

Web-based Hotel Reservation and Customer Feedback Management System for Sitara Hotel Weligama

W.A.K B. Sewwandi

2025



Web-based Hotel Reservation and Customer Feedback Management System for Sitara Hotel Weligama

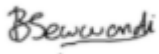
**A dissertation submitted for the Degree of Master of
Information Technology**

**W.A.K Buddhima Sewwandi
University of Colombo School of Computing
2025**

Declaration

Name of the student: W.A.K Buddhima Sewwandi
Registration number: MIT/2022/071
Name of the Degree Programme: Master's in information technology
Project/Thesis title: Web-based Hotel Reservation and Customer Feedback Management System for Sitara Hotel Weligama

1. The project/thesis is my original work and has not been submitted previously for a degree at this or any other University/Institute. To the best of my knowledge, it does not contain any material published or written by another person, except as acknowledged in the text.
2. I understand what plagiarism is, the various types of plagiarism, how to avoid it, what my resources are, who can help me if I am unsure about a research or plagiarism issue, as well as what the consequences are at University of Colombo School of Computing (UCSC) for plagiarism.
3. I understand that ignorance is not an excuse for plagiarism and that I am responsible for clarifying, asking questions and utilizing all available resources in order to educate myself and prevent myself from plagiarizing.
4. I am also aware of the dangers of using online plagiarism checkers and sites that offer essays for sale. I understand that if I use these resources, I am solely responsible for the consequences of my actions.
5. I assure that any work I submit with my name on it will reflect my own ideas and effort. I will properly cite all material that is not my own.
6. I understand that there is no acceptable excuse for committing plagiarism and that doing so is a violation of the Student Code of Conduct.

Signature of the Student	Date (DD/MM/YYYY)
	05/07/2025

Certified by Supervisor(s)

This is to certify that this project/thesis is based on the work of the above-mentioned student under my/our supervision. The thesis has been prepared according to the format stipulated and is of an acceptable standard.

	Supervisor 1	Supervisor 2	Supervisor 3
Name	Dr. Kasun Karunanayaka		
Signature			
Date	05/07/2025		

Abstract

Sitara Hotel, located in Weligama, currently lacks an official web application for managing hotel reservations and customer feedback, which has led to operational inefficiencies and increased costs. At present, the hotel relies on a third-party solution to handle reservations, resulting in high commission fees and limited control over the booking process. Additionally, staff operations, including managing reservations and handling customer feedback, are performed manually, increasing the chances of human error and reducing overall efficiency. This project aims to solve these challenges by developing a fully functional Hotel Reservation and Customer Feedback Management System tailored to Sitara Hotel's unique requirements.

The proposed system includes a web application to enable direct reservations, eliminating the need for third-party services and reducing operational costs. The reservation module allows guests to check room availability, make reservations. The system also includes a comprehensive feedback management feature, enabling guests to submit reviews and complaints, feedback directly through the platform. A key feature of the system is the lengthy review summarization function, which analyzes and summarizes guest reviews to provide hotel management with actionable insights into customer satisfaction and service improvements.

Additional reservation options, such as meal reservations, cab services, and spa bookings, are integrated into the platform to enhance the overall customer experience. The system follows the MVC (Model-View-Controller) architecture and was developed using the Laravel framework with PHP as the backend language. The front end is designed using Bootstrap and CSS to ensure a responsive and user-friendly interface. The project was developed using the Iterative and Incremental development model, allowing for continuous improvement and flexibility throughout the development process. The system also includes secure user authentication, data encryption, and role-based access control to ensure data privacy and security.

By implementing this system, Sitara Hotel can improve booking efficiency, reduce operational costs, enhance customer satisfaction, and establish a direct connection with its guests, thereby strengthening its market presence and service quality.

Acknowledgement

I would like to express my sincere gratitude to all those who supported and guided me throughout the completion of this project.

First and foremost, I am deeply grateful to my project supervisor, Dr. Kasun Karunanayaka, Senior Lecturer at the University of Colombo School of Computing (UCSC), for his invaluable guidance, continuous support, and constructive feedback throughout the development of this project. His expertise and insightful suggestions have been instrumental in refining and enhancing the quality of my work.

I would also like to extend my heartfelt thanks to the course coordinator of the MIT program and the academic staff of UCSC for providing me with the necessary knowledge, resources, and encouragement to complete this project successfully. Their support and expertise have been essential in helping me overcome various challenges during the project development.

I am especially thankful to Ms. Sandra Kelaart, the Manager of Sitara Hotel, for her valuable input and cooperation. Her insights into the hotel's operational challenges and requirements helped me design a system that meets the hotel's specific needs and improves its overall efficiency.

Finally, I would like to express my gratitude to my family and friends for their unwavering support, encouragement, and understanding throughout this journey. Their motivation and belief in my abilities have been a driving force behind the successful completion of this project.

Table of Contents

Declaration.....	i
Abstract.....	
Acknowledgement	iii
List of Abbreviations	xi
Chapter 1 – Introduction	1
1.1 Project Overview	1
1.2 Motivation.....	3
1.3 Objectives	4
1.4 Background of study	4
1.5 Scope of Project	5
1.6 Feasibility Study	6
1.7 Structure of the Dissertation	7
1.8 Summary	8
Chapter 2 – Background	9
2.1 Introduction.....	9
2.2 Data Collection Procedures.....	11
2.3 Requirement Analysis.....	12
2.3.1User Classes and Characteristics	13
2.3.2System Requirements.....	14
2.3.3 Design Implementation Constraints.....	15
2.3.4Functional Requirements	15
2.3.5Non-Functional Requirements	18
2.4 Review of Similar System	19
2.5 Related Technology	24

2.6 Summary	26
Chapter 3 -Design	27
3.1 Introduction.....	27
3.2 Related System Design Strategy	27
3.3 System Architecture.....	29
3.4 UML Diagrams for Proposed System.....	31
3.4.1 Use case Diagram	31
3.4.2. Use Case Narratives.....	34
3.4.3Activity Diagram	49
3.5 Database Design.....	53
3.5.1.ER Diagram	53
3.5.2 Class Diagram.....	55
3.5.3. Database Structure	56
3.4 Chapter Summary	56
Chapter 4 – Implementation.....	57
4.1 Introduction.....	57
4.2 Related Technologies and Tools	57
4.3 Modules of the System.....	58
4.4 Application Development Architecture	59
4.5 Middleware	60
4.6 Main User Interfaces and Coding	61
4.6.1 Room Reservation and Availability Checking	61
4.6.4. Additional Service Reservation (Meal, Cab Services, Payment Services).....	67
4.6.5. Add Review Page.....	69
4.6.6. Lengthy Review Summarization.....	69

4.6.7. Contact Us page	71
4.6.8. Dashboard	72
4.6.9. Employee Management	74
4.6.10. Room Management.....	75
4.6.11. Report Generation.....	76
4.6.13. Reservation Management.....	78
4.6.14. House Keeping Ticket Management.....	79
4.7. Code Structure	80
4.8. Summary	83
Chapter 5 – Testing and Evaluation.....	84
5.1 Introduction.....	84
5.2 Testing types	84
5.3 Test Cases and Test Case Design Techniques	85
5.4 Test Results	94
5.5 User Evaluation.....	96
5.6 Summary	102
Chapter 6 -Conclusion	103
6.1 Overview of the solution.....	103
6.2 Lesson Learnt.....	103
6.3 Future Improvement.....	104
6.4 Summary	105
References	106
Appendix A- Technical Documentation	108
Appendix B- MIS Report.....	109
Appendix C- User Manual	114

List of figures

Figure2 1:Current system in sitara hotel.....	10
Figure2 2:Booking.com	20
Figure2 3:Tripadvisor	21
Figure2 4:Hotels.com.....	21
Figure2 5:Anantaya.com.....	22
Figure3 1-UML Diagram.....	28
Figure3 2-System Architecture Diagram	29
Figure3 3:Use case Diagram.....	33
Figure3 4:Activity Diagram -Room reservation	50
Figure3 5:Activity Diagram-Spa Service reservation.....	51
Figure3 6:Activity Diagram-Additional meal reservation.....	52
Figure3 7:Feedback Management.....	53
Figure3 8:ER Diagram.....	54
Figure3 9:Class Diagram	55
Figure3 10:Database Design.....	56
Figure4 1:Modules of the system.....	58
Figure4 2:Application Development Architecture.....	59
Figure4 3:Middleware.....	60
Figure4 4:Auth Controller.....	60
Figure4 5:Room availability checking.....	61
Figure4 6:Room availability Checking -code	61
Figure4 7:Create Discounts.....	62
Figure4 8:Apply the Discount.....	62
Figure4 9:Discounts- code	63
Figure4 10:Payment Summary	64
Figure4 11:Payment Summary-code.....	65
Figure4 12:Payment Gateway Integration	65
Figure4 13:Payment Gateway Integration-code	66
Figure4 14:Meal Reservation UI	67
Figure4 15:Meal Reservation -code.....	67
Figure4 16:Cab service Reservation UI.....	67
Figure4 17:Cab service Status-code.....	68
Figure4 18:Spa service Reservation UI	68
Figure4 19:Spa Services management-code	68
Figure4 20:Add Review UI.....	69
Figure4 21:Lengthy review summarization UI.....	70
Figure4 22:Review Summarization-model	70
Figure4 23:Review Summarization-Model Integration	71

Figure4 24:Contact us client page.....	72
Figure4 25:Contact us management -Admin UI.....	72
Figure4 26:Contact us-code	72
Figure4 27:Admin Dashboard UI	73
Figure4 28:Admin Dashboard -code.....	73
Figure4 29:Front Desk Officer Dashboard-UI.....	73
Figure4 30:Front Desk Officer Dashboard-code	73
Figure4 31:Employee Management UI.....	74
Figure4 32:Employee Management -code	74
Figure4 33:Room Management UI	75
Figure4 34:Room Management-code	75
Figure4 35:Room Availability Report	76
Figure4 36:Room Availability UI.....	76
Figure4 37:Room Availability Report -code	76
Figure4 38:Employee Information UI	77
Figure4 39:Employee Information Report.....	77
Figure4 40:Employee Information Report-code	77
Figure4 41:Reservation Management-code	78
Figure4 42:Reservation Management UI.....	78
Figure4 43:Housekeeping task Management-code.....	79
Figure4 44:Housekeeping task Management UI.....	79
Figure4 45:Model_guest.php	80
Figure4 46:View_Register.blade.php	81
Figure4 47:Controller_guestcontroller.php.....	82
FigureA 1:Database connection.....	108
FigureB 1:Room availability reportFigure 5-6: Survey Feedback III	99
FigureB 2:Report Generation Menu	109
FigureB 3>User_ Interface to generate the room availability report	110
FigureB 4:Room availability report.....	110
FigureB 5>User interface to generate top claims report	111
FigureB 6:Top claims report.....	111
FigureB 7>User Interface to generate the hotel reservation report.....	112
FigureB 8:Hotel reservation report	112
FigureB 9>User Interface to generate the transaction report.....	113
FigureB 10:Transaction Report.....	113
FigureC 1:Check Availability	114
FigureC 2:Pick a Room.....	115
FigureC 3:Guest Detail Capturing UI.....	115
FigureC 4:Payment Confirmation UI.....	116
FigureC 5:Email Confirmation UI.....	116
FigureC 6:Meal Services Booking.....	117
FigureC 7:cab services booking.....	117
FigureC 8:spa Services booking	117

FigureC 9:Add Reviews.....	118
FigureC 10:Review Summarization.....	118
FigureC 11:Contact us UI	119
FigureC 12:Front Desk Officer Dashboard.....	119
FigureC 13:Mark Check in and check out	120
FigureC 14:House keeping ticket management	120
FigureC 15:Admin Dashboard.....	121
FigureC 16:Room Management UI	121
FigureC 17:Employee management UI	122

List of Tables

Table2 1:Hardware requirements.....	14
Table2 2:Software requirements	15
Table2 3:Related similar work comparison	23
Table3 1:Use case narratives for admin user create employee	34
Table3 2:Use case narratives for the employee login	35
Table3 3:Use case narratives for the admin user add rooms	36
Table3 4:Use case narrative for search room availability.....	37
Table3 5:Use case narrative for room selection.....	38
Table3 6:Use case narratives for the guest signup.....	40
Table3 7:Use case narratives for the guest sign in.....	41
Table3 8:Use case narratives for the make payment	42
Table3 9:Use case narratives for submit feedback	43
Table3 10:Use case narrative for the add complaints	44
Table3 11:use case narratives for the add reviews.....	45
Table3 12:Use case narratives for lengthy review summarization mechanism	46
Table3 13:Use case narratives for front desk officer mark the check in and check out	47
Table3 14:Use case narratives for viewing customer complaints.....	48
Table3 15:Use case narratives for housekeeping ticket management	49
Table4 1:Related Technologies and Tools.....	57
Table4 2:Testcase writing table	94

List of Abbreviations

UAT-User Acceptance Testing

UI -User Interface

GUI-Graphical User Interface

SDLC-Software Development Life Cycle

HTML-Hypertext Markup Language

SQL-Structured Query Language

UML-Unified Modeling Language

DBMS-Database Management System

CSS-Cascading Style Sheets

PHP-PHP Hypertext Preprocessor

UML-Unified Modeling Language

URL-Uniform Resource Locator

Chapter 1 – Introduction

1.1 Project Overview

Online booking platforms are essential for providing guests with convenience while optimizing hotel management and maximizing revenue (Pitt, 2021). However, Sitara Hotel Weligama currently faces significant challenges due to the absence of an official hotel reservation website. As a result, the hotel relies on third-party booking platforms, which charge substantial commission fees, impacting overall profitability. Additionally, the existing feedback collection mechanisms are limited, restricting the hotel's ability to gather and analyze valuable customer insights for service improvement. Additionally, manual systems for staff operations create inefficiencies, communication gaps, and risks of data loss, further hindering operational effectiveness.

The proposed system provides an integrated platform to optimize hotel operations, enhance customer satisfaction, and support data-driven decision-making. It features a room reservation module that allows guests to check real-time availability, view detailed room descriptions and photos, and make reservations. Real-time booking confirmations and email notifications streamline the process. Guests have payment options. They can either make an online payment or pay via cash or card upon check-in. Additionally, the system allows users to predefine their arrival date and time, reducing delays and ensuring a seamless check-in experience with the best possible service.

To enhance guest convenience, the hotel website will include a dedicated module for booking additional services such as cab transportation, meals, and other amenities. This feature will allow guests to customize their stay by selecting transportation options, meal preferences, and special services like spa treatments.

Review systems allow hotels to continuously monitor guest satisfaction, ensuring that improvements are made in real-time (Pitt, 2021). To achieve the system incorporating a review management module to collect, store, and analyze guest reviews. By leveraging machine learning algorithms, the system can summarize lengthy customer reviews, assisting future guests in making informed booking decisions and providing actionable insights for hotel management.

Sitara Weligama Hotel will implement an automated promotional module to attract new customers and retain existing guests through marketing campaigns. This strategy will improve customer engagement and drive repeat visits, strengthen brand loyalty, and contribute to overall business growth.

The Front Desk Officer is responsible for managing guest interactions, reservations, check-ins, check-outs, and customer inquiries. This module will automate the check-in and check-out processes, handle reservation confirmations, process guest requests, and update the status of rooms in real-time. The officer will also handle customer complaints and feedback, ensuring prompt action and maintaining a high level of guest satisfaction

The Admin Officer will oversee the hotel's daily operations, including staff management, reporting. This module will allow the Admin Officer to monitor hotel performance through dashboards, generate reports on bookings, guest feedback, and operational status. Housekeeping staff managing room cleaning This module will automate the assignment of cleaning tasks, allowing staff to update the status of rooms after cleaning is completed.

The project will follow the Iterative and Incremental Development Model to ensure flexibility, continuous improvement, and risk management. This approach allows the system to be developed in smaller, manageable increments, enabling early testing and feedback collection. Each iteration refines existing features while adding new functionalities, ensuring a user-friendly and efficient system. By addressing potential issues early and adapting to changing requirements, the model enhances overall system quality and reduces development risks.

Project includes Quality Assurance testing to ensures reliability, security, and user satisfaction. It includes functional testing, usability testing, Security testing, Retesting and Regression Testing.

By implementing this system, Sitara Hotel Weligama aims to significantly enhance operational efficiency, improve guest satisfaction, and establish a distinct digital identity in the hotel industry. The integrated approach addresses the needs of both guests and staff, fostering a more seamless, efficient, and satisfying hotel experience while strengthening the hotel's competitive edge in the hospitality industry

1.2 Motivation

In today's competitive hospitality industry, hotels must ensure efficient operations and exceptional customer experiences to sustain and grow their business. At Sitara Hotel Weligama, room reservations and customer feedback are currently managed using third-party platforms, which create several challenges. These external platforms impose significant commission fees, reducing the hotel's profitability, and limit the hotel's ability to directly engage with guests.

Similarly, manual processes used by the hotel staff for daily operations, such as maintaining guest details, managing bookings, and handling complaints, lead to inefficiencies, communication delays, and human errors. Paper-based records are also susceptible to loss, theft, or damage, further compromising the accuracy and security of critical data. The lack of an integrated feedback management system prevents timely collection and analysis of customer feedback, which is vital for identifying areas of improvement and enhancing the guest experience.

Developing a centralized, automated system that manages reservations, customer feedback, and staff operations can mitigate these challenges. Such a system would save valuable time for staff by automating repetitive tasks and streamlining operations. It would allow the hotel to focus on guest concerns proactively and ensuring memorable experiences. Additionally, a secure and efficient digital platform would eliminate the risks associated with paper-based records and third-party systems by improving data reliability and accessibility.

Web-based reservation and feedback management system, it can provide real-time booking options, offer tailored promotions, and generate actionable insights from customer feedback and reviews. Guests would enjoy enhanced convenience, such as easy access to room availability with additional customized options, secure online payments, and notifications, while staff would benefit from streamlined workflows and improved communication.

The motivation for implementing this system to optimize operational efficiency, enhance customer satisfaction, and leverage technology for data-driven decision making. Sitara Hotel Weligama can position itself as a modern and efficient hospitality provider. This transformation will not only improve profitability and brand identity but also benefit guests and staff by creating a seamless and satisfying user experience.

1.3 Objectives

- **Secure Data Management:** Provide a reliable and secure digital platform for storing, managing, and retrieving hotel booking details, guest feedback, and staff information, ensuring smooth and uninterrupted operations.
- **Enhance Customer Feedback Collection and Analysis:** Create a feedback system that collects stores and analyzes guest reviews in real time to drive continuous service improvements.
- **Streamline Daily Operations for Hotel Staff and Reservation and Booking processes**
- **Build a Recognizable Brand Identity through Unique System Features:** To create a distinctive system interface that reflects the Sitara Hotel brand, enhancing guest recognition and brand loyalty. integrate a review summarization feature powered by a machine learning algorithms to assist guests in making reservation decisions.
- **To provide comprehensive MIS reports and dashboards covering hotel bookings, customer satisfaction, and guest feedback, enabling informed decision-making and continuous service improvement.**

1.4 Background of study

The hospitality industry has undergone significant transformation in recent years, driven by the increasing need to enhance customer experience and optimize operational efficiency. A key aspect of this evolution is the integration of advanced technological solutions, particularly online booking systems and customer review management tools. These innovations not only facilitate streamlined operational workflows but also enable hotels to tailor their services based on real-time feedback from guests. In an increasingly competitive market, the ability to effectively manage and leverage guest reviews has become a crucial determinant of customer satisfaction, playing a pivotal role in enhancing service quality and operational performance.

Incorporating feedback management systems helps hotels to identify trends in guest satisfaction, improving both operational processes and guest loyalty (Bardi, 2011). Such systems allow hotel managers to stay informed about customer needs and ensure that any issues are addressed promptly, ensuring high service standards (Solomon, 2016). As the industry continues to move

towards a more customer-centric approach, understanding the importance of feedback and its impact on service quality is essential for hotel management and operational strategies (Shore, 2008).

1.5 Scope of Project

The Hotel Reservation and Customer Feedback Management System is designed to provide a comprehensive, centralized platform that streamlines the operational and customer engagement processes of Sitara Hotel Weligama. This system addresses critical aspects of hotel management, including room reservations, customer feedback handling, service bookings, and operational efficiency, with a focus on enhancing both guest experience and staff productivity.

The project encompasses the development of a web-based system that enables guests to seamlessly reserve rooms, cab services, and meals. Real-time room availability and pricing are displayed, allowing customers to make informed decisions. The system also supports multiple payment options, including online and on-arrival payments, to ensure a convenient and flexible booking experience. Reservation confirmations are automated through email notifications, reducing manual workload for front desk staff.

Customer feedback and review management are integral components of the system. Guests can submit feedback, complaints, and reviews through the platform, enabling the hotel to collect and analyze customer sentiments effectively. A machine learning-powered review summarization feature ensures that lengthy reviews are condensed into actionable insights, aiding management in decision-making and enhancing the guest experience.

The system also integrates promotional offers. This functionality fosters repeat bookings and strengthens customer relationships. Staff operations are further streamlined through features such as real time notifications, role-based access control, and an administrative dashboard for managing room details, reservations, and reporting. The reporting module provides analytics on key performance indicators (KPIs), enabling data-driven decisions to improve operational efficiency and service quality.

The proposed system ensures scalability, security, and user-friendliness to cater to the needs of both guests and staff. By transitioning from manual and third-party dependent processes to a

centralized digital solution, Sitara Hotel Weligama can modernize its operations, reduce operational inefficiencies, and establish a distinct brand identity in the competitive hospitality market

1.6 Feasibility Study

The feasibility study evaluates the technical, operational, economic, schedule, and resource aspects of implementing the Hotel Reservation & Customer Feedback Management System for Sitara Hotel Weligama. The study ensures that the proposed solution is viable and aligns with the objectives of the hotel.

Technical Feasibility

Laravel, an open-source PHP framework, is ideal for the proposed system at Sitara Hotel due to its scalability, reliability, and cost-effectiveness. Its compatibility with existing infrastructure ensures smooth integration, while built-in tools simplify development.

Operational Feasibility

The system aligns with the operational needs of Sitara Hotel, improving booking efficiency, customer feedback management, and staff coordination. It enhances operational processes with real-time room availability, automated booking, and centralized dashboards for staff. The system is designed for ease of use, ensuring minimal disruption to current operations.

Resource Feasibility

The resources required for the project are readily available and manageable within the hotel's budget. Since the system uses open-source tools, there is no significant cost associated with acquiring proprietary software. Additionally, the hotel's existing IT infrastructure supports the system's requirements. The availability of both human and technical resources ensures the project's feasibility from a resource perspective.

Economic Feasibility

The system is designed to be cost-effective, utilizing open-source tools to minimize development, deployment, and maintenance costs. By reducing reliance on third-party platforms, the hotel saves on commissions, leading to long-term cost savings.

1.7 Structure of the Dissertation

The dissertation is organized into six chapters, each providing a detailed aspect of the project.

Chapter 01: Introduction

This chapter provides an overview of the project, introducing the scope, objectives, Feasibility study and the proposed approach or study. It aims to capture the reader's interest and set the context for the work undertaken.

Chapter 02: Background

This chapter provides the background of the project is explained in detail. It covers existing work related to the topic and includes a critical review of previous research and findings.

Chapter 03: Design

This chapter provides the design of the system, illustrated using UML diagrams. It describes the overall system architecture and the design of various modules within the system.

Chapter 04: Implementation

The implementation chapter describes the techniques and tools used, including methodologies, useful tools, platform dependencies, and the chosen environment for the implementation.

Chapter 05: Testing and Evaluation

This chapter outlines the testing process, system evaluation, and the implementation of the solution. It highlights how the system was assessed for performance and functionality.

Chapter 06: Conclusion

The final chapter summarizes the key components of the project, offering insights into the outcomes and suggesting possible future work or improvements for the system.

1.8 Summary

The first chapter introduces the project to develop a Hotel Reservation and Customer Feedback Management System for Sitara Hotel Weligama. The system aims to enhance operational efficiency by automating bookings, managing customer feedback, and streamlining staff processes. Key features include real-time room availability, a review summarization tool, promotional offers, and a centralized platform for staff operations. The project is motivated by the need to reduce third-party commissions and operational inefficiencies. The feasibility study highlights technical, operational, resource, and economic considerations, confirming the project's viability.

Chapter 2 – Background

2.1 Introduction

Sitara Hotel Weligama currently depends on external third-party platforms for room reservations and feedback collection. These platforms impose substantial commission fees, thereby diminishing the hotel's profitability and restricting direct interaction with guests. Additionally, the manual management of reservations, guest preferences, and feedback increases the risk of errors, data loss, and delays in addressing customer inquiries or complaints. The front desk staff, marketing staff, and cleaning staff currently perform their day-to-day functions through manual processes.

In Sri Lanka's competitive hospitality industry, ensuring real-time booking availability and maintaining high customer satisfaction is essential. The existing system fails to provide an efficient and timely way to track reservations, manage guest feedback, and offer promotional services. Guests and staff are often required to use multiple platforms for reservations, service bookings, and feedback, leading to fragmented communication and delayed responses.

To address these issues, the proposed system automates the hotel's reservation and feedback management processes. The system will allow guests to view room availability in real-time, book rooms, and pay online, eliminating the need for external platforms and reducing commission fees. It will also collect and analyze customer feedback, providing the hotel with actionable insights. Additionally, the system will generate personalized offers based on guest history, fostering customer loyalty and encouraging repeat bookings. The manual system will enable staff to manage bookings, handle complaints, and provide seamless customer experience, enhancing operational efficiency and guest satisfaction.

The high-level case diagram for the current manual system is shown in Figure 2-1.

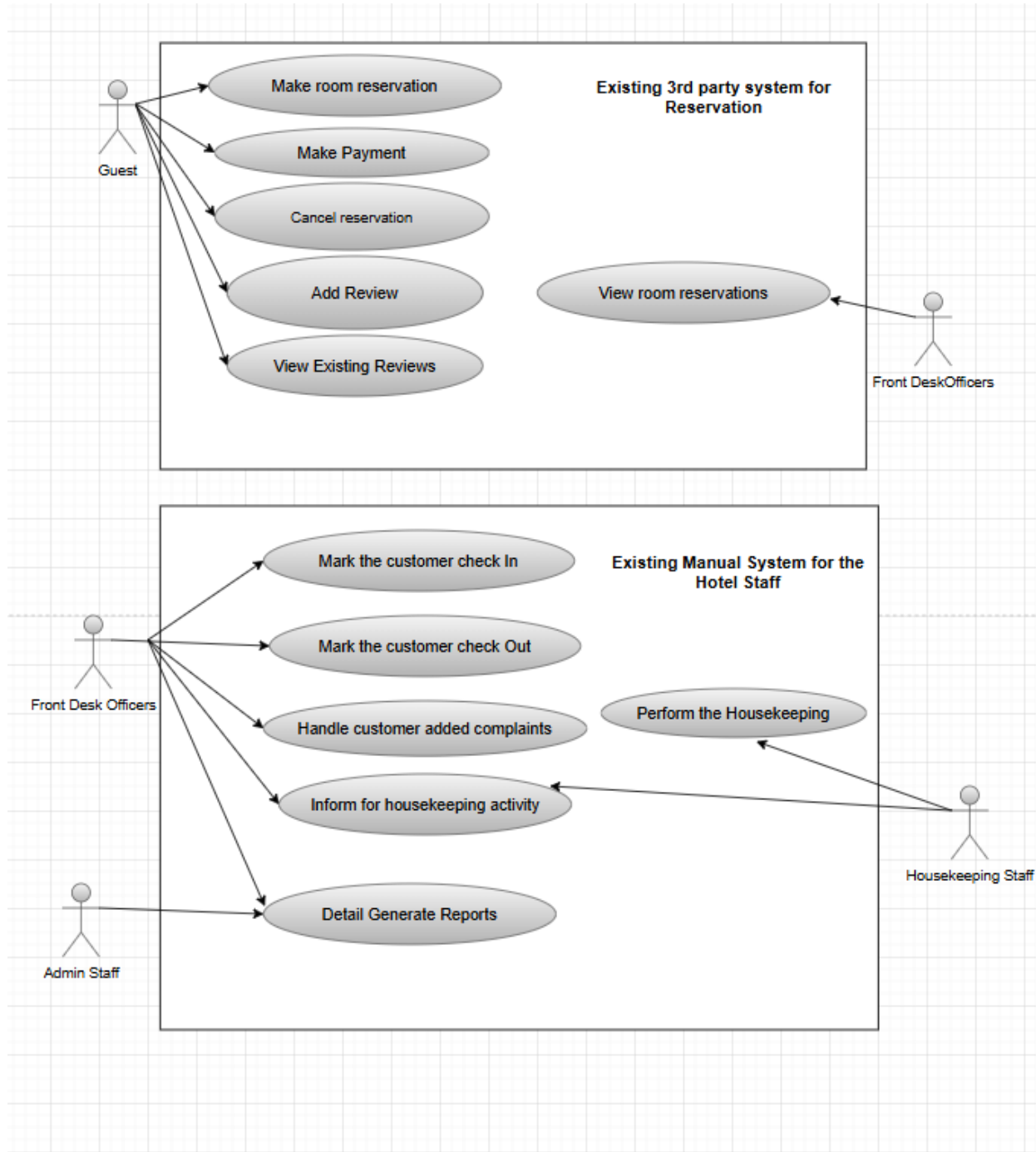


Figure2 1:Current system in Sitara hotel

2.2 Data Collection Procedures

Data collection is a critical step in the development of the Hotel Reservation & Customer Feedback Management System. The following steps outline the process for gathering requirements:

Identify Stakeholders

The first step was to identify the key stakeholders involved in the system, including hotel management (Admin), front desk staff, housekeeping staff, marketing staff, and guests. Each stakeholder's role in the system was analyzed to understand their specific needs and responsibilities.

Conduct Stakeholder Interviews

Interviews were conducted with each stakeholder group to gather detailed insights into their requirements and expectations. Open-ended questions were used to identify pain points in the current manual and third party-dependent processes, including reservations, feedback management, and service bookings. The information collected was documented accurately for further analysis.

Review Existing Documentation

Existing documentation, such as reservation logs, feedback forms, and payment records, was evaluated to identify inefficiencies and gaps in the current processes. The review provided a clearer understanding of how the new system could address challenges like delayed bookings, missed feedback, and errors in payment tracking.

Analyze Competitor Systems

Examined features from competitor hotel management systems, such as online booking platforms, feedback management tools, and loyalty program integrations. This analysis provided insights into industry best practices and features that could enhance the system's functionality.

Document Functional and Non-functional Requirements

The gathered data was consolidated into functional and non-functional requirements. Functional requirements include features like room availability display, real-time booking, secure online

payments, personalized offers, feedback collection, and review summarization. Non-functional requirements were documented to ensure security, usability, scalability, and performance criteria were met. These requirements were prioritized based on their importance and feasibility for implementation.

2.3 Requirement Analysis

The proposed system addresses various scopes based on the identified requirements and system features. The system users are classified as administrators, front desk officers, housekeeping staff, marketing staff, and guests. Each user group has specific functionalities tailored to their needs, ensuring seamless and efficient experience for both guests and hotel staff.

The administrator's scope includes managing user roles, updating room details, handling payment records, and maintaining reservation and feedback data. Administrators can monitor system activities, generate reports on reservations and feedback, and manage promotional campaigns to enhance customer retention. They also oversee the management of loyalty programs and personalized offers based on customer history.

Front desk officers are responsible for handling guest check-ins and check-outs, managing room availability, and making reservations on behalf of guests. They can access real-time data on room status and additional service bookings, such as cab services or meal orders, ensuring efficient guest service.

Housekeeping staff can receive updates on room status, cleaning schedules, and special guest requests, facilitating better coordination and timely service delivery. Similarly, marketing staff can analyze customer feedback and reviews to create targeted promotions and enhance the hotel's reputation.

Guests, as the end users, can use the system to view real-time room availability, make bookings, and securely process payments online. The system also allows guests to submit feedback, complaints, and reviews, which are summarized using machine learning algorithms to provide actionable insights to the hotel management.

The system integrates Python-based machine learning models to summarize customer reviews, offering flexibility and scalability. This feature enables guests to make informed decisions while assisting hotel staff in identifying trends and areas for improvement. The inclusion of secure online payments and personalized promotions ensures a user-friendly experience while enhancing operational efficiency.

2.3.1 User Classes and Characteristics

Admin

The admin holds the highest privileges within the system. They can create, view, modify, and delete user accounts, including staff and guest profiles. Admins can update room details, manage payment records. Additionally, they are responsible for generating reports, monitoring system usage, and handling escalated complaints or feedback.

Front Desk Officers

Front desk officers manage guest check-ins and check-outs, handle room reservations, and process additional service bookings such as cab services and meal orders. They have access to real-time room availability data and payment statuses. Front desk officers can also update reservation details on behalf of guests and assist with inquiries to ensure a seamless guest experience.

Housekeeping Staff

Housekeeping staff are notified about room cleaning schedules, special requests, and room availability status. They can update the system once tasks are completed, allowing for real-time coordination with front desk officers and ensuring rooms are prepared promptly for new guests.

Marketing Staff

Marketing staff use the system to analyze customer feedback, ratings, and summarized reviews. They can create and manage loyalty programs and targeted promotional offers based on guest preferences and booking history. Marketing staff are also responsible for monitoring customer trends and enhancing guest engagement through personalized promotions.

Guests

Guests have access to features like real-time room availability, secure booking and payment options, and service requests for amenities like cab services or meals. They can submit feedback,

reviews, and complaints directly through the system. Guests also benefit from personalized offers, loyalty rewards, and machine learning-powered review summaries to make informed booking decisions.

2.3.2 System Requirements

Hardware Requirements

Core i7 processor 10th generation or above	To do the complex processes in machine learning.
16 GB RAM or above	To control high volume of dataset and development.
Router or Internet Device	For Network
Disk space of 40GB or above	To store code bases, documents, and data.

Table 2 1: Hardware requirements

Software Requirement

Operating System (MacOS, Linux, Windows 7)	Windows 10 will be used
Visual Studio Code	Environment for Development of machine learning module
Bitbucket/GitHub	To backup code files.
Google Colab / Jupyter notebook	Initially the author will use google colab in order to train the model.
MS Office/ Google Docs/ Canva/ Figma	To create project documents, reports and diagrams related to the project.
Mendeley / Zotero	Mendeley will be used to save and backup. previous research papers and to manage.

	references.
HTML, CSS, JavaScript, jQuery, MySQL, Laravel, PHP	Environment for Development of proposed solution

Table 2: Software requirements

2.3.3 Design Implementation Constraints

Language Support

The system's graphical user interface (GUI) will only be available in English to ensure consistency and ease of use for all stakeholders.

Internet Dependency

The system requires a stable internet connection to function fully, as it relies on real-time data updates for room availability, payment processing, feedback submission, and notifications.

User Proficiency

The system assumes basic proficiency in using web applications among both guests and staff, with minimal training provided for internal users.

2.3.4 Functional Requirements

1. Room Reservations and Payment Handling

- Customers can make reservations by checking the availability of rooms.
- The system displays available room types and prices (e.g., Superior, Deluxe, Suite) along with detailed descriptions and recent photographs.
- Based on the customer selection of room types and additional services (e.g., Cab service) the system automatically calculates the total payment amount.

- Customers who complete a reservation successfully receive an email notification containing a reference number to confirm their booking details.
- The system also offers payment choices, permitting guests to settle the payment online or either during the check-in time as a card or cash.
- Users can cancel reservations and are eligible for a refund upon cancellation.

2. Reservations for the additional services (Cab service, Refreshment, meal and Spa)

- The system presents the available cab services for the tour, allowing customers to select a preferred location. Rates are adjusted according to the distance, after which the customer can proceed with the payment.
- The system displays an menu options, enabling customers to select meals. Based on the chosen items, the system calculates the total cost, allowing the customer to proceed with payment.
- The system displays spa services, enabling customers to select different spa services. Based on the chosen services, the system calculates the total cost, allowing the customer to proceed with payment.

3. Customer review management and review summarization using machine learning algorithms

- Customers who successfully log in can create a review by selecting a rating.
- Summarize extensive reviews submitted by current customers using the machine learning algorithms.

4. Customer Complaint Handling & Feedback management

- Implement a feature within the system for managing customer complaints, allowing customers to submit complaints along with necessary details directly through the system.
- Customers have the option to submit feedback via a form during the checkout process.

5. Check-In and check-out process Management.

- The reservation status is monitored in real time, requiring office users to separately record customer check-in and check-out details. This functionality enables accurate tracking of

reservation activities, ensuring up-to-date status information. By recording these details individually, the system maintains clear and reliable records of guest stays.

6. Functionality to arrange exclusive promotions

- Office users can record supplementary notes for special customer reservations. These notes may include specific requests from customers, such as room preferences or dietary needs, allowing for tailored experiences.
- Office users can easily identify eligible customers based on their booking history program status, allowing for targeted promotions. Communication of these special offers can occur through email notifications or during the booking process, ensuring customers are aware of available incentives.

7. Notifications

- The system will send confirmations and reminders to guests via email

8. Reporting Module

- The Report Module in the Hotel Reservation & Customer Feedback Management System provides comprehensive analytics for data-driven decision-making. It generates reports on customer feedback, operational metrics, reservation history allowing management to analyze key performance indicators (KPIs) and gain insights into customer satisfaction and service quality. This centralized data access enhances operational transparency, supports service improvements, and helps identify growth opportunities.

9. Dashboard Graphs

- A graphical dashboard will provide administrators and staff with an overview of system activities, including reservations, feedback trends, and service usage.
- The dashboard will display key performance indicators (KPIs) for decision-making.

10. User Role Management

- The admin is authorized to create office users for the portal.
- Access to the portal is limited to registered office users. Unregistered users are not allowed to log in.

- The admin user has the capability to manage room information within the system effectively. This includes adding new room details and updating existing information, such as room features and amenities, ensuring accurate and up-to-date data. This functionality enhances the system's flexibility, allowing for efficient adjustments to room offerings as needed.

11. House Keeping Ticket Management

The housekeeping ticket management system enhances efficiency by streamlining task allocation and tracking. It enables front desk staff to create and assign tasks to the housekeeping staff while categorizing them based on priority.

2.3.5 Non-Functional Requirements

Using the suggested system, it is concerned about the following nonfunctional requirements.

1. Security

The security requirement focus is to protect the suggested system against security threats, unauthorized access and data breaches. When considering confidentiality, it ensures sensitive data can access only the authorized users. According to the suggested system, Admin portal can only be accessed only for the registered office users. This can prevent unauthorized access from the outside users, so it secures the customer's sensitive information.

2. Usability

This requirement ensures the ease of learning, efficiently satisfying for the users. The interface design is simple with clear navigations. So, customers may easily make hotel reservations. Need minimal training required for the Agent users to perform the agent task. When considering efficiency, it measures the speed at which users can accomplish their task. It involves the minimizing number of steps to complete the task to reduce the human errors and optimizing responses times. It involves providing informative notifications to help the users to understand what went wrong and how to correct it

3. Reliability

Reliability ensures the user satisfaction and trust to the system. The system should be available for the users at any time. Need less downtime for the maintenance. Reliability testing is essential to ensure the reliability requirements. This includes the failure testing to understand how the system behaves under different conditions. Suggested system implemented according to the better design, testing and maintenance to maintain the desired level of reliability.

4. Performance

Need to measure the response time, resource allocation, scalability, throughput and reliability under load. Response to time means how it takes for the system to respond to the user request. This system is designed to manage a lot of user tasks quickly by avoiding delays or slow response time. Throughput measures the number of operations that the system can handle the expected workload. Performance is much more important to build a satisfactory user experience.

5. Maintainability

Maintainability ensures the ease with which suggested software can be modified, enhanced and maintained over a lifecycle. Modularity, understandability, simplicity, and testability attributes collectively contribute to the ease of maintaining the software system. The system should be well documented for maintenance and future enhancements. Codebase follows the best practices to ensure the ease of the future modification for developers.

2.4 Review of Similar System

Web-based hotel reservation systems have become an integral part of the hospitality industry, offering significant advantages over traditional reservation methods. These systems enable hotels to manage bookings more efficiently, reduce administrative overhead, and provide better customer experience. Research and development in this area have focused on various aspects:

User Experience: Research has demonstrated that user-friendly interfaces significantly impact customer satisfaction and booking conversion rates. Effective web-based systems incorporate intuitive design elements and seamless navigation to improve the user experience (Liu & Chen, 2009).

Booking.com

Booking.com is a leading online travel agency that allows users to search and book hotels, vacation rentals, flights, and car rentals worldwide. It offers a diverse range of accommodation and allows users to read reviews from previous customers, which helps in making well-informed decisions. The platform is known for its competitive pricing, easy-to-use interface, and flexibility in booking and canceling reservations (Booking.com, n.d.).



Figure 2: Booking.com

TripAdvisor

TripAdvisor is largely a review website, but it also lets customers make direct hotel reservations on its platform. Because it has a wealth of user evaluations and ratings, it is useful for assisting travelers in making judgments (TripAdvisor, n.d.).

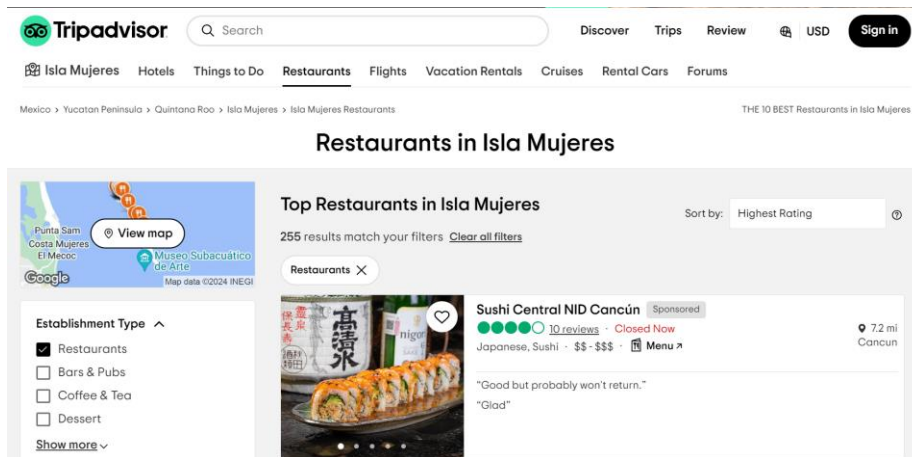


Figure2 3:Tripadvisor

Hotels.com

Hotels.com, which is owned by Expedia Group, specializes only in hotel reservations. It provides frequent discounts and bargains, a rewards program where you can earn free nights, and an easy-to-use interface (Hotels.com, n.d).

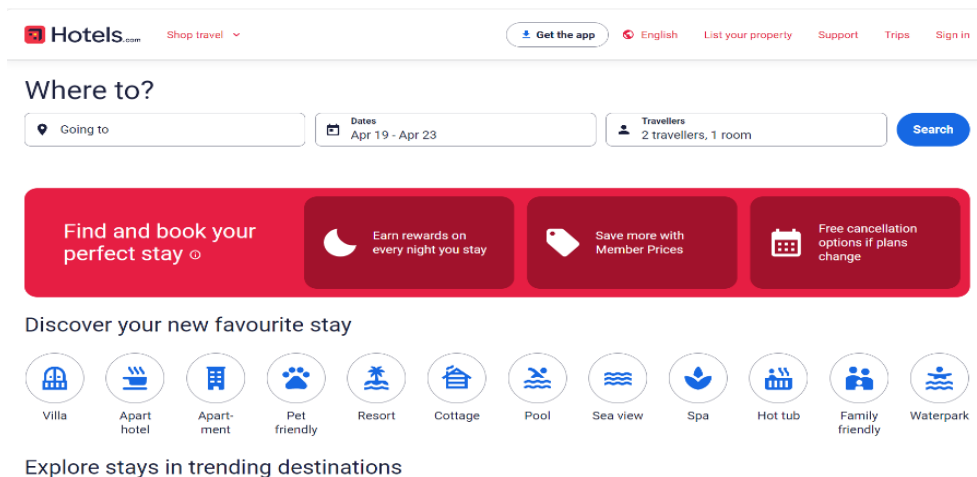


Figure2 4:Hotels.com

An Anantaya official web site

Anantaya Resort and Spa Chilaw or Anantaya Resort and Spa Pasikuda treated to great accommodation, fine dining, picturesque views, so they have a web application to make the reservation for customers and give the overview about in the hotel details about meetings, about us, Dinning, Gallery, Spa (Anantaya, n.d.).

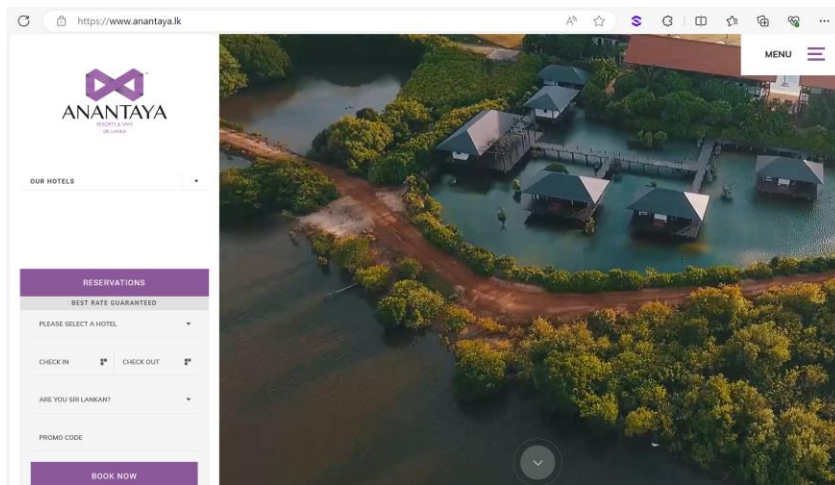


Figure2 5:Anantaya.com

Cloudbeds

A cloud-based property management tool called Cloudbeds was created to assist hotels save time while boosting income and guest pleasure. With its user-friendly interface, users may manage the entire property. It offers a powerful channel manager for easy booking management. It provides a price intelligence engine to automate techniques for managing rates across various channels, resulting in time savings and increased revenue. Its dashboard helps maximize occupancy and income utilizing real-time data and offers an overview of assets across many locations. (Cloudbeds, n.d.)

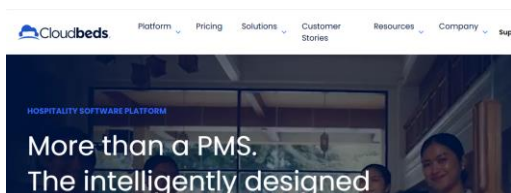


Figure2 6:cloudbeds

Related Similar work Comparison

When reviewing related works, no hotel reservation system was found that offers a feature to summarize lengthy reviews. Such a feature would simplify decision-making for customers while enabling hotel management to forecast trends and make informed decisions based on customer feedback.

Features	Similar Systems				Proposed System
	Booking.com	Hotels.com	Trip Advisor	Anantaya website	
Real-Time Room Availability	✓	✓	✓	✓	✓
Make a Reservation with attractive user Interfaces	✓	✓	✓	✓	✓
Add / view User Reviews	✓	✓	✓	✓	✓
Automatically summarizing the lengthy Reviews	✗	✗	✗	✗	✓
Promotion programs and rewards	✓	✓	✓	✗	✓
Online Payments	✓	✓	✓	✓	✓
Integration with Local Attractions & Services	✓	✓	✓	✗	✓
Email Notifications	✓	✓	✓	✓	✓
Detail Reports	✗	✗	✗	✗	✓
MIS Dashboard	✗	✗	✗	✗	✓

Table 2 3: Related similar work comparison

2.5 Related Technology

Web Application Front-End

HTML

HTML (Hypertext Markup Language) is used to structure the content and layout of web pages. It defines elements such as headings, tables, forms, and buttons, providing the foundation for user interface development. HTML integrates seamlessly with dynamic content rendering tools, ensuring a consistent structure and compatibility across browsers and devices.

CSS

CSS (Cascading Style Sheets) is used for styling and designing the system's interface. It enables clear separation of structure (HTML) from presentation, ensuring a cohesive and professional look throughout the application. CSS facilitates responsive designs, making the system accessible and visually appealing on desktops, tablets, and smartphones.

JavaScript

JavaScript enables dynamic and interactive features on the front end. It supports real-time content updates, form validation, and asynchronous communication with the server using AJAX. JavaScript enhances user experience by providing a seamless flow of actions, such as displaying real-time room availability and feedback submission without requiring page reloads.

Bootstrap

Bootstrap is a front-end framework that accelerates the development process with pre-designed components like buttons, forms, and navigation bars. Its responsive design features ensure that the system adapts to various screen sizes, providing a consistent experience across devices. The framework's customization options allow the system to align with the hotel's branding.

Back-End Development

Laravel Framework

Laravel, a PHP-based framework, is used for server-side development. It simplifies back-end operations such as handling user authentication, managing bookings, and processing payments. Laravel provides a modular architecture, robust security features, and scalability, ensuring the system can handle growing user demands.

MySQL

MySQL, a widely used relational database management system, serves as the back-end database. It efficiently stores and manages data for reservations, feedback, and payments. MySQL's indexing and query optimization features ensure quick access to data, even during peak usage periods.

Python and Machine Learning Libraries

Python-based machine learning libraries, such as TensorFlow or Scikit-learn, are employed for feedback analysis. These libraries enable the system to summarize lengthy customer reviews, offering actionable insights for hotel management and simplifying decision-making for guests.

User Interface and Design Tools

Figma

Figma is used for creating user interface prototypes and wireframes during the design phase. It allows collaborative design iterations, ensuring user-friendly interfaces aligned with the system's requirements and branding.

Additional Technologies

Notifications via Email

Email services are integrated for sending booking confirmations, reminders, and feedback requests. These notifications enhance communication between the hotel and its guests, ensuring timely updates.

2.6 Summary

This chapter outlines the current processes involved in the Hotel Reservation & Customer Feedback Management System at Sitara Hotel Weligama. It details the functional and non-functional requirements of the proposed system while highlighting the technologies chosen for implementation. Additionally, comparison with other existing systems emphasizes how this solution addresses key inefficiencies and introduces unique features, such as machine learning-based feedback summarization and real-time reservation management.

Chapter 3 -Design

3.1 Introduction

This chapter aims to present the design architecture of the proposed Hotel Reservation and Customer Feedback Management System. It provides a comprehensive overview of the system's analysis and design, serving as the foundation for transforming requirements into a functional solution. Requirements gathering and analysis are critical steps in ensuring the project's success, as they establish a clear understanding of the system's needs and scope. Key activities in this phase include understanding the domain, collecting and organizing requirements, classifying and structuring them, prioritizing their importance, and validating them against project goals.

The design phase of the software development lifecycle is a crucial step where the insights gained during the analysis are methodically shaped into a detailed framework for implementation. According to Sommerville (2011), "Software design is a creative activity in which you identify software components and their relationships, based on a customer's requirements." This phase serves as a bridge between high-level requirements and technical implementation, ensuring that all aspects of the system are clearly defined and aligned with the project's objectives.

This chapter includes the system architecture and its associated UML diagrams, such as the use case diagram, class diagram and activity diagram. These diagrams offer insights into the key structural components and their interrelationships, providing a holistic view of the system's design. The resulting system specification serves as a detailed roadmap for the implementation phase, ensuring clarity and precision in developing the Hotel Reservation and Customer Feedback Management System.

3.2 Related System Design Strategy

The design strategy outlines the approach adopted to create and structure the proposed system effectively. Various methodologies, such as prototyping, agile development, and modern structured design, are commonly used for system modeling. For the Hotel Reservation and

Customer Feedback Management System, the object-oriented design methodology was selected due to its flexibility, modular structure, and the ability to promote code reusability and easier system maintenance. This approach enables the integration of additional functionalities in the future and ensures the system remains scalable and manageable.

A key element of object-oriented design is the use of the Unified Modeling Language (UML), which provides a standardized visual approach for representing system components and their interactions. As Booch et al. (2005) state, "UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems." UML diagrams play a crucial role in simplifying complex object models, enabling developers to understand and communicate the system's architecture effectively. This visualization facilitates collaboration among stakeholders and ensures a streamlined development process.

The subsequent sections include detailed structural and behavioral UML diagrams, such as use case diagram, class diagram, and activity diagrams. These diagrams collectively depict the architecture and interactions within the Hotel Reservation and Customer Feedback Management System, providing a clear blueprint for its development and implementation.

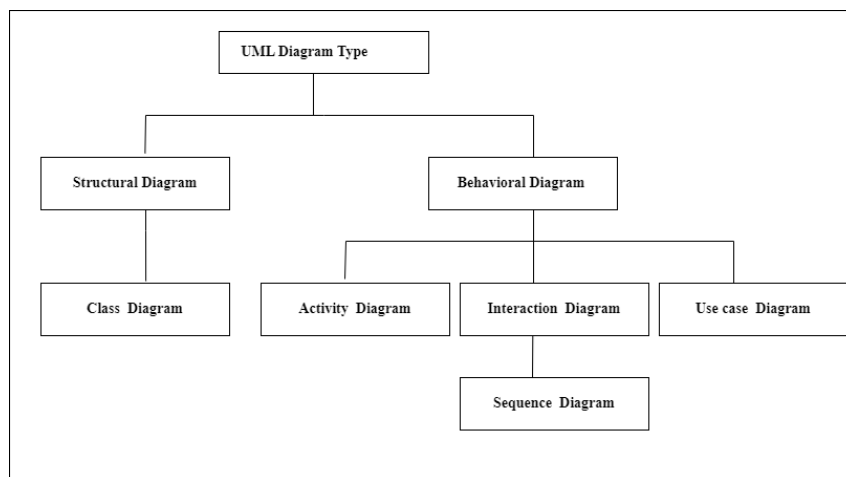


Figure3 1-UML Diagram

3.3 System Architecture

The architecture of the Hotel Reservation and Customer Feedback Management System leverages the Model-View-Controller (MVC) design pattern to ensure a structured and scalable approach to system development. The MVC architecture separates the system into three interconnected components: Model, View, and Controller. This division enhances modularity and maintainability while facilitating efficient communication between the client side and server side of the application. The proposed system architecture is depicted in Figure 3-2: System Architecture below.

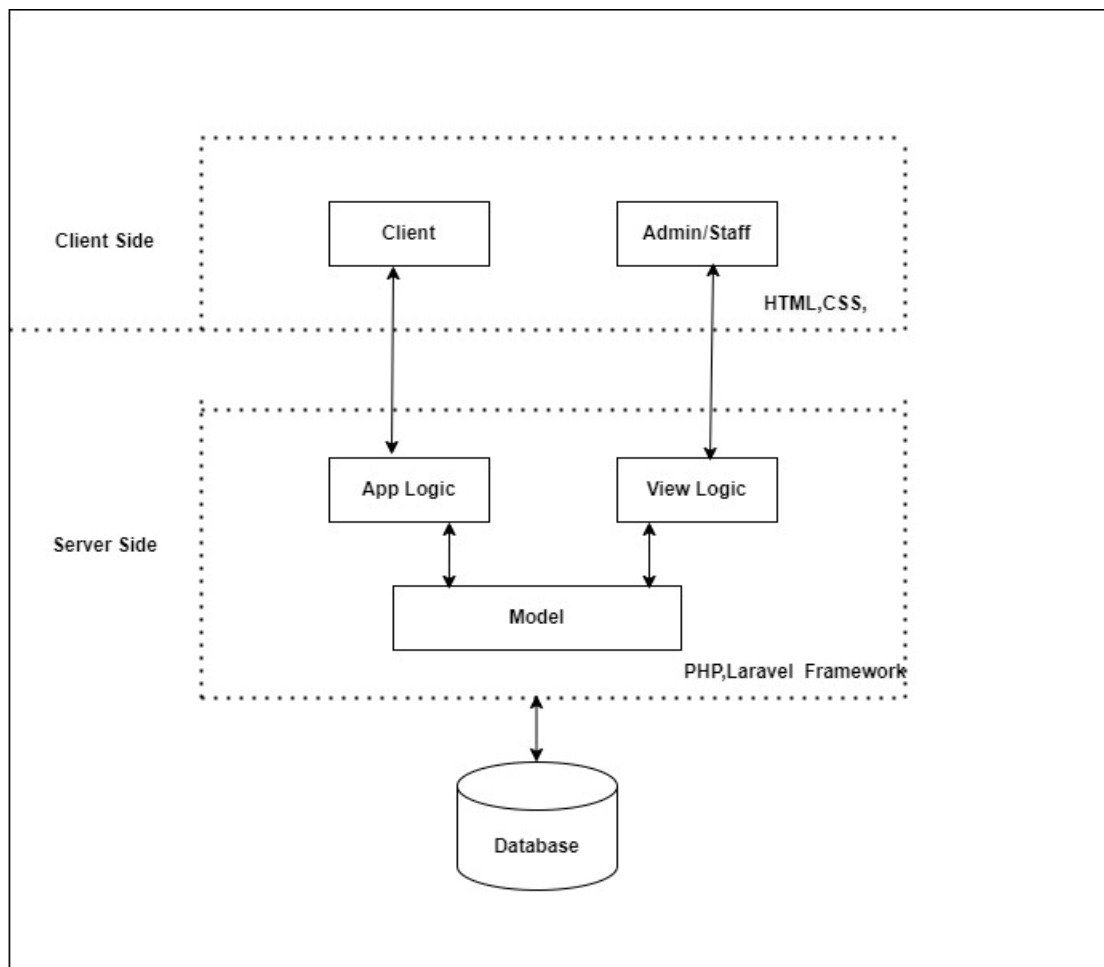


Figure3 2-System Architecture Diagram

Components of the MVC Architecture

Model

The model serves as the intermediary between the application and the database. It represents the logical data structure of the system and is responsible for managing data retrieval and updates. For the Hotel Reservation and Customer Feedback Management System, the model handles the database operations related to reservations, user feedback, customer information, and service availability. The database is implemented using relational database systems to store and manage structured data efficiently.

View Logic

The view component handles the user interface and presentation layer. It displays dynamic data retrieved from the model and ensures a user-friendly experience for both customers and administrative staff. For this system, the user interface is built using technologies like HTML and CSS, providing responsive and visually appealing designs tailored for clients and staff.

App Logic

The controller acts as the central component that manages the interaction between the model and the view. It processes input from the user, interacts with the model to retrieve or update data, and returns the appropriate response to the view. This layer ensures the seamless execution of business logic, such as room availability checks, reservation confirmations, and feedback processing.

Database

The database stores all critical information, including room details, reservation data, user feedback, and administrative records. It is an essential part of the system's backend and ensures data consistency, integrity, and security.

Security

The system incorporates robust security mechanisms to protect sensitive customer and operational data. This includes user authentication, role-based authorization, and encryption of data both at rest and in transit. These measures ensure secure access to the system by authorized users only.

Integration

To enhance operational efficiency, the system integrates with third-party services like payment gateways, email notifications, and analytics tools. This integration supports features such as online payments, reservation confirmations, and feedback reporting.

Reporting

The reporting component generates various reports for managerial and operational use, such as occupancy reports, revenue analysis, and customer feedback summaries. These reports aid in data-driven decision-making and system optimization.

Communication Mechanism

The system uses HTTP protocols to handle client-server communication. The client sends requests to the server, where the application logic processes them and interacts with the model as needed. The server then sends the appropriate response back to the client, ensuring a seamless user experience.

By leveraging the MVC architecture, this system provides a modular, scalable, and secure framework to manage hotel reservations and customer feedback effectively.

3.4 UML Diagrams for Proposed System

3.4.1 Use case Diagram

The use case model for the Hotel Reservation and Customer Feedback Management System represents the interaction between key actors and the system. The system's functionality is defined by various use cases that detail all possible interactions. The actors represent the roles users perform within the system to accomplish their tasks.

Actor	Description
Guest	The user who interacts with the system to search for rooms, make reservations, submit feedback, and view personalized offers, Review Summarizations
Front Desk Officer	The staff member responsible for managing reservations, reviewing customer feedback, handling approvals, and generating customer reports
Housekeeping Staff	The staff member who receives task notifications related to room preparation, cleaning schedules, and feedback-related requests.
Marketing Staff	The staff responsible for managing promotional offers, analyzing feedback, and reviewing customer reviews to improve the hotel's marketing strategies.
Admin Staff	Generated reports, makes data-driven decisions based on reservation and feedback insights, User Role management, Room management

Table 3-1: Actors in Use Case Diagram

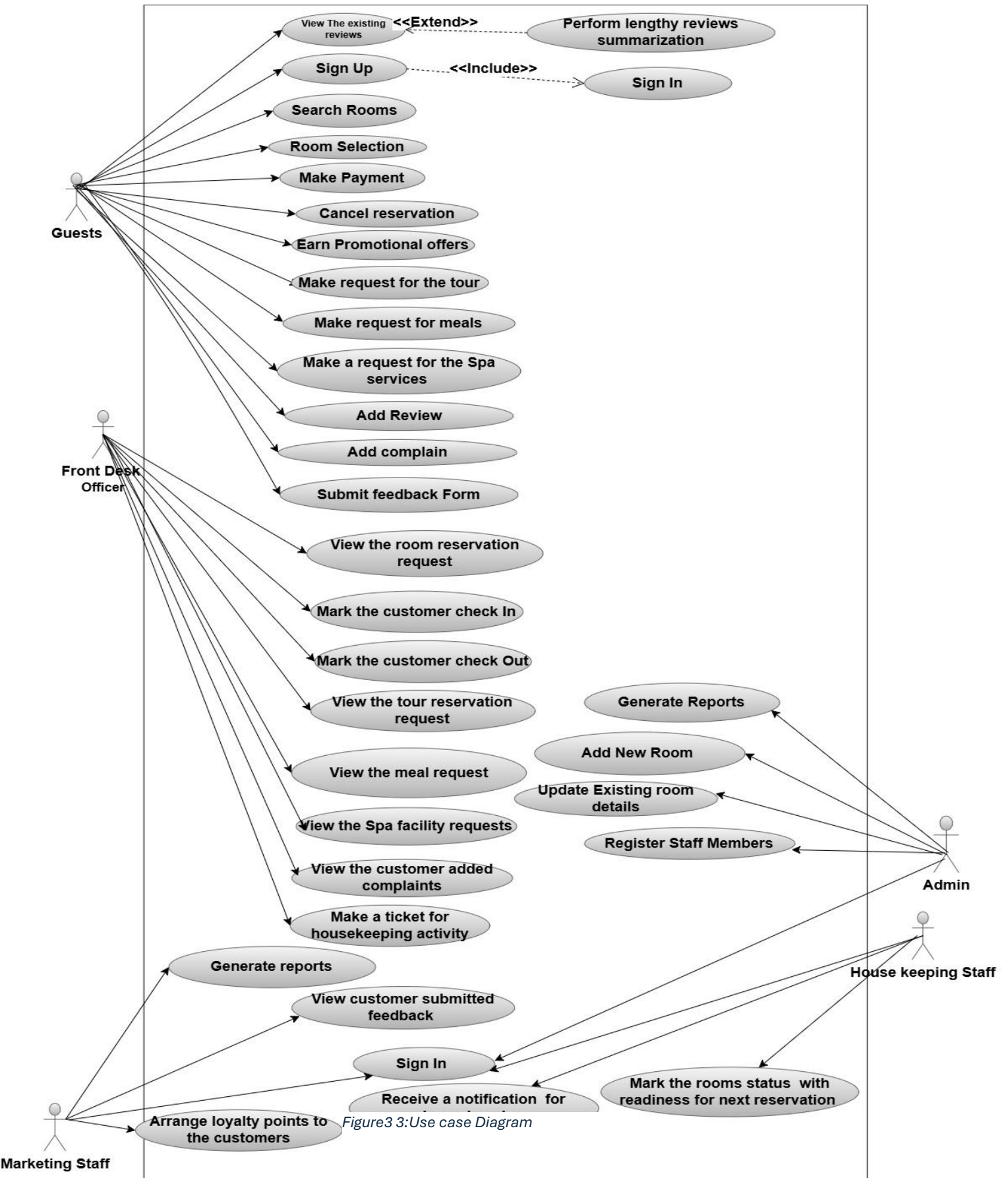


Figure3 3:Use case Diagram

3.4.2. Use Case Narratives

Use-case Number	UCE-01	
Use-Case Name	Create Employees	
Actor/s	Admin	
Description	This use case describes how an admin creates the employee account	
Precondition	The admin user should already login to the portal	
Post-condition	The created account can log in using their credentials.	
Basic course of Action	User Action	System Response
	1.Admin click on “Create Employee” section 2.Fill the Employee form and click on submit button • Name • Mobile • Email • Employee Type (Admin, Front Desk Staff, Marketing Staff, Housekeeping Staff) • Password	1. The system displays the "Create Employee" form 2. As the admin enters details, the system validates input fields in real-time (e.g., proper email format, no special characters in the mobile number). The system ensures that the "Employee Type" field is selected before proceeding 3.Display the success popup/ Error popup
Alternate course of Action	If the Admin attempts to add a user who is already registered, the system displays a validation message and prevents further action.	

Table3 1:Use case narratives for admin user create employee

Use-case Number	UCE-02	
Use-Case Name	Employee Login	
Actor/s	Admin, Front Desk, Marketing Staff	
Description	This use case describes how an employee logs into the Hotel Reservation & Customer Feedback Management System.	
Precondition	The Admin user created the user accounts	
Post-condition	The created account can log in using their credentials.	
Basic course of Action	User Action	System Response
	1. The employee accesses the login page. 2. The employee enters their email address and password. 3. The employee clicks on the "Login" button.	1. The system displays the login form with fields for Email and Password. 2. The system validates the credentials 3. If valid, the system logs the user
Alternate course of Action	If the email or password is incorrect, it display the an error message: <i>"Invalid email or password. Please try again."</i>	

Table3 2:Use case narratives for the employee login

Use-case Number	UCE-03	
Use-Case Name	Add Room.	
Actor/s	Admin	
Description	This use case describes how an Admin can add a new room	
Precondition	The Admin must be logged into the system and have access to the "Manage Rooms" section	
Post-condition	The new room is successfully added to the hotel	
Basic course of Action	User Action	System Response
	1. The admin navigates to the "Manage Rooms" section from the dashboard. 2. The admin clicks on the "Add Room" button. • Room Number • Room Type (e.g., Single, Double, Suite) • Price Per Night • Room Description • Status Availability 3. The admin fills in the required details and clicks "Submit."	1. The system displays an "Add Room" button. 2. The system displays a form to add a new room Form 3. If valid, the system saves the room details
Alternate course of Action	If any required field is left blank. - The system highlights the missing fields and displays an error message: <i>"Please fill in all required"</i>	

Table3 3:Use case narratives for the admin user add rooms

Use-case Number	UC-04	
Use-Case Name	Search for Available Rooms	
Actor/s	Guest	
Description	This use case describes how guests can search for available rooms by selecting check-in and check-out dates.	
Precondition	None	
Post-condition	The guest views the available rooms.	
Basic course of Action	User Action	System Response
	1.The guest selects the check-in and check-out dates 2.Click on the search button	1. The system displays the calendar to select the check-in and check-out dates 2. The system needs to display the available rooms 3. Use case Exit
Alternate course of Action	It needs to display a message as “please select the valid check in & check out date range” once user click on the search button without selecting the date range/ if the user selects the invalid date range. Previous dates need to be disabled	

Table3 4:Use case narrative for search room availability

Use-case Number	UC-05	
Use-Case Name	Room Selection	
Actor/s	Guest	
Description	This use case describes how guests can select rooms based on availability, price, and preferences.	
Precondition	The guest must have entered valid check-in and check-out dates on the "Room Selection" page.	
Post-condition	The selected rooms and preferences are saved, and the total price is displayed.	
Basic course of Action	User Action	System Response
	1. The guest views the list of available room types. 2. The guest selects the number of rooms for a specific type. 3. The guest specifies the number of adults and children for each room type. 4. The guest clicks the "Next" button to proceed.	1. The system displays available room types(Deluxe, Superior, Suite) with images, descriptions, prices, and availability. 2. The system updates the total price 3. The system verifies that the selection is within allowed limits and updates the price accordingly. 4. The system saves the selections and navigates to the "Guest Information" page.
Alternate course of Action	The guest does not select a room or number of occupants.- The system displays an error message: "Please select at least one room and specify occupants.	

Table3 5:Use case narrative for room selection

Use-case Number	UC-06	
Use-Case Name	Guest Sign Up	
Actor/s	Guest	
Description	This case describes how guest can sign up for the portal.	
Precondition	Load the portal for Hotel Reservation & Customer Feedback management portal	
Post-condition	The user is registered in the system	
Basic course of Action	User Action	System Response
	1.The guest clicks on the 'Sign Up' button. 2.It should display the following fields in the sign-up form ● First Name ● Last name ● Date Of Birth ● Email ● City ● Country ● Mobile Number ● Password 2.The guest fills in the required details and submits the form.	1.The system displays the sign-up form. 2. The system verifies the information and creates the account. 6. Use case Exit

Alternate course of Action	If the information is invalid, the system displays an error message and asks the user to try again.
----------------------------	---

Table3 6:Use case narratives for the guest signup

Use-case Number	UC-07	
Use-Case Name	Guest Sign In	
Actor/s	Guest	
Description	This use case describes how user can sign in to the portal.	
Precondition	The user must have an existing account.	
Post-condition	The user is logged into the system.	
Basic course of Action	User Action	System Response
	1.The guest clicks on the 'Sign In' button 2. The guest enters their credentials. (Email and password) 3.Click on the Login button	1. The system displays the login form to enter Username and Password. 2. The system verifies that all the filled have been filled out and valid. 3. The system successfully logged in the system. 4. Use case Exit

Alternate course of Action	If all fields are not filled out and not matched to the username and password the system notifies the actor a message “Invalid Username or Password”.
----------------------------	---

Table3 7:Use case narratives for the guest sign in

Use-case Number	UC_09	
Use-Case Name	Make Payment	
Actor/s	Guest	
Description	This use case describes how a guest completes a payment for services or bookings.	
Precondition	The user has finalized the selection of services or bookings and proceeds to payment.	
Post-condition	The payment is successfully processed	
Basic course of Action	User Action	System Response
	1. The guest selects the "Make Payment" option. 2.The guest enters payment details (e.g., card information or online transfer details). 3. The guest confirms the payment.	1. The system displays the payment summary and options. 2. The system validates the payment details. 3. The system processes the payment and displays a success message along with a receipt.

Alternate course of Action	
----------------------------	--

Table3 8:Use case narratives for the make payment

Use-case Number	UC_10	
Use-Case Name	Submit Feedback.	
Actor/s	Guest	
Description	This use case describes how a guest submits feedback through a feedback form.	
Precondition	Customer need to check In to the hotel.	
Post-condition	The feedback is successfully submitted.	
Basic course of Action	User Action	System Response
	1. The guest navigates to the "Submit Feedback" section. 2. The guest fills out the feedback form and submits it.	1. The system displays feedback form. (Overall Experience, Room Cleanliness, Staff Friendliness, Service Quality, would you recommend our hotel to others, Suggestions) 2. The system processes the submission 3. Use case exit

Alternate course of Action	If any mandatory field is left blank, display: “Please fill in all required fields.”
----------------------------	--

Table 3 9: Use case narratives for submit feedback

Use-case Number	UC_11	
Use-Case Name	Add Complaints.	
Actor/s	Guest	
Description	This case describes how a guest adds a complaint about their stay.	
Precondition	Customer login to the portal	
Post-condition	The complaint is successfully submitted.	
Basic course of Action	User Action	System Response
	1. The guest navigates to the "Add Complaint" section. Types of Complain (Service, Cleanliness, Maintenance, Staff Behavior, Others) Date and Time for the incident, Attachments, Feedback Agreement	1. The system displays the complaint form. (Enter the complaint and submit button) 2. The system processes the submission 3. Use case exit

	2. The guest fills out the form and submits it.	
Alternate course of Action	If any mandatory field is left blank, display: “Please fill in all required fields.”	

Table3 10:Use case narrative for the add complaints

Use-case Number	UC_12	
Use-Case Name	Add Review	
Actor/s	Guest	
Description	This use case describes how a guest adds a review.	
Precondition	Load the official web site	
Post-condition	The review is successfully added.	
	User Action	System Response

Basic course of Action	1. The guest navigates to the "Review" section. 2. Click on "Write a Review" button 3. Enter the booking Number, PIN code 4. Click on the Proceed Button 5. Fill the details for the Name, Relevant Date, Bring Your Topic to the Sitara Family Table, Your Story with Sitara Family How would you rate your experience out of 10? 6. Click on the Submit button	1. The system displays review section 2. Display the "Write a Review button" 3. Display a Write a Review Popup with Proceed button 4. Display the Write a Review Detail popup with submit button. 5. "Review should be successfully saved to the system
------------------------	---	---

Table3 11: use case narratives for the add reviews

Use-case Number	UC_13
Use-Case Name	View Summarized Customer Reviews.
Actor/s	Guest
Description	This case describes how guests can view summarized existing customer lengthy reviews.
Precondition	The system must have customer existing reviews and guests navigate to the reviews section.
Post-condition	If the use case was successful, users can view the summarize reviews

Basic course of Action	User Action	System Response
	1.The Actor selects a specific review by click on it 2. The Actor click on the “Summarize” button.	1. The system displays the button for the review summarization 2. The system retrieves and displays the summarized reviews. 6. Use case Exit
Alternate course of Action		

Table3 12:Use case narratives for lengthy review summarization mechanism

Use-case Number	UC_15	
Use-Case Name	Mark Customer Check-in and check out	
Actor/s	Front Desk Officer	
Description	This use case describes how the officer marks a customer check-in. and check out	
Precondition	The officer must be signed in and have access to the reservation system	
Post-condition	The customer is marked as checked in.	
	User Action	System Response

Basic course of Action	1. The officer selects a reservation. 2. The officer clicks "Check-In" to mark the customer.	1. The system displays the reservation details. 2. The system confirms and updates the check-in status. 3. Use case Exists
Alternate course of Action	None	

Table3 13:Use case narratives for front desk officer mark the check in and check out

Use-case Number	UC 18	
Use-Case Name	View Customer Complaints	
Actor/s	Front Desk Officer	
Description	This use case describes how the officer views complaints	
Precondition	The officer must be signed in and have access to complaints.	
Post-condition	The officer views customer complaints.	
	User Action	System Response

Basic course of Action	1. The officer navigates to "Customer Complaints." 2. The officer view complaint details.	1. The system displays all customer complaints.
Alternate course of Action	None	

Table3 14:Use case narratives for viewing customer complaints

Use-case Number	UC_19	
Use-Case Name	Make Housekeeping Notifications	
Actor/s	Front Desk Officer	
Description	This use case describes how housekeeping staff can add notifications regarding room cleaning, or special housekeeping requests.	
Precondition	The housekeeping staff must be logged into the system and have the necessary permissions to add notifications.	
Post-condition	The housekeeping notification is successfully sent to the Housekeeping Staff.	
	User Action	System Response

Basic course of Action	1. The staff navigates to the " Add Housekeeping Notifications " section. 2. The staff enters the required details (e.g., room number, task type, priority). 4. The staff submits the notification.	1. The system displays a form to add housekeeping notifications. 2. The system validates the input fields and ensures all mandatory details are filled. 5. The notification is sent to the housekeeping staff
Alternate course of Action	None	

Table3 15:Use case narratives for housekeeping ticket management

3.4.3Activity Diagram

Activity diagrams are used to represent the activities that make up a system process and the flow of control between them. A filled circle symbolizes the start of a process, while a filled circle enclosed within another circle indicates its end. These diagrams offer a clear depiction of the sequence of activities and the interactions between different components within the system.

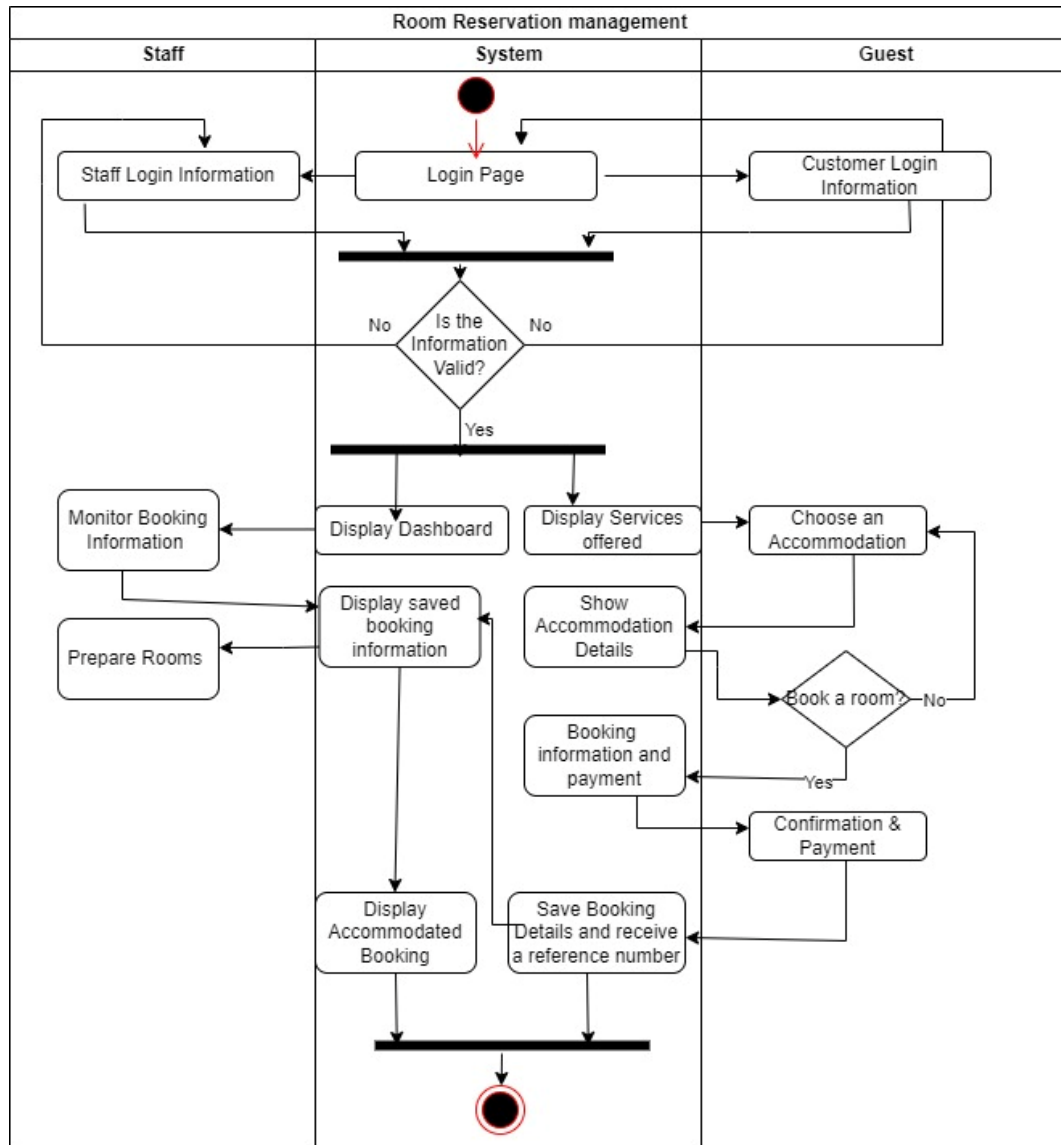


Figure3 4:Activity Diagram -Room reservation

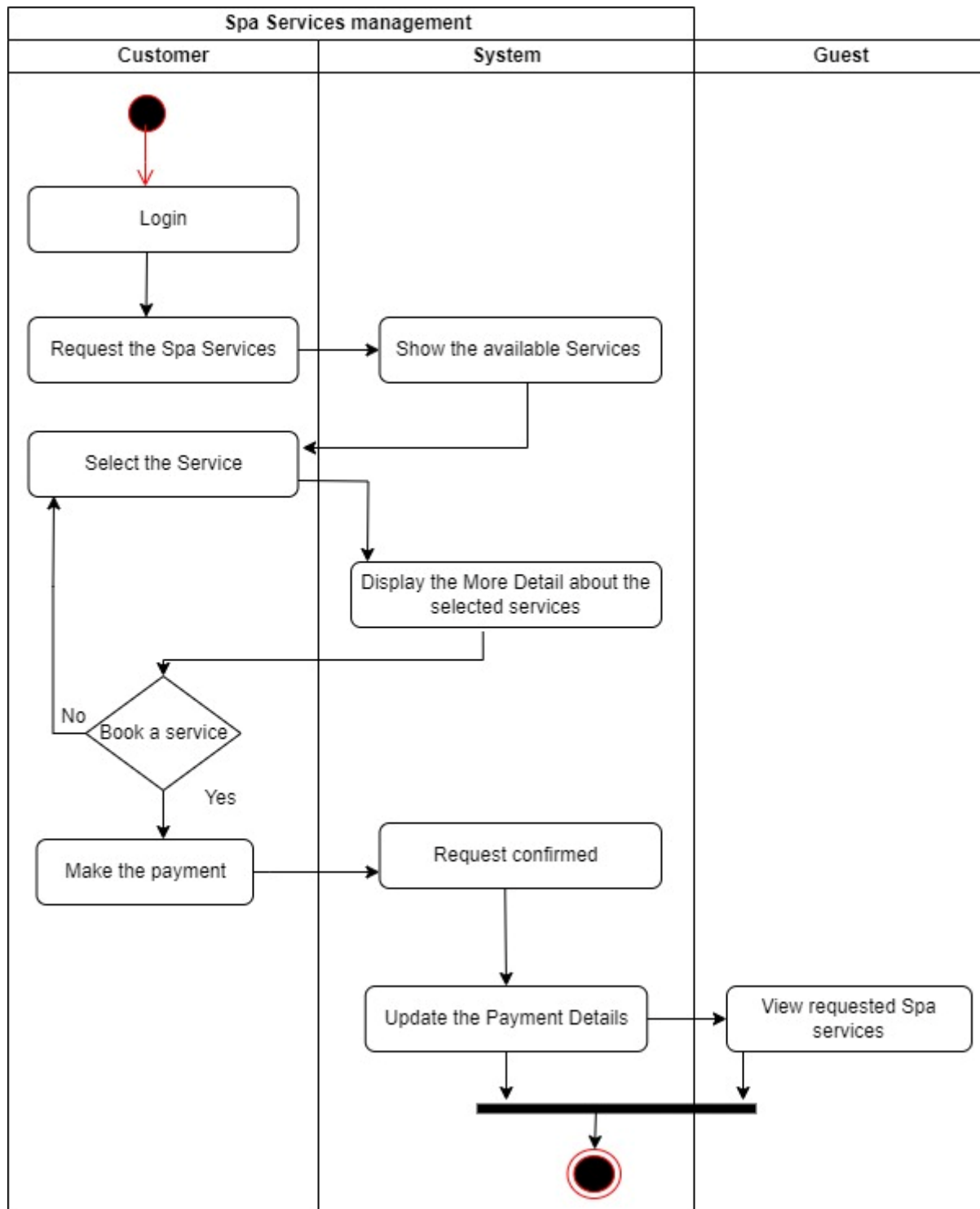


Figure3 5:Activity Diagram-Spa Service reservation

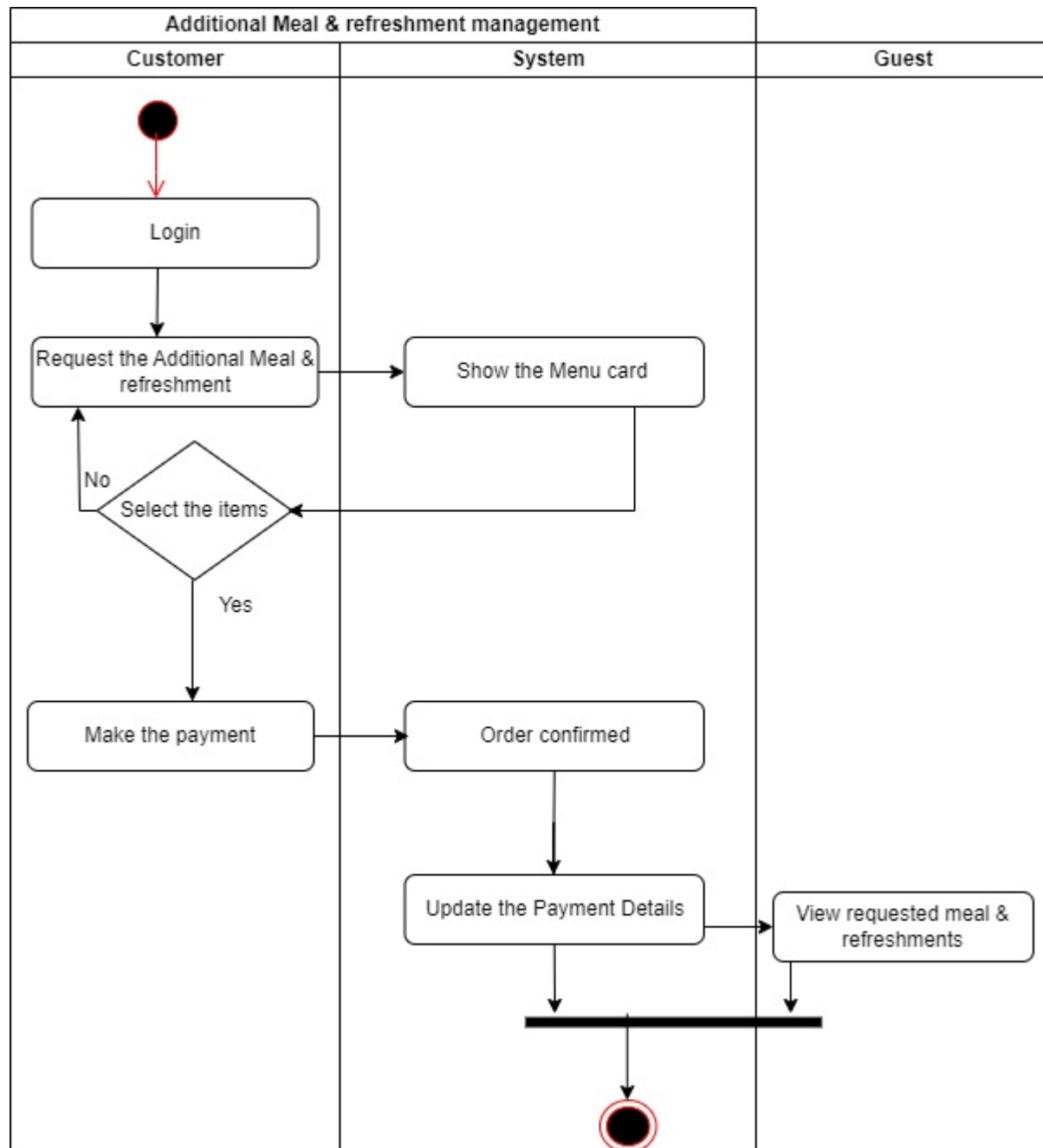


Figure3 6:Activity Diagram-Additional meal reservation

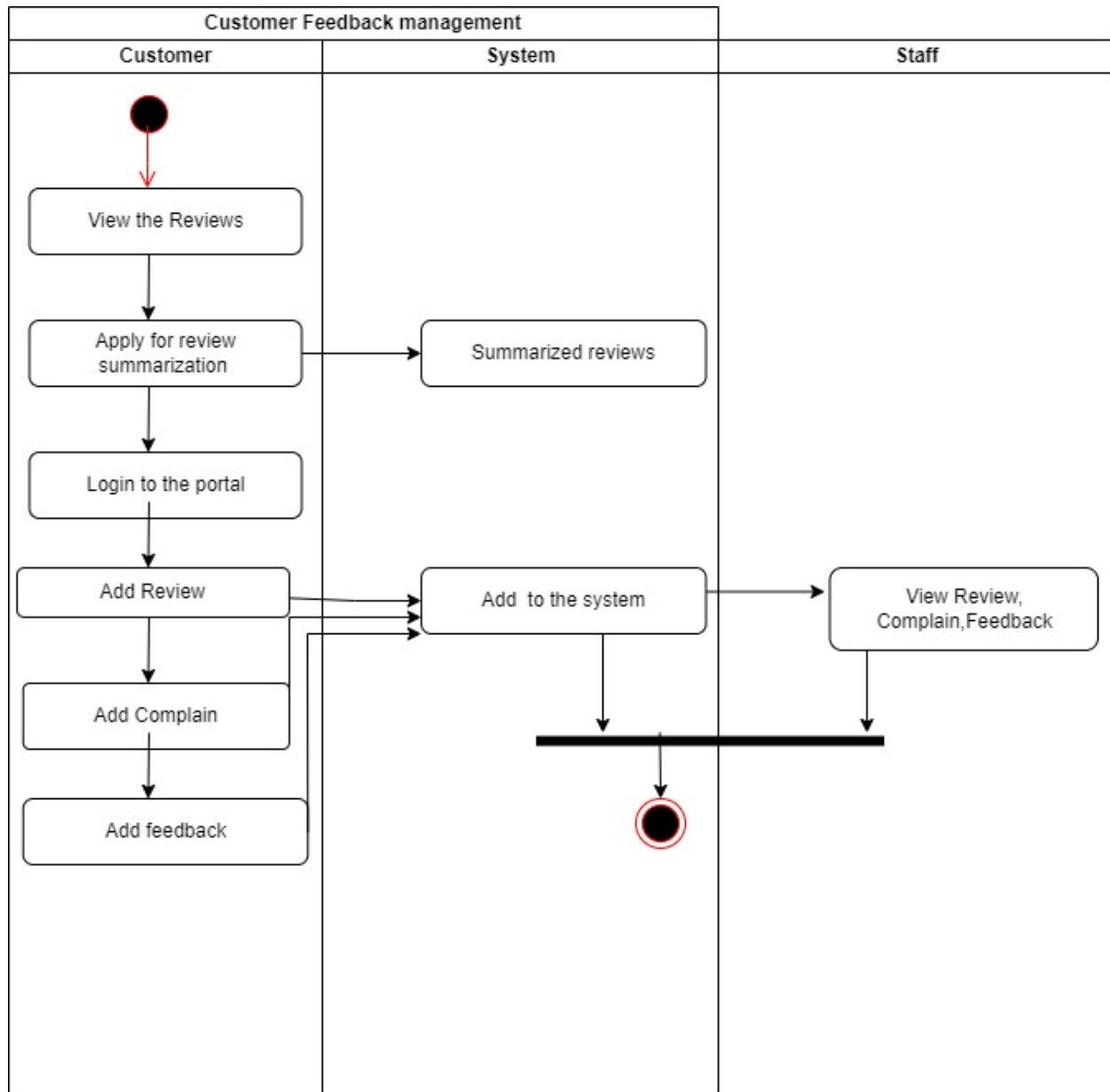


Figure3 7:Feedback Management

3.5 Database Design

3.5.1.ER Diagram

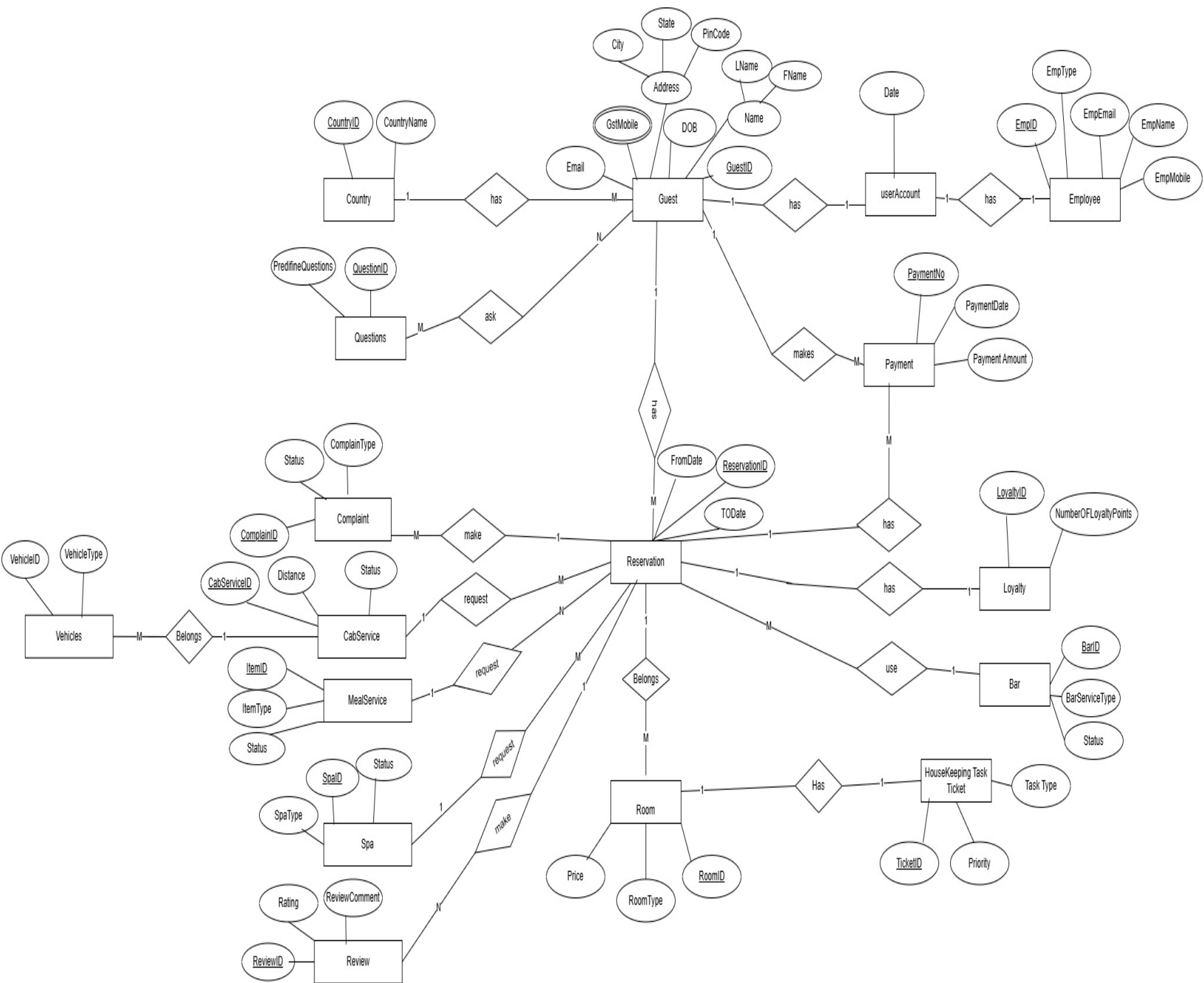


Figure3 8:ER Diagram

3.5.2 Class Diagram

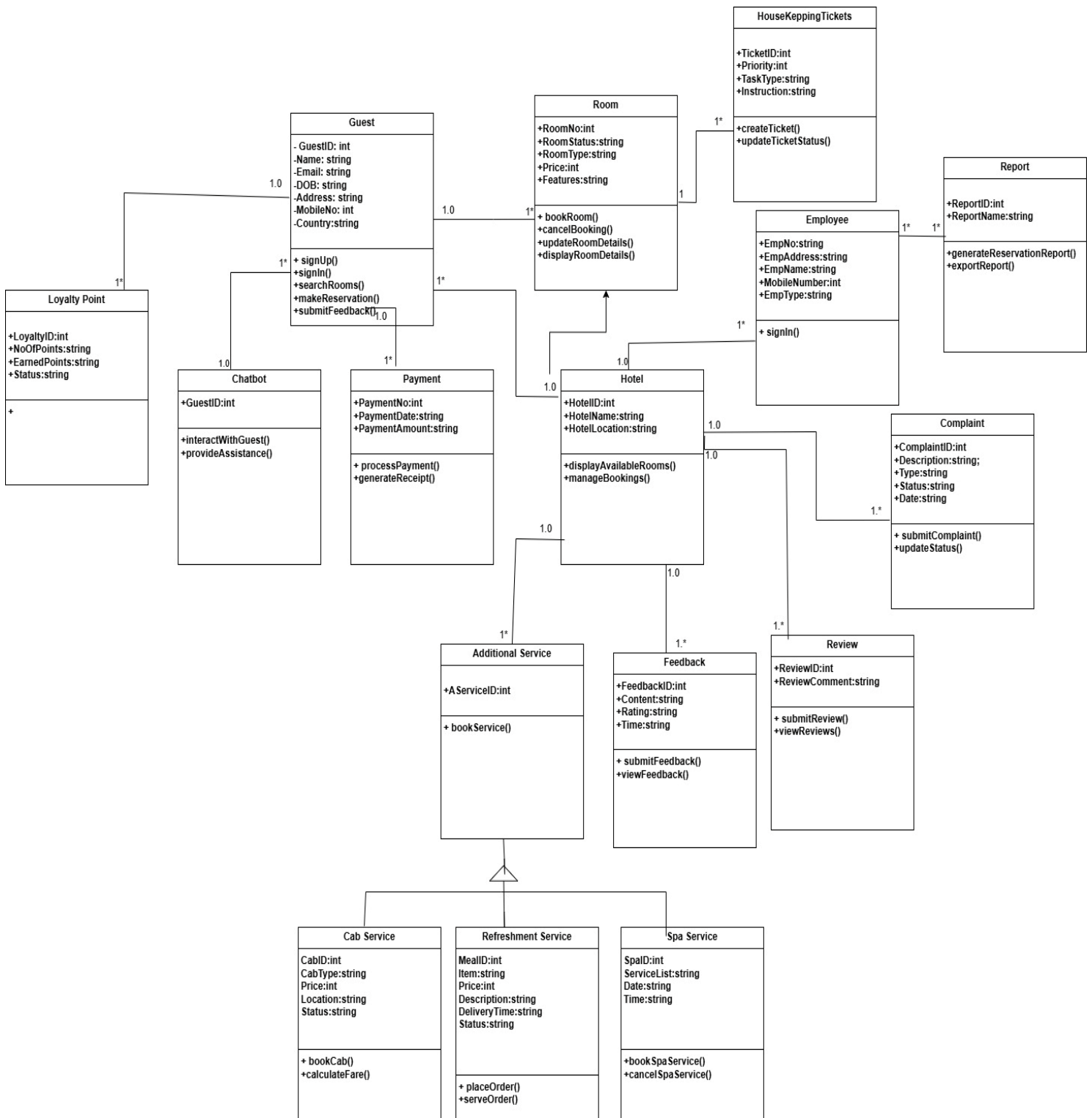


Figure3 9:Class Diagram

3.5.3. Database Structure

Structure	SQL	Search	Query	Export	Import	Operations	Privileges	Routines	Events	Triggers
☐ cancel_reservations	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	48.0 KiB	-
☐ complaints	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	32.0 KiB	-
☐ contacts	★	Browse	Structure	Search	Insert	Empty	Drop	7 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ countries	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ discounts	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ employees	★	Browse	Structure	Search	Insert	Empty	Drop	5 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ failed_jobs	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	32.0 KiB	-
☐ feedback	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ guests	★	Browse	Structure	Search	Insert	Empty	Drop	6 InnoDB utf8mb4_unicode_ci	64.0 KiB	-
☐ hotel_rooms	★	Browse	Structure	Search	Insert	Empty	Drop	9 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ hotel_room_reservation	★	Browse	Structure	Search	Insert	Empty	Drop	91 InnoDB utf8mb4_unicode_ci	64.0 KiB	-
☐ jobs	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	32.0 KiB	-
☐ job_batches	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ migrations	★	Browse	Structure	Search	Insert	Empty	Drop	73 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ password_reset_tokens	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ payments	★	Browse	Structure	Search	Insert	Empty	Drop	24 InnoDB utf8mb4_unicode_ci	32.0 KiB	-
☐ pickcabs	★	Browse	Structure	Search	Insert	Empty	Drop	6 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ pickguests	★	Browse	Structure	Search	Insert	Empty	Drop	208 InnoDB utf8mb4_unicode_ci	64.0 KiB	-
☐ pickmeals	★	Browse	Structure	Search	Insert	Empty	Drop	5 InnoDB utf8mb4_unicode_ci	32.0 KiB	-
☐ pickpaycashes	★	Browse	Structure	Search	Insert	Empty	Drop	1 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ pickpays	★	Browse	Structure	Search	Insert	Empty	Drop	0 InnoDB utf8mb4_unicode_ci	16.0 KiB	-
☐ pickreviews	★	Browse	Structure	Search	Insert	Empty	Drop	15 InnoDB utf8mb4_unicode_ci	16.0 KiB	-

Figure3 10:Database Design

3.4 Chapter Summary

The design chapter provides a structured framework for the Hotel Reservation & Customer Feedback Management System, detailing its architecture, use case diagram, ER Diagram, Class Diagram. It outlines the centralized database design for managing reservations, feedback, and user data efficiently.

Chapter 4 – Implementation

4.1 Introduction

The implementation phase involves converting the system’s specifications into executable code, ensuring that the Hotel Reservation and Customer Feedback Management System for Sitara Hotel is both functional and user-friendly. This phase translates the design into interactive front-end interfaces and structured back-end logic to handle system functionalities efficiently.

The system was implemented with a well-structured codebase, ensuring clarity and maintainability. This chapter details the technologies used, system architecture, coding structure, and key functionalities developed, ensuring a seamless integration of reservation management and customer feedback processing

4.2 Related Technologies and Tools

Application	Technology	Tools
Web-based Hotel Reservation and Customer Feedback Management System for the Sitara Hotel	Bootstrap	Visual Studio Code
	CSS	
	HTML	
	Java Script	
	Review Summarization Libraries	
	PHP	
	Python	
	MySQL	MySQL workbench 8.0
	Excel	MS Office

Table4 1:Related Technologies and Tools

4.3 Modules of the System

The Hotel Reservation & Customer Feedback Management System for Sitara Hotel, Weligama, consists of several key modules to enhance hotel operations and customer experience. The User Management Module handles user registration, authentication, and role-based access. The Room Reservation Module enables guests to search for available rooms, make bookings, and process payments. The Customer Feedback & Review Module allows guests to submit reviews, while the Hotel Services Module manages cab bookings, photo location reservations, and special offers. Additionally, the system includes Front Desk, Housekeeping, Marketing, and Reporting Modules to streamline hotel management, improve efficiency, and enhance decision-making.

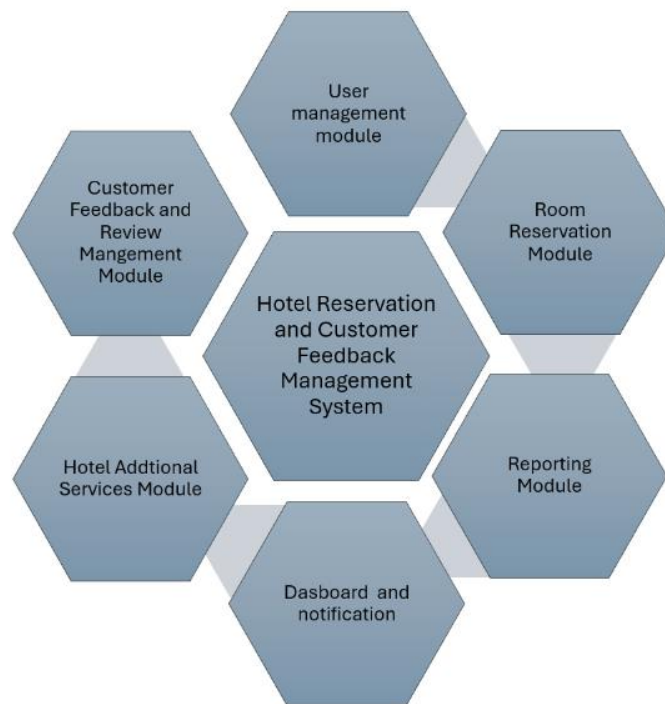


Figure4 1:Modules of the system

4.4 Application Development Architecture

The Hotel Reservation & Customer Feedback Management System for Sitara Hotel, Weligama is developed using the Model-View-Controller (MVC) Architecture, ensuring scalability, maintainability, and efficient separation of concerns. The Model layer handles the business logic and database operations, including storing and retrieving data related to reservations, customer feedback, and user accounts. The View layer provides the user interface, allowing guests, hotel staff, and management to interact with the system through web or mobile applications. The Controller layer acts as a bridge between the Model and View, processing user inputs, handling requests, and updating the data accordingly. This architectural approach enhances modularity, allowing independent modifications to the UI, logic, or data management without affecting other components. Additionally, it supports better testing, making the system more reliable and easier to maintain.

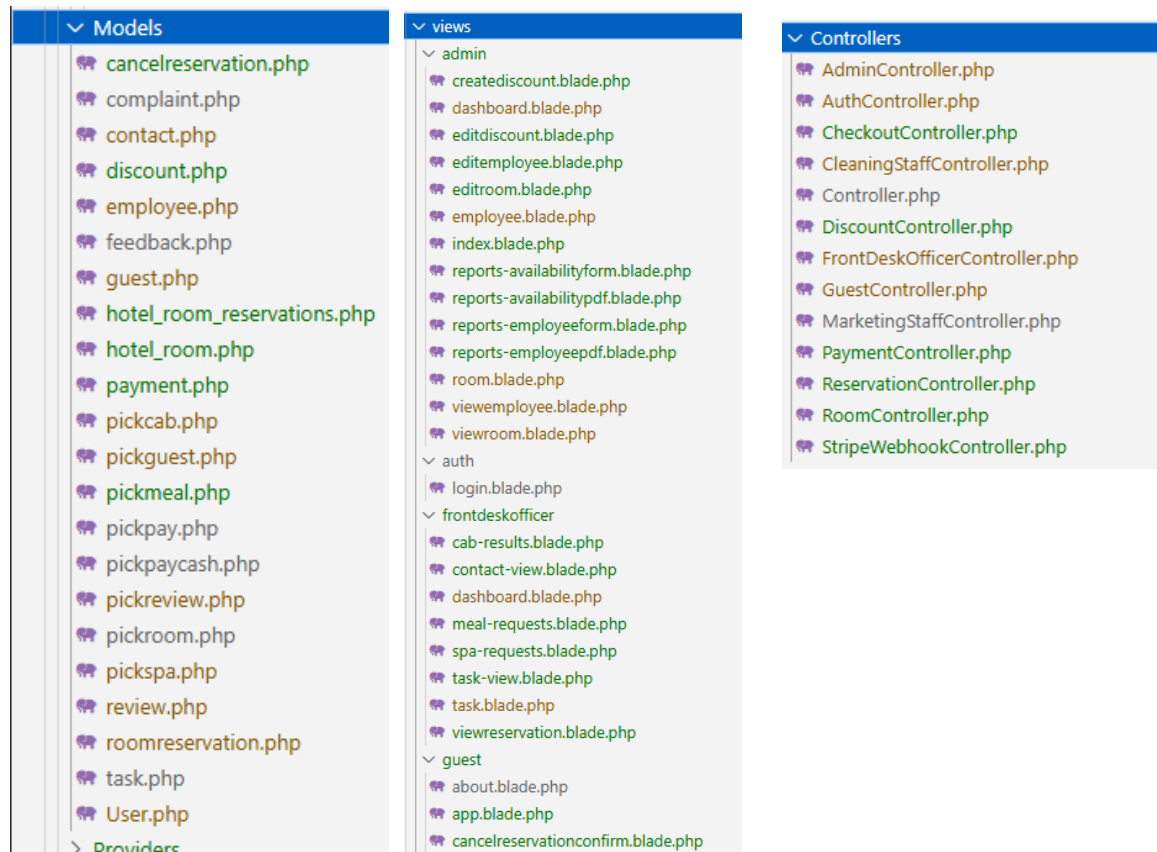


Figure4 2:Application Development Architecture

4.5 Middleware

In the Hotel Reservation & Customer Feedback Management System for Sitara Hotel, Weligama, middleware plays a crucial role in enabling seamless communication between the application's frontend, backend. It acts as an intermediary layer that manages authentication and data validation to ensure smooth data exchange. Middleware helps in handling user requests efficiently and managing communication between different user roles. By integrating middleware, the system ensures reliability and seamless user experience across all hotel management functionalities.



Figure4 3:Middleware

```
public function AuthLogin(Request $request): RedirectResponse
{
    if (Auth::attempt(credentials: ['email' => $request->email, 'password' => $request->password], remember: true)) {
        if (Auth::user()->user_type == 'Admin') {
            return redirect(to: 'admin/dashboard');
        }

        if (Auth::user()->user_type == 'FrontDeskOfficer') {
            return redirect(to: 'FrontDeskOfficer/dashboard');
        }

        if (Auth::user()->user_type == 'MarketingStaff') {
            return redirect(to: 'MarketingStaff/dashboard');
        }

        if (Auth::user()->user_type == 'guest') {
            return redirect(to: 'guest/dashboard');
        }

        if (Auth::user()->user_type == 'CleaningStaff') {
            return redirect(to: 'CleaningStaff/dashboard');
        }
    } else {
        return redirect()->back()->with(key: 'error', value: 'Invalid Email and Password');
    }
}
```

Figure4 4:Auth Controller

4.6 Main User Interfaces and Coding

4.6.1 Room Reservation and Availability Checking

Check Availability

The "Check Availability" function enables users to find available rooms by selecting their check-in and check-out dates, the number of guests, and the room count. It checks availability and displays the available room options with details, including the amount.

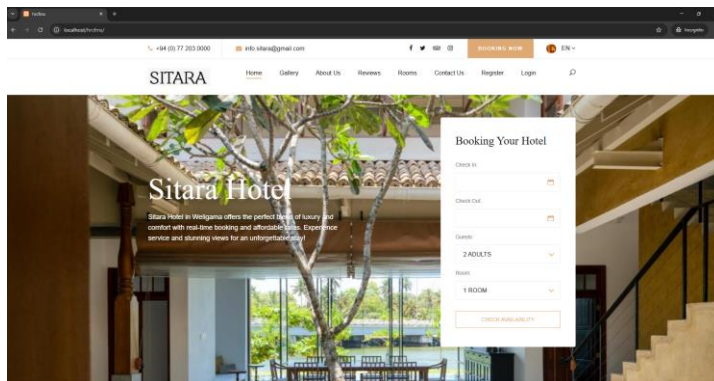


Figure4 5:Room availability checking

```
public function checkAvailability(Request $request): View
{
    // Validate the request
    $request->validate(rules: [
        'check_in_date' => 'required|date',
        'check_out_date' => 'required|date|after:check_in_date',
        'facilitate_adults' => 'required|integer|min:1',
        'facilitate_childs' => 'required|integer|min:0',
        'room_count' => 'required|integer|min:1',
    ]);

    // Retrieve the input values
    $checkInDate = $request->input(key: 'check_in_date');
    $checkOutDate = $request->input(key: 'check_out_date');
    $adults = $request->input(key: 'facilitate_adults');
    $children = $request->input(key: 'facilitate_childs');
    $roomCount = $request->input(key: 'room_count');
    $room_type1 = $request->input(key: 'Single');
    $room_type2 = $request->input(key: 'Double');
    $room_type3 = $request->input(key: 'Thribble');

    // Store search criteria in the session
    session(key: [
        'check_in_date' => $checkInDate,
        'check_out_date' => $checkOutDate,
        'facilitate_adults' => $adults,
        'facilitate_childs' => $children,
        'room_count' => $roomCount,
    ]);

    //Get rooms that are not reserved for the selected dates
    $availableRooms = hotel_room::where(column: 'facilitate_adults', operator: '>=', value: $adults)
        ->where(column: 'facilitate_childs', operator: '>=', value: $children)
        ->whereDoesntHave(relation: 'reservations', callback: function ($query) use ($checkInDate, $checkOutDate): void {
            $query->where(function ($q) use ($checkInDate, $checkOutDate): void {
                $q->where('check_out_date', '>', $checkInDate)
                    ->where('check_in_date', '<', $checkOutDate);
            });
        });
    ->get();
}
```

Figure4 6:Room availability Checking -code

Arrange Discounts

The system provides an interface that enables the admin user to create new discounts. Through this interface, the admin can define the discount name, specify the discount percentage, set the validity period (including start and end dates), and indicate whether the discount is exclusive to registered guests. This functionality allows the admin to offer discounts targeted either at specific loyal customers or to all customers, thereby supporting flexible promotional strategies within the system.

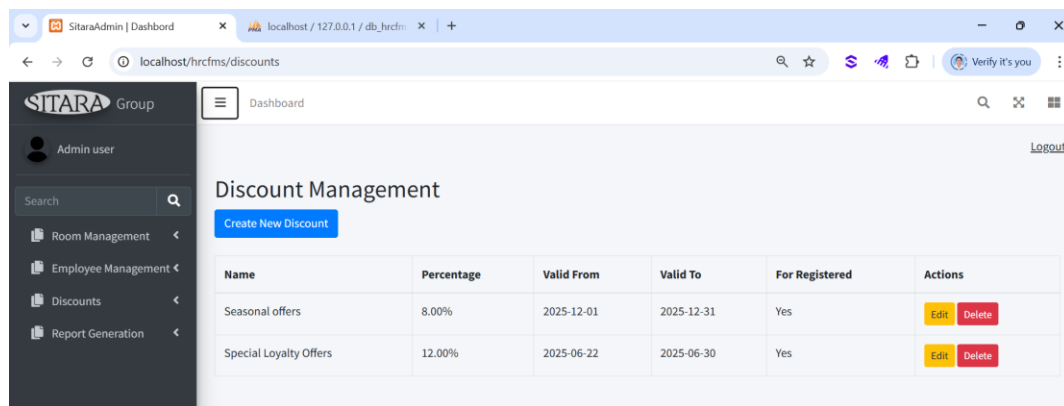


Figure4 7:Create Discounts

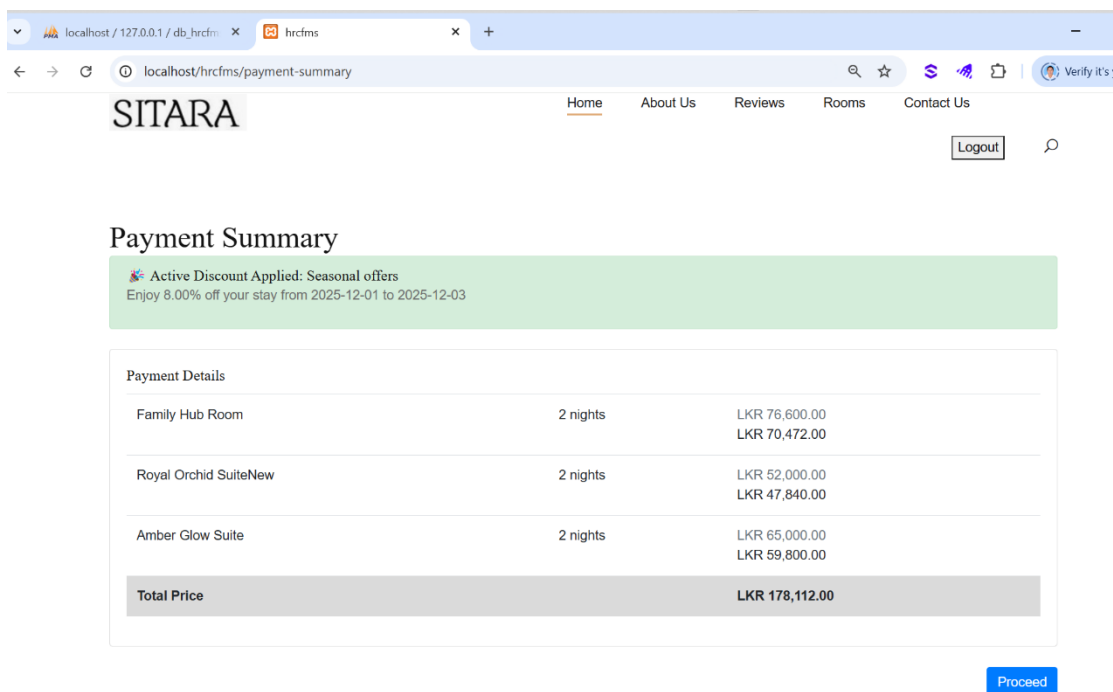


Figure4 8:Apply the Discount

```

1 reference | 0 overrides
public function store(Request $request): RedirectResponse
{
    // Validate request
    $validated = $request->validate(rules: [
        'name' => 'required|string|max:255',
        'percentage' => 'required|numeric|between:1,100',
        'valid_from' => 'required|date',
        'valid_to' => 'required|date|after:valid_from',
        'for_registered' => 'sometimes'
    ]);

    // Handle checkbox value
    $forRegistered = $request->has(key: 'for_registered'); // Returns true/false

    // Create discount
    Discount::create(attributes: [
        'name' => $validated['name'],
        'percentage' => $validated['percentage'],
        'valid_from' => $validated['valid_from'],
        'valid_to' => $validated['valid_to'],
        'for_registered' => $forRegistered // Store as boolean
    ]);

    return redirect()->route(route: 'admin.index')
        ->with(key: 'success', value: 'Discount created successfully');
}

```

```

public function update(Request $request, Discount $discount): RedirectResponse
{
    $validated = $request->validate(rules: [
        'name' => 'required|string|max:255',
        'percentage' => 'required|numeric|between:1,100',
        'valid_from' => 'required|date',
        'valid_to' => 'required|date|after:valid_from',
        'for_registered' => 'sometimes'
    ]);

    $discount->update(attributes: [
        'name' => $validated['name'],
        'percentage' => $validated['percentage'],
        'valid_from' => $validated['valid_from'],
        'valid_to' => $validated['valid_to'],
        'for_registered' => $request->has(key: 'for_registered')
    ]);

    return redirect()->route(route: 'admin.index')
        ->with(key: 'success', value: 'Discount updated successfully');
}

```

```

1 reference | 0 overrides
public function destroy($id): RedirectResponse
{
    $discount = Discount::findOrFail(id: $id);
    $discount->delete();

    return redirect()->route(route: 'admin.index')->with(key: 'success', value: 'Discount deleted successfully.');
```

Figure4 9:Discounts- code

Payment Handling

Payment Details		
Sea View RoomEdited	1 nights	LKR 39,000.00
Sun Shine Room	1 nights	LKR 22,500.00
Royal Orchid SuiteNew	1 nights	LKR 26,000.00
Total Price		LKR 87,500.00

Proceed

Figure4 10:Payment Summary

```
class PaymentController extends Controller
{
    1 reference | 0 overrides
    public function showPaymentSummary(): View
    {
        // Retrieve selected rooms from the session
        $selectedRooms = session(key: 'selectedRooms', default: []);
        $rooms = hotel_room::whereIn(column: 'id', values: $selectedRooms)->get();

        // Get reservation dates from session
        $checkInDate = session(key: 'check_in_date');
        $checkOutDate = session(key: 'check_out_date');

        // Convert to Carbon instances for proper date handling
        $checkIn = Carbon::parse(time: $checkInDate);
        $checkOut = Carbon::parse(time: $checkOutDate);
        //$nights = $checkOut->diffInDays($checkIn);
        $nights = $checkIn->diffInDays(date: $checkOut);
        //dd($nights);

        // Initialize discount variables
        $discount = 0;
        $discountName = '';
        $basePrices = [];
        $validDiscounts = collect();

        // Check for authenticated guest user
        if (Auth::check() && Auth::user()->user_type === 'guest') {
            $validDiscounts = Discount::where(column: 'for_registered', operator: true)
                ->where(column: function (Discount $query) use ($checkIn, $checkOut): void {
                    $query->where(column: function (Discount $q) use ($checkIn, $checkOut): void {
                        // Discount period completely covers reservation
                        $q->where(column: 'valid_from', operator: '<=', value: $checkIn)
                            ->where(column: 'valid_to', operator: '>=', value: $checkOut);
                    })->orWhere(column: function (Discount $q) use ($checkIn, $checkOut): void {
                        // Discount period overlaps with reservation
                        $q->whereBetween(column: 'valid_from', values: [$checkIn, $checkOut])
                            ->orWhereBetween(column: 'valid_to', values: [$checkIn, $checkOut]);
                    });
                })
                ->orderBy(column: 'percentage', direction: 'desc')
                ->get();
        }
    }
}
```

> signup

```

// Apply best available discount
if ($validDiscounts->isEmpty()) {
    $bestDiscount = $validDiscounts->first();
    $discount = $bestDiscount->percentage;
    $discountName = $bestDiscount->name;
}

// Calculate prices with discount
$totalPrice = 0;
foreach ($rooms as $room) {
    $basePrice = $room->price_per_night * $nights;
    $basePrices[] = $basePrice;

    $discountedPrice = $discount > 0
        ? $basePrice * (1 - ($discount / 100))
        : $basePrice;

    $totalPrice += $discountedPrice;
}
// dd($totalPrice);

return view(view: 'guest.summary', data: [
    'rooms' => $rooms,
    'totalPrice' => $totalPrice,
    'nights' => $nights,
    'discount' => $discount,
    'discountName' => $discountName,
    'basePrices' => $basePrices,
    'checkInDate' => $checkIn->format(format: 'Y-m-d'),
    'checkOutDate' => $checkOut->format(format: 'Y-m-d')
]);
}

```

Figure4 11:Payment Summary-code

Online Payment Gateway Integration

The screenshot shows a web browser window with the URL `localhost/hrcfms/stripe-payment`. The page header includes the SITARA logo, navigation links (Home, About Us, Reviews, Rooms, Contact Us, Register, Login), and contact information (+94 (0) 77 203 0000, info.sitara@gmail.com). A "BOOKING NOW" button is also present. The main content area displays a "Complete Payment" form with a Visa card number (4242 4242 4242 4242), expiration date (12 / 30), and a "Pay LKR 87500" button. A "Save with link" option is also visible.

Figure4 12:Payment Gateway Integration

```

class PaymentController extends Controller {
  public function handleStripePayment(Request $request): RedirectResponse|View
  {
    try {
      // Get necessary data from session
      $reservationData = [
        'hotel_room_id' => session(key: 'selected_room_id'),
        'guest_id' => auth()->id(),
        'check_in_date' => session(key: 'check_in_date'),
        'check_out_date' => session(key: 'check_out_date'),
        'status' => 'pending',
        'total_price' => session(key: 'totalPrice'),
        'nights' => session(key: 'nights'),
        'is_checked_in' => 0,
        'is_checked_out' => 0,
        'created_at' => now(),
        'updated_at' => now()
      ];

      // Create reservation first
      $reservation = hotel_room_reservations::create(attributes: $reservationData);

      // Proceed with Stripe payment
      $stripe = new StripeClient(config(key: 'services.stripe.secret'));
      $totalPrice = session(key: 'totalPrice');

      $paymentIntent = $stripe->paymentIntents->create(params: [
        'amount' => intval(value: $totalPrice * 100), // Convert to integer cents
        'currency' => 'usd',
        'payment_method_types' => ['card'],
        'confirmation_method' => 'automatic', // Optional but safe
        'metadata' => [
          'reservation_id' => $reservation->id,
          'guest_id' => auth()->id(),
        ]
      ]);

      return view(view: 'payment.stripe', data: [
        'clientSecret' => $paymentIntent->client_secret,
        'totalPrice' => session(key: 'totalPrice')
      ]);
    } catch (\Exception $e) {
      return redirect()->back()->withErrors(provider: ['error' => $e->getMessage()]);
    }
  }
}

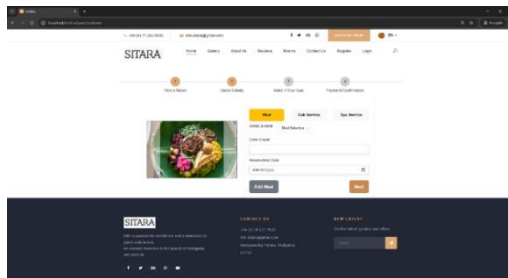
```

Figure4 13:Payment Gateway Integration-code

4.6.4. Additional Service Reservation (Meal, Cab Services, Payment Services)

Meal Reservation page

The Meal Reservation feature allows guests to make meal requests by selecting the meal type, specifying the quantity, and reservation date.



Meal Requests Management

Guest Name	Meal Selection	Reservation Date	Guest Count	Mobile	Status	Actions
sehansa samadi	Breakfast at Central	Mar 26, 2025	2	0767656541	Denied	Deny
serani pereia	Breakfast at Central	Mar 18, 2025	2	0767676564	Pending	Pending
sedani sehansa	Breakfast at Central	Mar 10, 2025	2	0767676541	Approved	Approve
Thineth Sehansa	Dinner at Central	Mar 25, 2025	3	0767676541	Approved	Approve
Dedunu Dehansa	Breakfast at Central	Mar 20, 2025	2	0767656471	Denied	Deny

Figure4 14:Meal Reservation UI

```
public function viewMealRequests(): View
{
    $mealRequests = pickmeal::orderBy(column: 'created_at', direction: 'desc')->get();
    return view(view: 'frontdeskofficer.meal-requests', data: compact(var_name: 'mealRequests'));
}

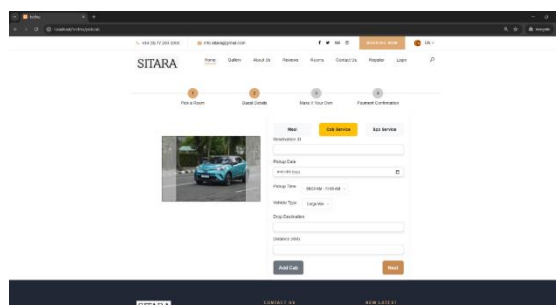
1 reference | 0 overrides
public function updateMealStatus(Request $request, Pickmeal $mealRequest): RedirectResponse
{
    $request->validate(rules: [
        'status' => 'required|in:Pending,Approved,Denied'
    ]);

    $mealRequest->update(attributes: [
        'status' => $request->status
    ]);

    return back()->with(key: 'success', value: 'Meal request status updated!');
}
```

Figure4 15:Meal Reservation -code

Cab Service Reservation page



Cab Service Requests

Guest Name	Vehicle Type	Pickup Date/Time	Destination	Mobile	Status	Actions
Sehansa samadi	Car	Mar 05, 2025 12:00 PM	Nuwara	0767288124	Approved	Approve
sehansa sandeepani	Car	Mar 16, 2025 6:30 AM	weligama	0767288124	Approved	Approve
Dehansa Damsarani	Car	Mar 10, 2025 5:30 PM	Pallana town	0767654341	Approved	Approve
Dehansa Dweethani	Car	Mar 25, 2025 6:40 AM	weligama town	0767676541	Denied	Deny
Sehuthi sathisara	Car	Mar 28, 2025 12:00 PM	Nuwaraeliya	0767676541	Approved	Approve
sehansa sathisara	Car	Mar 29, 2025 7:00 AM	nuwara	0767288124	Approved	Approve

Figure4 16:Cab service Reservation UI

```

public function viewCabRequests(): View
{
    $cabRequests = pickcab::orderBy(column: 'pickup_date', direction: 'asc')
    ->orderBy(column: 'pickup_time', direction: 'asc')
    ->get();

    return view(view: 'frontdeskofficer.cab-results', data: compact(var_name: 'cabRequests'));
}

1 reference | 0 overrides
public function updateCabStatus(Request $request, $id): RedirectResponse
{
    $request->validate(rules: [
        'status' => 'required|in:Pending,Approved,Denied'
    ]);

    $cabRequest = Pickcab::findOrFail(id: $id);
    $cabRequest->update(attributes: [
        'status' => $request->status
    ]);

    return redirect()->back()->with(key: 'success', value: 'Cab status updated successfully');
}

```

Figure4 17:Cab service Status-code

Spa Services Reservation page

The Spa Service Reservation feature allows guests to book spa treatments by selecting the service type, appointment date, and preferred time slot.

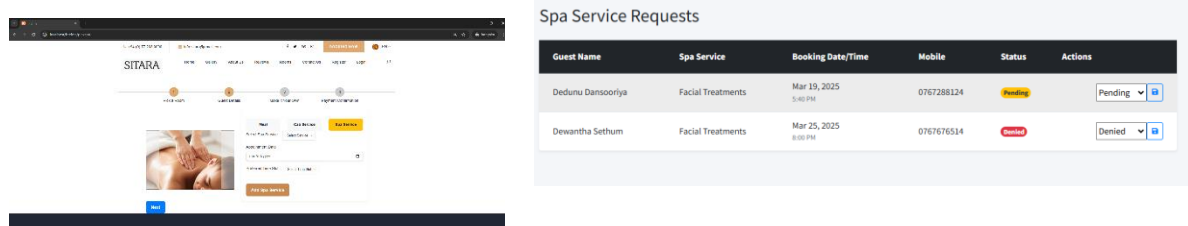


Figure4 18:Spa service Reservation UI

```

public function viewSpaRequests(): View
{
    $spaRequests = pickspa::orderBy(column: 'preferred_date', direction: 'asc')
    ->orderBy(column: 'preferred_time', direction: 'asc')
    ->get();

    return view(view: 'frontdeskofficer.spa-requests', data: compact(var_name: 'spaRequests'));
}

1 reference | 0 overrides
public function updateSpaStatus(Request $request, pickspa $spaRequest,$id): RedirectResponse
{
    $request->validate(rules: [
        'status' => 'required|in:Pending,Approved,Denied'
    ]);

    $spaRequest =pickspa::findOrFail(id: $id);
    $spaRequest->update(attributes: [
        'status' => $request->status
    ]);

    return redirect()->back()->with(key: 'success', value: 'Status updated successfully');
}

```

Figure4 19:Spa Services management-code

4.6.5. Add Review Page

The "Add Review" page in Sitara Hotel allows guests to provide feedback on their stay, offering a platform for users to share their experiences and rate the services. This feature plays a crucial role in improving service quality by gathering valuable insights that can be utilized for enhancing guest satisfaction and operational efficiency.

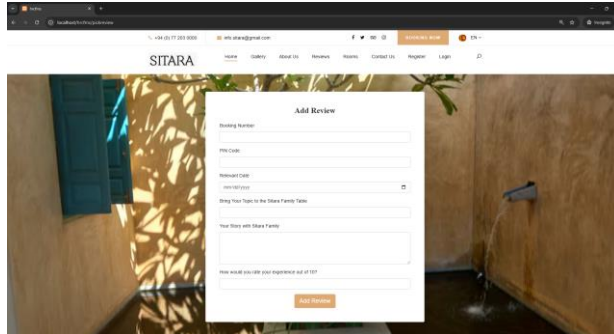


Figure4 20:Add Review UI

4.6.6. Lengthy Review Summarization

The Review Summarization functionality allows users to view both detailed reviews and their summarized versions. Each review includes a "Summarize Review" button, which, when clicked, displays a concise summary of the review. This summary highlights the key points of the guest's experience, such as the positive aspects of their stay and any minor issues they encountered. The summarization helps hotel management quickly assess guest feedback and gain insights into overall satisfaction without reading through lengthy reviews.

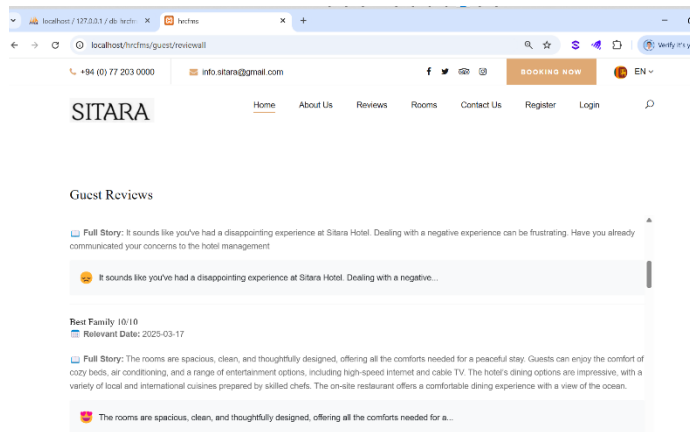


Figure4 21:Lengthy review summarization UI

```
from flask import Flask, request, jsonify
from transformers import pipeline

app = Flask(__name__)

# Load the summarization pipeline
summarizer = pipeline("summarization")

def get_summary(text):
    # Calculate max_length as 30% of the input text length
    input_length = len(text.split()) # Count words in the input text
    max_length = max(30, int(input_length * 0.3)) # Ensure a minimum length for very short texts
    min_length = max(10, int(max_length * 0.5)) # Set min_length to be 50% of max_length

    # Use the pipeline to summarize text
    summarized = summarizer(text, max_length=max_length, min_length=min_length, do_sample=False)
    return summarized[0]['summary_text']

@app.route('/summarize', methods=['POST'])
def summarize_text():
    data = request.get_json() # Get JSON data sent to the endpoint
    text = data.get('text') # Extract text field from JSON data

    if not text:
        return jsonify({'error': 'No text provided'}), 400

    # Get summary from the get_summary function
    summary = get_summary(text)
    print("Summary:", summary) # Print the summary to the console

    return jsonify({'summary': summary}) # Return the summary as JSON

if __name__ == '__main__':
    app.run(debug=True)
```

Figure4 22:Review Summarization-model

```

private function callSummarizationApi($text): mixed
{
    try {
        $client = new \GuzzleHttp\Client();
        $response = $client->post(uri: 'http://localhost:5000/summarize', options: [
            'json' => ['text' => $text],
            'timeout' => 10,
            'headers' => [
                'Accept' => 'application/json',
            ]
        ]);

        $data = json_decode(json: $response->getBody(), associative: true);
        return $data['summary'] ?? null;
    } catch (RequestException $e) {
        Log::error(message: 'API Request failed: ' . $e->getMessage());
        if ($e->hasResponse()) {
            Log::error(message: 'API Response: ' . $e->getResponse()->getBody());
        }
        return null;
    } catch (\Exception $e) {
        Log::error(message: 'General error: ' . $e->getMessage());
        return null;
    }
}

1 reference
private function localFallbackSummary($text): string
{
    $sentences = preg_split(pattern: '/(?<=[?!])\s+/', subject: $text);
    $fallback = Str::words(value: implode(separator: ' ', array: array_slice(array: $sentences, offset: 0, len: 2)), 15);
    return $fallback ?: 'Detailed review available';
}

```

Figure 4 23: Review Summarization-Model Integration

4.6.7. Contact Us page

The "Contact Us" page of Sitara Hotel Weligama serves as a direct communication channel between the hotel and its guests. It displays the hotel's physical address with clear location detail with Google map. Additionally, the page features a form where clients can submit their inquiries or questions. This form allows users to input relevant information, ensuring that their concerns are addressed promptly. By offering this platform, Sitara Hotel enhances customer service by facilitating easy and efficient communication with guests.

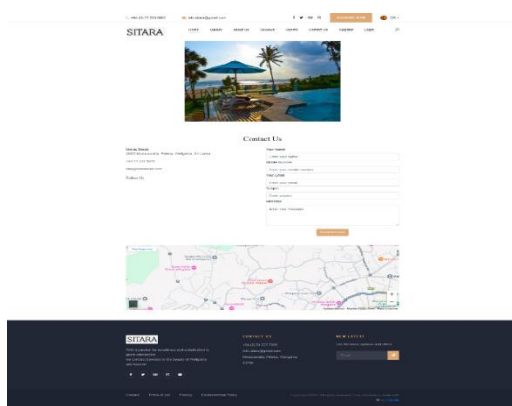


Figure4 24:Contact us client page

Contact Messages

Name	Contact Info	Subject	Message	Status	Actions
sehanza soca soca@gmail.com	0767288124	Anniversary	Is there any anniversary offers?	Pending	Pending ✕
dehani sameera dehani@gmail.com	076767654	Anniversary	Is there any offers for the meal reservation	Resolved	Resolved ✕
sewantha samadi sewantha@gmail.com	076767654	Meals	Can it arrange special meals?	Resolved	Resolved ✕
neha sadasaya neha@gmail.com	0767654541	Recipe	Can we arrange own recipe	Pending	Pending ✕
nthuli sadalka nthuli@gmail.com	076767654	Anniversary	Need about anniversary packages	Resolved	Resolved ✕
Dimuthi diwayang diwayang@gmail.com	0767288124	Anniversary Celebration	Need to prepare the room, having our wedding anniversary, is it possible	Resolved	Resolved ✕
Dahandi damasooriya dahandi@gmail.com	0767654342	Spa Services	Need Info about Spa Services	Resolved	Resolved ✕

Figure4 25:Contact us management -Admin UI

```
public function updateContactStatus(Request $request, $id): RedirectResponse
{
    $request->validate(rules: [
        'status' => 'required|in:Pending,In Progress,Resolved'
    ]);

    $contact = Contact::findOrFail(id: $id);
    $contact->update(attributes: [
        'status' => $request->status
    ]);

    return back()->with(key: 'success', value: 'Contact status updated successfully');
}
```

Figure4 26:Contact us-code

4.6.8. Dashboard

Admin Dashboard

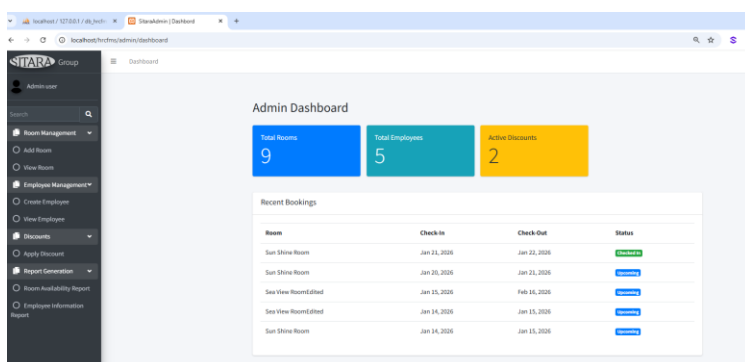


Figure4 27:Admin Dashboard UI

```
1 reference | 0 overrides
public function dashboard(): View
{
    $stats = [
        'totalRooms' => hotel_room::count(),
        'availableRooms' => hotel_room::where(column: 'availability', operator: 1)->count(),
        'totalEmployees' => Employee::count(),
        'activeDiscounts' => Discount::where(column: 'valid_to', operator: '>=', value: now())->count(),
        'recentBookings' => hotel_room_reservations::with(relations: 'room')
            ->orderBy(column: 'check_in_date', direction: 'desc')
            ->limit(value: 5)
            ->get()
    ];

    return view(view: 'admin.dashboard', data: $stats);
}
```

Figure4 28:Admin Dashboard -code

Front Desk Officer Dashboard

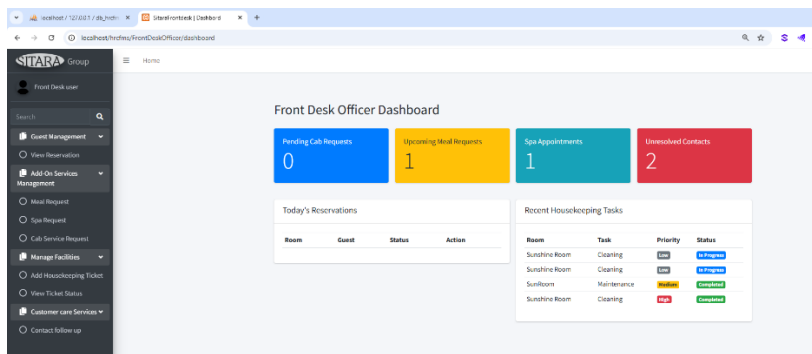


Figure4 29:Front Desk Officer Dashboard-UI

```
1 reference | 0 overrides
public function dashboard(): View
{
    $stats = [
        'pendingCabRequests' => pickcab::where(column: 'status', operator: 'Pending')->count(),
        'pendingMealRequests' => pickmeal::where(column: 'status', operator: 'Pending')->count(),
        'pendingSpaRequests' => pickspa::where(column: 'status', operator: 'Pending')->count(),
        'pendingContacts' => contact::whereIn(column: 'status', values: ['Pending', 'In Progress'])->count(),
        'recentReservations' => hotel_room_reservations::with(relations: ['room', 'guest'])
            ->whereDate(column: 'check_in_date', operator: today())
            ->orderBy(column: 'check_in_date')
            ->limit(value: 5)
            ->get(),
        'recentTasks' => Task::orderBy(column: 'created_at', direction: 'desc')
            ->limit(value: 5)
            ->get()
    ];

    return view(view: 'frontdeskoofficer.dashboard', data: $stats);
}
```

Figure4 30:Front Desk Officer Dashboard-code

4.6.9. Employee Management

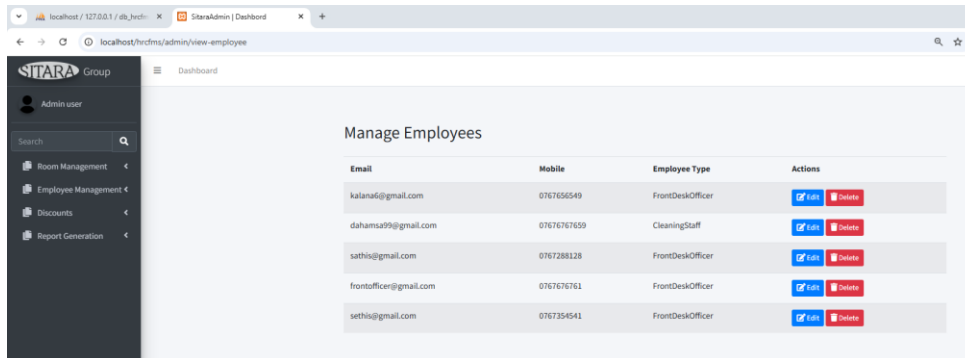


Figure4 31:Employee Management UI

```
public function updateEmployee(Request $request, $id): RedirectResponse
{
    $request->validate(rules: [
        'Fname' => 'required|string|max:255',
        'Lname' => 'required|string|max:255',
        'Mobile' => 'required|numeric',
        'City' => 'required|string|max:255'
    ]);

    $employee = Employee::findOrFail(id: $id);
    $user = User::where(column: 'email', operator: $employee->email)->first();

    // Update Employee
    $employee->update(attributes: [
        'Fname' => $request->Fname,
        'Lname' => $request->Lname,
        'Mobile' => $request->Mobile,
        'City' => $request->City,
        'name' => $request->Fname . ' ' . $request->Lname
    ]);

    // Update corresponding User
    if ($user) {
        $user->update(attributes: [
            'Fname' => $request->Fname,
            'Lname' => $request->Lname,
            'Mobile' => $request->Mobile,
            'City' => $request->City,
            'name' => $request->Fname . ' ' . $request->Lname
        ]);
    }

    return redirect()->route(route: 'admin.viewemployee')->with(key: 'success', value: 'Employee updated successfully!');
}

1 reference | 0 overrides
public function deleteEmployee($id): RedirectResponse
{
    $employee = Employee::findOrFail(id: $id);
    $user = User::where(column: 'email', operator: $employee->email)->first();

    if ($user) {
        $user->delete();
    }
    $employee->delete();

    return redirect()->route(route: 'admin.viewemployee')->with(key: 'success', value: 'Employee deleted successfully!');
}
```

Figure4 32:Employee Management -code

4.6.10. Room Management

Room ID	Room Name	Adults	Children	Price/Night	Actions
1	Sea View Room/Bed	3	2	\$38,000.00	Edit Delete
2	Sun Shine Room	2	1	\$22,500.00	Edit Delete
3	Family Hub Room	3	2	\$36,300.00	Edit Delete
4	SunGold Room	2	0	\$87,000.00	Edit Delete
5	Royal Orchid Suite/View	3	1	\$26,000.00	Edit Delete
6	Garden Retreat/View	3	2	\$53,700.00	Edit Delete
7	Palm Paradise Room	2	1	\$12,000.00	Edit Delete
8	Amber Glow Suite	2	1	\$52,500.00	Edit Delete
9	Coast Reef Suite	2	2	\$15,000.00	Edit Delete
10	FrontLake Room	2	1	\$41,000.00	Edit Delete

Figure4 33:Room Management UI

```
1 reference | 0 overrides
public function storeusers(Request $request): RedirectResponse
{
    $request->validate(rules: [
        'Fname' => 'required|string|max:255',
        'Lname' => 'required|string|max:255',
        'Email' => 'required|email|max:255|unique:users',
        'Password' => 'required',
    ]);
    // Create the User
    $user = User::create(attributes: [
        'Fname' => $request->input(key: 'Fname'),
        'Lname' => $request->input(key: 'Lname'),
        'DOB' => $request->input(key: 'DOB'),
        'email' => $request->input(key: 'Email'),
        'City' => $request->input(key: 'City'),
        'user_type' => $request->input(key: 'emp_type'),
        'name' => $request->input(key: 'Fname') . ' ' . $request->input(key: 'Lname'), // Full name
        'password' => Hash::make(value: $request->input(key: 'Password')),
        'Mobile' => $request->input(key: 'Mobile'),
    ]);
    // Create the Employee
    employee::create(attributes: [
        'Fname' => $request->input(key: 'Fname'),
        'Lname' => $request->input(key: 'Lname'),
        'DOB' => $request->input(key: 'DOB'),
        'email' => $request->input(key: 'Email'),
        'City' => $request->input(key: 'City'),
        'user_type' => $request->input(key: 'emp_type'),
        'name' => $request->input(key: 'Fname') . ' ' . $request->input(key: 'Lname'), // Full name
        'password' => Hash::make(value: $request->input(key: 'Password')),
        'Mobile' => $request->input(key: 'Mobile'),
    ]);
    //dd($request);
    return redirect()->route(route: 'admin.employee')->with(key: 'success', value: 'Employee added successfully!');
}

1 reference | 0 overrides
public function viewEmployee(): View
{
    $employees = Employee::select(columns: 'id','email', 'Mobile','user_type')->get();
    return view(view: 'admin.viewemployee', data: compact(var_name: 'employees'));
}

1 reference | 0 overrides
public function editEmployee($id): View
{
    $employee = Employee::findOrFail(id: $id);
    return view(view: 'admin.editemployee', data: compact(var_name: 'employee'));
}
```

Figure4 34:Room Management-code

4.6.11. Report Generation

Report generation includes reports on room availability, guest satisfaction, guest complaints, reservations, and transaction history. This feature facilitates the identification of trends, performance monitoring, and quality service improvement.

Hotel Room Availability Report

Hotel Room Availability Report					
Date Range: 2025-06-17 to 2025-06-24					
Generated At: 2025-06-22 01:39:55					
Room ID	Room Name	Adults Capacity	Children Capacity	Price/Night	Status
3	Family Hub Room	3	2	\$38,300.00	Available
4	SunGold Room	2	0	\$37,000.00	Available
5	Royal Orchid SuiteNew	3	1	\$26,000.00	Available
6	Garden RetreatNew	3	2	\$33,700.00	Available
7	Palm Paradise Room	2	1	\$12,000.00	Available
8	Amber Glow Suite	2	1	\$32,500.00	Available
9	Coral Reef Suite	2	2	\$12,500.00	Available
10	FrontLake Room	2	1	\$47,000.00	Available

Confidential Report - Laravel
Generated by Admin Portal

Figure4 35:Room Availability Report

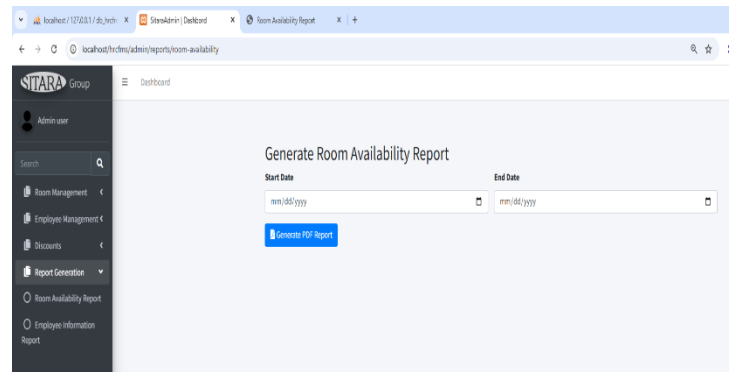


Figure4 36:Room Availability UI

```
public function generateAvailabilityReport(Request $request): Response
{
    $validated = $request->validate(rules: [
        'start_date' => 'required|date',
        'end_date' => 'required|date|after_or_equal:start_date'
    ]);

    $startDate = Carbon::parse(time: $validated['start_date']);
    $endDate = Carbon::parse(time: $validated['end_date']);

    $availableRooms = hotel_room::whereDoesntHave(relation: 'reservations', callback: function ($query) use ($startDate, $endDate): void {
        $query->where(function ($q) use ($startDate, $endDate): void {
            $q->where('check_out_date', '>', $startDate)
            ->where('check_in_date', '<', $endDate);
        });
    })->get();

    $pdf = PDF::loadView(view: 'admin.reports-availabilitypdf', data: [
        'startDate' => $startDate->format(format: 'Y-m-d'),
        'endDate' => $endDate->format(format: 'Y-m-d'),
        'availableRooms' => $availableRooms,
        'generatedAt' => now()->format(format: 'Y-m-d H:i:s')
    ]);

    return $pdf->download(filename: 'admin.reports-availabilitypdf'.now()->format(format: 'YmdHis').'.pdf');
```

Figure4 37:Room Availability Report -code

Employee Information Report

Employee Information Report						
All Employees						
First Name	Last Name	Email	Mobile	City	Date of Birth	Type
kalanaedit	sathsara	kalana6@gmail.com	0767656549	chilawedit	18/03/2025	FrontDesk
Dahamsa	Dewangi	dahamsa99@gmail.com	07676767659	chilaw	22/01/2025	CleaningSt
Sathis		sathis@gmail.com	0767288128	welimada	20/03/2025	FrontDesk
dehani		frontofficer@gmail.com	0767676761	Chilaw	08/05/2024	FrontDesk
sethis		sethis@gmail.com	0767354541	chilaw	13/03/2025	FrontDesk

Generated at: 2025-06-22 01:48:35
Total Employees: 6

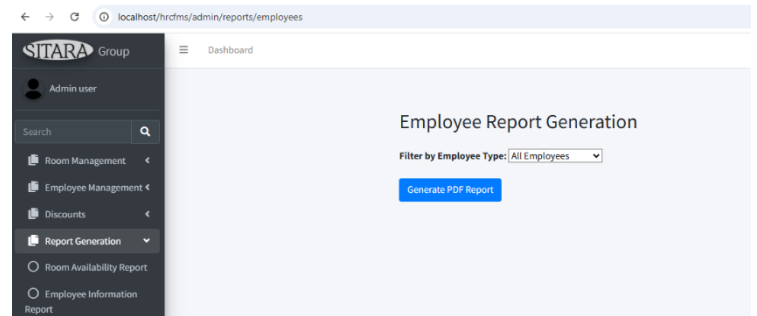


Figure4 39:Employee Information Report

Figure4 38:Employee Information UI

```

public function showEmployeeReportForm(): View
{
    return view(view: 'admin.reports-employeeform');
}

1 reference | 0 overrides
public function generateEmployeeReport(Request $request): Response
{
    $validated = $request->validate(rules: [
        'emp_type' => 'required|in:all,Admin,FrontDeskOfficer,MarketingStaff,CleaningStaff'
    ]);

    $empType = $validated['emp_type'];

    $employees = Employee::when(value: $empType !== 'all', callback: function(Builder $query) use ($empType): Builder {
        return $query->where(column: 'user_type', operator: $empType);
    })->get();

    $pdf = PDF::loadView(view: 'admin.reports-employeeepdf', data: [
        'employees' => $employees,
        'empType' => $empType,
        'generatedAt' => now()->format(format: 'Y-m-d H:i:s')
    ]);

    return $pdf->download(filename: 'admin.reports-employeeepdf'.now()->format(format: 'YmdHis').'.pdf');
}

```

Figure4 40:Employee Information Report-code

4.6.13. Reservation Management

The Front Desk User has the capability to mark the check-in and check-out times of guests. This functionality ensures accurate tracking of guest stays, facilitates efficient room management.

Reservation Management					
Room Number	Guest Name	Check-in Date	Check-out Date	Status	Actions
Sun Shine Room	sesa sasi	Jan 21, 2026 00:00	Jan 22, 2026 00:00	Checked In	Mark Check-out
Sun Shine Room	sathis sehanka	Jan 20, 2026 00:00	Jan 21, 2026 00:00	Checked Out	
Sea View RoomEdited	Sethas sehanka	Jan 15, 2026 00:00	Feb 16, 2026 00:00	Checked Out	
Sea View RoomEdited	neha sehansa	Jan 14, 2026 00:00	Jan 15, 2026 00:00	Upcoming	Mark Check-in

Figure4 42:Reservation Management UI

```
reference | overrides
public function viewreservation(): View
{
    $reservations = hotel_room_reservations::with(relations: ['room', 'guest'])
        ->orderBy(column: 'check_in_date', direction: 'desc')
        ->get();

    return view(view: 'frontdeskofficer.viewreservation', data: compact(var_name: 'reservations'));
}

1 reference | 0 overrides
public function checkIn(hotel_room_reservations $reservation): RedirectResponse
{
    $reservation->update(attributes: [
        'is_checked_in' => true,
        'actual_check_in' => now()
    ]);

    return back()->with(key: 'success', value: 'Check-in marked successfully');

    //dd($reservation);
}

1 reference | 0 overrides
public function checkOut(hotel_room_reservations $reservation): RedirectResponse
{
    $reservation->update(attributes: [
        'is_checked_out' => true,
        'actual_check_out' => now()
    ]);

    return back()->with(key: 'success', value: 'Check-out marked successfully');
}
```

Figure4 41:Reservation Management-code

4.6.14. House Keeping Ticket Management

The Housekeeping Ticket Management feature allows front desk officers to create and assign housekeeping tasks.

Housekeeping Tasks					
Room #	Task Type	Priority	Description	Status	Actions
Sunshine Room	Cleaning	Low	Wet Floors	In Progress	In Progress ▼
Sunshine Room	Cleaning	Low	AC Repairing	In Progress	In Progress ▼
SunRoom	Maintenance	Medium	AC Repairing	Completed	Completed ▼
Sunshine Room	Cleaning	High	Need to do it immediately	Completed	Completed ▼

Figure4 44:Housekeeping task Management UI

```
1 reference | 0 overrides
public function viewTasks(): View
{
    $tasks = task::orderBy(column: 'created_at', direction: 'desc')->get();
    return view(view: 'housekeepingstaff.task-results', data: compact(var_name: 'tasks'));
}

1 reference | 0 overrides
public function updateStatus(Request $request, Task $task): RedirectResponse
{
    $request->validate(rules: [
        'status' => 'required|in:Pending,In Progress,Completed'
    ]);

    $task->update(attributes: [
        'status' => $request->status
    ]);

    return back()->with(key: 'success', value: 'Task status updated successfully!');
}
```

Figure4 43:Housekeeping task Management-code

4.7. Code Structure

The MVC architecture in the Guest Registration functionality separates concerns into three components. The Model (`guest.php`) defines guest attributes and relationships. The View (`guest.blade.php`) presents a user-friendly registration form with validation messages. The Controller (store method) handles form validation, user creation, and guest record storage. This structure improves maintainability, scalability, and code reusability in the application.

Guest Registration

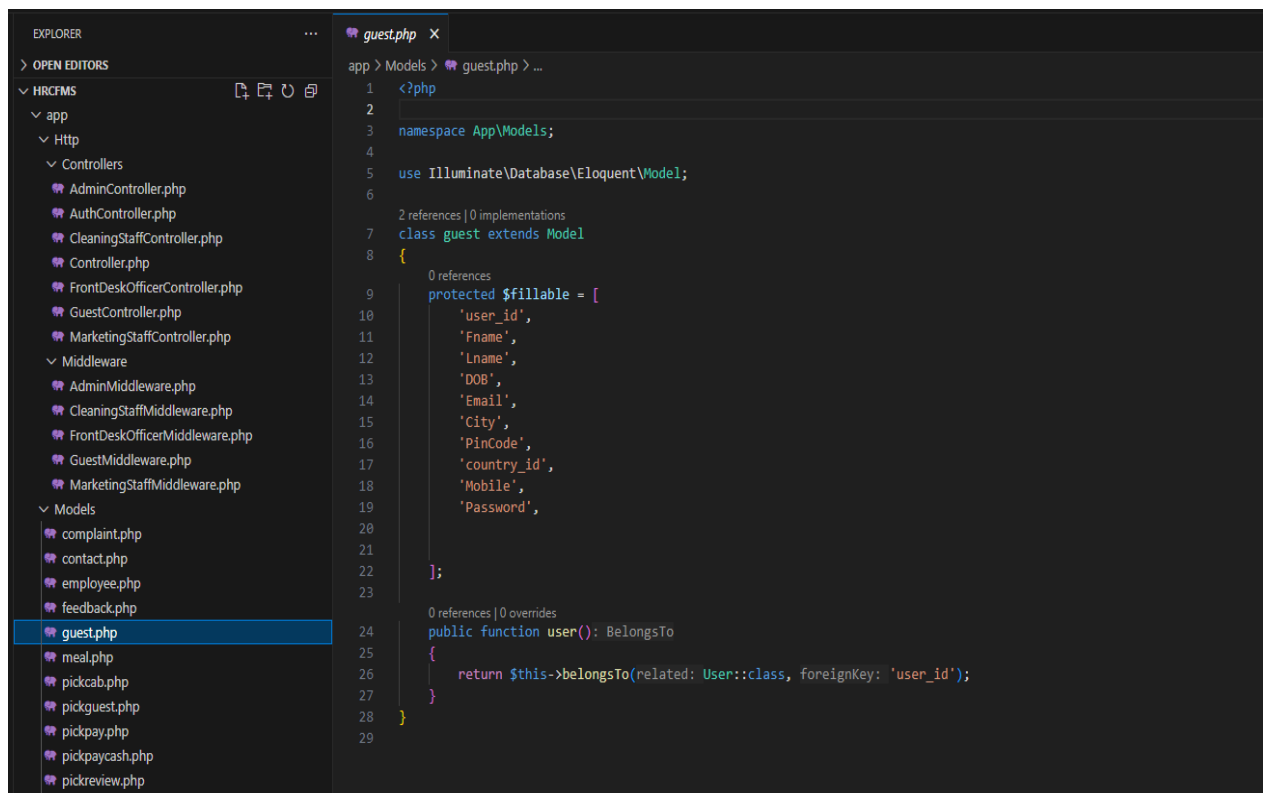


Figure4 45:Model_guest.php

EXPLORER

resources

views

auth

login.blade.php

frontdeskofficer

guest

about.blade.php

cab.blade.php

complaint.blade.php

contact.blade.php

dashboard.blade.php

deluxe.blade.php

facility.blade.php

feedback.blade.php

gallery.blade.php

meal.blade.php

pickcab.blade.php

pickguest.blade.php

pickown.blade.php

pickpay.blade.php

pickpaycash.blade.php

pickreview.blade.php

pickroom.blade.php

pickspa.blade.php

promotion.blade.php

register.blade.php

review.blade.php

roomreservation.blade.php

spa.blade.php

success.blade.php

successbooking.blade.php

suite.blade.php

superior.blade.php

term.blade.php

layouts

vendor

resources > views > guest > register.blade.php

2 @section(section: 'content')

42 <section class="aboutus-section spad" style="padding-top: 20px !important;">

43 <div class="container">

44 <div class="row">

45 <div class="col-lg-12">

46 <form method="post" action="{{route(name: 'guest.store')}}" class="row g-3 novalidate">

47 @csrf

48

49 <div class="col-md-6">

50 <label for="Fname" class="form-label">First Name</label>

51 <input type="text" class="form-control" id="Fname" name="Fname" placeholder="First Name">

52 @error('Fname')

53 {{ \$message }}

54 @enderror

55 </div>

56

57 <div class="col-md-6">

58 <label for="Lname" class="form-label">Last Name</label>

59 <input type="text" class="form-control" id="Lname" name="Lname" placeholder="Last Name">

60 @error('Lname')

61 {{ \$message }}

62 @enderror

63 </div>

64

65 <div class="col-md-6">

66 <label for="DOB" class="form-label">Date of Birth</label>

67 <input type="date" class="form-control" id="DOB" name="DOB">

68 </div>

69

70 <div class="col-md-6">

71 <label for="Email" class="form-label">Email</label>

72 <input type="email" class="form-control" id="Email" name="Email" placeholder="Email">

73 @error('Email')

74 {{ \$message }}

75 @enderror

76 </div>

77

78 <div class="col-md-6">

79 <label for="Mobile" class="form-label">Mobile Number</label>

80 <input type="text" class="form-control" id="Mobile" name="Mobile" placeholder="Mobile Number">

81 @error('Mobile')

82 {{ \$message }}

2 @section(section: 'content')

42 <section class="aboutus-section spad" style="padding-top: 20px !important;">

43 <div class="container">

44 <div class="row">

45 <div class="col-lg-12">

104 @error('Password')

105 {{ \$message }}

106 @enderror

107 </div>

108

109 <!-- Terms and Conditions Section -->

110 <div class="col-12">

111 <div class="form-check">

112 <input class="form-check-input" type="checkbox" id="terms" name="terms" required>

113 <label class="form-check-label" for="terms">

114 I have read and agree to the

115

116 Terms and Conditions & the Privacy policy

117

118

119 </label>

120 </div>

121 </div>

122 @error('terms')

123 {{ \$message }}

124 @enderror

125 </div>

126

127 <div class="col-12 d-flex justify-content-center mt-4">

128 <button type="submit" class="btn btn-primary"

129 style="background-color: #dfa974; border-color: #dfa974;">Register</button>

130 </div>

131 </div>

132 </div>

133 </div>

134 </div>

135 </section>

136 <!-- About Us Section End -->

137

138 @endsection

Figure4 46:View_Register.blade.php

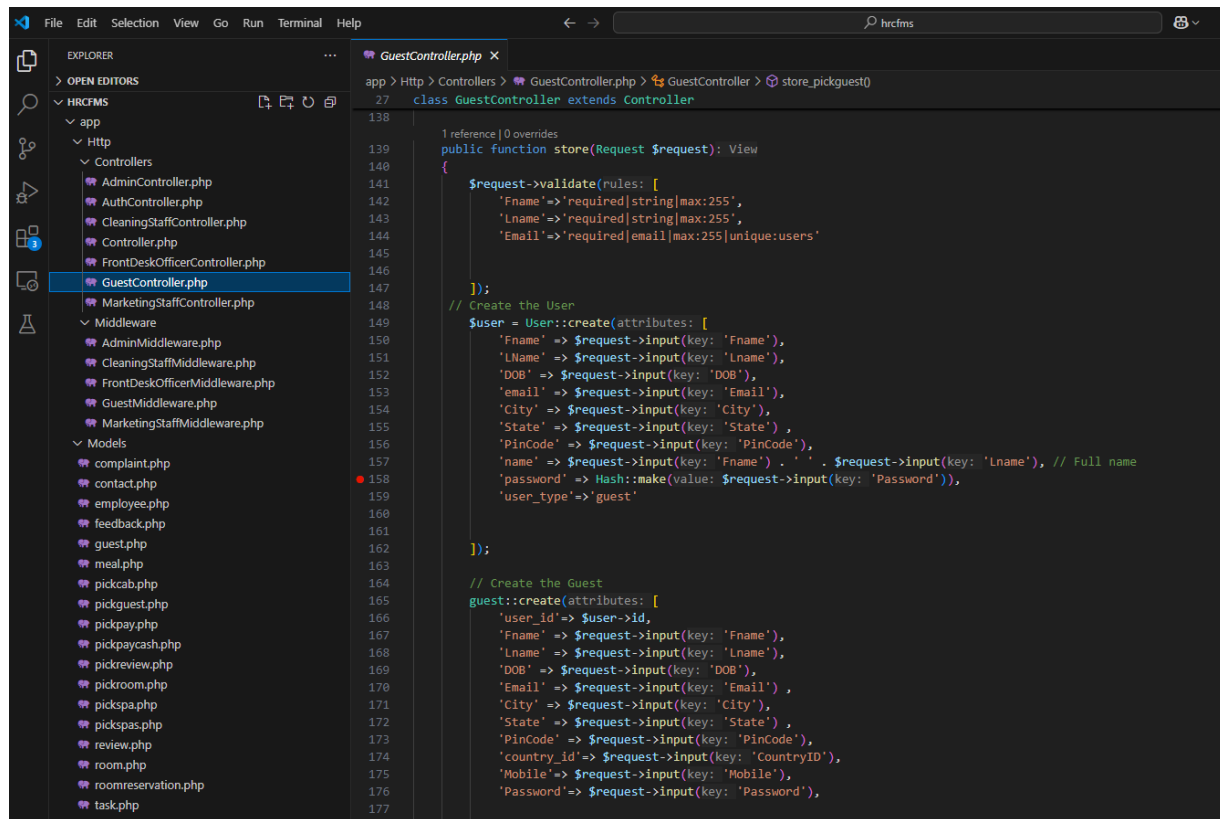


Figure4 47:Controller_guestcontroller.php

4.8. Summary

The implementation phase translates the system's design into functional code, ensuring the Sitara Hotel Reservation & Customer Feedback Management System operates efficiently. Technologies such as PHP, Python, MySQL, Bootstrap, and the Laravel framework were used in development. The system includes key modules like user management, room reservations, and customer feedback, following the MVC architecture for scalability. Middleware facilitates smooth communication between components, and user interfaces were designed for guests, staff, and administrators to enhance operational efficiency.

Chapter 5 – Testing and Evaluation

5.1 Introduction

The "Testing and Evaluation" chapter focuses on verifying the functionality, performance, and usability of the Hotel Reservation and Customer Feedback Management System. The evaluation process helps identify potential issues and optimization opportunities. It also highlights the system's performance under different conditions. The results from testing are used to improve the system, ensuring it meets user requirements and objectives.

5.2 Testing types

Unit testing is performed to verify the correctness of individual components or modules of the system. Each function, such as room reservation, feedback submission, and user authentication, is tested in isolation to identify and fix errors at an early stage.

Integration testing ensures that different modules of the system, such as reservation management, user authentication, and payment processing, function correctly when combined. This testing verifies seamless communication between components, ensuring that data flows correctly across different system layers.

Functional testing is conducted to validate that the system meets all specified requirements. Test cases cover critical functions, including room availability checking, booking confirmation, and feedback management. This testing type ensures that each feature works as intended and meets both user expectations and business logic.

Usability testing focuses on assessing the user experience and ease of navigation within the system. Real users interact with the platform to identify usability issues, ensuring that booking procedures, feedback submission, and system interactions are intuitive and user-friendly. The results help improve the system's overall accessibility and efficiency.

Security testing is crucial to safeguarding user data and protecting the system from vulnerabilities. Penetration testing is conducted to identify potential security flaws, such as unauthorized access and data breaches. Encryption methods, secure authentication protocols, and data protection mechanisms are evaluated to ensure compliance with security standards.

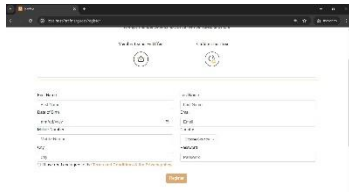
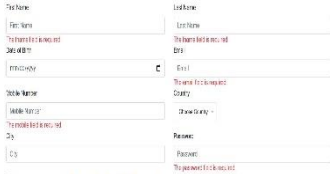
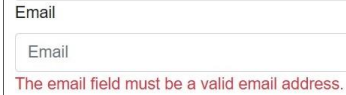
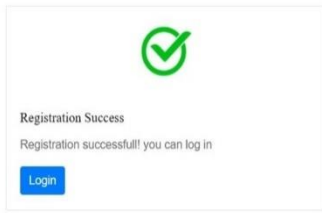
Retesting is a critical testing process conducted to confirm that previously identified defects have been effectively resolved. Once an issue is addressed, the same test cases that initially detected the defect are re-executed to ensure that the problem no longer persists. Regression testing ensures that bug fixes or modifications do not affect the system's existing functionalities.

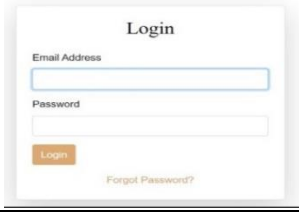
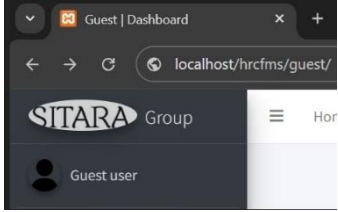
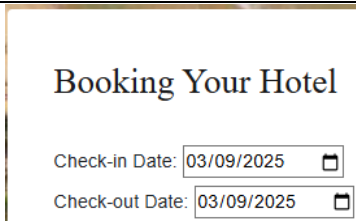
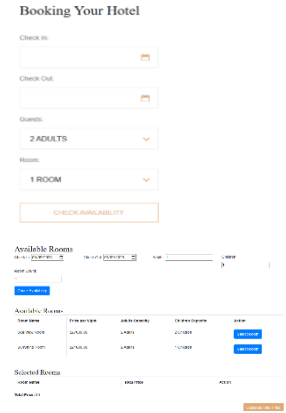
Acceptance testing is performed to validate whether the system meets business requirements and user expectations. End-users and stakeholders conduct real-world tests to assess system functionality, identifying any gaps that need to be addressed before deployment.

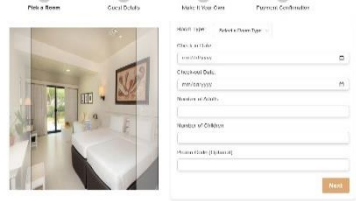
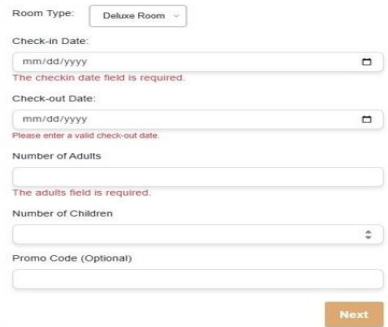
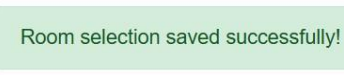
5.3 Test Cases and Test Case Design Techniques

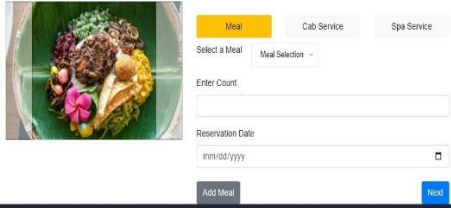
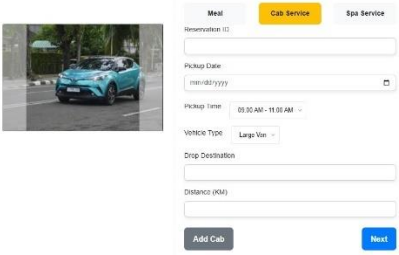
Test case design techniques ensure the reliability and accuracy of the Hotel Reservation and Customer Feedback Management System at Sitara Hotel Weligama. Equivalence partitioning and boundary value analysis help validate input ranges, while decision table testing ensures correct responses to various conditions. State transition testing verifies changes in booking statuses and use case testing covers end-to-end scenarios like reservations and feedback submission. Additionally, error guessing helps identify potential issues by simulating unexpected user actions.

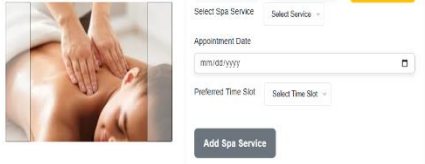
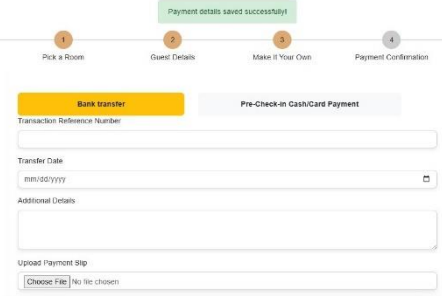
Test Cases

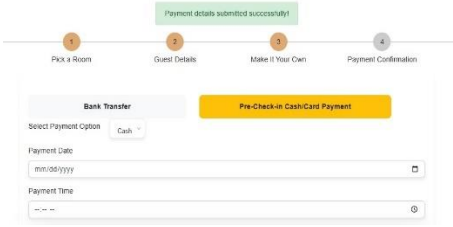
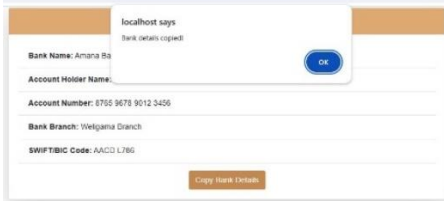
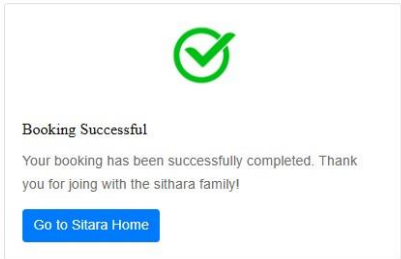
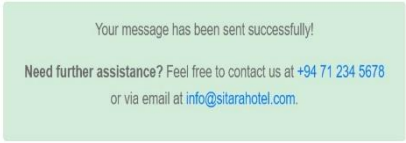
Test Case ID	Test Case Name	Test Procedure	Expected Result	Actual Result	Priority
SH_001	Verify User Registration page UI	1.Click on the Register link from Home Page 2.Verify the UI	User should display the Register page		Medium
SH_002	Verify the required validations in the Registration page UI	1.Navigate to the Registration page 2.Click on Register button without entering the required details	User should display the validation messages for the required fields		Medium
SH_003	Verify user not able to submit the registration form with Invalid Email Address	1.User navigates to the Registration page 2.Enter the Invalid format email 3. Click on Register button and verify	User should display the validation message as "The email field must be a valid email address"		High
SH_004	Verify guest can register to the portal	1.Verify user navigate to the Registration page 2.Enter the valid user details 3.Click on Register button and verify	1.User should be registered successfully 2.Should display the success message		High

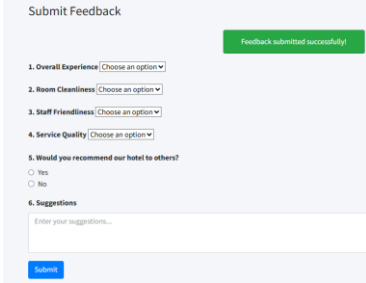
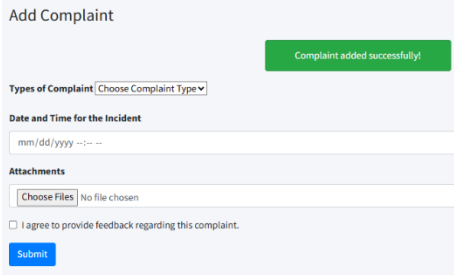
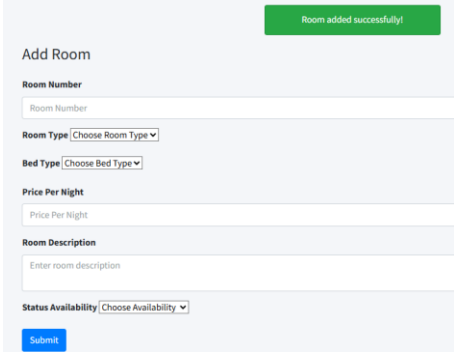
SH_005	Verify the Login page UI	1.User register with the portal 1.Click on the login link and verify	User should display the Login Page with Email and password		Medium
SH_006	Verify user not able to login to the portal by entering the Invalid Email and Password	1.Navigate to the Login page 2.Enter the Invalid Email and Password 3.Click on the Login button	User should display the validation message as "Invalid Email and Password"		High
SH_007	Verify user can login to the portal by entering valid Email and Password	1.Navigate to the Login page 2.Enter the Invalid Email and Password 3.Click on the Login button	User should be able to login to the dashboard		High
SH_008	Verify guest not able to select the past date for the Check-in Date and Check-out Date.	1.Load the Home page 2.Click on the calendar option for the Check-in Date and Check-out date	User should not able select the pass date for the check in and check out date		
SH_009	Verify guests can check the room availability by applying the searching criteria.	1.Load the Home page 2.Click on check availability button by adding applicable date, adults, child and room count	Should display only the available rooms		

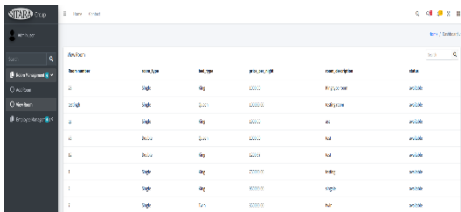
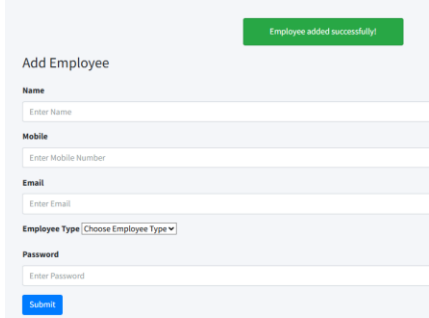
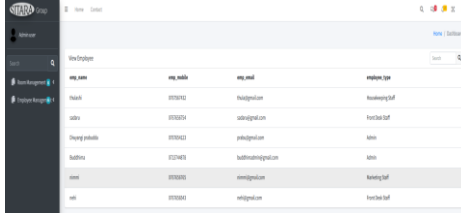
SH_010	Verify User can Start the Booking process by clicking on the "Booking Now " button	1.Click on "Booking Now" button from Home page	User should display the Booking form with Four steps. (Pick a Room, Guest Details, Make It Your Own, Payment Confirmation)		High
SH_011	Verify the validation messages in the "Pick a Room" page	1.Click on "Booking Now" button from Home page 2.Select the Room type 3.Click on the Next button without filling the other required fields	User should display the required messages for check-in date, Check-out date and adults field		Medium
SH_012	Verify user can be able to successfully submit the room selection	1.Click on "Booking Now" button from Home page 2.Enter the valid data for the Pick a Room page 3.Click on the Next button	1.User should be able to submit the form 2.User should navigate to the Guest Detail page		Medium

SH_01 3	Verify user can be able to submit the guest detail page	1.Navigate to the Guest Detail page 2.Enter the valid guest details and click on the next button	1.User should be able to submit the form 2.User should navigate to the Make it Your Own page		Medium
SH_01 4	Verify user can navigate to the Meal reservation page	1.Guest User click on the "Booking Now" 2.Submit the room and guest selection 3.Click on Next button	User should display the meal selection page		Medium
SH_01 5	Verify user can request a Meal reservation	1.Navigate to the Meals page 2.Filling the Meals required form 3.Click on the Add Meal button	1.User should be able to make a Meal reservation 2.Should display the success message		Medium
SH_01 6	Verify user can navigate to the Cab service selection page	1.Guest User click on the "Booking Now" 2.Submit the room and guest selection 3.Click on Next button 4.Click on the Cab Service	User should display the Cab service page		Medium

SH_017	Verify user can request a Cab Service reservation	1.Navigate to the Cab Service request page 2.Fill the cab service form 3.Click on the Add Cab service button	1.User should be able to request the cab service 2.Should display the success message		Medium
SH_018	Verify user can navigate to the Spa service selection page	1.Guest User click on the "Booking Now" 2.Submit the room and guest selection 3.Click on Next button 4.Click on the Spa Service	User should display the Spa services selection page		Medium
SH_019	Verify user can request a Spa service reservation	1.Navigate to the Spa Service request page 2.Fill the Spa service form 3.Click on the Add spa Service button	User should be able to request the spa service Should display the success message		Medium
SH_020	Verify guest can submit the application by selecting the Bank transfer payment option	1.Guest navigate to the Payment details page 2.Select the Bank Transfer option 3.Fill the Form and click on Add Bank Transfer button	User should be able to submit the bank transfer form. Should display a success message		High

SH_02 1	Verify guest can submit the application by selecting the Pre-Check-in Cash, Card option	1.Guest navigate to the Payment details page 2.Select the Pre-Check-in Cash, Card option 3.Fill the Form and submit	User should be able to submit the payment application		High
SH_02 2	Verify Guest can copy the bank details	1.Guest navigate to the Payment details page 2.Select the Bank Transfer option 3.Click on the copy bank details button	User can copy the bank details		Low
SH_02 3	Verify guest can submit the form by selecting room reservation with additional services	1.Guest click on the Booking process button 2. Fill all four steps with the Additional request 3.Click on Confirm booking	1.User should be able to submit the booking form 2.Success message should be displayed		High
SH_02 4	Verify guest user can submit the contact Us form	1.Navigate to the Contact Us page 2.Enter the Details 3.Click on the Send Message button	1.User should be able to submit the contact us form 2.Success Message should be displayed		Medium

SH_02 5	Verify guest user can add a review	1.Navigate to the Add Review page 2.Enter the Details 3.Click on Add Review button	1.User should be able to add a review 2.Success Message should be displayed		Medium
SH_02 6	Verify guest can submit the Feedback form	1. Guest navigate to the profile UI 2.Click on the Feedback 3.Fill the Feedback form and click on the submit	User should be able to submit the feedback form Should display the success message		Medium
SH_02 7	Verify guest can add a complain	1. Guest navigate to the profile UI 2.Click on the Complain 3.Fill the Complain form and click on the submit	User should be able to submit the complaint form 2.should display the success message		Medium
SH_02 8	Verify Admin user can be able to add a Room	1.Admin login to the portal select Room Management 3.Click on Add Room 4.Fill the form with valid room detail 5.Click on Add Room button	Admin should be able to Add a new room		Medium

SH_02 9	Verify Admin user can view the already added room	1.Admin login to the portal by adding a room 2.Click on the Room Management 3. Click on View Room	Admin should be able to view the added room details		Medium
SH_03 0	Verify Admin user can be able to add an Employee	1.Admin login to the portal 2.Click on the Room Management 3. Click on Add Room 4.Fill the form with valid room detail 5.Click on Add Employee button	Admin should be able to add a new employee to the system		High
SH_03 1	Verify Admin user can view the already added Employee details	1.Admin login to the portal by adding an employee 2.Click on the Employee Management 3. Click on View Employee	Admin should be able to view the already added Employee details		High

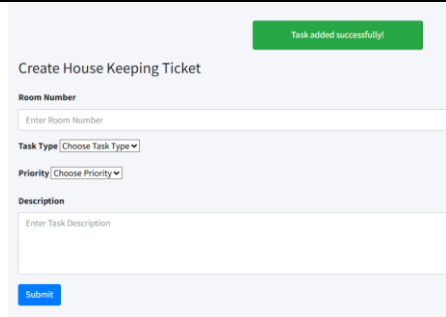
SH_03 2	Verify Front Desk Officer can add a ticket for the House Keeping task	1.Front Desk Officer Login to the portal 2.Click on the Add House Keeping Ticket navigation option 3.Fill the form with valid data 4.Click on the Add House keeping ticket button	Front desk officer should be able to add a House keeping ticket		Medium
------------	---	--	---	--	--------

Table4 2:Testcase writing table

5.4 Test Results



Runs (Summary) 3/9/2025

Project: Sitara Hotel Reservation and Customer Feedback Management System
By WANNIYA SEWWANDI, 3/9/2025 3:56 AM

Run: SH - Test Run 3/9/2025

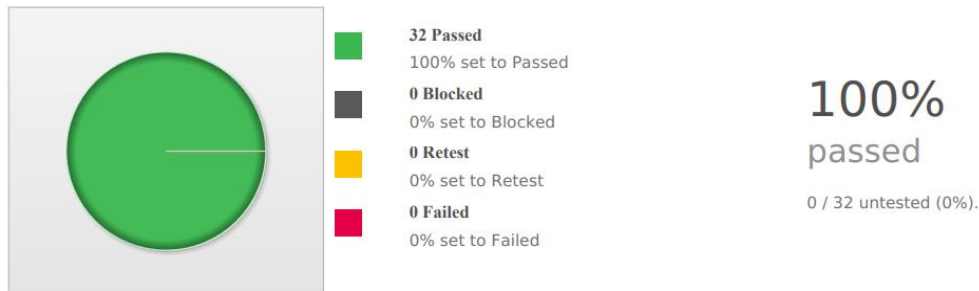


Figure5 1:Test summary report using TestRail

Tests & Results

Test Cases (32)

ID	Title	Status
T673	[SH_001] Verify User Registration page UI	Passed
T674	[SH_002] Verify the required validations in the Registration page UI	Passed
T675	[SH_003] Verify user not able to submit the registration form with Invalid Email Address	Passed
T676	[SH_004] Verify guest can register to the portal	Passed
T677	[SH_005] Verify the Login page UI	Passed
T678	[SH_006] Verify user not able to login to the portal by entering the Invalid Email and Password	Passed
T679	[SH_007] Verify user can login to the portal by entering valid Email and Password	Passed
T703	[SH_008] Verify guest not able to select the past date for the Check-in Date and Check-out Date.	Passed
T704	[SH_009] Verify guests can check the room availability by applying the searching criteria.	Passed
T680	[SH_010] Verify User can Start the Booking process by clicking on the "Booking Now " button	Passed
T681	[SH_011] Verify the validation messages in the "Pick a Room" page	Passed
T682	[SH_012] Verify user can be able to successfully submit the room selection	Passed
T683	[SH_013] Verify user can be able to able to submit the guest detail page	Passed
T684	[SH_014] Verify user can navigate to the Meal reservation page	Passed
T685	[SH_015] Verify user can request a Meal reservation	Passed
T686	[SH_016] Verify user can navigate to the Cab service selection page	Passed
T687	[SH_017] Verify user can request a Cab Service reservation	Passed
T688	[SH_018] Verify user can navigate to the Spa service selection page	Passed
T689	[SH_019] Verify user can request a Spa service reservation	Passed
T690	[SH_020] Verify guest can submit the application by selecting the Bank transfer payment option	Passed
T691	[SH_021] Verify guest can submit the application by selecting the Pre-Check-in Cash, Card option	Passed
T692	[SH_022] Verify Guest can copy the bank details	Passed
T693	[SH_023] Verify guest can submit the form by selecting room reservation with additional services	Passed
T694	[SH_024] Verify guest user can submit the contact Us form	Passed

T695	[SH_025] Verify guest user can add a review	Passed
T696	[SH_026] Verify guest can submit the Feedback form	Passed
T697	[SH_027] Verify guest can add a complain	Passed
T698	[SH_028] Verify Admin user can be able to add a Room	Passed
T699	[SH_029] Verify Admin user can view the already added room	Passed
T700	[SH_030] Verify Admin user can be able to add an Employee	Passed
T701	[SH_031] Verify Admin user can view the already added Employee details	Passed
T702	[SH_032] Verify Front Desk Officer can add a ticket for the House Keeping task	Passed

Figure5 2:Test Execution results using the testrail

Figure5 3:Test Execution results using the testrail

5.5 User Evaluation

The developed Hotel Reservation and Customer Feedback Management System for Sitara Hotel was evaluated by users to determine its effectiveness in meeting the specified requirements. The user evaluation process followed User Acceptance Testing (UAT), which involved a structured questionnaire distributed via a Google Form survey. The primary actors involved in the evaluation process were selected from key user groups, including hotel staff and management. The evaluation aimed to assess various aspects of the system, including its functionality, usability, and overall user experience, ensuring it aligned with the operational needs of Sitara Hotel.

The evaluation was conducted by selecting representatives from the hotel's different departments, ensuring a comprehensive review of all functions. The participants included:

Guest-4

Front Desk Officers-3

Admin-2

Marketing-2

House Keeping-2

SITARA Hotel Reservation & Customer Feedback System – User Evaluation Form

Dear User/Hotel Staff,

Thank you for using the **SITARA Hotel Reservation & Customer Feedback System**. We appreciate your time in providing valuable feedback to help us improve the system. Please rate the system's usability and features based on your experience. Your responses will be kept confidential.

buddhipraba94@gmail.com

Switch account

Not shared

* Indicates required question

Your Role (Select one) *

☐ Admin

☐ Front desk officer

☐ Marketing officer

☐ House keeping staff

☐ Guest

How frequently do you use the system? *

☐ Daily

☐ Weekly

☐ Occasionally

☐ Rarely

Usability & User Experience

How easy was it to navigate through the system? *

☐ Very Easy

☐ Easy

☐ Neutral

☐ Difficult

☐ Very Difficult

How intuitive were the system's features? *

☐ Very Intuitive

☐ Somewhat Intuitive

☐ Neutral

☐ Somewhat Confusing

☐ Very Confusing

How would you rate the overall user interface (UI) design? *

Figure 5-2: User Evaluation Form -Part I

Figure 5-2: User Evaluation Form -Part I

System Performance & Reliability

How satisfied are you with the system's speed and performance? *

☐ Very Satisfied
☐ Satisfied
☐ Neutral
☐ Dissatisfied
☐ Very Dissatisfied

Did you encounter any technical issues while using the system? *

☐ No Issues
☐ Minor Issues (Did not affect work)
☐ Major Issues (Required IT Support)
☐ Frequent Issues (System disruptions)

Features & Functionality

How effective is the reservation system in managing hotel bookings? *

☐ Very Effective
☐ Effective
☐ Neutral
☐ Ineffective
☐ Very Ineffective

How useful is the customer feedback module in gathering guest opinions? *

☐ Very Useful
☐ Somewhat Useful
☐ Neutral
☐ Not Useful
☐ Not Used

Suggestions & Overall Satisfaction

What improvements would you suggest for the system? (Open-ended)

Your answer

Overall, how satisfied are you with the SITARA Hotel System? *

1 2 3 4 5
(Not Satisfied) ☐ ☐ ☐ ☐ ☐ (Highly Satisfied)

Figure 5-3: User Evaluation Form -Part II

Your Role (Select one)
13 responses

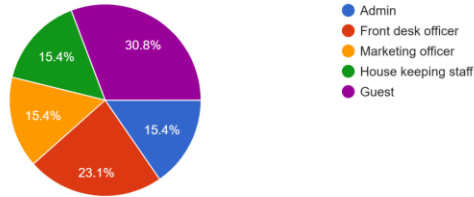


Figure 5-4: Survey Feedback -I

How frequently do you use the system?
13 responses

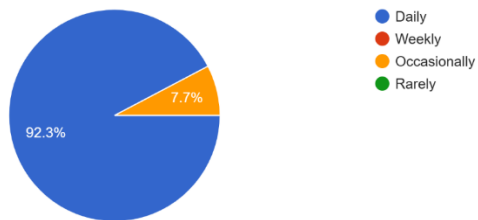


Figure 5-5: Survey Feedback II

How easy was it to navigate through the system?
13 responses

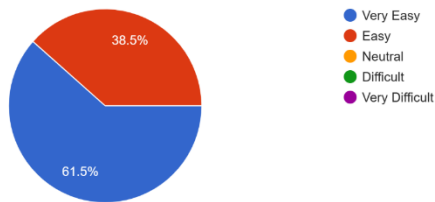


Figure 5-6: Survey Feedback III

FigureB 1:Room availability report
Figure 5-6: Survey Feedback III

How intuitive were the system's features?
13 responses

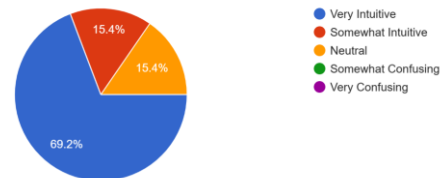


Figure 5-7: Survey Feedback IV

How would you rate the overall user interface (UI) design?
13 responses

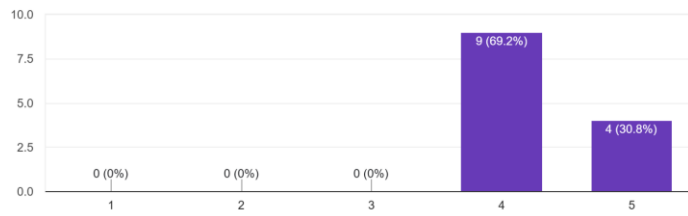


Figure 5-8: Survey Feedback V

How satisfied are you with the system's speed and performance?
13 responses

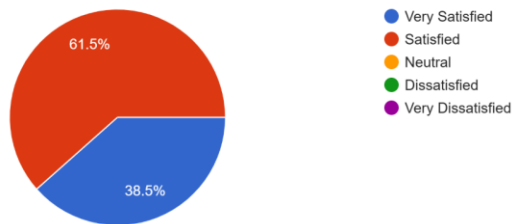


Figure 5-9: Survey Feedback VI

Did you encounter any technical issues while using the system?
13 responses

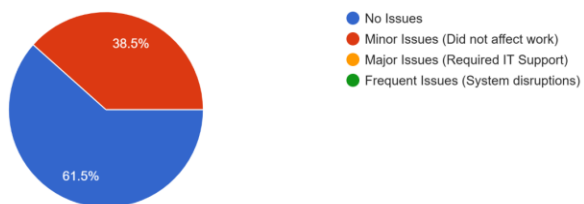


Figure 5-10: Survey Feedback VII

How effective is the reservation system in managing hotel bookings?
13 responses

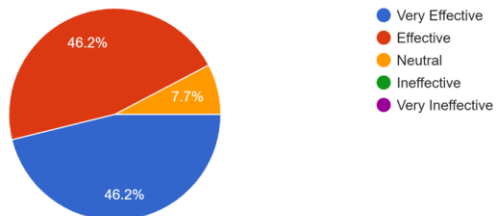


Figure 5-11: Survey Feedback VIII

How useful is the customer feedback module in gathering guest opinions?
13 responses

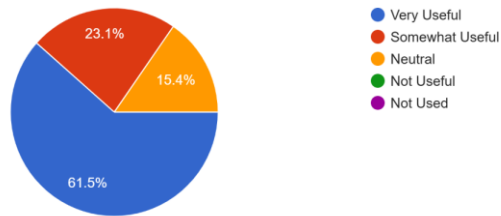


Figure 5-12: Survey Feedback VIII

Suggestions & Overall Satisfaction

What improvements would you suggest for the system? (Open-ended)

3 responses

Dynamic Pricing Model

Mobile Application Integration

Multi-Language Support

Figure 5-13: Survey Feedback IX

Overall, how satisfied are you with the SITARA Hotel System?

13 responses

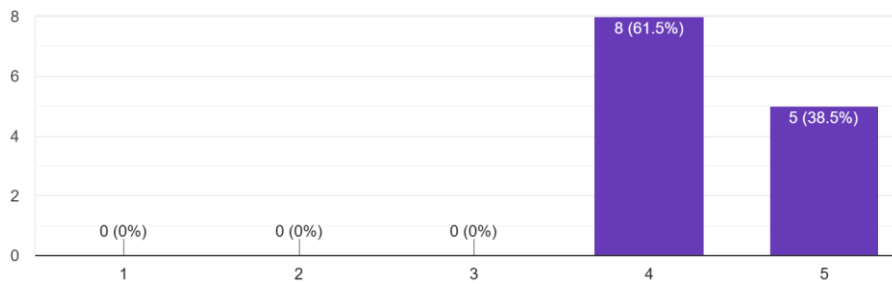


Figure 5-14: Survey Feedback X

The majority of users reported high satisfaction with the system, particularly in terms of its ease of navigation and intuitive design. A substantial portion (92.3%) of respondents used the system daily, and the user interface received favorable ratings, with 69.2% rating it positively. Furthermore, the reservation system was deemed effective, with 92.3% of users finding it beneficial for managing bookings. The customer feedback module also garnered high marks for its usefulness in gathering guest opinions, with 61.5% of participants considering it very useful. While the system generally performed well, users suggested several enhancements, such as the integration of a dynamic pricing model, mobile application support, and multi-language capabilities. Overall, the evaluation highlights the system's strengths in functionality and user-friendliness, while also providing valuable insights into potential areas for future improvements.

5.6 Summary

Testing and Evaluation chapter focuses on testing the functionality, performance, and usability of the Hotel Reservation and Customer Feedback Management System. It covers various testing types such as component, integration, functional, and security testing to ensure system reliability. Test cases are designed using techniques like boundary value analysis and decision table testing. User evaluation through User Acceptance Testing (UAT) shows high satisfaction with the system, especially its booking and feedback features. Suggestions for improvement include mobile app support and dynamic pricing.

Chapter 6 -Conclusion

This conclusion chapter provides a summary of the project's objectives, achievements, challenges faced, and its impact on Sitara Hotel's daily operations, with a focus on the benefits it brings to both guests and hotel staff.

6.1 Overview of the solution

The Hotel Reservation and Customer Feedback Management System developed for Sitara Hotel, Weligama, presents a comprehensive solution aimed at addressing the inherent limitations of the current hotel management practices. At present, Sitara Hotel relies heavily on third-party reservation platforms, which result in significant financial expenditures and limit the hotel's ability to cultivate and promote its unique brand identity through its own website. Moreover, the hotel staff's operational tasks, including the management of bookings and customer feedback, are predominantly carried out manually using paper-based processes, leading to inefficiencies, time consumption, and the potential for human error.

The newly implemented system serves to eliminate these inefficiencies by introducing an integrated, automated reservation platform. This platform enables guests to view real-time room availability, make direct reservations via the hotel's website, and receive immediate confirmation of their bookings. In addition to reducing the dependency on costly third-party reservation services, this solution empowers Sitara Hotel to establish and promote its online presence, thereby fostering the development of its unique brand. Furthermore, the system enhances the efficiency of staff operations by eliminating paper-based workflows, facilitating streamlined booking management, and incorporating a review summarization mechanism to check the lengthy reviews.

6.2 Lesson Learnt

Throughout the development and implementation of the Hotel Reservation and Customer Feedback Management System for Sitara Hotel, several key lessons were learned that have contributed to the overall success of the project. One of the primary lessons learned was the importance of thorough requirements gathering and stakeholder engagement early in the project.

Understanding the specific needs and challenges faced by hotel staff and management allowed for the design of a system that was both functional and user-friendly. Regular feedback from users during the development phase ensured that the system met the practical needs of the hotel.

The project offered significant learning opportunities in both backend and frontend technologies. The backend was developed using the Laravel framework with PHP, which facilitated the construction of a scalable system based on the Model-View-Controller (MVC) architecture. This approach ensured the efficient management of business logic, data processing, and overall system functionality. On the frontend, the system utilized Bootstrap and CSS to design a responsive and user-friendly interface, thereby ensuring smooth and seamless interaction between the users and the system.

The implementation of the system provided an opportunity to address business process flaws, streamlining operations that were previously managed manually. This experience allowed me to work through the entire System Development Life Cycle (SDLC), gaining valuable insights into the phases of system design, development, testing, and deployment.

6.3 Future Improvement

According to the findings from the user acceptance test conducted during the evaluation phase, the feedback from users was largely positive, indicating that the system effectively meets the operational needs and enhances the guest experience. However, several areas for future enhancement have been identified to further optimize the system and address evolving user needs. The following improvements are planned for the future development of the system.

Mobile Application Integration

The development of a mobile application to provide guests with the ability to make reservations, access real-time room availability, and submit feedback directly from their smartphones.

Dynamic Pricing Model

Implementation of a dynamic pricing algorithm to adjust room rates based on demand, seasonality, and other market factors, enhancing revenue management capabilities.

Sentiment Analysis Tool

Integration of sentiment analysis features to automatically assess guest reviews, providing real-time insights into customer satisfaction and identifying areas for service improvement.

Multi-Language Support

Expansion of the system to support multiple languages, allowing the hotel to cater to a broader international clientele and improve the accessibility of the system.

6.4 Summary

This chapter provides a summary of the project's objectives, achievements, challenges, and its impact on Sitara Hotel's operations. The developed Hotel Reservation and Customer Feedback Management System addresses inefficiencies by replacing third-party reservation platforms and manual processes, allowing for real-time booking management and streamlined operations. Key lessons learned include the importance of thorough requirements gathering, stakeholder engagement, and working through the entire System Development Life Cycle (SDLC). Future improvements, based on positive user feedback, include the integration of a mobile application, dynamic pricing model, sentiment analysis tool, and multi-language support to further enhance the system's functionality and accessibility.

References

- Agyei, M. and Owusu, A., 2022. Developing a smart hotel management system: An overview of modern technologies. *Journal of Hospitality Technology*, 15(2), pp. 150-163.
- Bardi, J.A., 2011. *Hotel front office management*. Hoboken, N.J.: John Wiley & Sons.
- Booch, G., 2017. *Unified Modeling Language User Guide*.
- Booking.com, n.d. Booking.com: Hotels, flights, car rentals, and more. Available at: <https://www.booking.com> [Accessed 1 Mar. 2025].
- Cockburn, A., 2001. *Writing effective use cases*. Boston: Addison-Wesley.
- Cloudbeds, n.d. Cloudbeds: Property management software for hotels. Available at: <https://www.cloudbeds.com> [Accessed 1 Mar. 2025].
- Cheng, W. and Lee, H., 2020. Building a hotel reservation system using modern web technologies. *Journal of Web Development*, 35(4), pp. 123-145.
- Enz, C.A., 2019. *Hospitality strategic management: Concepts and cases* (2nd ed.). Hoboken, N.J.: John Wiley & Sons.
- Figma, n.d. Collaborative Interface Design Tool. Available at: <https://www.figma.com> [Accessed 12 Dec. 2024].
- Fogg, B.J., 2003. *Persuasive technology: Using computers to change what we think and do*. San Francisco: Morgan Kaufmann.
- GeeksforGeeks, 2024. MVC Architecture System Design. Available at: <https://www.geeksforgeeks.org/mvc-architecture-system-design/>.
- Hotels.com, n.d. Hotels.com: Hotel rooms & travel deals. Available at: <https://www.hotels.com> [Accessed 1 Mar. 2025].
- Koch, A.S., 2005. *Agile software development: Evaluating the methods for your organization*. Boston; London: Artech House, Cop.
- Laraveldaily.com, 2025. 01 - Install Laravel: Composer or Laravel Installer | Laravel Daily. Available at: <https://laraveldaily.com/lesson/laravel-beginners/install-composer-installer> [Accessed 1 Mar. 2025].
- Lu, Y. and Lee, J., 2021. Optimizing hotel booking systems using cloud-based services. *Journal of Cloud Computing & Hospitality*, 28(5), pp. 56-72.

- Larman, C., 2005. Applying UML and Patterns. Pearson Deutschland GmbH.
- McCool, S., 2012. Laravel starter: The definitive introduction to the Laravel PHP web development framework. Birmingham: Packt Publishing, November.
- Meister, C.W., 2021. DevOps - Successfully Combining Development and IT Operations. BoD – Books on Demand.
- Nielsen, J., 1994. Usability Engineering. Amsterdam: Morgan Kaufmann.
- Nguyen, A. and Le, T., 2022. Customer feedback systems in hospitality: Trends and developments. International Journal of Hospitality Management, 47(1), pp. 50-62.
- Pitt, C., 2021. PRO PHP 8 MVC: Model view controller architecture-driven application development. S.L.: Apress.
- Pink, D.H., 2018. Drive: The Surprising Truth about What Motivates Us. S.L.: Canongate Books Ltd.
- Pressman, R.S. and Maxim, B.R., 2020. Software engineering: A practitioner's approach. New York, NY: McGraw-Hill Education.
- Robertson, S. and Robertson, J., 2012. Mastering the requirements process. Upper Saddle River, N.J: Addison-Wesley.
- Solomon, M., 2016. The Heart of Hospitality. SelectBooks, Inc.
- Shore, J. and Warden, S., 2008. The art of agile development. Beijing: O'Reilly.
- Sturgis, H.E., 1988. Software. University Park: Pennsylvania State Univ.
- Sommerville, I., 2011. Software Engineering. Pearson Higher Ed.
- Torres-Moreno, J-M., 2014. Automatic Text Summarization. London: ISTE.
- TripAdvisor, n.d. TripAdvisor: Read reviews, compare prices, and book. Available at: <https://www.tripadvisor.com> [Accessed 11 Mar. 2025].
- Whitaker, K., 2010. Principles of software development leadership: Applying project management principles to agile software development leadership. Boston, MA: Charles River Media.
- Zhao, X., 2019. Leveraging customer feedback to enhance hotel service quality. International Journal of Hospitality & Tourism Research, 25(4), pp. 100-112.

Appendix A- Technical Documentation

System Architecture

The system follows a three-tier architecture:

- **Presentation Layer:** The front-end user interface built using HTML, CSS, and JavaScript. It uses Bootstrap for responsive design and ensures seamless user experience across devices.
- **Business Logic Layer:** The backend is powered by PHP and Laravel Framework. It handles the business logic for processing reservations, handling feedback, and managing services like meals, cab, and spa requests.
- **Data Layer:** The database is managed with MySQL. It stores all user data, booking details, feedback, and related information.

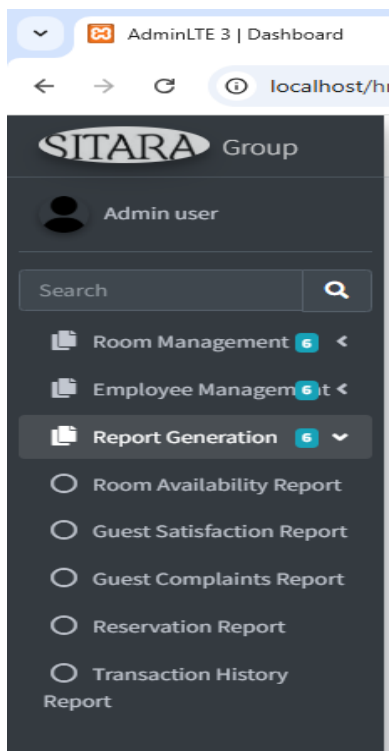
Database connection

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=db_hrcfms
DB_USERNAME=root
DB_PASSWORD=root
```

FigureA 1:Database connection

Appendix B- MIS Report

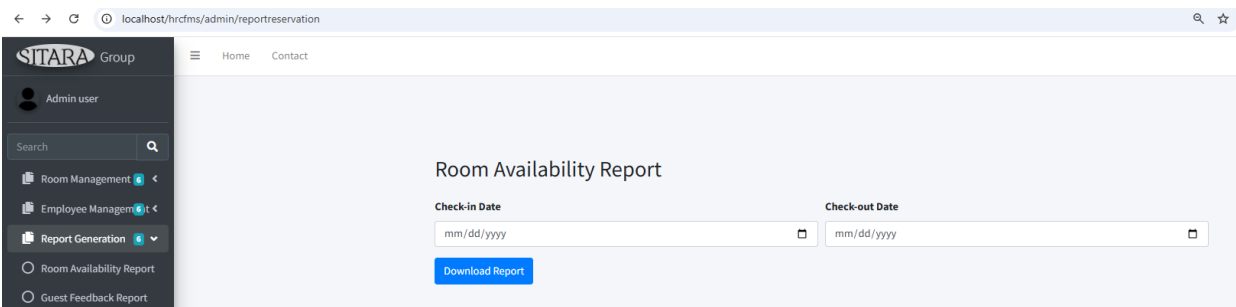
The MIS (Management Information System) reports at Sitara Hotel Weligama provide structured data from various hotel operations, such as reservations, room availability, and customer feedback. These reports help management monitor and control key performance aspects, enabling data-driven decision-making. By analyzing these reports, management can track trends, improve operational efficiency, and enhance guest satisfaction, ultimately supporting the hotel's growth and success.



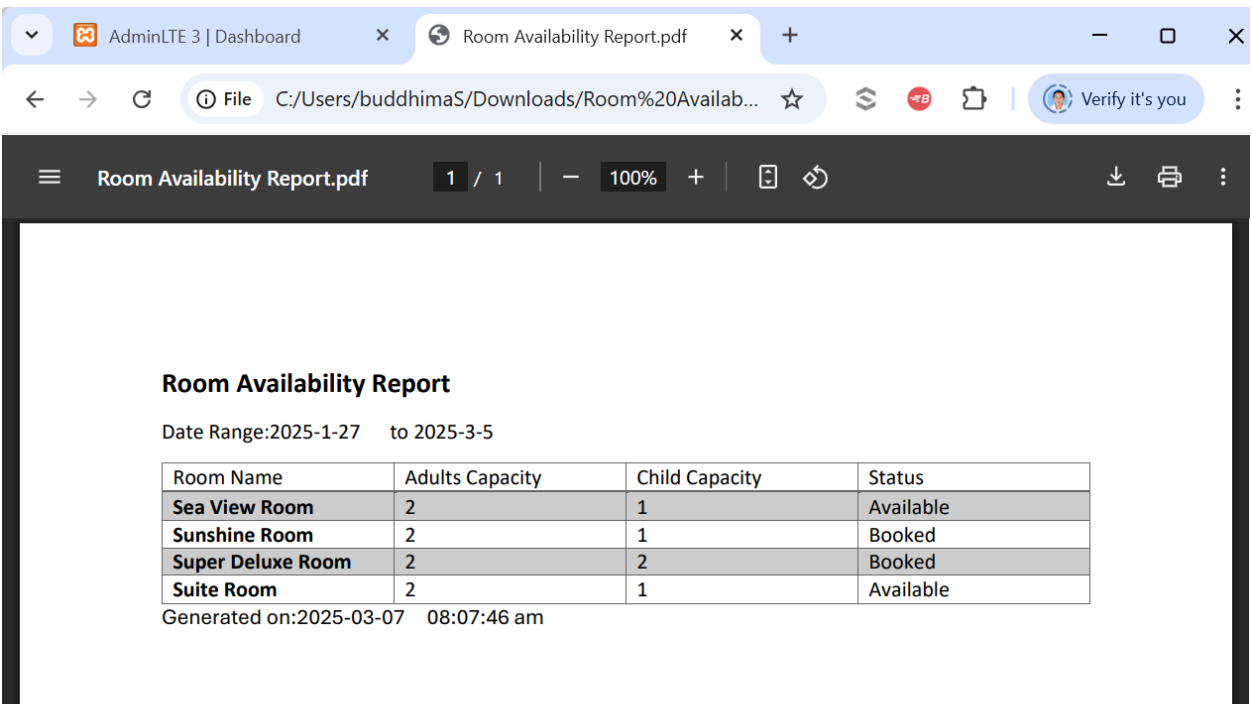
FigureB 2:Report Generation Menu

Room Availability Report

The Room Availability Report provides a detailed overview of room availability at Sitara Hotel Weligama, including room names, adult and child capacities, and current availability status. It displays the data for a specified date range, helping management monitor booking trends. The report is essential for efficiently managing room occupancy and guest reservations.



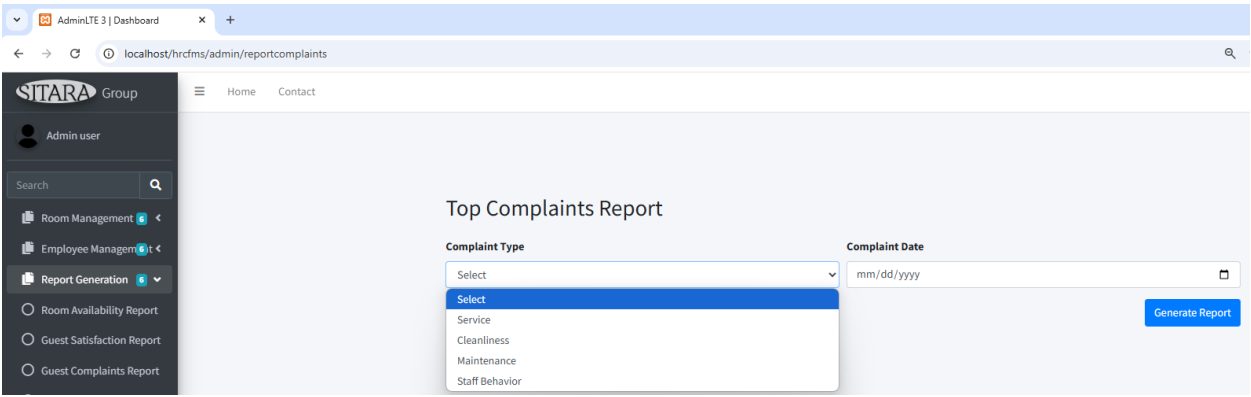
FigureB 3:User_Interface to generate the room availability report



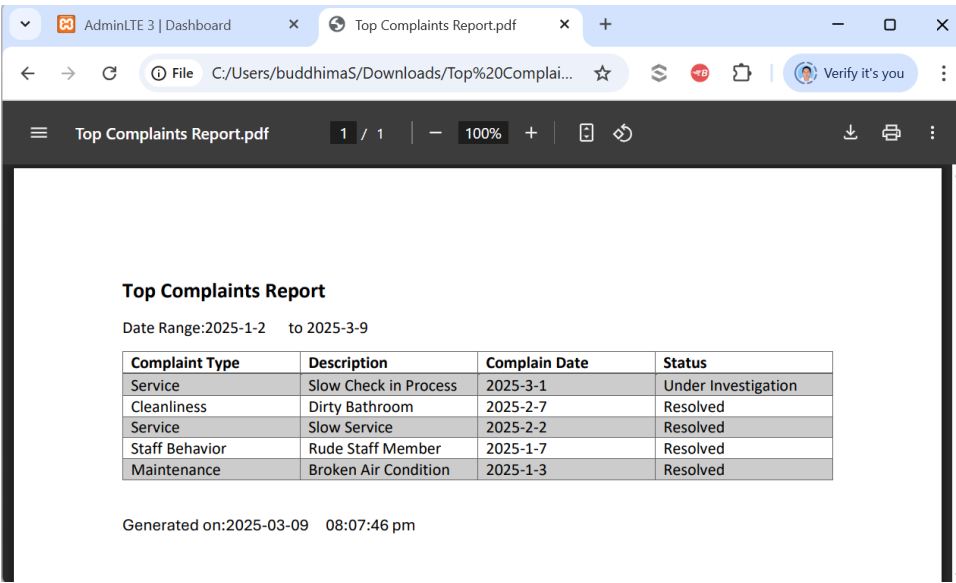
FigureB 4:Room availability report

Top Complaints Report

The Top Complaints Report provides an overview of customer complaints received over a specific date range. The report includes details such as the Complaint Type, Description, Complaint Date, and Status. It provides an overview of customer complaints. This report is useful for hotel management to track ongoing issues, ensure timely resolutions, and improve overall guest satisfaction.



FigureB 5:User interface to generate top claims report



FigureB 6:Top claims report

Hotel Reservation Report

The Hotel Reservation Report provides an overview of reservations within a specified date range, including Reservation ID, Check-in Date, Check-out Date, and Reservation Status. It helps management track and manage bookings, monitor reservation statuses.

The screenshot shows the 'Hotel Reservation Report' page in the SITARA Group AdminLTE 3 dashboard. The page has a sidebar with navigation links: Room Management, Employee Management, Report Generation (selected), Room Availability Report, Guest Satisfaction Report, Guest Complaints Report, and Reservation Report. The main content area contains the report generation form with the following fields:

- Check-in Date:** A date input field with a calendar icon, showing 'mm/dd/yyyy'.
- Check-out Date:** A date input field with a calendar icon, showing 'mm/dd/yyyy'.
- Reservation Status:** A dropdown menu with 'Select' as the current option.
- Generate Report:** A blue button to generate the report.

FigureB 7:User Interface to generate the hotel reservation report

The screenshot shows the generated 'Hotel Reservation Report.pdf' document. The report includes the following information:

- Hotel Reservation Report**
- Date Range:** 2025-1-10 to 2025-3-1
- Table:**

Reservation ID	Check-in Date	Check-out Date	Reservation Status
00005	2025-2-7	2025-2-8	Under Investigation
00004	2025-2-5	2025-2-7	Resolved
00003	2025-2-1	2025-2-2	Resolved
00002	2025-1-5	2025-1-7	Resolved
00001	2025-1-2	2025-1-3	Resolved

Generated on:2025-03-08 05:07:46 pm

FigureB 8:Hotel reservation report

Transaction Report

The Transaction Report provides details about transactions within a specified date range, including Booking ID, Payment ID, Booking Date, and Payment Status. It helps management track the status of payments for bookings, showing whether they are Completed or Pending. This report is essential for ensuring accurate transaction monitoring and financial management.

The screenshot shows the 'Transaction Report' page in the AdminLTE 3 dashboard. The page has a sidebar with navigation links: Room Management, Employee Management, Report Generation (selected), Room Availability Report, Guest Satisfaction Report, Guest Complaints Report, Reservation Report, and Transaction History Report. The main content area has a 'Transaction Report' title and two date input fields labeled 'Start Date' and 'End Date', both with a placeholder 'mm/dd/yyyy'. A blue 'Generate Report' button is located to the right of the 'End Date' field.

FigureB 9:User Interface to generate the transaction report

The screenshot shows the 'Transaction Report.pdf' file opened in a web browser. The PDF content includes the title 'Transaction Report', the date range 'Date Range:2025-1-1 to 2025-3-9', and a table with 4 columns: BookingID, PaymentID, Booking Date, and Payment Status. The table contains 4 rows of data. Below the table, it says 'Generated on:2025-03-08 02:07:46 pm'.

BookingID	PaymentID	Booking Date	Payment Status
00005	P0004	2025-3-8	Pending
00004	P0003	2025-2-7	Completed
00003	P0002	2025-2-2	Completed
00002	P0001	2025-1-7	Completed

Generated on:2025-03-08 02:07:46 pm

FigureB 10:Transaction Report

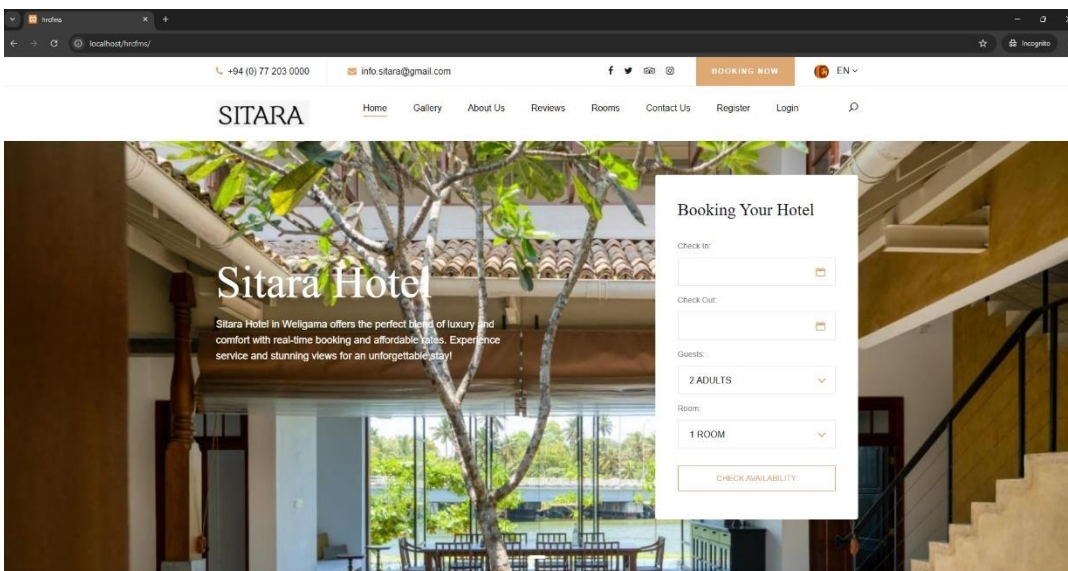
Appendix C- User Manual

This comprehensive guide is designed to facilitate the efficient navigation and utilization of the Sitara Hotel Management System. The system incorporates a wide range of functionalities, including the management of guest reservations, tracking check-ins and check-outs, monitoring room availability, generating guest satisfaction reports, and overseeing marketing campaigns. It is intended for use by all roles within the hotel, including Guests, Front Desk Officers, Marketing Staff, and Admin Users. This manual will provide detailed instructions to ensure users can fully leverage the system's capabilities, thereby enhancing the efficiency of hotel operations and the quality of customer service.

Guest: Room Reservation

Step1: Check Availability

The Booking Your Hotel section allows users to input essential details for making a reservation, including check-in and check-out dates, the number of adults, children, and rooms required. Once the information is entered, users can click the "Check Availability" button to check the availability of rooms.

The image is a screenshot of a web browser displaying the Sitara Hotel website. The browser's address bar shows 'localhost/hotels/'. The website's header includes the Sitara logo, a navigation menu with links like Home, Gallery, About Us, Reviews, Rooms, Contact Us, Register, and Login, and contact information: '+94 (0) 77 203 0000' and 'info.sitara@gmail.com'. A 'BOOKING NOW' button and a language selector 'EN' are also present. The main content area features a large background image of a hotel interior with a tree. Overlaid on this is a 'Booking Your Hotel' form. The form contains the following fields: 'Check In:' with a calendar icon, 'Check Out:' with a calendar icon, 'Guests:' with a dropdown menu showing '2 ADULTS', and 'Room:' with a dropdown menu showing '1 ROOM'. At the bottom of the form is a 'CHECK AVAILABILITY' button. Below the form, there is a small text block: 'Sitara Hotel in Weligama offers the perfect blend of luxury and comfort with real-time booking and affordable rates. Experience service and stunning views for an unforgettable stay!'

FigureC 1:Check Availability

Step2: Pick a Room

The system then displays the available rooms, each accompanied by a "Select Room" button. Once the user clicks on the "Select Room" button, the chosen room is added to the Selected Rooms section, where the user can view the total cost calculation for the selected room. This feature enables users to calculate the overall cost of their stay.

SITARA Home Gallery About Us Reviews Rooms Contact Us Register Login

Available Rooms
Check In: Check Out: Adults: Children:
Room Count:
[Check Availability](#)

Room Name	Price per Night	Adults Capacity	Children Capacity	Action
Sea View Room	\$27000.00	2 Adults	2 Children	Select Room
SunShine Room	\$21000.00	2 Adults	1 Children	Select Room

Selected Rooms

Room Name	Total Price	Action
Total Price: \$0		

[Calculate Total Price](#)

FigureC 2:Pick a Room

Step3: Enter the Guest Detail

SITARA Home Gallery About Us Reviews Rooms Contact Us Register Login

Guest Details

1 Pick a Room 2 Guest Details 3 Make it Your Own 4 Payment Confirmation

Title:

First Name: Last Name:

Country: Mobile Number:

Email: Confirm Email:

Special Request:

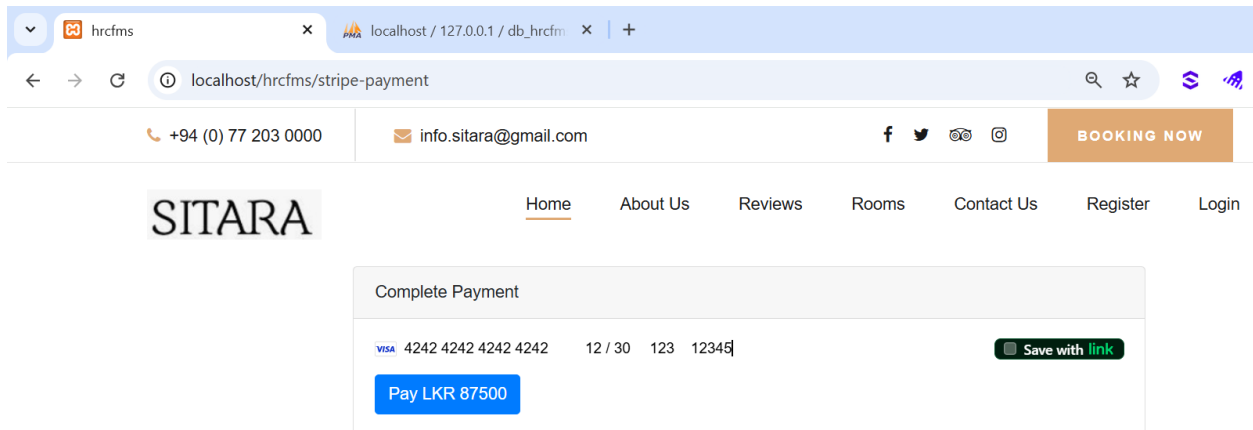
Arrival Time: To make your stay even more convenient, let us know your expected arrival time! This allows us to prepare for your arrival.

Join Sitara
To enjoy member benefits, exclusive member rates, and more

FigureC 3:Guest Detail Capturing UI

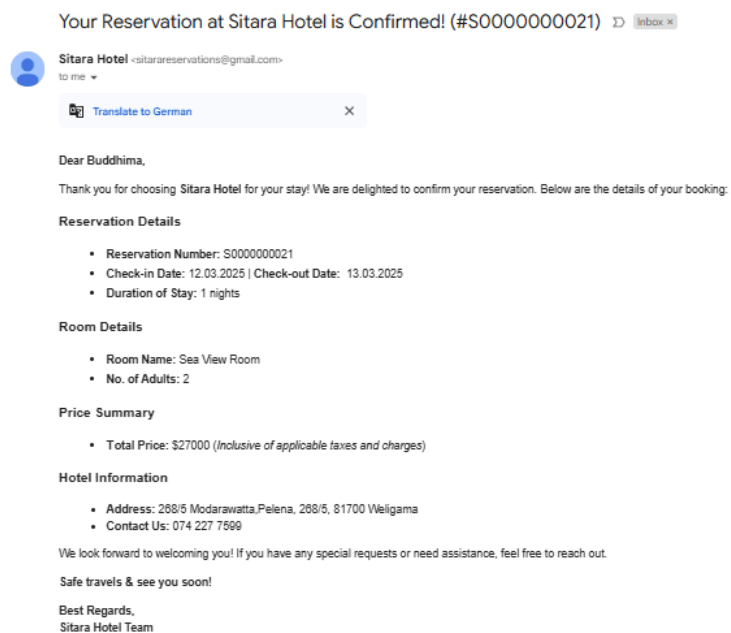
Step3: Payment Confirmation

Guests can choose between two payment methods: Online payment or Pre Check-in Cash/Card and specify the expected arrival time with their chosen payment option.



FigureC 4:Payment Confirmation UI

Step3: Guest Receive the Email Confirmation



FigureC 5:Email Confirmation UI

Guest: Additional Services Booking.

Guests have the option to make additional reservations for meals, cab services, or spa treatments, enhancing their overall stay experience.

The screenshot displays the SITARA website's booking interface. At the top, there's a navigation bar with links like Home, Gallery, About Us, Reviews, Rooms, Contact Us, Register, and Login. Below this is a progress bar with four steps: 1. Pick a Room, 2. Guest Details, 3. Make it Your Own, and 4. Payment Confirmation. The current step is 'Make it Your Own', which contains three sub-sections: Meal, Cab Service, and Spa Service. The 'Meal' section is active, showing a form with fields for 'Select a Meal', 'Meal Selection', 'Reservation Date', and 'Reservation Time'. A 'Book' button is at the bottom right of the form. The footer includes the SITARA logo, contact information, and a 'NEW LATEST' section.

FigureC 6:Meal Services Booking

This screenshot shows the 'Cab Service' booking form on the SITARA website. The form is located under the 'Make it Your Own' section. It includes fields for 'Reservation ID', 'Pickup Date', 'Pickup Time', 'Vehicle Type', 'Drop Description', and 'Distance (KM)'. A 'Book' button is positioned at the bottom right of the form. The website's header and footer are consistent with the previous screenshots.

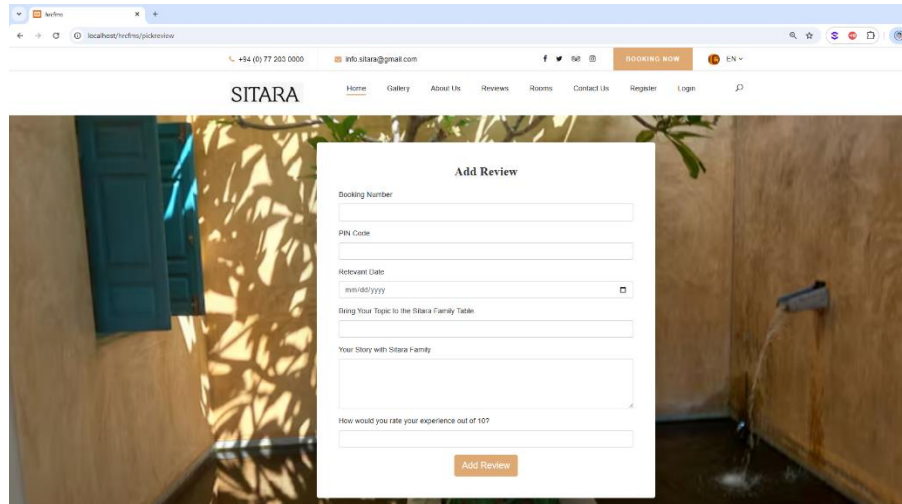
FigureC 7:cab services booking

The screenshot displays the 'Spa Service' booking form on the SITARA website. The form is part of the 'Make it Your Own' section. It features fields for 'Select a Service', 'Spa Service', 'Appointment Date', and 'Appointment Time Slot'. A 'Book' button is at the bottom right of the form. The website's header and footer are consistent with the previous screenshots.

FigureC 8:spa Services booking

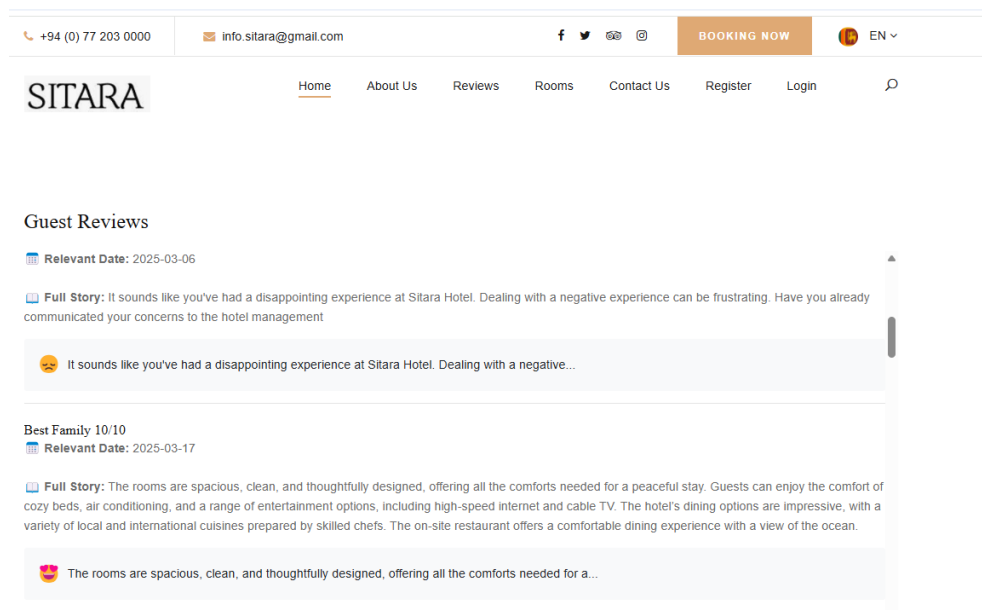
Guest: Add Reviews

This section allows guests to add reviews by entering their booking number, PIN code, relevant date, and sharing their experience. They can also rate their experience out of 10 and submit the review to be shared with the Sitara family



FigureC 9:Add Reviews

Guest: View Existing Reviews and Review Summarization



FigureC 10:Review Summarization

Guest: Send Message Via Contact Us section

Cera Truck
2605 Mokolawatha, Pinnawa, Weligama, Sri Lanka
+94 71 224 9878
info@starahotel.com
Follow Us

Contact Us

Your Name
Enter your name

Mobile Number
Enter your mobile number

Your Email
Enter your email

Subject
Enter subject

Message
Enter your message

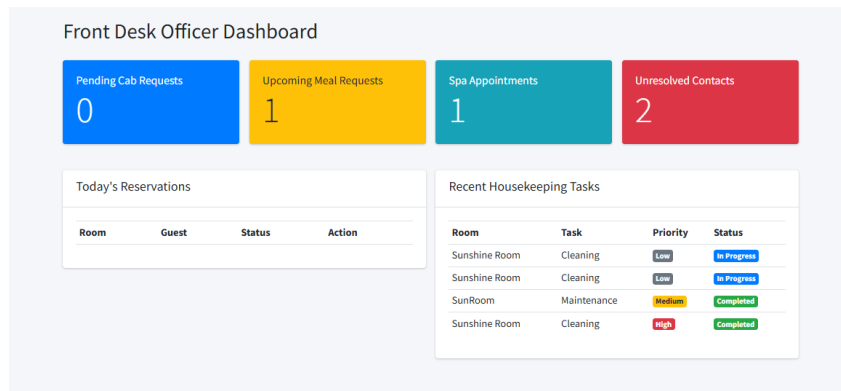
Send Message

FigureC 11:Contact us UI

Step1: Click on the Contact Us Button

Step2: Fill the Form and click on the Send Message button

Front Desk Officer: Dashboard



FigureC 12:Front Desk Officer Dashboard

Step1: Login to the Portal

Step2: View the Dashboard

Front Desk Officer: Mark the Check-In and Check-Out

Reservation Management					
Room Number	Guest Name	Check-in Date	Check-out Date	Status	Actions
Sun Shine Room	sesa sasi	Jan 21, 2026 00:00	Jan 22, 2026 00:00	Checked In	Mark Check-out
Sun Shine Room	sathis sehanka	Jan 20, 2026 00:00	Jan 21, 2026 00:00	Checked Out	
Sea View RoomEdited	Sethas sehanka	Jan 15, 2026 00:00	Feb 16, 2026 00:00	Checked Out	
Sea View RoomEdited	neha sehansa	Jan 14, 2026 00:00	Jan 15, 2026 00:00	Upcoming	Mark Check-in

FigureC 13:Mark Check in and check out

Step1: Login to the Portal

Step2: Navigate to the Reservation Management section

Step3: Mark the Check in and Check Out Status

Housekeeping Officer: Housekeeping ticket

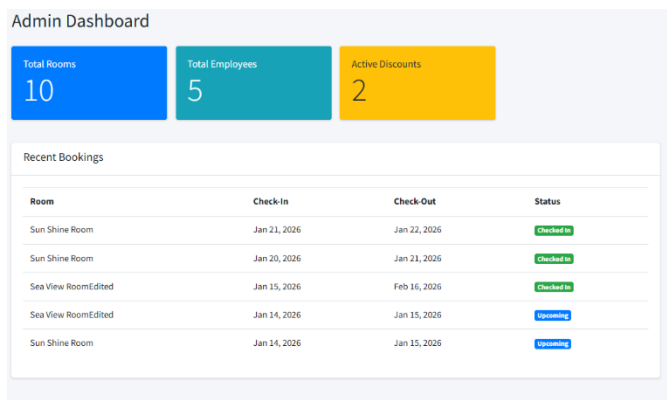
Step1: Login to the portal and click on the housekeeping task management.

Step2: Select the Action from the task

Housekeeping Tasks					
Room #	Task Type	Priority	Description	Status	Actions
Sunshine Room	Cleaning	Low	Wet Floors	In Progress	In Progress ▼
Sunshine Room	Cleaning	Low	AC Repairing	In Progress	In Progress ▼
SunRoom	Maintenance	Medium	AC Repairing	Completed	Completed ▼
Sunshine Room	Cleaning	High	Need to do it immediately	Completed	Completed ▼

FigureC 14:House keeping ticket management

Admin: Dashboard



FigureC 15:Admin Dashboard

Step1: Login to the Portal

Step2: View the Dashboard

Admin: Add and Edit New Room Details

The 'Add Room' form includes fields for Room Number, Room Type (dropdown), Price Per Night, Room Description, and Status Availability (dropdown). A 'Submit' button is at the bottom.

Manage Rooms					
Room ID	Room Name	Adults	Children	Price/Night	Actions
1	Sea View RoomEdited	3	2	\$39,000.00	Edit
2	Sun Shine Room	2	1	\$22,500.00	Edit
3	Family Hub Room	3	2	\$38,300.00	Edit
4	SunGold Room	2	0	\$37,000.00	Edit

FigureC 16:Room Management UI

Step1: Login to the Portal and navigate to the room management section.

Step2: Click on Add Room, Enter the details and click on submit button/ Click on Edit button and change the already added data.

Admin: Add and Edit Employee Details

The screenshot shows a web browser window with the URL 'localhost:3000/admin/employees'. The left sidebar contains a menu with 'Admin Portal', 'Employee Management', 'Add Employee', 'Edit Employee', 'Delete Employee', and 'New Employee'. The main content area is titled 'Add Employee' and contains the following form fields:

- Name:
- Mobile:
- Email:
- Employee Type:
- Password:

A blue 'Submit' button is located at the bottom of the form.

Manage Employees

Email	Mobile	Employee Type	Actions
kalana6@gmail.com	0767656549	FrontDeskOfficer	Edit Delete
dahamsa99@gmail.com	07676767659	CleaningStaff	Edit Delete
sathis@gmail.com	0767288128	FrontDeskOfficer	Edit Delete

FigureC 17:Employee management UI

Step1: Login to the Portal and navigate to the Employee management section

Step2: Click on Add Employee, Enter the details and click on submit button/ Click on Edit button and change the already added data