VIETNAM INTERNATIONAL UNIVERSITY – HO CHI MINH CITY

INTERNATIONAL UNIVERSITY

WEB APPLICATION DEVELOPMENT PROJECT

ONLINE MOTEL MANAGEMENT

By

Huỳnh Minh Duy - Student ID: ITCSIU21174

Trương Trí Dũng - Student ID: ITCSIU21126

Advisor: Dr. Nguyen Van Sinh

A report submitted to the School of Computer Science and Engineering in
partial fulfillment of the requirements for the Final Project
in Web Application Development course - Fall 2024

Ho Chi Minh city, Vietnam, 2024

Contents

I. INTRODUCTION 3

II. REQUIREMENT ANALYSIS AND DESIGN 9

1. REQUIREMENT ANALYSIS 9

2. FUNCTIONAL REQUIREMENTS 14

Use Case 1: Manage Deals 14

Use Case 2: Manage Rooms 15

Use Case 3: Manage Rates 16

Use Case 4: Manage Guests 17

3. NON-FUNCTIONAL REQUIREMENTS 18

a. Operational requirements: 18

b. Legal requirements: 18

c. Usability requirements: 18

d. Humanity Requirements: 18

e. Performance Requirements: 19

f. Maintainability Requirements: 20

g. Support Requirements 20

h. Security requirements: 21

i. Interface requirements: 21

III. DESIGN 22

IV. IMPLEMENTATION 51

V. REQUIREMENT ANALYSIS AND DESIGN 57

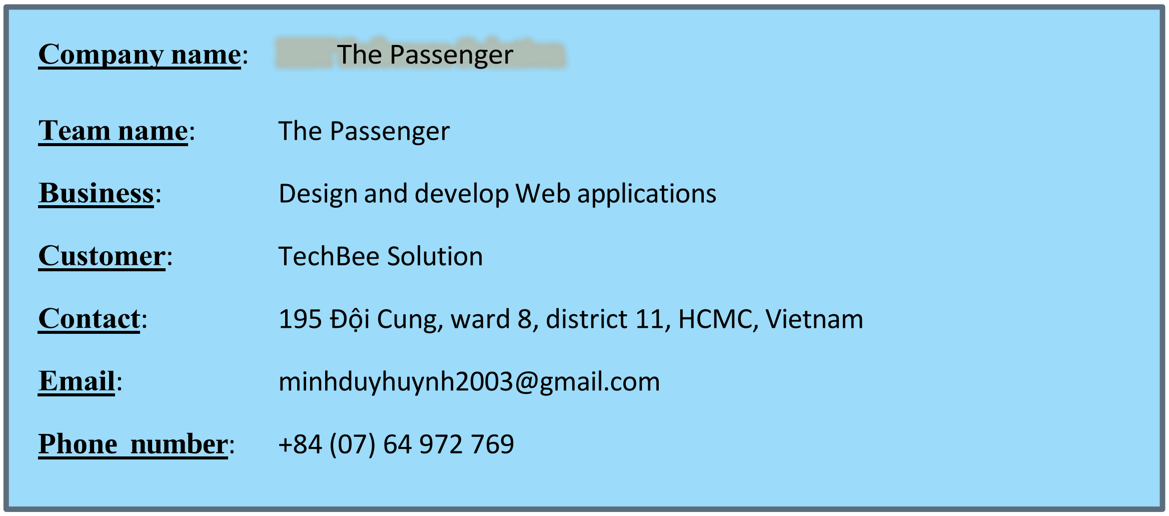
VI. REFERENCES 58

I. INTRODUCTION

Our Motel Management Project aims to streamline and enhance the operational efficiency of motel businesses through a comprehensive software solution. This project includes features for room booking management, guest check-in/check-out, billing and invoicing, and inventory control. The system will also provide detailed reporting and analytics to help motel managers make informed decisions. By automating routine tasks and improving data accuracy, the project seeks to improve the overall guest experience and optimize resource utilization for motel operators.

1. ABOUT US

Welcome to The Passenger, your trusted partner in modern motel management. Our mission is to revolutionize the hospitality industry with a cutting-edge platform that simplifies room bookings, guest management, billing, and inventory control. Designed for efficiency and ease of use, The Passenger helps motel owners and managers enhance operational performance and deliver exceptional guest experiences. Join us on our journey to set new standards in motel management.



Company name:	The Passenger
Team name:	The Passenger
Business:	Design and develop Web applications
Customer:	TechBee Solution
Contact:	195 Đội Cung, ward 8, district 11, HCMC, Vietnam
Email:	minhduyhuynh2003@gmail.com
Phone number:	+84 (07) 64 972 769

Figure 1

2. THE PRODUCT’S INFORMATION

Introducing The Passenger, an all-in-one motel management solution designed to streamline your operations and elevate guest experiences. Our platform offers robust features for room booking management, seamless check-in/check-out processes, efficient billing and invoicing, and comprehensive inventory control. With intuitive reporting and analytics, The Passenger empowers motel managers to make data-driven decisions and optimize resource utilization. Experience the future of motel management with The Passenger.

3. WORK BREAKDOWN STRUCTURE

The structure of our project can be expressed in the ***Figure 3.***

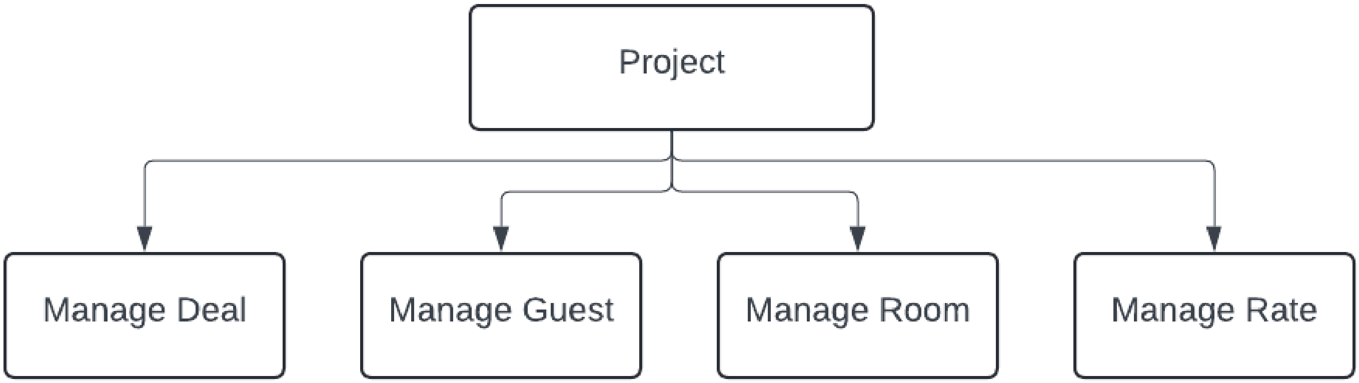


Figure 3.1

In this tree, we have divided it into smaller subtrees which describe the tasks needed to be accomplished of each team member.

The tasks for the Manage Deal are described in the ***Figure 3.2***:

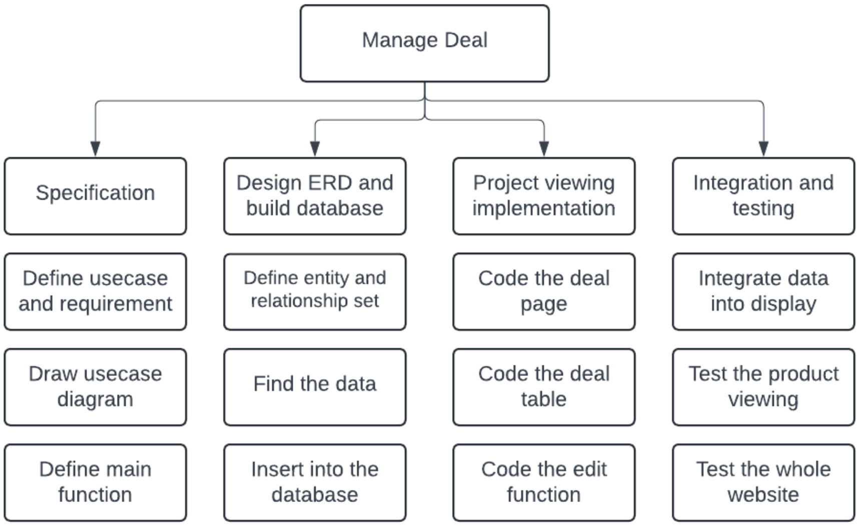


Figure 3.2

The tasks for the Manage Guest are described in the ***Figure 3.3***:

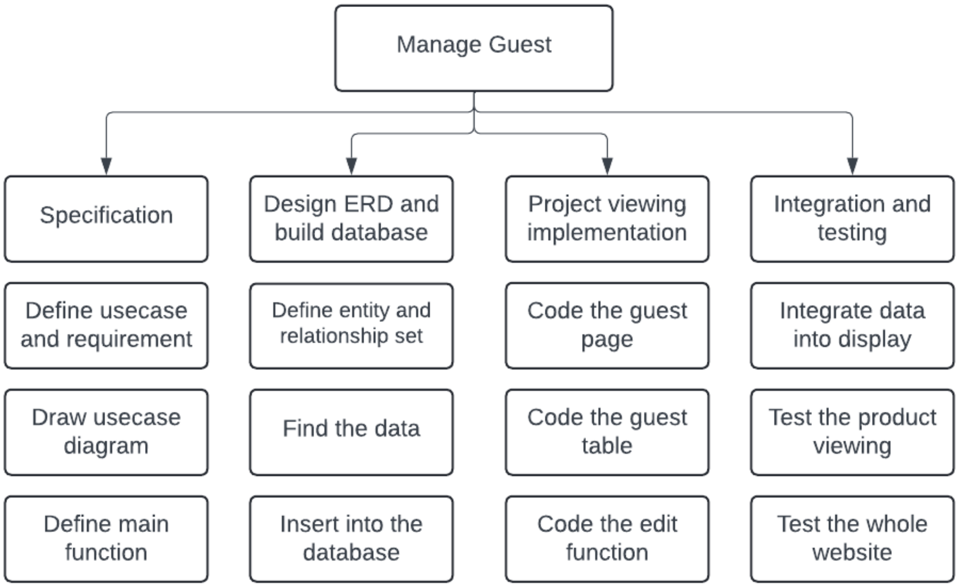


Figure 3.3

The tasks for the Manage Room are described in the ***Figure 3.4***:

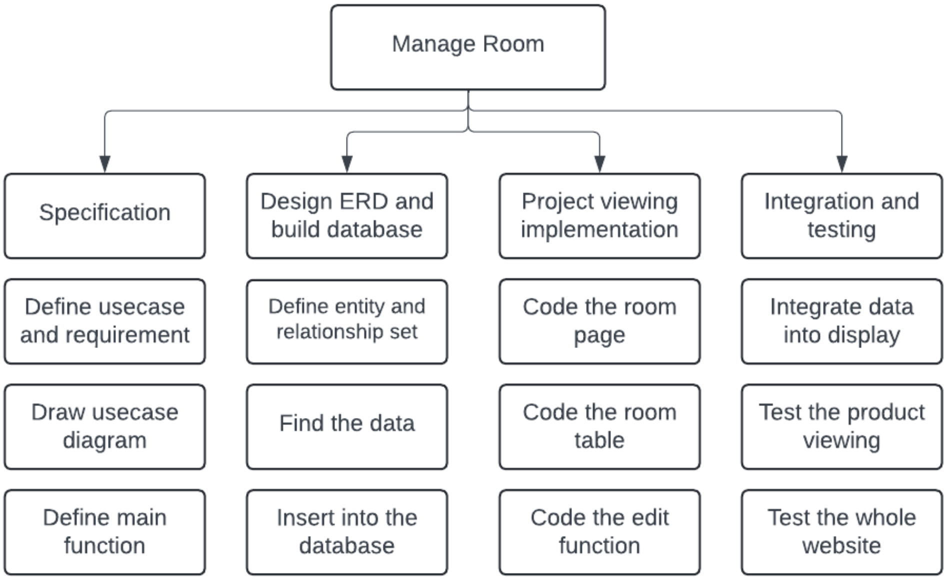


Figure 3.4

The tasks for Manage Rate are described in the ***Figure 3.5***:

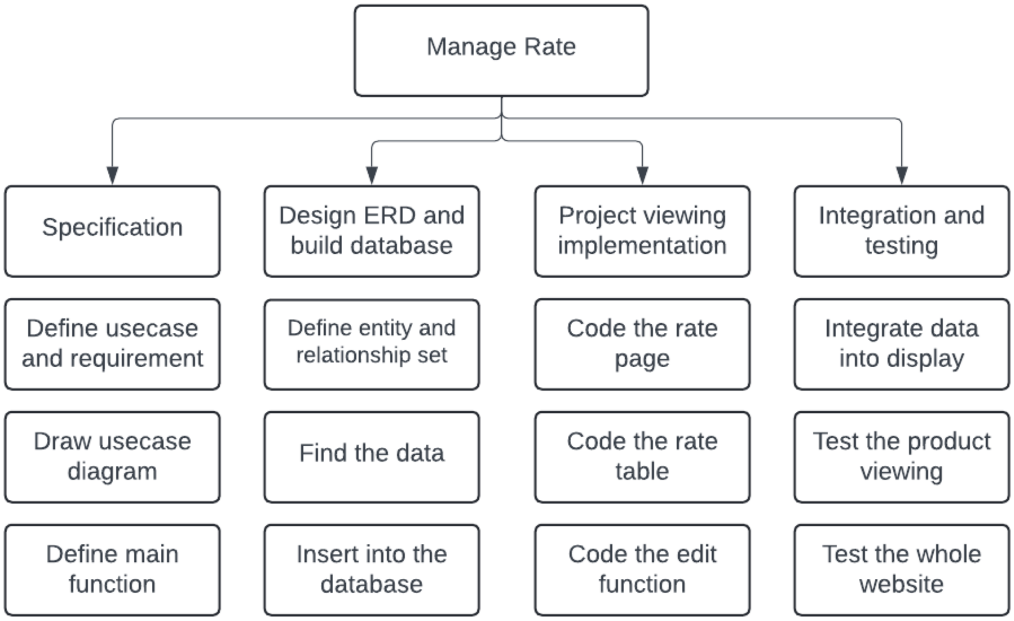


Figure 3.5

These tasks are expected to be completed and thoroughly tested after each iteration of the development process since we aim at applying the agile method rather than the traditional one which separates the steps.

4. DEVELOPMENT PROCESS

At The Passenger, we follow an Agile development methodology to ensure our motel management solution meets the dynamic needs of the hospitality industry. This iterative approach allows us to deliver high-quality features quickly and adapt to feedback in real-time. Through regular sprints, collaborative teamwork, and continuous testing, we refine our platform to provide the most efficient and user-friendly experience. Our commitment to agility ensures that The Passenger evolves with your business, offering reliable and innovative tools to manage your motel effectively.

Our method can be described in the ***Figure 4.1***:

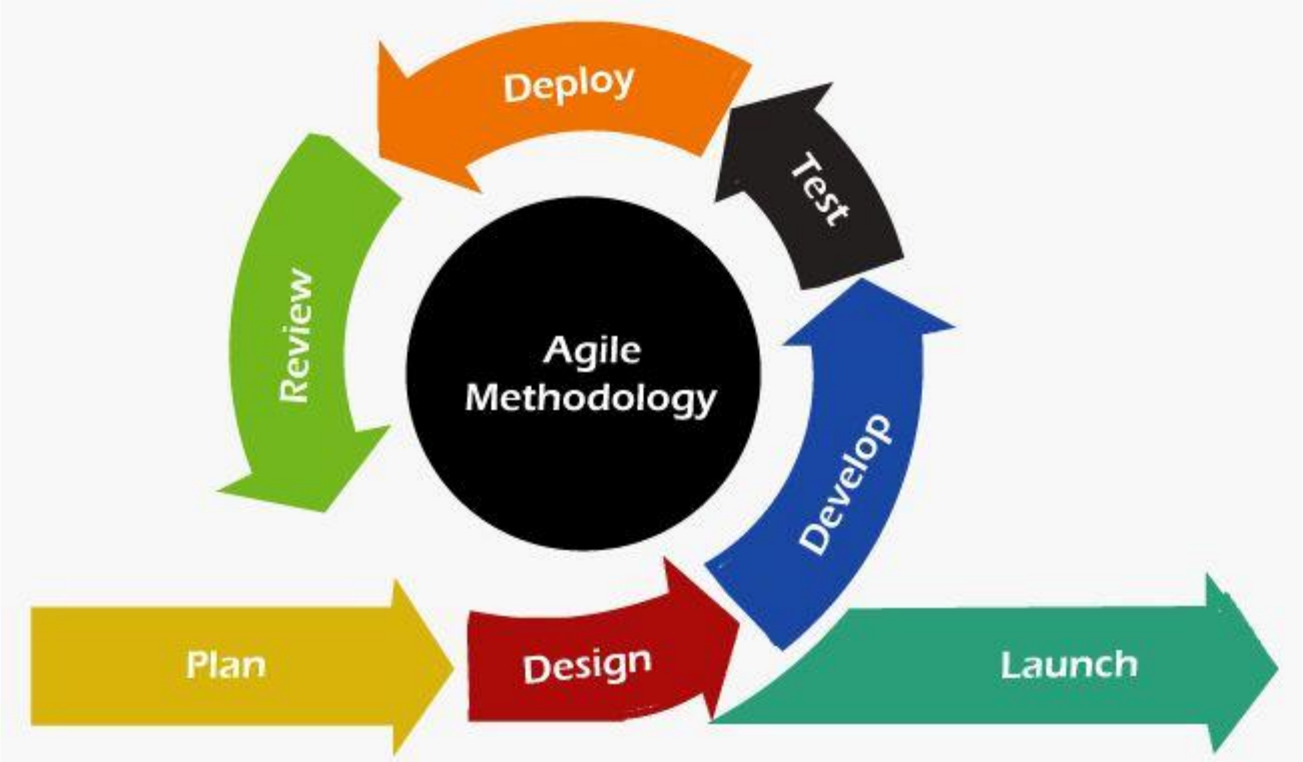


Figure 4.1

5. DEVELOPMENT ENVIRONMENT

Since this is a web-based product, the project is conducted using some Web Design and Programming Language under the model of MVC.

The following Programming Languages and Frameworks are used within our system:

1. **HTML:** to create an outline of the webpages
2. **CSS:** to design the looks of the webpages
3. **JavaScript:** to create animations for the webpages
4. **JSX:** to create the data table and the UI for the webpages.
5. **Reactjs:** the framework to develop the frontend.
6. **Nodejs:** the framework to develop the backend.
7. **MySQL:** to store the database of the system

The code is implemented according to the MVC model in the ***Figure 5.1***:

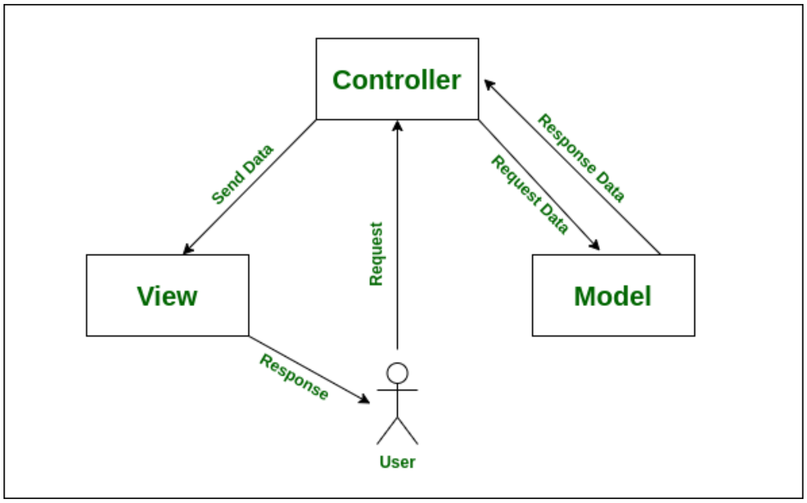


Figure 5.1

1. ***Model:*** the data of the system stored in the database and external system
2. ***View:*** the user interface
3. ***Control:*** the logics and algorithms used to develop a dynamic website with required functionalities

Moreover, we use UML tools to help us draw Entity Relationship Diagram as well Use Case needed in the project such as: LucidChart.

Here is the link of our shared Lucid used in this project: ***https://lucid.app/folder/invitations/accept/inv_e9d39fc6-2115-443b-a586-692ad2c2cc8c***

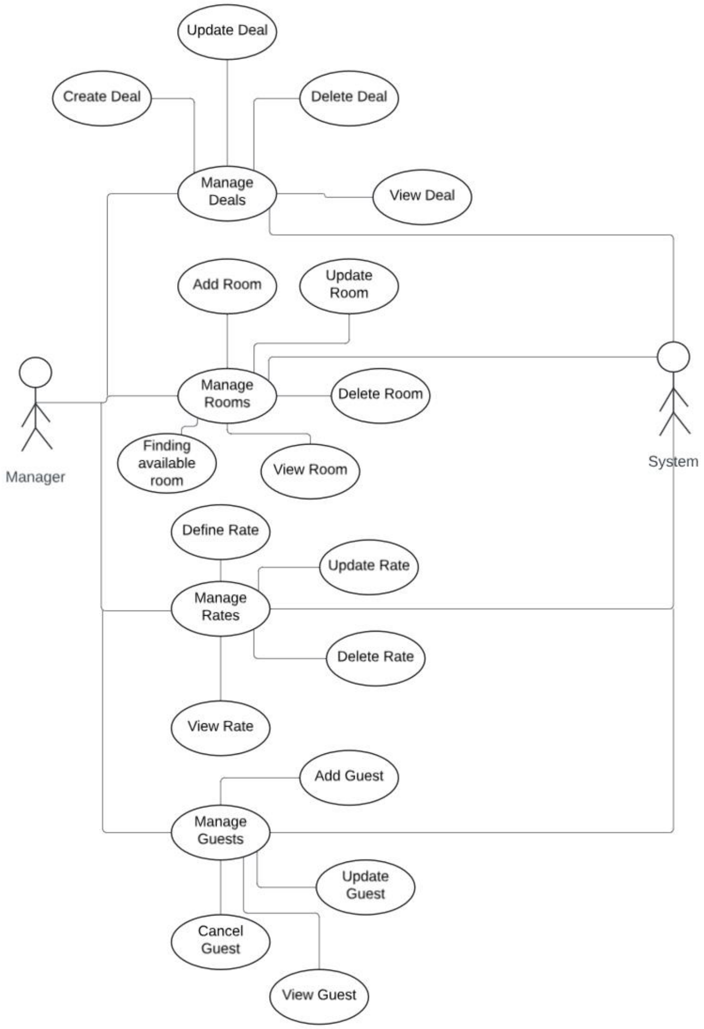
Moreover, we also make use of github to keep track of the overall progress as well as each team member's work performance.

II. REQUIREMENT ANALYSIS AND DESIGN

The Passenger begins with a thorough requirement analysis, engaging stakeholders to understand the unique needs of motel management. This ensures our solution addresses real-world challenges. In the design phase, we transform these insights into an intuitive, user-centric platform through wireframes, prototypes, and iterations. Our meticulous approach guarantees The Passenger delivers exceptional usability and performance.

1. REQUIREMENT ANALYSIS

USE CASE DIAGRAM



Use Case 1:

Name: Manage Deals

Identifier *UC1*

Inputs: The search data

Outputs:

1. The data table	[If homepage of deal]
2. The filtered data	[If search]
3. The update notification success	[If update]
4. The delete notification success	[If delete]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open the deal page	1.1. Display the data table.
2. Enter the data wanted to search	2.1. Check for the data. 2.2. Display the data table after filtered.
3. Update the data	3.1. Check for the data. 3.2. Enter the data wanted to update. 3.3. Update the database 3.4. Give a notification
4. Delete the data	4.1. Enter the ID for the data to delete 4.2. Update the database 4.3. Give a notification

Precondition

1. User accesses the website via the link.

Post condition

1. Display the data table or the notification.

User story: As a motel manager, I want to access the data table for the deal to register for the customer…

Use Case 2:

Name: Manage Rooms

Identifier *UC2*

Inputs: The search data

Outputs:

1. The data table	[If homepage of deal]
2. The filtered data	[If search]
3. The update notification success	[If update]
4. The delete notification success	[If delete]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open the room page	1.1. Display the data table.
2. Enter the data wanted to search	2.1. Check for the data. 2.2. Display the data table after filtered.
3. Update the data	3.1. Check for the data. 3.2. Enter the data wanted to update. 3.3. Update the database 3.4. Give a notification
4. Delete the data	4.1. Enter the ID for the data to delete 4.2. Update the database 4.3. Give a notification

Precondition

1. User accesses the website via the link.

Post condition

1. Display the data table or the notification.

User story: As a motel manager, I want to access the data table for the room to register for the customer…

Use Case 3:

Name: Manage Rates

Identifier *UC3*

Inputs: The search data

Outputs:

1. The data table	[If homepage of deal]
2. The filtered data	[If search]
3. The update notification success	[If update]
4. The delete notification success	[If delete]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open the rate page	1.1. Display the data table.
2. Enter the data wanted to search	2.1. Check for the data. 2.2. Display the data table after filtered.
3. Update the data	3.1. Check for the data. 3.2. Enter the data wanted to update. 3.3. Update the database 3.4. Give a notification
4. Delete the data	4.1. Enter the ID for the data to delete 4.2. Update the database 4.3. Give a notification

Precondition

1. User accesses the website via the link.

Post condition

1. Display the data table or the notification.

User story: As a motel manager, I want to access the data table for the ratel to give the customer an overview…

Use Case 4:

Name: Manage Guests

Identifier *UC4*

Inputs: The search data

Outputs:

5. The data table	[If homepage of deal]
6. The filtered data	[If search]
7. The update notification success	[If update]
8. The delete notification success	[If delete]

Basic Course

Actor: User (Customer/Store Manager)	System
1. Open the guest page	1.1. Display the data table.
2. Enter the data wanted to search	2.1. Check for the data. 2.2. Display the data table after filtered.
3. Update the data	3.1. Check for the data. 3.2. Enter the data wanted to update. 3.3. Update the database 3.4. Give a notification
4. Delete the data	4.1. Enter the ID for the data to delete 4.3. Update the database 4.3. Give a notification

Precondition

1. User accesses the website via the link.

Post condition

1. Display the data table or the notification.

User story: As a motel manager, I want to access the data table for the guests to inform, edit the customer in detail and register for new customer…

2. FUNCTIONAL REQUIREMENTS

Use Case 1: Manage Deals

1. *The Scope of the Work*

- This occurs in the sprint 4 in the process
- 4 tasks needed for this function
- 20 hours of effort is needed for this function

2. ***The Scope of the Product***: This is the main part of showing deal data function.

3. *Functional and Data Requirements*

a. *Functional Requirement*

_ Shall display the deal table which the user can see.

_ Shall achieve the information of the data which is in the database

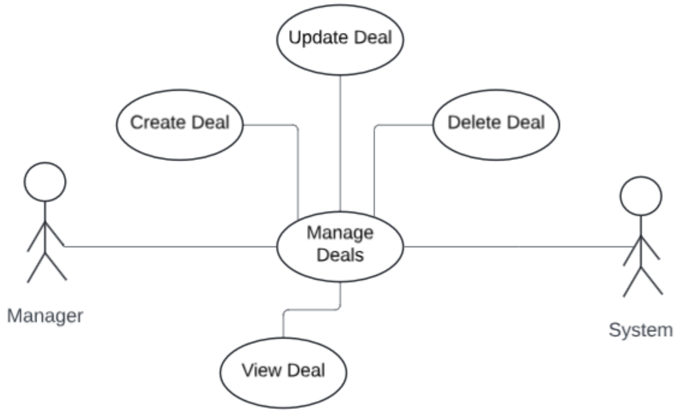
_ Shall be able to search

_ Shall be able to update

_ Shall be able to delete

b. *Data Requirement*: The data must be according to the database.

4. *Use Case Diagram*



Use Case 2: Manage Rooms

1. The Scope of the Work

- This occurs in the sprint 4 in the process
- 4 tasks needed for this function
- 20 hours of effort is needed for this function

2. The Scope of the Product: This is the main part of showing room data function.

3. Functional and Data Requirements

a. *Functional Requirement*

_ Shall display the room table which the user can see.

_ Shall achieve the information of the data which is in the database

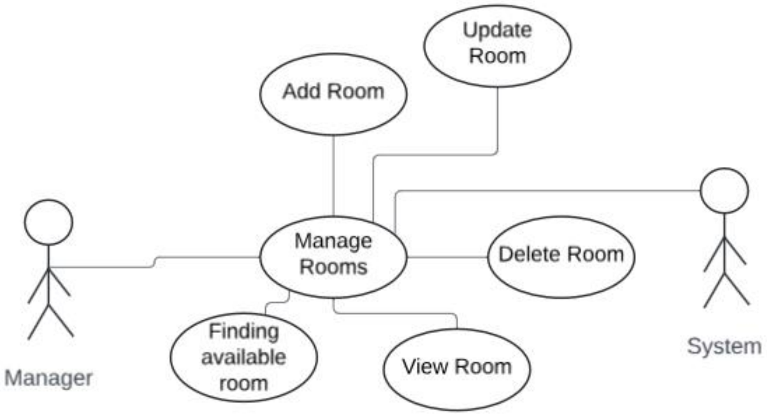
_ Shall be able to search

_ Shall be able to update

_ Shall be able to delete

b. *Data Requirement*: The data must be according to the database.

4. Use Case Diagram



Use Case 3: Manage Rates

1. The Scope of the Work

i. This occurs in the sprint 5 in the process

ii. 4 tasks needed for this function

iii. 20 hours of effort is needed for this function

2. The Scope of the Product: This is the main part of showing rate data function.

3. Functional and Data Requirements

a. *Functional Requirement*

_ Shall display the rate table which the user can see.

_ Shall achieve the information of the data which is in the database

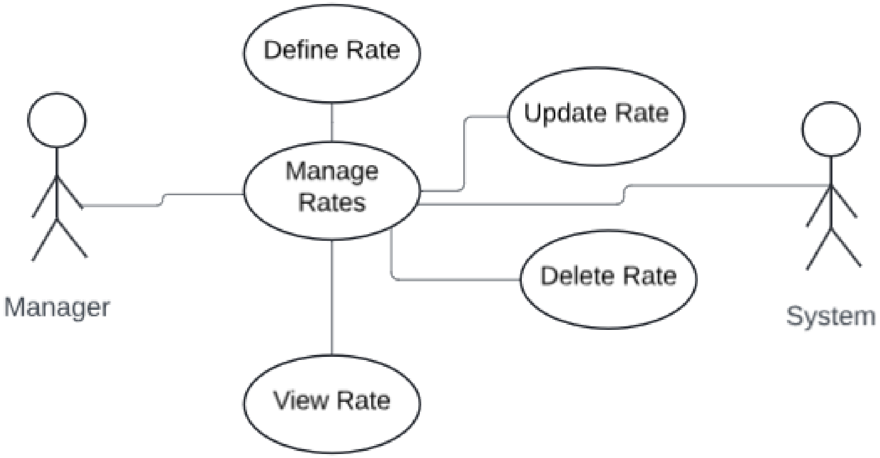
_ Shall be able to search

_ Shall be able to update

_ Shall be able to delete

b. *Data Requirement*: The data must be according to the database.

4. Use Case Diagram



Use Case 4: Manage Guests

1. The Scope of the Work

ιϖ. This occurs in the sprint 5 in the process

ϖ. 4 tasks needed for this function

ϖι. 20 hours of effort is needed for this function

2. The Scope of the Product: This is the main part of showing guest data function.

3. Functional and Data Requirements

a. *Functional Requirement*

_ Shall display the guest table which the user can see.

_ Shall achieve the information of the data which is in the database

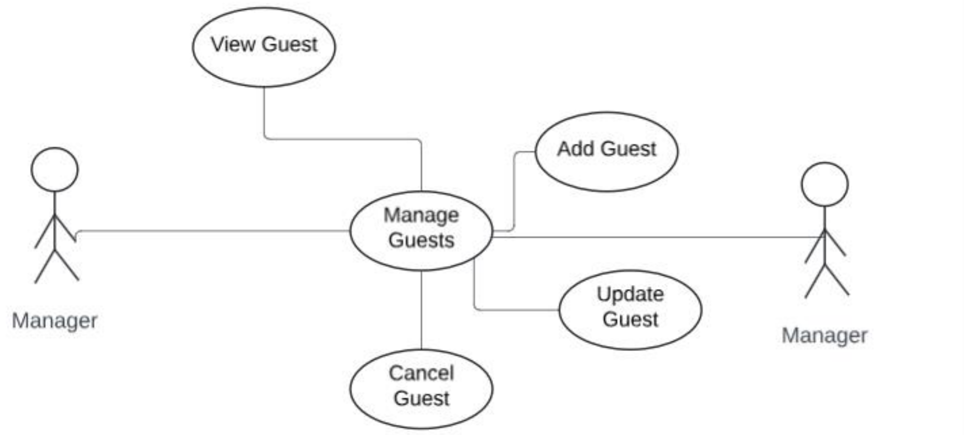
_ Shall be able to search

_ Shall be able to update

_ Shall be able to delete

b. *Data Requirement*: The data must be according to the database.

4. Use Case Diagram



3. NON-FUNCTIONAL REQUIREMENTS

a. Operational requirements:

ι. Ensure all software components operate effectively.

ιι. Maintain licensed hosting to handle all requests efficiently.

ιιι. Appoint an administrator to regularly update the software and check for system errors.

b. Legal requirements:

ι. Cite all integrated components to avoid copyright violations.

ιι. Keep proprietary source code confidential to prevent unauthorized use.

c. Usability requirements:

ι. Ensure the system is easy for users to learn, input data, and interpret outputs.

ιι. Provide well-structured user manuals.

ιιι. Include informative error messages with hints for solutions.

ιϖ. Offer comprehensive help facilities.

ϖ. Design intuitive graphical user interfaces that are easy to learn and navigate.

ϖι. Ensure tasks can be accomplished efficiently with minimal user errors.

d. Humanity Requirements:

ι. Assume users have little to no computer background.

ιι. Create detailed documentation for easy understanding.

ιιι. Design user-friendly graphical interfaces that are easy to learn and use.

ιϖ. Ensure the interface is attractive and responsive to enhance user experience.

e. Performance Requirements:

Performance Requirements:

a. ***Response Requirements:***
   - I. Ensure fast login response times for both customers and managers.
   - II. Minimize the time required for database updates, enabling seamless changes across multiple tables.
   - III. Ensure quick processing of services offered by the system, with the capability to handle 10 different transactions simultaneously.
   - IV. Achieve fast website loading times, with the ability to correctly generate and load over 50 products per page promptly.

b. ***Throughput Requirements:***
   - I. Handle a large volume of requests from multiple customers simultaneously without delays, supporting over 1000 visitors browsing the store page at the same time.
   - II. Enable quick handling of multiple database modifications.

c. ***Availability Requirements:***
   - I. Ensure efficient memory management, utilizing programming language garbage collection to maintain system availability.

Description:

Our motel management system is designed to handle heavy traffic, with the potential for thousands of visitors each day. To ensure smooth operation, the server must efficiently manage all incoming requests. The booking system must process multiple reservations simultaneously without delays or overloads. The platform is optimized to handle various requests from numerous customers using different operating systems without any postponement.

f. Maintainability Requirements:

The motel management system requires continuous updates and changes. Users have limited technical knowledge, so it's crucial to develop the system for easy manipulation by both users and maintenance teams.

i. Provide user-friendly functions to update and manage product and customer information.
ii. Write clear and well-commented source code to enhance understanding.
iii. Design a clear and stable database that facilitates easy maintenance.

g. Support Requirements

Users, particularly motel managers, lack technical expertise, necessitating robust support mechanisms from the development team:

i. Offer a hotline for immediate feedback from motel managers in case of issues.
ii. Provide remote maintenance tools (such as TeamViewer or remote-control software) for direct support.
iii. Conduct regular system checks and maintenance (monthly or every 2-3 months) to ensure stability.

h. Security requirements:

ι. Ensure data protection against unauthorized access.

ιι. Maintain system integrity to prevent accidental or malicious damage.

ιιι. Restrict system data changes to administrators only.

ιϖ. Log all modification events with details such as date, time, user, action, object, prior value, and new value...

v. Encrypt all communications between the data server and clients (e.g., SSL, SSH for HTTP).

ϖι. Encrypt customer information before storing it in the database.

ϖιι. Backup all system data, including products and customer information, every 24 hours with copies stored securely off-site.

ϖιιι. Information of the customers must be encrypted before adding to the database.

ιξ. Backup all system data, including products and customer information, every 24 hours with copies stored securely off-site.

ξ. Enforce privacy policies to restrict third-party access to customer information.

i. Interface requirements:

ι. Describe technical requirements affecting interfaces, including protocol management, scheduling, directory services, broadcasts, message types, error and buffer management, and security, each assigned a unique ID.

ιι. Some non-functional requirements of interface:

+ Buttons should visually represent their functionalities with appropriate symbols.

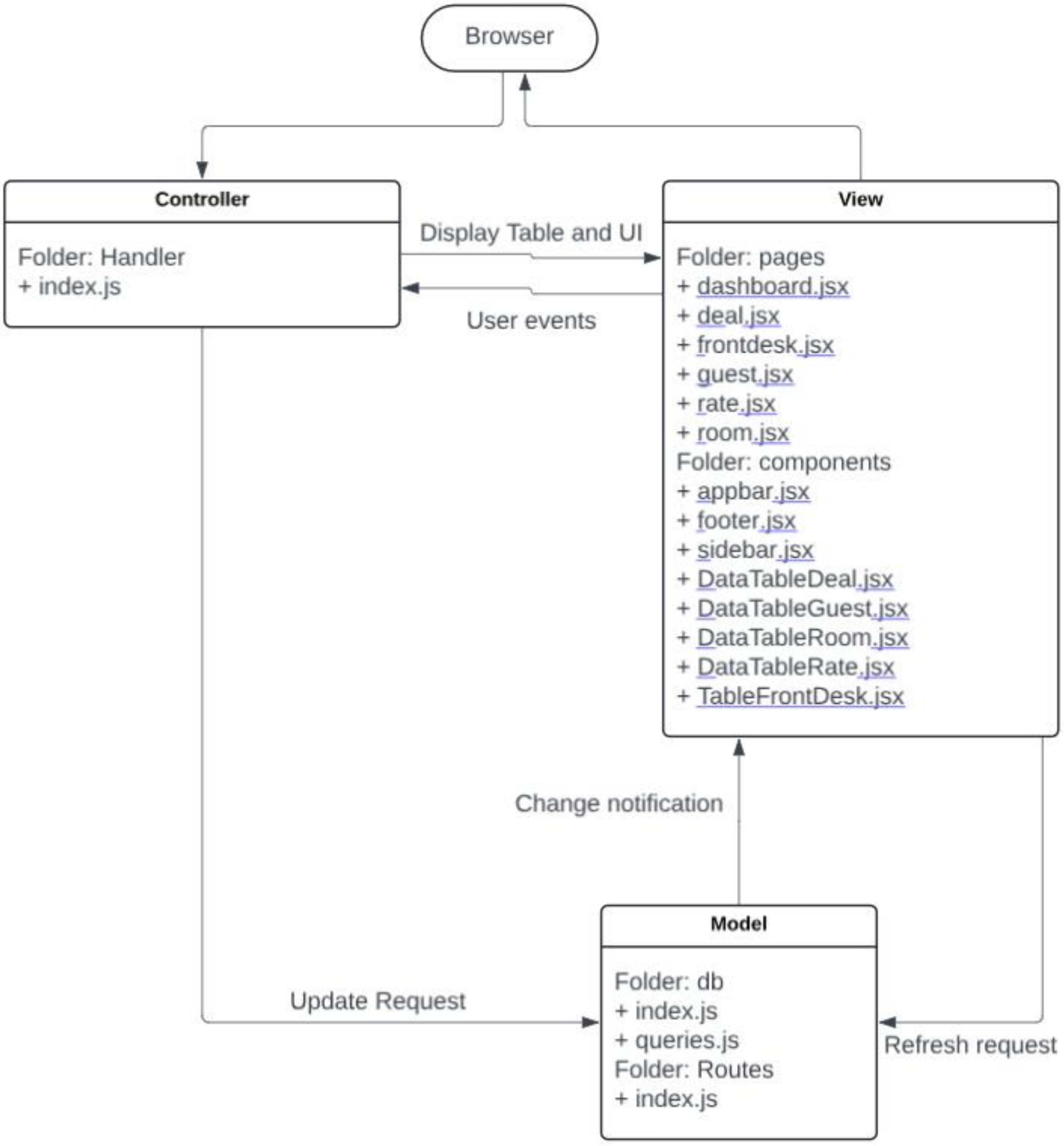
+ Sales reports must be generated for every successful transaction.

+ Java should support various Look and Feel libraries for website aesthetics.

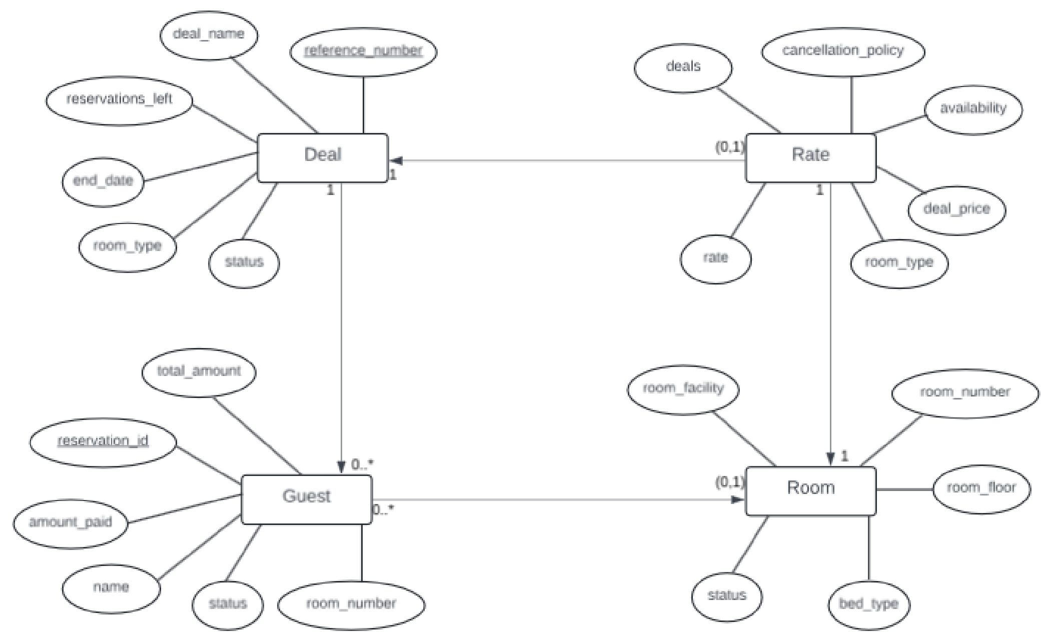
+ Provide language options for both native and foreign visitors/customers.

III. DESIGN

System Architecture Model

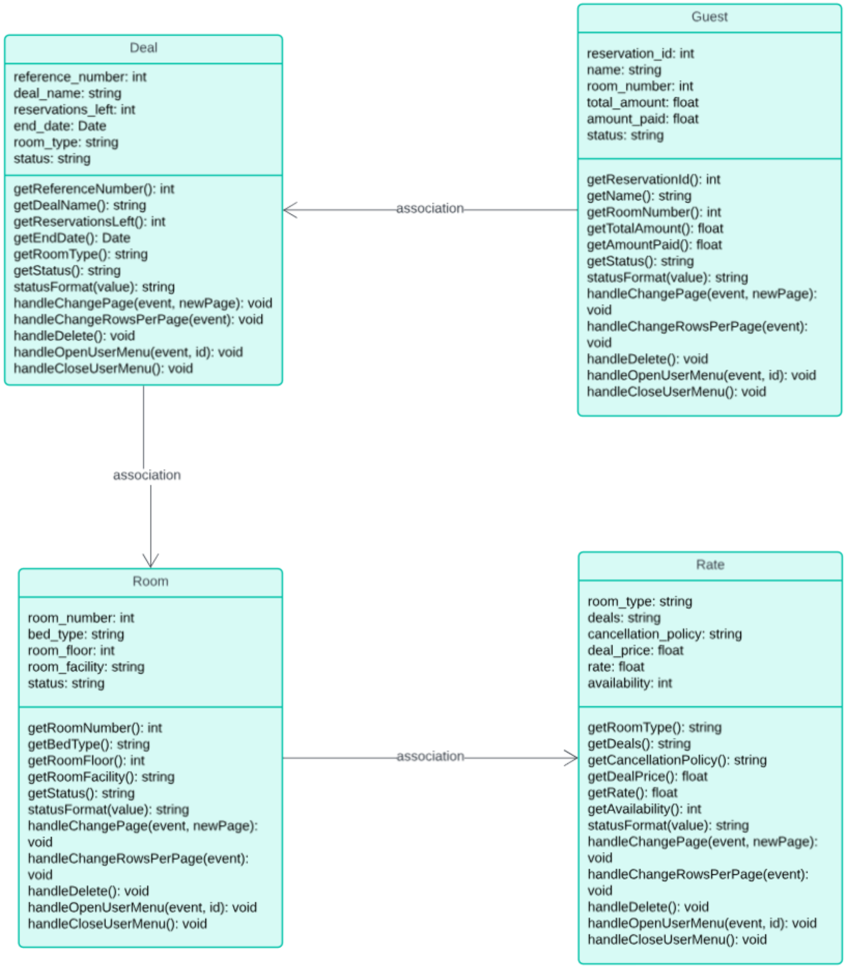


Entity-Relationship Diagram (ERD)

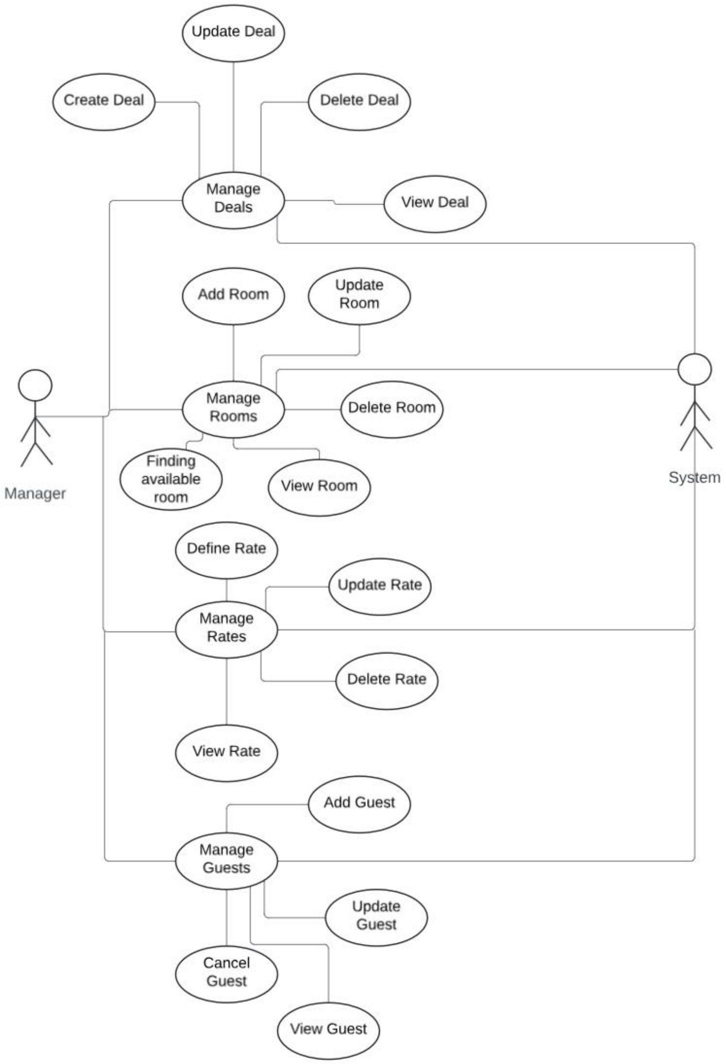


Deal(reference_number, deal_name, reservations_left, end_date, room_type, status)
Guest(reservation_id, name, room_number, total_amount, amount_paid, status)
Rate(room_type, deals, cancellation_policy, deal_price, rate, availability)
Room(room_number, bed_type, room_floor, room_facility, status)

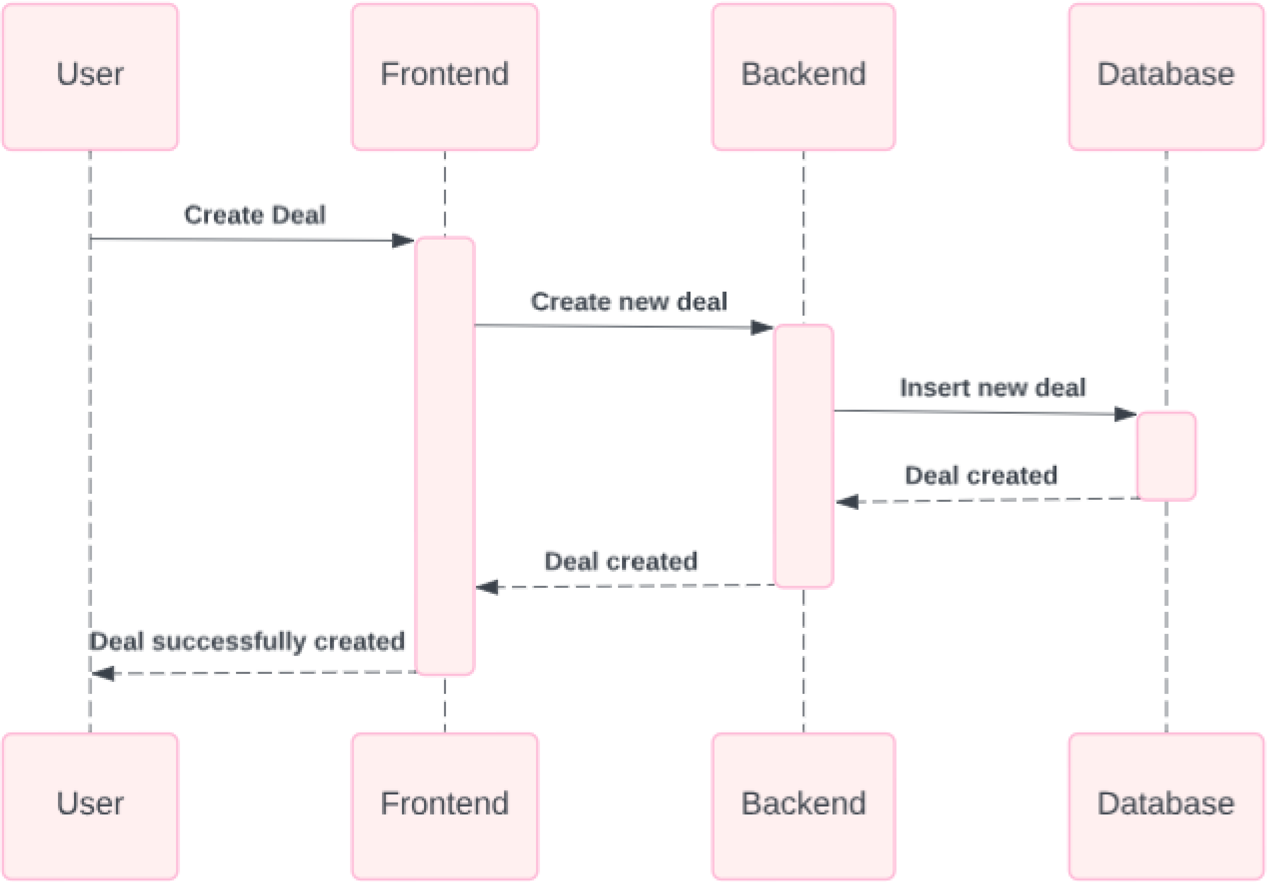
Class Diagram



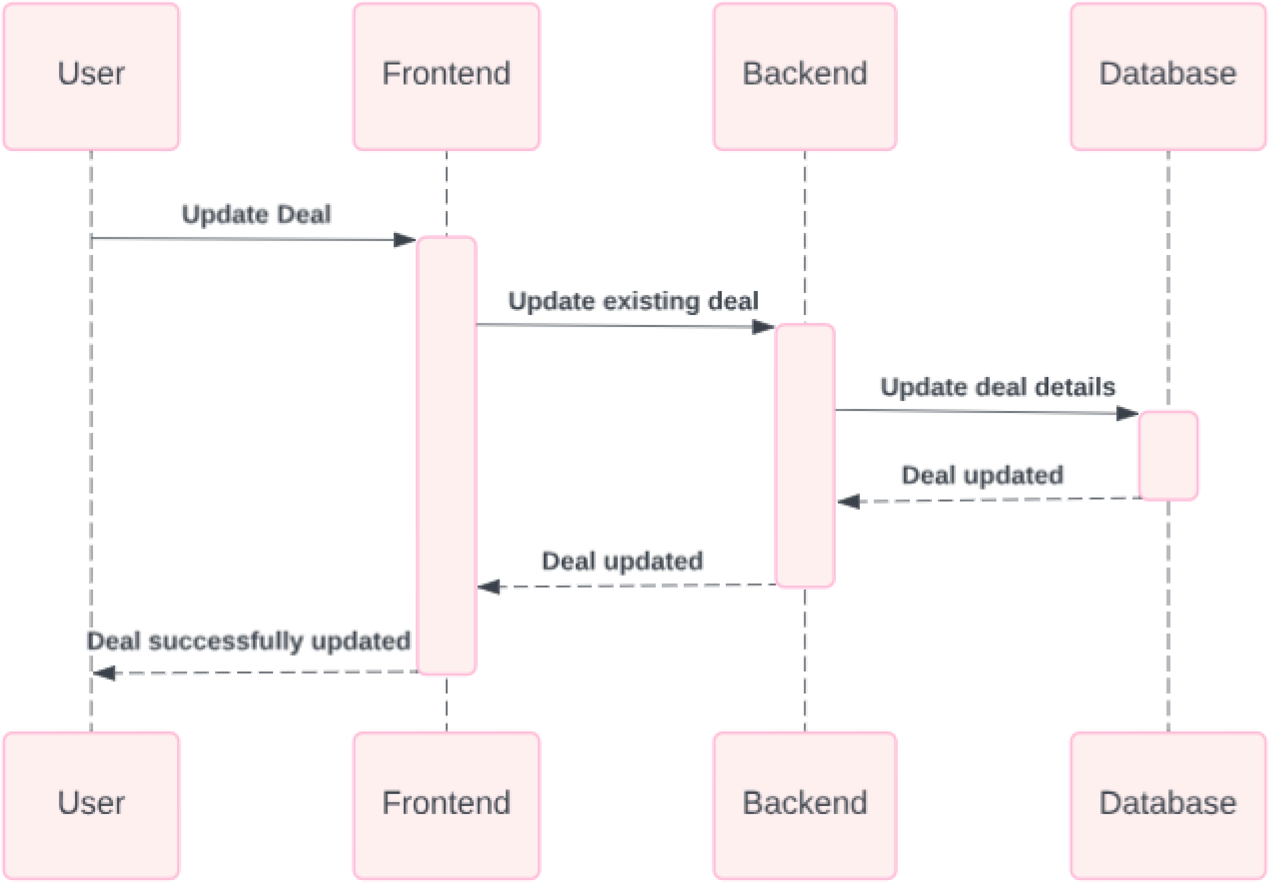
Use Case Diagram



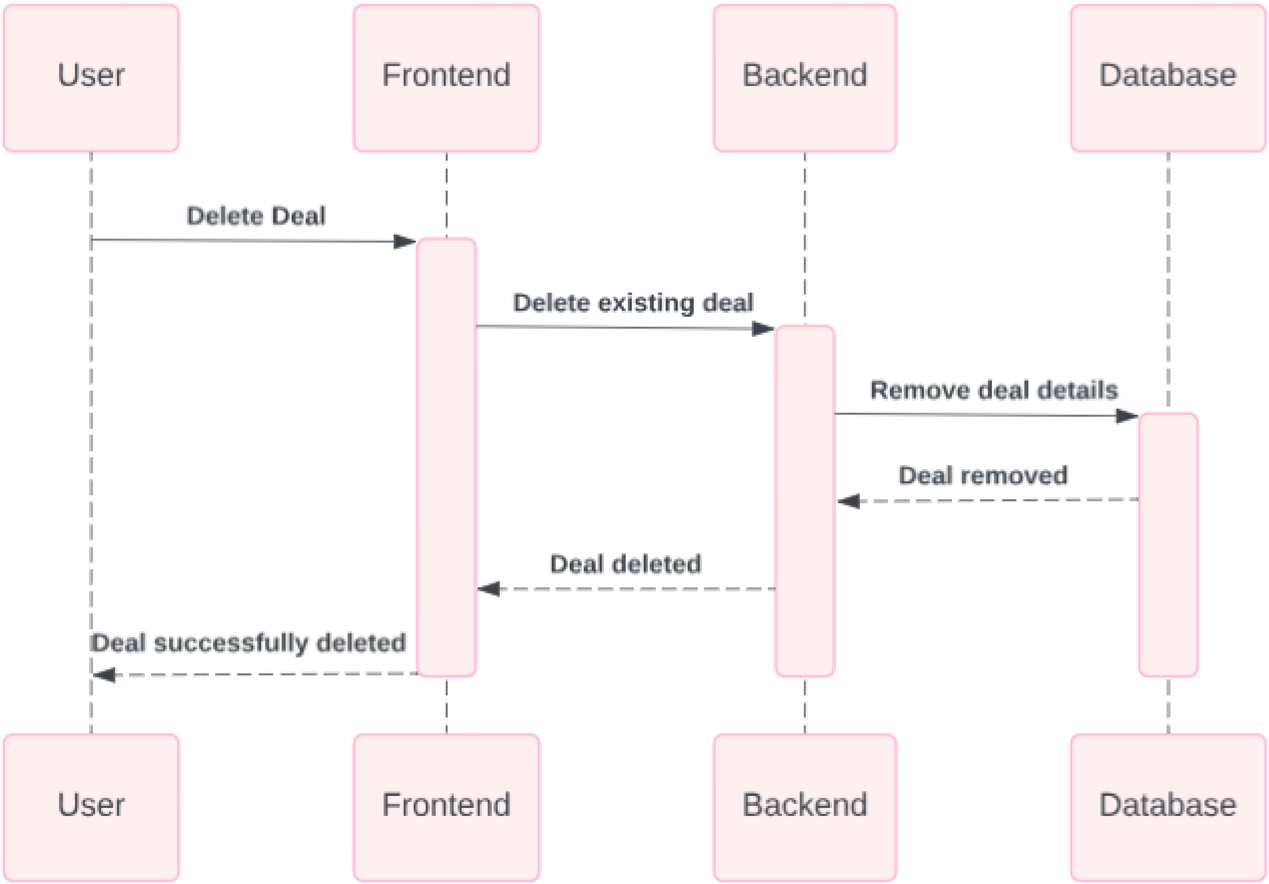
Sequence Diagram 1: Create Deal



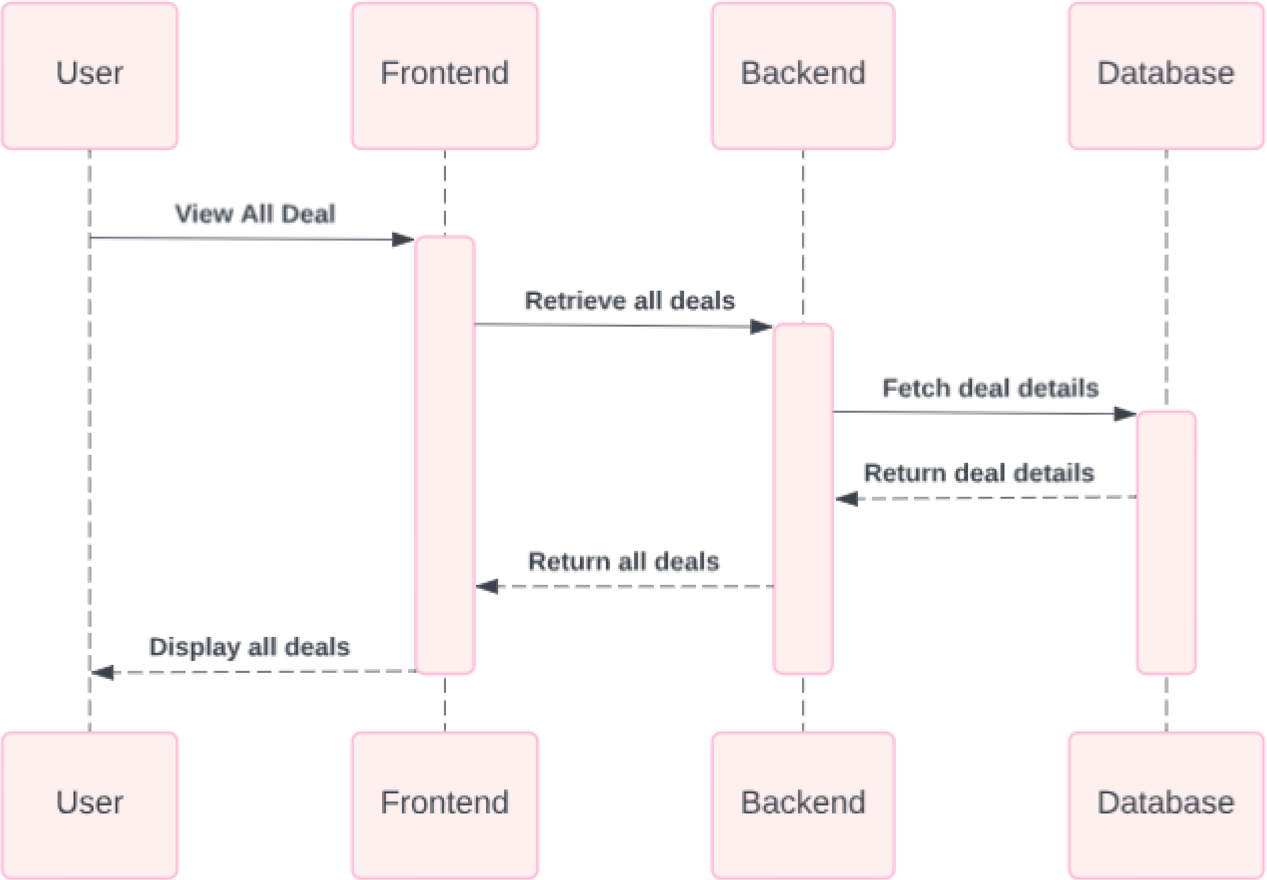
Sequence Diagram 2: Update Deal



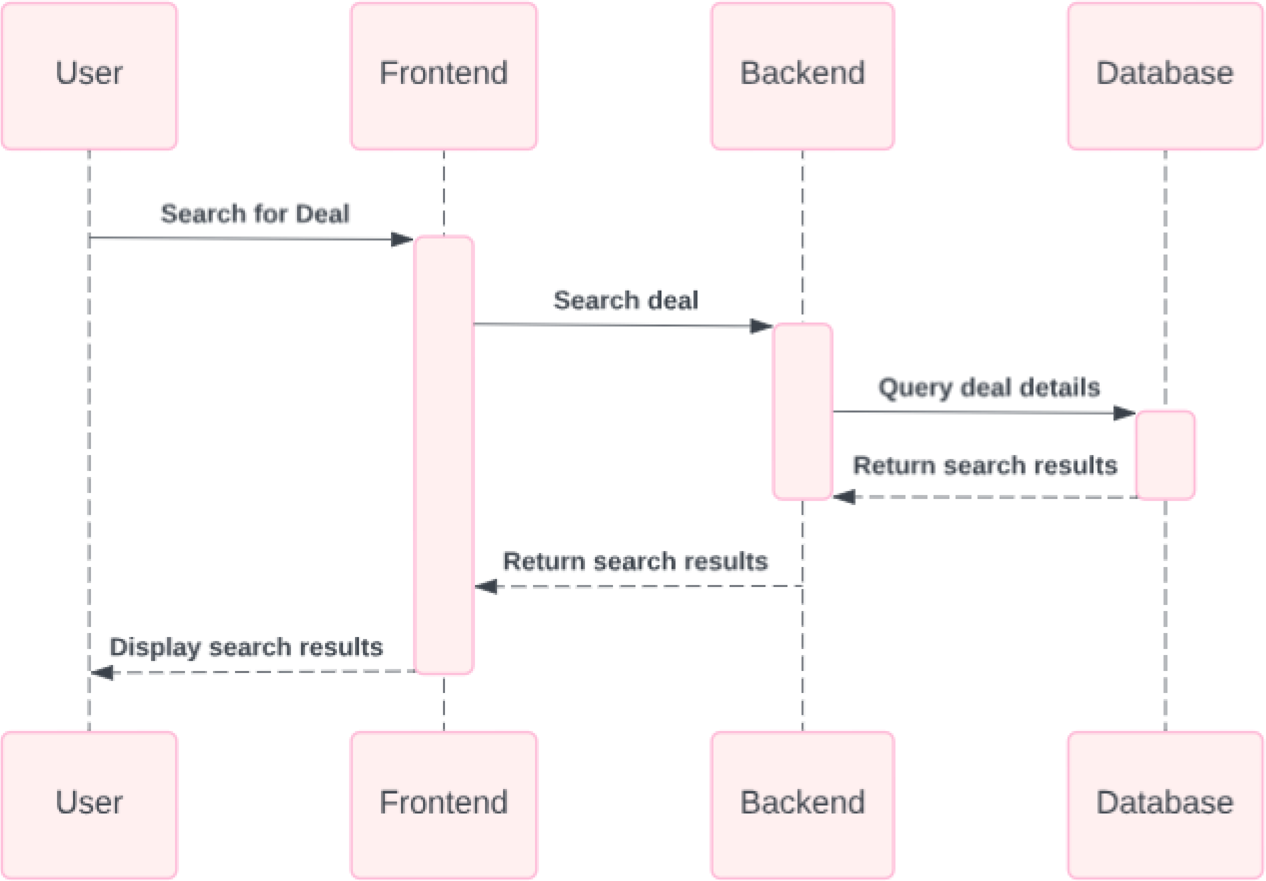
Sequence Diagram 3: Delete Deal



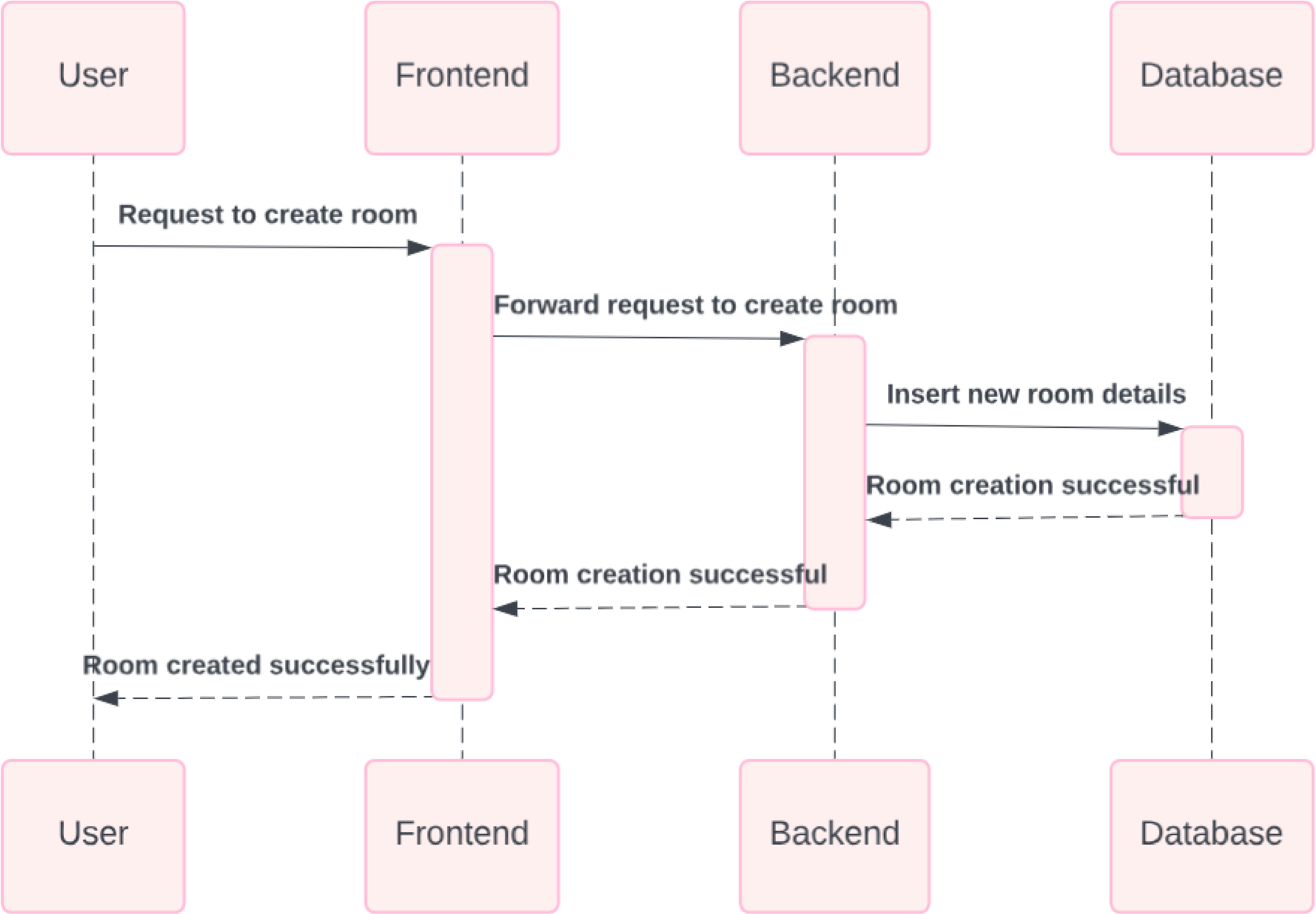
Sequence Diagram 4: View All Deal



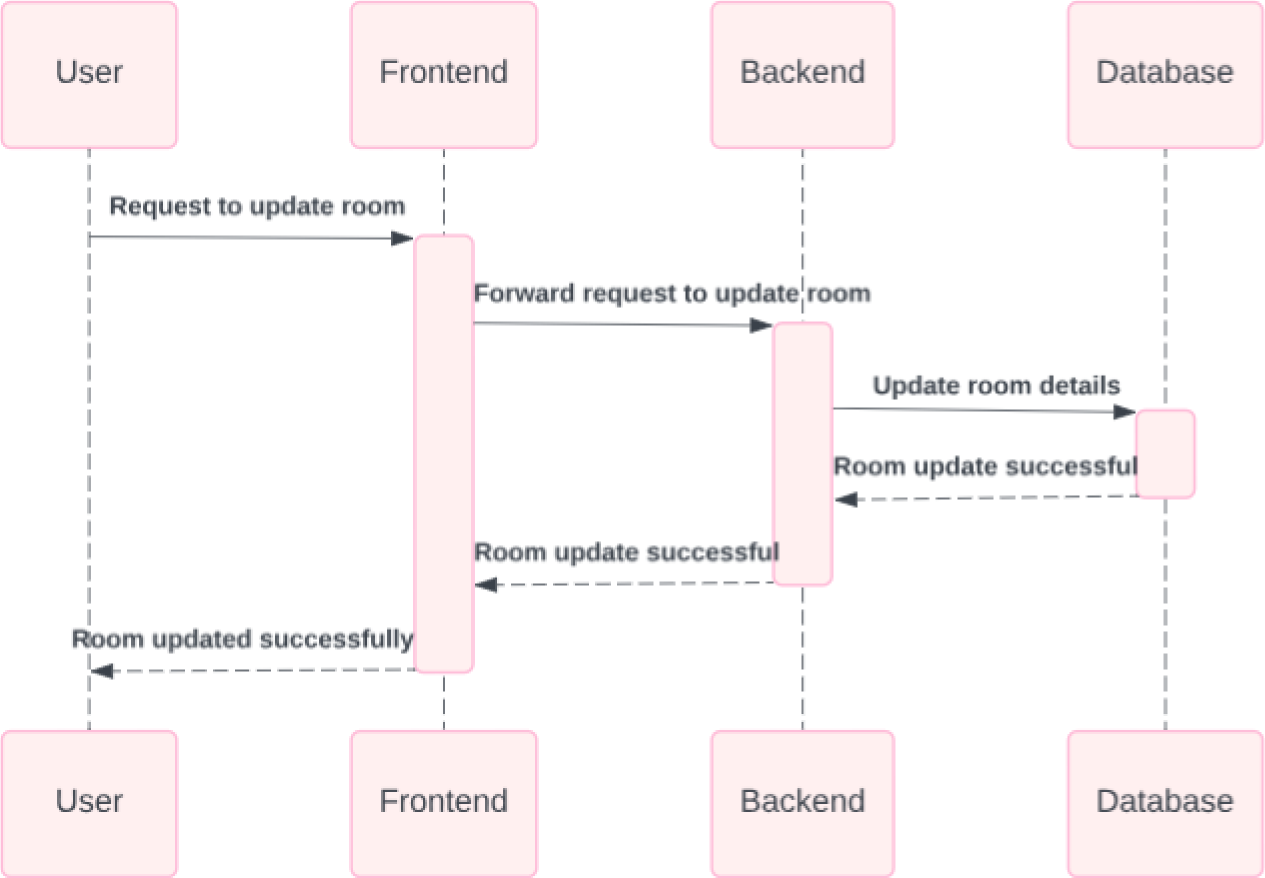
Sequence Diagram 5: Search for Deal



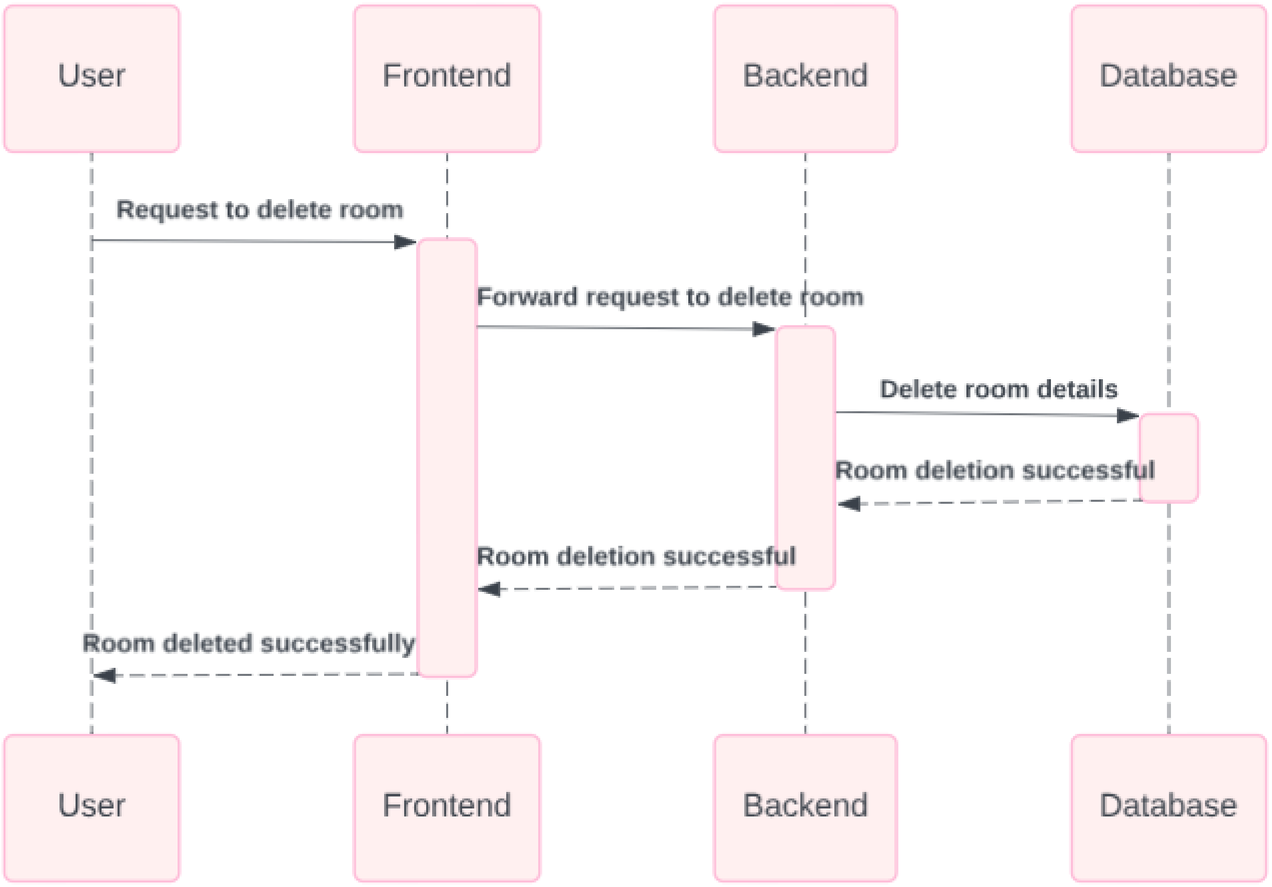
Sequence Diagram 6: Create Room



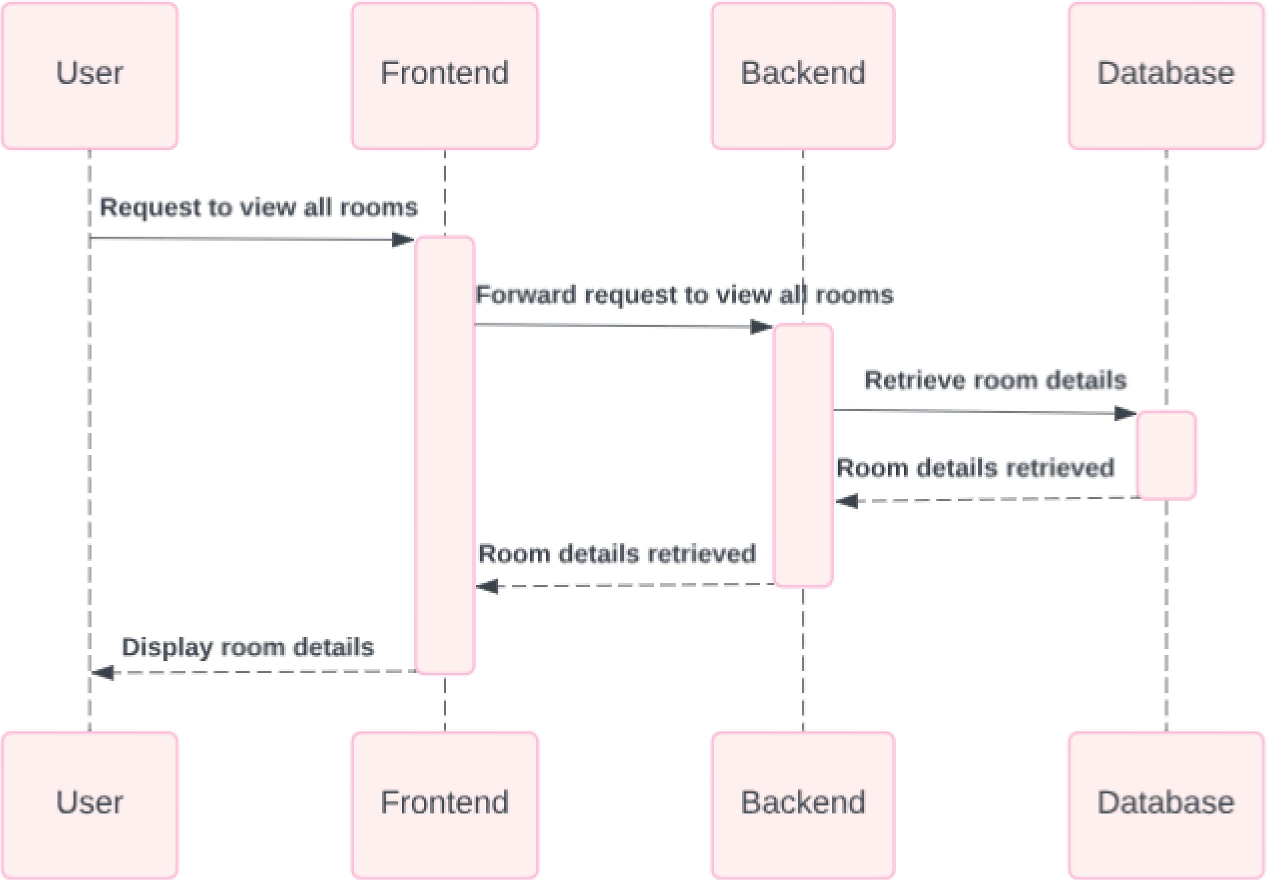
Sequence Diagram 7: Update Room



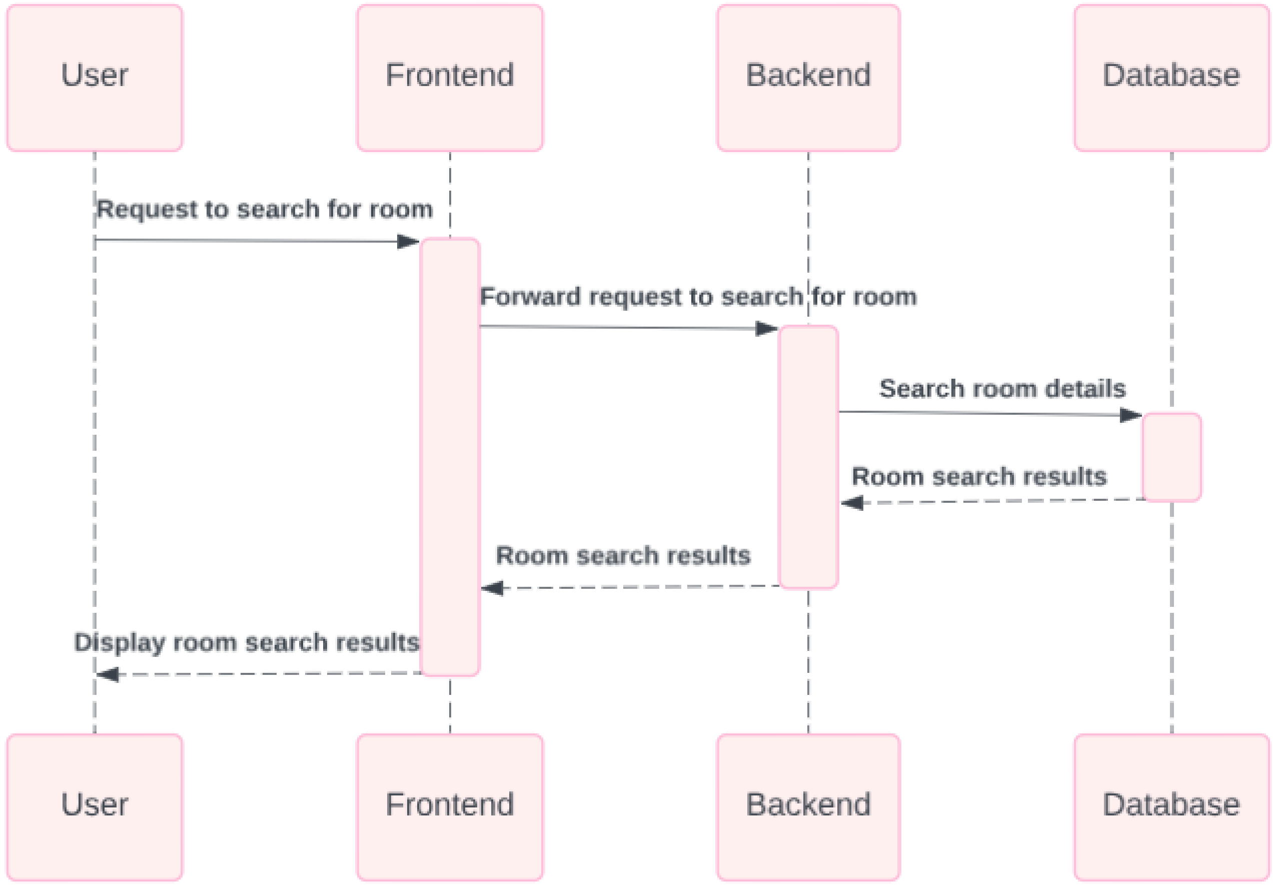
Sequence Diagram 8: Delete Room



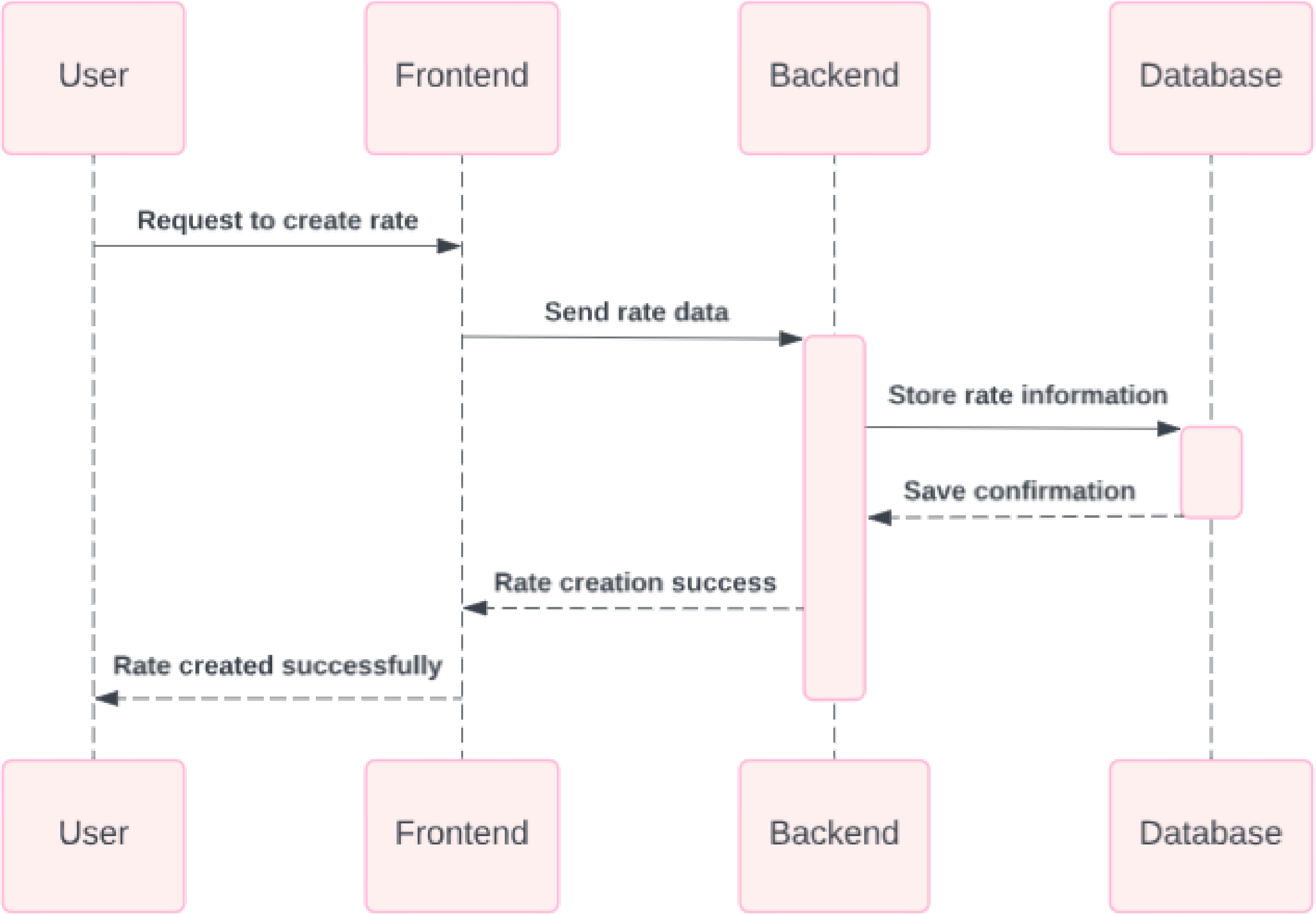
Sequence Diagram 9: View all Room



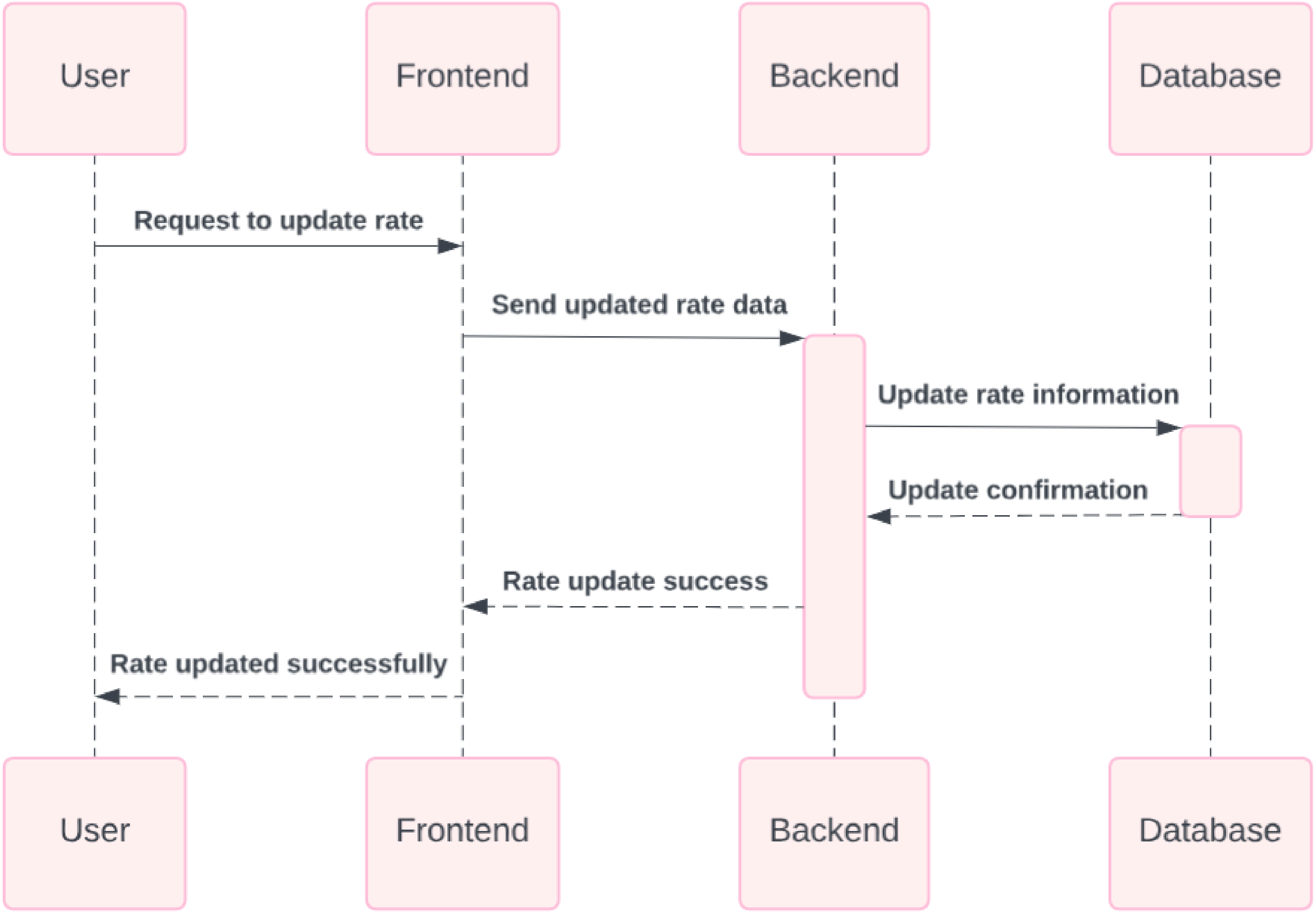
Sequence Diagram 10: Search for Room



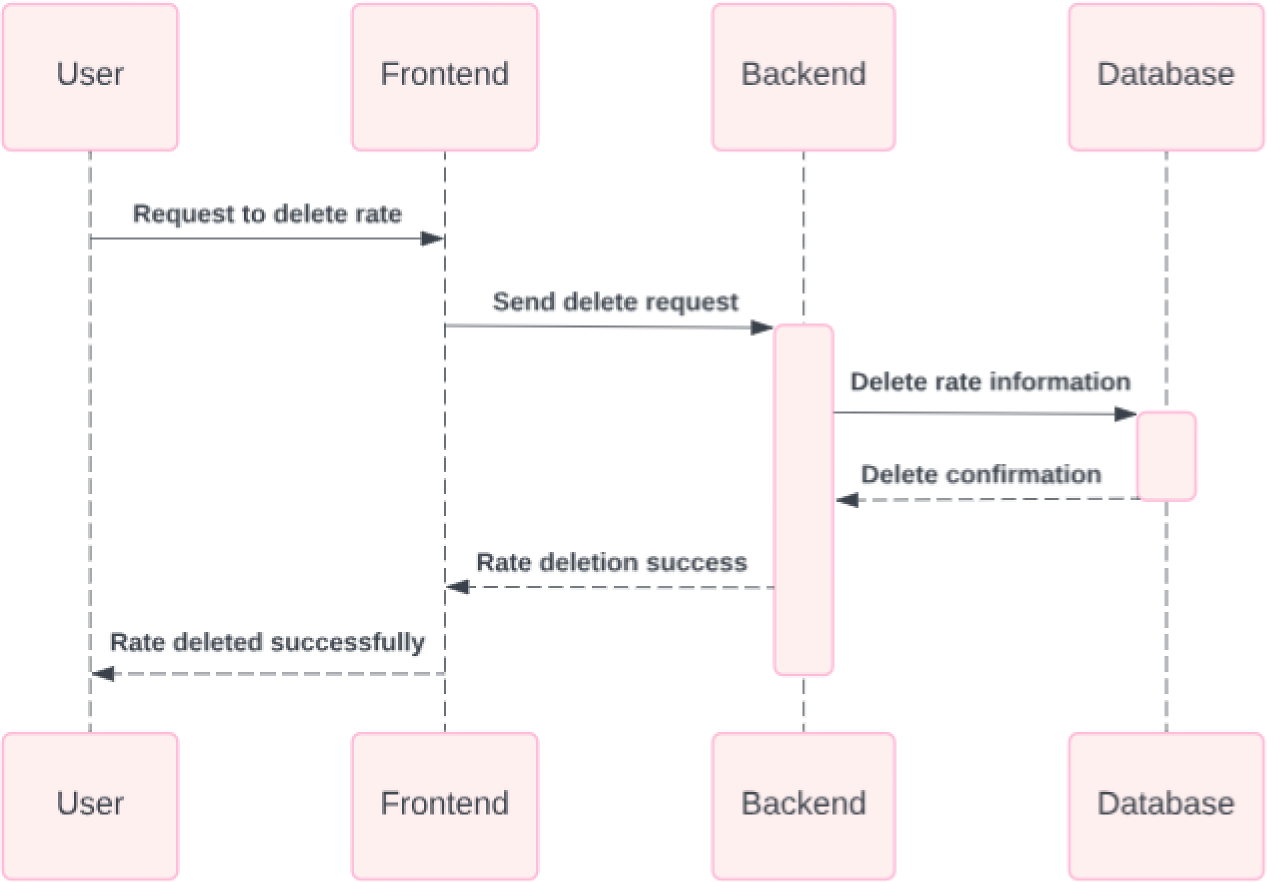
Sequence Diagram 11: Create Rate



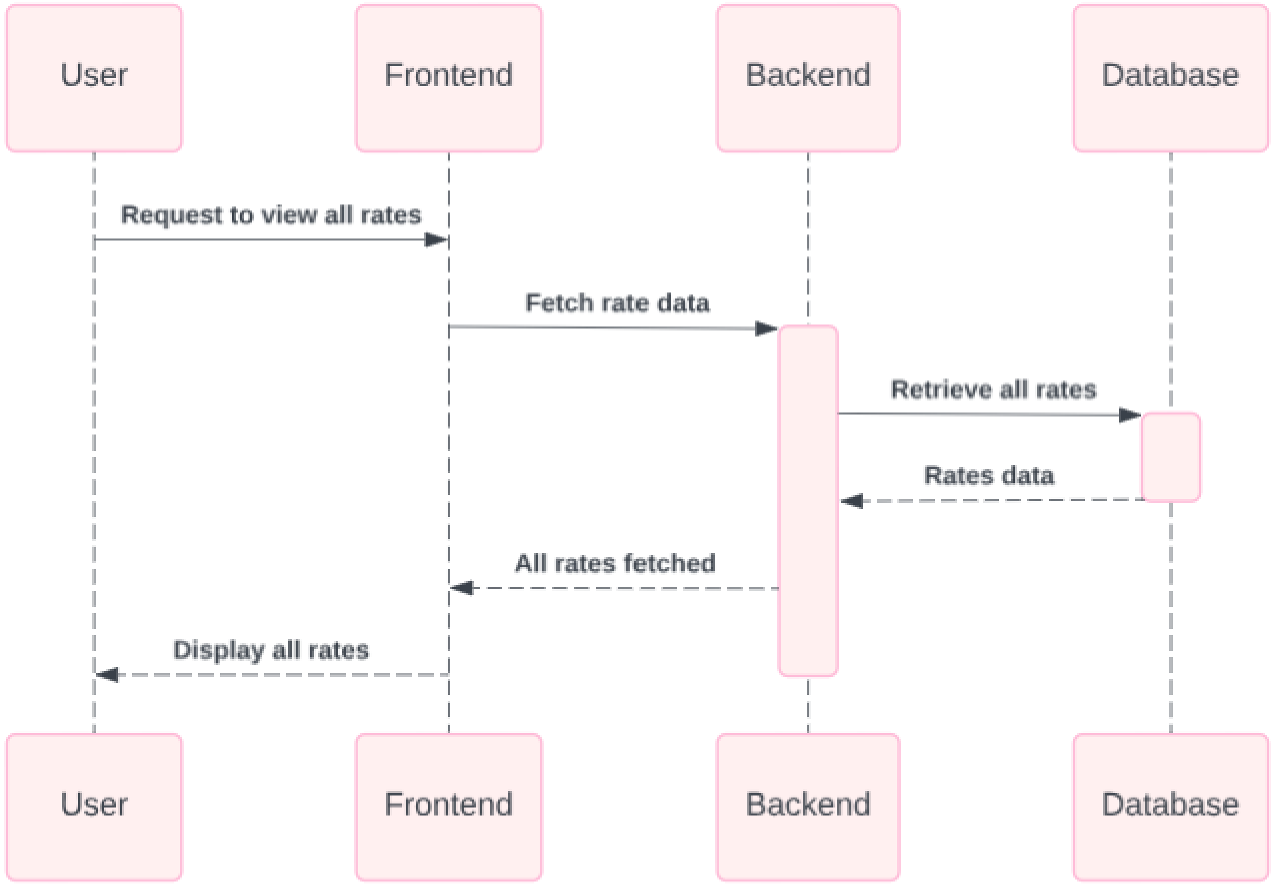
Sequence Diagram 12: Update Rate



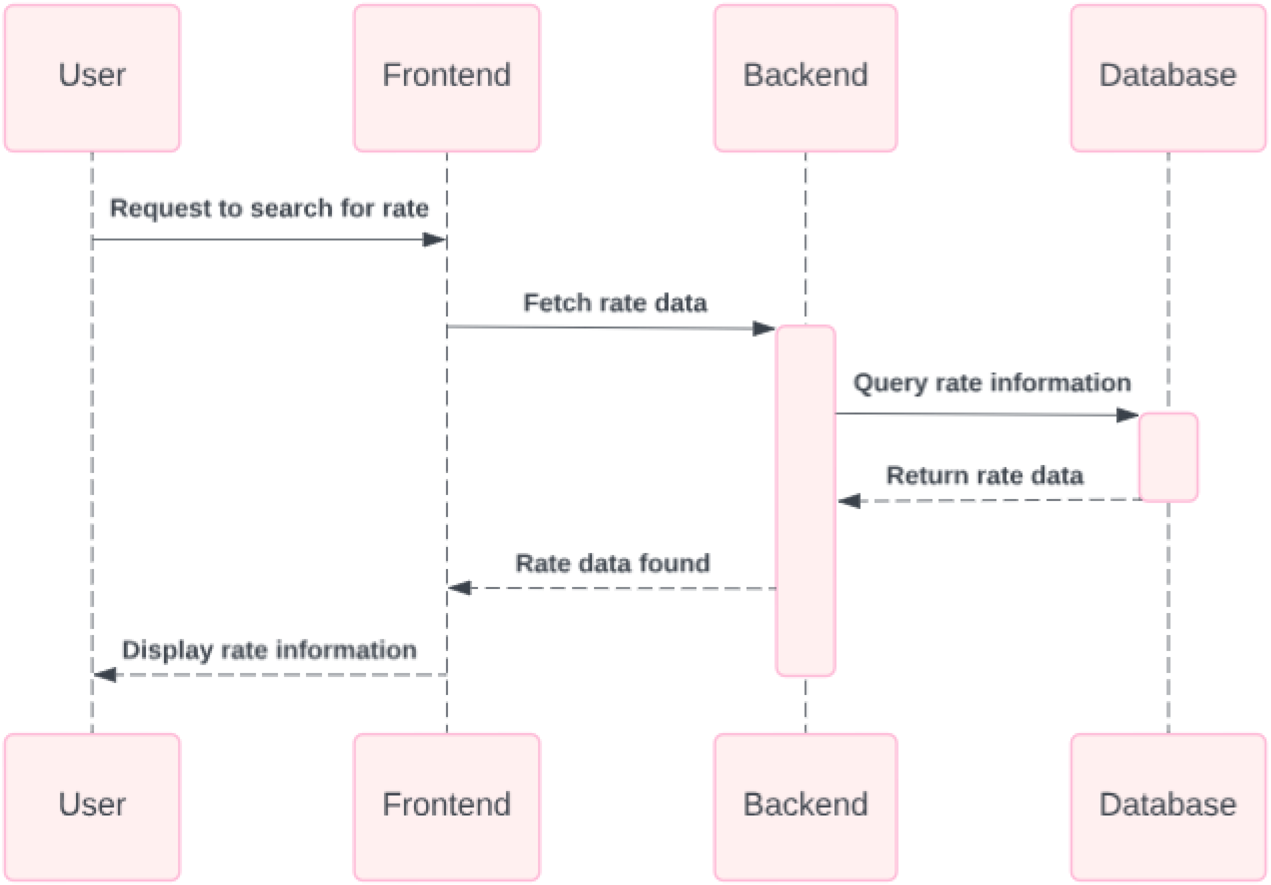
Sequence Diagram 13: Delete Rate



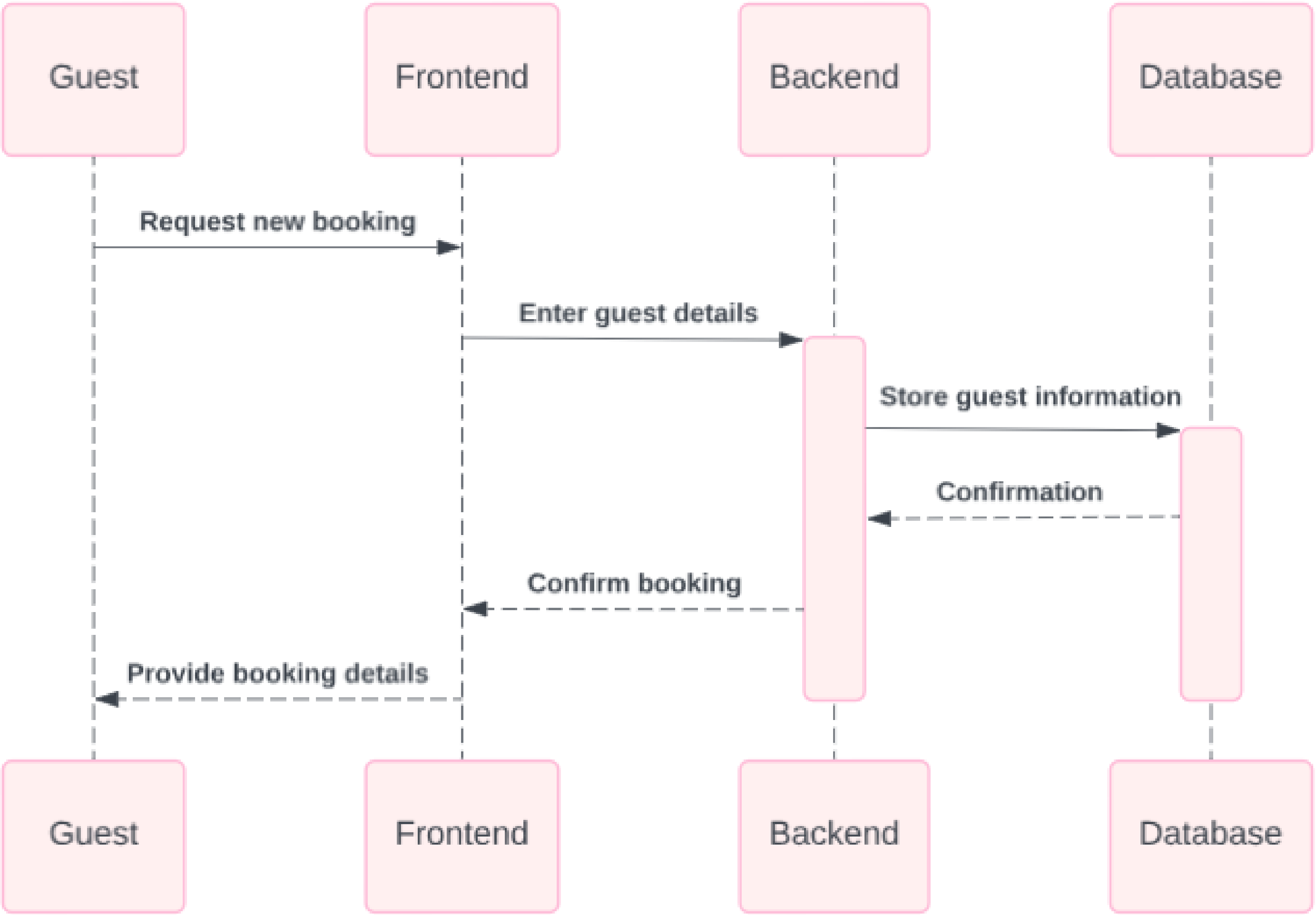
Sequence Diagram 14: View All Rate



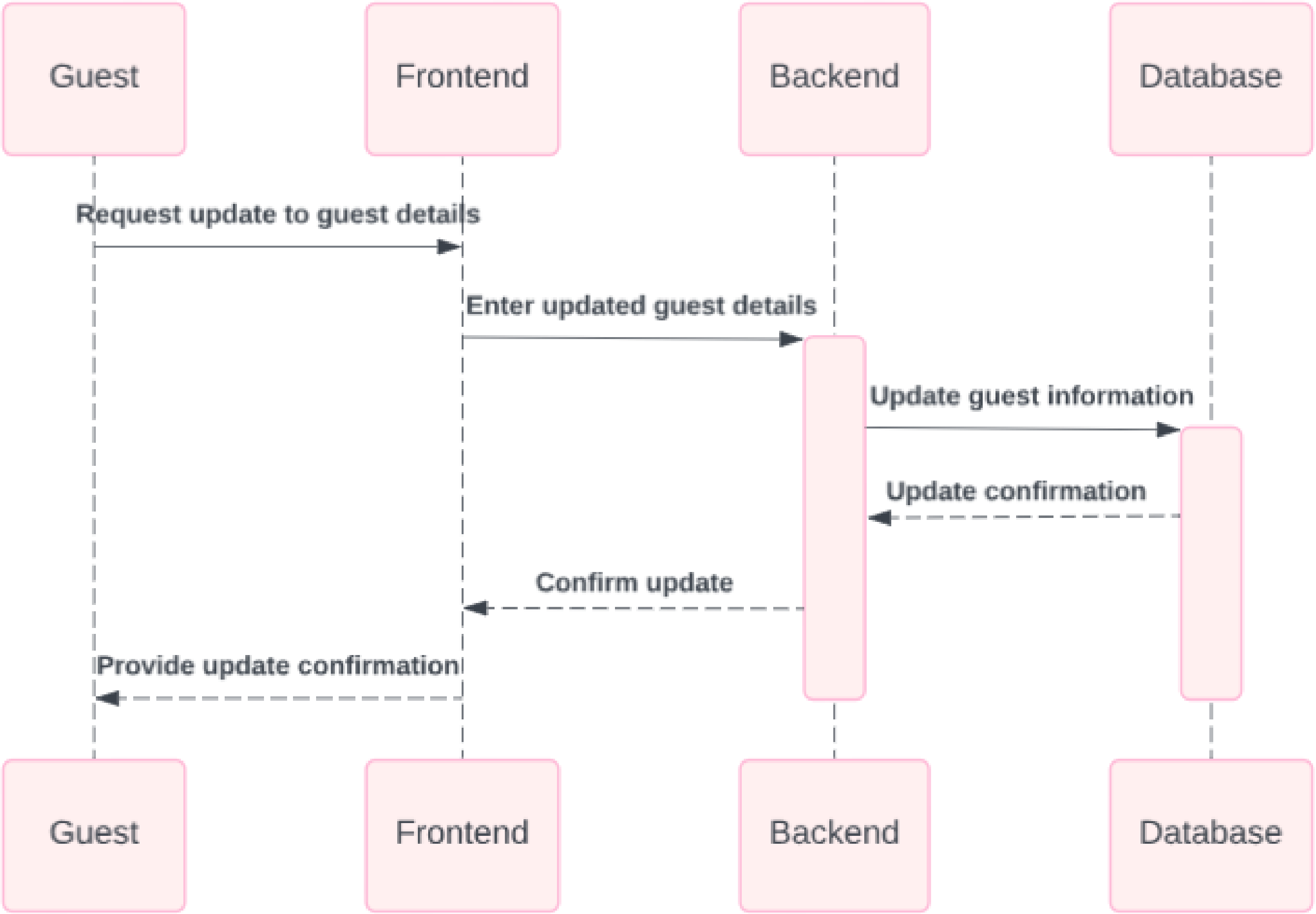
Sequence Diagram 15: Search for Rate



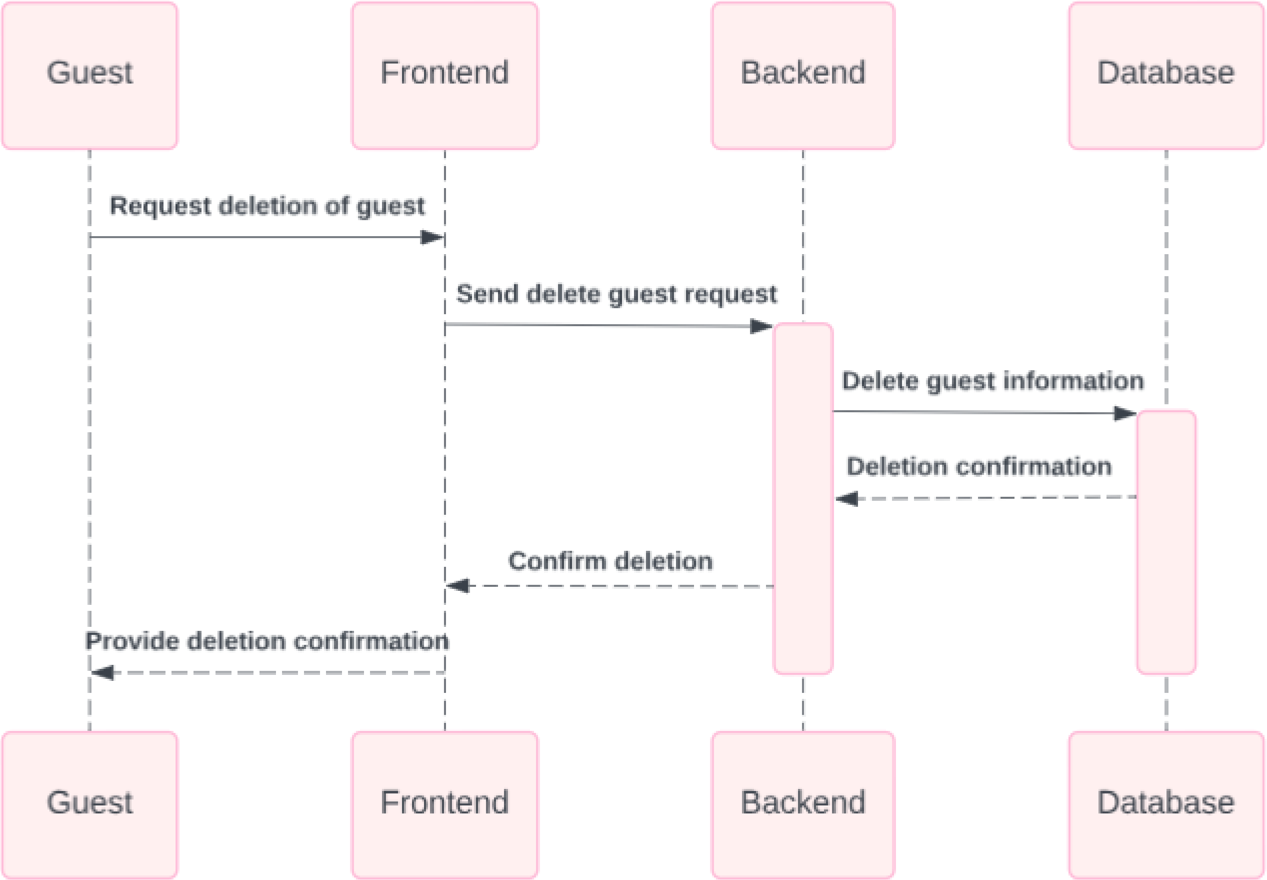
Sequence Diagram 16: Create Guests



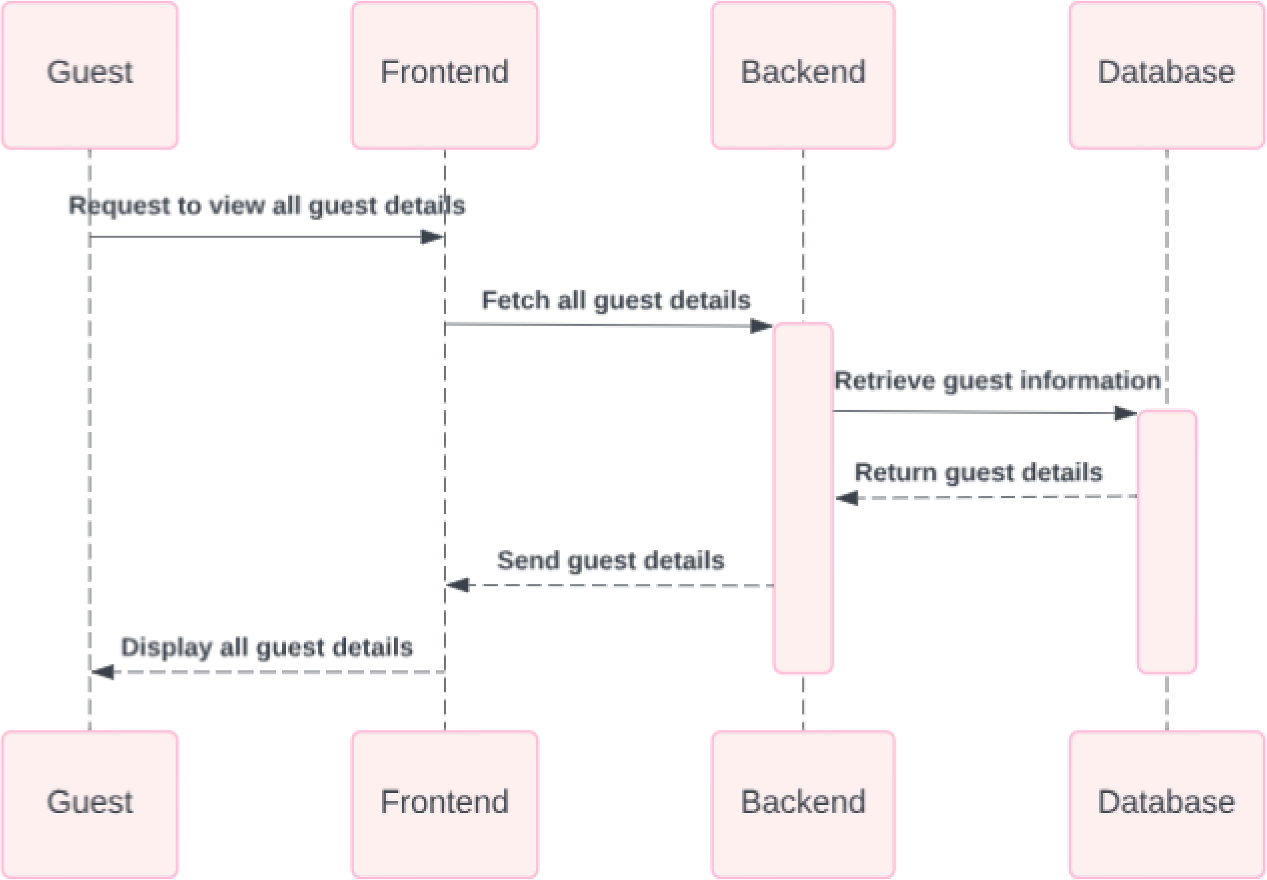
Sequence Diagram 17: Update Guests



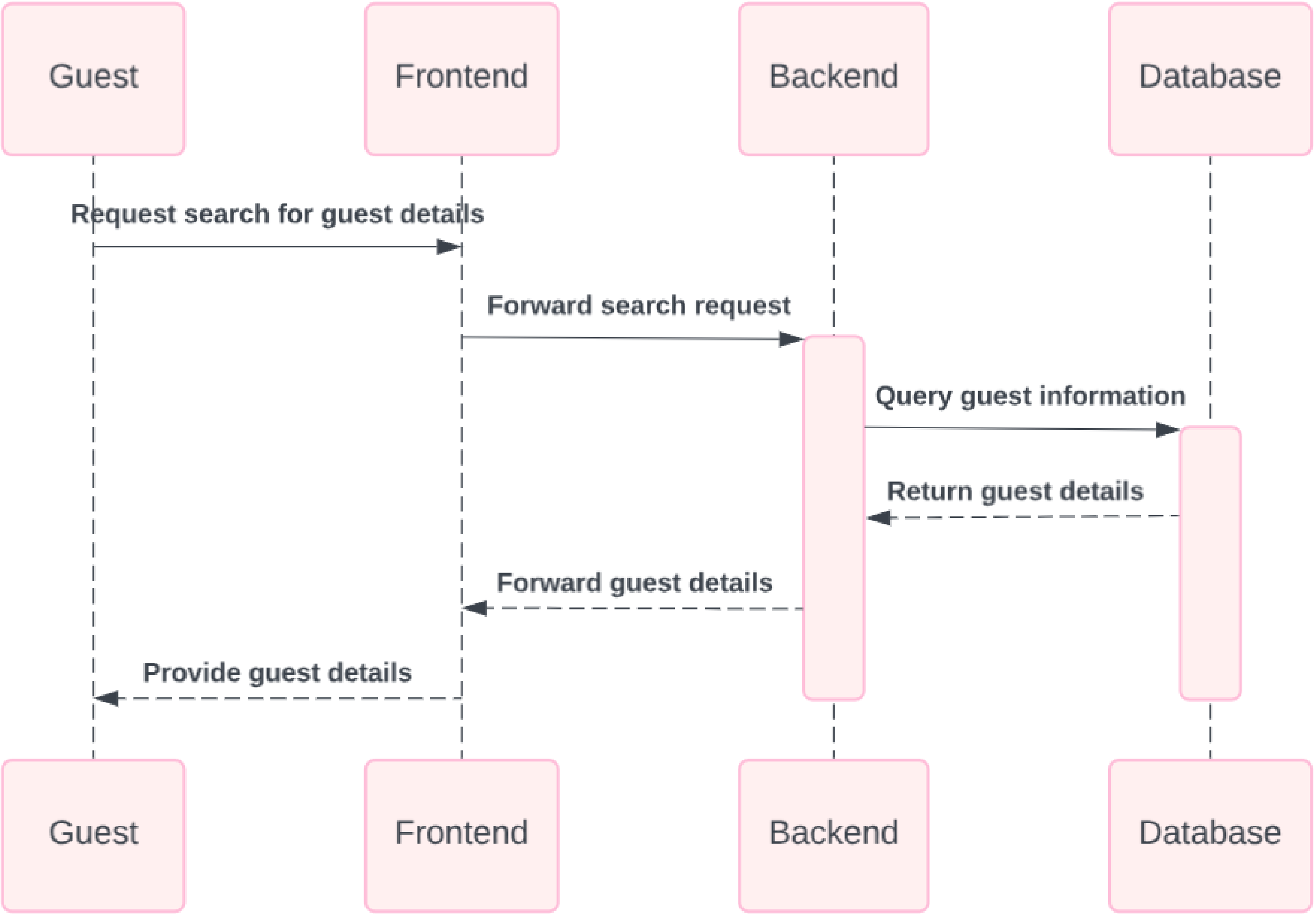
Sequence Diagram 18: Delete Guests



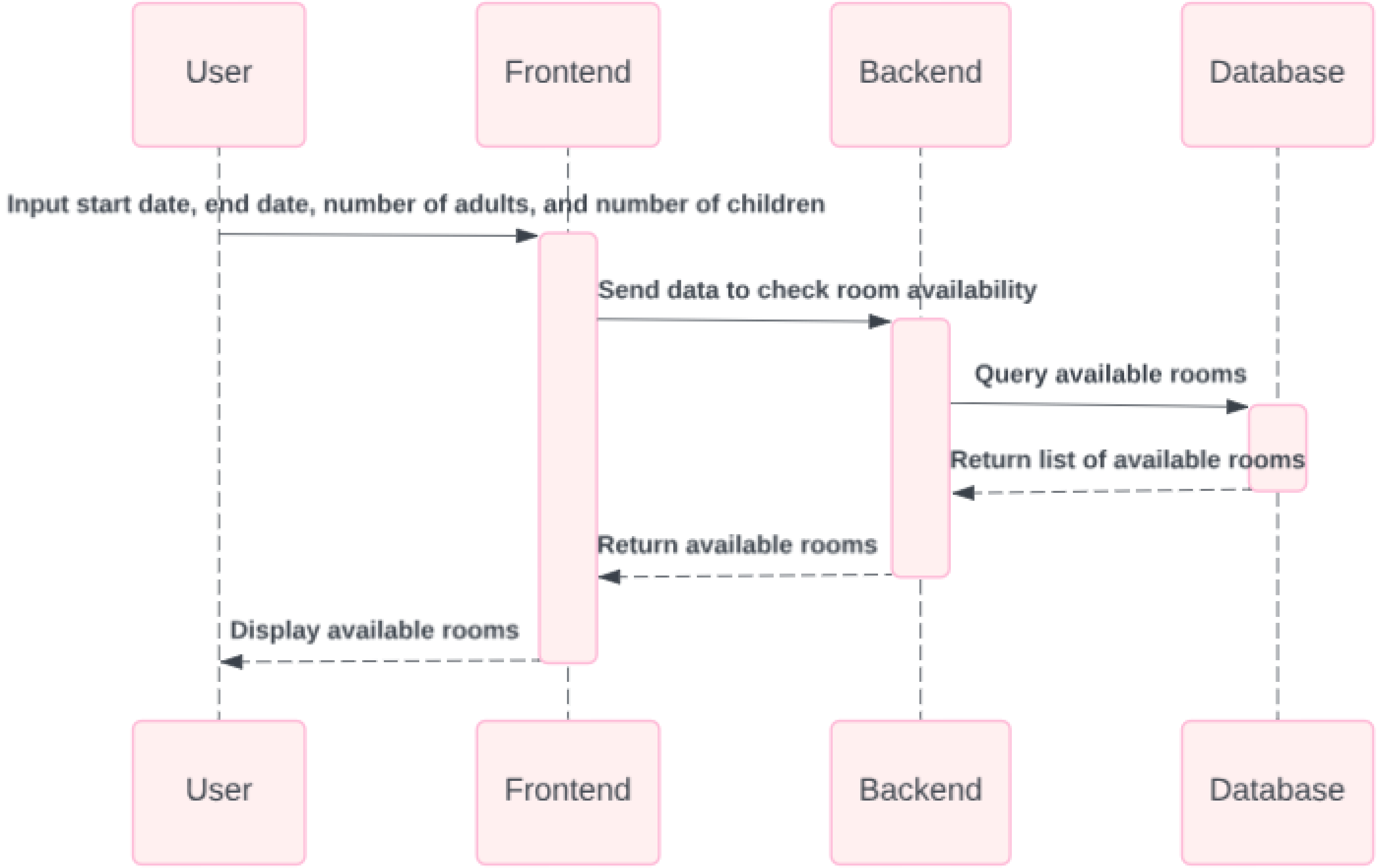
Sequence Diagram 19: View All Guests



Sequence Diagram 20: Search for Guests

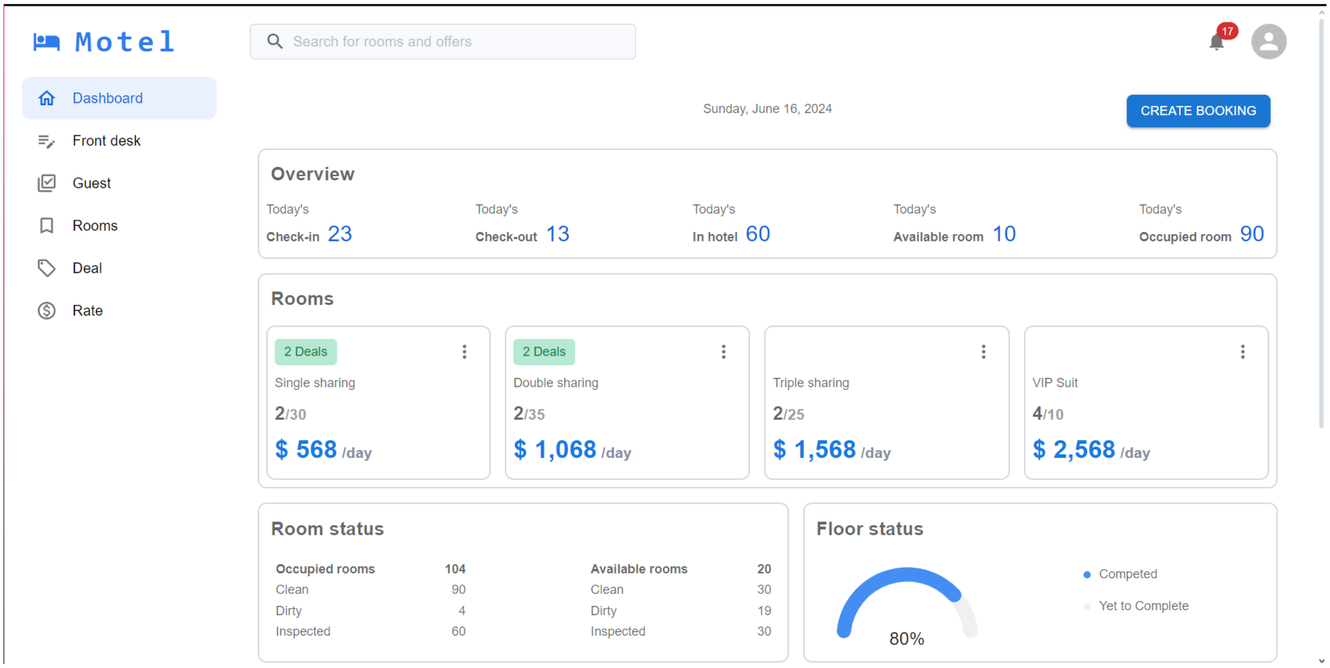


Sequence Diagram 21: Finding available room feature


User
Frontend
Backend
Database
Input start date, end date, number of adults, and number of children
Send data to check room availability
Query available rooms
Return list of available rooms
Return available rooms
Display available rooms
User
Frontend
Backend
Database

IV. IMPLEMENTATION

1. Dashboard UI



Sunday, June 16, 2024

Overview

Today's	Today's	Today's	Today's	Today's
Check-in 23	Check-out 13	In hotel 60	Available room 10	Occupied room 90

Rooms

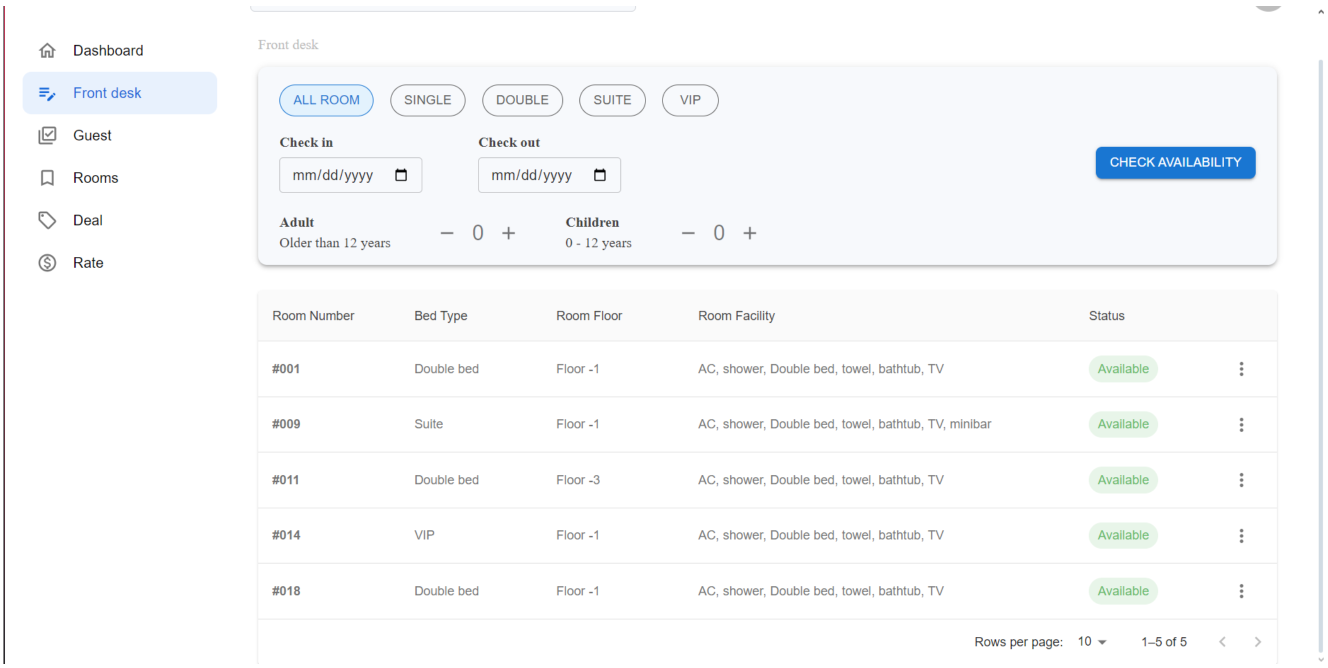
2 Deals	2 Deals		
Single sharing	Double sharing	Triple sharing	VIP Suit
2/30	2/35	2/25	4/10
$ 568 /day	$ 1,068 /day	$ 1,568 /day	$ 2,568 /day

Room status

Occupied rooms	104	Available rooms	20
Clean	90	Clean	30
Dirty	4	Dirty	19
Inspected	60	Inspected	30

Floor status

2. Manage Frontdesk Functions



Check in: mm/dd/yyyy

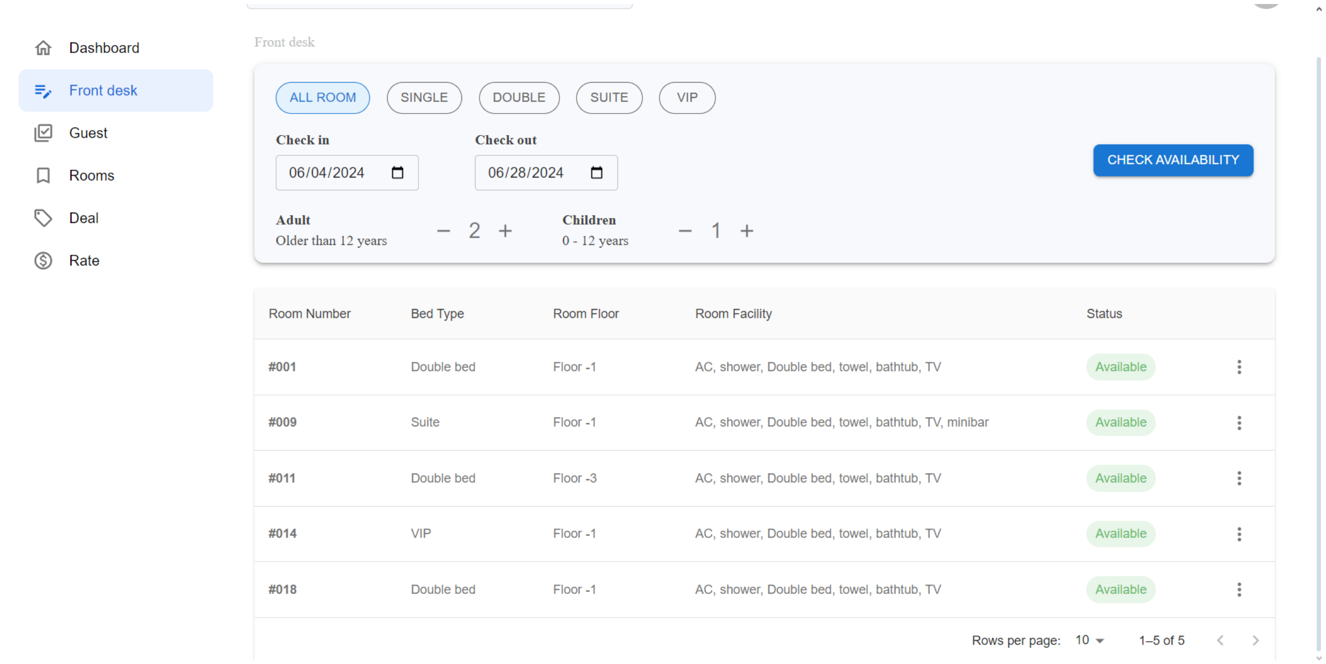
Check out: mm/dd/yyyy

Adult (Older than 12 years): 0

Children (0 - 12 years): 0

Room Number	Bed Type	Room Floor	Room Facility	Status
#001	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#009	Suite	Floor -1	AC, shower, Double bed, towel, bathtub, TV, minibar	Available
#011	Double bed	Floor -3	AC, shower, Double bed, towel, bathtub, TV	Available
#014	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#018	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available

Rows per page: 10 1–5 of 5



Check in: 06/04/2024

Check out: 06/28/2024

Adult (Older than 12 years): 2

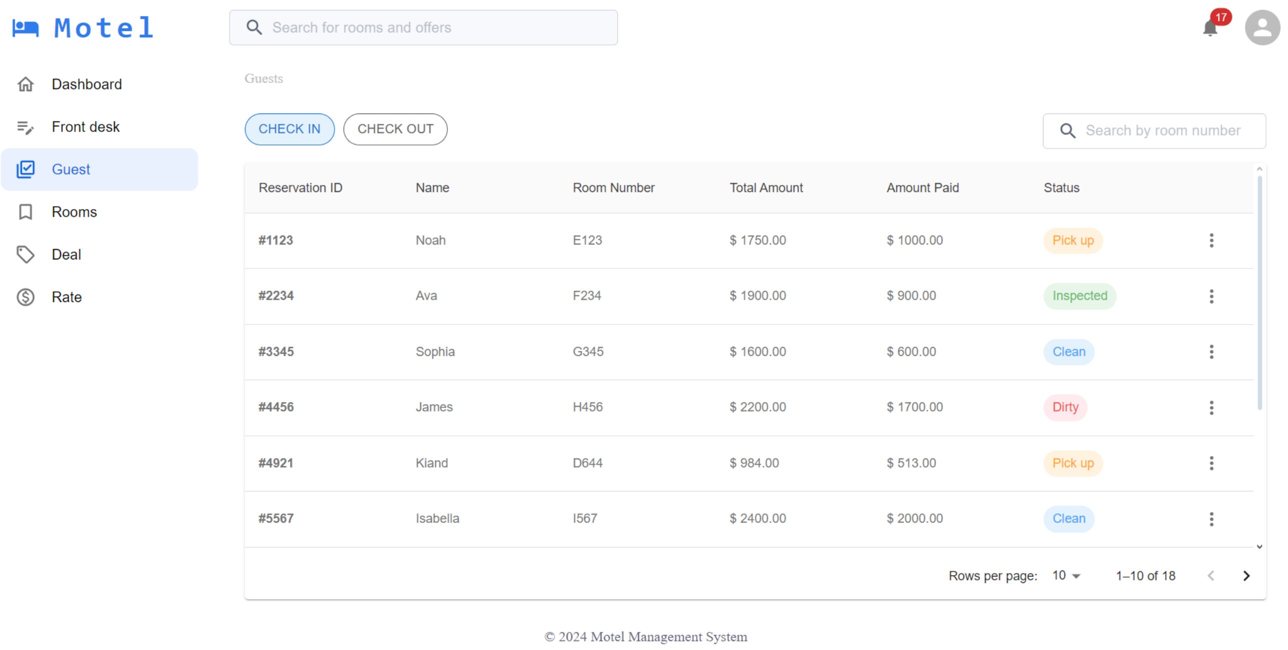
Children (0 - 12 years): 1

Room Number	Bed Type	Room Floor	Room Facility	Status
#001	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#009	Suite	Floor -1	AC, shower, Double bed, towel, bathtub, TV, minibar	Available
#011	Double bed	Floor -3	AC, shower, Double bed, towel, bathtub, TV	Available
#014	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#018	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available

Rows per page: 10 1–5 of 5

On this page, you can search for an available room by choosing check in or check out date to filter the table below. You can also adjust the adults and the children to choose an appropriate room.

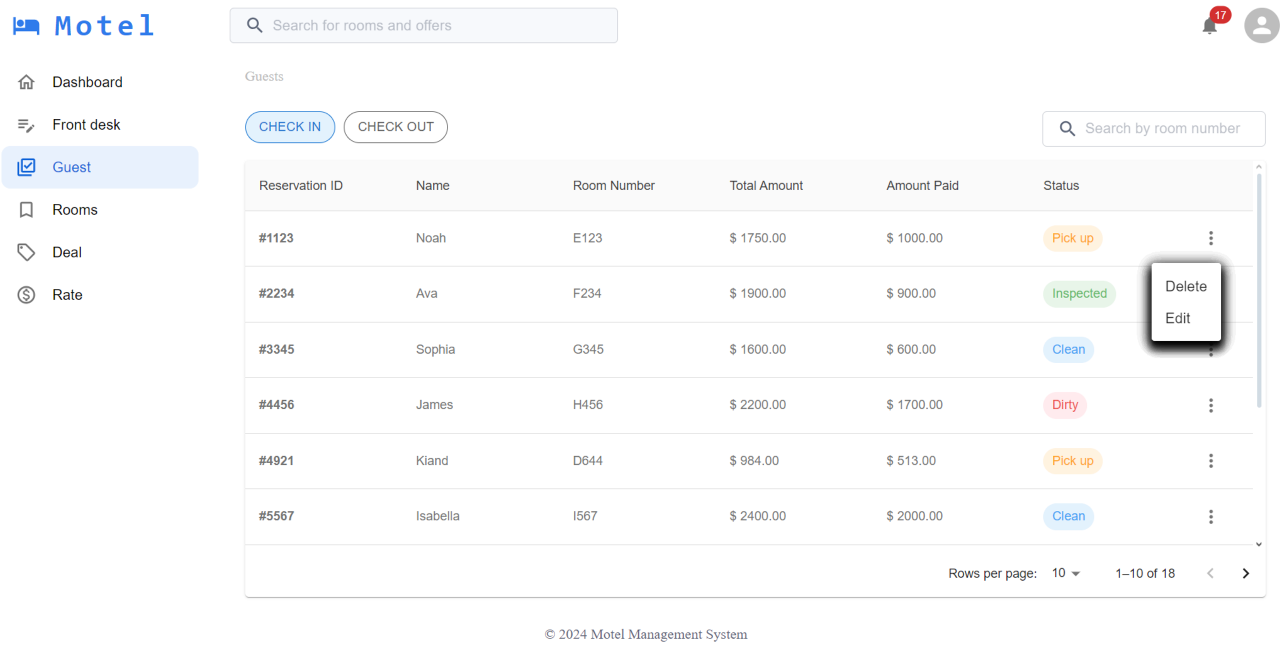
3. Manage Guest Functions



Reservation ID	Name	Room Number	Total Amount	Amount Paid	Status
#1123	Noah	E123	$ 1750.00	$ 1000.00	Pick up
#2234	Ava	F234	$ 1900.00	$ 900.00	Inspected
#3345	Sophia	G345	$ 1600.00	$ 600.00	Clean
#4456	James	H456	$ 2200.00	$ 1700.00	Dirty
#4921	Kland	D644	$ 984.00	$ 513.00	Pick up
#5567	Isabella	I567	$ 2400.00	$ 2000.00	Clean

Rows per page: 10 1–10 of 18

© 2024 Motel Management System



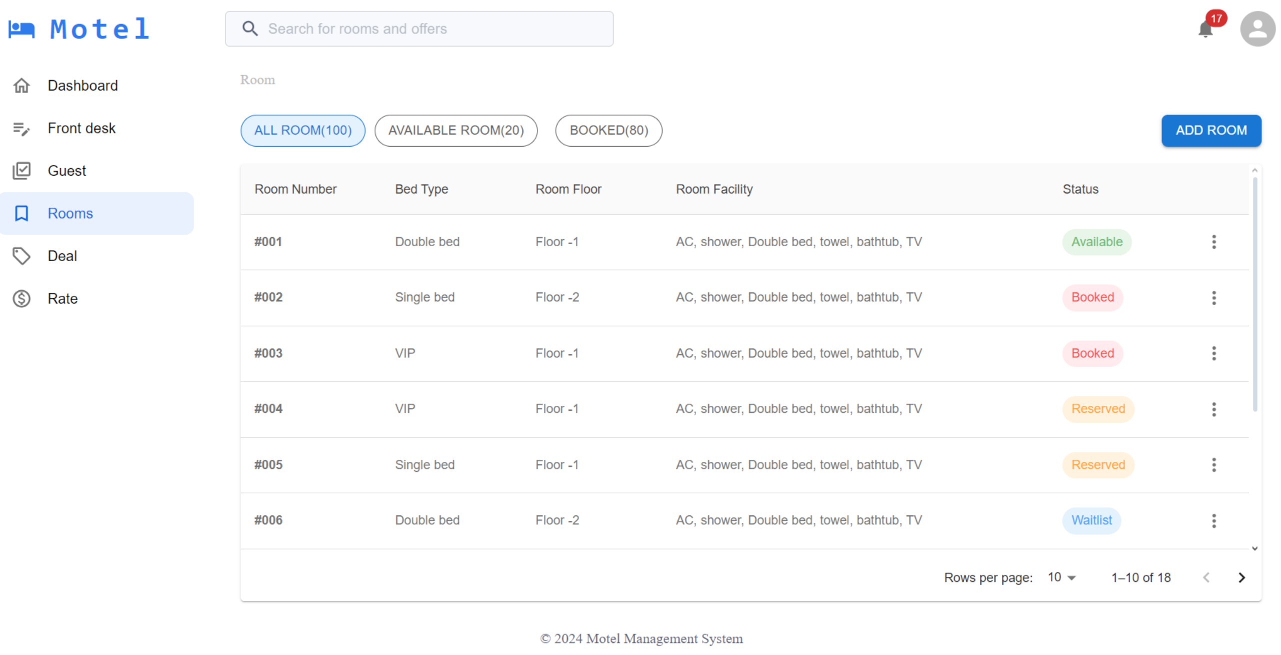
Reservation ID	Name	Room Number	Total Amount	Amount Paid	Status
#1123	Noah	E123	$ 1750.00	$ 1000.00	Pick up
#2234	Ava	F234	$ 1900.00	$ 900.00	Inspected
#3345	Sophia	G345	$ 1600.00	$ 600.00	Clean
#4456	James	H456	$ 2200.00	$ 1700.00	Dirty
#4921	Kland	D644	$ 984.00	$ 513.00	Pick up
#5567	Isabella	I567	$ 2400.00	$ 2000.00	Clean

Rows per page: 10 1–10 of 18

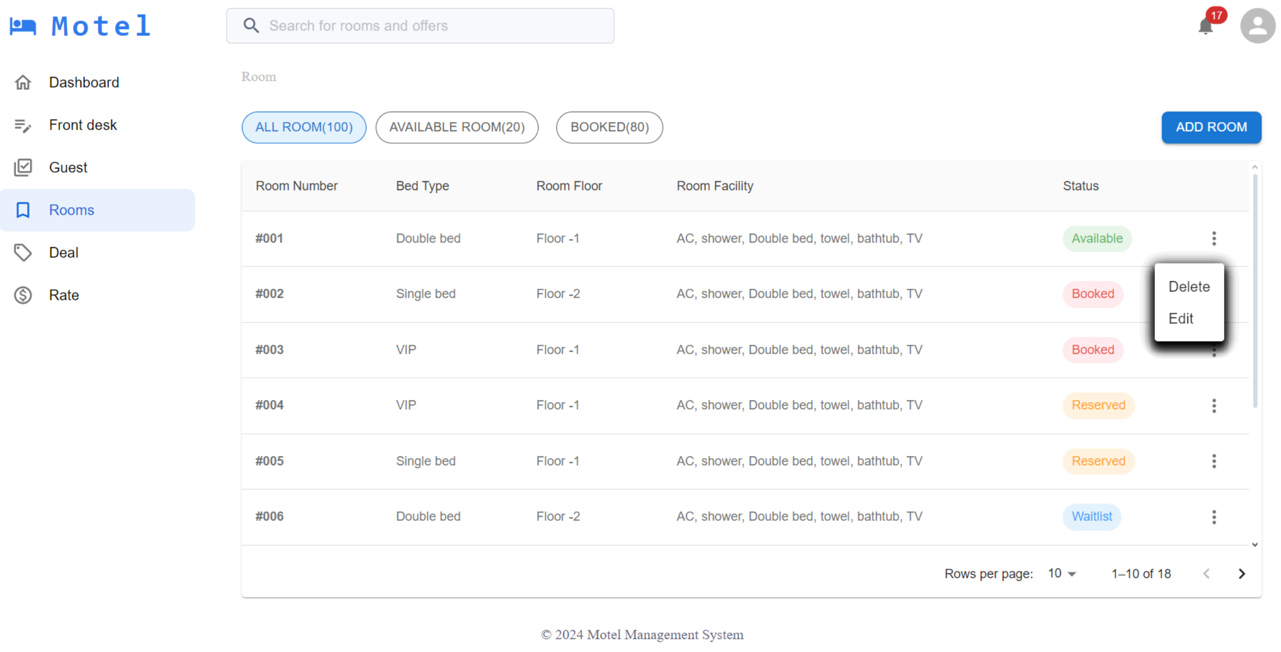
© 2024 Motel Management System

On this page, you can search for a guest by filling in the search bar on the upright corner. You can also view and update, delete a guest by pressing the three dots at the action's column

4. Manage Guest Functions



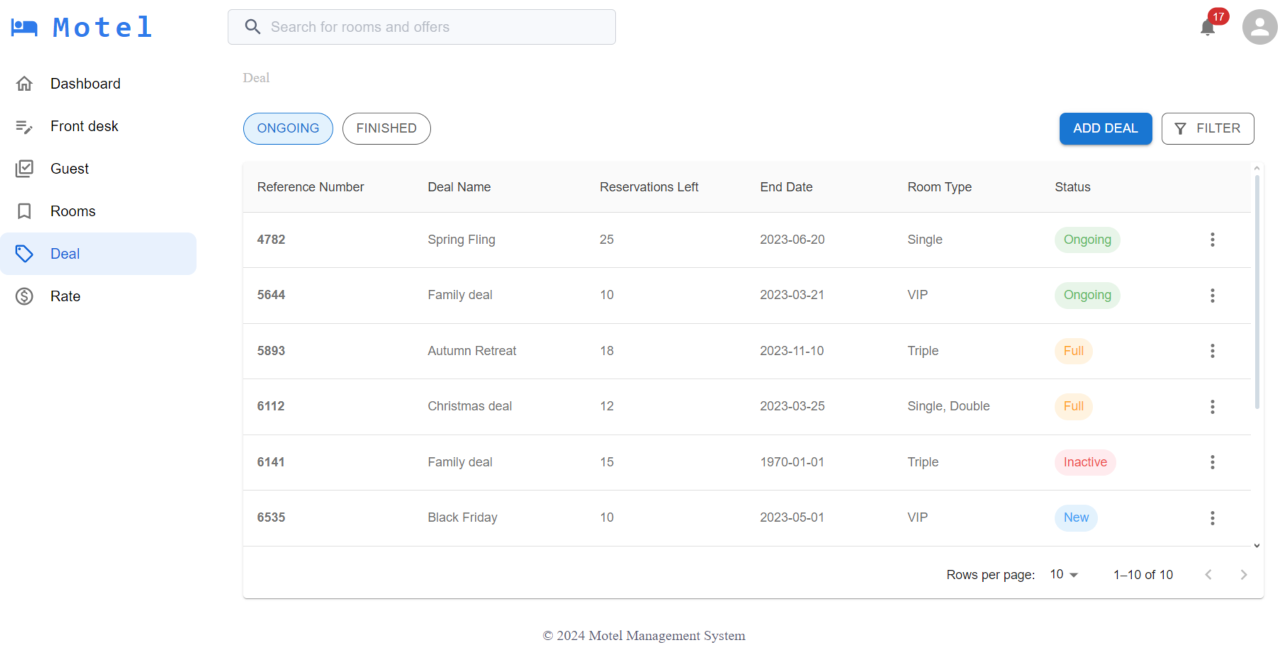
Room Number	Bed Type	Room Floor	Room Facility	Status
#001	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#002	Single bed	Floor -2	AC, shower, Double bed, towel, bathtub, TV	Booked
#003	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Booked
#004	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Reserved
#005	Single bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Reserved
#006	Double bed	Floor -2	AC, shower, Double bed, towel, bathtub, TV	Waitlist



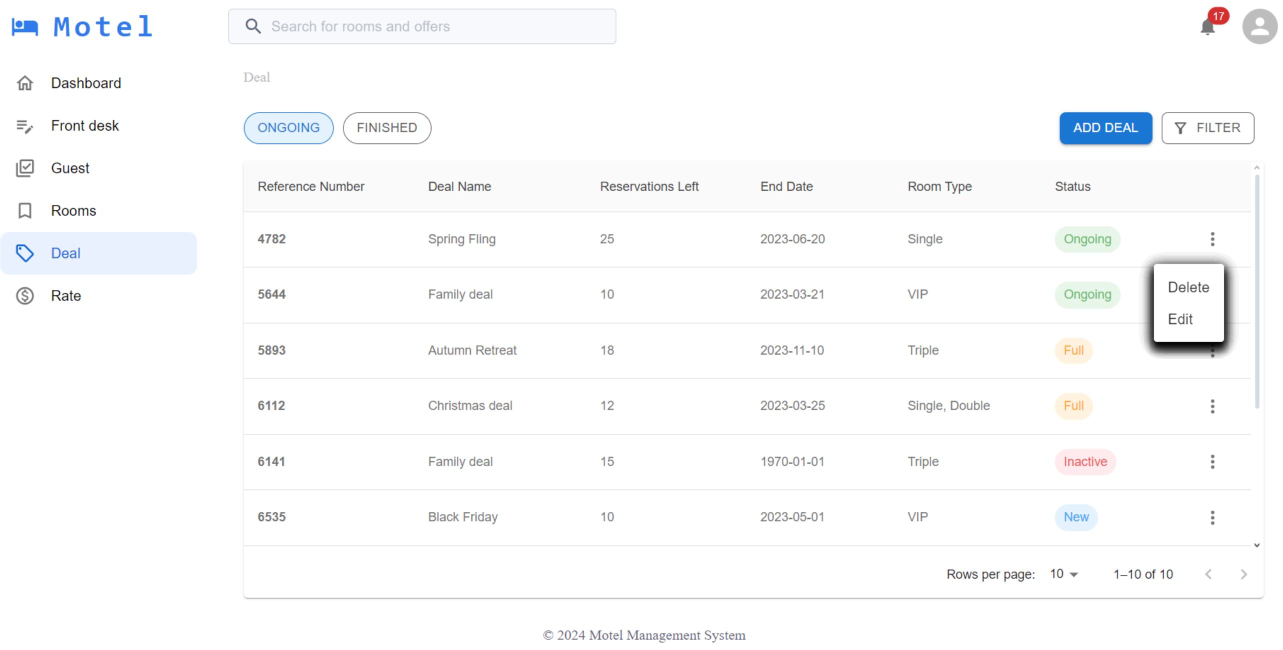
Room Number	Bed Type	Room Floor	Room Facility	Status
#001	Double bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Available
#002	Single bed	Floor -2	AC, shower, Double bed, towel, bathtub, TV	Booked
#003	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Booked
#004	VIP	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Reserved
#005	Single bed	Floor -1	AC, shower, Double bed, towel, bathtub, TV	Reserved
#006	Double bed	Floor -2	AC, shower, Double bed, towel, bathtub, TV	Waitlist

On this page, you can search for a room by filling in the search bar on the upright corner. You can also view and update, delete a room by pressing the three dots at the action's column.

5. Manage Deal Functions



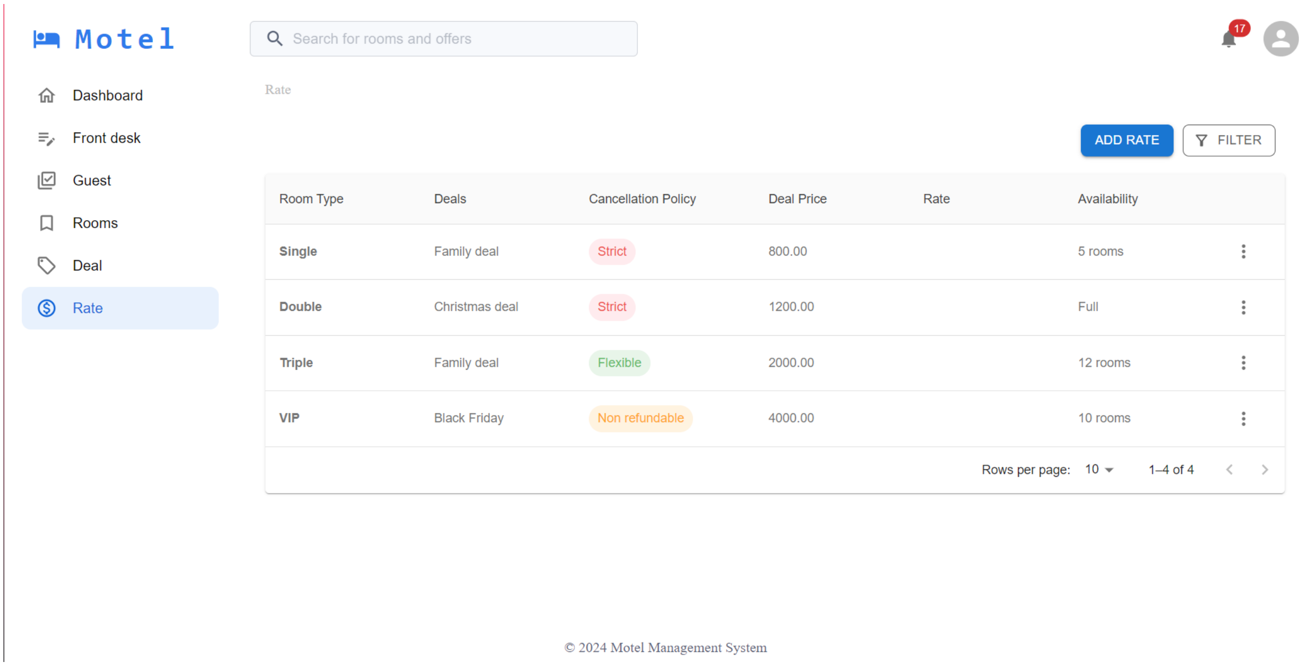
Reference Number	Deal Name	Reservations Left	End Date	Room Type	Status
4782	Spring Fling	25	2023-06-20	Single	Ongoing
5644	Family deal	10	2023-03-21	VIP	Ongoing
5893	Autumn Retreat	18	2023-11-10	Triple	Full
6112	Christmas deal	12	2023-03-25	Single, Double	Full
6141	Family deal	15	1970-01-01	Triple	Inactive
6535	Black Friday	10	2023-05-01	VIP	New



Reference Number	Deal Name	Reservations Left	End Date	Room Type	Status
4782	Spring Fling	25	2023-06-20	Single	Ongoing
5644	Family deal	10	2023-03-21	VIP	Ongoing
5893	Autumn Retreat	18	2023-11-10	Triple	Full
6112	Christmas deal	12	2023-03-25	Single, Double	Full
6141	Family deal	15	1970-01-01	Triple	Inactive
6535	Black Friday	10	2023-05-01	VIP	New

On this page, you can search for a deal by filling in the search bar on the upright corner. You can also view and update, delete a deal by pressing the three dots at the action's column.

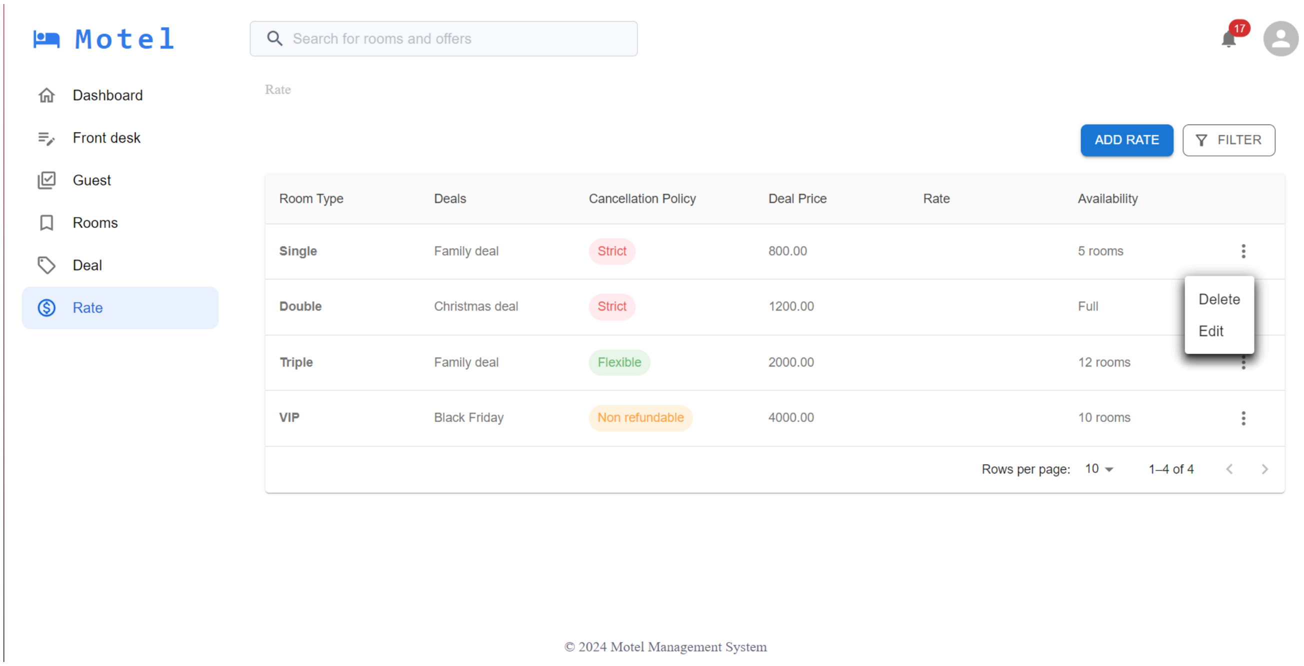
6. Manage Rate Functions



Room Type	Deals	Cancellation Policy	Deal Price	Rate	Availability
Single	Family deal	Strict	800.00		5 rooms
Double	Christmas deal	Strict	1200.00		Full
Triple	Family deal	Flexible	2000.00		12 rooms
VIP	Black Friday	Non refundable	4000.00		10 rooms

Rows per page: 10 1–4 of 4

© 2024 Motel Management System



Room Type	Deals	Cancellation Policy	Deal Price	Rate	Availability
Single	Family deal	Strict	800.00		5 rooms
Double	Christmas deal	Strict	1200.00		Full
Triple	Family deal	Flexible	2000.00		12 rooms
VIP	Black Friday	Non refundable	4000.00		10 rooms

Rows per page: 10 1–4 of 4

© 2024 Motel Management System

On this page, you can search for a rate by filling in the search bar on the upright corner. You can also view and update, delete a rate by pressing the three dots at the action’s column

V. REQUIREMENT ANALYSIS AND DESIGN

The development of our motel management project has been guided by rigorous analysis, design, and implementation phases aimed at addressing the diverse needs of motel operations. Throughout this project, we have prioritized efficiency, usability, and security to enhance the overall guest experience and operational effectiveness.

In the discussion, we have highlighted key aspects such as operational, legal, usability, maintainability, support, security, and interface requirements. These requirements have shaped our approach to system development, ensuring robust performance under heavy workloads, ease of maintenance, and comprehensive support for non-technical users.

Our implementation of stringent security measures, including data encryption, access restrictions, and regular backups, underscores our commitment to safeguarding guest information and maintaining system integrity.

In conclusion, our motel management project aims to set new standards in the hospitality industry by offering a user-friendly interface, reliable performance, and responsive customer support. Moving forward, continuous feedback and iteration will be crucial to further refining our system and meeting the evolving needs of motel operators and their guests.

VI. REFERENCES

- **GitHub:**
  - https://github.com/dungtruong151/motel-management/
- **Deployed web online:**
  - https://dungtruong151.github.io/motel-management/
- **Agile**
  - http://agilemethodology.org/
- **Database Management System (DBMS)**
  - https://dev.mysql.com/
- **HTML, CSS, JavaScript techniques**
  - www.w3schools.com
- **React and Nodejs data**
  - https://react.dev
  - https://nodejs.org/en
- **Material UI**
  - https://mui.com/material-ui/