

---

# **Final Design Document**

**for**

## **<Hotel Management System>**

**Prepared by <Hoang Dang Khai>**

**<Vu Viet Hung>**

**<Tran Duy Tuan Anh>**

**<Le Van Luong>**

**<Nguyen Duy Anh Quoc>**

**<University of Engineering and Technology>**

**11-05-2024**

# Table of Contents

I. System architecture:	2
II. Sequence diagram	6
III. Class diagram	11
IV. User interface design:	18
V. Database design:	40

## I. System architecture:

### 1) Domain Driven Design (DDD) Layered Architecture

This system is built with Domain Driven Design (DDD), there are 4 main layers:

- **Presentation Layer** (User Interface Layer): This layer is used to display information to the user and receive new data. It can be done by Web application, Desktop or by any presentation technology. This layer depends on the Application layer.
- **Application Layer**: This layer contains business logic. It is responsible for monitoring and coordinating application activities that are executed on the Domain. Does not contain logic or business, does not store the state of the business. This layer depends on the Domain layer.
- **Domain Layer**: This layer is the heart of the software business. Businesses and logic are placed in this layer. Entity (Class) states and operations that are used to describe operations are placed here. This is information exchanged with different systems, or describes objects that can be saved to a database, or used to forward to an infrastructure layer such as a repository.
- **Database Layer** (Infrastructure Layer): This class acts as a support library for all current classes. It provides communication between layers, implements business persistence for business objects.

Domain-Driven Design (DDD) brings benefits and solves a lot of system problems of Hotel Management System:

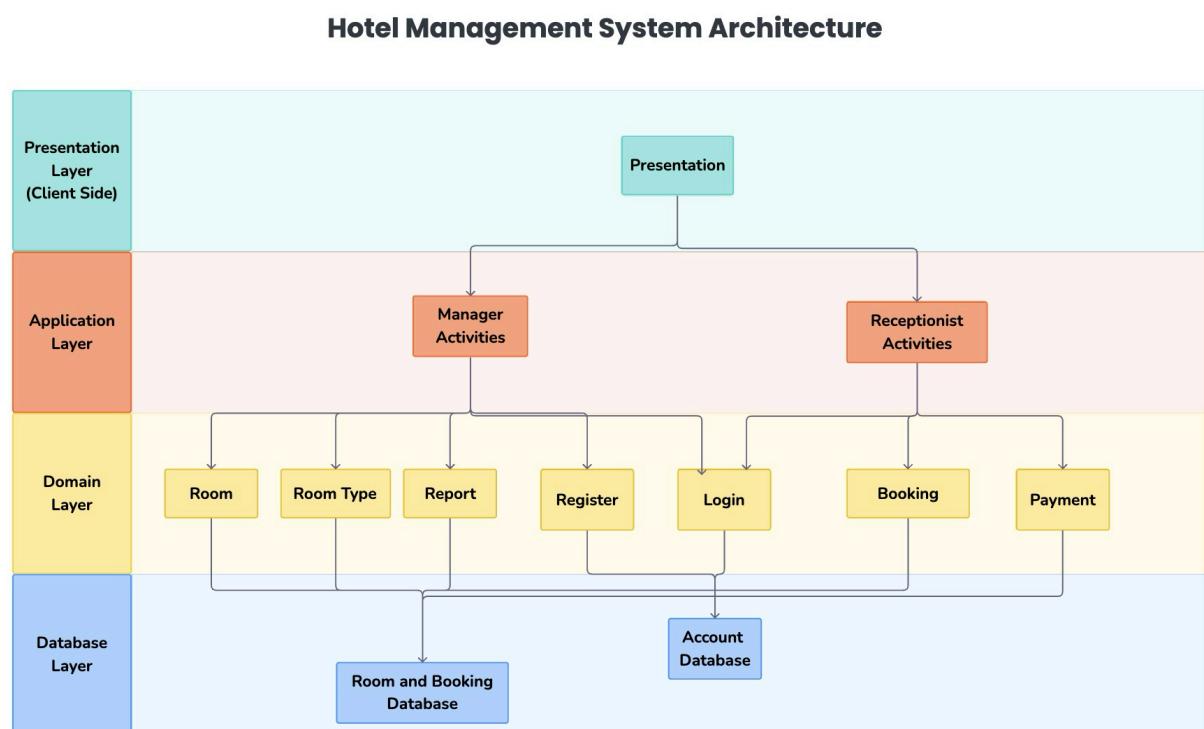
- Separation of Concerns: Layered Architecture helps separate different concerns within the application. In the context of a Hotel Management System, this separation allows us to organize our code into distinct layers, each with a specific purpose.
- Scalability and Modularity: Layered Architecture enables independent development and deployment of different layers. If you need to enhance or replace a specific layer (e.g., UI or data access), it won't impact the entire system. This flexibility is crucial for maintaining and evolving

- large applications.
- Enforcement of Restrictions: By enforcing communication restrictions between layers, we achieve encapsulation. When a layer changes or is replaced, only the affected layers need to be updated. This minimizes the ripple effect across the entire application.
- Reduced Dependency and Easier Testing: Layered Architecture reduces dependencies between components. Each layer has a well-defined responsibility, making it easier to modify and maintain the application over time. Testing becomes simpler because individual components can be tested independently.

Domain-Driven Design Layered Architecture promotes modularity, scalability, and separation of concerns, making it a valuable choice for building complex systems like Hotel Management System.

## 2) Architectural Model:

Domain Driven Design (DDD) Layered Architecture



### **3) Domain Component:**

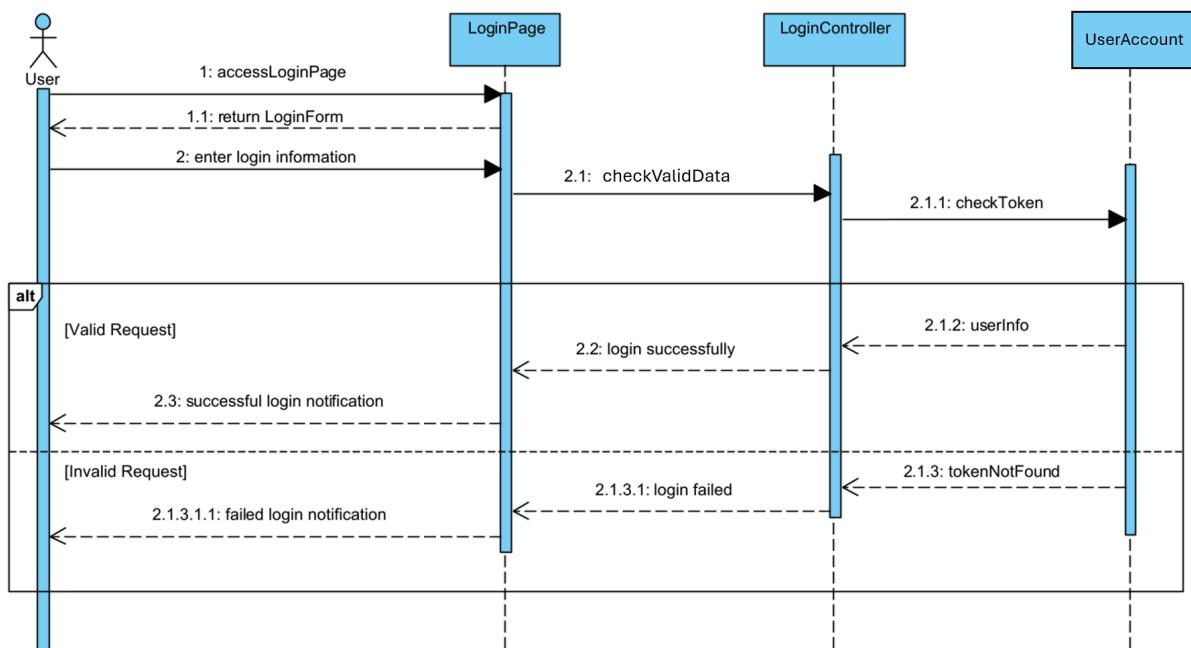
Name	Description
Login	<p>The Login component in a Hotel Management System often plays an important role in controlling access to system functions and data. The Login component allows users to enter their credentials, including username and password, to authenticate their identity and access the system. Users are manager and receptionist. The Login component typically includes a form that allows users to enter their login information. It can also provide options like “Forgot Password” to help users recover their login information if needed. The Login component needs to be secure to prevent unauthorized access. This is typically done through encrypting credentials and allowing only valid access. Furthermore, it maintains user sessions and permissions, enabling secure access to platform features and resources while protecting against unauthorized access.</p>
Register	<p>The Registration component in the Hotel Management System is responsible for creating a new account for the user. The Registration component allows the user to create a new account by providing necessary information such as name, date of birth , phone number, account, password, email. The user can be a new customer or a new employee who needs to create an account to access the system. The Registration component typically includes a form that allows users to enter their registration information. It may also provide a “Sign up by email” option to create an account. The Registry component needs to be secure to prevent unauthorized access. This is typically done through encrypting registration information and allowing only valid access.</p>
Room Type	<p>The Room Type component in the Hotel Management System plays an important role in managing and classifying hotel room types. This component provides a list of room types and their information,</p>

	allowing managers to create new room types, edit information about room types, and delete room types.
Room	The Room component in the Hotel Management System plays an important role in managing and classifying hotel rooms. This component provides a list of rooms and their information, allowing managers to create new rooms, edit information about rooms, and delete rooms.
Booking	The Booking component in the Hotel Management System plays an important role in managing and tracking hotel reservations. The Reservations component allows users to book rooms, services, and amenities. It also allows us to manage bookings, prices, customer information and payments. The user is a hotel receptionist who needs to manage reservations. The Booking component allows users to search for available rooms, view room information, book and pay online. The Booking component needs to be secure to prevent unauthorized access. This is typically done through encrypting payment information and allowing only authorized access.
Payment	The Payment component in the Hotel Management System plays an important role in processing payment transactions. The Payments component allows users to make payments for services they have used, including rooms, room service, restaurant services, and others. It also allows hotels to add incidental charges to the final bill and quickly generate invoices and receipts. The user is a hotel receptionist who needs to process payment transactions. The Payment component includes an intuitive interface that allows users to make payments online or at the reception desk, allowing for extraction and printing of invoices. The Payments component needs to be secure to prevent unauthorized access. This is done through encrypting payment information and allowing only valid access.
Report	The Report component in the Hotel Management System plays an important role in providing detailed and up-to-date information about the system's

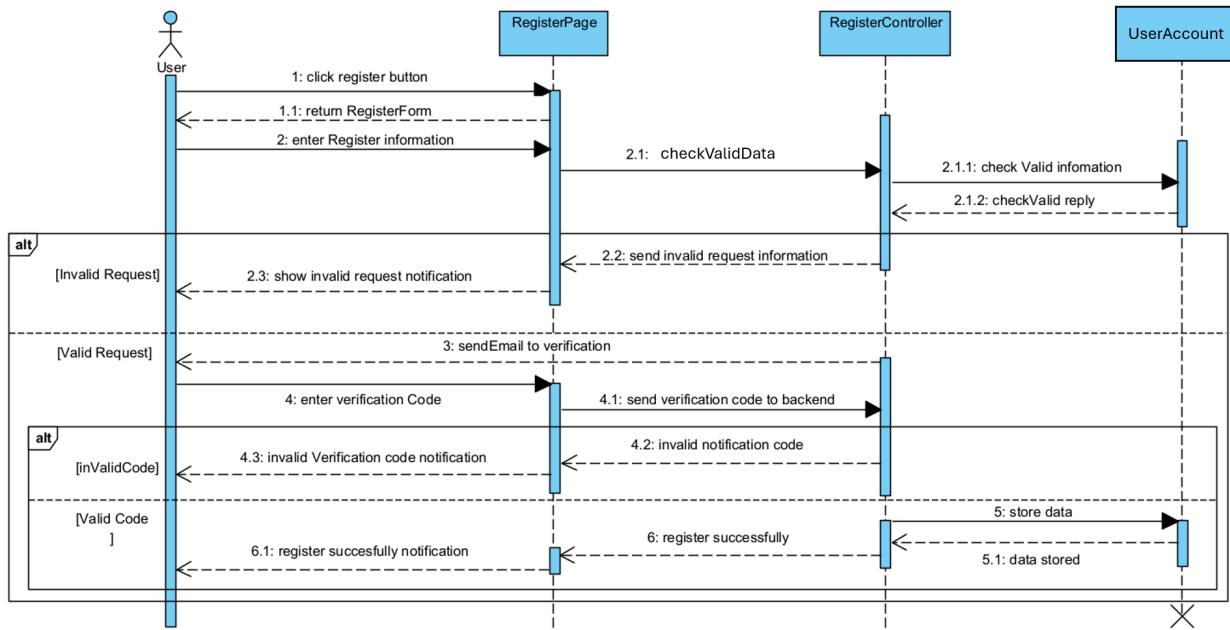
performance. The Report component allows users to generate detailed reports on various activities in the system, including room management, reservations, and other services. This helps in monitoring performance and managing resources effectively. The user is a manager and needs to analyze data and information from the system. The Report component includes a user-friendly interface that allows users to select different reporting criteria, and generate reports based on these criteria. It can also provide options like “Export Report” to help users save and share reports. The Report Generation component needs to be secure to prevent unauthorized access to sensitive data and information. This is typically done through controlling access and allowing only users with valid permissions to create and view reports.

## II. Sequence diagram

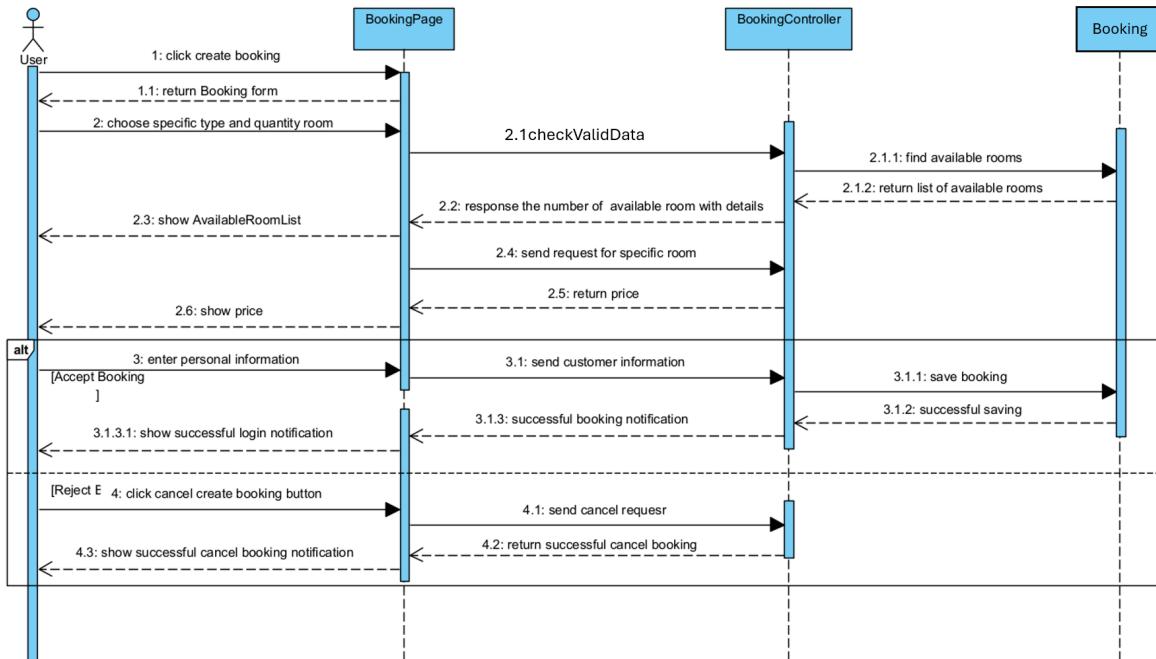
### 2.1 Login



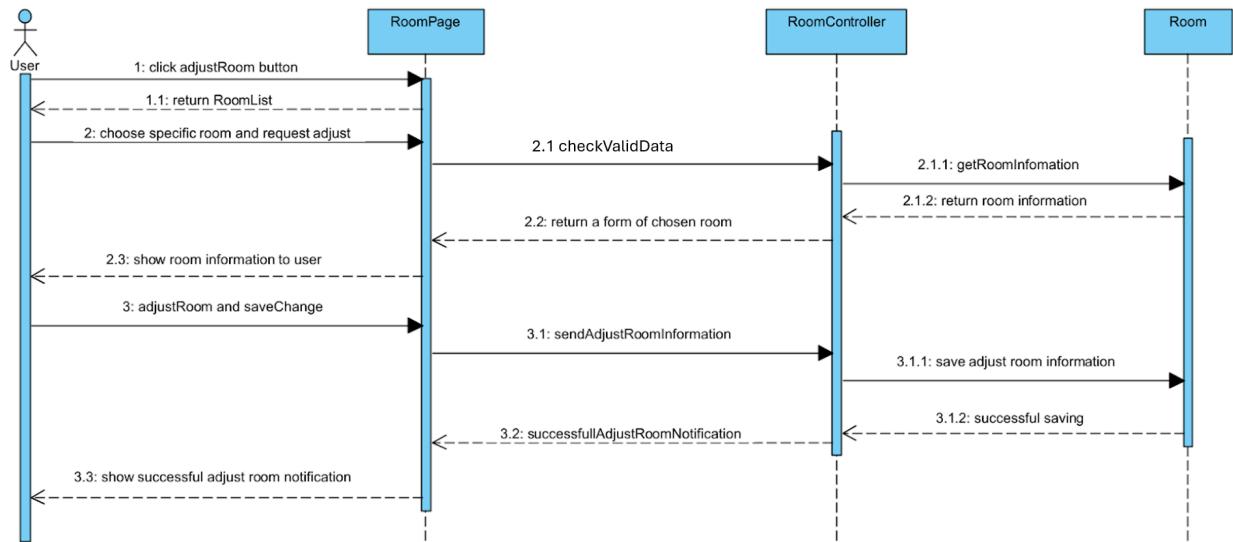
## 2.2 Register



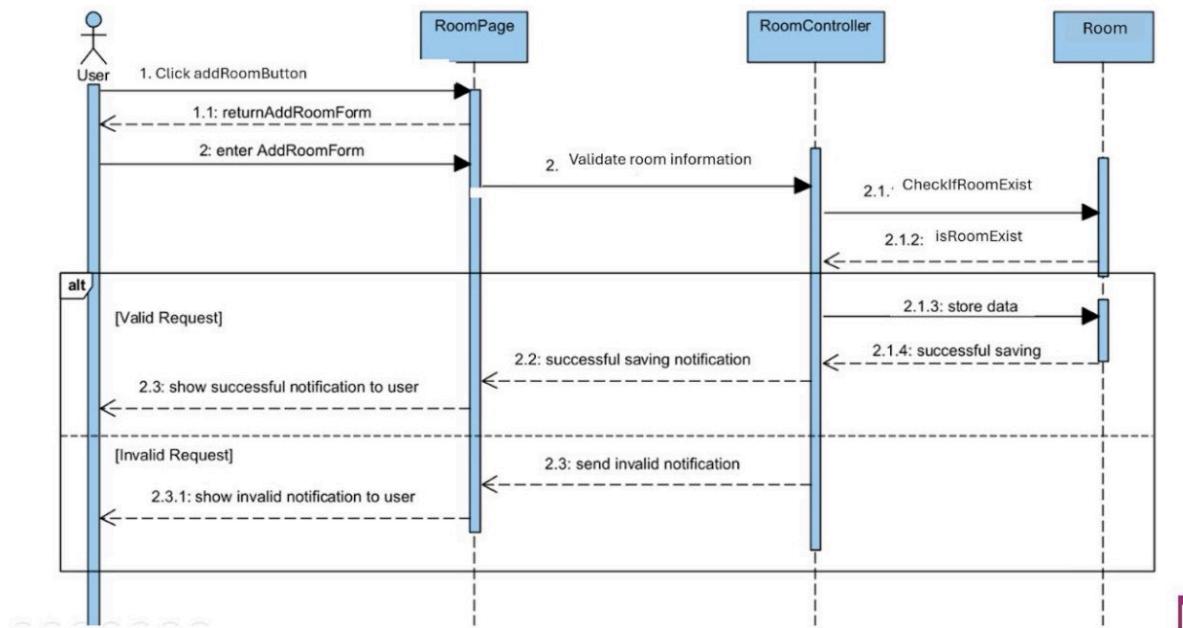
## 2.3 Create Booking



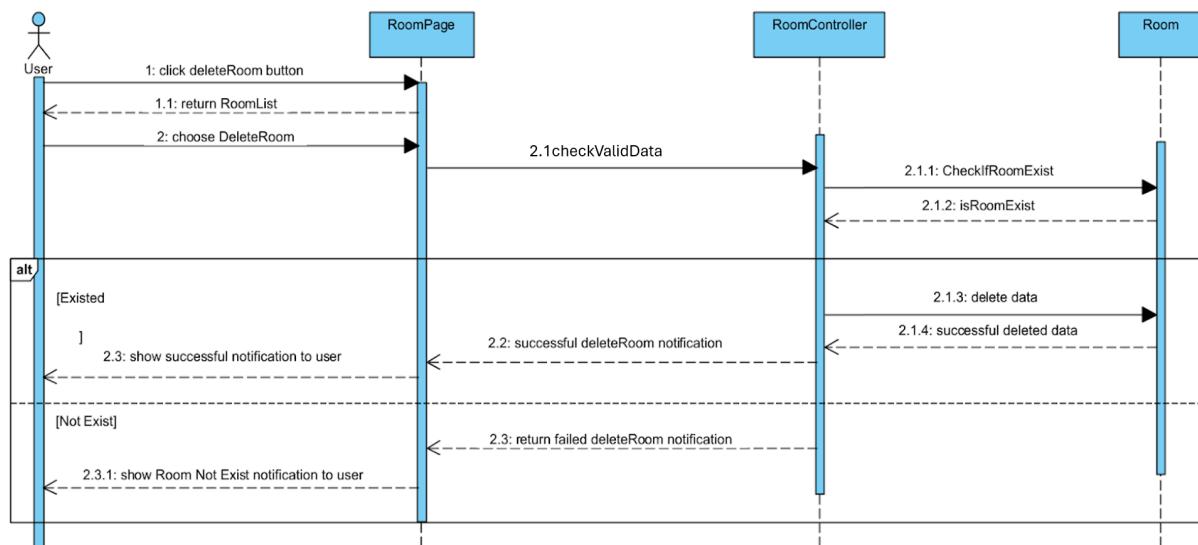
## 2.4 Adjust Room



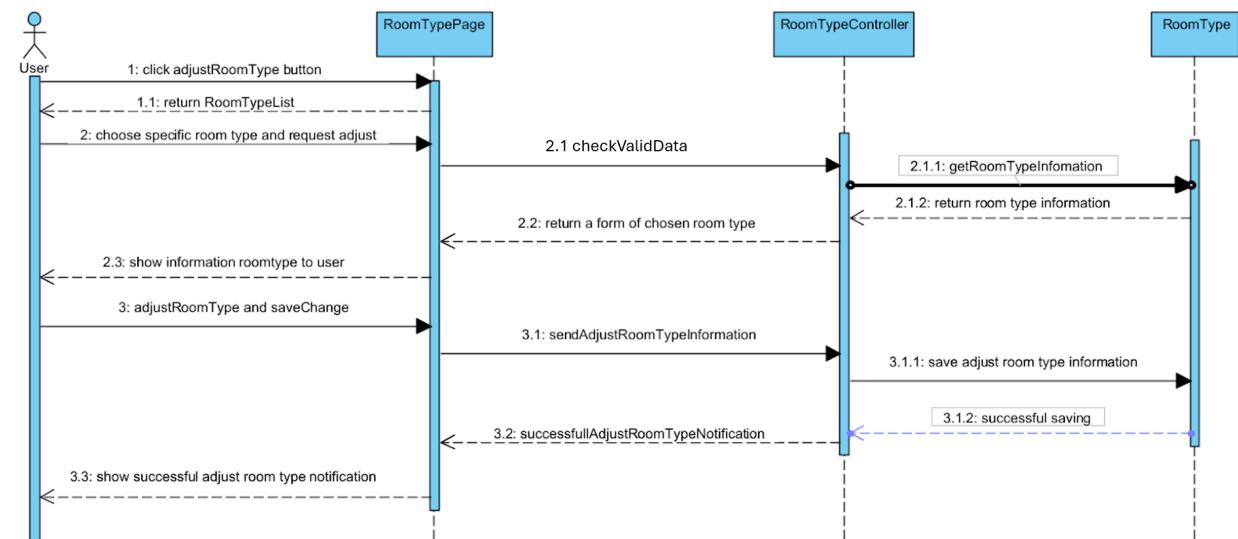
## 2.5 Add Room



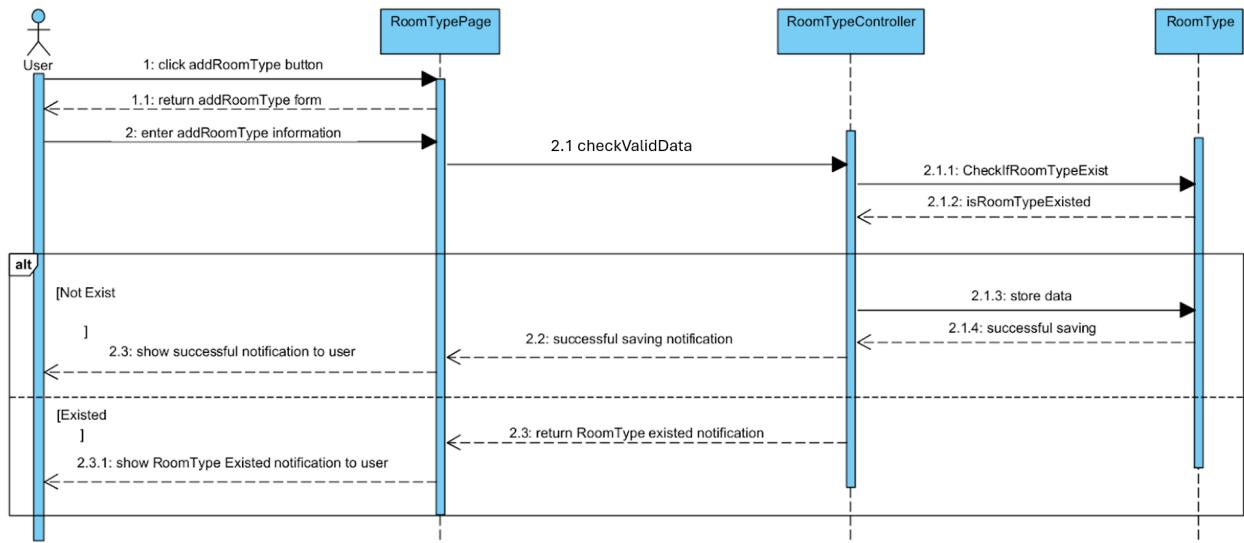
## 2.6 Delete Room



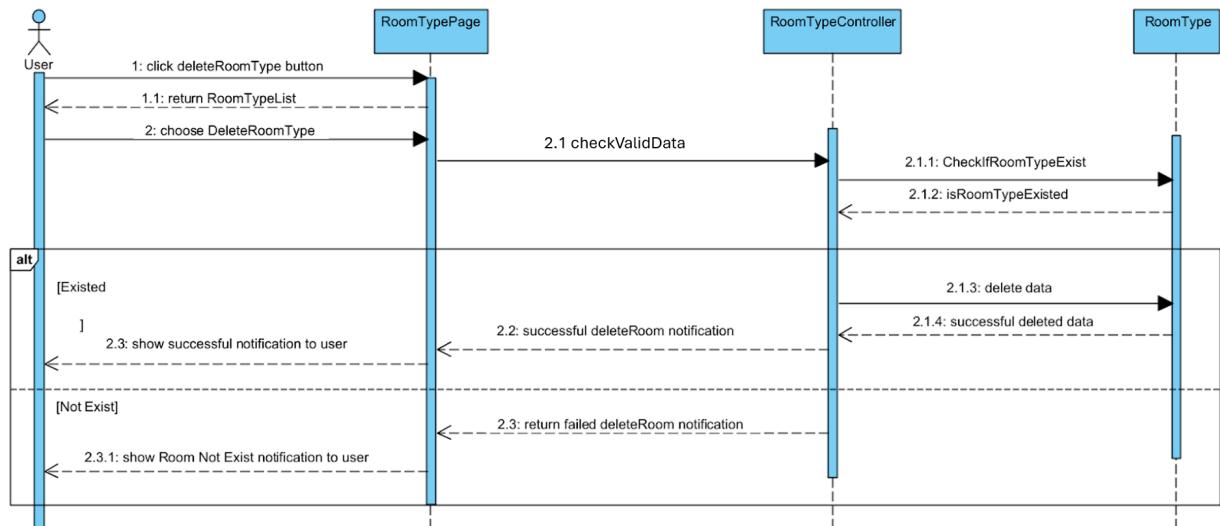
## 2.7 Adjust Room Type



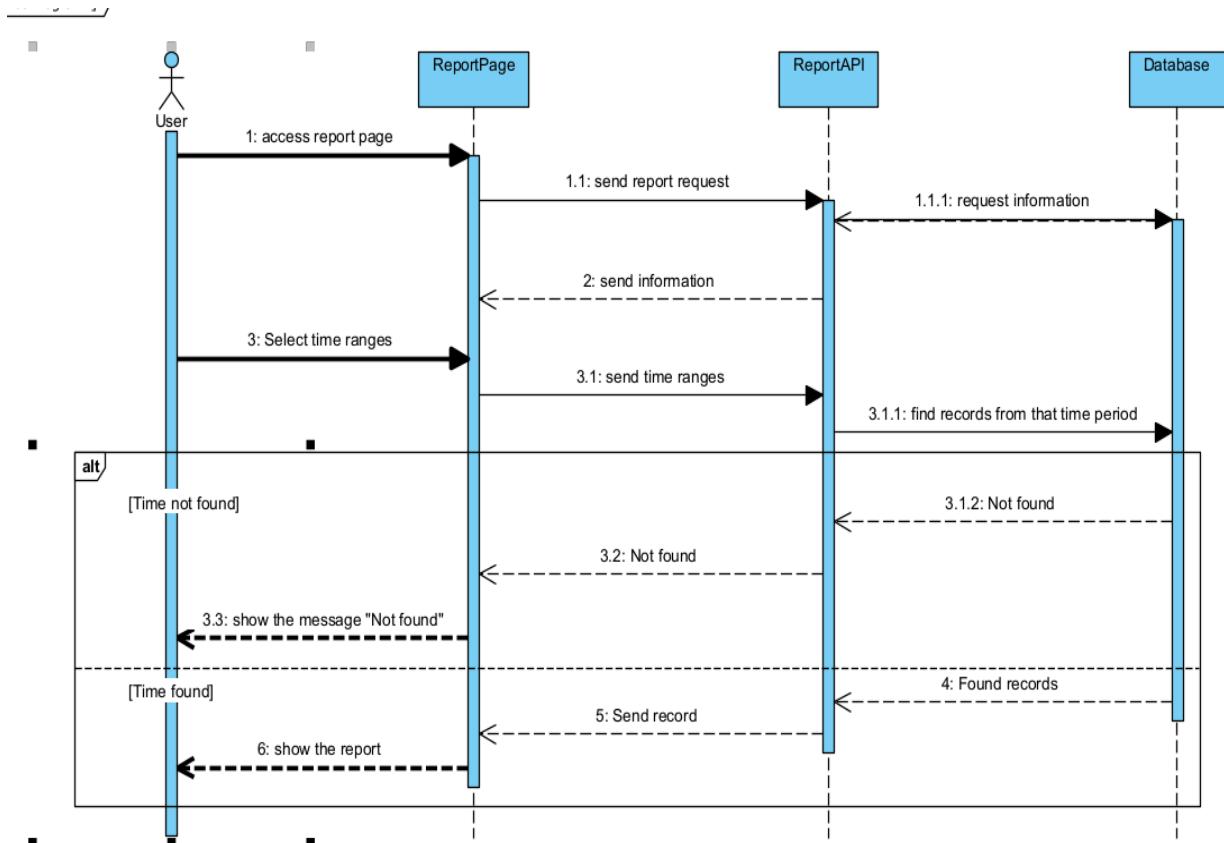
## 2.8 Add RoomType



## 2.9 Delete Room Type

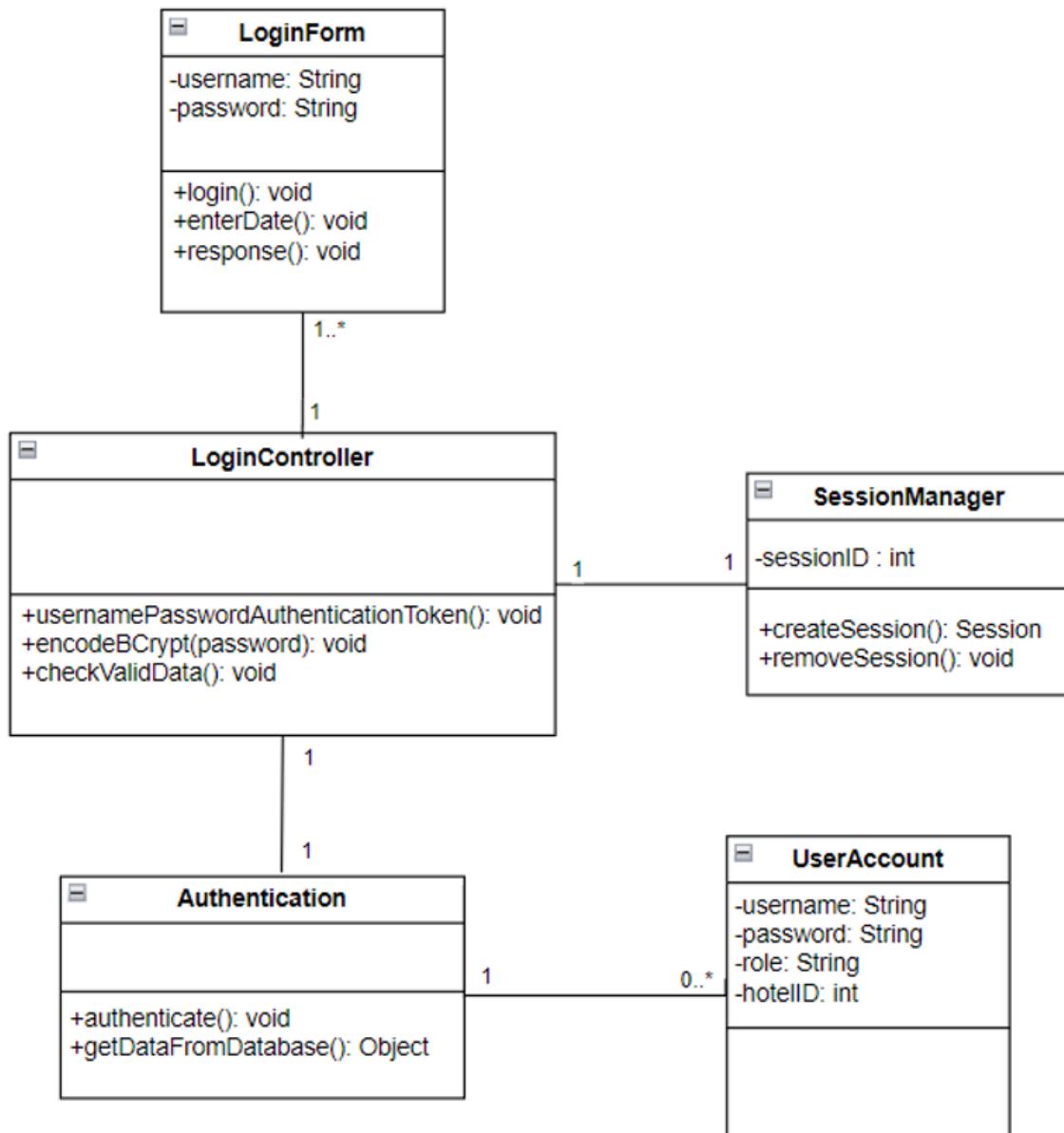


## 2.10 Generate Report

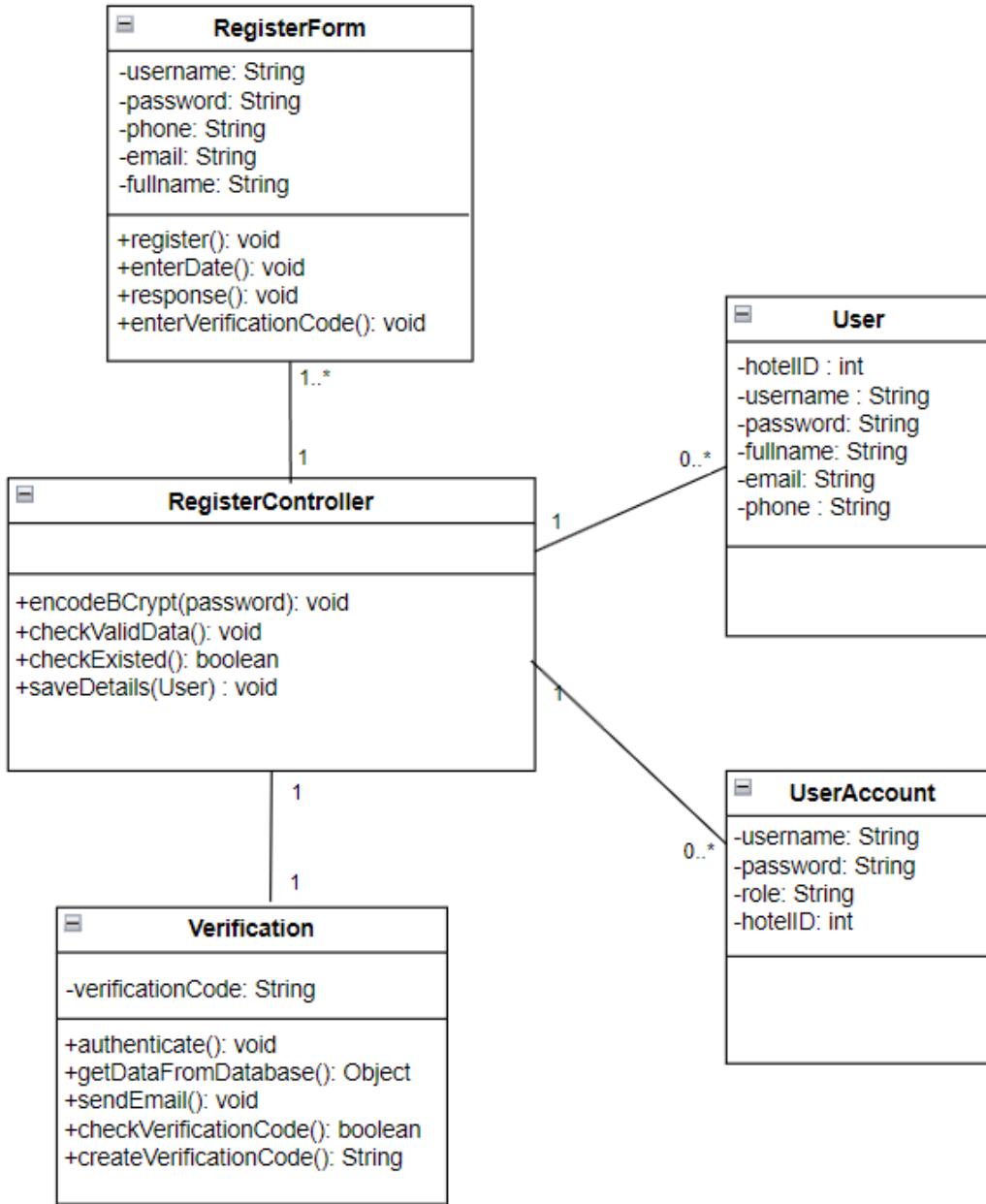


## III. Class diagram

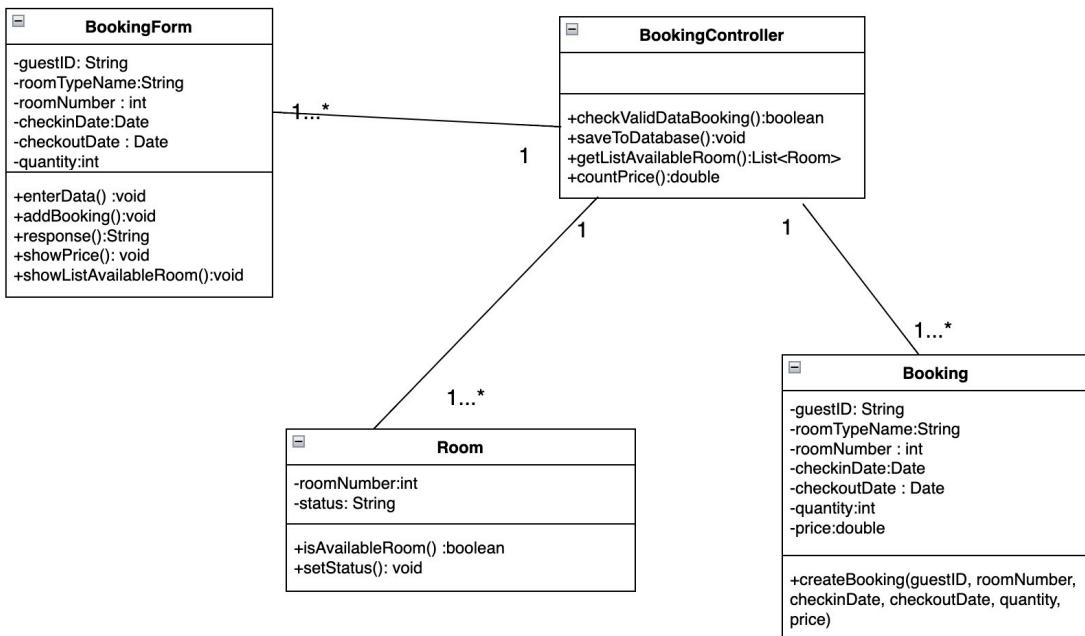
### 3.1 Login



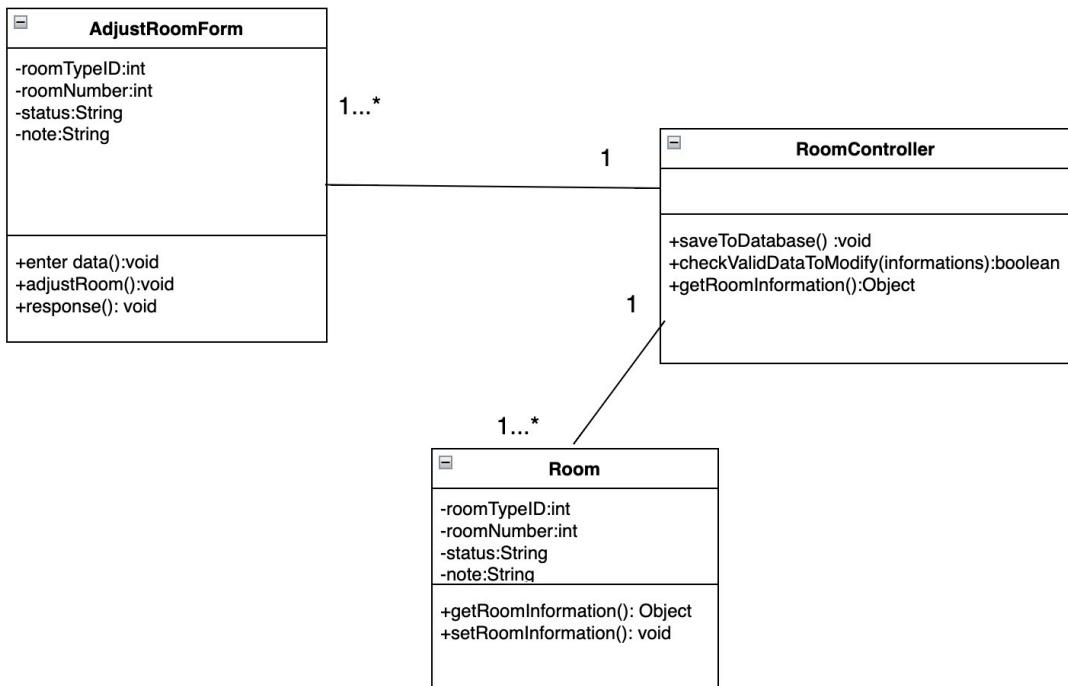
### 3.2 Register



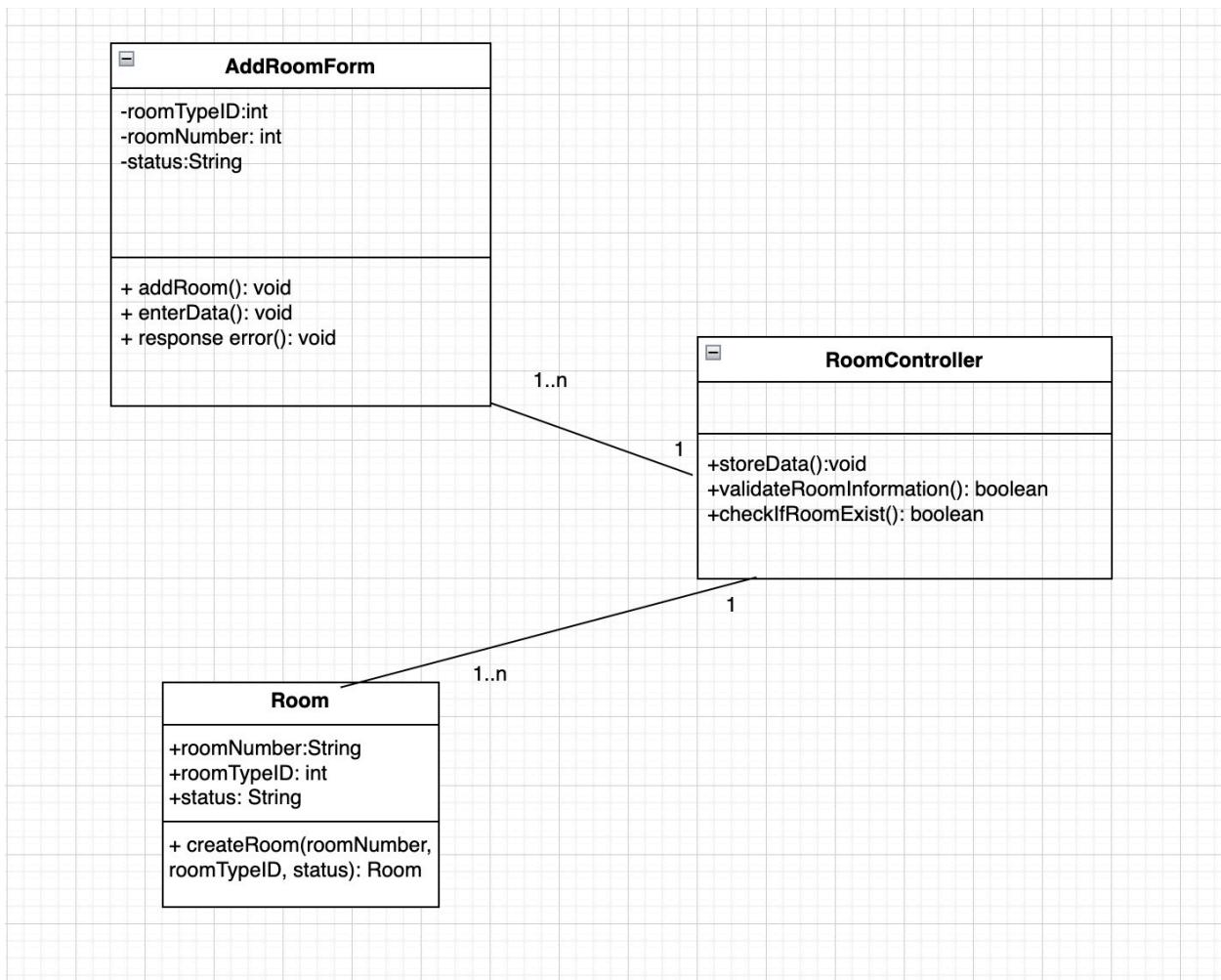
### 3.3 Create booking



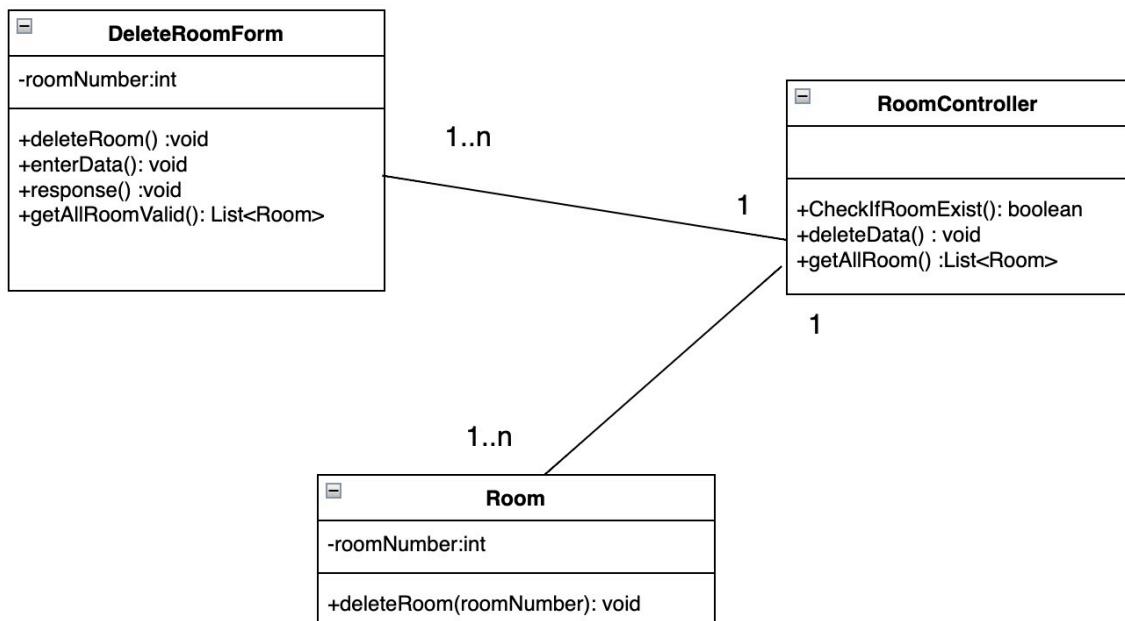
### 3.4 Adjust room



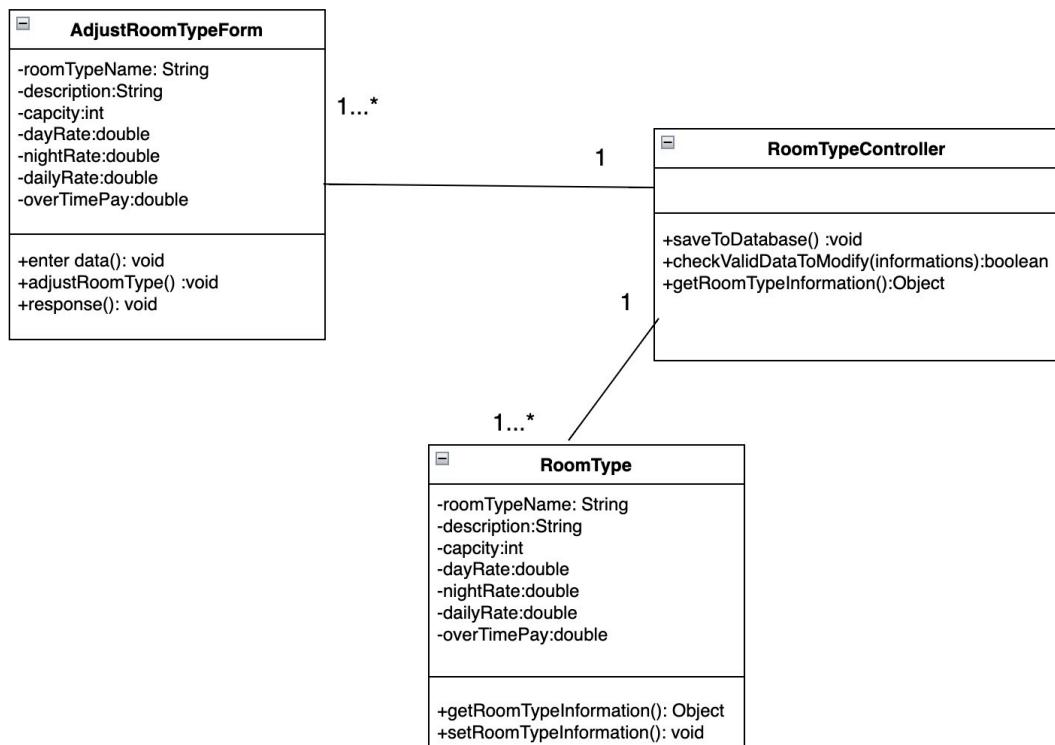
### 3.5 Add room



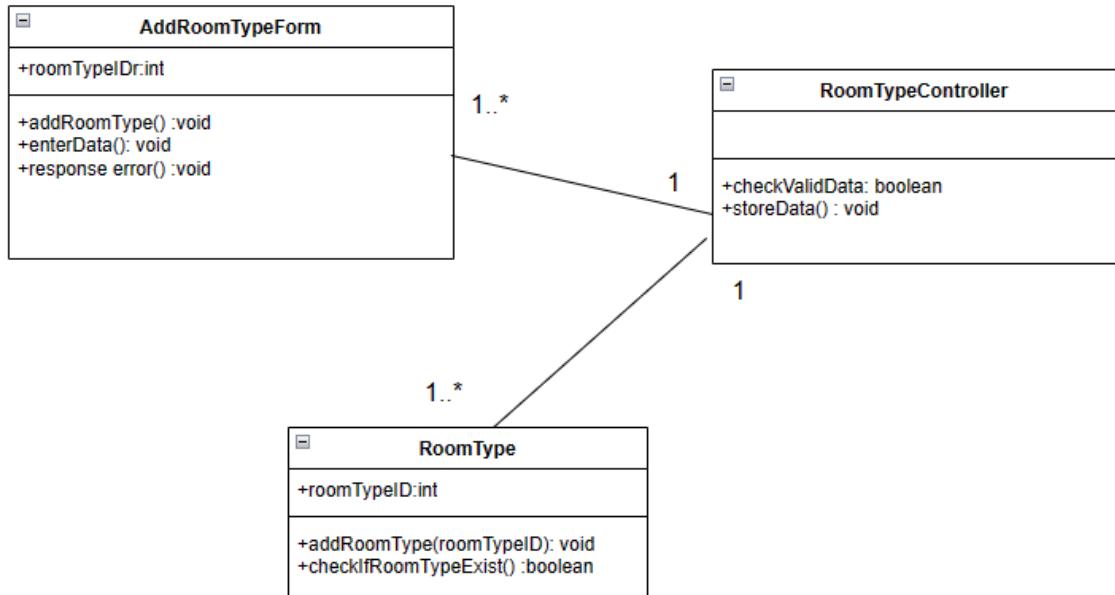
### 3.6 Delete room



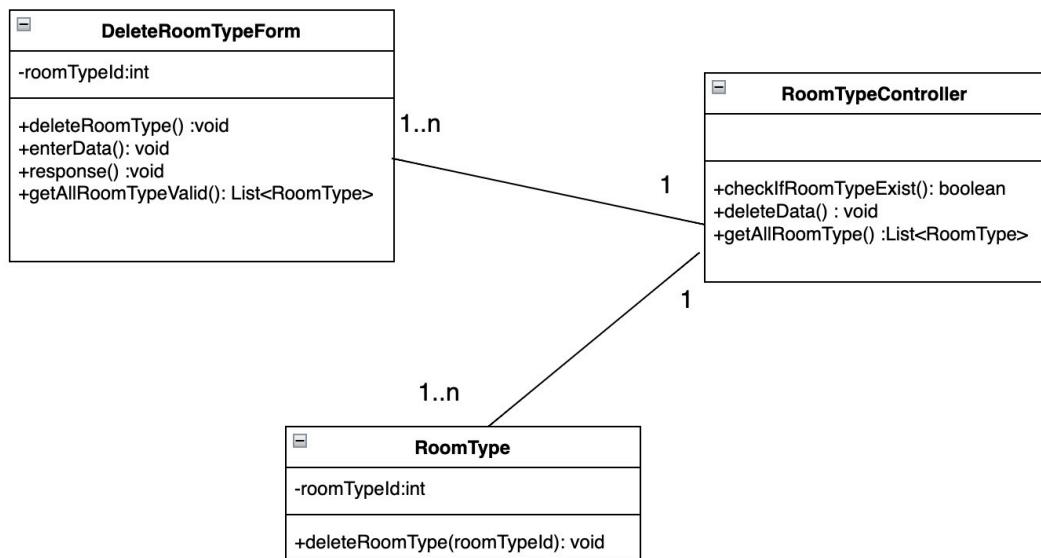
### 3.7 Adjust Room Type



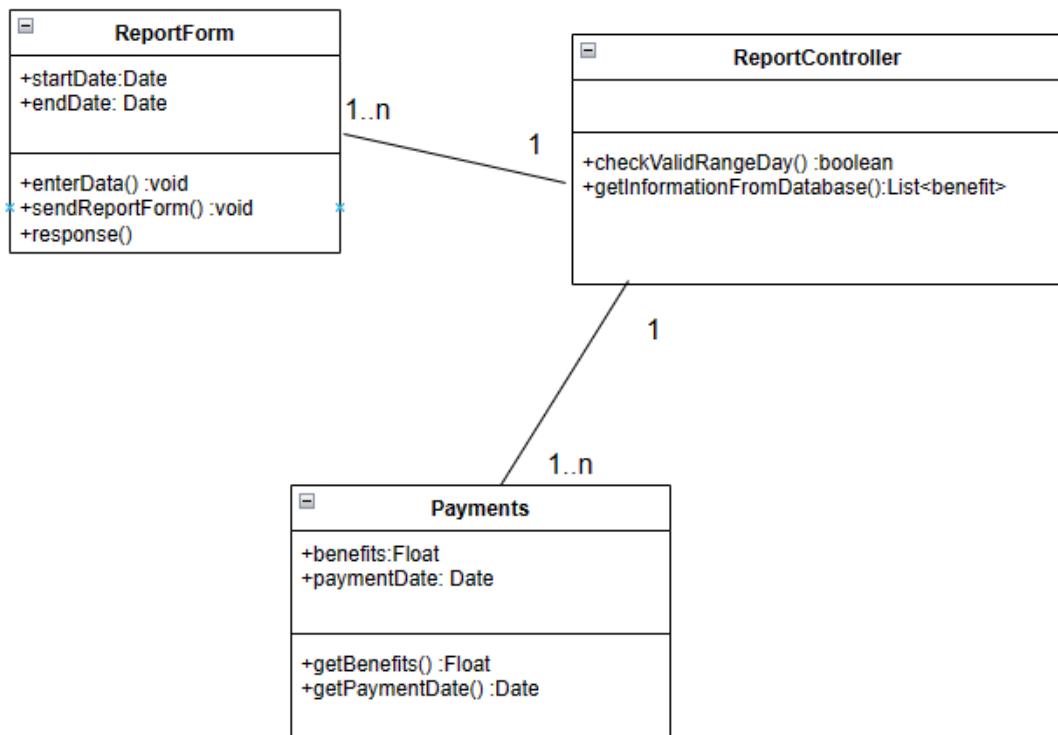
### 3.8 Add RoomType



### 3.9 Delete RoomType



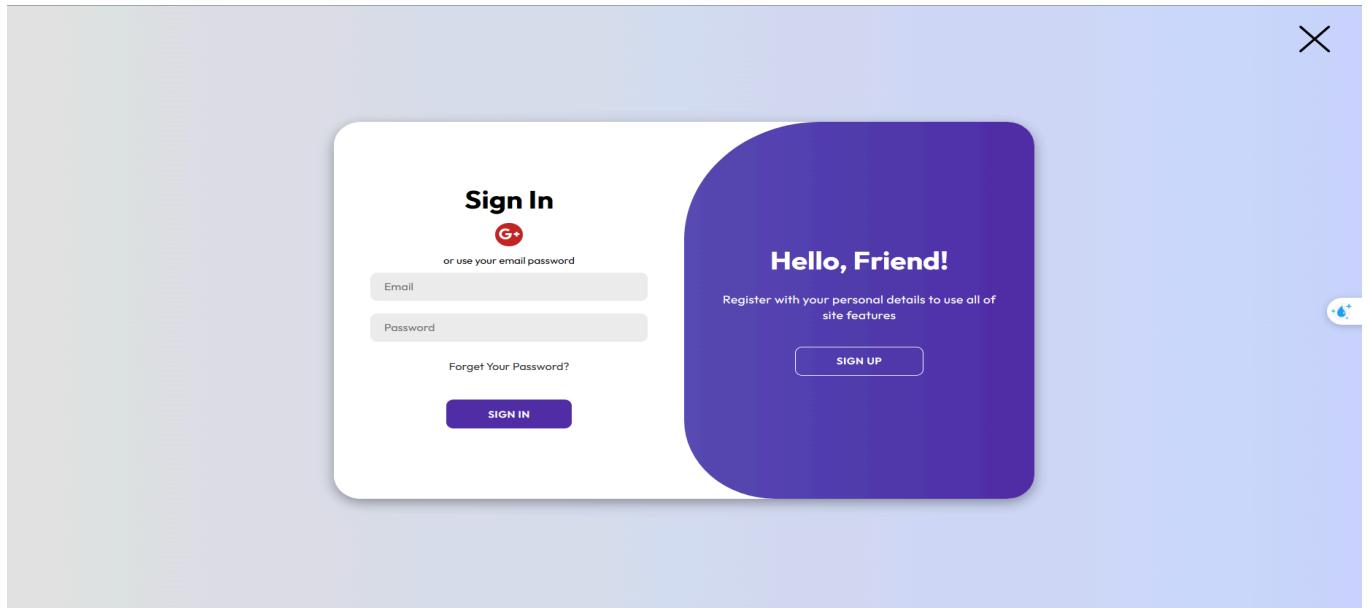
### 3.10 Generate Report



## IV. User interface design:

## 4.0 Login and Register

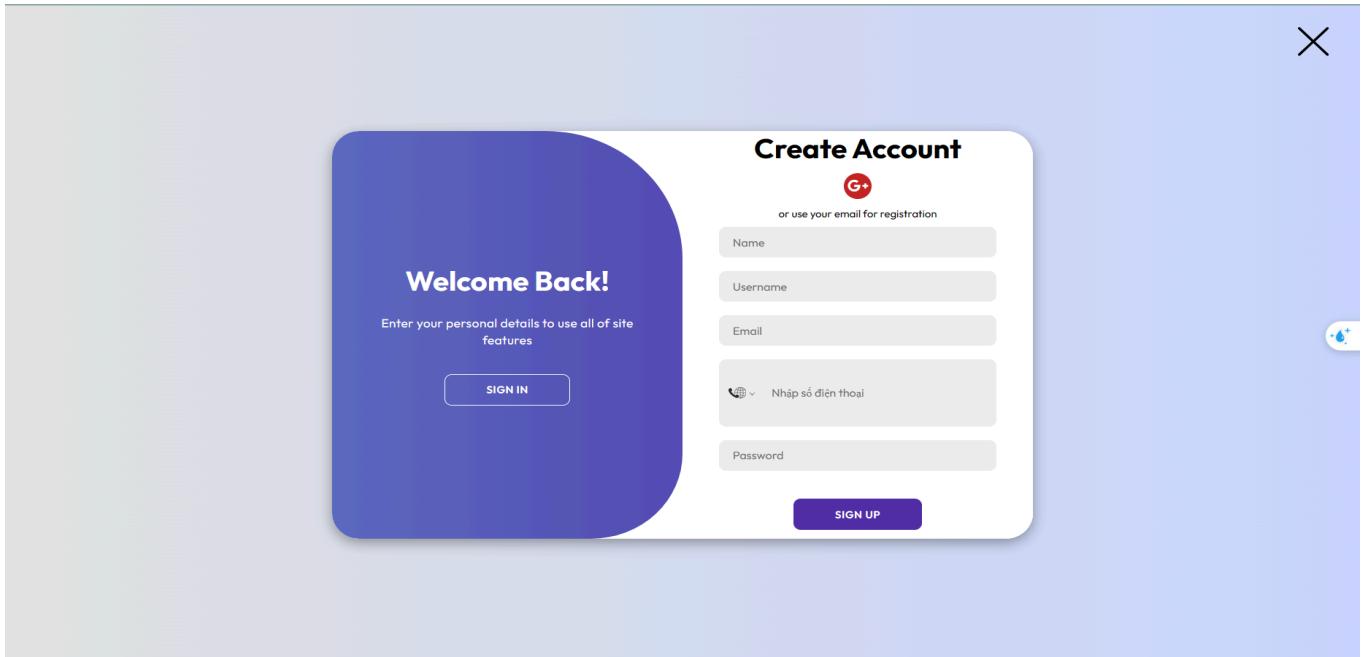
The Sign In interface provides 2 fields. The first field is the email and the second field is the password to log in



*Illustration of the Sign In interface*

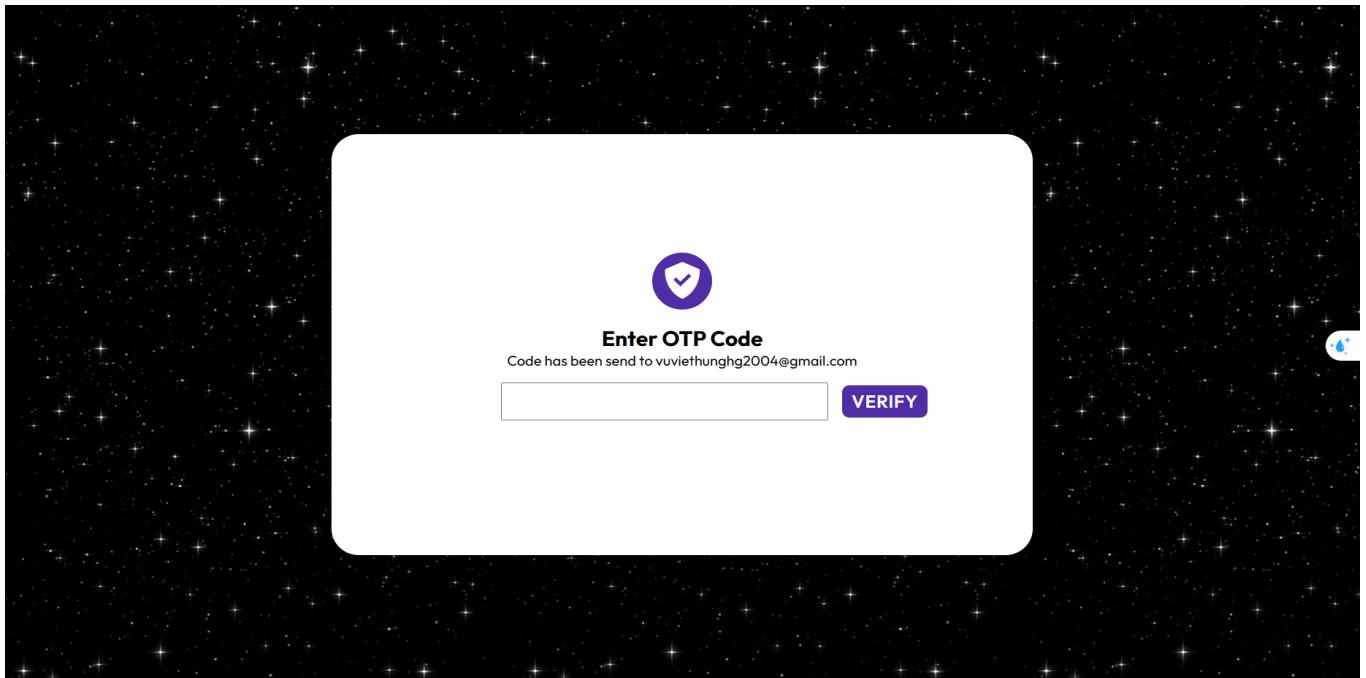
When you press the Sign Up button, it will switch to the Sign Up interface. The Sign Up interface will provide 5 fields for account creation:

- Name: Full name of the user
- Username: User's account name
- Email
- Phone number
- Account password



*Illustration of the Register interface*

After pressing the Sign Up button, you will move to the verification page. An OTP code will be sent to the email the user used to register their account. The user will enter the OTP code and press the Verify button.



*Illustration of the OTP code verification interface*

## 4.1 Room Type

### 4.1.1 Main interface of room type

		ROOM TYPE		ROOM LIST						
ID	Name	Description		Day Rate	Night Rate	Daily Rate	Overtime Pay \$/h	Capacity	Edit	Delete
1	Standard	Standard room with basic amenities		50	100	150	2	2		
2	Deluxe	Deluxe room with ocean view		30	150	200	2	2		
3	Suite	Luxurious suite with living area		50	250	300	2	4		
4	Family Room	Spacious room suitable for families		60	200	250	4	4		
5	Junior Suite	Has an area of 60 - 70m2, with beautiful views		100	250	300	4	4		
6	Executive Suite	Has an area of up to 80m2, equipped with both standing and lying bathtubs		45	300	350	4	6		
7	Bungalow	One-story house, with unique architecture and beautiful view		60	270	320	1	4		
8	Superior	Has better quality than Standard rooms		40	120	170	1	3		

*Illustration of room list interface*

### 4.1.2 Add Room Type

Users click on the "+" icon in the upper right corner of the screen. The filling form will automatically appear

**Room Category**

ID	Name
1	Standard
2	Deluxe
3	Suite
4	Family Room
5	Junior Suite
6	Executive Suite
7	Bungalow
8	Superior

Has better quality than Standard rooms

40    120    170    1    3    [Edit](#)    [Delete](#)

**Add New Type Of Room**

Name \*

Description \*

Day Rate  \$

Night Rate  \$

Daily Rate  \$

Overtime Pay  \$/h

Capacity \*

[SAVE](#)

*Illustration of the Add new type of Room form*

After filling in all the information:

**Room Category**

ID	Name
1	Standard
2	Deluxe
3	Suite
4	Family Room
5	Junior Suite
6	Executive Suite
7	Bungalow
8	Superior

Has better quality than Standard rooms

40    120    170    1    3    [Edit](#)    [Delete](#)

**Add New Type Of Room**

Name \*  VIP

Description \*  For Government

Day Rate  \$ 1000

Night Rate  \$ 2000

Daily Rate  \$ 5000

Overtime Pay  \$/h 1000

Capacity \*  2

[SAVE](#)

The user clicks Save and the new room type will be rendered on the screen

9	VIP	For Government	1000	2000	5000	1000	2	<a href="#">Edit</a>	<a href="#">Delete</a>
---	-----	----------------	------	------	------	------	---	----------------------	------------------------

#### 4.1.3 Edit Room Type

The user clicks on the Edit icon right on the row where they want to edit the information, an Edit Form will appear containing all the user's current information.

The screenshot shows a modal dialog titled "Edit Type Of Room" over a "Room Category" list. The modal contains fields for Name, Description, and various rates. The "Name" field is populated with "Suite". The "Description" field contains "Luxurious suite with living area". Under "Day Rate", the value is "\$ 50". Under "Night Rate", the value is "\$ 250". Under "Daily Rate", the value is "\$ 300". Under "Overtime Pay", the value is "\$/h 2". The "Capacity" field is currently set to "4". At the bottom right of the modal is a "SAVE" button. The background list shows room types: Standard, Deluxe, Suite, Family Room, Junior Suite, Executive Suite, Bungalow, and Superior. The "Suite" row is highlighted, indicating it is selected for editing. The "Edit" and "Delete" icons are visible next to each row in the list.

*Illustration of the Edit type of Room form*

Then users customize as desired. After adjustment (eg: Capacity:5). The interface will be updated after clicking SAVE.

3	Suite	Luxurious suite with living area	50	250	300	2	5		
---	-------	----------------------------------	----	-----	-----	---	---	--	--

#### 4.1.4 Delete Room Type

The user clicks on the DELETE button right on the row they want to delete, then a deletion confirmation message will appear (Suppose the room with ID is 2 is selected).

Room Category									
ID	Name	Description	Day Rate	Night Rate	Daily Rate	Overtime Pay \$/h	Capacity	Edit	Delete
1	Standard	Standard room with basic amenities	50	100	150	2	2		
2	Deluxe	Deluxe room with ocean view	30	150	200	2	2		
3	Suite	Luxurious suite with living area			300	2	4		
4	Family Room	Spacious room suitable for families			250	4	4		
5	Junior Suite	Has an area of 60 - 70m2, with beautiful views	100	250	300	4	4		
6	Executive Suite	Has an area of up to 80m2, equipped with both standing and lying bathtubs	45	300	350	4	6		
7	Bungalow	One-story house, with unique architecture and beautiful view	60	270	320	1	4		
8	Superior	Has better quality than Standard rooms	40	120	170	1	3		

*Error message is displayed*

When the user selects “Yes” the room will disappear from the table. When the user clicks "No" everything will remain the same. (Suppose you click YES) line number 2 will disappear from the table

Room Category									
ID	Name	Description	Day Rate	Night Rate	Daily Rate	Overtime Pay \$/h	Capacity	Edit	Delete
1	Standard	Standard room with basic amenities	50	100	150	2	2		
3	Suite	Luxurious suite with living area	50	250	300	2	4		
4	Family Room	Spacious room suitable for families	60	200	250	4	4		
5	Junior Suite	Has an area of 60 - 70m2, with beautiful views	100	250	300	4	4		
6	Executive Suite	Has an area of up to 80m2, equipped with both standing and lying bathtubs	45	300	350	4	6		
7	Bungalow	One-story house, with unique architecture and beautiful view	60	270	320	1	4		
8	Superior	Has better quality than Standard rooms	40	120	170	1	3		

*Room Category after deleting the room has ID 2*

## 4.2 Room List

4.2.1 Main interface of room list: Provides information about room name, room type, price if rented during the day, night or by day and capacity

Room List											
	Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand
<input type="checkbox"/>	101	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	102	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	103	Standard	50	100	100	Unavailable	2	2			
<input type="checkbox"/>	104	Standard	50	100	100	Unavailable	2	2			
<input type="checkbox"/>	201	Deluxe	30	150	150	Available	2	2			
<input type="checkbox"/>	202	Deluxe	30	150	150	Unavailable	2	2			
<input type="checkbox"/>	203	Deluxe	30	150	150	Available	2	2			
<input type="checkbox"/>	204	Deluxe	30	150	150	Available	2	2			

*Illustration of Room List*

### 4.2.2 Room Filter

Is the sidebar part of the room list. When selecting table fields, they will only be displayed according to the selected field. (For example, select Type: Standard)

Room List											
	Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand
<input type="checkbox"/>	101	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	102	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	103	Standard	50	100	100	Unavailable	2	2			
<input type="checkbox"/>	104	Standard	50	100	100	Unavailable	2	2			

### 4.2.3 Room Table Sort

- In the table header next to each label there will be an arrow corresponding to the currently applied sort (the arrow only appears when moving the mouse to the label, otherwise the default will be sort by Room Name). Suppose in this case, choose Sort by Day Rate:
- Interface before sorting (Day Rate increasing - arrow direction going up)

		ROOM TYPE		ROOM LIST									
Name:		Type	Room Name	Type	↑ Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximun Capacity	Pay	Edit	Expand
			201	Deluxe	30	150	150	Available	2	2			
			202	Deluxe	30	150	150	Unavailable	2	2			
			203	Deluxe	30	150	150	Available	2	2			
			204	Deluxe	30	150	150	Available	2	2			
			801	Deluxe	30	150	150	Available	2	2			
			802	Deluxe	30	150	150	Available	2	2			
			601	Executive Suite	45	300	300	Available	4	6			
			602	Executive Suite	45	300	300	Unavailable	4	6			

Rows per page: 8 ▾ 1–8 of 39 < >

Dense padding

- After sorting (Day Rate decreasing - arrow direction going down):

Room List												
	Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand	
<input type="checkbox"/>	501	Junior Suite	100	250	250	Available	4	4				
<input type="checkbox"/>	502	Junior Suite	100	250	250	Unavailable	4	4				
<input type="checkbox"/>	503	Junior Suite	100	250	250	Available	4	4				
<input type="checkbox"/>	504	Junior Suite	100	250	250	Unavailable	4	4				
<input type="checkbox"/>	705	Junior Suite	100	250	250	Unavailable	4	4				
<input type="checkbox"/>	401	Family Room	60	200	200	Available	4	4				
<input type="checkbox"/>	402	Family Room	60	200	200	Available	4	4				
<input type="checkbox"/>	403	Family Room	60	200	200	Available	4	4				

Rows per page: 8 ▾ 1–8 of 39 < >

Dense padding

Users can click next to change the desired Sort order.

#### 4.2.4: Add Room

Users click on the "+" icon in the upper right corner of the screen. (The "+" sign only appears when there are no checkboxes selected. Use the checkbox for deleting a room. When at least one checkbox is selected, the "+" icon will change to the DELETE icon). After clicking, a form to fill in information will appear.

Room List												
	Room Name	Type	Day	Night	Daily	Status	Overtime	Maximum	Pay	Edit	Expand	
<input type="checkbox"/>	204	Deluxe	30	150	150	Available	2	2				

Add a new Room

Name \*

Type \*

Status\*

Available  Unavailable

Notes

**SAVE**

Dense padding

*Illustration of Add a new Room form*

Next, the user fills in all the necessary information to create a new room and click Save. However, if the necessary information has not been filled in, an Alert will appear. In addition, when setting up a new name, if that name is duplicated, you will be warned. Besides, when users select Type for the room, the necessary information will be displayed. Suppose in this case the user fills in the name 203 (duplicate name), selects the type "Suite" and deliberately does not select Status, the interface will be as follows:

The screenshot shows a modal dialog titled 'Add a new Room'. At the top, there is a red alert box with the text '① Please fill in all required fields.' Below it, the 'Name\*' field contains '203', which is highlighted in red with the message 'Name already exists.' The 'Type\*' dropdown is set to 'Suite'. Under 'Status\*', there are three radio buttons: 'Available', 'Unavailable', and 'Both', none of which are selected. A note section is empty. To the right of the form, a sidebar displays room pricing details: Day Rate: 50, Night Rate: 250, Daily Rate: 300, Overtime Pay: 2, and Capacity: 4. At the bottom right of the modal is a 'SAVE' button.

*Illustration of the Add a new Room form showing an error message*

If the user fills in the appropriate information, after clicking Save the room will be displayed at the bottom of the list and sorted in the correct position for the next render.

<input type="checkbox"/>	205	Suite	50	250	300	Available	2	4			
Rows per page: 8 1–8 of 39 < >											

Dense padding

#### 4.2.5: Edit Room

Users click directly on the Edit icon on the room they want to edit, an information form will appear with the room's current information pre-set. The user will then edit the information if necessary (Note that if you set the same name, you will receive a Warning). Suppose you choose room 102, the interface will be as follows:

ROOM TYPE    ROOM LIST

Name:

Type

- STANDARD
- FAMILY ROOM
- ALL

Status

- Available
- Unavailable
- Both

### Edit room information

Name \*

Type \*

- Standard

Status

- Available
- Unavailable

Notes \*

<input type="checkbox"/>	203	Deluxe	30	150	150	Available	2	2	<input type="button" value="Edit"/>	<input type="button" value="▼"/>
<input type="checkbox"/>	204	Deluxe	30	150	150	Available	2	2	<input type="button" value="Edit"/>	<input type="button" value="▼"/>

Price of room is applied follow type of room:

Day Rate:	50
Night Rate:	100
Daily Rate:	100
Overtime Pay:	2
Capacity:	2

Edit    Expand

*Illustration Edit room information form*

Warning if editing invalid name:

ROOM TYPE    ROOM LIST

Name:

Type

- STANDARD
- FAMILY ROOM
- ALL

Status

- Available
- Unavailable
- Both

### Edit room information

Name \*

Name already exists.

Type \*

- Standard

Status

- Available
- Unavailable

Notes \*

<input type="checkbox"/>	203	Deluxe	30	150	150	Available	2	2	<input type="button" value="Edit"/>	<input type="button" value="▼"/>
<input type="checkbox"/>	204	Deluxe	30	150	150	Available	2	2	<input type="button" value="Edit"/>	<input type="button" value="▼"/>

Price of room is applied follow type of room:

Day Rate:	50
Night Rate:	100
Daily Rate:	100
Overtime Pay:	2
Capacity:	2

Edit    Expand

After clicking save, the room information will be updated

Room List											
	Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand
<input type="checkbox"/>	101	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	90	Standard	50	100	100	Available	2	2			
<input type="checkbox"/>	103	Standard	50	100	100	Unavailable	2	2			
<input type="checkbox"/>	104	Standard	50	100	100	Unavailable	2	2			
<input type="checkbox"/>	201	Deluxe	30	150	150	Available	2	2			
<input type="checkbox"/>	202	Deluxe	30	150	150	Unavailable	2	2			
<input type="checkbox"/>	203	Deluxe	30	150	150	Available	2	2			
<input type="checkbox"/>	204	Deluxe	30	150	150	Available	2	2			

#### 4.2.6 Room Image

Users click on the EXPAND icon right on the line containing the room they want to see photos. The images will be displayed and automatically arranged in a masonry layout for art

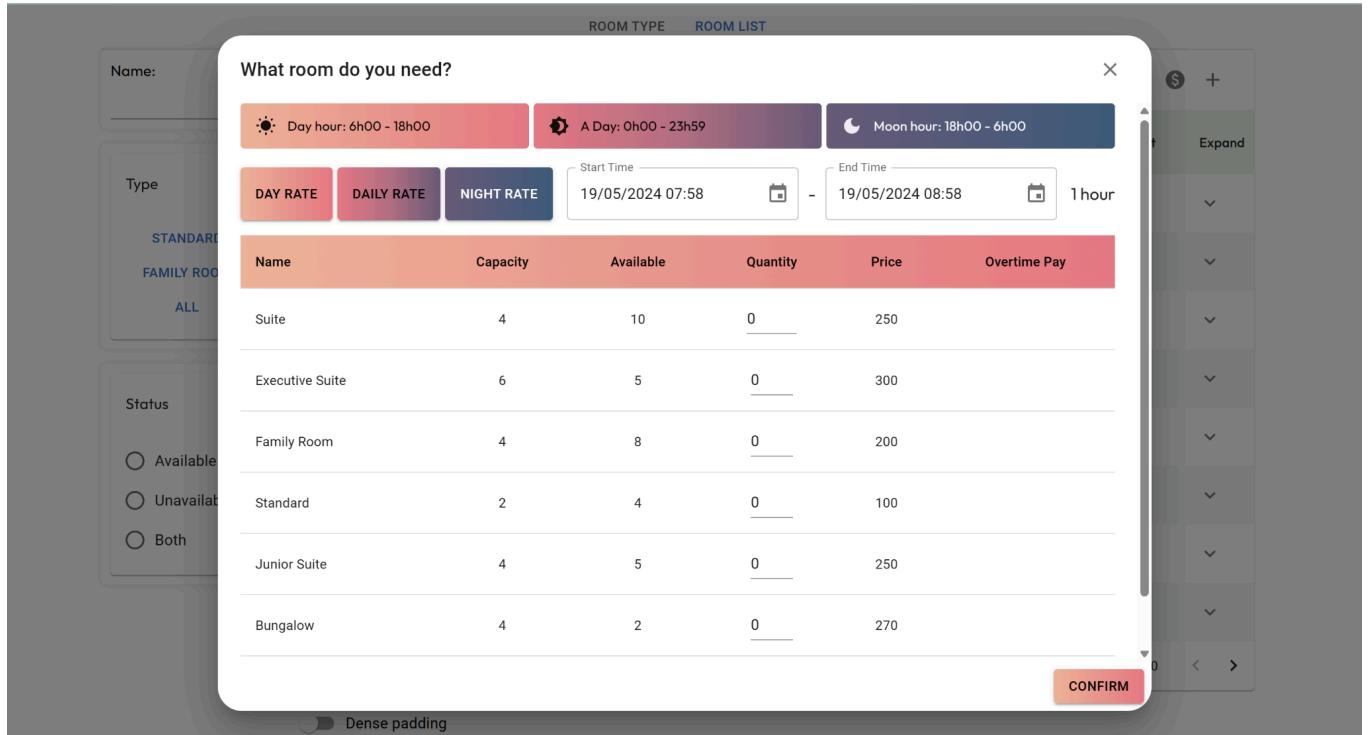
Room List											
	Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand
<input type="checkbox"/>	101	Standard	50	100	100	Available	2	2			
IMAGES											
						Bed					
<input type="checkbox"/>	90	Standard	50	100	100	Available	2	2			

Room List interface after pressing expand in a specific room

## 4.3 Booking

Users click on the "\$" icon in the top right corner of the screen to create a new booking.

### 4.3.1. Booking main interface



*Booking main interface*

When the user changes the Rate (Day Rate, Daily Rate, Night Rate), the interface will change as follows

ROOM TYPE    ROOM LIST

What room do you need?

Day hour: 6h00 - 18h00 A Day: 0h00 - 23h59 Moon hour: 18h00 - 6h00

DAY RATE DAILY RATE NIGHT RATE

Start Time: 19/05/2024 07:58 - End Time: 19/05/2024 08:58 O day

Name	Capacity	Available	Quantity	Price	Overtime Pay
Suite	4	10	0	250	
Executive Suite	6	5	0	300	
Family Room	4	8	0	200	
Standard	2	4	0	100	
Junior Suite	4	5	0	250	
Bungalow	4	2	0	270	

CONFIRM

Dense padding

Booking form interface if Day Rate is selected

ROOM TYPE    ROOM LIST

What room do you need?

Day hour: 6h00 - 18h00 A Day: 0h00 - 23h59 Moon hour: 18h00 - 6h00

DAY RATE DAILY RATE NIGHT RATE

Start Time: 19/05/2024 07:58 - End Time: 19/05/2024 08:58 1 hour

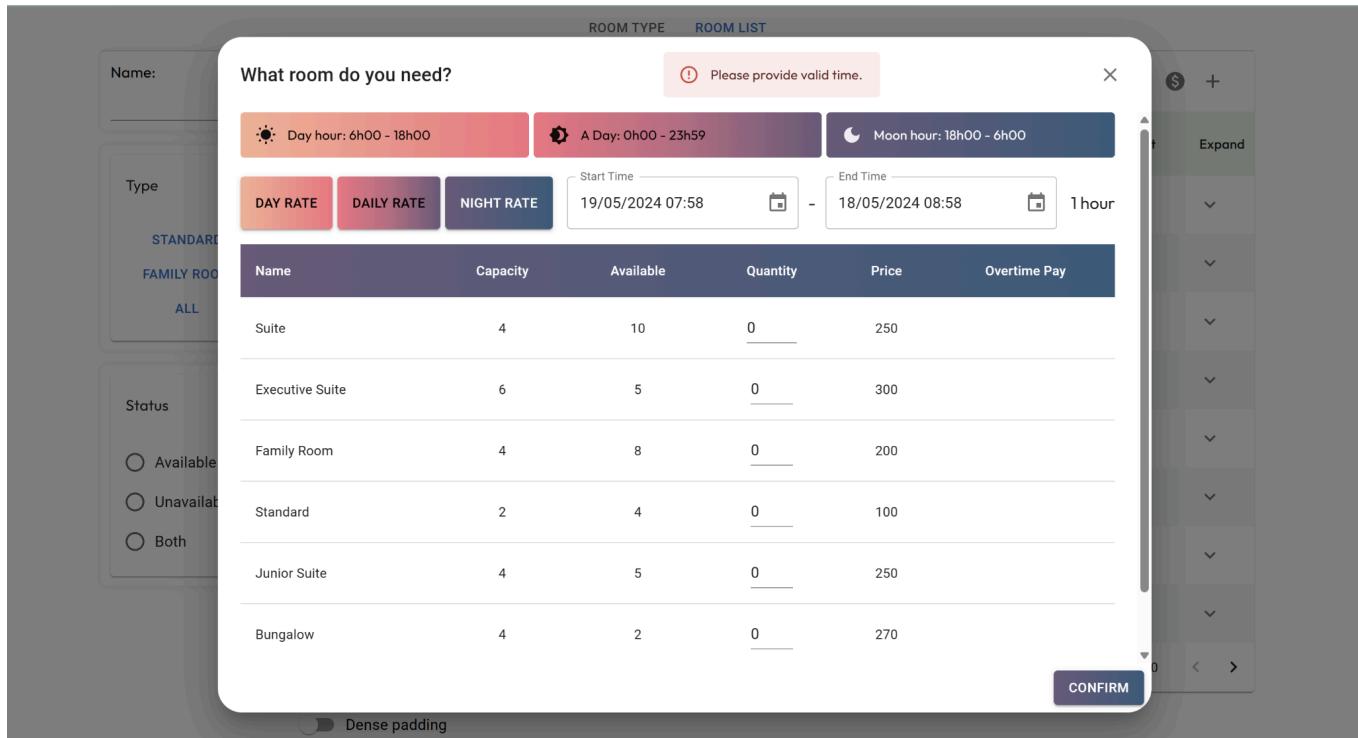
Name	Capacity	Available	Quantity	Price	Overtime Pay
Suite	4	10	0	250	
Executive Suite	6	5	0	300	
Family Room	4	8	0	200	
Standard	2	4	0	100	
Junior Suite	4	5	0	250	
Bungalow	4	2	0	270	

CONFIRM

Dense padding

Booking form interface if selecting Night Rate

If the user provides an unknown time (for example, an end date before a start date) a warning will appear



#### 4.3.2: Booking details interface

After the user selects the appropriate room type (Quantity of the selected room types  $> 0$ ) and clicks CONFIRM, the Booking details interface will appear. Here, users continue to provide information including: customer information and appropriate room selection.

Type	Quantity	Rooms	Rate	Start	End	Anticipated	Money <sup>9</sup>
Suite	1	Select room	Invalid time range	07:58/19/05/2024	08:58/26/05/2024	168 hour	0
Executive Suite	1	Select room	Invalid time range	07:58/19/05/2024	08:58/26/05/2024	168 hour	0

Customer notes

0 USD

Rows per page: 8 ▾ 1-8 of 40 < >

Dense padding

Detail Booking form interface

#### 4.3.2.1: Giao diện thông tin khách hàng

The user clicks on the "+" icon on the screen. A Form providing information will appear

Then the user needs to fill in the necessary information. If not, you will be warned

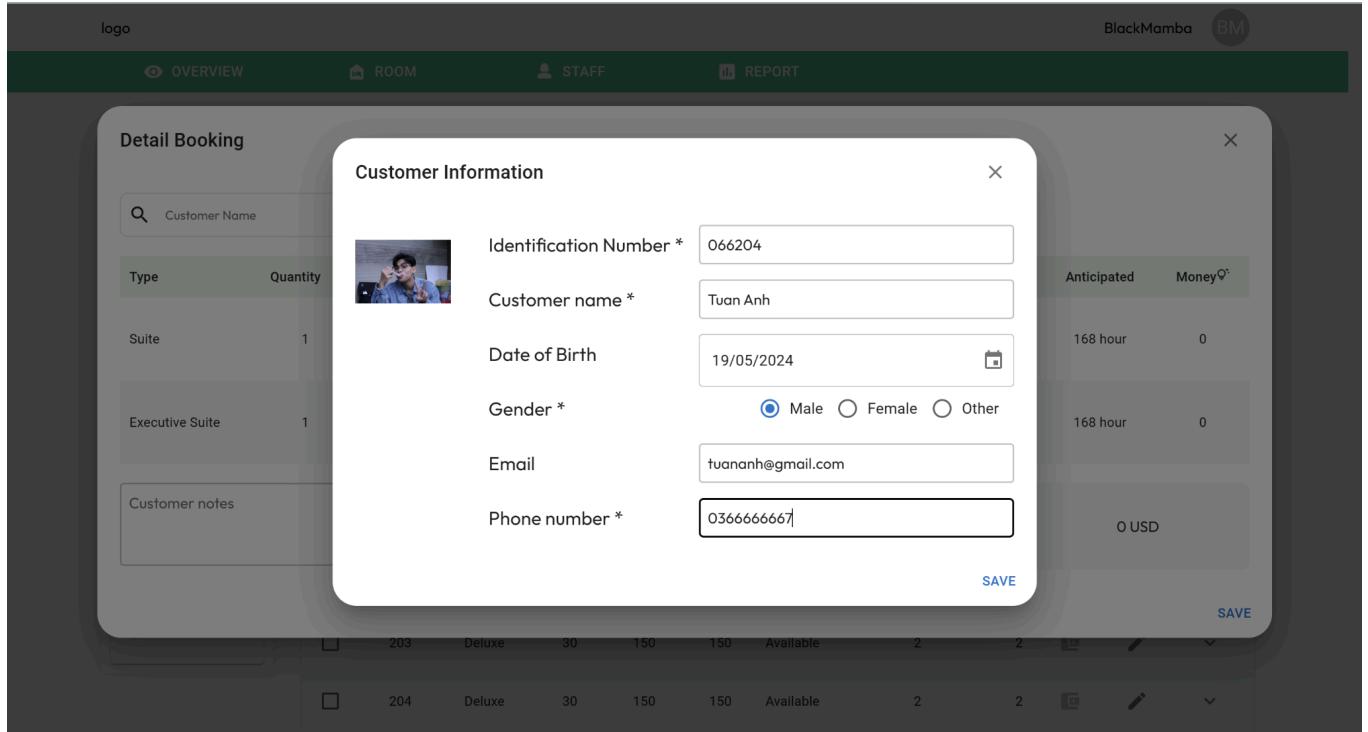
The screenshot shows a mobile application interface for managing room bookings. At the top, there are tabs for "ROOM TYPE" and "ROOM LIST". Below the tabs, the title "Room List" is displayed. On the left, a sidebar titled "Detail Booking" lists room types and quantities: "Suite" (1) and "Executive Suite" (1). A "Customer notes" section is also present. In the center, a modal window titled "Customer Information" is open. It contains fields for "Identification Number", "Customer name \*", "Date of Birth" (set to "19/05/2024"), "Gender \*", "Email", and "Phone number \*". There is a note at the top of the modal saying "Please provide enough necessary information". Below the fields are two "SAVE" buttons. The background shows a grid of room details with columns for "Anticipated" and "Money". At the bottom of the screen, there is a footer with "Rows per page: 8" and a page number "1-8 of 40". A note "Dense padding" is visible near the bottom left.

*The Customer Information form displays a message asking for complete information*

If the user provides all the necessary information, after clicking Save the information will be saved and the name will be displayed on Detail Booking.

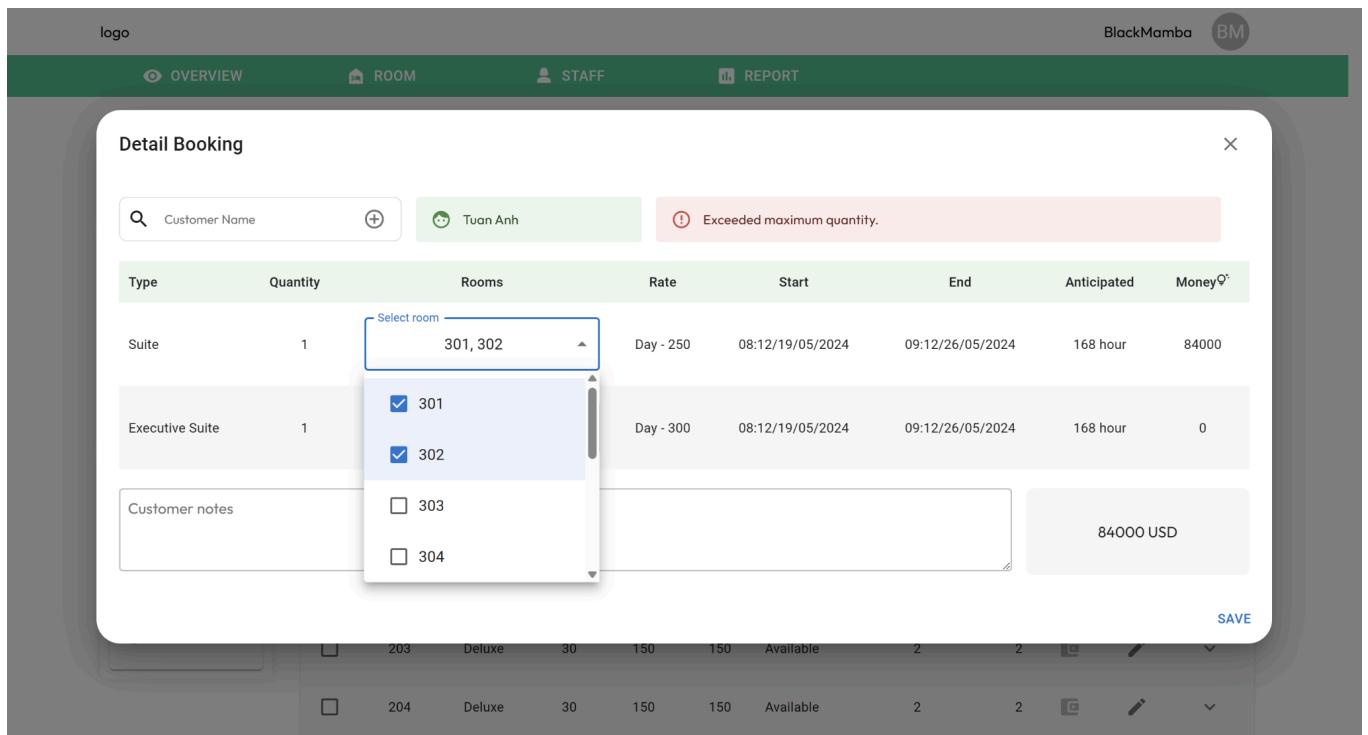
#### 4.3.2.2

Detail Booking interface after having customer information



*Customer Information interface when full customer information is available*

Next, the user selects the rooms. However, if you select more than the initially entered quantity, you will be warned. Suppose in this case you deliberately choose 2 Suite rooms when you initially only choose 1. If there is nothing wrong, when you click Save the booking will be saved and the room will be adjusted from "Available" to "Unavailable".



*The Detail Booking interface displays an error message that exceeds the number entered*

### 4.3.3 Payment interface

Users click directly on the Pay icon on the room they want to pay. (The icon will be green if the room is Unavailable and available for payment, and gray when Available). After pressing a Dialog will appear

**Room List**

Room Name	Type	Day Rate	Night Rate	Daily Rate	Status	Overtime Pay	Maximum Capacity	Pay	Edit	Expand
101	Standard	50	100	100	Available	2	2			
102	Standard	50	100	100	Available	2	2			
103	Standard	50	100	100	Unavailable	2	2			
104	Standard	50	100	100	Unavailable	2	2			
201	Deluxe	30	150	150	Available	2	2			
202	Deluxe	30	150	150	Unavailable	2	2			
203	Deluxe	30	150	150	Available	2	2			
204	Deluxe	30	150	150	Available	2	2			

Rows per page: 8 1-8 of 40

Dense padding

*In the RoomList interface, only green icons can pay*

Contains information about the current customer and the amount to be paid

**PAYMENT OF ROOM: 303**

Customer Information	
Customer name	Tuan Anh
Identification Number	2004
Gender	male
Phone	0123496894

Type	Check In	Check Out	Time	Money
Suite	23:00/16/05/2024	03:15/17/05/2024	4.25 hours	1000

Room Money: 1000  
Surcharge: \_\_\_\_\_  
Discount (%): \_\_\_\_\_  
Total amount due: 1000.00  
Payment Method:  
 Cash  Bank Transfer

**Payment interface**

Users can choose to pay in cash or pay by bank transfer (when transferring, a QR will appear)

The screenshot shows the 'PAYMENT OF ROOM: 303' interface. It includes sections for 'Customer Information' (Customer name: Tuan Anh, Identification Number: 2004, Gender: male, Phone: 0123496894), 'Room Money' (1000), 'Surcharge' (200), 'Discount (%)' (3), and 'Total amount due' (1170.00). The 'Payment Method' section shows 'Cash' and 'Bank Transfer' (selected). A QR code is displayed for bank transfer. The bottom right corner has a 'SAVE' button.

*The Payment interface displays QR*

After clicking save, the information will be loaded for 3 seconds to complete and a success message will appear and the room will be changed to Available status.

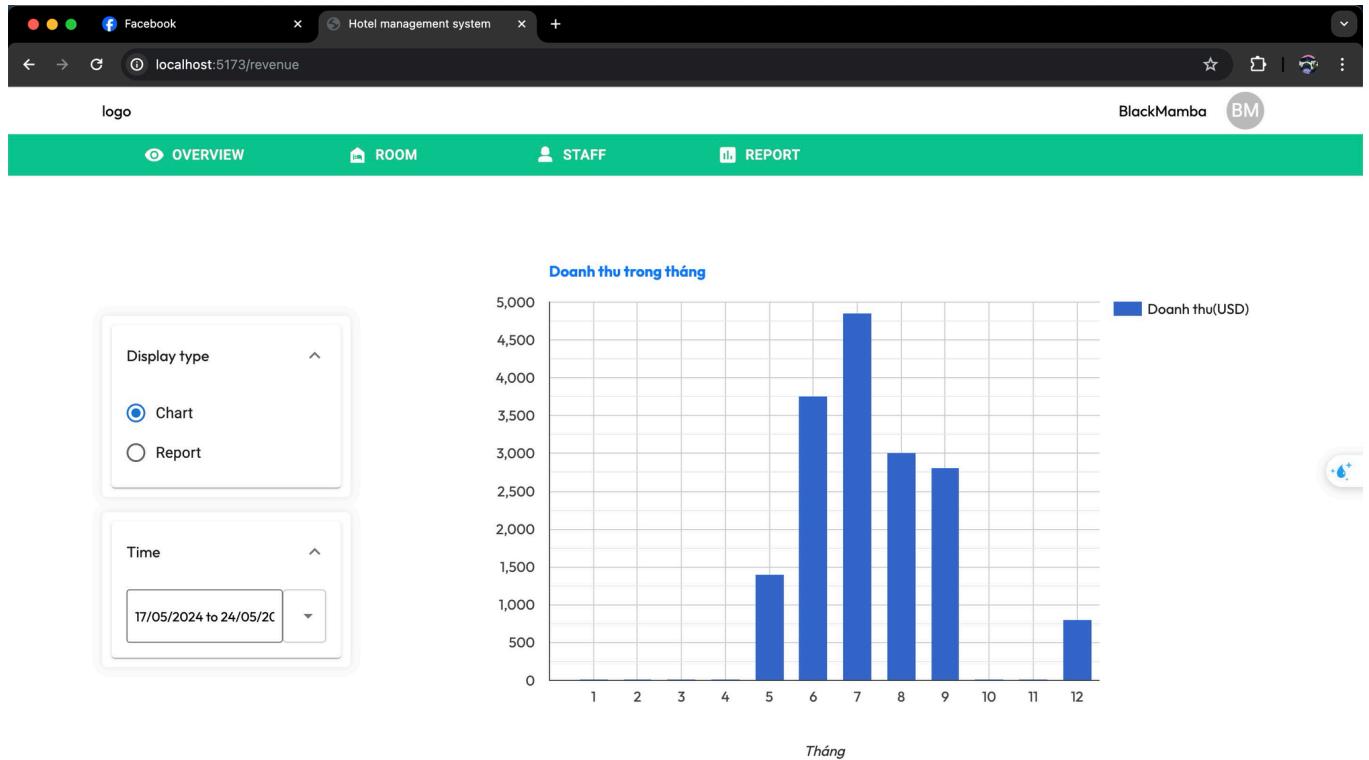
The screenshot shows the 'Room List' interface with a header 'ROOM TYPE' and 'ROOM LIST'. A green circular notification bar at the top center says 'Payment success'. The main table lists rooms with columns: Room Name, Type, Day Rate, Night Rate, Daily Rate, Status, Overtime Pay, Maximum Capacity, Pay, Edit, and Expand. The table contains 8 rows of room data. At the bottom, there are buttons for 'Rows per page' (set to 8), '1-8 of 40', and navigation arrows.

Dense padding

*Successful transaction*

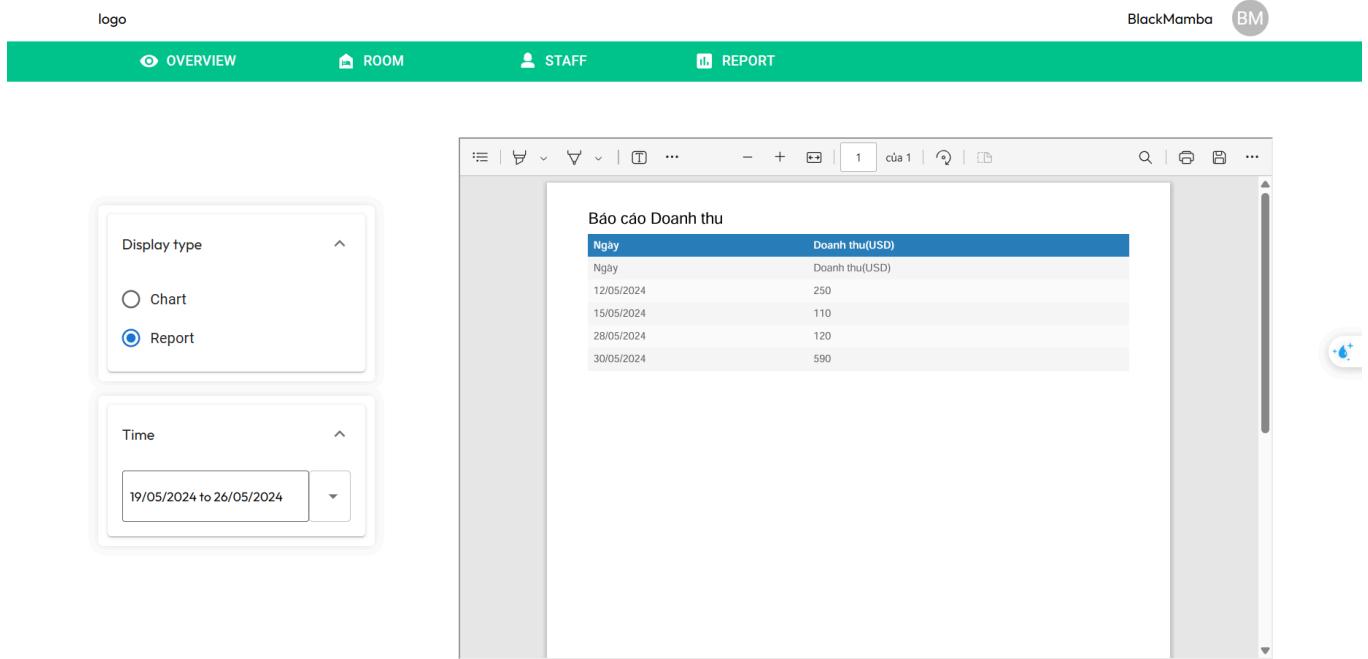
## 4.3 Report

The interface displays the hotel's revenue report so that the hotel can monitor revenue and make reasonable policies to increase revenue. The default initial view of revenue will be displayed in graph form:



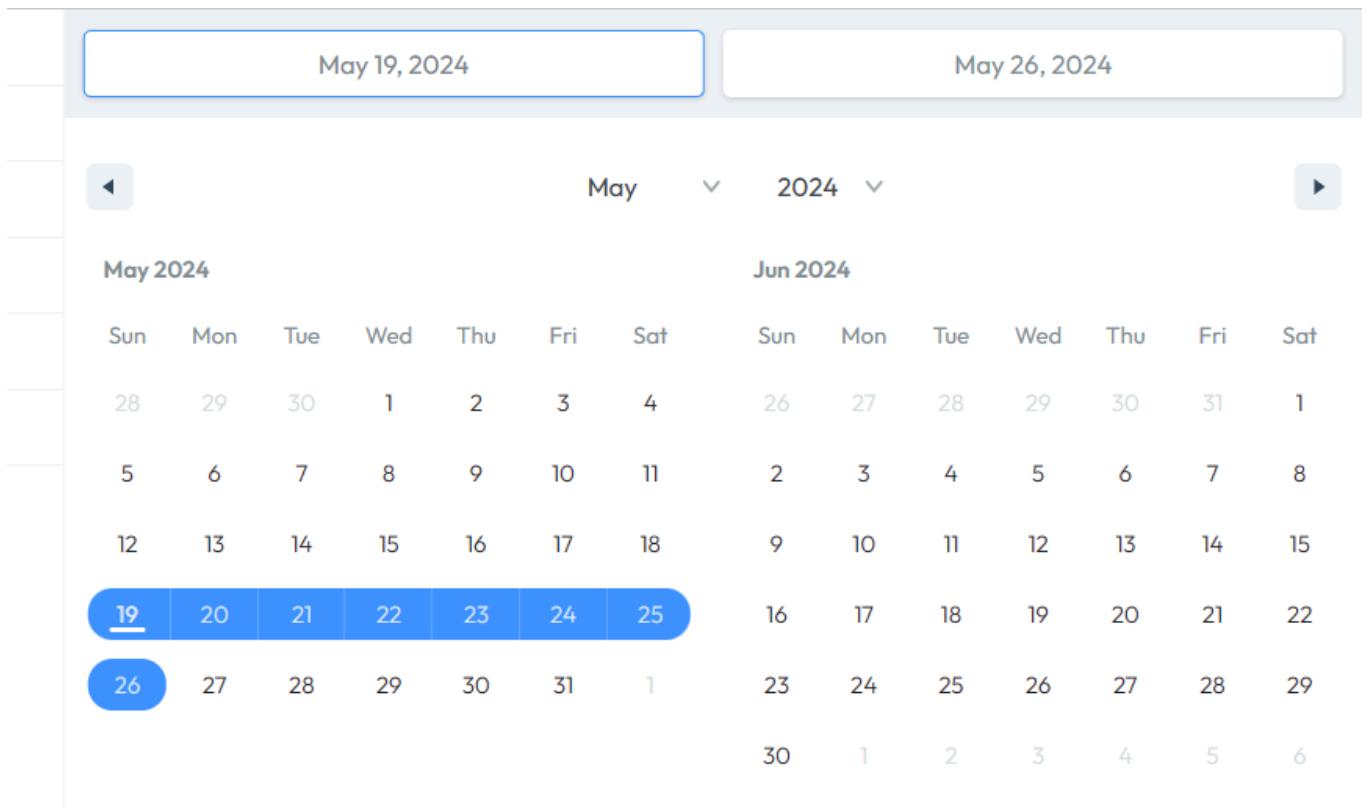
*The main interface of the Revenue report is in chart form*

The report can also be displayed in PDF format, when the user selects Report format in the Display Type menu.



## *Revenue report interface when in PDF format*

## Revenue report interface when in PDF format



## Date Range Picker

# V. Database design:

## 5.1. Description

For hotel management systems, MySQL is used to manage the database. Basically, the system's data requires mutual constraints, so MySQL stores data in SQL form. a good choice. Because it can be queried quickly, bringing a good experience to users. Besides, MySQL has a large user community, with many people solving problems and optimizing performance.

MySQL is also available on a variety of platforms, including Windows, Linux, and MacOS, making it a versatile choice for software development environments.

In the design of MySQL there are 4 concepts:

- Schema: Defines and describes the organizational structure of the database, including many Tables
- Table: Is a table with many columns and rows, represented in the database
- Column: Are the attributes of the entity
- Row: is a set of data records of an entity

## 5.2. Design:

a, Booking is the place to store information about a booking:

Field	Means
bookingId (int)	Primary Key
checkin_date (datetime)	Booking date
checkout_date(datetime)	Check-out date
guestid(int)	FK, lis the id of the booking customer, references to the Guest table
room_number(int)	Room number booked

total_price(float)	Total amount expected to be
--------------------	-----------------------------

b, Guest is the place to store information about the guest:

Field	Means
guestid(int)	Primary Key
date_of_birth(datetime)	customer's birthday
email(varchar)	Customer's email
first_name(varchar)	Customer's first name
last_name(varchar)	Customer's last name
phone(varchar)	Customer's phone number
gender(varchar)	gender
cccd(varchar)	Identification Code
image (varchar)	Image
notes(varchar)	More information?

c, Hotel is the place to store information about the hotel:

Field	Means
hotelid(int)	Primary Key

address(varchar)	address of Hotel
name(varchar)	hotel's name
phone(varchar)	phone number
work_time_start(varchar)	The hotel is now open for service
work_time_end(varchar)	The hotel now stops accepting services
email(varchar)	Hotel email
stars(int)	Evaluate hotel quality

d, Payment is where information about payment invoices is stored:

Field	Means
paymentid(int)	Primary Key
amount(float)	The money have to pay
bookingid(int)	FK, What booking does it refer to?
payment_date(varchar)	payment date
payment_method(varchar)	payment method

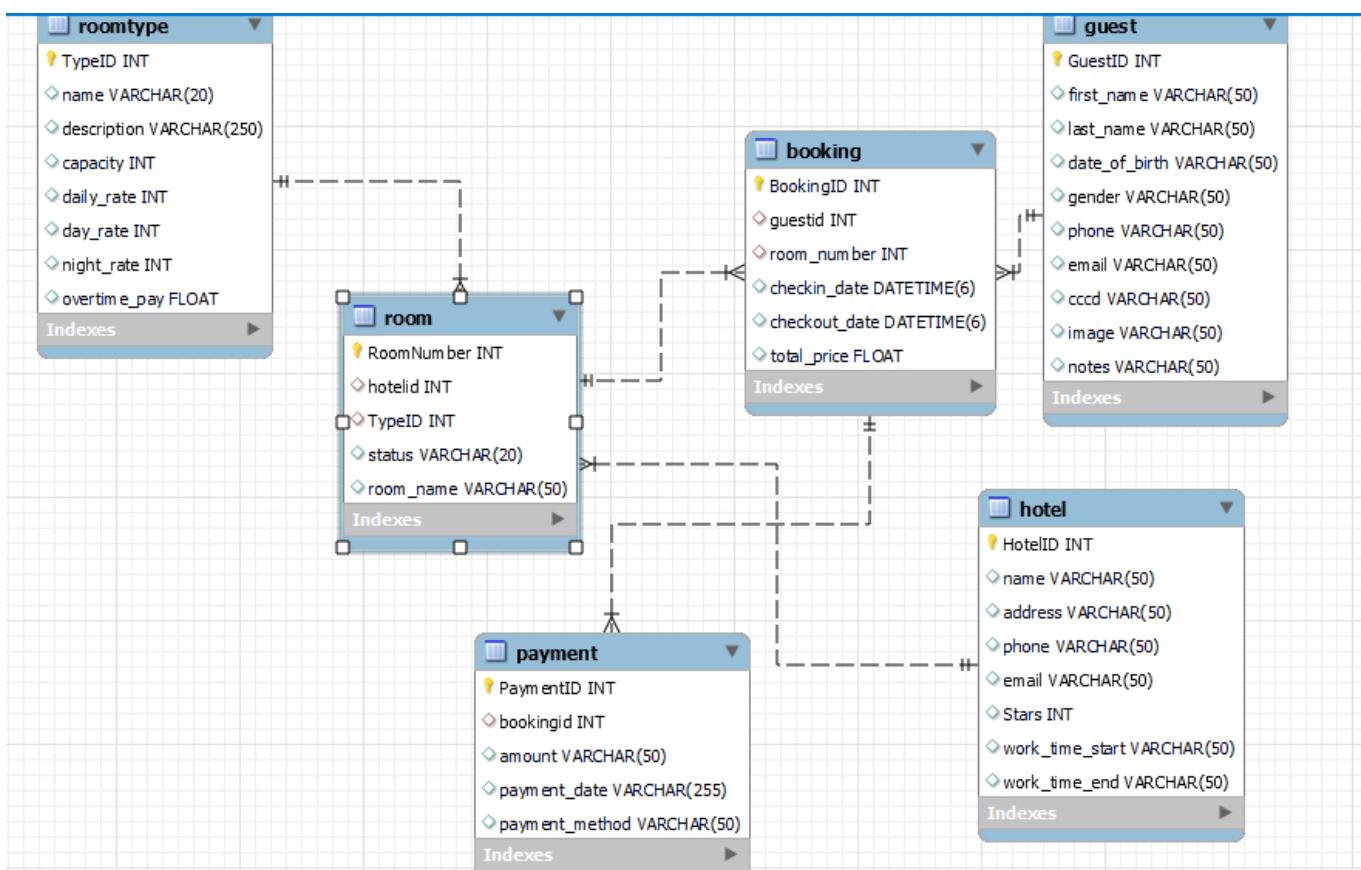
e, Room is the place to store information about rooms in the hotel:

<b>Field</b>	<b>Means</b>
room_number(int)	Primary Key
status(varchar)	room status
hoitelid(int)	FK, Which hotel is the reference to?
typeid(int)	FK, reference to the room type in the RoomType table
room_name(varchar)	room name

f, RoomType stores room types:

<b>Field</b>	<b>Means</b>
type(int)	Primary Key
capacity(int)	Number of people in 1 room allowed
description(varchar)	Describe room features
name(varchar)	Room type's name
night_rate(float)	How much does it cost at night?
daily_rate(float)	How much does it cost during the day?

day_rate(float)	How much does it cost all the day?
overtime_pay(float)	How much to pay for overtime:



g, User is the place to store account information representing a hotel

Field	Means
hotelId(int)	Primary Key
email (varchar)	Hotel representative account email

full_name (varchar)	Full name of representative
phone (varchar)	phone Number
user_name (varchar)	The account name that represents the hotel
user_password (varchar)	Password to log in to the hotel representative account

h, UserAccount is the place to store accounts for users, including managers, receptionists and admins

Field	Means
user_name (varchar)	Primary Key
password (varchar)	Account login password
active (bit)	Check if the account is locked or not
hotelId (int)	Hotel ID
role (varchar)	account's role

i, UserSession is where login sessions are stored after a user successfully logs in

Field	Means
sessionId (int)	Primary Key - session ID

role (varchar)	account's role
user_name (varchar)	username