

***Bookify Hotel Reservation System with ASP.NET Core***

***Graduation Project***

***Prepared by***

*Ibrahim Abdullah Elsaid Ahmed Hamada Abdo*

*Kawther Hamdy Mohamed Mariam Tarek Ahmed*

**Supervised by**

*Eng. Mohamed Saad*

**1**



### Introduction

The hospitality and tourism industry is a vital sector that plays a significant role in the

economy, facilitating millions of travels and accommodations worldwide. During our market

research, we observed that while online hotel booking platforms are widely available, many still

lack a seamless, user-friendly experience, robust management capabilities for hotel staff, and the integration of modern design patterns that ensure scalability and maintainability.

The proposed **Bookify Hotel Reservation System** aims to address these gaps by providing a comprehensive, robust, and scalable web application for hotel bookings. This platform will offer significant benefits for both customers and hotel administrators. For instance, customers can effortlessly search for available rooms, view detailed information, and complete bookings with an integrated, secure payment system. Meanwhile, hotel staff gain powerful tools to manage room inventory, track bookings, and oversee operations efficiently through a dedicated admin panel.

Furthermore, the application will provide a trustworthy platform where users can make informed decisions based on transparent information, such as room details, and secured payment methods.

The core services provided by the **Bookify** application are:

1. **Room Browsing & Search Service:**  
   Allows users to view all available rooms and filter them based on criteria such as date, price, room type, and capacity.
2. **Booking & Reservation Service:**  
   Enables customers to select their desired rooms, choose dates, and securely complete their reservations.
3. **Secure Payment Integration:**  
   Facilitates a safe and reliable payment process during booking confirmation using the Stripe payment gateway, ensuring transaction security.
4. **Admin Management Service:**  
   Provides an powerful admin panel for hotel administrators to manage rooms, room types, view all bookings, and monitor the system efficiently.
5. **User Account & History Service:**  
   Allows registered users to view their personal profiles and access a complete history of their past and upcoming bookings.

### *Motivation*

In this section, we present the motivations behind developing the Bookify Hotel Reservation

System. Through our personal and family experiences with hotel bookings, we observed that most difficulties encountered were related to one of the following issues: either the inability to find available rooms during desired dates, or the challenge of assessing hotel and room quality before making a reservation.

As a first step in our design process, we created an online survey using Google Forms that

focused on the importance of providing a comprehensive hotel booking system. This survey was distributed to approximately 100 people. It included multiple-choice questions about how

frequently they faced similar challenges and gathered opinions about additional services such as implementing a comprehensive hotel and room rating system.

**The survey results are as follows:**

* **80%** of participants reported problems finding available rooms during peak seasons as shown in Figure 1.
* **63%** found it difficult to assess hotel and room quality before making reservations as shown in Figure 2.
* **85%** agreed on the importance of having a comprehensive rating system showing others' experiences as shown in Figure 3.
* **73%** of respondents strongly agreed on developing a comprehensive web application for hotel booking services as shown in Figure 4.

[Figure 1: Participants who faced problems finding available rooms]

[Figure 2: Participants who faced difficulty assessing hotel quality before booking]

[Figure 3: Participants who supported having a comprehensive rating system]

[Figure 4: Participants who supported developing a comprehensive web application]

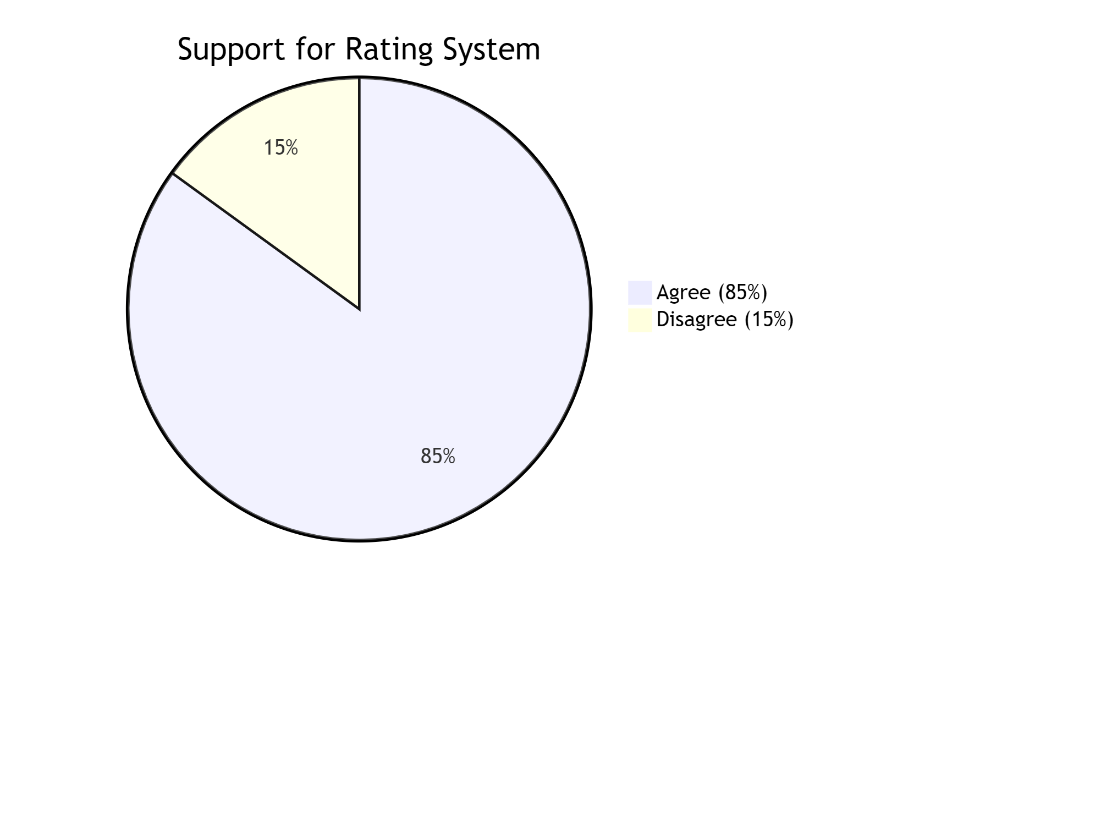
A pie chart with a number of percentages

AI-generated content may be incorrect. A pie chart with a number of percentages

AI-generated content may be incorrect.

**Figure 1: Participants who faced problems finding available rooms**

**Figure 2: Participants who faced difficulty assessing hotel quality before booking**

 A pie chart with percentages

AI-generated content may be incorrect.

**Figure 3: Participants who supported having a comprehensive rating systemFigure 4: Participants who supported developing a comprehensive web application**

Based on the survey results, we found that travelers need a system that enables them to:

* Search for available rooms easily and quickly
* Know hotel quality and ratings before booking
* Experience a smooth and secure booking process
* Have an integrated and reliable payment system

These results confirmed the urgent need to develop the Bookify system, which will contribute to solving these challenges and generally improve the hotel booking experience.

### Project Objectives

### This section presents the main objectives of the Bookify Hotel Reservation System.

### These objectives are:

### Streamline the Hotel Booking Process Simplify and digitalize the entire reservation workflow, making it effortless for customers to find and book hotel rooms while providing hotel staff with efficient management tools.

### Ensure Transparency and Build Trust Provide comprehensive hotel and room information, genuine customer reviews, and clear pricing to help users make informed decisions and build confidence in their booking choices.

### Enhance User Experience with Smart Features Implement intelligent search filters, real-time availability checks, and personalized recommendations to help travelers find ideal accommodations quickly and avoid booking frustrations.

### Create a Reliable Platform for Both Travelers and Hotels Develop a secure, scalable system that offers guests easy access to quality accommodations while providing hotels with effective occupancy management and revenue generation opportunities.

### 

### Project Scope

Our application will be developed initially as a web application using [ASP.NET](https://asp.net/) Core MVC. It's primarily

focused on the Egyptian hotel market, connecting travelers with local hotels and accommodations. Nevertheless,

the overall architecture and system design have been made as generic and scalable as possible, so that the

application can be expanded in the future to include mobile applications and other regions.

We chose a web application approach for several strategic reasons:

1. **Cross-Platform Compatibility**: Web applications can be accessed from any device with a browser,including Windows, macOS, iOS, and Android systems, ensuring maximum
2. reach without platform restrictions.
3. [**ASP.NET**](https://asp.net/)**Core Expertise**: Our development team has strong expertise in [ASP.NET](https://asp.net/)  Core and the Microsoft technology stack, which aligns perfectly with our project requirements for building robust, scalable web applications.
4. **Proven Technology Stack**: [ASP.NET](https://asp.net/) Core provides excellent performance, security features, and scalability options that are essential for handling hotel reservation traffic and payment processing.
5. **Market Research**: Based on our analysis of the Egyptian travel market, most hotel bookings are made through web platforms, with a significant portion of mobile users accessing these platforms through mobile browsers rather than dedicated apps.

A graph of a hotel

AI-generated content may be incorrect.

*Figure 5: Primary platforms used for hotel bookings in the Egyptian market.*

### 1.5 Document Organization

This project documentation consists of five comprehensive chapters. These chapters are systematically organized to reflect the scientific methodology and developmental steps toward achieving our primary objectives. **Chapter 1: Introduction**

This chapter introduces the project foundation, including the project objectives, motivation behind development, scope of work, and overall project layout and

organization.

**Chapter 2: Literature Review**

This section provides readers with a comprehensive overview of existing research, technologies, and market analysis relevant to hotel reservation systems and web applications.

**Chapter 3: System Analysis**

This chapter delivers detailed system analysis, encompassing both functional and non-functional requirements, use case diagrams with comprehensive description cards, sequence diagrams, and the development methodology employed in the project.

**Chapter 4: System Design**

In this section, we present the architectural design patterns, detailed class diagrams, database schema, and user interface prototypes and samples.

**Chapter 5: Conclusion and Future Work**

The final chapter summarizes the project achievements, presents conclusions drawn from the development process, and outlines potential future enhancements and expansion opportunities.





## 2.1 Introduction

A literature review is a comprehensive discussion and analysis of published information within a specific subject area, often focusing on developments during a particular time period. In this chapter, we present a detailed literature review of existing hotel reservation systems and web applications in the hospitality industry. This review examines various technological approaches, architectural patterns, and implementation strategies employed in current booking platforms, with particular focus on systems utilizing similar technologies to our Bookify application, including [ASP.NET](https://asp.net/) Core, Entity Framework, and modern web development frameworks.

The analysis will cover both commercial solutions and academic research in the field of online hotel reservation systems, highlighting strengths, limitations, and emerging trends that inform our development approach.

### Related work

Hotel reservation systems exist in the Egyptian market as both standalone applications and integrated services that can be used by both travelers and hotel managers. According to our market research, the majority of small to medium-sized hotels in Egypt currently lack comprehensive digital reservation systems.

Our research analyzed 9 leading hotel booking platforms, including 5 regional applications operating in Egypt and 4 global applications from international markets. In Egypt, it's not common to find unified reservation systems that serve multiple hotels across different regions, as most solutions are developed individually for specific hotel chains or properties. Regionally, we found 5 applications operating but with limited market penetration and geographical coverage. Globally, all 4 applications we studied demonstrate large-scale operations and extensive market support.

Details about these global hotel reservation systems can be found in Section 2.2.1

Regarding standalone booking applications, we have identified several platforms that provide hotel reservation services in the Egyptian market. We selected three prominent examples for detailed analysis:

1. [**Booking.com**](https://booking.com/) (Global platform with significant presence in Egypt)
2. **EgyRooms** (Local Egyptian booking platform)
3. **MakeMyTrip** (International platform serving the Egyptian market)

Details about these hotel reservation services and their features can be found in Section 2.2.2

#### 

#### 2.1.1 Global Hotel Reservation Systems

#### "Booking.com" (Global):

#### A global online travel reservation platform offering accommodation bookings

#### across hotels, apartments, villas, and other properties. Features include real-time

#### availability, guest reviews, integrated payment processing, loyalty programs, and

#### seamless access to multiple travel services under one account.

#### Advantages:

#### - Available 24/7 with instant confirmation

#### - Extensive property selection worldwide

#### - Comprehensive guest review system

#### - Multi-language and multi-currency support

#### Disadvantages:

#### - High commission fees for property owners

#### - Customer service can be slow during peak seasons

#### - Some users report hidden fees

#### "Expedia" (USA/Global):

#### Provides complete travel packages including flights, hotels, and car rentals.

#### The platform offers virtual tours, flexible cancellation policies, reward

#### programs, and integration with airline and car rental services.

#### Advantages:

#### - Bundle deals for cost savings

#### - Rewards program for frequent users

#### - User-friendly mobile app

#### - Strong customer protection policies

#### Disadvantages:

#### - Complex cancellation policies

#### - Customer service challenges

#### - Limited local market knowledge in some regions

#### "Agoda" (Asia/Global):

#### An online travel booking platform with strong presence in Asia, offering

#### competitive pricing, promotional deals, and a focus on Asian markets.

#### Features include mobile-exclusive deals, price comparisons, and instant

#### confirmation.

#### Advantages:

#### - Competitive pricing in Asian markets

#### - Frequent promotional offers

#### - User-friendly interface

#### - Strong mobile presence

#### Disadvantages:

#### - Limited customer support options

#### - Payment issues reported in some regions

#### - Less comprehensive in Western markets

#### "Airbnb" (Global):

#### Offers unique accommodations including private homes, apartments, and

#### experiences. The platform includes host and guest verification, secure

#### messaging, and integrated payment systems.

#### Advantages:

#### - Unique accommodation options

#### - Direct host communication

#### - Competitive pricing

#### - Local experiences available

#### Disadvantages:

#### - Quality consistency issues

#### - Safety concerns in some cases

#### - Limited hotel options

#### 2.1.2 Regional Hotel Reservation Applications in Egypt

#### "EgyRooms":

#### A comprehensive Egyptian hotel booking platform that lets users book

#### accommodations across Egypt, compare prices, view property photos, and

#### access customer support. Includes hotel profiles, guest ratings, and seasonal

#### discounts.

#### Advantages:

#### - Focus on Egyptian market

#### - Local payment options

#### - Arabic language support

#### - Competitive local pricing

#### Disadvantages:

#### - Limited international properties

#### - Basic user interface

#### - Occasional booking confirmation delays

#### "Jumia Travel" (Now Jumia):

#### A service within the Jumia ecosystem focused on helping people book hotels

#### across Africa and specifically in Egypt. Emphasizes simplicity with filtering

#### by location, rating, price, and available dates.

#### Advantages:

#### - Integration with Jumia ecosystem

#### - Local market understanding

#### - Mobile-friendly platform

#### - Regular promotional offers

#### Disadvantages:

#### - Limited advanced features

#### - Customer service inconsistencies

#### - Fewer international chain hotels

#### "Jazeera Hotels":

#### An Egyptian hotel booking app that lets users browse and book hotels across

#### Egypt. Includes verified guest reviews, allows filtering by governorate and date,

#### and offers travel content and tips.

#### Advantages:

#### - Specialized in Egyptian hotels

#### - Arabic-first interface

#### - Local customer support

#### - Competitive commission rates

#### Disadvantages:

#### - Limited international coverage

#### - Basic technology infrastructure

#### - Fewer payment options

#### - Limited integration with global systems

### The Proposed System

As highlighted in our analysis of existing solutions, current hotel reservation platforms face

several limitations including platform restrictions, lack of essential services, unreliable rating

systems, complex registration processes, limited booking windows, and disproportionately

high service fees. Additionally, many small to medium-sized hotels in our target market

lack integrated digital reservation systems.

**Key Features of Bookify:**

**1. Simplicity and User-Friendly Design:**

* Quick registration and profile setup with minimal steps
* Instant access to comprehensive hotel and room listings
* Intuitive navigation and search functionality

**2. Transparent Rating System:**

* Verified guest reviews and ratings based on actual stays
* Detailed hotel and room information with high-quality images
* Real-time availability and pricing updates

**3. Mutual Benefits for Travelers and Hotels:**

* Streamlined hotel registration and management portal
* Easy inventory management and booking oversight
* Direct communication channel between guests and hotels
* Performance analytics and customer feedback for continuous improvement

**4. Comprehensive Service Integration:**

* Secure payment processing with multiple options
* Real-time booking confirmation
* Flexible cancellation and modification policies
* Loyalty programs and special offers

**5. Advanced Technology Integration:**

* AI-powered recommendation engine based on user preferences
* Smart search filters and personalized results
* Mobile-responsive design for seamless cross-platform experience
* Real-time notification system for booking updates

**System Architecture:**

Bookify employs a robust N-Tier architecture with:

* **Presentation Layer:** [ASP.NET](https://asp.net/) Core MVC with responsive UI
* **Business Logic Layer:** Dedicated service classes handling core operations
* **Data Access Layer:** Repository pattern with Entity Framework Core
* **Security Layer:** [ASP.NET](https://asp.net/) Identity with role-based access control

**Competitive Advantages:**

* **Accessibility:** Web-based platform accessible across all devices and operating systems
* **Cost-Effectiveness:** Lower commission rates compared to global platforms
* **Local Focus:** Tailored specifically for the Egyptian market while maintaining global

standards

* **Reliability:** Secure transactions and verified information
* **Scalability:** Architecture designed for future expansion and feature enhancements

Bookify aims to revolutionize the hotel booking experience in Egypt by providing a trustworthy,

efficient, and comprehensive platform that addresses the gaps in existing solutions while

introducing innovative features that benefit both travelers and hotel operators.





### Introduction

### This chapter presents the system analysis for the Bookify Hotel Reservation System, including both functional and non-functional requirements specification. It also covers the system design through

### ERD (Entity Relationship Diagram), use case diagram, and sequence diagrams to specify the system functions in detail.

### Requirements Specification

### Requirements specification is the comprehensive description of what

### the system is supposed to do, including visible functions in the application and underlying concepts the application must consider.

#### Functional Requirements

#### Functional requirements define the required operations of the system that

#### describe how it works. The functional requirements of the Bookify system are:

#### User Registration: Creating new user accounts, including full name, password, email address, and personal profile with essential information used by other system functions.

#### Hotel Registration: Creating new hotel accounts with hotel details, owner information, and administrative approval process.

#### Profile Management: The process of editing user profile data including personal information, preferences, and contact details.

#### Hotel Management: The process for hotel owners to add and manage hotel information, room types, amenities, and pricing.

#### Admin Approval Notification: The process of sending notifications to administrators about new hotel registration requests pending approval.

#### Administrator Approval: The process for administrators to review and approve/reject new hotel registration requests with full access to hotel details.

#### Room Inventory Management: The process of adding and managing room availability, types, prices, and capacity for each hotel.

#### Schedule Updates: The process of editing room availability, pricing, and special offers based on seasonal demand.

#### Booking Notifications: The process of sending notifications to users when there are changes to their bookings or new offers matching their preferences.

#### Low Rating Alert: The process of sending notifications to administrators about hotels receiving consistently low ratings.

#### Booking Records View: The process for administrators to access comprehensive booking records with details like hotel, dates, pricing, and customer information.

#### Hotel Performance Monitoring: The process for administrators to view all hotels, filter by ratings, and communicate with hotel managers for improvements.

#### Review Display: The process of showing verified guest reviews and ratings for each hotel and room type.

#### Review Submission: The process for guests to write and submit reviews with ratings and comments after completing their stay.

#### Hotel Search: The process of searching and selecting hotels based on various criteria.

#### Advanced Filtering: The process of filtering available hotels by location, price range, amenities, dates, and guest ratings.

#### Reservation Booking: The process for users to book rooms by selecting preferred dates, room types, and special requirements.

#### Secure Payment Processing: The process for users to complete online payments through integrated payment gateways with multiple payment options.

#### Customer Support Chat: The process for users to communicate with hotel staff and customer support for inquiries, special requests, and post-booking assistance.

#### Loyalty Program Management: The process of managing reward points, special offers, and exclusive deals for frequent users.

#### Real-time Availability Check: The process of providing instant room availability updates and confirmation during the booking process.

#### Cancellation and Modification: The process for users to modify or cancel bookings according to hotel policies with automated refund processing.

#### Non-Functional Requirements

"Non-Functional requirements - also known as system qualities - are constraints and quality attributes of the system".

The non-functional requirements of the Bookify Hotel Reservation System are:

1. **Role-based Access Control**: Only verified hotel owners/managers can register as hotel partners, while any user can register as a guest/customer.
2. **Booking Conflict Prevention**: Users cannot book multiple rooms at different hotels for overlapping dates and times.
3. **Capacity Management**: Each room type has a limited inventory per date; once fully booked, the system automatically shows as unavailable.
4. **Verified Reviews System**: Users can only submit reviews and ratings after completing their stay and checkout.
5. **Booking Modification Policy**: Hotel managers cannot modify booking details within 24 hours of check-in time.
6. **Payment Flexibility**: Online payment is optional, but users with repeated no-shows receive warnings and temporary account restrictions.
7. **Reliability**: The system maintains 99.5% uptime with real-time availability updates and instant booking confirmations.
8. **Usability**: The application features an intuitive, user-friendly interface with clear navigation and minimal steps for booking.
9. **Security**: Implements high-level security measures including encrypted payment processing, secure authentication, and data protection compliance.
10. **Dependability**: Provides a single, reliable platform for all hotel bookings with 24/7 accessibility and customer support.
11. **Maintainability**: The system is built with clean, modular code following N-Tier architecture and design patterns for easy future maintenance and feature expansion.
12. **Accessibility**: The web application is fully responsive and accessible from any device (desktop, tablet, mobile) across different browsers and operating systems.
13. **Performance**: The system handles peak traffic during holiday seasons with response times under 2 seconds for search and booking operations.
14. **Scalability**: The architecture supports horizontal scaling to accommodate growing numbers of users, hotels, and transactions.

**3.2.3 Use Case Diagram**

A use case diagram is a graphical description of the interactions between system

elements. There are three main components in the use case diagram: actors, use cases,

and relationships.

**Primary Actors:**

* Customer (Authenticated User)
* Administrator (System Admin)

**Sample Use Cases:**

* Search Hotels
* Make Reservation
* Process Payment
* Manage Hotel Profile
* Approve Hotel Registration
* Write Review
* Manage Bookings

Each use case in the diagram requires detailed description cards that specify the

use case name, involved actors, preconditions, postconditions, and the flow of events.

#### *Figure 3.1: Use Case Diagram for Bookify Hotel Reservation System*

#### [Use Case Diagram would be inserted here showing interactions between actors

#### and system functionalities]

#### Example Use Case Description Card:

#### Use Case: Make Reservation Actors: Customer, Payment System Preconditions: User is logged in, room is available Main Flow:

#### Customer selects dates and room type

#### System shows availability and pricing

#### Customer enters guest details

#### System processes payment

#### Booking confirmation generated

#### Notifications sent to both parties Postconditions: Reservation created, room inventory updated

#### 

**Figure9: Use case diagram**

Each use case diagram must have description cards that describe each use case, showing its name, actors involved and flow of events.

**Table 1: Register Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Register |
| Actors involved: | Customer |
| Flow of events: | 1. Customer accesses registration page 2. System displays registration form 3. Customer enters ID, name, and email 4. System validates data 5. Account is created 6. Confirmation message displayed |

**Table 2:  Search Rooms Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Search Rooms |
| Actors involved: | Customer |
| Flow of events: | 1. Customer enters room number or type  2. System queries database  3. Matching rooms displayed  4. Customer can select room for details |

**Table 3: Filter Rooms Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Filter Rooms |
| Actors involved: | Customer |
| Flow of events: | 1. Customer applies filters (date, price, type, capacity)  2. System refines results based on filters  3. Filtered rooms displayed  4. Customer can adjust filters |

**Table 4: Add to Cart Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Add to Cart |
| Actors involved: | Customer |
| Flow of events: | 1. Customer selects room  2. Chooses booking dates (from-to)  3. System checks availability for dates  4. Room added to cart  5. Cart updated with selection |

**Table 5: Make Payment Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Make Payment |
| Actors involved: | Rooms in cart, dates selected |
| Flow of events: | 1. Customer proceeds to checkout  2. System displays total amount  3. Customer enters payment details  4. System processes payment  5. Booking confirmed  6. Receipt generated |

**Table 6: Manage Room Types Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Manage Room Types |
| Actors involved: | Administrator |
| Flow of events: | 1. Admin accesses room types management  2. System displays existing room types  3. Admin can add new room type (name, price, capacity)  4. Admin can update existing room types  5. Admin can delete room types  6. Changes saved to system |

**Table 7:** **Add New Room Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Add New Room |
| Actors involved: | Admin |
| Flow of events: | 1. Customer accesses review section  2. System displays review form  3. Customer provides rating and comments  4. System validates review content  5. Review is submitted and stored  6. Hotel rating is updated |

**Table 8:  View System Bookings Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | View System Bookings |
| Actors involved: | Admin |
| Flow of events: | 1. Admin accesses bookings page  2. System displays all system bookings  3. Admin can view booking details  4. Admin can filter bookings  5. System shows booking statistics |

**Table 9:  Update Room Use Case Description Card**

|  |  |
| --- | --- |
| Use case name: | Update Room |
| Actors involved: | Admin |
| Flow of events: | 1. Admin selects room to update  2. System displays current room data  3. Admin modifies room details  4. Admin can update photos  5. System saves changes  6. Confirmation displayed |

### Sequence diagram

A sequence diagram is a graphical description of the interaction between system

components for a specific function. Software engineers use sequence diagrams to illustrate

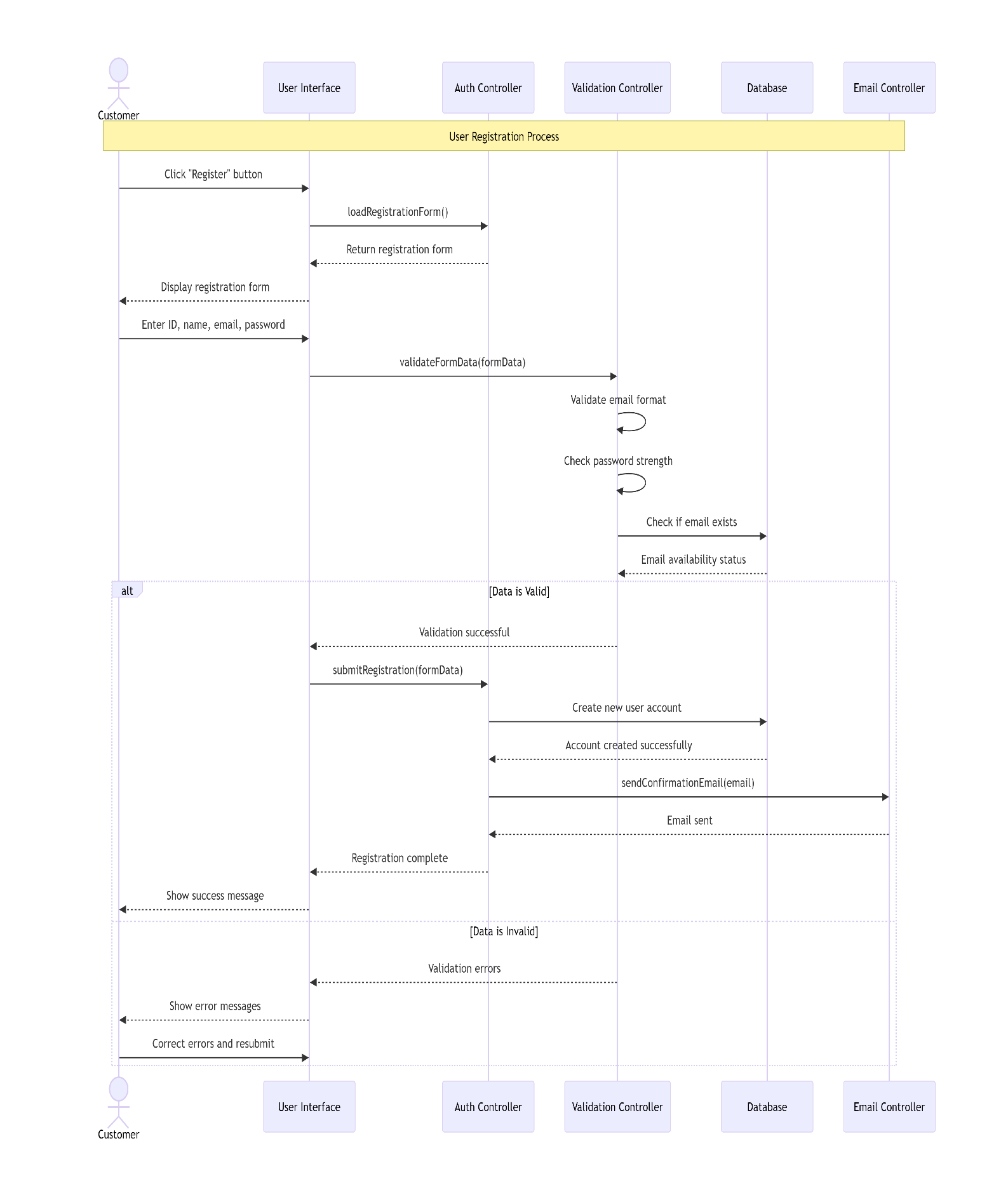
complex use cases and visualize the flow of messages between objects over time.

**Sequence Diagram for "Register" Use Case**

The sequence diagram for the Register use case is shown in Figure 3.2.

This diagram illustrates the complete user registration process from form access to account

confirmation.



**Figure10: Sequence diagram for Register Use Case**

### Development Methodology

### The Bookify system was developed using the Incremental

### Development Approach, building the system through sequential phases:

### Development Phases:

### First Increment: Foundation

### User authentication and registration

### Basic hotel search functionality

### Core database structure

### Second Increment: Booking Engine

### Room reservation system

### Booking management

### Payment integration

### Third Increment: Advanced Features

### Enhanced search filters

### Review and rating system

### Admin management panel

### Fourth Increment: Final Polish

### Performance optimization

### Security enhancements

### User experience improvements

### Benefits of Incremental Approach:

### Early delivery of working features

### Flexibility to adapt to changing requirements

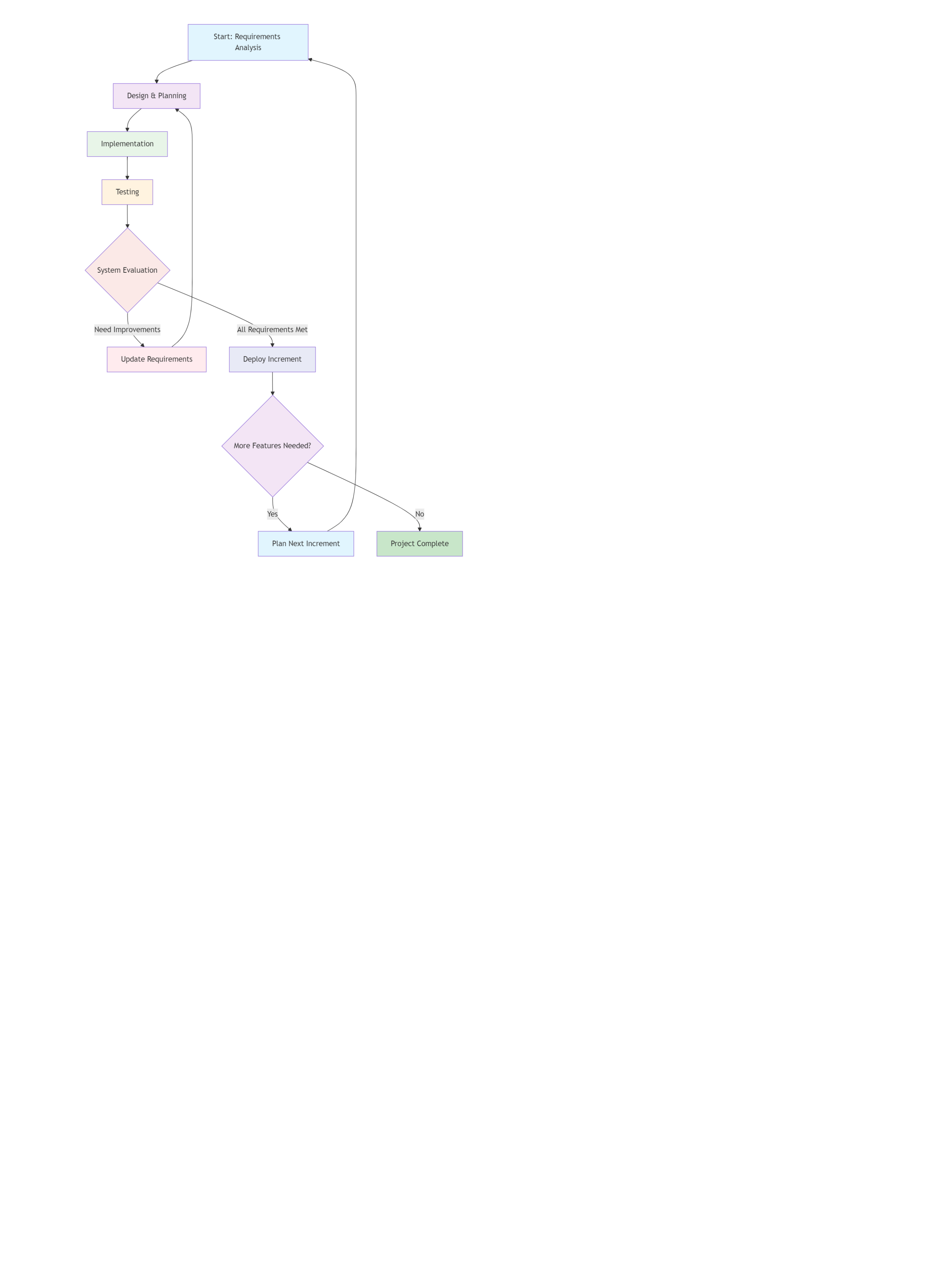
### Effective risk management

### Continuous improvement based on feedback

### The development process involved continuous iteration between analysis

### and design phases until achieving the complete system meeting all

### functional and non-functional requirements.

****

**Figure 11: Incremental model life cycle**

### Introduction



This chapter presents the comprehensive system design of the Bookify Hotel Reservation

System. It covers the architectural framework, system modules and components, their interfaces,

and the data flow throughout the system. The design follows a structured N-Tier architecture

pattern to ensure scalability, maintainability, and separation of concerns.

### Architectural design

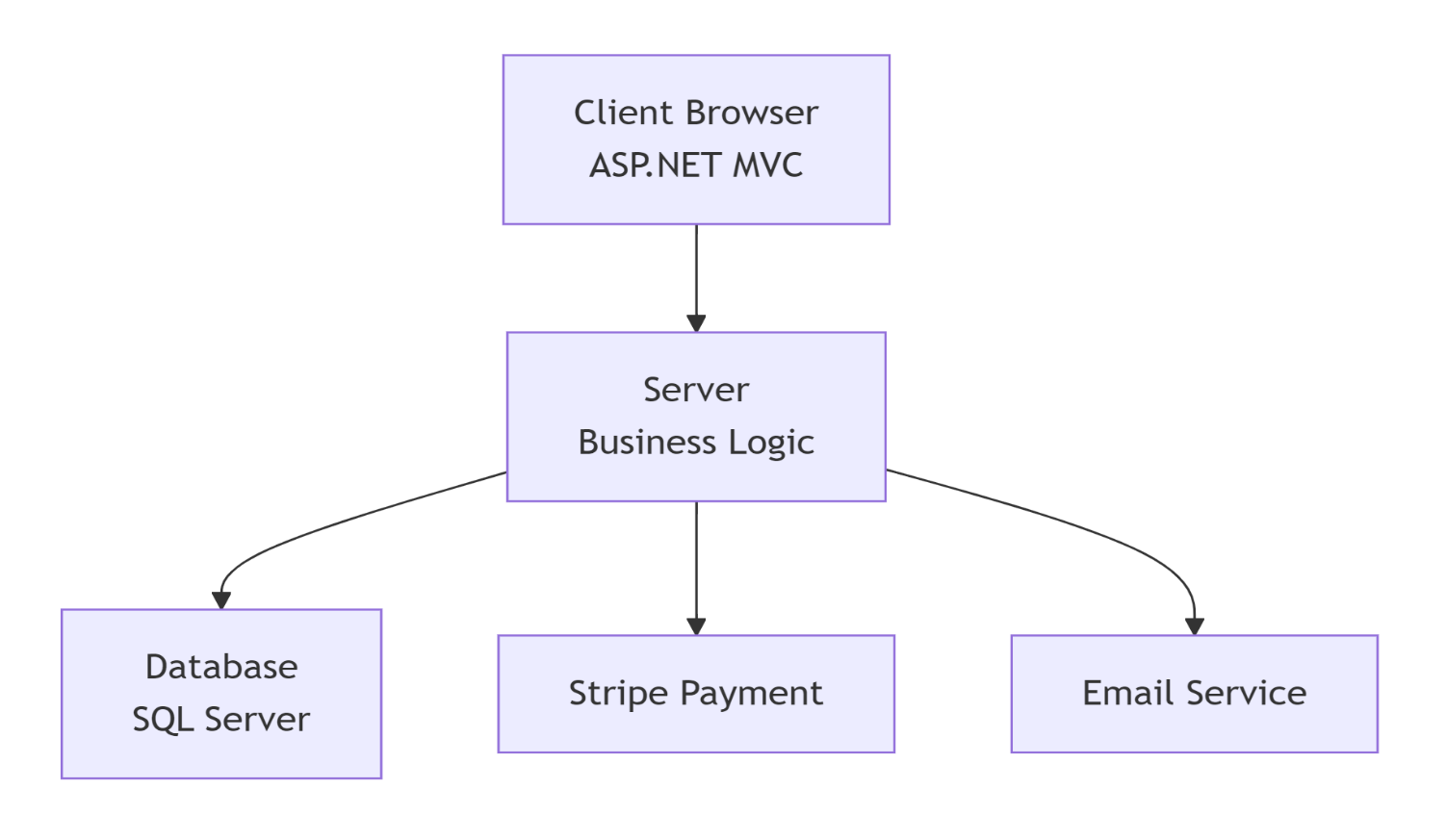
Bookify employs a client-server architecture structured around a three-tier model consisting of presentation, business logic, and data layers. The client side is built with [ASP.NET](https://asp.net/) Core MVC

handling all user interactions and interface rendering. The server side contains the core business

logic managing reservations, payments, and authentication through structured services. Data

persistence is achieved via SQL Server with Entity Framework Core using the Repository pattern for secure data access. External integrations include Stripe for payment processing and dedicated

services for email notifications and cloud storage. This architectural approach ensures clear separation of concerns, enhanced security, and scalable maintainability while delivering a responsive

user experience. 

**Figure12: System architecture**

* + 1. **Client components**

A screenshot of a phone

AI-generated content may be incorrect.

User Management:

This component handles all user-related operations

including registration, profile management, and

authentication. It interacts with the Users entity

and manages user data through the UserService

class, which processes user registration, login, and

profile updates.

Hotel Browser:

 This component provides comprehensive

hotel discovery and exploration capabilities.

It enables users to search, filter, and browse

hotels based on various criteria including

location, ratings, amenities, and price range.

The component integrates real-time availability

checks and displays verified customer reviews and ratings to help users make informed booking decisions.

A screenshot of a computer

AI-generated content may be incorrect.

Booking Engine:

This core component manages the entire reservation lifecycle from

room selection to booking

confirmation. It handles real-time

availability checks, room inventory management, pricing calculations, and booking status tracking. The engine ensures no double bookings occur and provides instant confirmation with detailed booking summaries.

Payment Processing :

A screen shot of a phone

AI-generated content may be incorrect.

 This secure component handles all

financial transactions within the system.

It integrates with multiple payment gateways,

Processes credit/debit card payments, and manages

Payment verification and refund processing. The

component ensures PCI compliance and provides

encrypted transaction handling for maximum

security.

Review and Rating :

This component facilitates the customer feedback system, allowing users to share their experiences through ratings and reviews. It includes features for review moderation, verified booking validation, and helpfulness voting. The system calculates aggregate ratings and displays authentic customer feedback to build trust and transparency.

A screenshot of a review

AI-generated content may be incorrect.

#### Server components:

#### A diagram of a diagram AI-generated content may be incorrect. The BookingService and PaymentService work in close

#### coordination to deliver a seamless booking experience.

#### The BookingService initiates the reservation process and

#### delegates payment processing to the PaymentService.

#### Once payment is confirmed, the BookingService finalizes

#### the reservation and updates all relevant system statuses.

#### This separation of concerns ensures that financial

#### operations remain secure while booking management

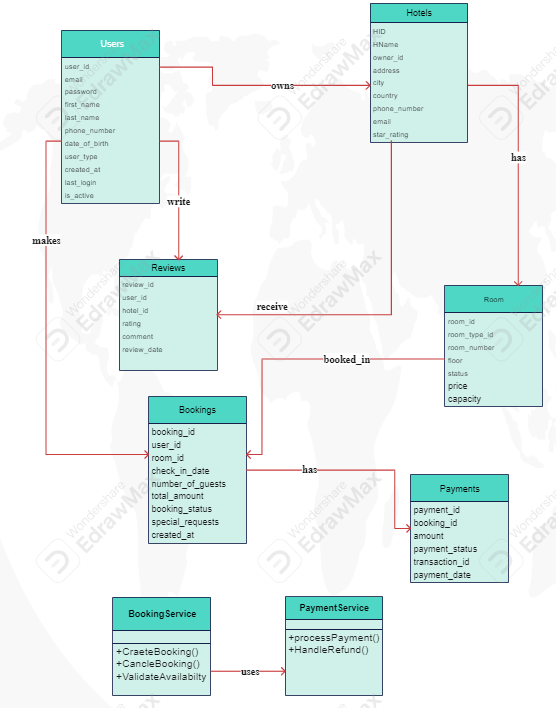
#### maintains business logic integrity.

### Object Design

#### Class Diagram

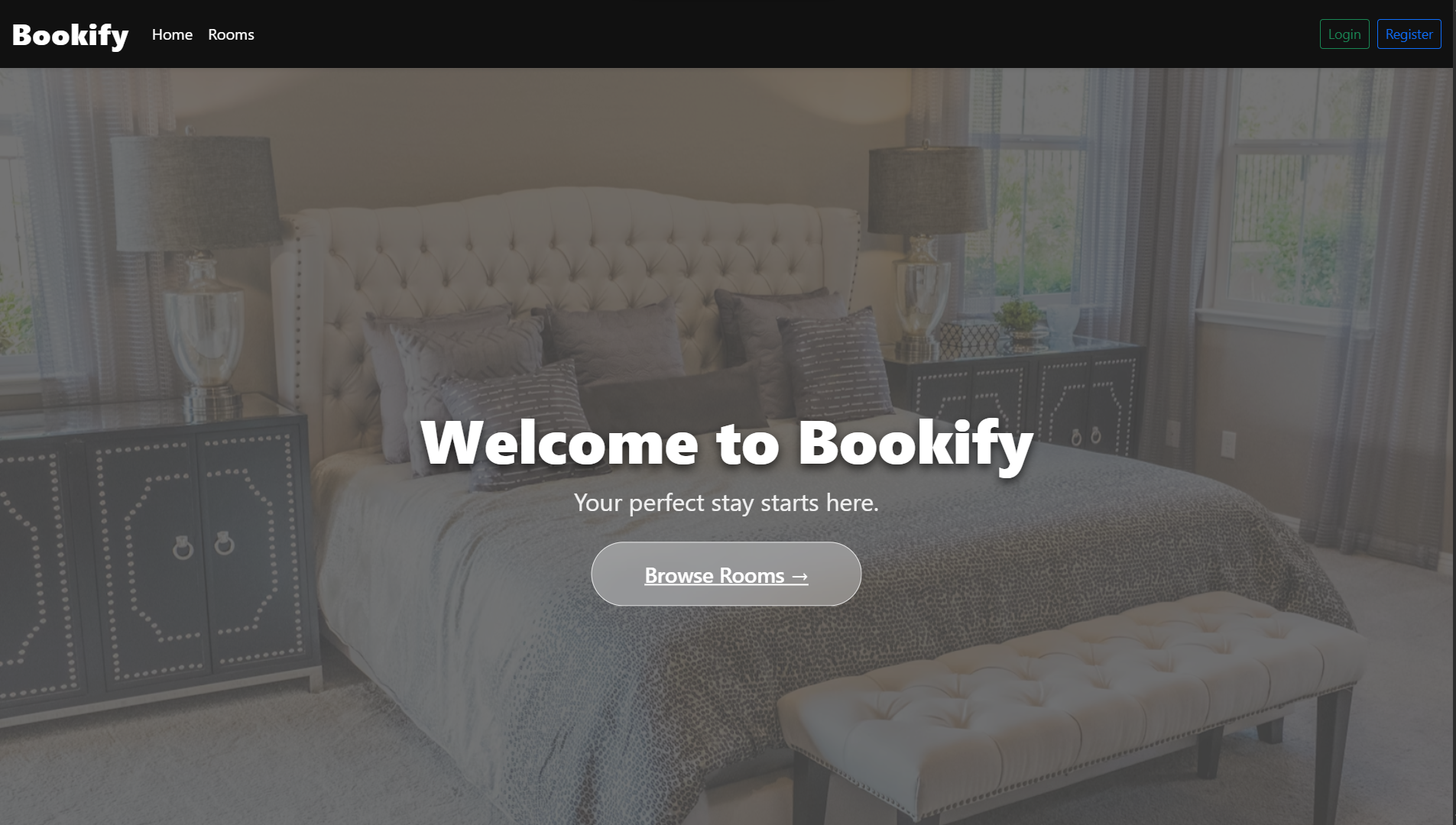
A class diagram is a graphical representation of the system's structure via its classes and their operations, attributes and the relationships between them.

Figure below presents  class diagram.



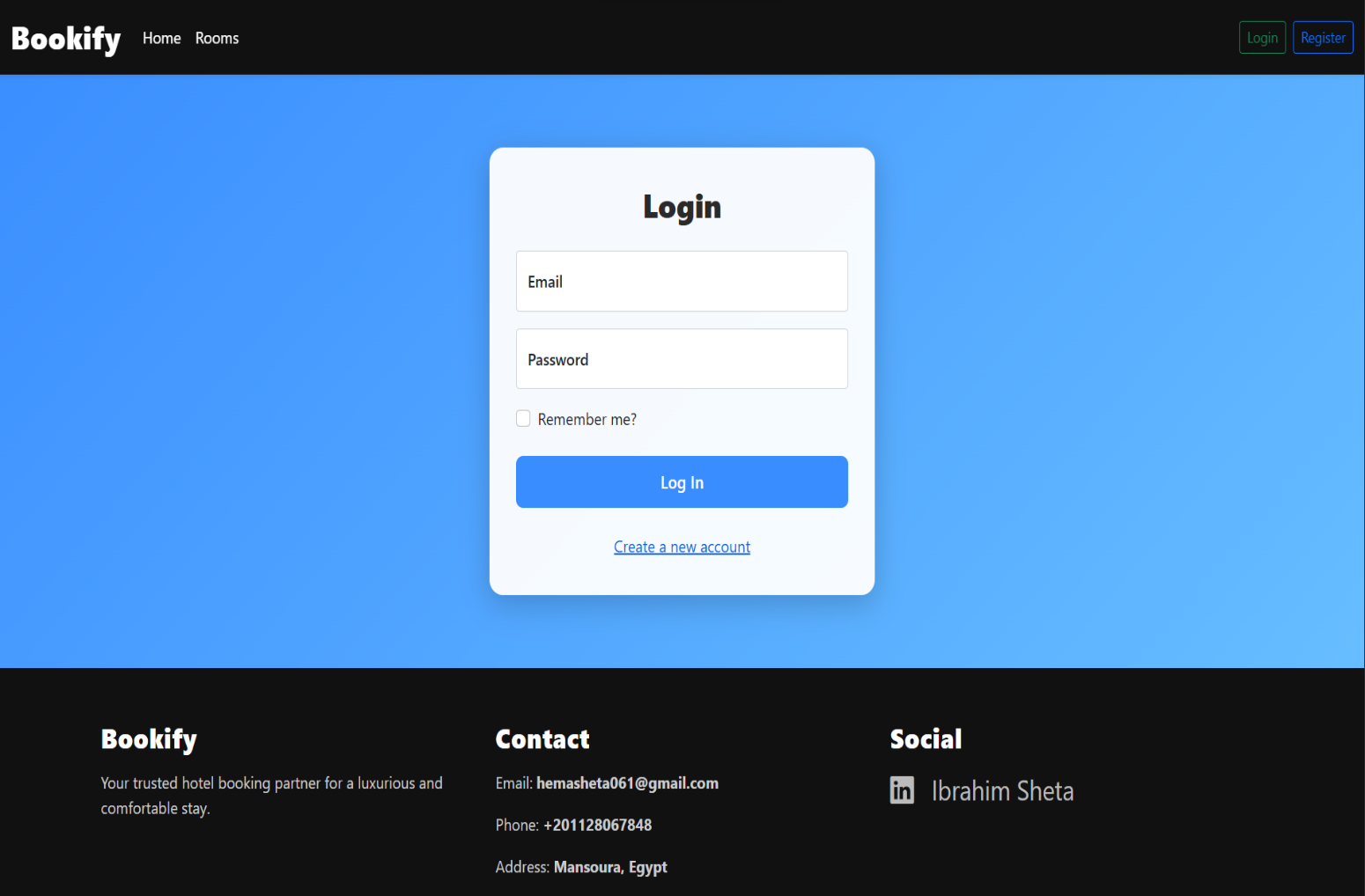
### User Interface Design

User Interface is visual components in the application that allow the user to interact with it. represents our application's UI.

**Home page:** this screen shows when the application starts

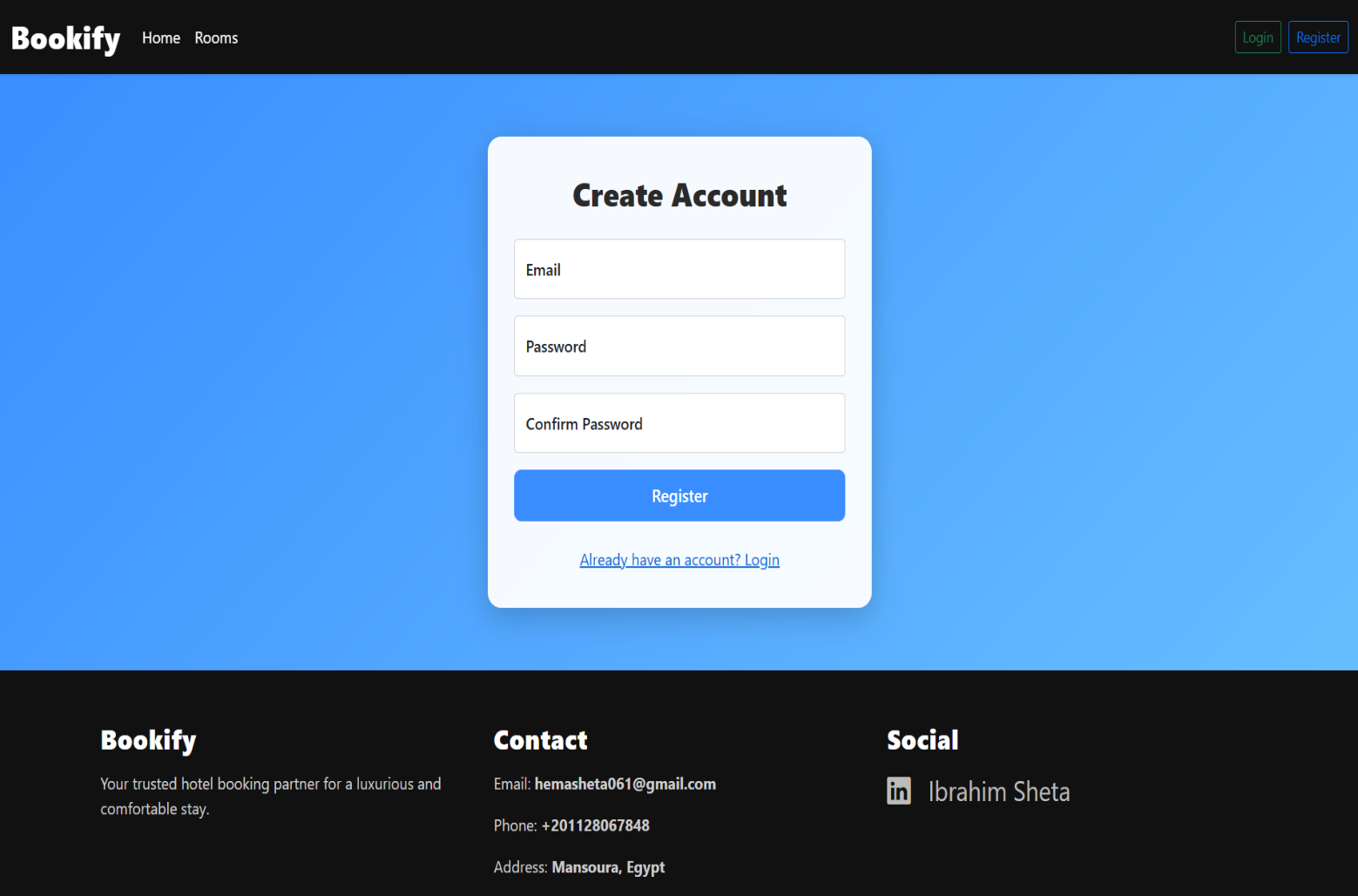
**Figure 20: home UI**

**Login screen:** this screen shows when the application starts, it is responsible for authenticate the user and give him access to the application, the user should enter a valid email and password to log in the application.



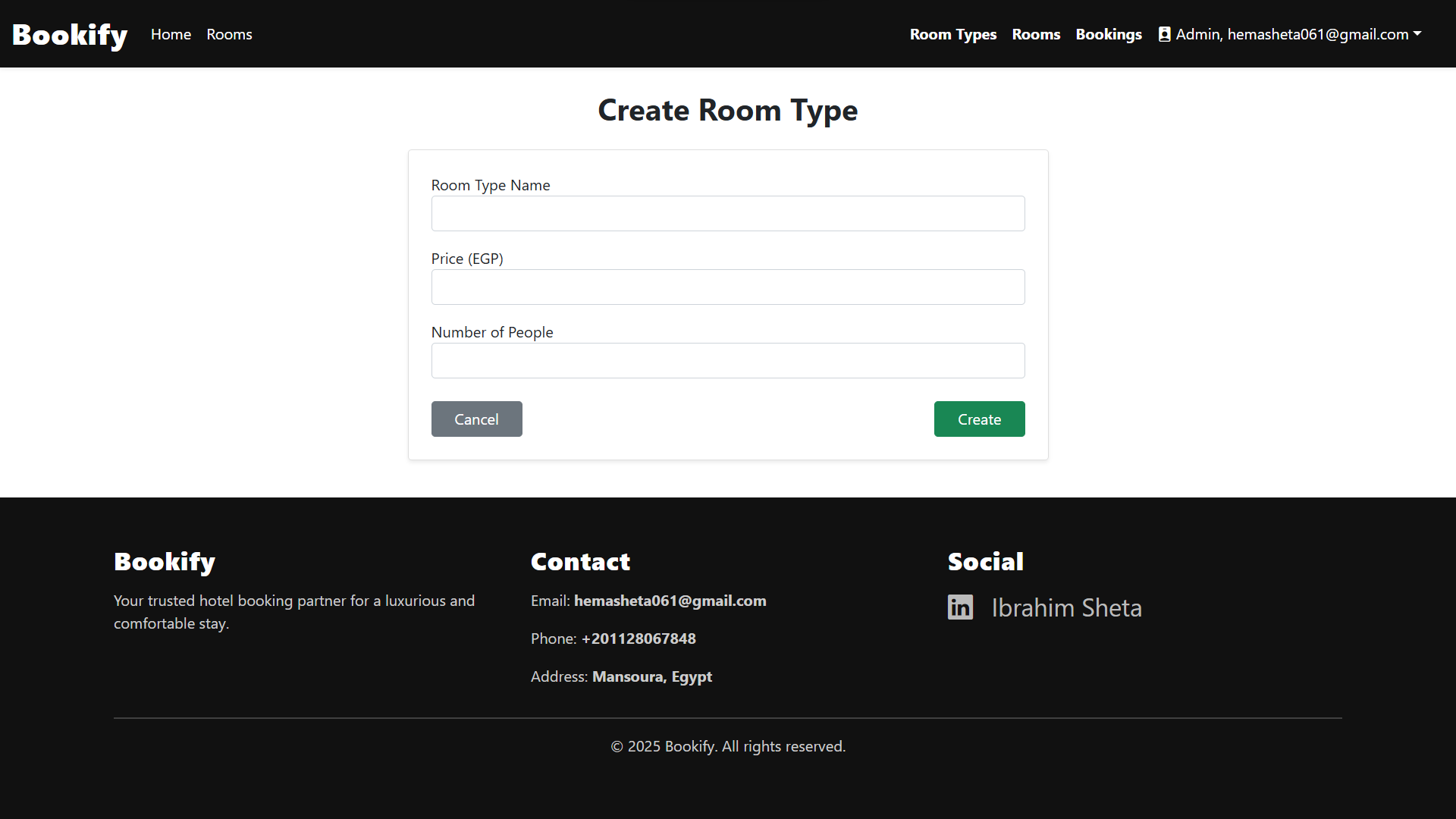
**Figure 21: Login UI**

**Signup screen:** Responsible for getting the required data from the user to create a new account, it can be shown through the Login screen.



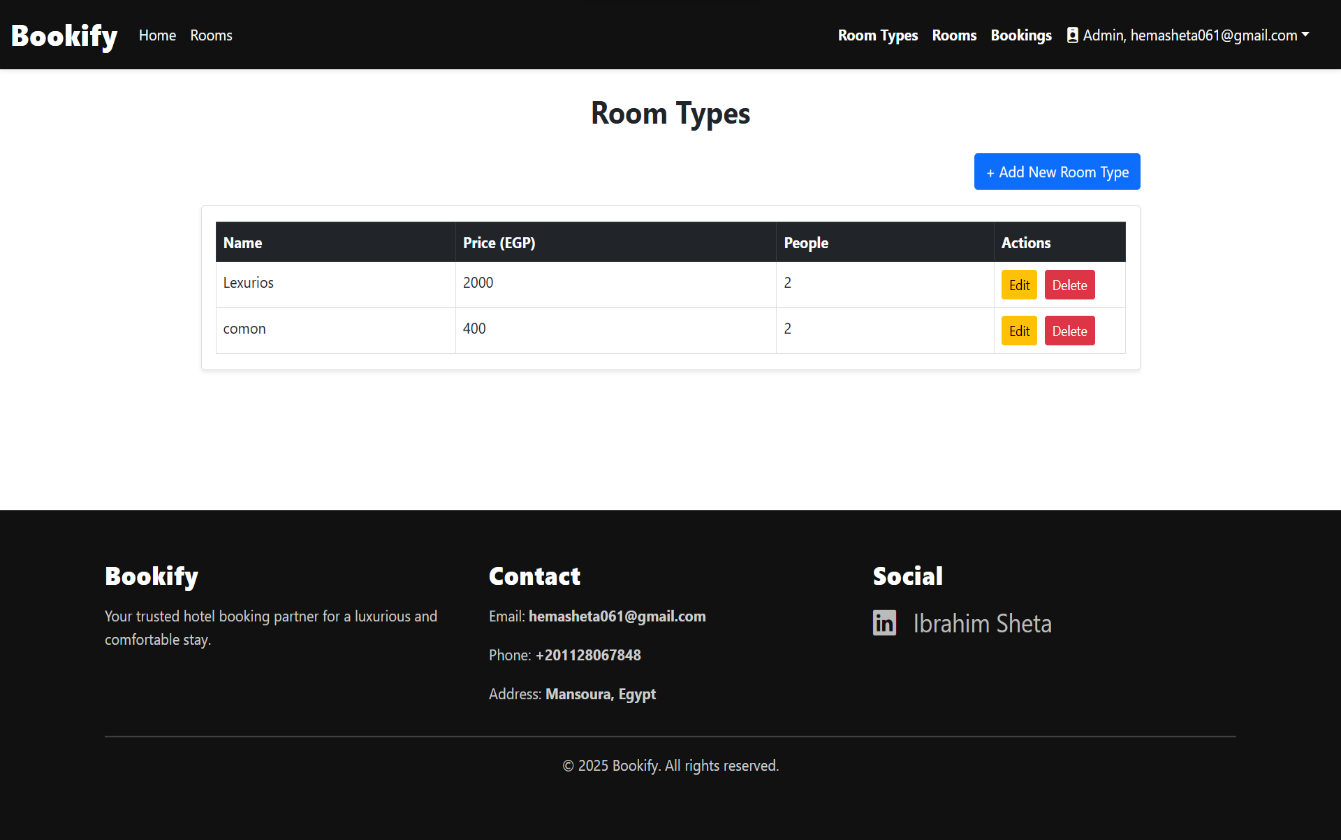
**Figure 22: Sign up UI**

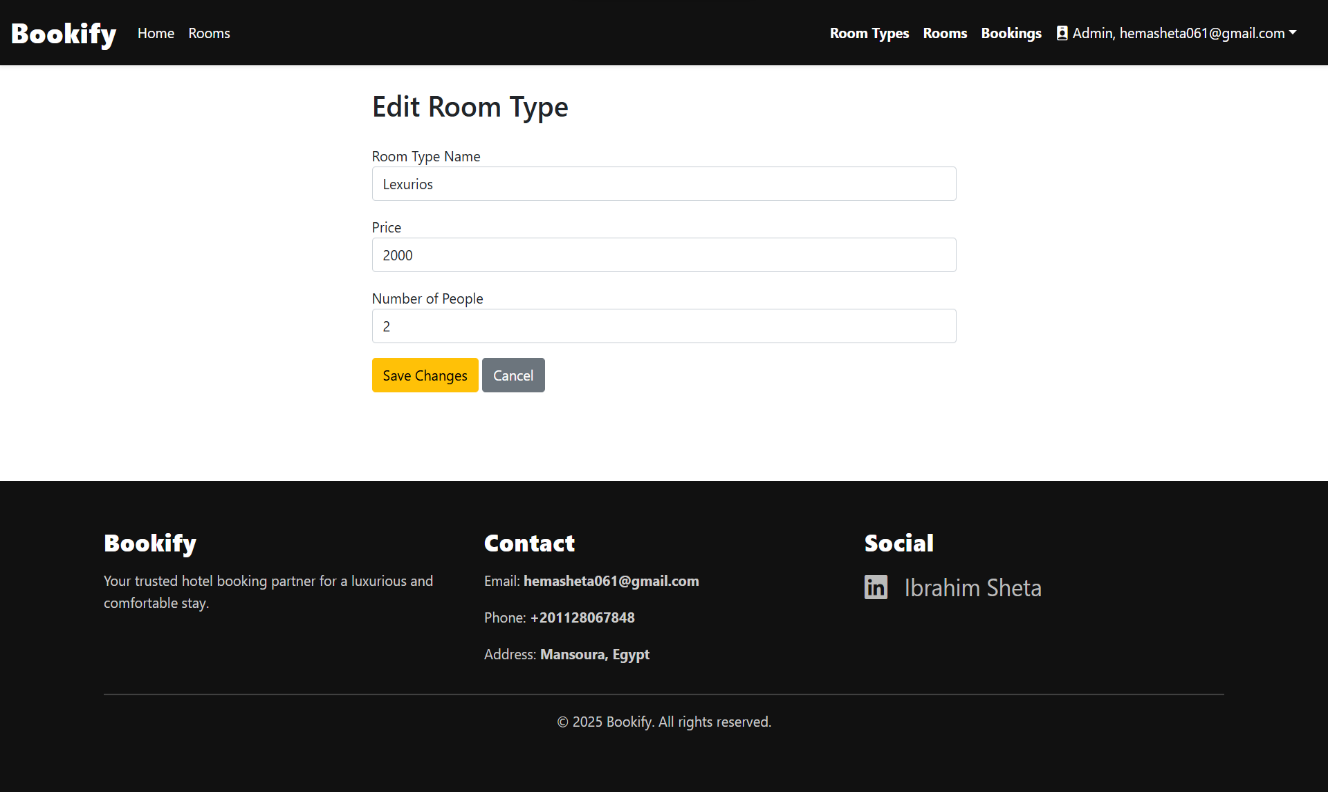
**Create room type screen:** Responsible for getting the required data from the owner to create a new room type



**Figure 23: create room type UI**

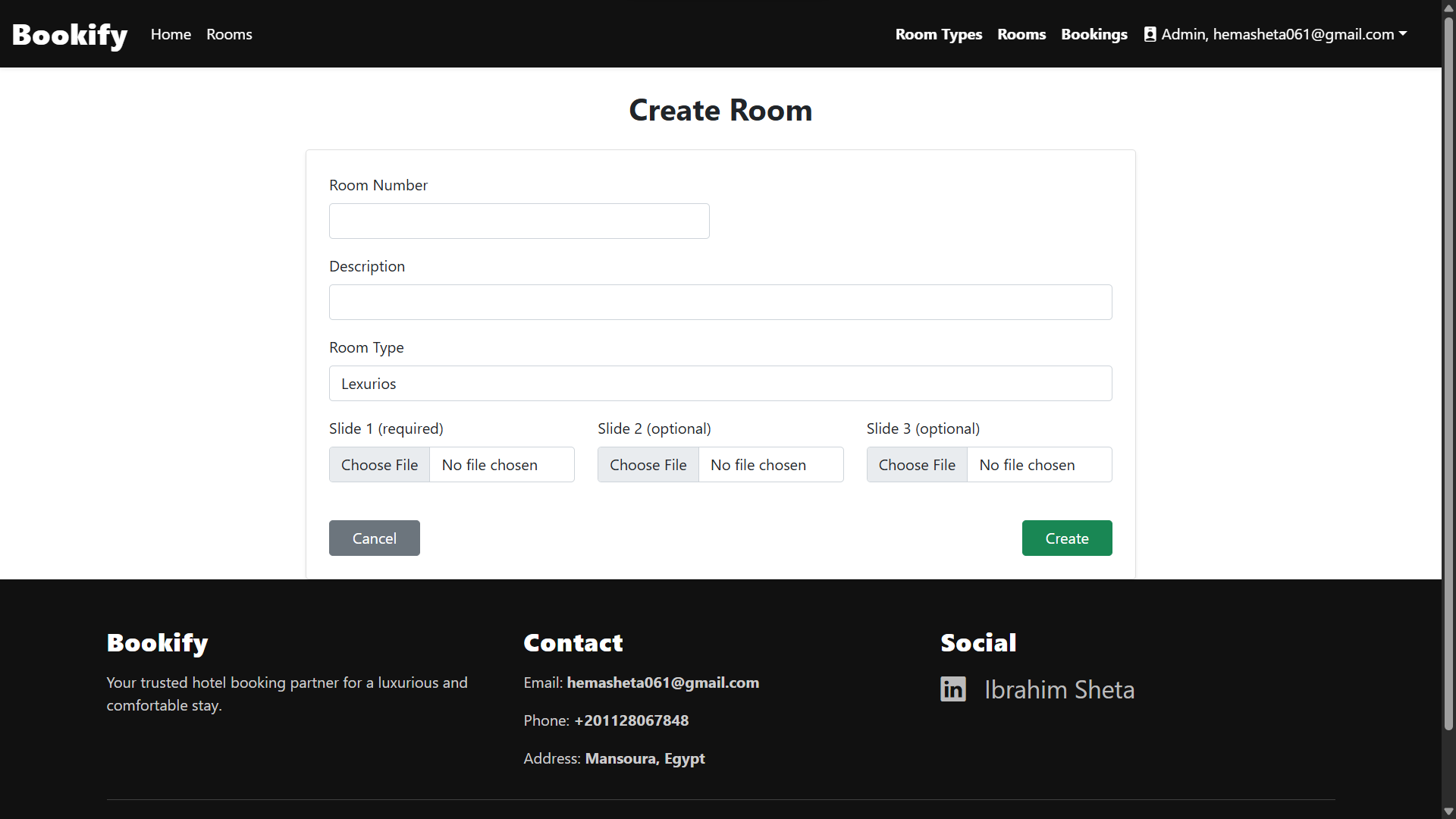
**Editing types screens:** Responsible for editing or delete the types of the room by the owner





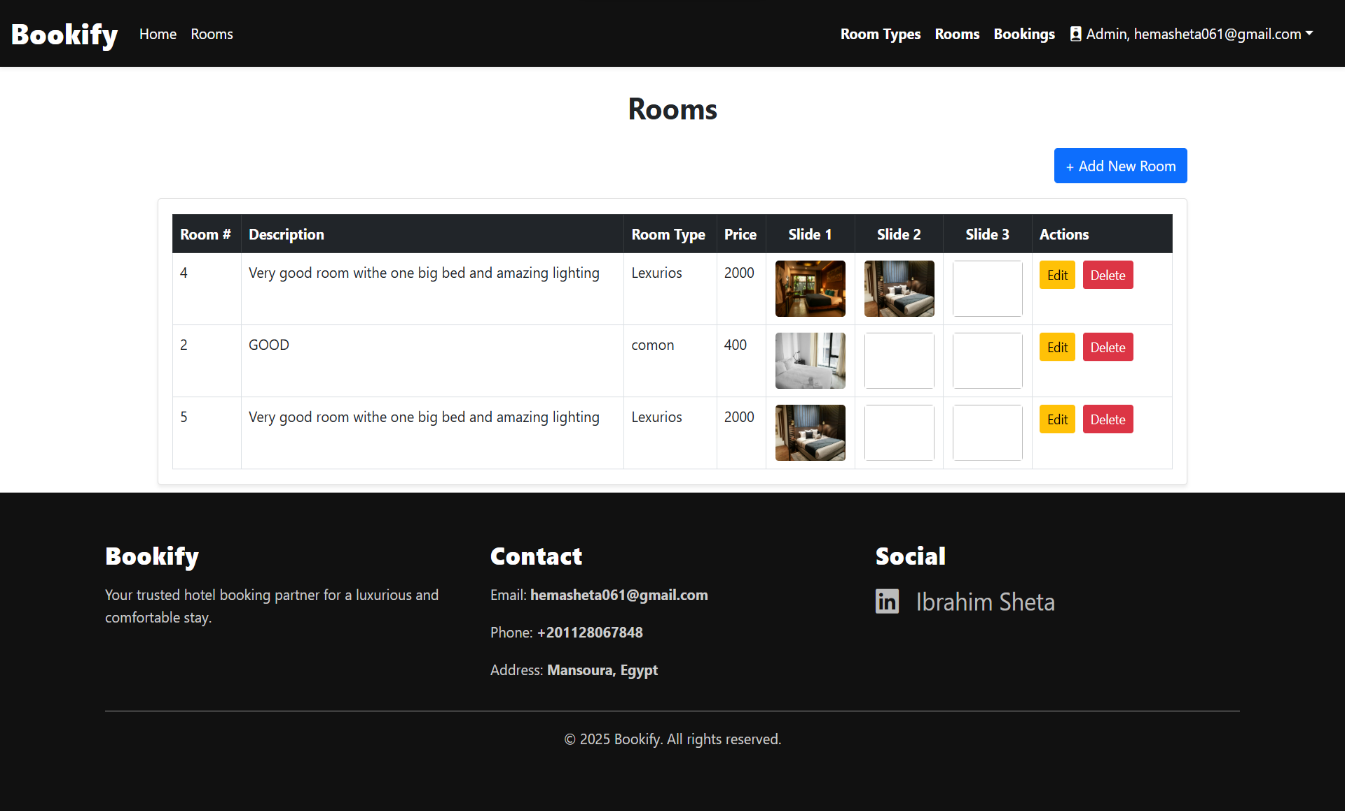
**Figure 24 , 25: editing types UI**

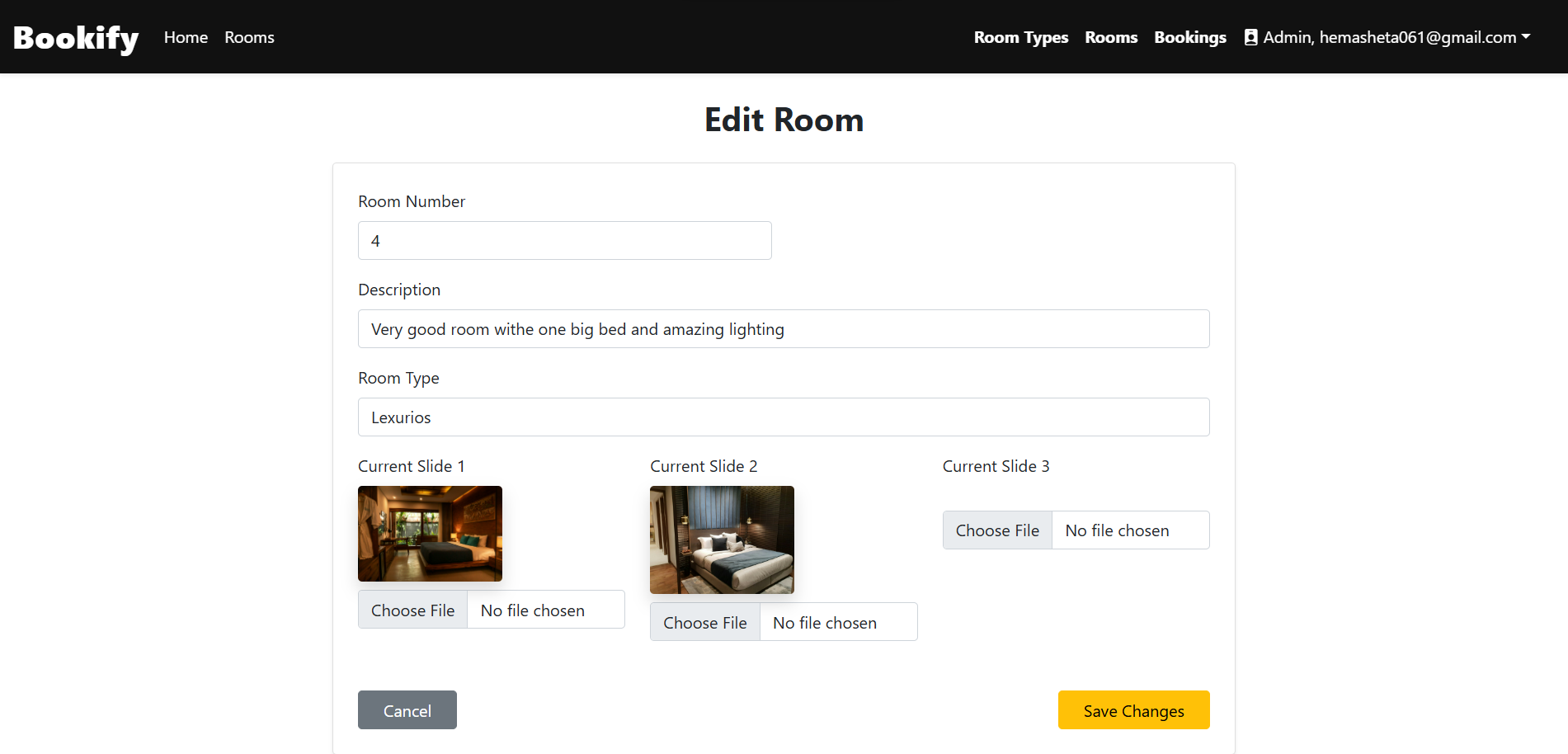
**Create room screen:** Responsible for getting the required data from the owner to create a new room, it can be shown through the customer room page.



**Figure 26: create room UI**

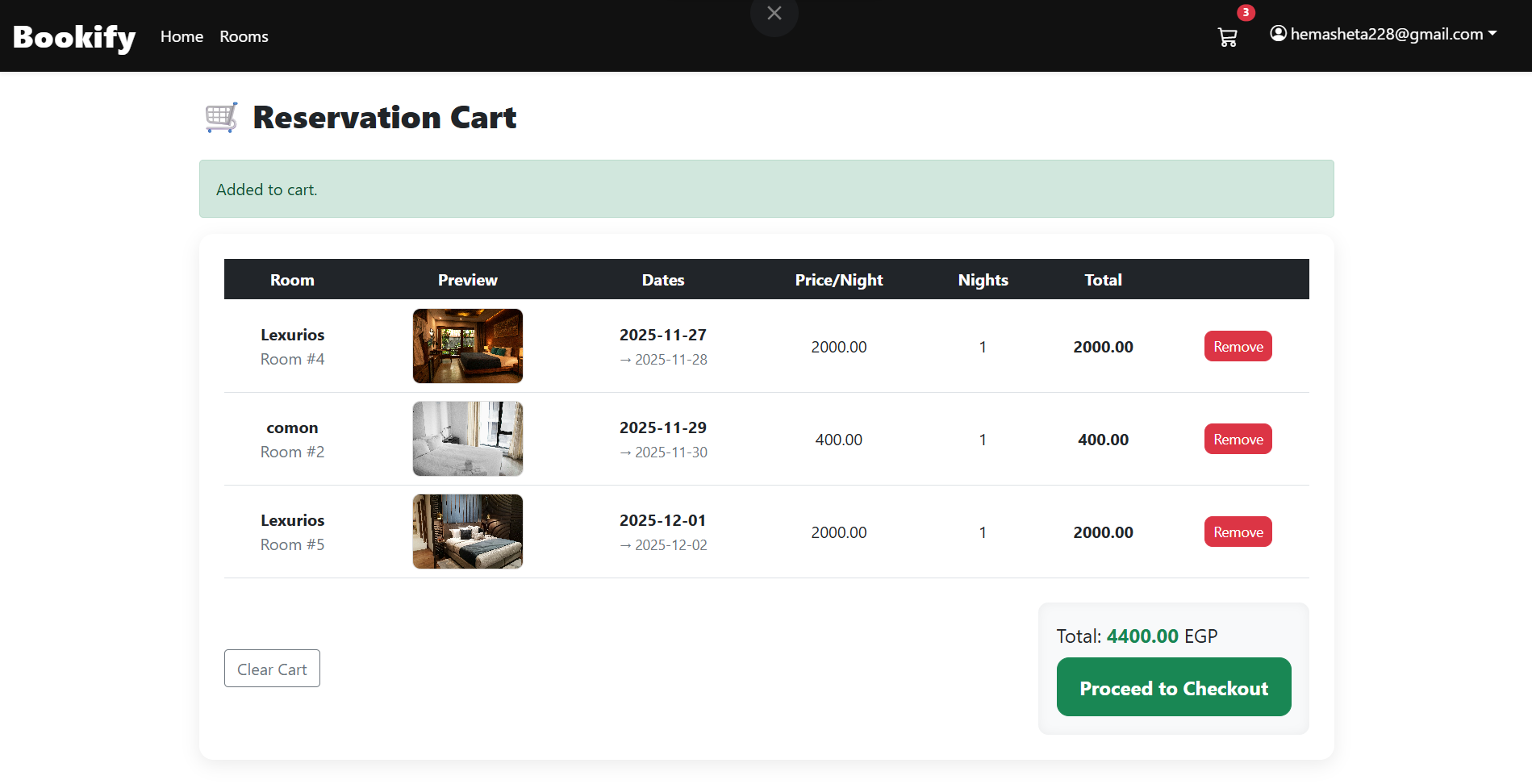
**Editing screens:** Responsible for editing or delete any room by the owner





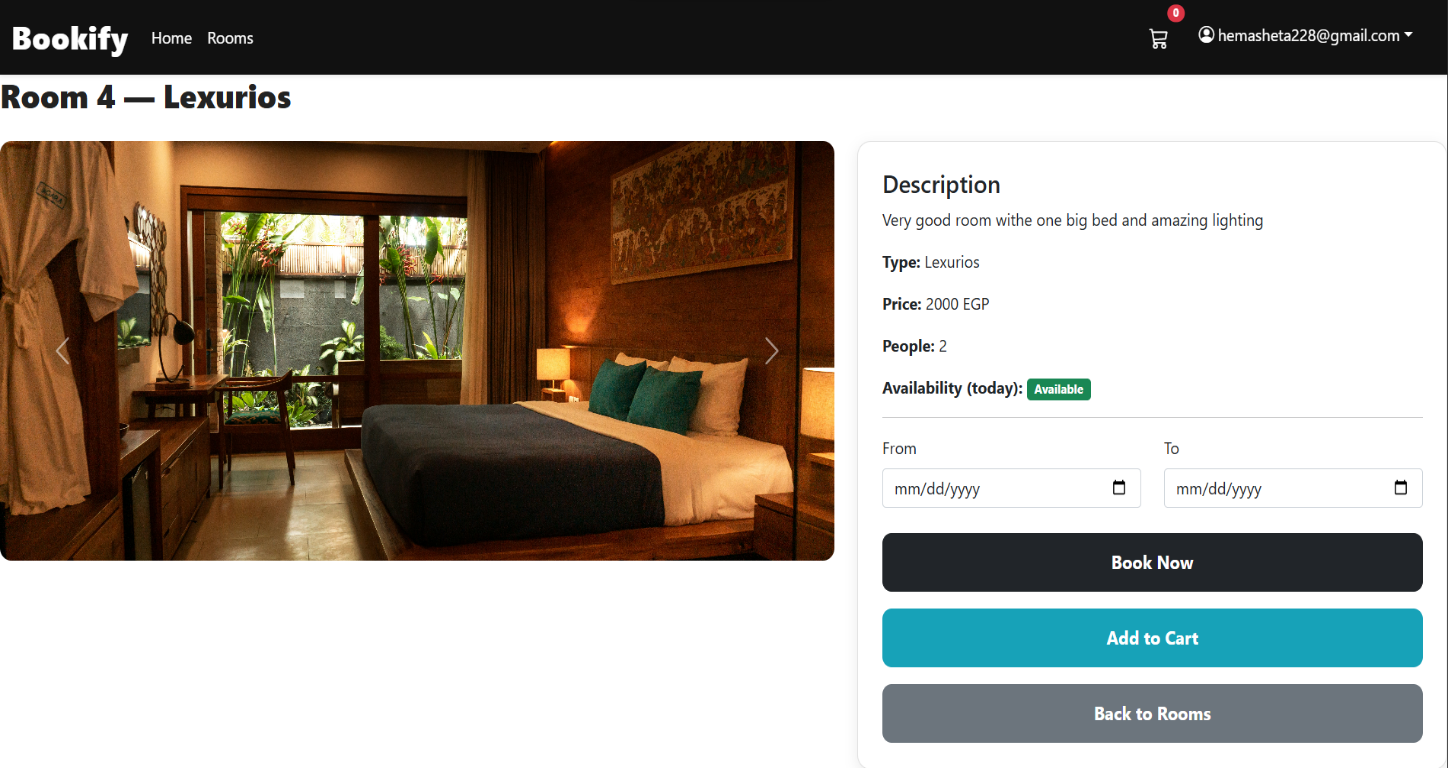
**Figure 27 , 28: editing room UI**

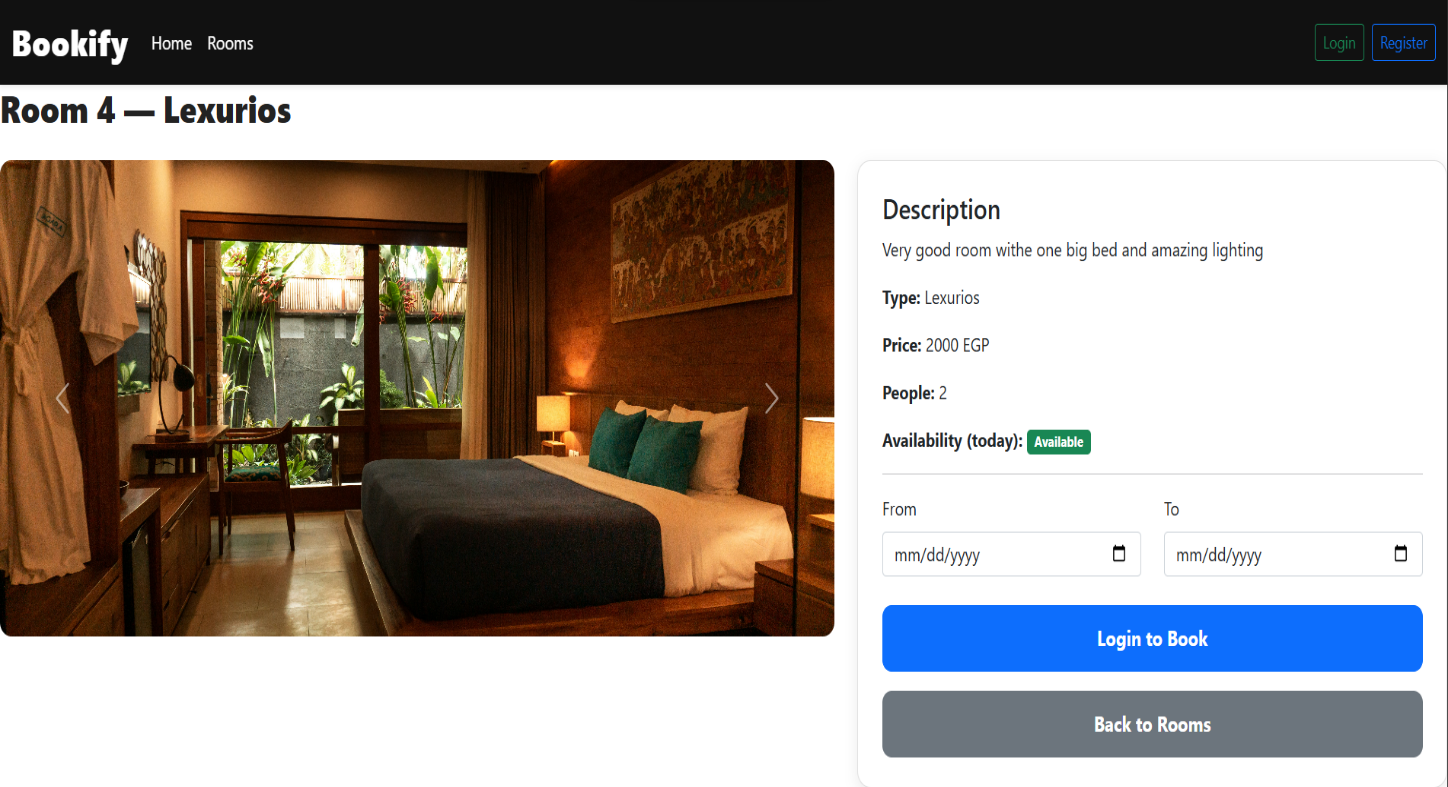
**Cart screen:** This page shows the user the selected rooms and the final price he will pay.



**Figure 29: cart UI**

**Room details screen :** The user can book his room through it and specify the date and duration.





**Figure 30 , 31: room details UI**





### Introduction

This chapter will cover the implementation side of the project such as: the tools and languages used to develop the system, most important codes, results of the project and system testing including unit tests, integration tests and usability tests.

### Tools and Languages

In *Help Me* application we used Android Studio [6] as an IDE (Integrated development environment)and Java as a language for developing the client side of the application, regarding the server side we used Hostinger [8] to host our server. For developing the server-side application and to manage the communications between the database and client side we used PHP language. We managed the database through php My Admin using MySQL language. Regarding the modeling, we used *Draw.io* online tool [3] to draw all the diagrams.

### Main/Most Important Codes

### 

**Admin add room type page**

This code is responsible for allowing the administrator to create a new type of room by adding the room

type name, price, and number of people for this room.

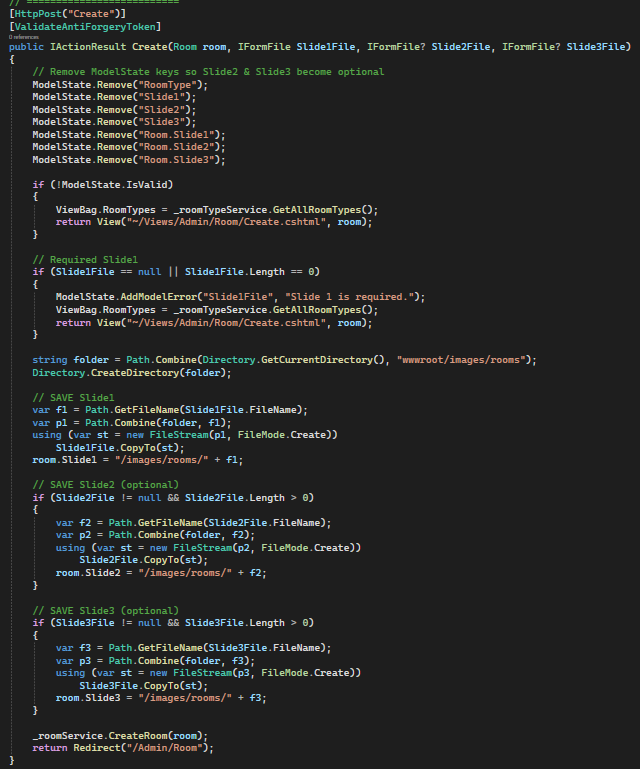


**Figure 28: Admin add room type page**

**Second:** **Admin add room**

This code is responsible for allowing the administrator to create a new room by adding the room number,

description, type, and uploading some pictures of it.



**Figure 29: Add New Request Method Code**

**Customer room page code**

It provides the user with all the necessary information to make a booking decision and enable them to

finalize the reservation for the selected room.

#### Its Main Functions:

1. **Display Details:** It offers a comprehensive view of the room, including multiple images, a written description, room type, price per night, and capacity.

**Show Availability:** It gives an instant indicator of today's availability (**Available**) and allows the user to check availability for specific durations by entering **"From"** and **"To"** dates.

**Facilitate Direct Booking:** The **"Book Now"** button allows the user to proceed directly to the payment

process for their selected dates.

**Support Multi-Booking Shopping:** The **"Add to Cart"** button allows the user to save this room to the cart

(as seen in the previous page) to continue browsing other rooms before finalizing payment for all reservations together.

****

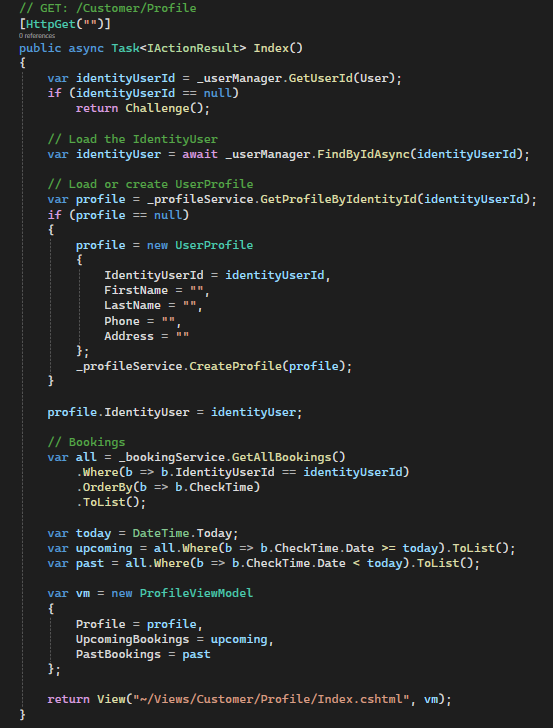
**Profile code**

The primary function of this page is **to collect the necessary information from the user to create a new profile**

**in the website's database**, which grants them the ability to book rooms and track their reservations.

* **Registration:** It asks the user to enter their **Email**, **Password**, and then **Confirm Password**, and then click the **"Register"** button to complete the account creation process.
* **Return to Login:** It provides an **"Already have an account? Login"** link to redirect the user to the login

page if they already possess an account.

****

**Profile edit code**

**To Identify the Identity**: The system first identifies the currently logged-in user.  
  
**Data Retrieval**: The system uses this identity to retrieve all personal details saved for the client (such as name,

phone number, and address) from the system records.  
**Display**: The retrieved data is placed in a modification form, so that the client sees their current information

and is ready to change whatever they want.  
**Security:** If the system cannot find a record for the client, it redirects them away from the edit page to their

main page.



**Cart – final booking code**

The function of this page is to **manage and confirm the process of booking rooms before payment.**

The "Reservation Cart" page is an essential step in the online booking flow, and its main functions are summarized

as follows:

**1. Reviewing and Modifying the Reservation**

* **Displaying Bookings:** It allows the user to review all the rooms they have chosen to book, including the room name, number, and a preview image.
* **Confirming Dates and Prices:** It displays details of the booked dates (Check-in $\to$ Check-out), the price

per night, and the number of nights.

* **Removing Items:** It provides a **"Remove"** button for each booking to delete a specific room from the cart if

the user changes their mind.

* **Emptying the Cart:** It offers a **"Clear Cart"** button to delete all added reservations at once.

**2. Summarizing the Cost**

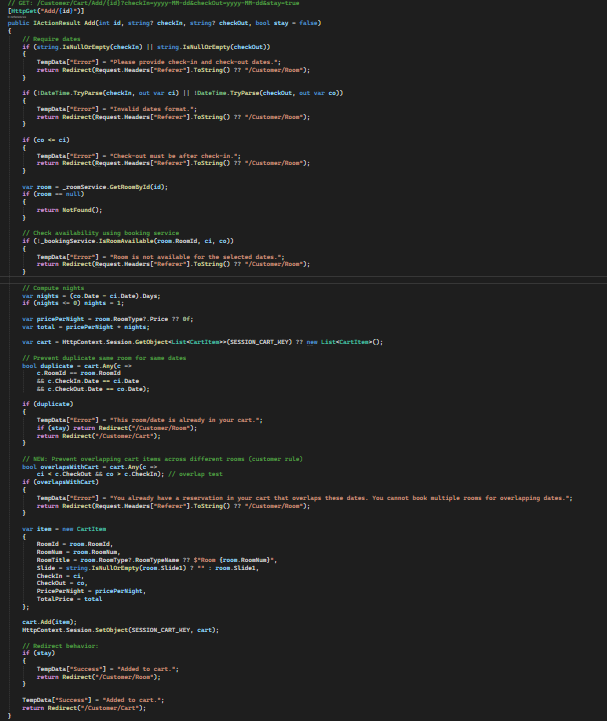
* **Calculating the Total:** It automatically calculates the overall total cost of all added reservations in the cart

(**Total: 4400.00 EGP**), which is the amount to be paid later.

**3. Proceeding to Finalize the Purchase**

* **Moving to Checkout:** It provides the **"Proceed to Checkout"** button, the most important function, which

moves the user to the next step to enter payment details and confirm the final reservation.

****

### System Testing

#### Unit testing:

We are going to address the unit tests of the system:

**Table 10: User Test Plan**

|  |  |
| --- | --- |
| **Unit Test plan (In scope Out of scope)** | |
| **In scope functions** | **Out of scope functions** |
| **Signup Login**  **Create new request** | Rest of system functions does not have multiple cases to be tested |

**Table 11: Signup Unit Test**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit function: Signup** | | |  |
| **Test case** | **Input value** | **Expected output** | **Pass / Fail** |
| **Enters valid data** | All entries are correct and matches the syntax | System creates a new profile record in the database with the  data input | pass |
| **Enters invalid data** | Entries dose not match the syntax or required field left empty | System does not allow the user to proceed and ask him to enter valid  inputs | pass |

**Table 12: Login Unit Test**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit function: Login** | | |  |
| **Test case** | **Input value** | **Expected output** | **Pass / Fail** |
| **Enters valid data** | Enter a correct email and password | System allows  access to the next page | pass |
| **Enters invalid data** | Enters a wrong email or a wrong password | System show that the email or the password are  incorrect and stays in the Login  page | pass |

**Table 13: Create New Request Unit Test**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit function: create new request** | | |  |
| **Test case** | **Input value** | **Expected output** | **Pass / Fail** |
| **Enters valid data** | User enters all required fields correctly | System creates  new request  record to the database | pass |
| **Enters invalid data** | User leaves some required fields empty | System shows an error massage and asks the user to fill in the required  fields | pass |

#### System scenarios:

**First, User (Customer) Scenario:**

After the user registers or logs in with their email and password (user data includes number, name, and email),

they are directed to the home screen. The user can browse available rooms, search/filter them, add to cart, and

proceed to booking with payment.

**1. Login/Register:** The user opens the app and sees a splash screen. They tap "Login" or "Register". For

register, they enter name, email, password, and phone number. For login, they enter email and password.

Upon success, they are redirected to the home screen (dashboard showing available rooms).

**2. View Available Rooms**: On the home screen, the user sees a list of available rooms with images, types,

prices, and capacities (e.g., "Deluxe Room - $100/night - Up to 2 people"). Rooms are displayed in a grid or

list view with scrollable cards. Micro-interaction: Rooms fade in with a subtle animation as they load.

**3. Search for Rooms:** The user taps a search bar at the top. They can enter a room number (e.g., "101") or

type (e.g., "Suite"). The app filters the list in real-time, highlighting matching rooms. If no matches, a "No

results" message appears with a suggestion to adjust filters.

**4. Filter Rooms:** The user taps a filter icon (e.g., funnel). A pop-up or side panel appears with filters:

- Available dates (calendar picker for check-in/check-out).

- Price range (slider).

- Room type (dropdown: e.g., Standard, Deluxe, Suite).

- Number of people (dropdown: 1-4).

They apply filters, and the room list updates dynamically. Micro-interaction: Filters animate in/out smoothly.

**5. Add to Cart:** The user selects a room card and taps "Add to Cart". A confirmation toast appears (e.g.,

"Room added!"). They can add multiple rooms. Tapping the cart icon (bottom nav) shows the cart screen with

selected rooms, quantities, and total price. They can edit quantities or remove items. Micro-interaction: Cart

items slide in when added.

**6. Proceed to Booking:** In the cart, the user taps "Book Now". A booking screen appears where they select

check-in/check-out dates (calendar picker), confirm details, and proceed to payment. If dates conflict,

an error message shows (e.g., "Room unavailable on selected dates").

**7. Payment and Confirmation:** The user enters payment details (e.g., card info or integrated payment

gateway like Stripe). Upon successful payment, a confirmation screen appears with booking ID, details, and

a "View Bookings" button. They receive an email/SMS notification. Micro-interaction: Payment

progress bar animates during processing.

**8. Logout:** From any screen, the user taps the profile icon and selects "Logout", returning to the login screen.

Throughout, the app ensures accessibility (e.g., high contrast for colors, screen reader support) and

responsiveness (adapts to phone/tablet sizes).

**Second, Admin Scenario:**

After the admin logs in with email and password, they access a dashboard to manage room types,

rooms, and bookings. Admins can add/edit/delete everything, including login/register/logout for their account.

**1. Login/Register**: Similar to user, but admin accounts are pre-created or registered via a special code. Upon

login, they are directed to the admin dashboard.

**2. View Dashboard:** The dashboard screen shows an overview: total bookings, revenue, and quick links to

manage rooms/types/bookings. It's a tabbed interface (e.g., Rooms, Types, Bookings) with charts or lists.

**3. Manage Room Types**: The admin taps "Room Types" tab. They see a list of existing types (e.g.,

"Standard - $50/night - Up to 2 people"). Tapping the plus (+) button opens a form pop-up: Enter name,

price, and max people. They submit to add. For editing, tap a type to modify fields and save. Delete via a

trash icon with confirmation. Micro-interaction: Forms slide in/out.

**4. Add/Edit Rooms:** In the "Rooms" tab, the admin sees a list of rooms with details. Tapping plus (+) opens

a form: Enter room number, select type (from dropdown), description, and upload up to 3 images. Submit to

add. Edit by tapping a room card, updating fields/images. Delete with confirmation. Micro-interaction: Image

uploads show progress bars.

**5. View and Manage Bookings:** In the "Bookings" tab, the admin sees a list of all system bookings (user

name, room, dates, status). They can tap to view details, edit (e.g., change dates/status), or delete. Filters/

search by date/user/room. Micro-interaction: List items expand on tap for details.

**6. Logout:** Tap profile icon and "Logout", back to login.

The admin dashboard is secure, with role-based access (only admins see these features). All changes sync

in real-time, and the app logs actions for auditing. Accessibility and responsiveness are maintained, with bold

colors for a lively feel (e.g., green for confirmations, red for deletes





In summary, this hotel room booking app delivers a seamless, user-friendly experience for both customers and admins, emphasizing ease of navigation, real-time updates, and secure transactions.

For users, the flow from browsing and filtering rooms to booking and payment ensures quick,

enjoyable interactions with vibrant UI elements and micro-interactions. Admins benefit from a

powerful dashboard for efficient management of rooms, types, and bookings, all while maintaining

data integrity and accessibility standards.

This design promotes a lively, colorful interface that reflects a modern hotel brand, with responsive layouts for mobile devices