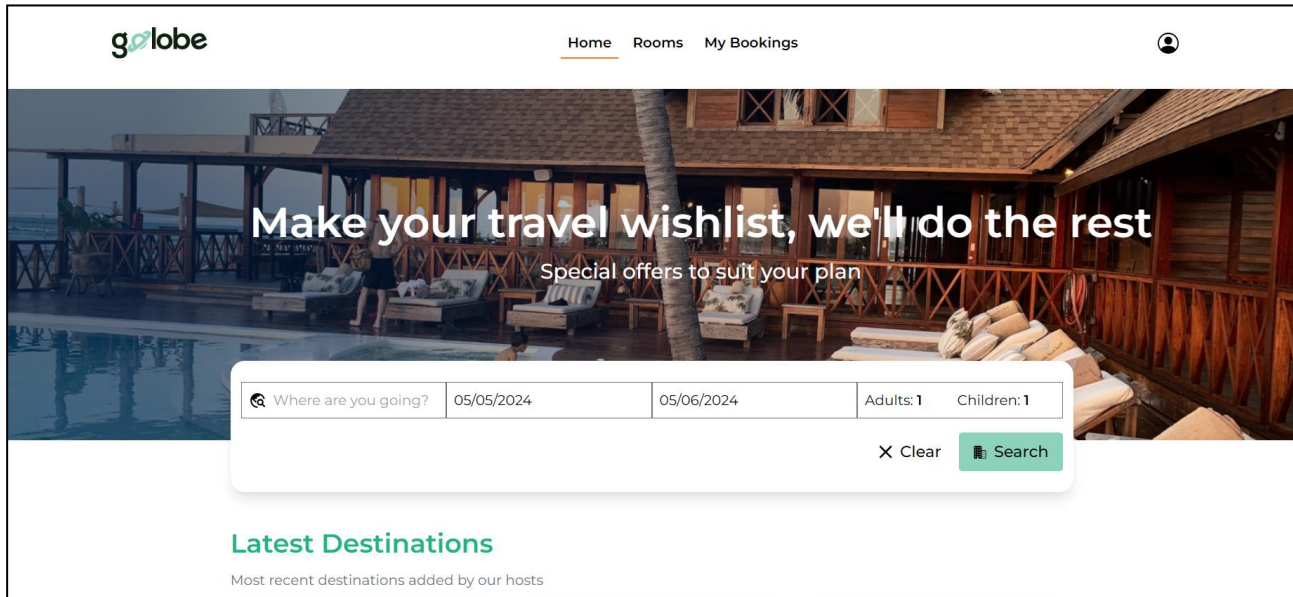


Figure 20. Landing Page

3.4 Search Page

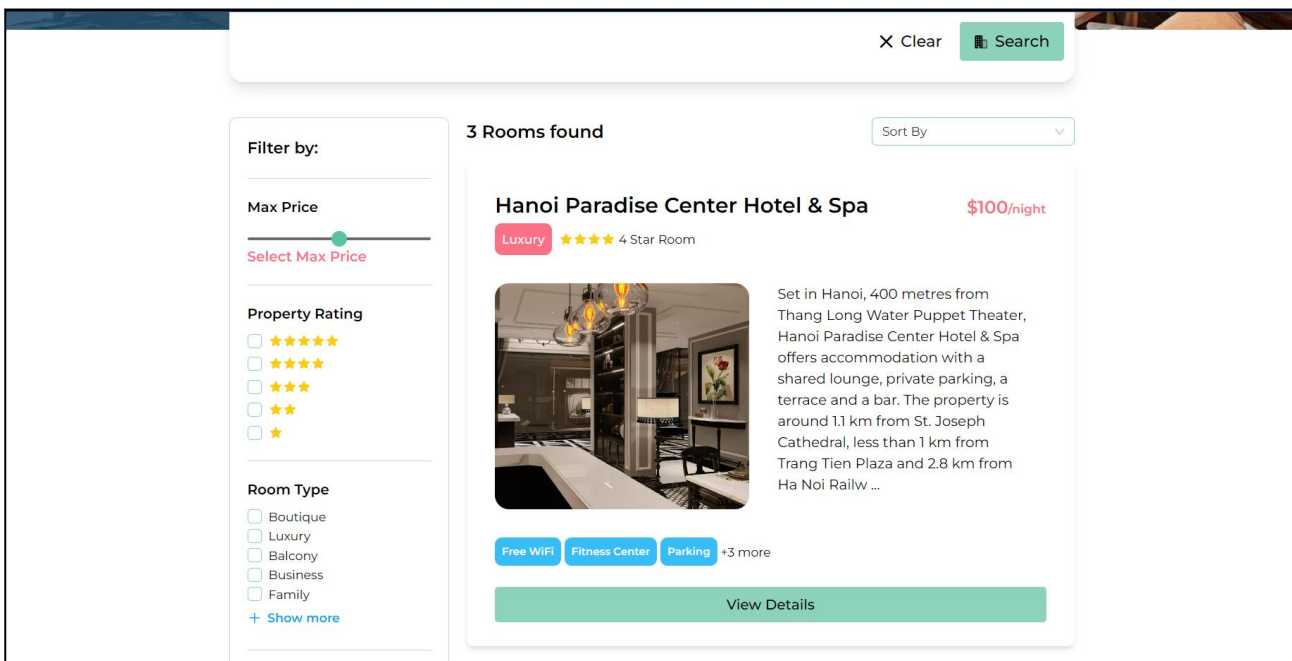


Figure 21. Search results

3.5 View Bookings

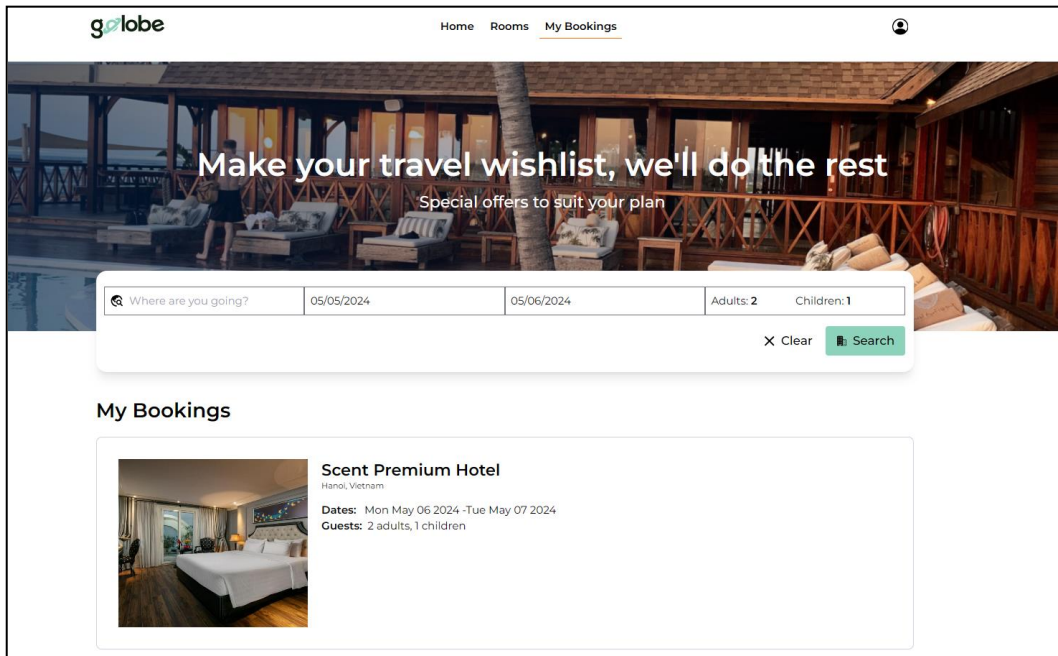


Figure 23. View current account's bookings

3.6 Booking Confirmation

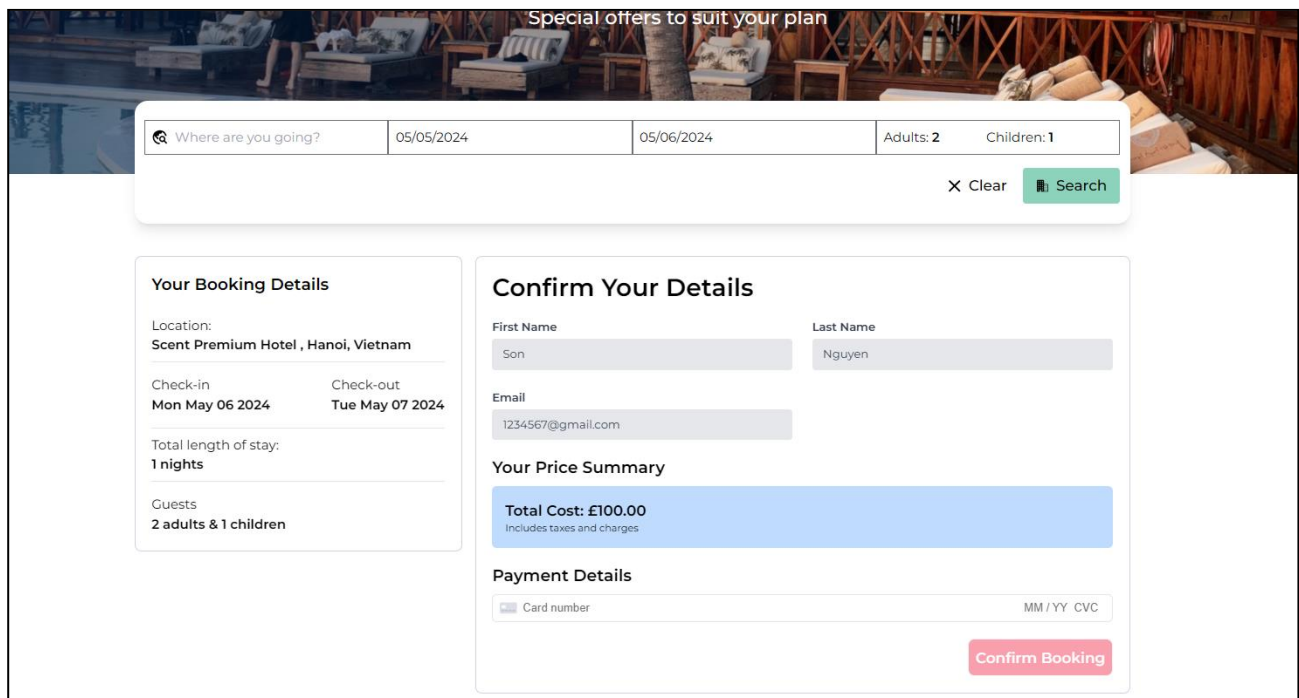


Figure 22. Booking confirmation page

3.7 Hotel Detail Page





Vietnam > Hanoi > Hanoi Paradise Center Hotel & Spa

Luxury

★★★★★ 4 Star Room

\$100/night

Hanoi, Vietnam

Overview

Set in Hanoi, 400 metres from Thang Long Water Puppet Theater, Hanoi Paradise Center Hotel & Spa offers accommodation with a shared lounge, private parking, a terrace and a bar. The property is around 1.1 km from St. Joseph Cathedral, less than 1 km from Trang Tien Plaza and 2.8 km from Ha Noi Railway station. The accommodation features a 24-hour front desk, airport transfers, room service and free WiFi.

Facilities

- Free WiFi
- Fitness Center
- Parking
- Airport Shuttle
- Outdoor Pool
- Spa

05/05/2024

05/05/2024

Adults: 1 Children: 1

Book Now

Figure 24. View homestay detail

3.8 User Profile

The screenshot shows the 'golobe' web application interface. At the top, there is a navigation bar with the logo 'golobe' on the left, and links for 'Home', 'Rooms', and 'My Bookings' in the center. A user profile icon is on the right. Below the navigation bar is a large hero image of a resort pool area with the text 'Make your travel wishlist, we'll do the rest' and 'Special offers to suit your plan'. The main content area is divided into two sections. On the left, there is a sidebar with two links: 'Personal Details' (highlighted with a magnifying glass icon) and 'Change Password' (with a padlock icon). On the right, the 'Personal details' form is displayed. It has a title 'Personal details' and a subtitle 'Keep your details current to ensure seamless communication and services'. The form contains three input fields: 'First Name' with the value 'Son', 'Last Name' with the value 'Nguyen', and 'Email' with the value '1234567@gmail.com'. At the bottom of the form is a green 'Edit' button.

Figure 25. View personal details

3.9 Change Password

The screenshot shows the 'golobe' web application interface. At the top, there is a navigation bar with the logo 'golobe' on the left, and links for 'Home', 'Rooms', and 'My Bookings' in the center. A user profile icon is on the right. Below the navigation bar is a large hero image of a resort pool area with the text 'Make your travel wishlist, we'll do the rest' and 'Special offers to suit your plan'. The main content area is divided into two sections. On the left, there is a sidebar with two links: 'Personal Details' and 'Change Password' (highlighted with a magnifying glass icon). On the right, the 'Change Password' form is displayed. It has a title 'Change Password' and a subtitle 'Keep your password secure and up to date'. The form contains three input fields: 'Current Password', 'New Password', and 'Confirm Password'. At the bottom of the form is a green 'Save' button.

Figure 26. Change password

3.10 View Host's Rooms

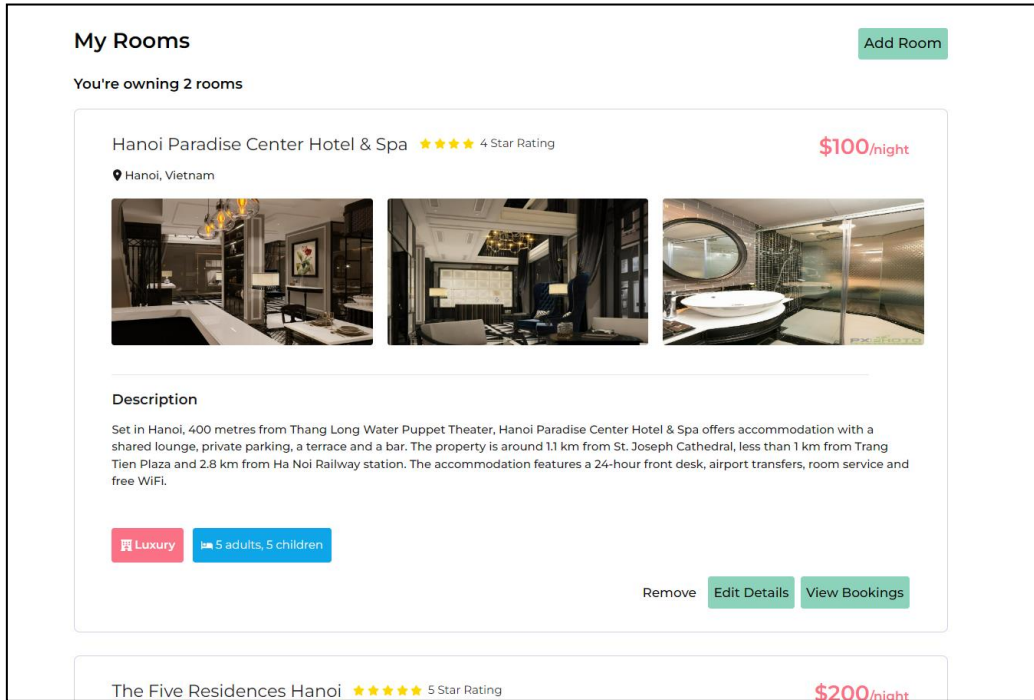


Figure 27. View list of host's rooms

3.11 View Guests' Bookings

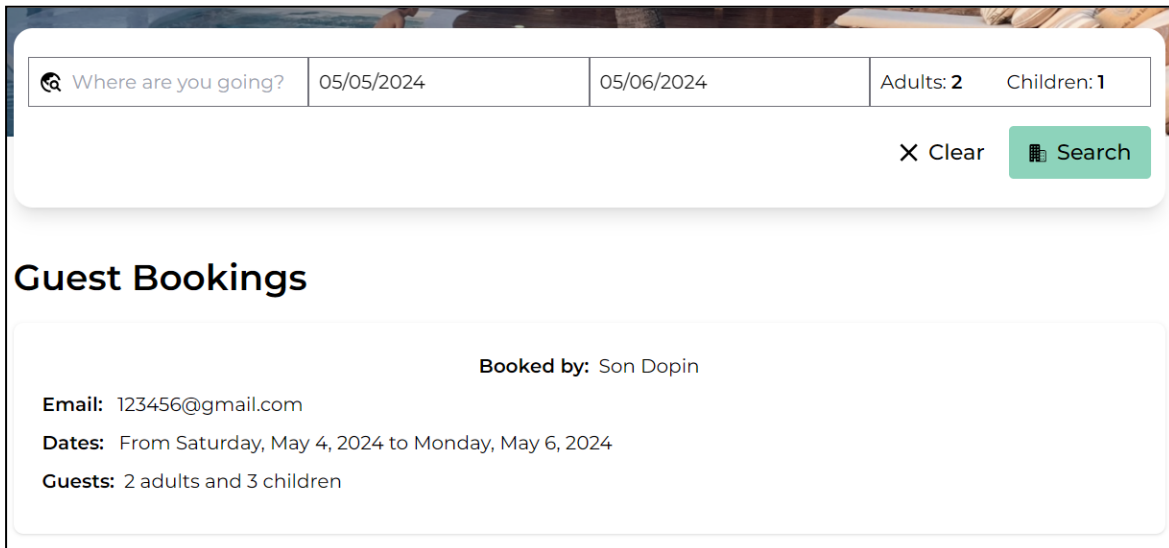


Figure 28. View guests' bookings

3.12 Add Room

Add Room

Name

City

Country

Description

Price Per Night

Star Rating

Type

Boutique

Luxury

Balcony

Business

Family

Romantic

Breakfast

Cabin

Non-Smoking

Bath

Motel

All Inclusive

Pet Friendly

Self Catering

Facilities

☐ Free WiFi

☐ Parking

☐ Airport Shuttle

☐ Outdoor Pool

☐ Spa

☐ Fitness Center

Guests

Adults

Children

Images

Choose Files

No file chosen

Save

Figure 29. Add room page

4. Data Structures and Algorithms

There are some notable data structures and alogirthms used in the project:

- Bcrypt hasing function: a cryptographic algorithm that hashes passwords in a way that is secure agianst brute-force attacks.
- sessionStorage: a web storage API that's used to store data on the client side for the duration of the session. In this project, sessionStorage is used to persist the state of destination, checkIn, checkOut variables across page reloads.

5. External Interfaces

a) GeoDB Cities API:

- **Description:** The GeoDB Cities API provides a comprehensive database of cities worldwide. It allows us to retrieve city-related data based on user input, such as city name or geographical coordinates.
- **Functionality:** The API supports various endpoints for querying cities by name, country code, or location. It returns detailed information about cities, including their name, country, population, coordinates, and more.
- **Usage in the system:** When a user inputs a location, we'll send a request to the GeoDB Cities API to fetch relevant city data. This data will then be used to populate the UI with available location options.

b) REST Countries API:

- **Description:** The REST Countries API provides information about countries worldwide. It offers data such as country name, capital, population, languages, currencies, and more.
- **Functionality:** This API allows us to retrieve country-related data based on the city selected by the user. We'll use it to obtain additional information about the country corresponding to the chosen city.
- **Usage in the system:** After the user selects a city, we'll send a request to the REST Countries API to obtain data about the corresponding country. This data can then be displayed alongside the selected city information in the UI, providing users with additional context.

6. Database Schema Diagram

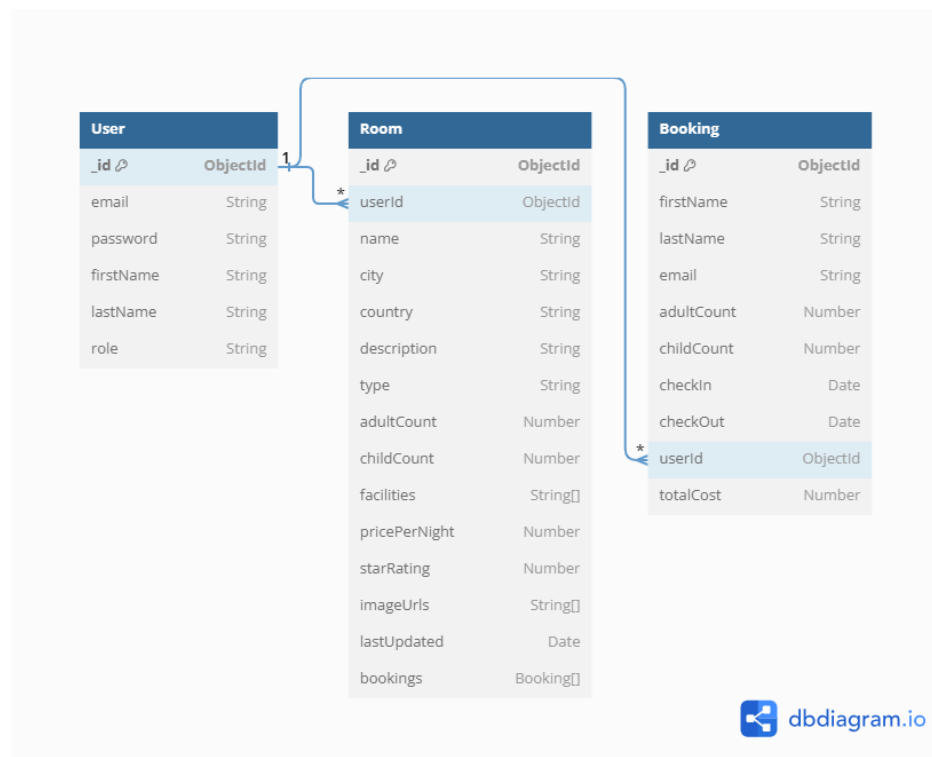


Figure 30. Database schema diagram

The above diagram represents the structure of three MongoDB collections: User, Hotel, and Booking. Each collection is like a table in a relational database, and each document in a collection is like a row in a table. The relationships between the collections are as follows:

- User to Hotel: One-to-Many. A user can own multiple hotels, but each hotel is owned by one user. This is represented by the `userId` field in the Hotel collection, which references the `_id` field in the User collection.
- User to Booking: One-to-Many. A user can have multiple bookings, but each booking is made by one user. This is represented by the `userId` field in the Booking document, which references the `_id` field in the User collection.
- Hotel to Booking: One-to-Many. A hotel can have multiple bookings, but each booking is made at one hotel. This is represented by the `bookings` array in the Hotel collection, which contains embedded Booking documents.