BENEDICT XVI
CATHOLIC INSTITUTE OF HIGHER EDUCATION
NEGOMBO,
SRI LANKA

Group Software Project – BSIT 22063

Hotel Management System

Project Report

Project ID: Group 3

Submitted by:

(S.E.M Rodrigo)
(W.H.N Fernando)
(M.A.P Buddhima)

<<29th June 2023>>

Abstract

This report provides an overview of the successful implementation of a Hotel Management System (HMS) at Rico Shadow Guest House, a manually operated hotel. The customized HMS was designed to address the specific needs of Rico Shadow Guest House, incorporating two administrative roles for room management and kitchen operations. The HMS implementation commenced with a thorough analysis of the existing manual processes and identified areas for improvement. By leveraging technology, the system aimed to automate and streamline operations while enhancing overall efficiency.

The system will encompass various key functionalities, including room booking and food ordering. Additionally, we will integrate multiple admin controls, allowing the main admin to handle room bookings and enabling the kitchen system to sufficiently manage food orders for customers.

The introduction of the HMS at Rico Shadow Guest House has revolutionized their operations, enhancing efficiency, guest satisfaction, and profitability. The tailored administrative roles for room and kitchen management have empowered the staff, enabling smoother operations and improved coordination between departments.

Acknowledgement

We would like to express our deepest gratitude to all those who have contributed to the successful completion of this project on the hotel management system. Their support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, we would like to extend my sincere appreciation to Mr. Sohan Fernando (Supervisor), for their invaluable guidance and expertise. We are also thankful to the faculty members of BCI campus for their valuable advice. We extend our appreciation to the hotel management team at RICO SHADOW GUEST HOUSE for their cooperation and insights. Special thanks to the participant who volunteered their time and provided feedback.

Thank you all for your unwavering support.

Sincerely,

[Group 3 members]

Declaration

We declare that this project report or part of it was not a copy of a document done by any organization, university any other institute, or a previous student project group at BCI and was not copied from the Internet or other sources.

Project Details

Project Title	Hotel Management System
Project ID	

Group Members

Reg. No	Name	Signature
[redacted]	S.E.M Rodrigo	[redacted]
	W.H.N Fernando	
	M.A.P Buddhima	

Table of Contents

List of Figures

List of Acronyms and Abbreviations

HMS – Hotel Management System
API - Application Programming Interface
UI - User Interface

1. Introduction

1.1 Problem Statement

The problem at hand is the manual management of various tasks within our hotel, such as room bookings and food orders. The current manual system is prone to errors, time-consuming, and inefficient. Therefore, there is a need to develop a comprehensive hotel management system that can automate and streamline these tasks, leading to improved efficiency, accuracy, and overall customer satisfaction.

To address this problem, we have created a hotel management system that aims to centralize and automate the management of room bookings and food orders. The system will provide an intuitive user interface for hotel staff, allowing them to perform their tasks more efficiently.

1.2 Product Scope

Our project aims to develop a web-based hotel management system that addresses the challenges of our current manual system. The system will encompass various key functionalities, including room booking and food ordering. Additionally, we will integrate multiple admin controls, allowing the main admin to handle room bookings and enabling the kitchen system to sufficiently manage food orders for customers. By automating these processes, we seek to improve customer satisfaction, reduce errors, and enhance the overall management of our hotel operations.

- Room Management: The system will allow the hotel admin to manage room availability, assign rooms to guests, and track room statuses.

- Reservation and Booking: Admin will be able to make room reservations of customers through the system. The system will handle the reservation process, including checking room availability, and confirming reservations.

- Check-In and Check-Out: The system will facilitate the check-in and check-out process for guests.

- Food Ordering: Kitchen admin will have the option to place food orders of customers (for currently booked rooms) through the hotel management system. The system will manage the order processing and delivery tracking.

1.3 Project Report Structure

The project report is organized as follows:

❖ **Chapter 2 Methodology**

- System Requirements Analysis

  - This chapter provides a detailed analysis of the requirements for the hotel management system. It includes functional and non-functional requirements, user requirements, and system constraints.

- System Design

  - This chapter describes the architectural design of the hotel management system. It covers the system's overall structure, database design, UI design, and integration with external systems.

- Implementation and Testing

  - Here, we discuss the implementation details of the system and the testing methodologies employed. It includes information about the technologies used, coding practices, and the testing approach followed to ensure the system's quality.

- **Chapter 3: Evaluation**

- This chapter concludes the project report by evaluating the achieved results, reflecting on lessons learned, and providing guidance for future work. It helps to summarize the project's outcomes and provide insights for continuous improvement and future endeavors.

- Assessment of the Project Results

  - Overview of the achieved goals and objectives
  - Evaluation of the project's outcomes and deliverables
  - Analysis of the project's performance against key metrics
  - Identification of strengths and weaknesses of the project results

- Lessons Learned

  - Project management lessons
  - Technical lessons
  - Team collaboration and communication lessons
  - Stakeholder engagement and feedback

- Future Work

  - Identified areas for improvement.
  - Recommendations for future projects
  - Potential next steps
  - Project sustainability and long-term implications

❖ **Chapter 4: Conclusion**

  - The final chapter concludes the project report, summarizing the key findings and highlighting the success of the hotel management system. It also includes recommendations for further improvements and potential areas of future research.

❖ **Chapter 5: Reference**

- It is important to consult the specific style guide or referencing guidelines.

❖ **Appendix: Additional Supporting Materials**

- The appendix contains supplementary materials such as system diagrams, sample UI, and any other relevant documentation to provide a comprehensive understanding of the hotel management system.

2. Methodology

2.1 Requirements and Analysis

Use Case Diagram

(Zahid, n.d.)

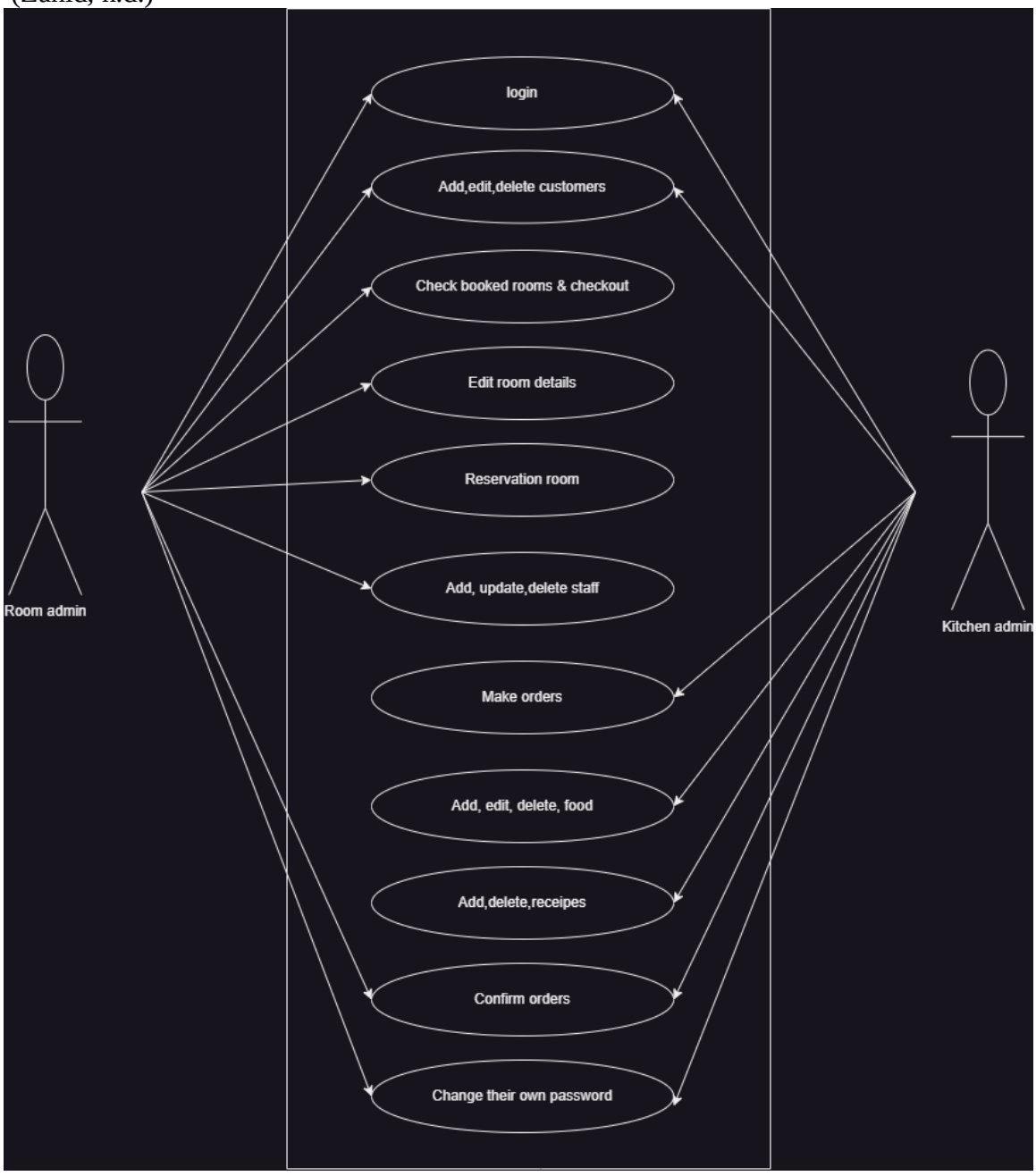


Figure 1 Use Case Diagram

Activity Diagram
(Nym, 2022)

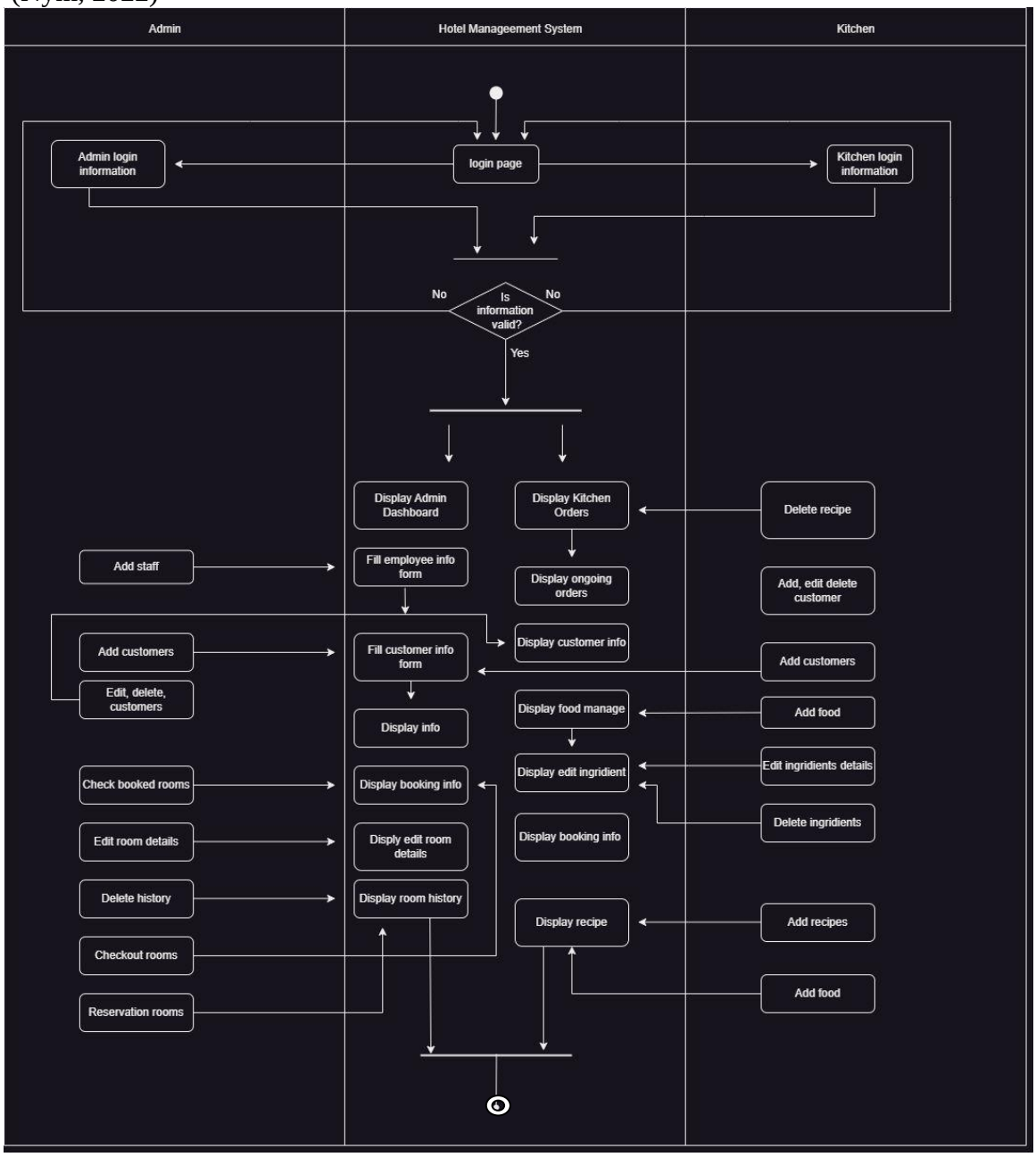


Figure 2Activity Diagram

2.2 Design

2.2.1 ER Diagram

Table: admin
admin_id (int)
username (varchar)
password (varchar)

Table: bookedroom
bookedid (int)
roomid (int)
total (double)

Table: customer
customerid (varchar)
cfname (varchar)
clname (varchar)
cmobile (varchar)
cemail (varchar)
cdob (date)
cregisterday (datetime)
health (text)
cdescription (text)
cid (int)

Table: food
food_id (int)
food_name (varchar)
food_quntity (int)
food_weight (double)
food_measurement (varchar)
food_unit_price (double)
food_img (varchar)
food_description (text)
food_resgister_date (datetime)
food_total (double)
.

Table: history
historyid (int)
historycustomerid (varchar)
historyroomid (int)
historydescription (text)
historyrigisterdate (datetime)
historygivendate (datetime)
historypayement (double)

Table: kitchen
kitchen_id (int)
username (varchar)
password (varchar)

Table: placedorder
order_id (int)
order_roomnumid (int)
order_recipe_id (int)
order_customerName (varchar)
order_time (datetime)
order_status (tinyint)
order_quantity (int)
order_total (double)
order_placedBy (varchar)

Table: recipe
recipe_id (int)
recipe_name (varchar)
recipe_ingredients (text)
recipe_price (double)
realCost (double)

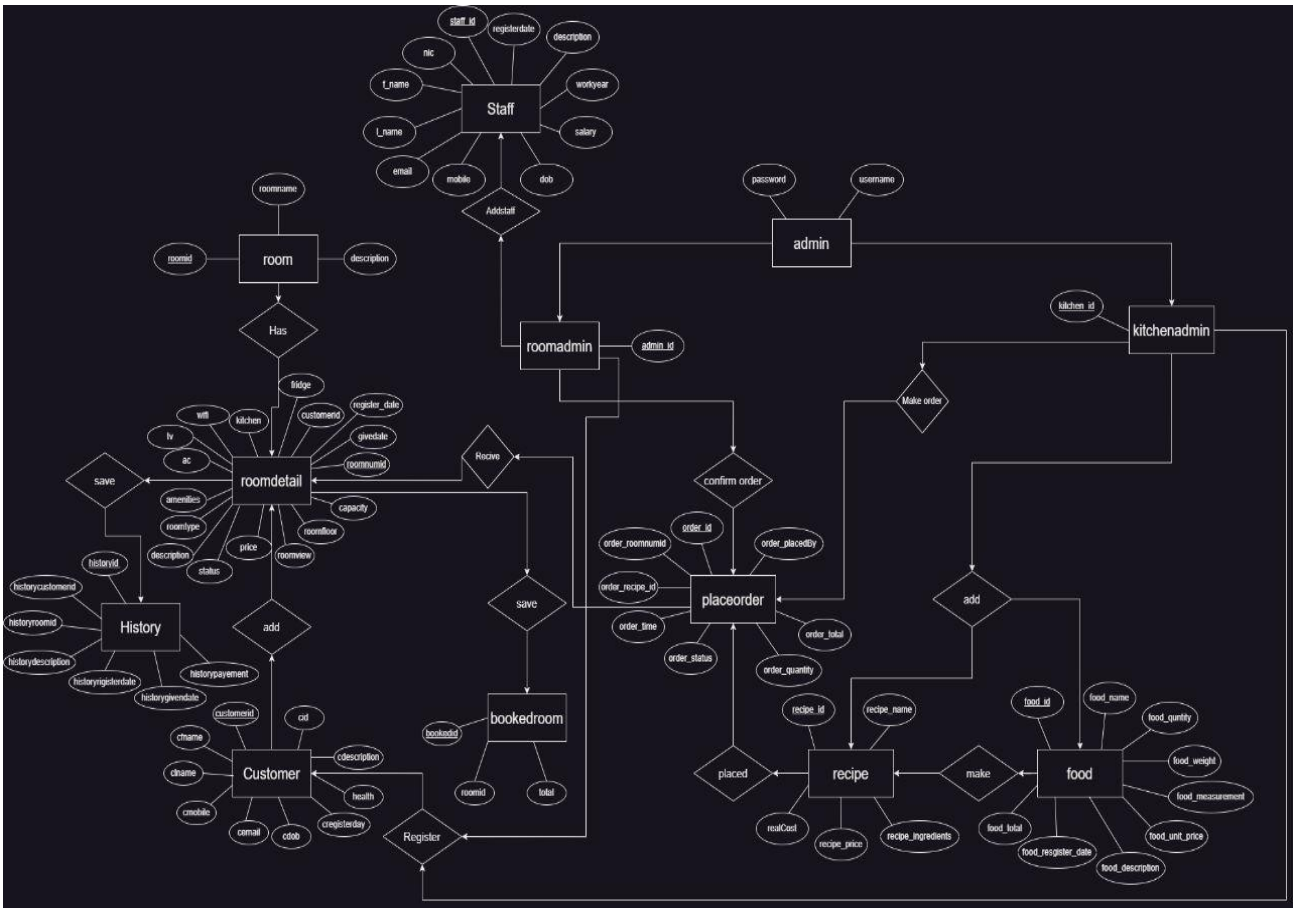


Figure 3ER Diagram

2.2.2 High-Level Architecture Diagram

IPO model

- The IPO model provides a clear understanding of the system's input, the processing steps involved, and the resulting output. It helps in visualizing the flow of data and the transformation that occurs within the system.

Input:

Guest information (name, contact details.)

Staff information (name, position, etc.)

Room inventory and availability

Reservation and booking details.

Process:

Room management (Edit room details, availability, etc.)

Reservation and booking management (Check in and check out bookings)

Food orders management (Place order, confirm orders, add recipes and ingredients)

Output:

Booking confirmations

Make orders.

Class Diagram

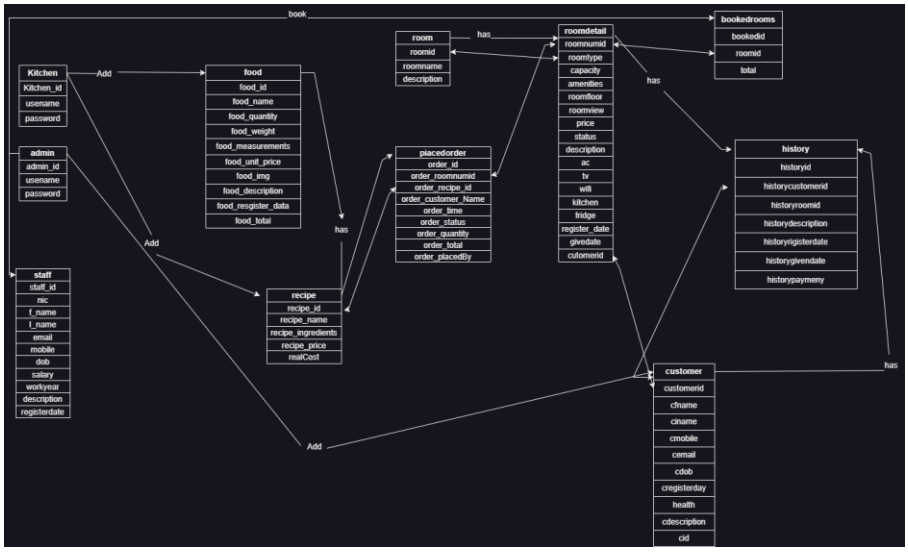


Figure 4Class diagram

2.2.3 User Interface Design

Figure 5Login page

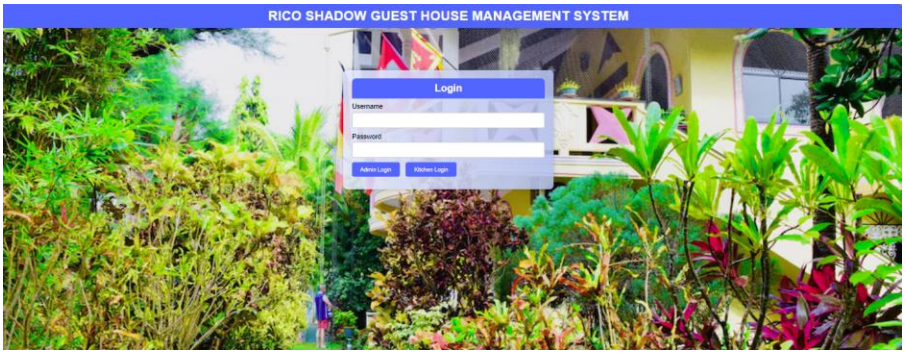


Figure 6Admin dashboard (Room admin)

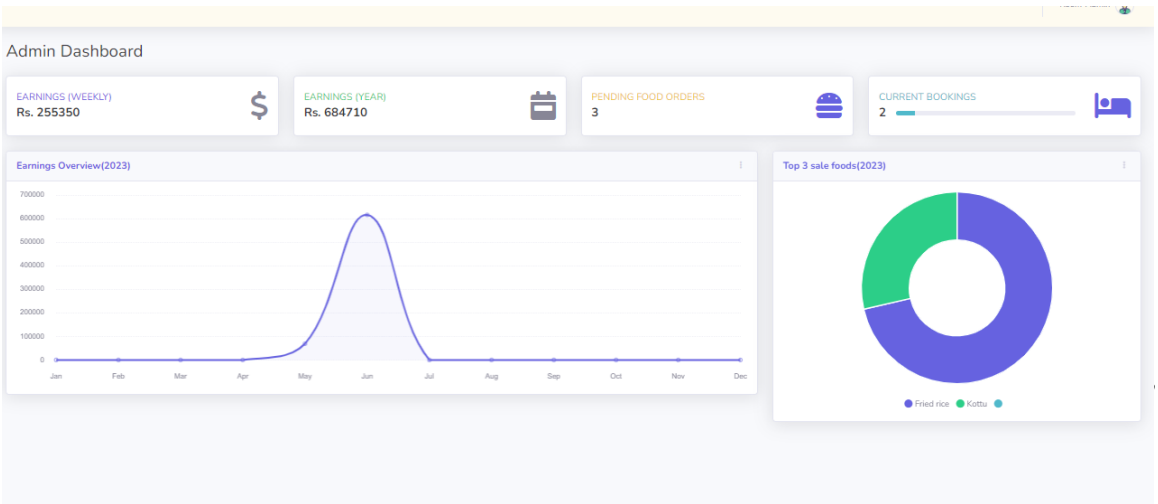


Figure 7Room booking page. (Room admin)

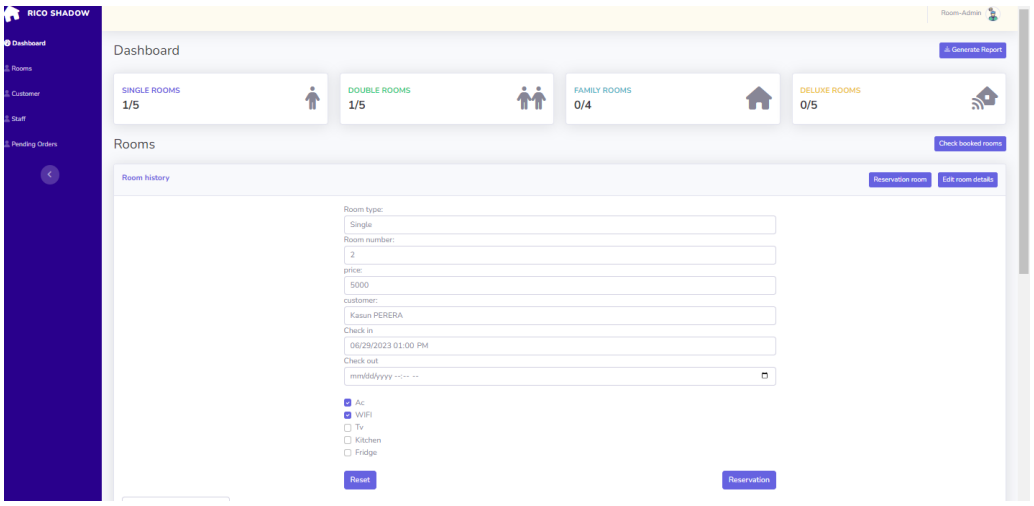


Figure 8add customer. (Room admin)

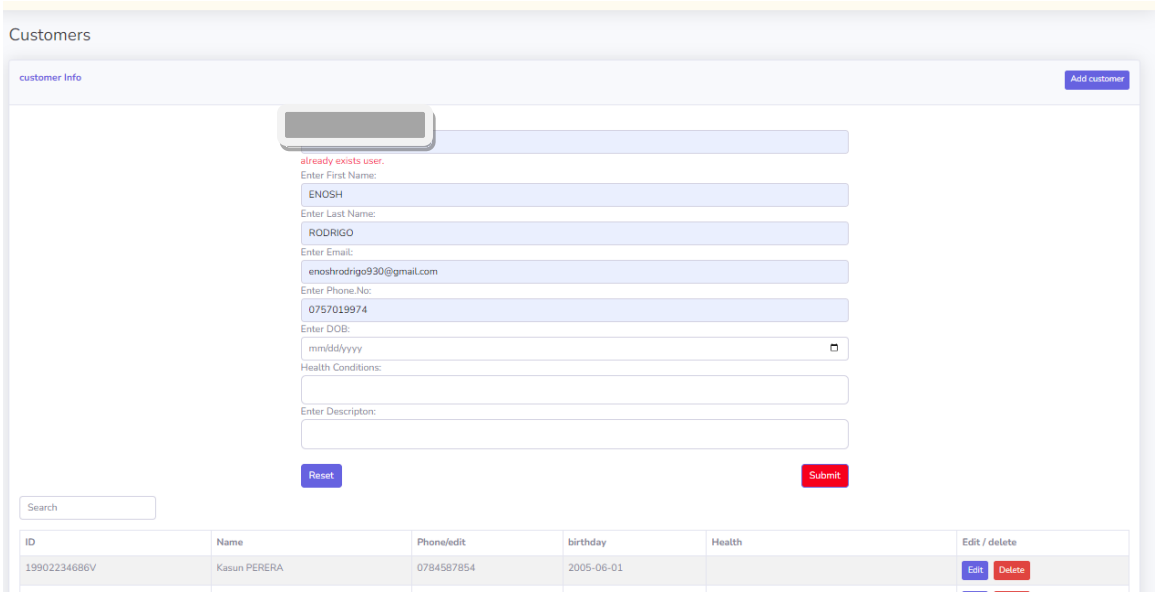


Figure 9Make Orders UI (Kitchen admin)

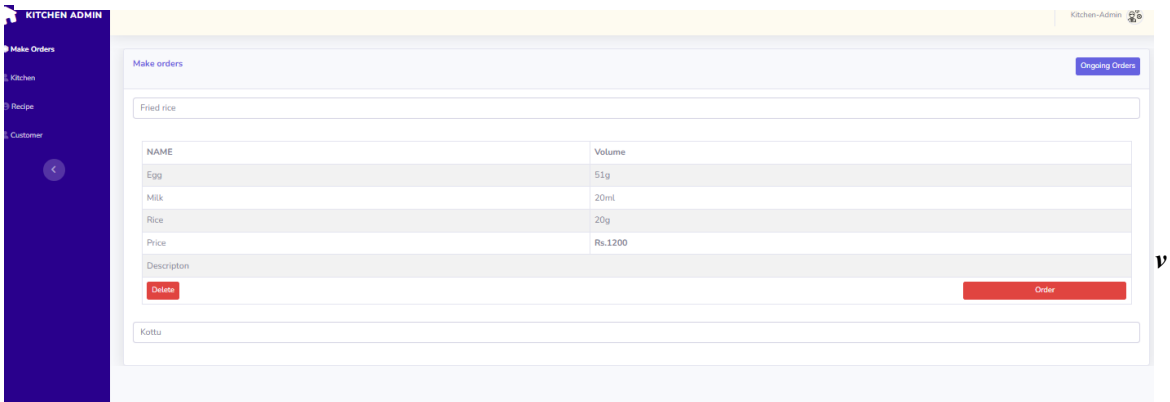


Figure 10Add ingredients Page (Kitchen admin)

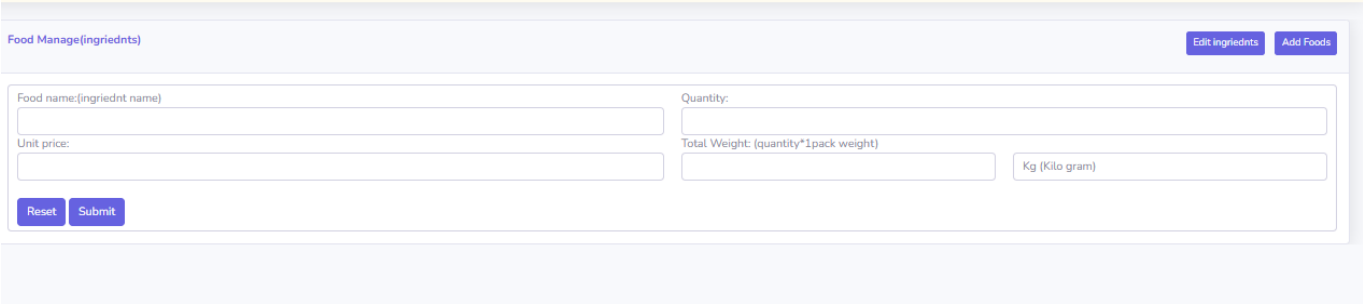
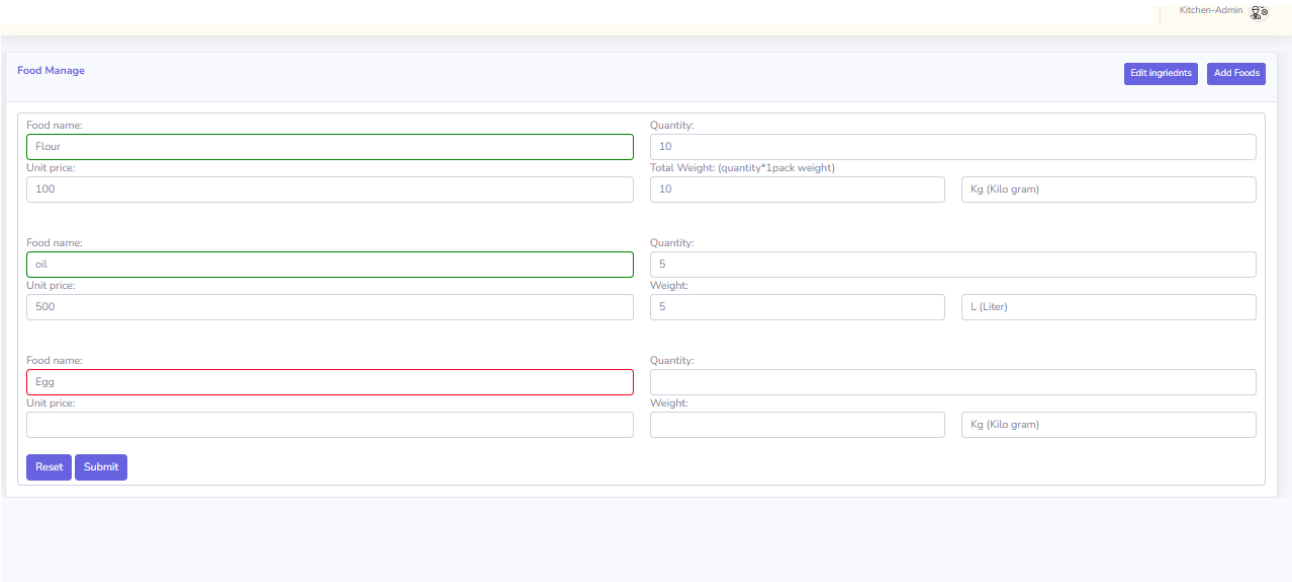


Figure 11Add recipes UI. (Kitchen admin)



2.3 Implementation

1) Reservation Module:

   a) Design and create a database to store reservation details, guest information, and room availability.
   b) Develop a user interface for admi n to search for available rooms, select check out date, and make reservations.
   c) Implement features to handle room availability and conflicts.
   d) Create database tables to store guest information, room assignments, and check-in/check-out details.
   e) Implement features to manage room assignments, including room upgrades and changes.
   f) Include functionality for managing guest profiles!

2) Food Ordering Module:

   a) Design database tables to store food items, recipes, pricing, and order details.
   b) Develop a user-friendly interface for admin to browse food, and place orders for customers.
   c) Implement features for managing food inventory, including stock levels and ingredient tracking.
   d) Integrate with the reservation and billing modules to link orders to guest profiles and room charges.
   e) Develop reporting functionality to track top food sales, and revenue generated.
   f) Create a database to store inventory data, including food items, quantities, and related information.
   g) For managing the inventory, adding new ingredients, updating quantities, and tracking stock levels.

Database: MySQL

Backend Development*: Use PHP as the programming language for the server-side development.*

Frontend Development*: JavaScript, HTML, and CSS*

Security*: user authentication mechanisms (Room admin ,kitchen admin).*

2.4 Testing

We conducted a comprehensive test plan to ensure that all aspects of the hotel management system were thoroughly tested. Our test plan included various types of testing.

- **Unit Testing:**

Test Case 1: Verify that the room booking module correctly registers a new booking.

Test Case 2: Ensure that the kitchen system accurately processes and records food orders.

- **Integration Testing:**

Test Case 3: Validate the seamless integration between the room booking and kitchen system modules.

Test Case 4: Test the synchronization of data between the database and the user interface.

- **System Testing:**

Test Case 5: Perform end-to-end testing of the entire hotel management system, including room booking, food ordering, and data storage.:

- **User Acceptance Testing:**

Test Case 6: Collect input from kitchen staff on the efficiency and accuracy of the food order management system.

- **Usability Testing:**

Test Case 7: Verify that the user interface is user-friendly and intuitive for both admin users.

Test Case 8: Evaluate the system's accessibility and responsiveness across different devices and screen sizes.

- **Security Testing:**

Test Case 9: Assess the system's security measures to ensure protection against unauthorized access.

Test Case 10: Verify the confidentiality and integrity of sensitive data stored in the database.

- **Error Handling Testing:**

Test Case 11: Validate the system's ability to handle and display appropriate error messages for invalid inputs.

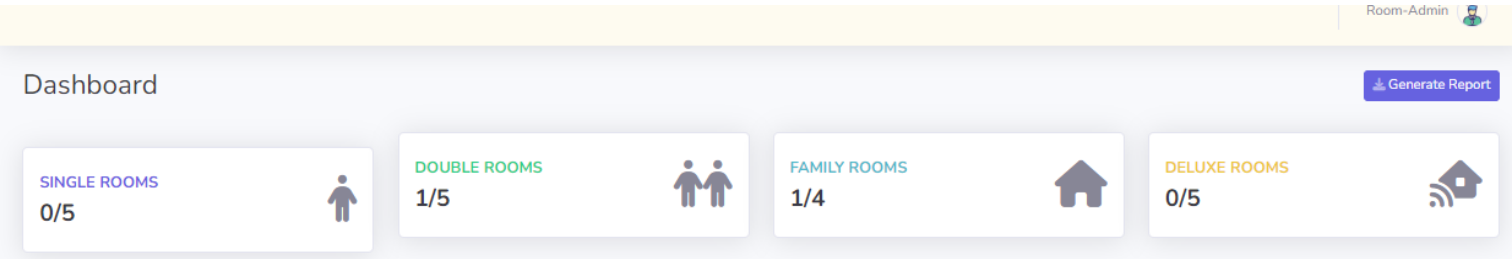
- **Performance Testing:**

Test Case 12: Measure the system's response time for various operations, such as room booking and food ordering.

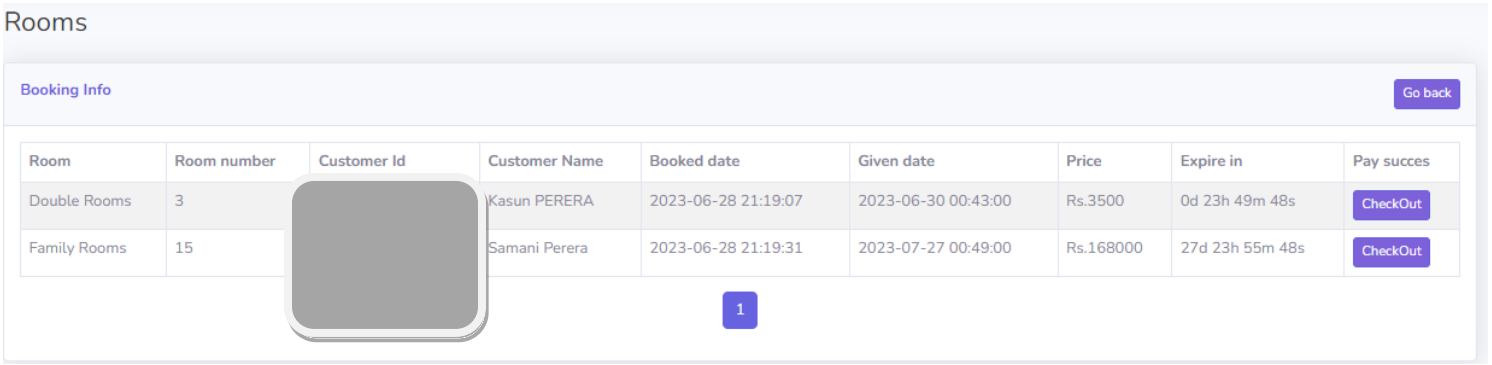
2.4.1 Evidence that all aspects of the system have been tested.

2.4.1.1 Room Dashboard

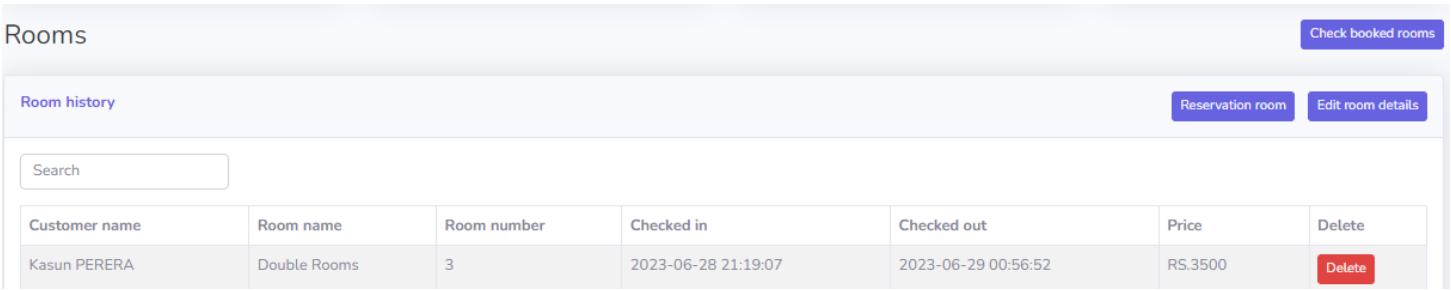
Room booking (Displays current booking categories)



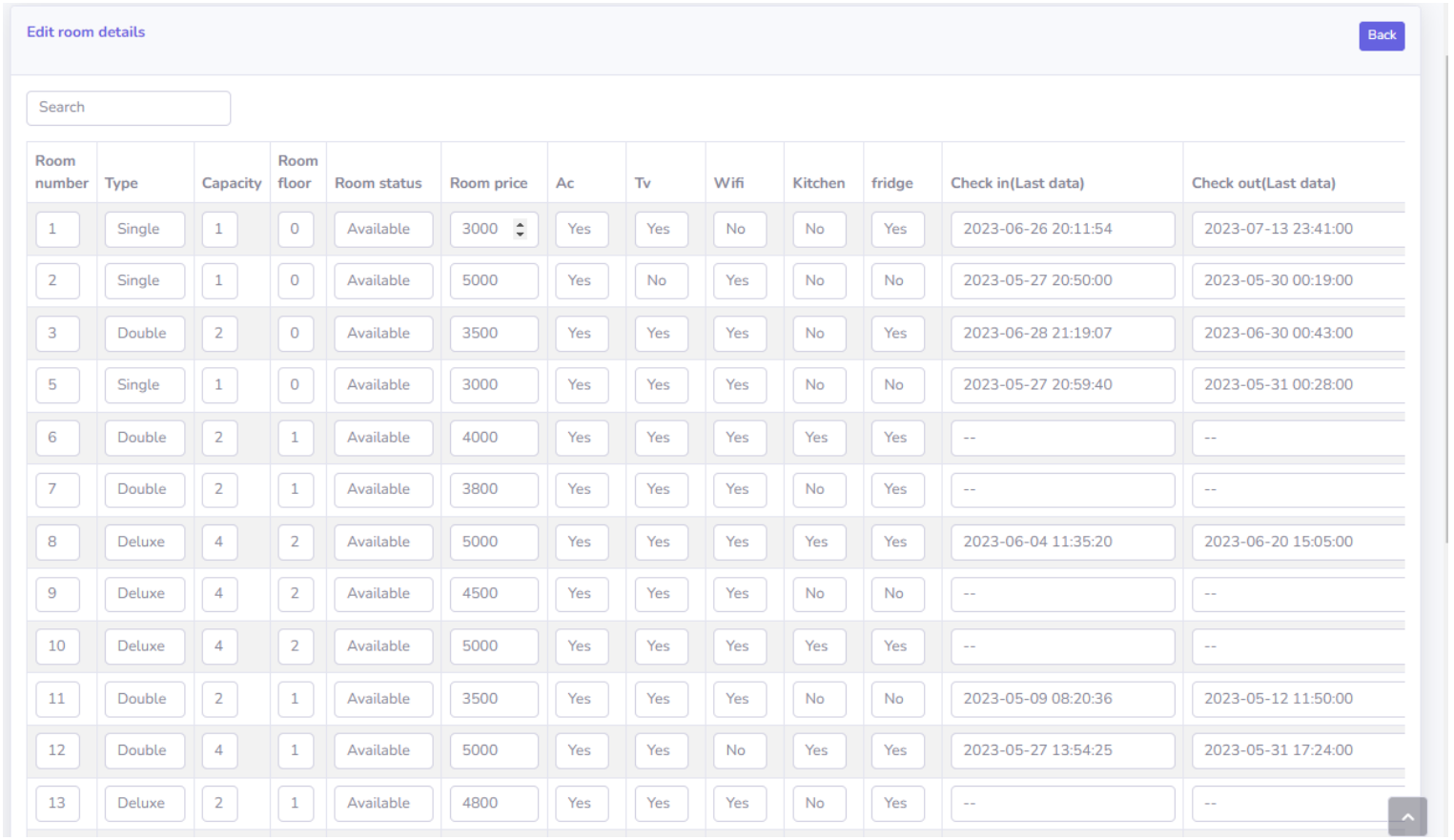
Display current bookings and prices.



Checkout and save in the history table.



Edit room details and availability.



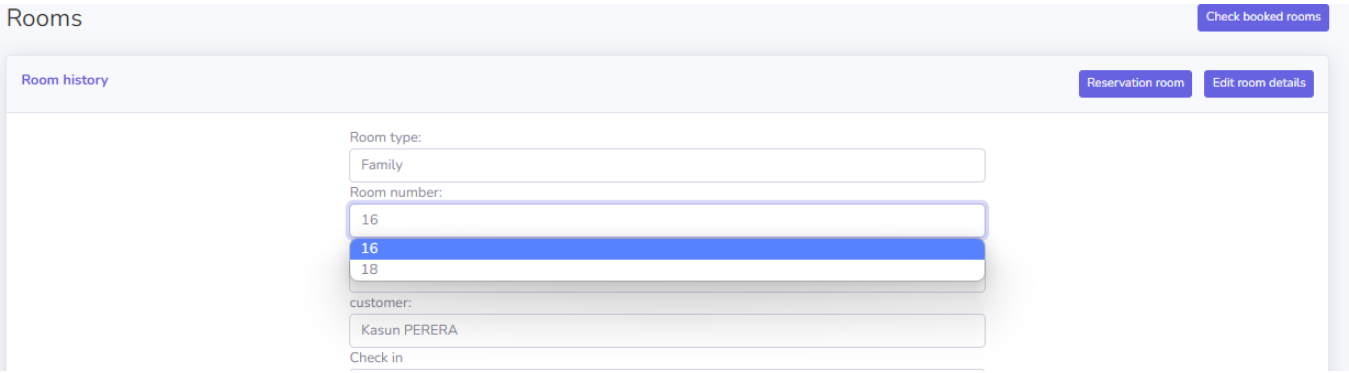
Room number	Type	Capacity	Room floor	Room status	Room price	Ac	Tv	Wifi	Kitchen	fridge	Check in(Last data)	Check out(Last data)
1	Single	1	0	Available	3000	Yes	Yes	No	No	Yes	2023-06-26 20:11:54	2023-07-13 23:41:00
2	Single	1	0	Available	5000	Yes	No	Yes	No	No	2023-05-27 20:50:00	2023-05-30 00:19:00
3	Double	2	0	Available	3500	Yes	Yes	Yes	No	Yes	2023-06-28 21:19:07	2023-06-30 00:43:00
5	Single	1	0	Available	3000	Yes	Yes	Yes	No	No	2023-05-27 20:59:40	2023-05-31 00:28:00
6	Double	2	1	Available	4000	Yes	Yes	Yes	Yes	Yes	--	--
7	Double	2	1	Available	3800	Yes	Yes	Yes	No	Yes	--	--
8	Deluxe	4	2	Available	5000	Yes	Yes	Yes	Yes	Yes	2023-06-04 11:35:20	2023-06-20 15:05:00
9	Deluxe	4	2	Available	4500	Yes	Yes	Yes	No	No	--	--
10	Deluxe	4	2	Available	5000	Yes	Yes	Yes	Yes	Yes	--	--
11	Double	2	1	Available	3500	Yes	Yes	Yes	No	No	2023-05-09 08:20:36	2023-05-12 11:50:00
12	Double	4	1	Available	5000	Yes	Yes	No	Yes	Yes	2023-05-27 13:54:25	2023-05-31 17:24:00
13	Deluxe	2	1	Available	4800	Yes	Yes	Yes	No	Yes	--	--

Set room number17 price 0 for unavailable to room. (Check availability and unavailability)

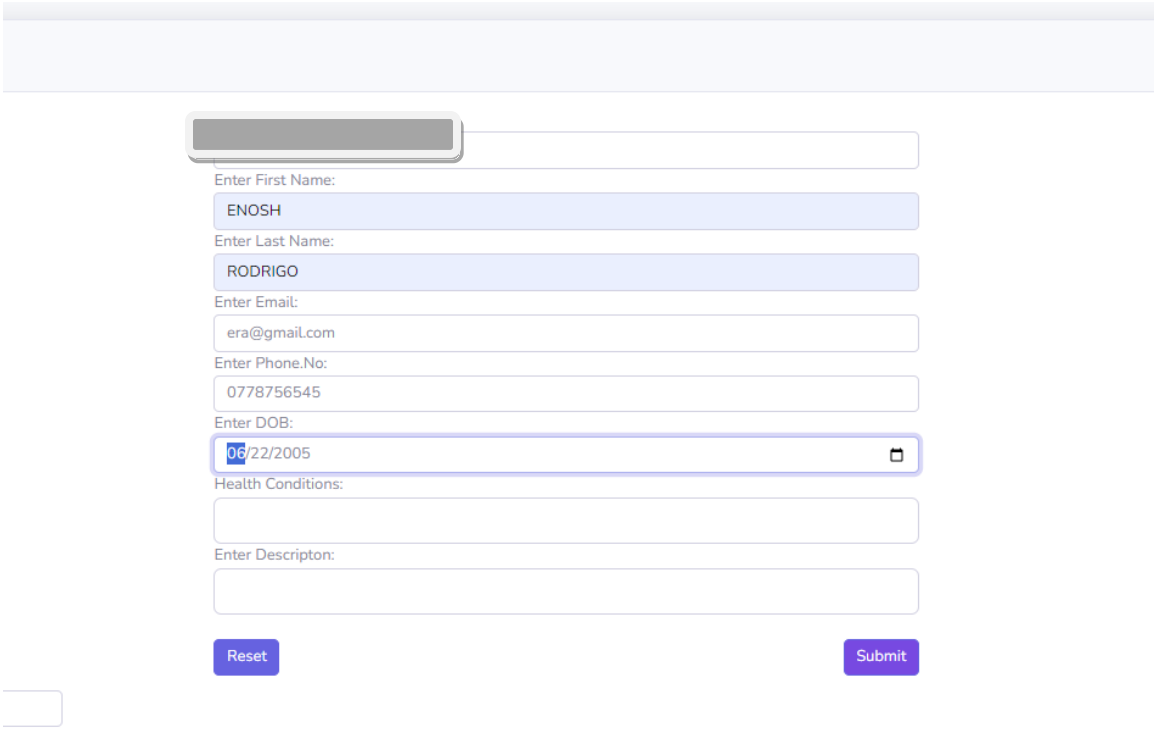
13	Deluxe	2	1	Available	4800	Yes	Yes	Yes	No	Yes	--	--
14	Deluxe	2	1	Available	5000	Yes	Yes	Yes	Yes	Yes	2023-05-27 21:01:24	2023-06-02 00:35:00
15	Family	4	2	Booked	6000	Yes	Yes	Yes	No	No	2023-06-28 21:19:31	2023-07-27 00:49:00
16	Family	4	2	Available	8000	Yes	Yes	Yes	No	Yes	2023-06-03 23:39:43	2023-06-16 03:09:00
17	Family	6	2	Available	0	Yes	Yes	Yes	Yes	Yes	2023-06-24 09:30:17	2023-06-29 13:00:00
18	Family	4	2	Available	7000	Yes	Yes	Yes	Yes	Yes	--	--
19	Single	1	1	Available	3800	Yes	Yes	Yes	No	Yes	2023-04-09 14:50:20	2023-04-11 18:19:00
20	Single	1	1	Available	3500	Yes	Yes	Yes	Yes	No	2023-05-12 07:37:40	2023-05-12 11:06:00

x

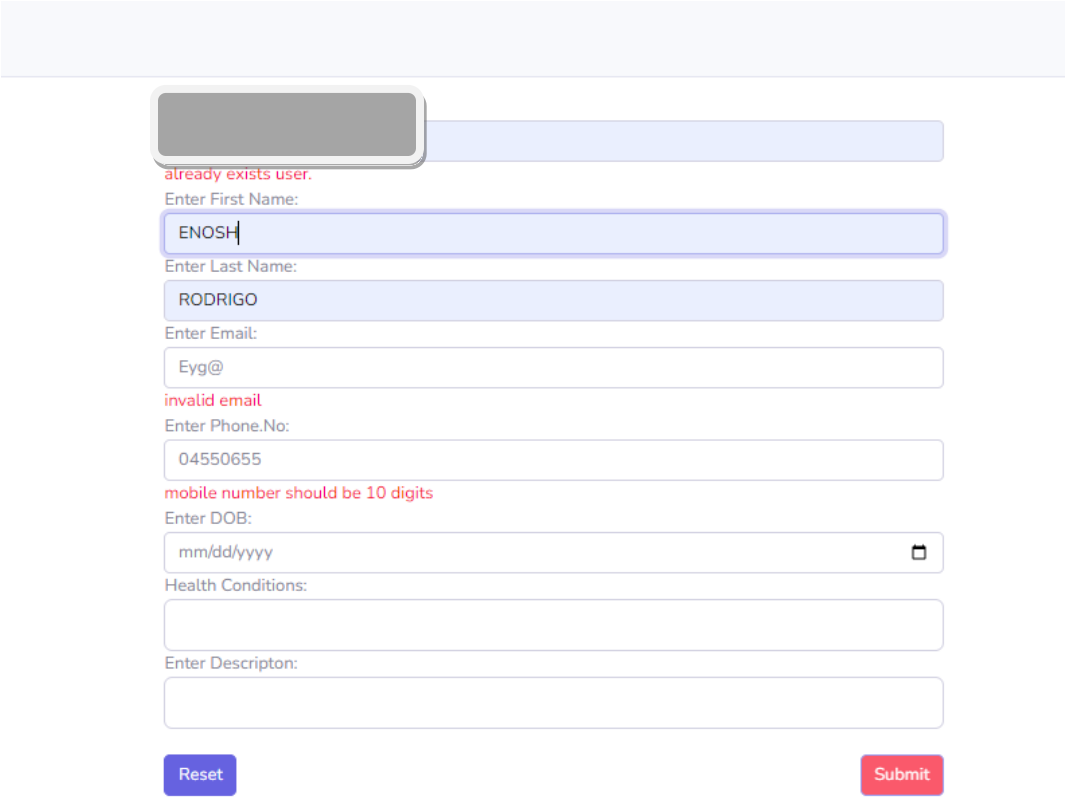
Now below screenshot room 17 not listed in available rooms.



Add customers to the system.

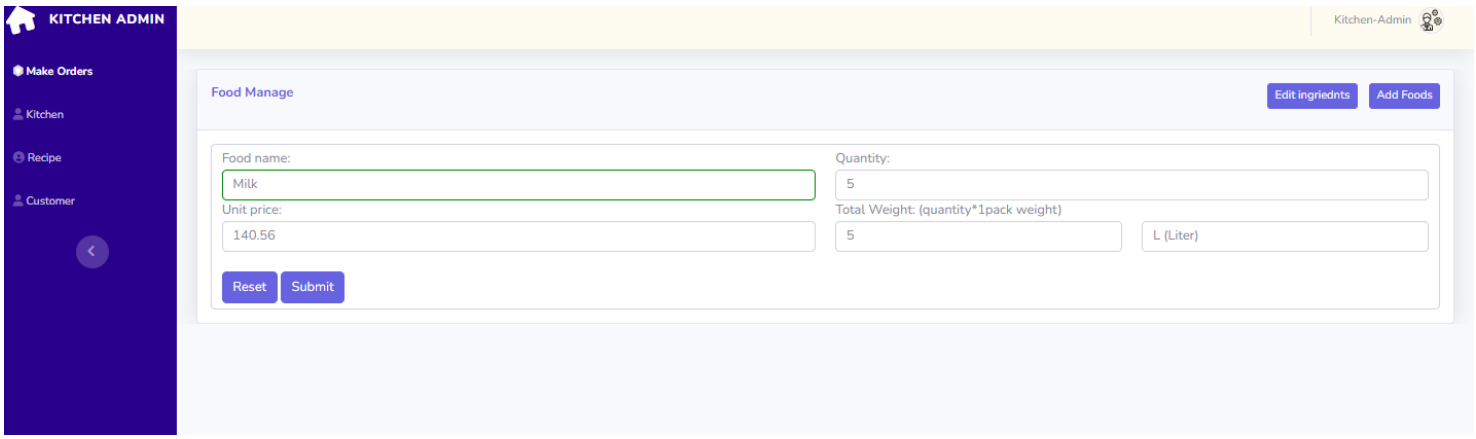


Verify input fields data.

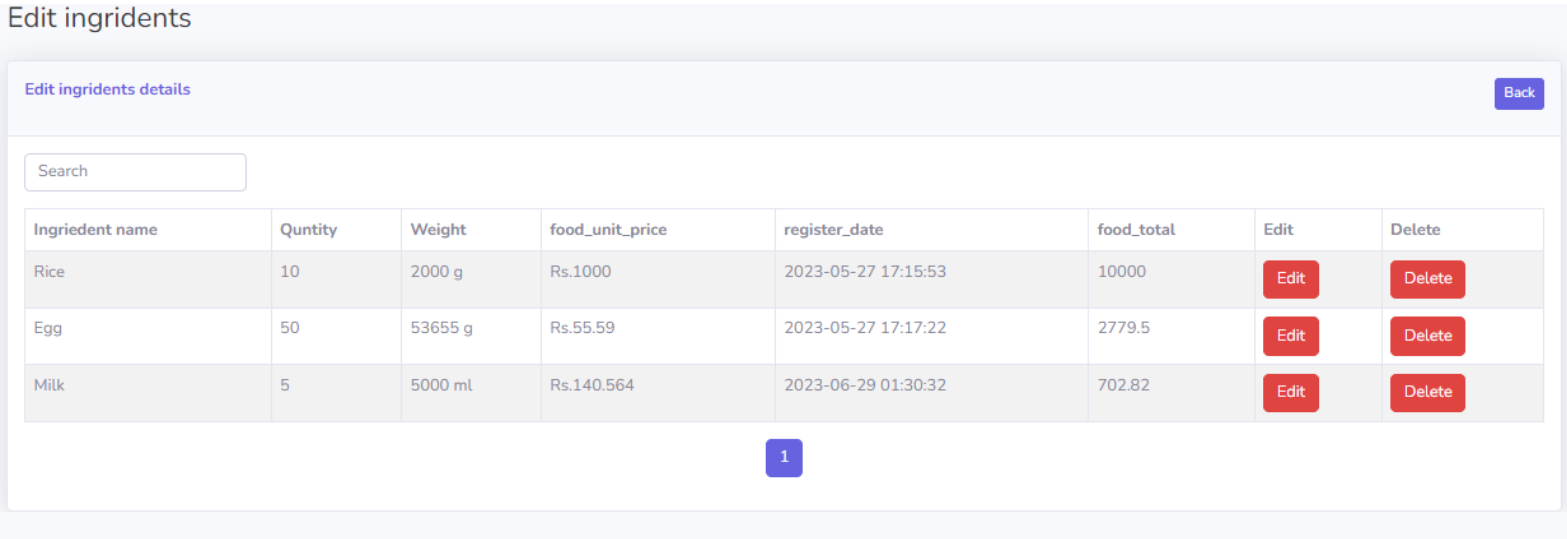


2.4.1.2 Kitchen Dashboard testing

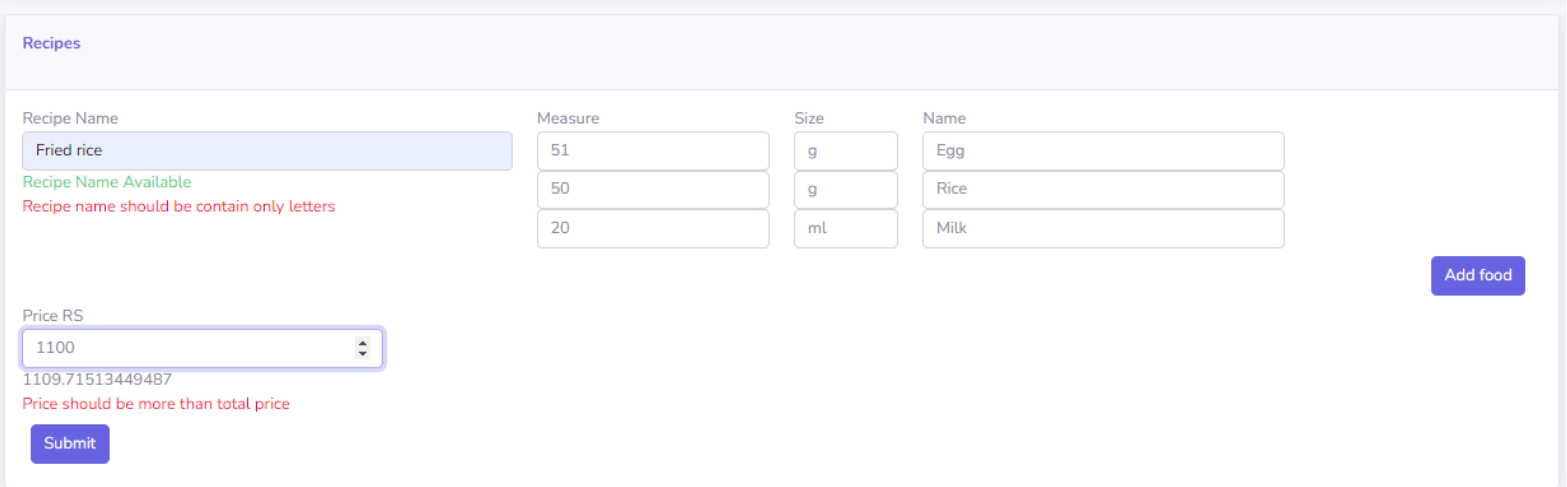
Add food(ingredients) to the system.



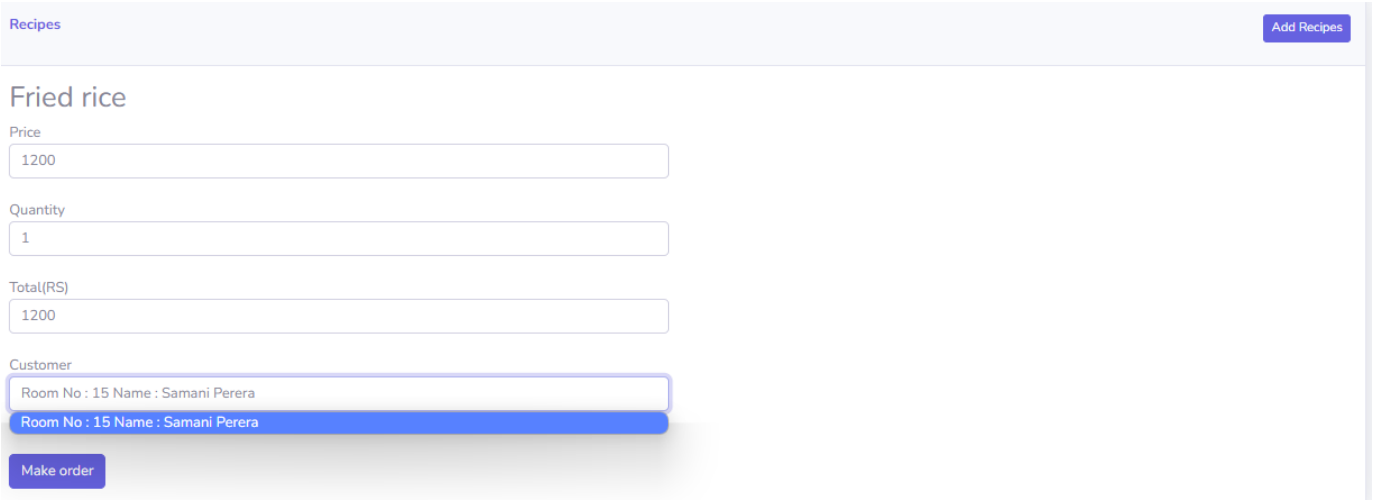
Edit and delete ingredients (Successfully added ingredients to the system)



Make recipes using ingredients. (Display minimum recipe price based on entered values according to food prices in database)



Make orders for current bookings.



When make an order ingredients values are decrease in database

Before the order (ingredients values in database)

food_name	food_quntity	food_weight	food_measurement	food_unit_price	food_img	food_description
Rice	10	2000	g	1000	NULL	
Egg	50	53655	g	55.59	NULL	
Milk	5	5000	ml	140.564	NULL	NULL

Edit Copy Delete Export

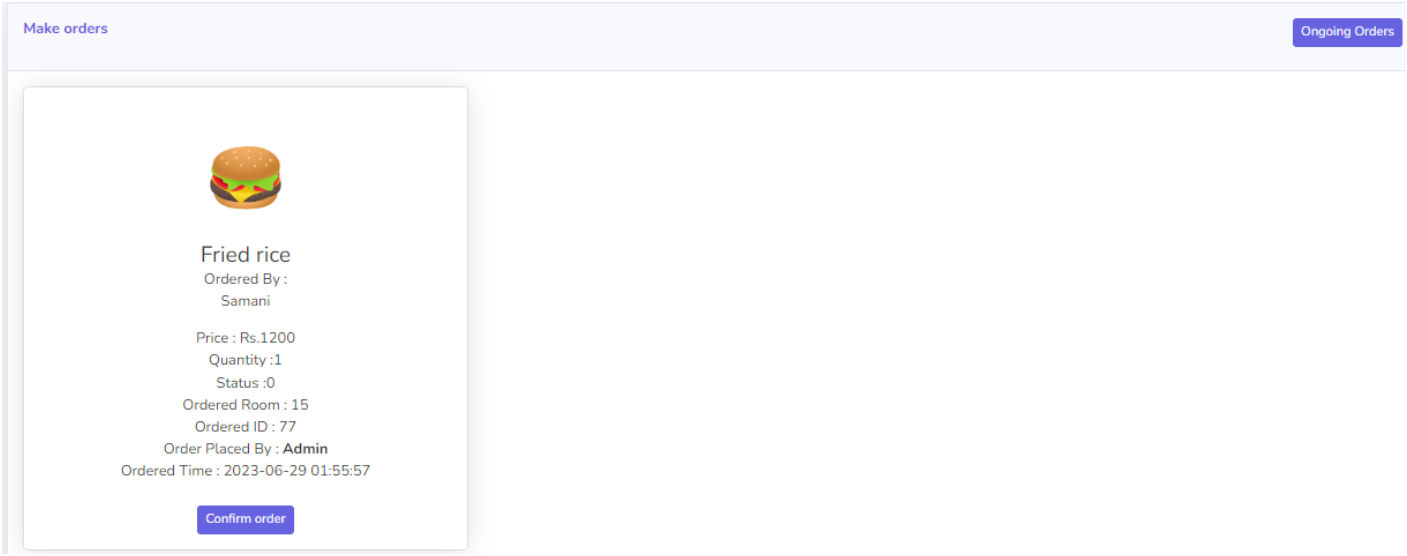
After the order (ingredients values in database)

	food_name	food_quntity	food_weight	food_measurement	food_unit_price	food_img
7	Rice	10	1980	g	1000	NULL
9	Egg	50	53604	g	55.59	NULL
1	Milk	5	4980	ml	140.564	NULL

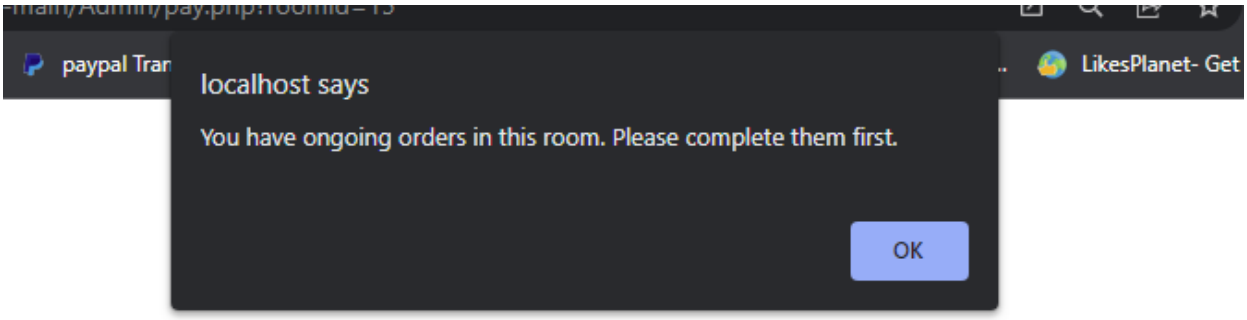
Edit Copy Delete Export

Filter rows: Search this table Sort by key: None

Displaying ongoing orders and confirming orders.

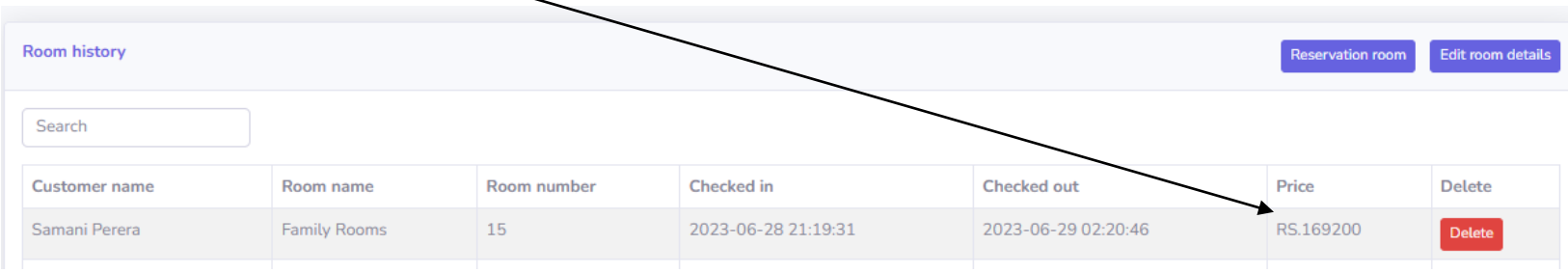


With ongoing orders in rooms admin can't checkout the rooms



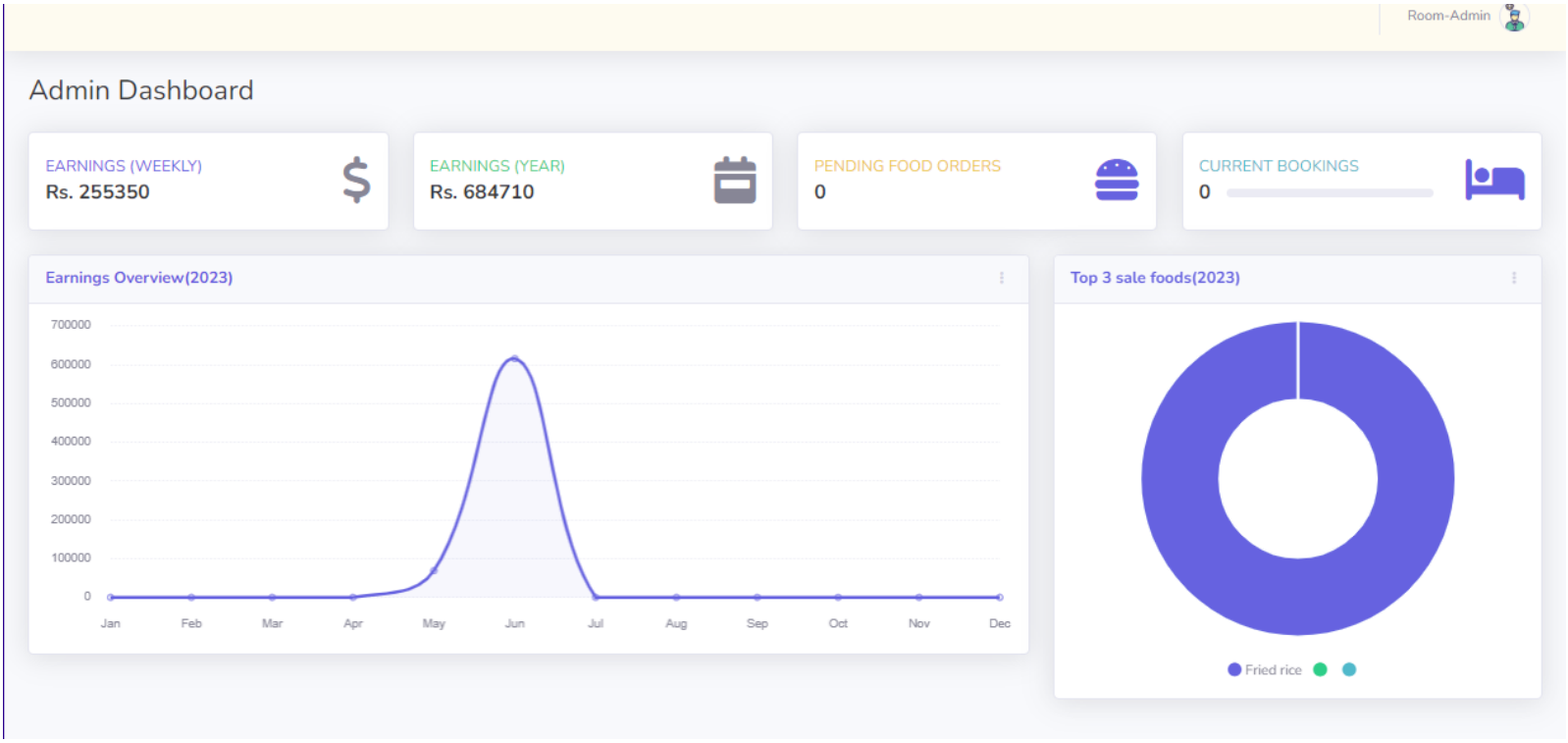
After checking out the room it's shown in the history table with the total price.

Total Price= (1day price of a room * Number Of days Booked) +Ordered food totals



Main dashboard

Display total earnings (weekly, yearly)
Display pending food orders.
Display current bookings.
Earnings overview yearly.
Top 3 sales foods.



3. Evaluation

3.1 Assessment of the Project results

- Overview of the achieved goals and objectives

One of the project goals was to increase customer satisfaction by implementing a user-friendly interface for the hotel management system. This goal was achieved by conducting user research, incorporating user feedback into the design process, and ensuring intuitive navigation and functionality.

- Evaluation of the project's outcomes and deliverables

The project delivered a fully functional hotel management system that included features such as room booking and food ordering. The system was thoroughly tested and met all specified requirements, resulting in a reliable and efficient solution for hotel operations.

- Analysis of the project's performance against key metrics

Key metrics used to evaluate the project's performance included adherence to the project timeline, budget utilization, and customer feedback. The project was completed within the allocated time and budget, and customer feedback indicated a high level of satisfaction with the system's usability and functionality.

- Identification of strengths and weaknesses of the project results

A strength of the project was the successful integration of the hotel management system with existing systems, resulting in improved data accuracy and efficiency.

3.2 Lessons Learned

- Project management lessons

Effective project planning and scheduling ensured that tasks were completed within the designated timeframes, preventing bottlenecks and delays.

Clear communication channels and regular project updates improved stakeholder engagement and minimized misunderstandings.

- Technical lessons

Integrating third-party APIs presented technical challenges. Through thorough testing and collaboration with API providers, compatibility issues were resolved.

Inadequate server capacity caused performance issues. Scaling up server resources and optimizing database queries resolved the performance bottlenecks.

- Team collaboration and communication lessons

Regular team meetings and progress updates fostered a collaborative environment, enabling timely issue resolution and ensuring everyone was aligned with project objectives.

- Stakeholder engagement and feedback

Engaging key stakeholders early in the project and incorporating their feedback during development ensured the final product met their requirements and expectations.

Conducting post-project surveys and soliciting feedback from stakeholders provided valuable insights for future improvements and highlighted areas of success.

3.3 Future Work

- Identified areas for improvement.

1. Enable booking for future dates: Currently, the system only allows room admins to book rooms for the current date. Implementing a feature that enables booking for future dates will provide more flexibility and convenience to customers.
2. Enhance check-out process: Integrate a check-out date field and develop a pricing algorithm based on the duration of the stay. This improvement will ensure accurate billing and streamline the check-out process.

- Recommendations for future projects

1. Develop a customer-facing interface: Create a dedicated interface for customers to make room reservations and place food orders. This user-friendly interface will enhance the overall customer experience.
2. Implement order confirmation for kitchen admins: Introduce a Delete step for kitchen admins when Confirm orders. This will improve order accuracy.

- Potential next steps

1. Integrate online payment functionality: Incorporate secure online payment options, such as credit card or digital wallet integration, to facilitate convenient and secure payments for bookings and orders.
2. Implement room availability tracking: Develop a real-time room availability tracking feature that allows Customers to check the current availability status of rooms before making a booking.

- Project sustainability and long-term implications

1. Regular maintenance and updates: Establish a maintenance plan to ensure the system remains secure, reliable, and up to date with the latest technologies and security patches.
2. User feedback and continuous improvement: Actively seek feedback from users and engage in continuous improvement efforts to address their needs and preferences, ensuring the long-term success and relevance of the system.

By considering these areas for improvement, recommendations for future projects, potential next steps, and the project's sustainability and long-term implications, we can further enhance the hotel management system and provide an even better experience for both the hotel staff and customers.

4. Conclusion

The Hotel Management System project has successfully achieved its objectives and goals, providing a comprehensive solution to streamline and automate hotel operations. The system will encompass various key functionalities, including room booking and food ordering. Throughout the project, we addressed the challenges faced by the hospitality industry, resulting in improved efficiency, reduced errors, and enhanced guest satisfaction.

However, it is important to acknowledge certain weaknesses and limitations of the proposed technique. These include:

- Integration Challenges: The Hotel Management System may face integration challenges with existing systems or third-party applications used by the hotel. It is crucial to ensure seamless integration and data synchronization to avoid data inconsistencies and operational disruptions.

- Scalability: As the hotel's operations grow, the system should be scalable to accommodate increasing demands and a larger volume of data. This may require additional resources and infrastructure to support the growing needs of the hotel.

To address these limitations, the following solutions can be considered:

- Customization and Integration Support: Provide customization options and integration support to tailor the system according to the specific requirements of each hotel. This includes providing APIs or interfaces to seamlessly connect the Hotel Management System with other existing systems.

- Scalable Architecture: Design the system architecture to be scalable and flexible, allowing for easy expansion and accommodating a larger user base and increased data volumes. Implementing cloud-based solutions and adopting technologies like containerization can help ensure scalability.

Benefits of developing this project for the client organization:

- Improved Operational Efficiency: The Hotel Management System automates manual processes, reducing the time and effort required for administrative tasks. This leads to increased operational efficiency, allowing hotel staff to focus more on delivering exceptional guest experiences.

- Accurate Reporting and Decision-Making: The system provides comprehensive reports and analytics, offering valuable insights into key performance indicators, occupancy rates, revenue. This empowers management to make data-driven decisions, identify areas for improvement, and optimize hotel operations.

- Increased Revenue Generation: By efficiently managing room reservations, optimizing pricing strategies, and capturing additional revenue through ancillary services, the Hotel Management System helps maximize revenue generation for the hotel.

5. References

Nym. (2022, 09 17). *Activity Diagram for Hotel Management System*. Retrieved from ITsourcecode: https://itsourcecode.com/uml/hotel-management-system-activity-diagram-uml/

S.A.M.A.P.D, S. (2015). *Web based hotel management system for hotel araliya.*

Zahid, M. (n.d.). *Hotel Management System UseCase* . Retrieved from Creately: https://creately.com/diagram/example/hijr9igf1/hotel-management-system-usecase-classic