Module Code	**:**	AAPP003-4-2 (Introduction to Software Engineering)
Intake Code	**:**	UCDF2307ICT(SE)
Lecturer Name	**:**	Ts. Dr. Law Foong Li
Hand in Date	**:**	7th December 2024

Group Details

Student Name	TP Number
1. Vincent Teo Kai Qi	TP075160
2. Wang Jim Loong	TP075739
3. Wang Zi Jie	TP076251
4. Wong Jun Ming	TP076078

Group Assignment Instruction

1. The assignment must be submitted according to the given hand-in-date.
2. Late submissions will not be assessed and awarded **ZERO (0)** marks.
3. This assignment consists of **TWO (2)** components, Group Component & Individual Component. These 2 components must be combined into **ONE (1)** documentation.
4. Submission must be done through Moodle.

Table of Contents

Abstract 5
1.0 Introduction 6
1.1 Project Background 6
1.2 Objectives 8
1.3 Scopes 9
2.0 Planning and Requirements Analysis 10
2.1 Explanation of Software Process 10
2.1.1 Software Specification 10
2.1.2 Software Design and Implementation 13
2.1.3 Software Validation 15
2.1.4 Software Evolution 18
2.2 Functional and Non-Functional Requirements 19
2.2.1 Functional Requirements 19
2.2.2 Non-Functional Requirements 22
3.0 Logical Design 24
3.1 Use Case Diagram 24
3.2 Use Case Description 25
3.3 Class Diagram 58
4.0 Physical Design (Wire-framing) 59
5.0 Implementation 80
5.1 Hardware Requirements 80
5.2 Software Requirements 84
6.0 Prototype 87
7.0 Conclusion 88
8.0 Methodology (Individual Part 1): 90
8.1 Spiral Model (VINCENT TEO KAI QI – TP075160) 90
8.1.1 Implementation of Spiral Model 91
8.1.2 Advantages and Disadvantages of Spiral Model 93
8.1.3 Justification of Spiral Model 94
8.2 Scrum Model (WANG JIM LOONG – TP075739) 95
8.2.1 Implementation of Scrum Model 96
8.2.2 Advantages and disadvantages of the methodology 98
8.2.3 Justification of Scrum Model: 98
8.3 Rational Unified Process (RUP) (WANG ZI JIE – TP076251) 99

8.3.1 Implementation of the Rational Unified Process (RUP): 99
8.3.2 Advantages and Disadvantages of the Rational Unified Process (RUP) 101
8.3.3 Justification of the Rational Unified Process (RUP) 101
8.4 Waterfall Methodology (WONG JUN MING – TP076078) 102
8.4.1 Implementation of the waterfall model 102
8.4.2 Advantages and Disadvantages of the waterfall model 104
8.4.3 Justification of Waterfall Model 105
8.5 Final Methodology Chosen 106
9.0 Activity Diagram (Individual Part 2) 107
9.1 Vincent Teo Kai Qi (TP075160) 107
9.1.1 Introduction 107
9.1.2 Implementation 109
9.2 Wang Jim Loong (TP075739) 110
9.2.1 Introduction 110
9.2.2 Implementation 111
9.3 Wang Zi Jie (TP076251) 112
9.3.1 Introduction 112
9.3.2 Implementation 114
9.4 Wong Jun Ming (TP076078) 115
9.4.1 Introduction 115
9.4.2 Implementation 117
10.0 Testing (Individual Part 3) 118
10.1 Vincent Teo Kai Qi (TP075160) 118
10.1.1 Testing Method 118
10.1.2 Test Script 119
10.2 Wang Jim Loong (TP075739) 120
10.2.1 Testing Method 120
10.2.2 Test Script 121
10.3 Wang Zi Jie (TP076251) 122
10.3.1 Testing Method 122
10.3.2 Test Script 123
10.4 Wong Jun Ming (TP076078) 124
10.4.1 Testing Method 124
10.4.2 Test Script 125
References 126

Appendix 134
Appendix 1: Gantt Chart 134
Appendix 2: Workload Matrix 135
Appendix 3: Wireframe and Prototype 136
Appendix 4: Additional Features – Chatbot (Zapier.com) 136

Abstract

Swiss Garden Hotel, one of the booking and reservation systems in Kuala Lumpur, provides various services such as room reservations, event hosting, dining, and recreational activities. However, due to its manual processes, the hotel faces inefficiencies in managing reservations, guest information and records, events, and billings. Especially when the hotel's guest and event numbers grow, all these manual methods become more and more difficult to handle, affecting both efficiency and productivity and, at the same time decreasing customer satisfaction because of the time-consuming. Mr. David, the hotel's owner, decides to implement an integrated information system to automate the manual processes, streamline operations, and enhance customer service. The contents will focus more on how a development team develops and implements the system for the Swiss Garden Hotel in order to make it more efficient and productive. The potential users of the booking reservation system are the customers, coordinators, receptionists, room cleaners, and managers. Moreover, a variety of additional features will be added also such as an AI-advanced chatbot, multilingual and multi-currency supports, and parking features.

1.0 Introduction

1.1 Project Background

Swiss Garden Hotel is a well-known accommodation facility located in the heart of downtown Kuala Lumpur. It has become one of the good choices for guests and clients to make room reservations, book spaces for hosting events and meetings, and enjoy dining and recreational activities. Despite its popularity and central location, the hotel still relies on a manual system for managing critical business processes such as room reservations, guest records, event bookings, and billing.

All these manual methods have become increasingly difficult to manage with the growing number of guests and events, affecting the hotel's overall efficiency and productivity. Meanwhile, customer experience and satisfaction are also being impacted because of the slower service, high error rates, and poor responsiveness. For instance, manually checking room availability, event bookings and processing payments will be more time-consuming, leading to a longer waiting time for guests. (Booking plugins vs. Manual scheduling, 2024). At the same time, manual data entry will result in more mistakes than automated systems.

Therefore, Mr. David, the owner of Swiss Garden Hotel decided to implement an automated information system to streamline the hotel operations and enhance customer service which can improve both efficiency and productivity for the staff, receptionists, and managers. (King, n.d.). In addition, the implementation could highly increase accuracy by reducing human-make errors, time efficiency, and cost-effectiveness which might enhance customer experience and satisfaction. (Lavelle, 2024). An automated system will handle duties like reservations, billing, and event administration, minimizing the need for human labor and the hotel's overall workload.

The proposed system will implement various modern technologies to improve capabilities and functionalities, which might simultaneously enhance guest experience. For instance, the programming languages chosen for developing the website are **HyperText Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript, HyperText Processor (PHP), and Structured Query Language (SQL)**.

HTML, one of the famous programming languages provides the fundamental structure of web pages, defining elements like text, images, links, and forms. It not only enables content to be accessible across all browsers and devices but also seamlessly integrates with other languages. Moreover, a powerful stack for developing dynamic, data-driven, and interactive web applications with efficient server-side processing, robust database management, seamless

client-side interactivity, and responsive, visually appealing design will be created by using CSS, JavaScript, PHP, and MySQL (Nixon, 2022). Besides, relational databases like MySQL will handle guest and reservation data efficiently, making it easy to generate reports. For the coding part, **Visual Studio Code (VS Code)** will be used because of its relatively lightweight compared to full IDEs, providing a smooth and responsive experience without consuming too much system memory.

One of the new features, chatbots, was implemented using **Zapier.com** to improve workflows and integrate numerous services with chatbots. The chatbot is built with natural language processing (NLP), machine learning (ML), and Frame-based Knowledge Representation. NLP is a software program that allows computers to mimic human talk while communicating with humans (Pollock, 2023) while ML is a subfield of artificial intelligence that applies trained algorithms to establish moulds that allow machines to carry out tasks typically achievable by humans (Coursera, 2023). Furthermore, chatbots offer a wide range of features and functionalities, including the ability to provide instant information with high accuracy and provide 24/7 customer care even on weekdays and public holidays, which can significantly improve the customer experience.

Cloud computing is the internet-based on-demand access to computing resources such as physical or virtual servers, data storage, networking capabilities, application development tools, software, AI-powered analytic tools, and more at pay-per-use pricing (Susnjara & Smalley, 2024). It will offer scalability, real-time data access, and security, ensuring the system operates smoothly and cost-effectively. According to the research, **Amazon Web Service (AWS)** will be the preferred cloud server for the Swiss Garden Hotel Management System due to its worldwide reach, excellent reliability, and solid security measures. Its extensive network of availability areas provides excellent performance and low latency, making it a popular choice for major companies requiring scalable and stable cloud solutions (Mufti, Mittal, & Gupta, 2021).

In addition, **iPay88**, one of the contactless payment gateways that support various online payment methods such as credit and debit cards, and e-wallets like TnG e-wallet will improve speed, security, and convenience by reducing human contact and integrating multiple payment options. For multilingual and multi-currency support, the **WordPress Multilingual Plugin (WPML)** and **Fixer.io** will be used to allow customers to change their preferred language and currency.

1.2 Objectives

One of the objectives of developing an automated system for the Swiss Garden Hotel's manual methods is to streamline hotel operations by automating tasks including room reservation, event booking, guest records, billing management, payment handling, and report generation. By automating these processes, guest experience and satisfaction will be enhanced by offering a seamless and convenient way to interact with the hotel. The system will also increase accuracy and reduce errors by minimizing the need for human intervention in routine processes, preventing issues such as double bookings or billing mistakes.

Another important goal is real-time data access, which helps hotel management to swiftly make well-informed decisions about things like creating financial reports or keeping an eye on available rooms. Furthermore, secure payment management integrates contactless and mobile payment alternatives for increased simplicity and security, guaranteeing the safety of guests' financial transactions. For example, traditional offline payment methods should be replaced by the most famous online payment methods available in Malaysia such as Touch n' Go e-wallet, Grab Pay, Apple Pay, and credit or debit cards (Dunse, 2021).

Furthermore, the implementation of a chatbot supports 24/7 availability which would enable customers to get immediate responses. The customer can receive immediate assistance from the chatbot by asking some relevant questions without waiting for human staff availability even during holidays and weekends (Hils & T, 2024). Besides, it can also increase the productivity of the hotel management system since the customer does not need to wait for human responses and simultaneously increase the customer's overall experience and satisfaction.

Last but not least, the multilingual and multi-currency support allows users to select their chosen language and currency while exploring the hotel's official website. These might highly enhance customer satisfaction by leading them to a more personalized and satisfying experience. Additionally, both multilingual and multi-currency can broaden the market reach to a global presence and boost the hotel's profitability at the same time by attracting more international travelers.

1.3 Scopes

The proposed hotel management system for the Swiss Garden Hotel includes several core functionalities aimed at modernizing and enhancing the hotel's operational efficiency and guests. First and foremost, the system will be able to handle room reservation management, which includes bookings, cancellations, guest details, preferred check-in and check-out dates, and checking room availability. Furthermore, guests can use the system for event bookings by providing information such as date, time, expected number of attendees, and specific requirements. The customers can also view the dining options and recreation activities available at the hotel.

There are a total of 3 types of staff which are receptionists, room cleaners and coordinators. The receptionists can use the system to assist customers with booking modifications and special requests. The room cleaner can view the room assigned by the admin that need to be cleaned, update the cleaning status, and report issues. Not only that, but the coordinator can manage customer bookings and parking requests.

The admin should have permission for guest records management to keep track of the guest's history. Moreover, the system can handle billing management, displaying the total bill for each guest and booking, and then storing it in the system's database. The system will also integrate contactless payment options to streamline financial transactions, enhancing security and convenience for guests. This will support offline and online payment methods. Once the payment is complete, the system will update the status automatically and generate an online receipt as a PDF file.

2.0 Planning and Requirements Analysis

2.1 Explanation of Software Process

The software process is a sequence of processes that lead to the production of software products, ensuring its quality, capabilities, functionalities, and efficiency. In software engineering, it refers to the methods and techniques used to develop and maintain software. (GeeksforGeeks, Software Processes in Software Engineering, 2024). For the implementation of software for the Swiss Garden Hotel, there are a total of four key stages including specification, design and implementation, validation, and evolution.

2.1.1 Software Specification

The Swiss Garden Hotel operates processes manually including room and event booking, guest details and reservation management, payment processing, and report generation. These manual processes will cause inefficiency and lower productivity, especially when the number of visitors and events increases. Therefore, Mr David suggests implementing an online system to automate and streamline the daily operation. The objective of the system is not only to enhance the efficiency of whole manual processes but simultaneously can improve the hotel's overall performance.

For the **operational** feasibility study, the development team should be able to know whether the system developed can meet the hotel's needs and improve its current operation. According to requirements, the newly developed system shall be able to include a booking automation service that allows guests and receptionists to book rooms or events either online or in a physical way. Besides that, automation for guest records management, billing, and payment handling is significant for the system. All these services are required to be reliable, high-performance, accurate information, and fast response time. For example, the manual check-in or check-out process currently takes over 10 minutes, but with automation, it can be completed in under 2 minutes. It shows how effective the developed system is. In order to ensure that all staff know how to use and operate the management system, staff training is required. This is necessary to ensure a smooth transition from manual to automated processes.

The **technical** feasibility study is a process of determining whether the organization has the technology resources to develop, purchase, install, and operate the system. Before starting to implement the real system, the development team shall decide what are the hardware requirements for the Swiss Garden Hotel Management System. Since the hotel requires an efficient online system that can handle room booking, payment processing, multilingual

features, and chatbot integration, the team need to prepare sufficient hardware and tools. For instance, a Database Management System (DBMS) is required for the hotel management system not only to store the guest records, booking history, and other critical data. Therefore, the development team decided that MySQL would be used for database management, and the cloud server, Amazon Web Services (AWS) would be used for storing data on the cloud.

Furthermore, the **schedule** feasibility study is the process of assessing the degree to which the potential time frame and completion dates for all major activities within a project meet organizational deadlines and constraints for affecting change. The development team shall separate the development and implementation of the system into phases including software specification, software design and implementation, software validation, and software evolution, then give the estimated time for each phase. For example, the requirement collection and analysis phase is expected to take 2-3 weeks, with the design and prototyping phase taking less than a month. The development and integration phase may last 8-10 weeks, with a focus on implementing basic functionality like booking automation, payment processing, and multilingual assistance. The testing and validation process will take three to four weeks to ensure that all components function effectively. Finally, the deployment and staff training phase will last a further 2-3 weeks to ensure a smooth transition from the manual method.

Last but not least, an **economic** feasibility study will be conducted to establish whether the project can be completed within the provided budget and to calculate the cost and benefit (including profit) of the project. The development team need to list out the budget prepared for the system development and implementation and calculate the initial investment including the cost of hardware (computers, POS devices) and software (MySQL, cloud services), licensing fees for third-party services, and staff training. Additionally, the team must consider the extra variable costs such as ongoing maintenance, software updates, and support services. By estimating the financial benefits that can be brought to the Swiss Garden Hotel, which contains the revenue and reduces labor costs, the team can create a cost-benefit analysis (CBA) that will assist in determining whether the project is economically feasible and will provide an impressive return on investment for the hotel.

To gather the users' requirements, there are a variety of research methods including interviews, usability testing, focus groups, surveys, and A/B testing. For example, the development team might conduct interviews with different range of potential users such as customers, staff, and admins. Through interviews with the potential users and asking some

open-ended questions focused on behavior and attitude, the team might gain more detailed insights into specific features and functionalities for each user group, like booking preferences for customers, daily operational needs for staff, and management operations for admins. This approach helps to collect detailed, qualitative feedback and ensures a user-centered design.

After gathering the users' requirements, the development team need to do an analysis and document all types of requirements gathered. The team shall be able to list out the functional and non-functional requirements of the implemented software. For instance, one of the functional requirements is real-time room availability which is provided for updating a room or hall status such as has been booked, available, or under maintenance in real-time. Besides that, a real-time report is one of the system's essential capabilities for generating reports on room and event hall occupancy, bookings, revenue, and financial transactions. It can also do performance analytics by providing data insights for hotel management to make informed decisions regarding operations and financial performance.

For non-functional requirements, security is considered one constraint to the Swiss Garden Hotel Management System. For login purposes, both guests and staff are required to register an account first before accessing the website. Complying with data protection regulations requires the secure handling of guest data, especially personal information and payment details. According to Robertson, the system must employ role-based access control (RBAC) to restrict sensitive areas, such as billing and guest records, to authorized personnel only (Robertson, 2023). Reliability is another key constraint as the system must maintain 24/7 availability, especially during peak times like public holidays to ensure minimal downtime. The system developers should implement regular backups and an automated data recovery process to prevent data loss in the event of system failure. This guarantees that the hotel's operations continue smoothly without interruptions, providing a dependable experience for both guests and staff.

2.1.2 Software Design and Implementation

Software design and implementation are one of the software processes in software engineering. It is the process of creating an executable system from a system specification. Problem-solving and planning for a software solution are the processes that make software design (Kulak, Cebi, & Kahraman, 2010)The process of defining a system's architecture, interfaces, data, and components is known as software design. This process entails organizing and structuring the system. Software that is designed effectively is easy to understand, reliable, and adaptable. (Why software design is important, 2024). However, software implementation is a process that converts a system's design into codes. Implementation entails a few procedures that include installation, modification, configuration, and integration to ensure it fulfills the objectives and requirements. (Sahoo, 2024). Software design and implementation involve four key activities: architectural design, interface design, component design, and database design.

In software engineering, "architectural design" refers to the high-level process of describing the overall structure of the system. Architectural design is a process of determining the subsystems that comprise a system, as well as the framework for their control and communication. (Sommerville Software Engineering (10 ed), n.d.). It breaks down the system into components or modules, detailing how the components interact with each other, and defining the system's interfaces and data flows. For example, the room reservation feature on the Swiss Garden Hotel's website is a part of the architectural design and works in conjunction with the payment feature. When users select their desired room and services, the website will calculate and show the total price of the booking for the user's reference. After the user clicks the "Proceed to Checkout" button, the total price data is passed to the payment feature. This integration means that the user does not have to manually enter the amount again when paying the bill; instead, the system automatically charges the price based on the data received from the room reservation feature.

Interface design is the process that communicates boundaries between different components of a system. This process interacts with components by specifying their data formats, protocols, methods, and rules for managing the exchange of information. Once the interface specifications (i.e., data formats, and communication method) are clearly defined by the development team, the components can be developed independently at the same time, as the team already knows how each component will communicate with others. However, developers need to ensure their work still stays on track according to the agreed-upon interface

throughout the development process. The Swiss Garden Hotel's website interface design ensures smooth communication between each component. For example, the room reservation system will pass the booking information (i.e., room type, number of pax, total price) to the payment system. The payment system will process the information to proceed with the payment. Meanwhile, the guest records management system will receive the customer's details (i.e., first name, last name, contact number, and email) from the room reservation system.

The main objective of component design is to break down complicated software systems into smaller, reusable units, or simply modules. (Anshika, 2024). Component design emphasizes the function of each component, how they will interact with one another, and what types of data it will manage. This process enhances flexibility and scalability, allowing the formation of dynamic systems made up of smaller parts that interact with each other. (Anshika, 2024). On the Swiss Garden Hotel's website, each function is developed well with component design. For instance, the room reservation module manages the room booking process including room availability, room types, and reservation confirmation. However, the payment module is responsible for handling payment transactions with various payment methods. These modules will interact with each other to process the payment once the booking is confirmed by the customer.

Last, database design is the process of organizing data and defining the structure, management, and storage of information in a database system. It entails deciding how data will be stored and accessed. A good database design will greatly enhance the efficiency of data retrieval, security, and scalability. A correctly structured database provides you with access to up-to-date and accurate information. (Microsoft, 2024). A good database design enables and guarantees the accuracy and integrity of information, it also meets the needs of data processing and reporting. (Microsoft, 2024). Database is created by 3 components which are entities, relationships, and tables. Entities represent real-world objects or concepts. Relationships represent how entities interact with each other. While tables in a database represent data (i.e., attributes and entities) in an organized way. For example, the Swiss Garden Hotel's system has an entity called "Customer". It stores customer's personal information (i.e., first name, last name, contact number, and email). Meanwhile, the system also stored an entity called "Reservation". The "Reservation" entity stores room reservation information (i.e., check-in date, check-out date, room type, and number of pax). The "Customer" entity has a one-to-many relationship with the "Reservation" entity because a customer can make multiple reservations but each reservation is linked to only one customer.

2.1.3 Software Validation

Validation is the process of determining recognized evidence to confirm that the computer system has been correctly installed, meets user requirements, and operates normally according to its intended use. (O'Donnell, 2024). The purpose of validation is to check whether the software operates correctly according to the user's requirements and expectations and to ensure the developed system can provide the required functions and is free of faults or weaknesses to avoid impacting performance. (iderawpadmin, 2022)

Three types of validation testing are crucial to the software process that are component/unit testing, system testing, and acceptance testing. (GeeksforGeeks, Unit Testing | Software Testing, 2019). Unit testing is a fundamental aspect of software testing that independently tests various components or functions of software applications. (GeeksforGeeks, Unit Testing | Software Testing, 2019). This method ensures that each unit of the software operates as expected. For example, the room availability module of the Swiss Garden Hotel management system will be tested to ensure that when a room is booked, it is marked as unavailable. The billing calculation function should also be tested to verify that it correctly adds up the costs of the room stay, parking fee, and any additional services. This can ensure that when guests make room reservations and choose parking, the system can accurately update the number of available parking spaces. The multi-language and currency functions must be tested to ensure that the system not only can accurately translate all relevant information into supported languages without losing its meaning or context but also accurately convert the price to the selected currency based on the latest exchange rates. For the chatbot function, the hotel management system should validate it to make sure that it can accurately answer frequent questions related to room reservations, parking, events, and billing in a wide range of languages. (Gotra, 2021). The developer tests individual units by using mock inputs to ensure they produce the correct outputs.

System testing is a type of software testing aimed at evaluating a fully integrated system to verify whether it meets the corresponding requirements (GeeksforGeeks, System Testing - GeeksforGeeks, 2019). System testing tests the design and behavior of the system, as well as the client's prospects. For instance, system testing not only tests the complete booking workflow, from creating a reservation, updating availability, and generating guest records to producing the final bill but also tests the complete process of parking reservation, including booking a room with a parking space, confirming the parking space, and checking availability

in real-time. It will check the integration between the event booking system, billing system, and report generation to ensure that event details are accurately captured and reflected in the reports. For multi-language and currency functions, system testing will test that users can switch between languages and currencies seamlessly without disrupting their session or losing any previously entered data, and ensure that the final bill reflects the correct currency (m-kauppinen, 2024). This type of test is conducted after all modules have been integrated, checking end-to-end functionality, including usability, security, performance, and the data flow between components (GeeksforGeeks, System Testing - GeeksforGeeks, 2019).

Acceptance testing is a process used to evaluate whether a system meets user requirements, and business processes, allowing users and clients to establish the system's acceptability (Ahmad, 2023). Acceptance testing enables end users and clients to formally test user needs, requirements, and business processes to determine whether they should be accepted. For instance, Swiss Garden Hotel can conduct user acceptance testing (UAT) to allow the hotel to interact with the system for booking rooms, managing events, generating bills, and creating reports (Vijay, What is Acceptance Testing (A Complete Guide), 2023). For parking functions, guests can use the system to make a parking reservation to check for ease of use and accuracy of the parking status. For the chatbot, hotel staff and guests should interact with the chatbot to verify whether it provides useful, accurate, and timely responses. Feedback should be collected to ensure that the chatbot can enhance the user experience. For the multi-language and currency function, hotel staff should validate whether the system can accurately handle various payment methods like e-wallets, credit cards, and online payments in different currencies. The final receipt and report should reflect the correct currency (Checkout.com, n.d.). These are due to verify that the system operates as required. The acceptance testing also helps to check the system's ability to handle real-world scenarios such as making reservations simultaneously during peak seasons or generating large event reports. This type of test can be run by the user to perform specific tasks under normal operating conditions, and feedback is gathered. If the system passes acceptance testing, it is considered ready for production (Ahmad, 2023).

Validation is important because it can help product development teams ensure that their products meet customer needs and expectations. Validation can also help software developers identify and fix errors or direct other areas for improvement before the product is released (Insights, 2022). For example, without proper verification, the Swiss Garden Hotel may face serious issues such as double-booking rooms, incorrect billing, or slow response times for

generating reports. These issues may lead to guest dissatisfaction and reduced operational efficiency. These issues may lead to guest dissatisfaction and reduced operational efficiency. Validation ensures that the Swiss Garden Hotel Management System is functional and reliable to reduce the chances of system failures during critical operations like peak check-in/check-out periods or large events.

2.1.4 Software Evolution

Software evolution refers to the initial process of developing software, constantly updated for various reasons, such as adding new features or removing outdated functionalities (Engineering, 2024). Software evolution is a process of continuously updating and developing software to meet market needs (Khrupa, 2022). The system needs to constantly evolve to satisfy the users, market, and business requirements without compromising the integrity of the system's organization (Khrupa, 2022).

Besides that, the software evolution process is important in the whole development cycle (Khrupa, 2022). It helps keep the product up to date and fixes the current issues in the software (Khrupa, 2022). Furthermore, the latest updated software makes it more stable to reduce and defend the cyberattacks (Khrupa, 2022).

Moreover, there are several laws used for software evolution that need attention, which are the Law of Continuing Change, Law of Increasing Complexity, Law of Conservation of Organization Stability, and Law of Conservation of Familiarity (Engineering, 2024). The Law of Continuing Change means that any real-world reality software system will slowly become useless due to the continuous nature of environmental changes (Engineering, 2024). The Law of Increasing Complexity refers to the evolution of program changes that will make the whole structure more complex (Engineering, 2024).

Law of Conservation of Organizations Stability refers to the fact that the fact that during the life cycle of a program, the development program rate is mostly constant, and it is not relevant to the system development resources (Engineering, 2024). The Law of Conservation of Familiarity refers to the fact that the fact that during the effective life cycle of the program, the subsequent changes made are mostly constant (Engineering, 2024).

Lastly, software evolution is crucial for the Swiss Garden Hotel Management System as the hotel expands and customer demands evolve. The owner of Swiss Garden Hotel, Mr. David, should pay attention to the importance of the software evolution by adding new features such as an AI-advanced chatbot, multilingual and multi-currency support, customizable room support, parking fee support, and a loyalty program to enhance the overall user experience.

In conclusion, software evolution can enhance the usability and functionality of the Swiss Garden Hotel Management System to maintain a high level of customer satisfaction and operational efficiency over time

2.2 Functional and Non-Functional Requirements

Functional requirements are the key requirements that the system must include all fundamental facilities that the end user particularly requests, as these features are necessary under the terms and condition (Functional vs. Non Functional Requirements, 2024).

2.2.1 Functional Requirements

Table 2.2_1 Functional Requirements

Actor	Functional Requirement
Customer	The customer shall be able to register by entering the customer's details such as first and last name, date of birth, email address, and contact number to create an account.
	The customer shall be able to log in by entering the username and password to access the customer homepage and log out to terminate the session.
	The customer shall be able to view the rooms available on the homepage with room pictures, names, and details.
	The customer shall be able to chat with the AI-advanced chatbot if they face any issues.
	The customer shall be able to select their preferred language and currency shown on the website.
	The customer shall be able to select their preferred check-in and check-out dates with the number of guests.
	The customer shall be able to view the list of hotel's available rooms and details, as well as the price for each.
	The customer shall be able to book the room selected.
	The customer shall be able to select the number of room booked, breakfast and waterpark are optional.
	The customer shall be able to fill in the car's plate number if they want the parking facilities.
	The customer shall be able to see the summary of price details before making payment.
	The customer shall be able to fill in the booking information such as name, phone number, and email address to confirm the booking.

	The customer shall be able to make payment by selecting the payment method.
	The customer shall receive the booking confirmation letter after payment is made.
	The customer shall be able to view the event halls of the hotel with detailed information including floor plan and seating capacity.
	The customer shall be able to fill in the request form and submit if they want to book the event hall.
	The customer shall be able to view the restaurants available at the hotel with the restaurant details and food menus.
	The customer shall be able to view the recreational facilities with the activity details.
Admin	The admin shall be able to log in by entering the correct username and password to access the admin homepage.
	The admin shall be able to view the selections such as guest information, room/event report, and check the availability of rooms and event halls.
	The admin shall be able to view and print the customer details with their booking history.
	The admin shall be able to generate and print financial reports, including revenue breakdown and occupancy rates.
	The admin shall be able to view each room and event hall report with information such as the room occupancy and revenue of each month.
	The admin shall be able to make changes and modify the rooms or event halls' availability such as available, pending, or unavailable.
	The admin shall be able to make approval to the event hall request form such as approve, pending, or reject it.
	The admin shall be able to assign and manage room cleaning tasks by creating a task list for room cleaners and specifying room numbers.
	The admin shall be able to save the changes they made to the system.
Coordinator	The coordinator shall be able to view the current booking information on all lists including check-in and check-out dates, number of guests, contact details. and the room/hall arranged for each booking.

	The coordinator shall be able to arrange and manage the assignment of room numbers for each booking based on the customer's requirements.
	The coordinator shall be able to arrange bookings for event halls and review customer requests for event spaces.
	The coordinator shall be able to manage guest parking requests by verifying car plate numbers and ensuring availability in the hotel's parking facilities.
	The coordinator shall be able to contact the customer if they face any issues while processing the customer's booking.
Room Cleaner	The room cleaner shall be able to receive notifications for room cleaning or special requests.
	The room cleaner shall be able to update the cleaning status such as cleaned, or pending.
	The room cleaner shall be able to report issues or damages in the rooms to the admin.
Receptionist	The receptionist shall be able to manage check-in and check-out processes including calculating total cost and collecting money.
	The receptionist shall be able to assist customers with booking modifications and special requests.

2.2.2 Non-Functional Requirements

Non-functional requirements are the limitations that define the software's quality attribute (GeeksforGeeks, 2024). To gathering the non-functional requirements, we use the method observation to complete the table below:

Table 2.2_2 Non-Functional Requirements

Product Requirements

Types	Title	Manual system	New system
Performance requirements	Report Generation	Spend at least 30 minutes to generate the report.	Spend no more than 1 minute to generate the report.
	Payment Processing	Handled manually and may take 5 minutes to complete each payment.	Payment will be processed in less than 3 seconds.
	Verify user	Manuals verify each customer take several minutes to hours.	Verify the user in less than 3 seconds.
	Booking Processing	Booking confirmations require 15-20 minutes manually.	Bookings are processed and confirmed in 3 minutes.
	Data backup	No consistent backup process.	Automatically execute data backups every day within 30 minutes.
	Concurrent User Support	The manual system is limited to one user at a time.	Supports up to 2000 concurrent users.
	Notification Delivery	Send notification to customer via calls, taking about 5 minutes.	Notification will be sent automatically in 5 seconds.

	Response Time	Take several minutes to hours.	AI-chatbot should respond in less than 5 seconds.
	Form Submission Time	Filling and processing registration forms takes around 15 minutes	Registration forms are processed in less than 3 minutes.
Reliability Requirements	Transaction Accuracy	Manual calculations are prone to errors.	All financial transactions should be processed with 100% accuracy.
	System Downtime	The system may be unavailable for hours during maintenance.	System downtime should not exceed 5 minutes per day.
	System Maintenance	The entire system fails to operate during maintenance tasks.	System maintenance should not exceed 48 hours per month, and the system can continue operating during maintenance.
	Data Validation	No automated checks, requiring admin to verify registration details manually.	The system automatically validates input fields by the customer.

This table compares the manual system and the new system, which shows the new system was more efficient and had better accuracy.

3.0 Logical Design

3.1 Use Case Diagram

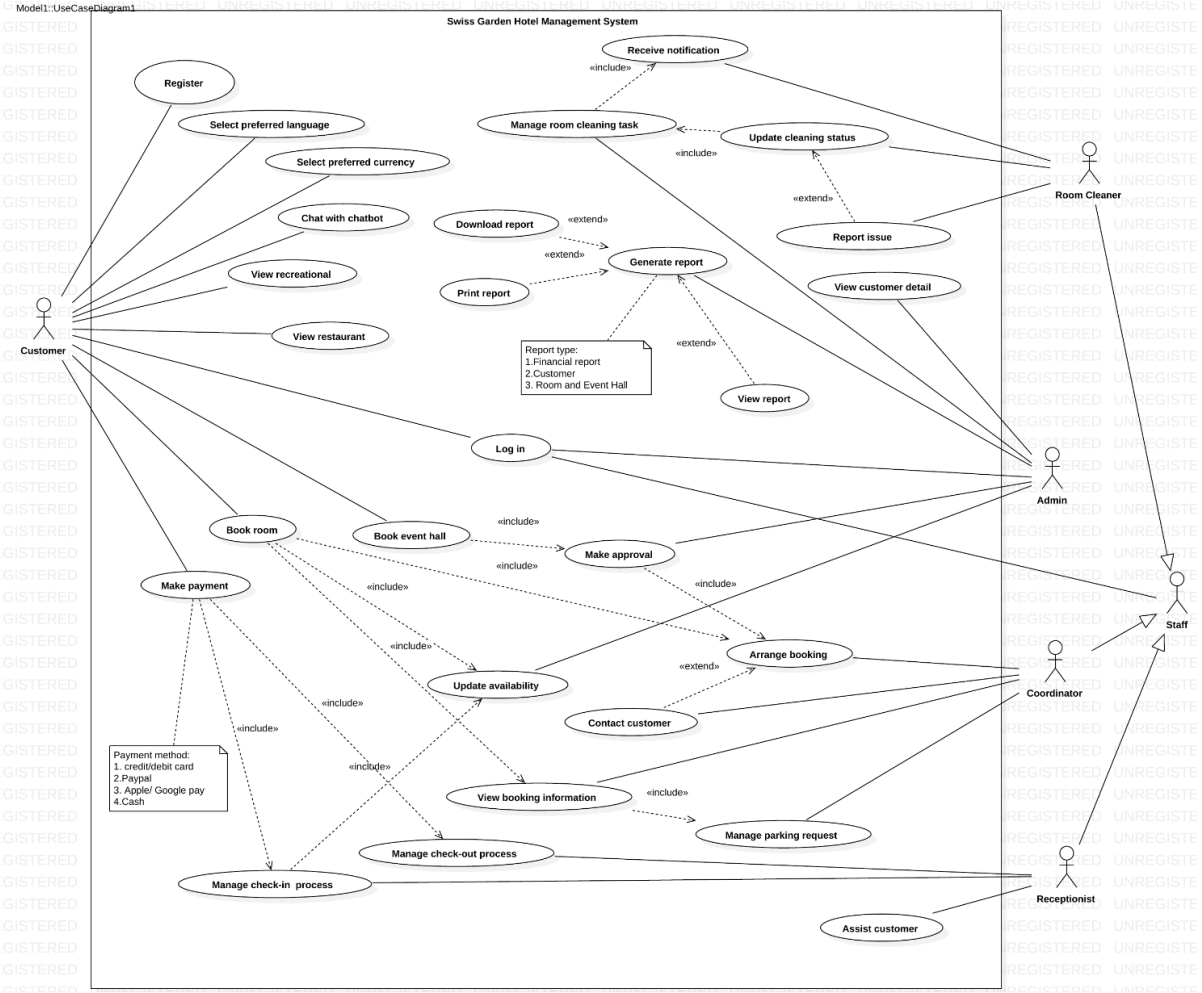


Figure 3.1_1 Use Case Diagram

3.2 Use Case Description

UC1: Register

Actor: Customer

Description: Customer registers an account in the hotel management system.

Precondition:

- System is accessible

Main Flow:

1. Customer navigates to the registration page
2. System shows the registration form.
3. Customer enters their personal details.
4. System validates the information.
5. System creates an account for the customer.

Alternative Flows:

- A1: Invalid Information Provided

1. System displays an error message to indicate missing or incorrect information.
2. Customer corrects the information.
3. Returns to step 4 of the main flow.

Postconditions:

- Customer account is created.
- Customer receives a confirmation notification.

UC2: Log in

Actor: Customer, Staff, Admin

Description: Users log into the hotel management system to access their own services.

Precondition:

- User has an account
- System is accessible

Main Flow:

1. User navigates to the log-in page.
2. System displays the login page.
3. User enters their username and password.
4. System validates credentials.
5. System grants access based on the user's role.

Alternative Flows:

- A1: Invalid Credentials
    1. System displays the error message.
    2. User re-enters credentials.
    3. Returns to step 4 of main flow.

Postconditions:

- User is logged into the system.
- - Appropriate access level is granted.

UC3: Select Preferred Language

Actor: Customer

Description: Customer selects their preferred language for system use.

Precondition:

- Customer is logged in.

Main Flow:

1. Customer navigates to the navigation bar.
2. System displays language options.
3. Customer selects their preferences.
4. System confirms the changes.

Alternative Flows:

- A1: Preferences Not Confirmed
    1. System shows an error message indicating that the preferences have not been selected.
    2. Customer re-selects their preferences.
    3. Returns to step 4 of the main flow.

Postconditions:

- Language preferences are updated.

UC4: Select Preferred Currency

Actor: Customer

Description: Customer selects their preferred currency for system use.

Precondition:

- Customer is logged in.

Main Flow:

1. Customer navigates to the navigation bar.
2. System displays currency options.
3. Customer selects their preferences.
4. System confirms the changes.

Alternative Flows:

- A1: Preferences Not Confirmed
    4. System shows an error message indicating that the preferences have not been selected.
    5. Customer re-selects their preferences.
    6. Returns to step 4 of the main flow.

Postconditions:

- Currency preferences are updated.

UC5: Chat with Chatbot

Actor: Customer

Description: Customer interacts with a chatbot for assistance.

Precondition:

- Customer is logged in.

Main Flow:

1. Customer initiates a chat with the chatbot.
2. System displays the chat interface.
3. Customer asks a question.
4. Chatbot provides an appropriate response.

Alternative Flows:

- A1: Chatbot Cannot Answer

1. Chatbot informs the customer it cannot answer the question.
2. Customer is given the option to contact customer support.
3. Customer chooses to proceed to customer support or ends the chat.

Postconditions:

- Customer receives assistance through the chatbot.

UC6: Book Room

Actor: Customer

Description: A customer books a room in the hotel.

Precondition:

- Customer is logged in.
- Rooms are available for booking.

Main Flow:

1. Customer navigates to the room booking section.
2. System displays available rooms.
3. Customer selects a room and desired dates.
4. Customer fills in booking information
5. System calculates the price and displays booking details.
6. Customer confirms the booking.
7. System processes the payment and confirms the booking.

Alternative Flows:

- A1: Room Not Available

1. System displays a message indicating unavailability.
2. Customer selects a different room or dates.
3. Returns to step 3 of the main flow.

Postconditions:

- Room is successfully booked.
- Customer receives a booking confirmation.

UC7: Make Payment

Actor: Customer

Description: A customer makes a payment for a booked services.

Precondition:

- Customer has a valid booking or service.

Main Flow:

1. Customer navigates to the payment section.
2. System displays payment options (credit/debit card, PayPal, Apple/Google Pay, cash).
3. Customer selects a payment method.
4. System processes the payment.
5. System confirms the payment and provides a receipt.

Alternative Flows:

- A1: Payment Failure

1. System displays an error message.
2. Customer selects a different payment method or retries.

Postconditions:

- Payment is completed.

- Booking or service is confirmed.

UC8: View Restaurant

Actor: Customer

Description: Customer views information about the hotel restaurant.

Precondition:

- Customer is logged in.

Main Flow:

1. Customer navigates to the restaurant section.
2. System displays restaurant details.

Alternative Flows:

- A1: Restaurant Information Not Available

1. System displays a message indicating that restaurant details are unavailable.
2. Customer may contact support for further information.

Postconditions:

- Customer views restaurant information.

UC9: View Recreational

Actor: Customer

Description: Customer views available recreational activities.

Precondition:

- Customer is logged in.

Main Flow:

1. Customer navigates to the recreational section.

2. System displays available recreational activities.

Alternative Flows:

- A1: No Recreational Activities Available

1. System displays a message indicating no recreational activities are available.

2. Customer may contact support for alternatives.

Postconditions:

Customer views the recreational options

UC10: Manage Check-in Process

Actor: Receptionist, Customer

Description: The receptionist manages the check-in processes for customers.

Precondition:

- Customer has a valid booking

Main Flow:

1. Customer arrives at the hotel.
2. Receptionist verifies the booking details.
3. Receptionist completes the check-in process and provides room access.

Alternative Flows:

- A1: Booking Not Found

1. Receptionist informs the customer.
2. Customer provides additional information.
3. Receptionist re-checks the booking details.

Postconditions:

- Customer is checked in successfully.
- System updates room availability.

UC11: Manage Check-out Process

Actor: Receptionist, Customer

Description: The receptionist manages the check-out processes for customers.

Precondition:

- Customer has a valid booking

Main Flow:

1. Receptionist verifies the booking details.
2. Receptionist verifies the room status.
3. Receptionist finalizes the check-out and processes any outstanding payments.

Alternative Flows:

- A1: Booking Not Found

1. Receptionist informs the customer.
2. Customer provides additional information.
3. Receptionist re-checks the booking details.

Postconditions:

- Customer is checked out successfully.
- System updates room availability.

UC12: Receive Notification

Actor: Room Cleaner

Description: Room cleaner receives notification for the cleaning task.

Precondition:

- Room cleaner is logged in.

Main Flow:

1. System sends a notification.
2. Room cleaner views the notification.

Alternative Flows:

- A1: Notification Not Received

1. Room cleaner refreshes or re-checks the notification section.
2. If still not received, the room cleaner contacts the admin or the system administrator for assistance.

Postconditions:

- Room cleaner receives and views relevant notifications.

UC13: Update Cleaning Status (Base Use Case)

Actor: Room Cleaner

Description: Room cleaner updates the cleaning status of a room.

Precondition:

- Room is assigned for cleaning.

Main Flow:

1. Room cleaner navigates to the room status section.
2. System displays room details.
3. Room cleaner updates the status to reflect the current cleaning progress.

Alternative Flows:

- A1: Unable to Update Status

1. System shows an error message.
2. Room cleaner re-attempts to update the status or contacts the admin for assistance.

Postconditions:

- Room cleaning status is updated.

UC14: Report Issue (Extended Use Case)

Extends: Update Cleaning Status

Actor: Room Cleaner, Admin

Description: Room cleaner reports any issue or damage in rooms or facilities.

Extension Points:

- Extends the base use case when the user reports the issue or damage.

Precondition:

- User is logged in.

Main Flow:

1. User navigates to the issue reporting section.
2. System displays the issue reporting form.
3. User enters details of the issue or damage.
4. System logs the issue and notifies relevant personnel.

Alternative Flows:

- A1: Unable to Submit Report

1. System shows an error message due to missing required information.
2. User corrects the error and submits the report.

Postconditions:

- Issue or damage is reported and logged.
- Relevant personnel are notified.

UC15: View Customer Detail

Actor: Admin

Description: Admin views customer details for reference or assistance.

Precondition:

- Admin is logged.

Main Flow:

1. Admin navigates to the customer details section.
2. System displays the customer information.

Alternative Flows:

- A1: Customer Information Not Found

1. System shows an error message indicating no customer data is available.
2. Admin checks the database or updates the issue to support.

Postconditions:

- Customer details are viewed by the user.

UC16: Manage Room Cleaning Task

Actor: Admin, Room Cleaner

Description: Admin manages the cleaning status of rooms.

Precondition:

- Room is assigned for cleaning.

Main Flow:

1. Admin navigates to the cleaning management section.
2. System displays the list of rooms assigned for cleaning.
3. Room cleaner updates the cleaning status such as cleaned, or pending.
4. System saves the updated status.

Alternative Flows:

- A1: Unable to Update Status

1. System displays an error indicating that the status cannot be updated.
2. Admin or room cleaner re-attempts the update or contacts support.

Postconditions:

- Room cleaning status is updated in the system.

UC17: Generate Report (Base Use Case)

Actor: Admin

Description: Admin generates different types of reports such as financial, customer, and room/event hall reports.

Precondition:

- Admin is logged in with the necessary permissions.

Main Flow:

1. Admin navigates to the report generation section.
2. System displays report options such as financial, customer, and room/event hall.
3. Admin selects the type of report to generate.
4. System retrieves the relevant data and generates the report.
5. Admin views, downloads, or prints the report.

Alternative Flows:

- -A1: No Data Available

1. System displays a message indicating no data is available for the selected report type.
2. User selects a different report type or date range.

Postconditions:

- Report is generated and available for viewing or printing.

UC18: Download Report (Extended Use Case)

Extends: Generate Report

Actor: Admin

Description: Admin downloads a generated report.

Extension Points:

- Extends the base use case when the admin downloads the report.

Precondition:

- Report is generated.

Main Flow:

1. Admin selects the download option for the report.
2. System downloads the report to the user's device.

Alternative Flows:

- A1: Download Failure

1. System shows an error indicating the report could not be downloaded.
2. Admin retries the download.

Postconditions:

- Report is downloaded to the admin’s device.

UC19: Print Report (Extended Use Case)

Extends: Generate Report

Actor: Admin

Description: Admin prints a generated report.

Extension Points:

- Extends the base use case when the admin prints the report.

Precondition:

- Report is generated.
- Printer is connected and accessible

Main Flow:

1. User selects the print option for the report.
2. System sends the report to the printer.
3. Report is printed.

Alternative Flows:

- A1: Printer Not Connected

1. System displays a message indicating no printer is connected.
2. Admin connects the printer and retries.

- A2: Print Failure

1. System detects an issue with the printer.

2. Admin resolves the issue or contacts technical support.

Postconditions:

- Report is printed successfully.

UC20: View Report (Extended Use Case)

Extends: Generate Report

Actor: Admin

Description: Admin views a generated report.

Extension Points:

- Extends the base use case when the admin view the report.

Precondition:

- Report is generated.

Main Flow:

1. Admin selects the view option for the report.
2. System displays the report.

Alternative Flows:

- A1: Report Not Available

1. System shows an error indicating the report is unavailable.
2. Admin retries the action.

Postconditions:

- Report is viewed by the admin.

UC21: Book event hall

Actor: Customer

Description: Customer can book the event hall

Precondition:

- Event hall details such as availability and pricing are accessible in the system.

Main Flow:

1. Customer navigates to the "Event Hall Booking" section.
2. System displays available event halls and booking options.
3. Customer selects a hall, and date, and fills in booking details.
4. Customer submits the request.

Alternative Flows:

- A1: Event Hall Unavailable:

1. The system notifies the customer of unavailability for the selected date/time.
2. The customer selects a different event hall or adjusts the booking date/time.
3. The process returns to Step 4 of the Main Flow.

Postconditions:

- The event hall is successfully booked.
- The booking details are stored in the system.
- The customer receives a confirmation notification.

UC22: Make Approval

Actor: Admin, Customer

Description: Admin approves requests such as event hall.

Precondition:

- Request is submitted and pending approval.

Main Flow:

1. Admin navigates to the approval section.
2. System displays pending requests.
3. Admin reviews and approves the request.

Alternative Flows:

- A1: Request Rejected

1. Admin rejects the request due to certain conditions (e.g., unavailability of resources).
2. System sends a rejection notice to the customer or requester.

Postconditions:

- Request is approved or rejected.

UC23: Update Availability

Actor: Admin, Receptionist, Customer

Description: Admin updates the availability status of rooms.

Precondition:

- Admin is logged in.

Main Flow:

1. Admin navigates to the availability management section.
2. System displays the current status of rooms.
3. Admin updates the availability as required.
4. System saves the changes and updates the availability information.

Alternative Flows:

- A1: Unable to Update Availability
- System displays an error message indicating the availability update failed.
- Admin re-attempts the update or contacts support.

Postconditions:

- Room availability status is updated.

UC24: View Booking Information

Actor: Coordinator, Customer

Description: Coordinator view details of their bookings.

Precondition:

- Booking is made and confirmed.

Main Flow:

1. Coordinator navigates to the booking information section.
2. System displays booking details.

Alternative Flows:

- A1: Booking Information Not Found

1. System shows an error indicating no booking information available.
2. Coordinator re-checks the booking details or contacts customer service.

Postconditions:

- Booking information is viewed by the Coordinator.

UC25: Arrange Booking (Base Use Case)

Actor: Coordinator, Admin

Description: Coordinator arranges a booking for the customer, such as for rooms or event halls.

Precondition:

- Customer request for booking is received.

Main Flow:

1. Coordinator navigates to the booking management section.
2. System displays available options.
3. Coordinator will arrange the room or event hall.
4. If the room or event hall has problem, the coordinator will rearrange the booking for the customer

Alternative Flows:

- A1: No Available Rooms/Event Halls

1. System displays no availability.

2. Coordinator informs the customer or re-arranges the booking.

Postconditions:

- Booking is arranged successfully

UC26: Contact Customer (Extended Use Case)

Extends: Arrange booking

Actor: Coordinator

Description: Coordinator contacts the customer if they face any issues while processing the customer's booking.

Extension Points:

- Extends the base use case when the Coordinator contacts the customer.

Precondition:

- Customer contact information is available.

Main Flow:

1. Staff navigates to the customer contact section.
2. System displays customer contact details.
3. Staff contacts the customer.

Alternative Flows:

- A1: Contact Information Missing

1. System displays a message indicating contact information is unavailable.
2. Staff contacts admin or uses other means to reach the customer.

Postconditions:

- Customer is contacted successfully.

UC27: Manage Parking Request

Actor: Coordinator

Description: Coordinator manages parking requests from customers.

Precondition:

- Customer has made a parking request.

Main Flow:

1. Coordinator navigates to the parking management section.
2. System displays pending parking requests.
3. Coordinator allocates parking space and updates the status.

Alternative Flows:

- A1: Parking Space Unavailable

1. Coordinator informs the customer about the lack of parking availability.
2. Coordinator arranges for alternate parking or escalates the issue.

Postconditions:

-- Parking request is managed successfully.

UC28: Assist Customer

Actor: Receptionist

Description: Receptionist assists the customer with various requests or inquiries.

Precondition:

- Customer has a request or inquiry.

Main Flow:

1. Customer contacts staff for assistance.
2. Staff listens to the request or inquiry.
3. Staff provides the necessary assistance or information.

Alternative Flows:

- A1: Unable to Assist

1. Staff informs the customer that the request cannot be fulfilled.
2. Staff suggests alternative solutions or escalates the issue.

Postconditions:

- Customer is assisted successfully.

Non-functional requirement

1. Performance:

- The system should process orders and user requests within 5 seconds.
- Data input and output for processing payroll and generating reports should occur in less than 10 seconds.
- The status of tasks, orders, or requests should be updated in real time to ensure users can track progress efficiently.

2. Security:

- All sensitive data including payment details, employee information, and payroll details, must be encrypted during storage.
- Users must authenticate using multi-factor authentication (MFA) before accessing any system features.
- Payment transactions should be sent through a secure and compliant gateway.

3. Usability:

- The system should provide a simple, intuitive, and responsive interface, suitable for the website.
- The system should display clear and helpful error messages in case of user input errors.

4. Reliability:

- All transactions and data entries should be backed up in real time to ensure no data loss in case of system failures.
- The system should automatically recover from failures like server crashes, and network disruptions with minimal user disruption.

5. Scalability:

- The system should handle peak usage periods without performance degradation.

- The system should support multiple instances to allow different departments or clients to run within the same infrastructure.

6. Privacy:

- Personal information collected from users should comply with privacy laws such as GDPR.
- The system should comply with relevant data protection and privacy regulations to ensure proper handling of customer and employee data.

3.3 Class Diagram

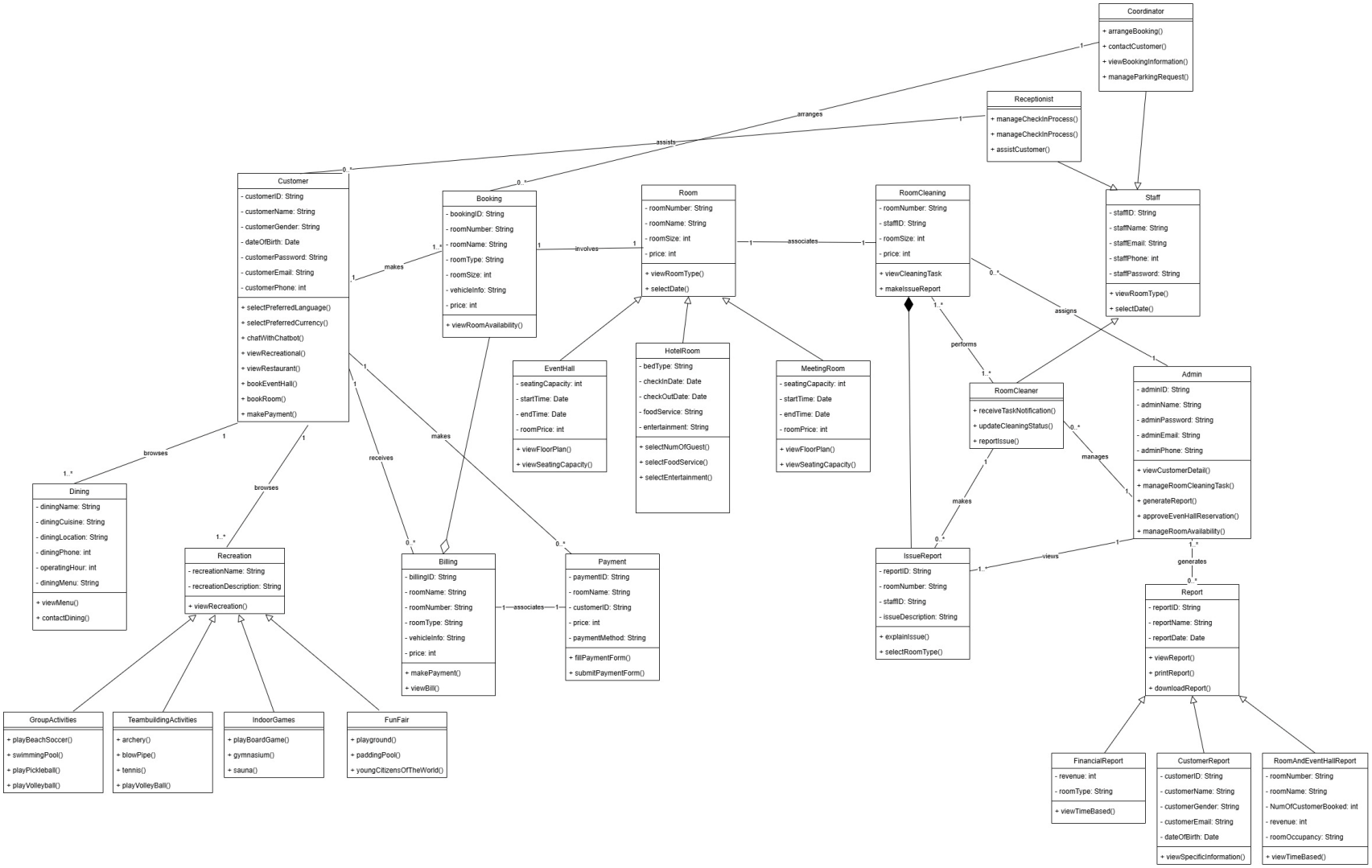


Figure 3.3_1 Class Diagram

4.0 Physical Design (Wire-framing)

Figure 4.0_1 Home Page without Login

Before logging in, users can change the language and currency rate using the 'Language & Currency' button. Sample room pictures and descriptions are displayed, with a 'Book Now' button for instant booking. A chatbot icon provides information about the Swiss Garden Hotel. At the top right, the 'Login/Sign Up' button allows users to log in or create a new account.

Figure 4.0_2 Home Page after Login

After logging in, users gain access to the 'Event Reservation' button to book event halls or ballrooms at the Swiss Garden Hotel, and the 'Room Reservation' button to view and book suitable rooms. The 'Dining' button enables users to explore available restaurants, while the 'Recreation' button showcases entertainment activities offered at the hotel.

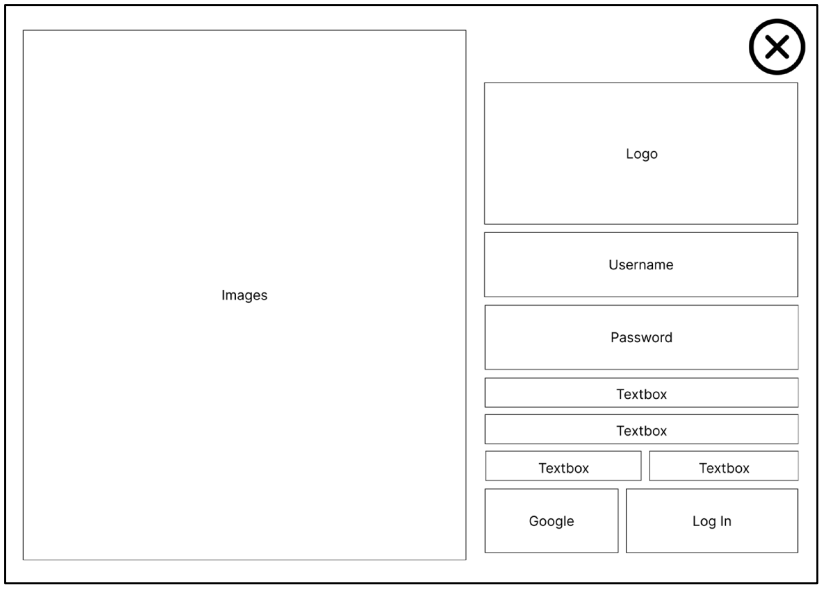


Figure 4.0_3 Login Page

On the login page, users can enter their username and password or log in with an existing Google account. A special login button is available for admin and staff. 'Register' and 'Forgot Password' buttons are also displayed on this page.

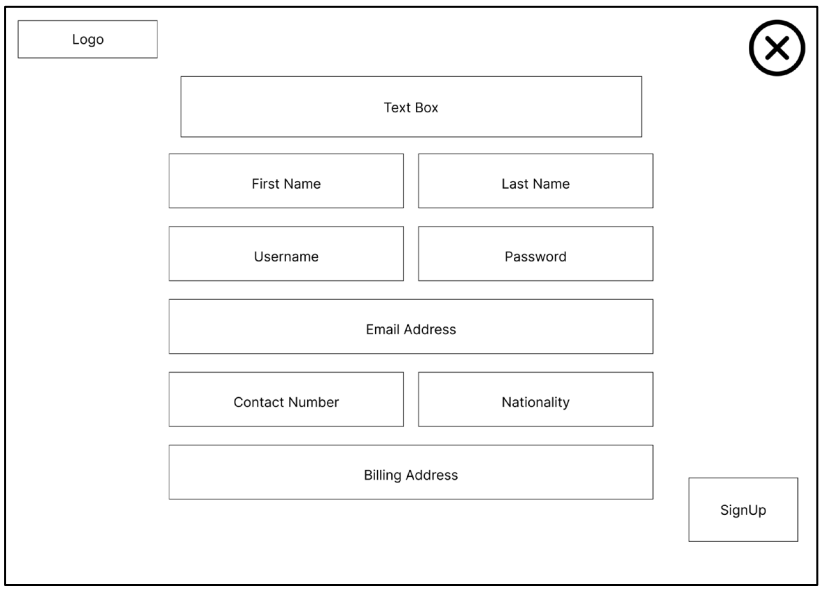


Figure 4.0_4 Registration Page

The registration page enables users to enter their personal information to create a new account. Users shall fill in all boxes shown on the page including their first and last name, username, password, email address, contact number, nationality, and billing address. After filling in, they must click the “SignUp” button to complete the registration process.

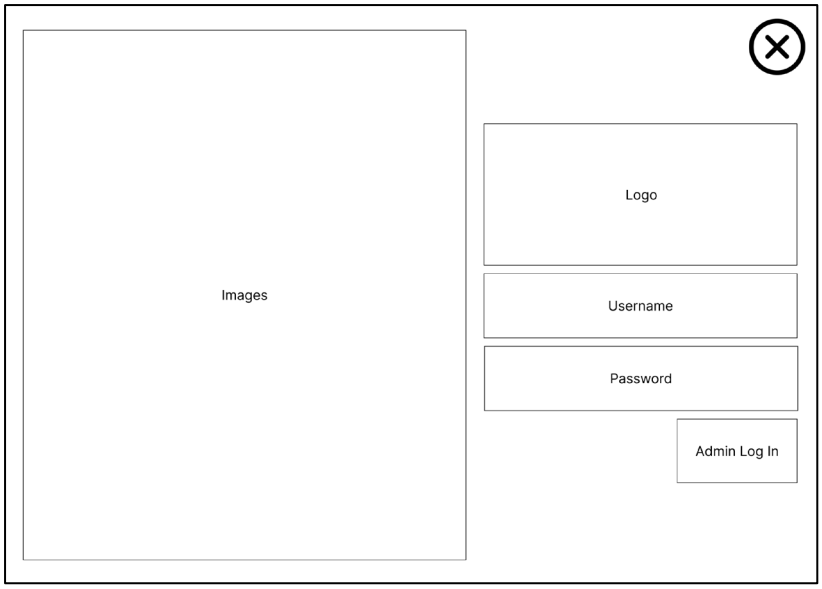


Figure 4.0_5 Admin Login Page

Admins must enter their username and password to log in to the system. After logging in with the correct preset username and password, they can access all functions available on the website.

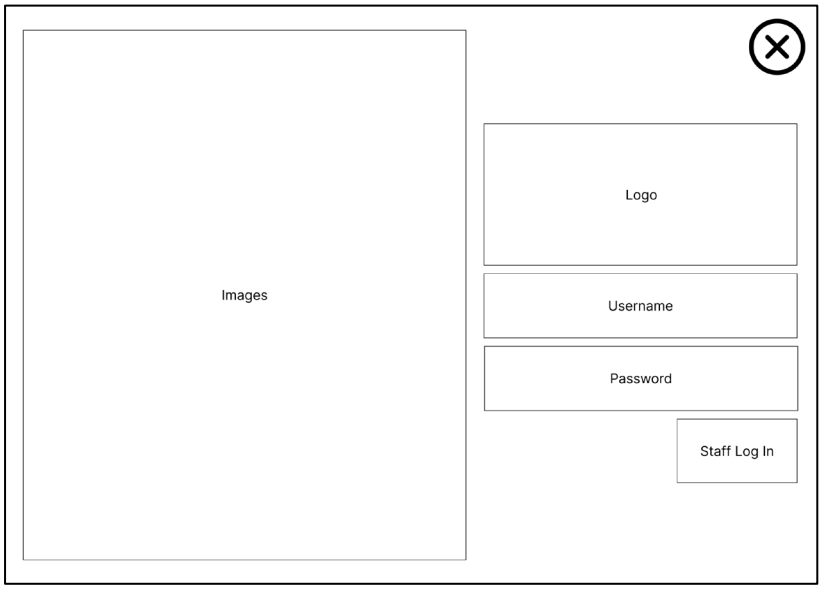


Figure 4.0_6 Staff Login Page

Staff (receptionists, room cleaners, and coordinators) must enter their username and password to log in to the system. After logging in with the correct preset username and password, they can access all functions available on the website.

Figure 4.0_7 Searching Page

This page allows users to enter their check-in and check-out dates and specify the number of guests in the room.

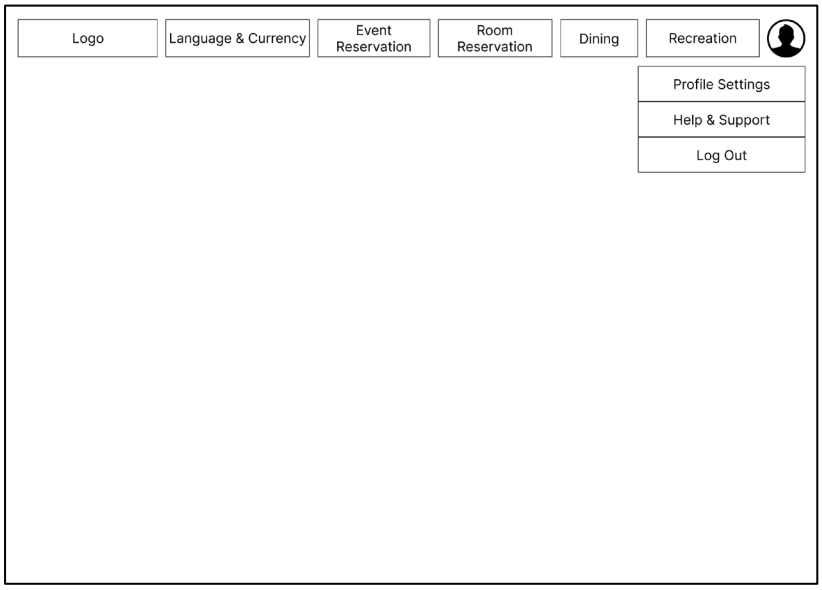


Figure 4.0_8 Profile Icon

After clicking the user icon in the top right corner, a menu appears with three options: Profile Settings, Help & Support, and Log Out.

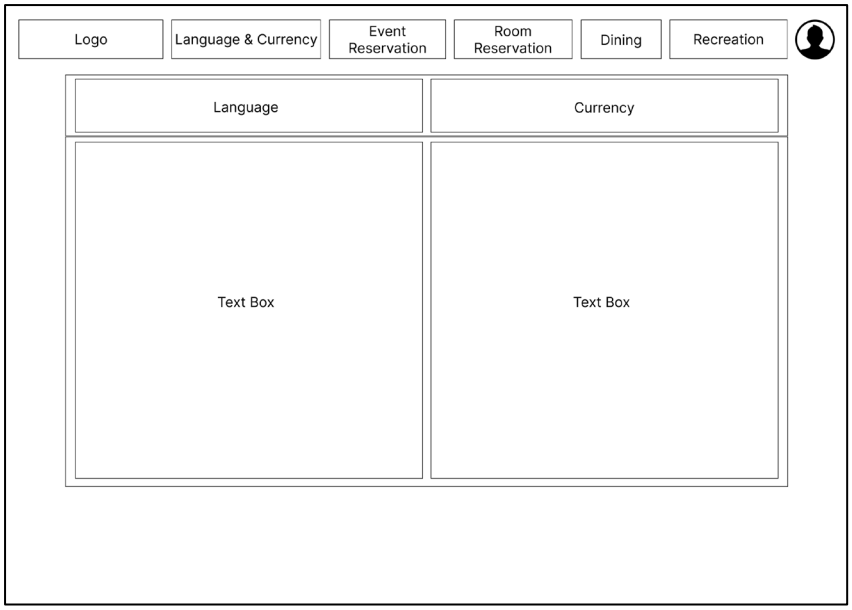


Figure 4.0_9 Language and Currency

This page allows users to change the system's language and currency. The users might choose their preferred language and currency.

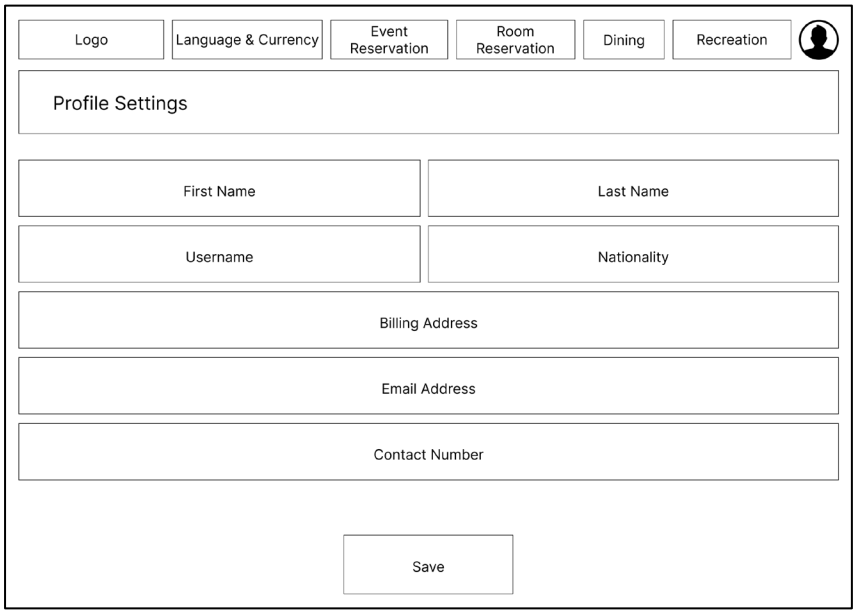


Figure 4.0_10 Profile Page

This page displays the user's personal details (E.g. first and last name, username, nationality, and email address) and allows them to make changes. After making changes, the user can select to save the changes or just go to another page to discard changes.

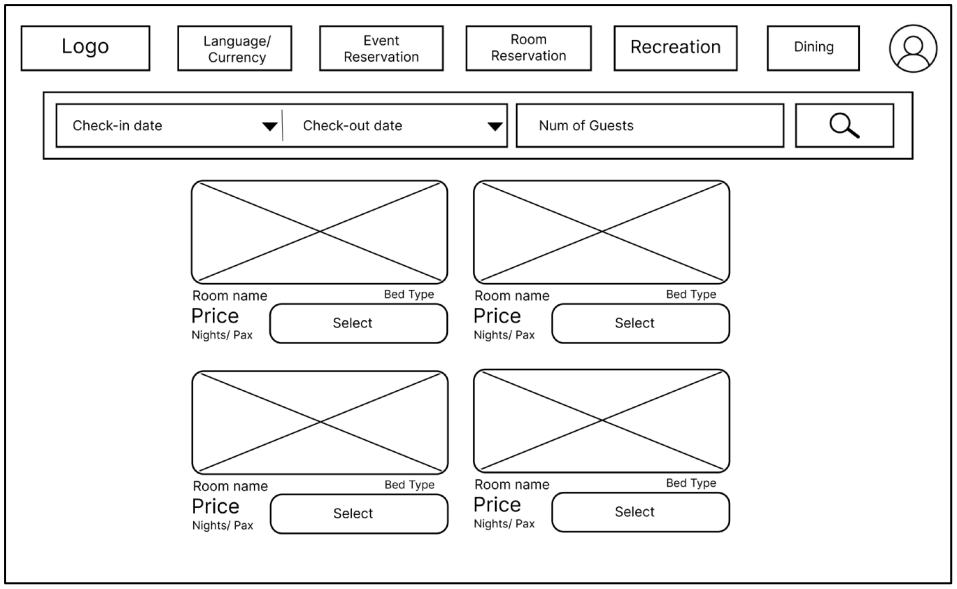


Figure 4.0_11 Booking Main Page

This page allows users to select the room they wish to stay in during their trip to the hotel.

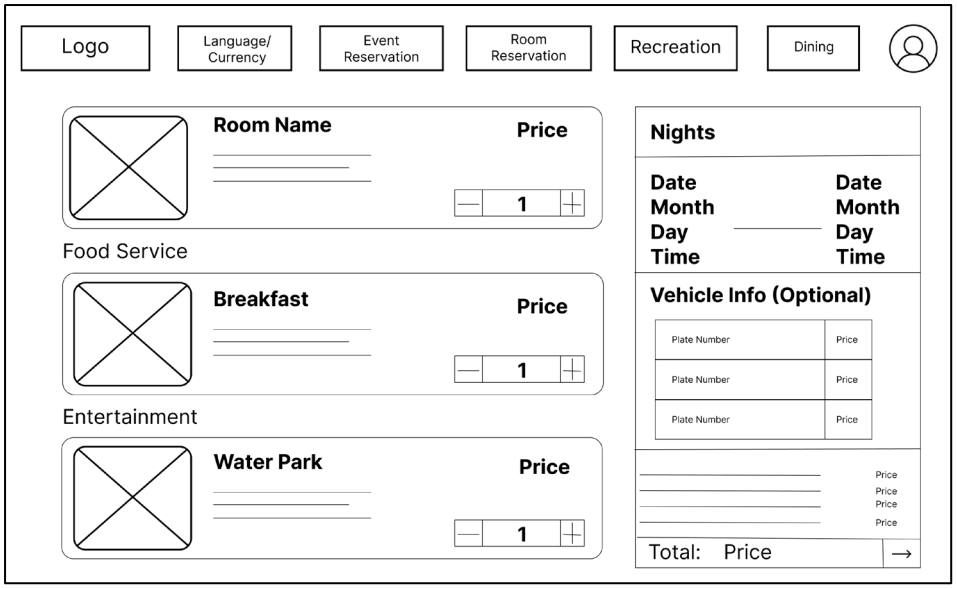


Figure 4.0_12 Room Settlement Page

After selecting their desired room type, users will be prompted to this page to choose room volume, food service, and entertainment options. Stay information and prices will be displayed on the right side of the page.

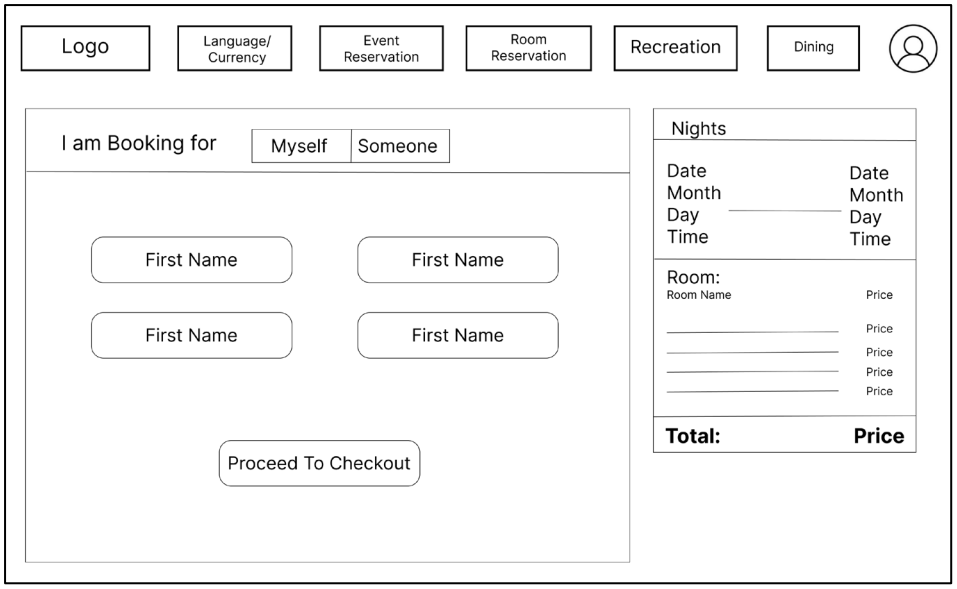


Figure 4.0_13 Proceed to checkout page

After completing room selection, users will need to enter their personal information before proceeding to checkout.

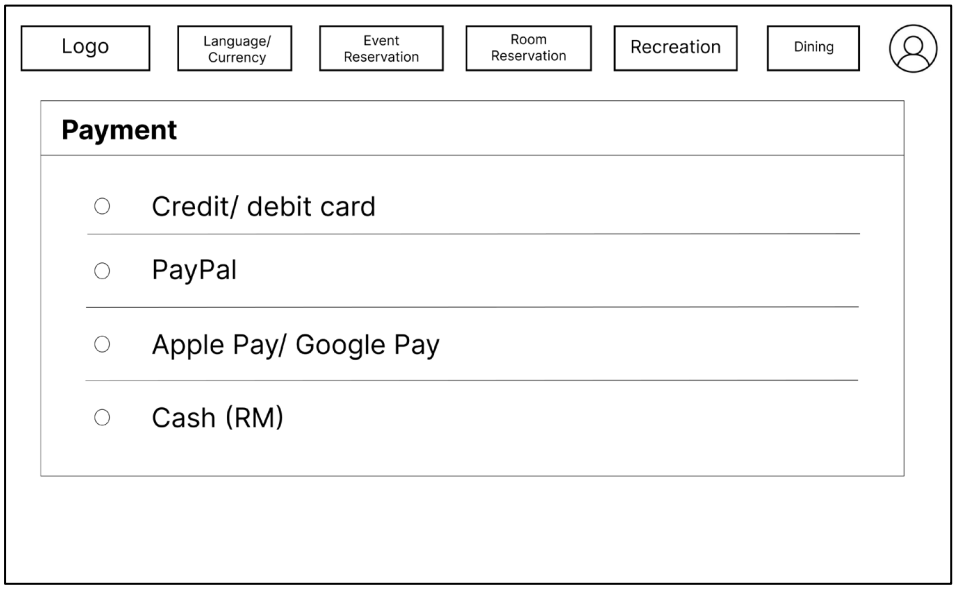


Figure 4.0_14Payment method page

This page allows users to select their preferred payment method to settle the bill. There are a total of four online secured payment methods that supported by the developed system.

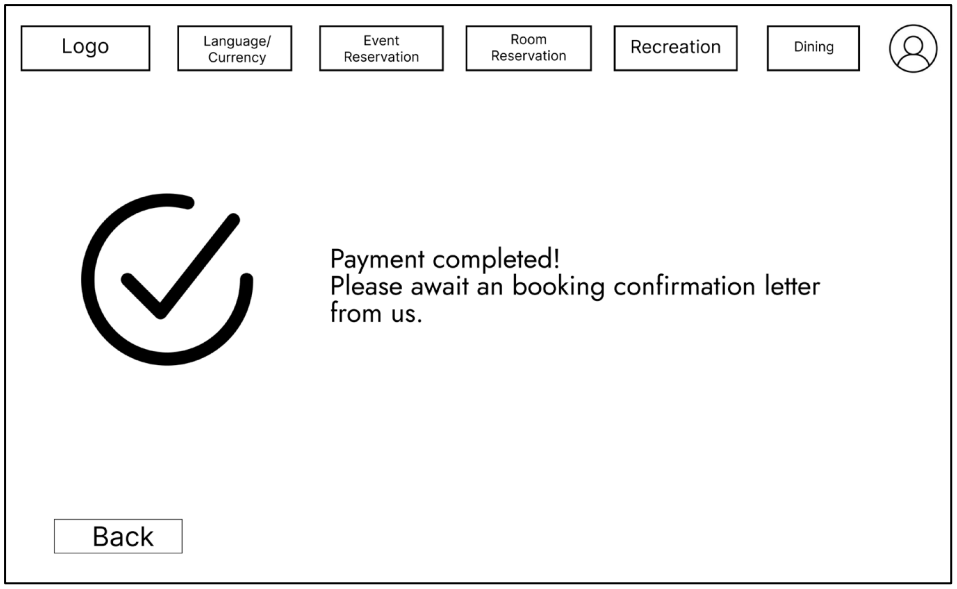


Figure 4.0_15 Done Payment Page

After users complete the payment, this page will be prompted to notify them that the payment has been successfully made.

Dining page

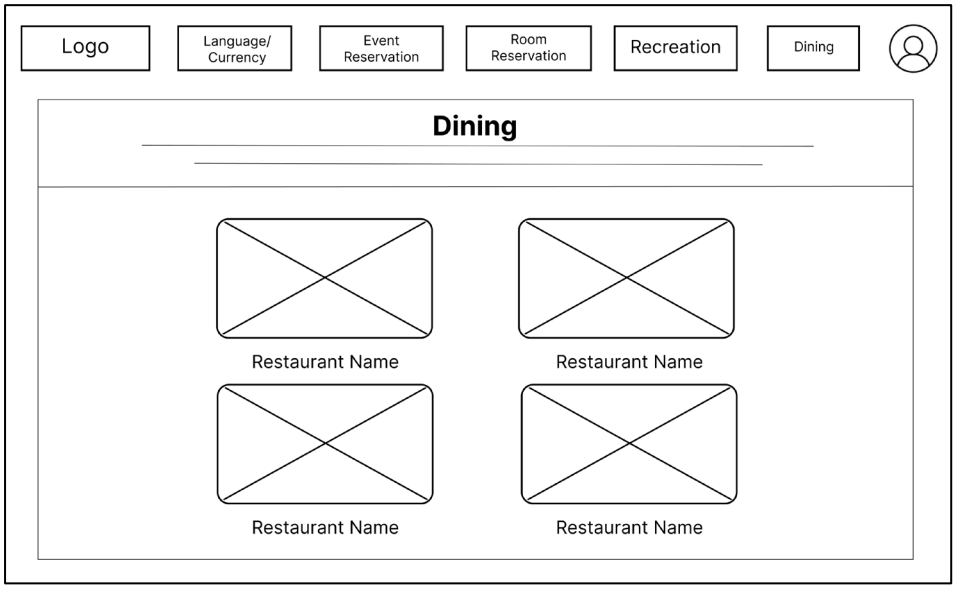


Figure 4.0_16 Dining Page

This page displays the restaurants available at the hotel. There are the restaurant names and images for each restaurant, and the customers might select one restaurant to view the details.

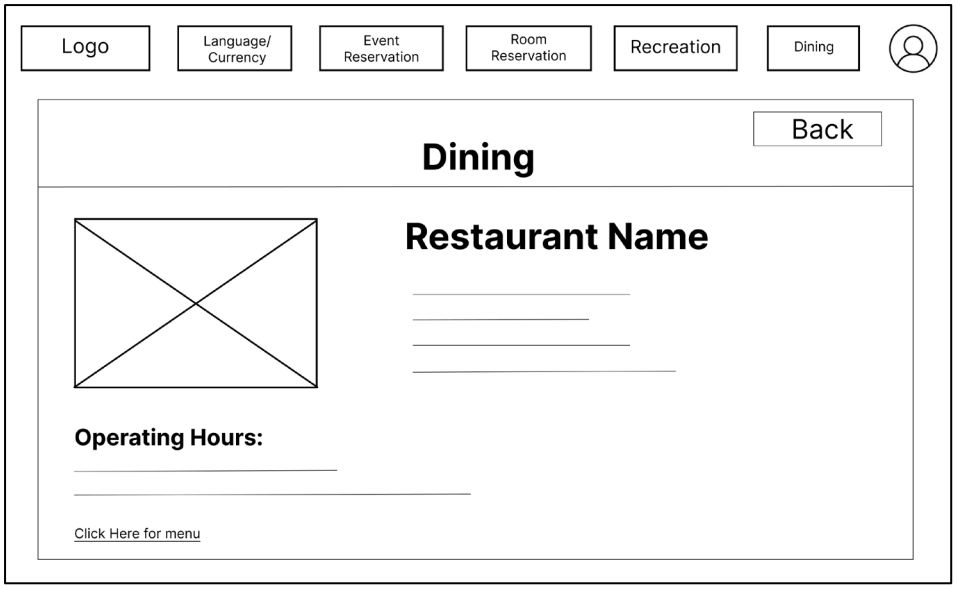


Figure 4.0_17 Restaurant Page

This page provides detailed information about a specific restaurant.

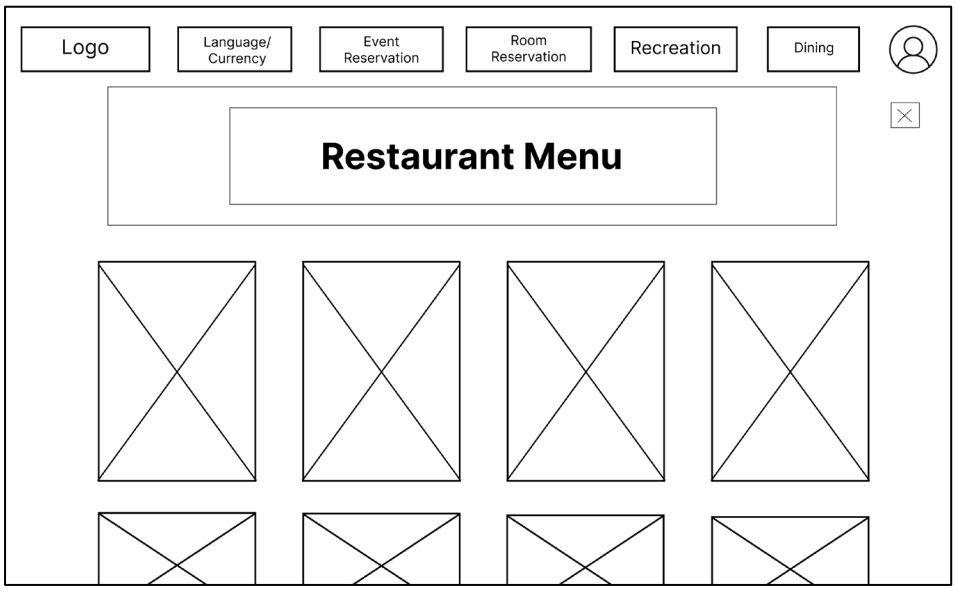


Figure 4.0_18 Restaurant Menu Page

This page displays the menu for a specific restaurant.

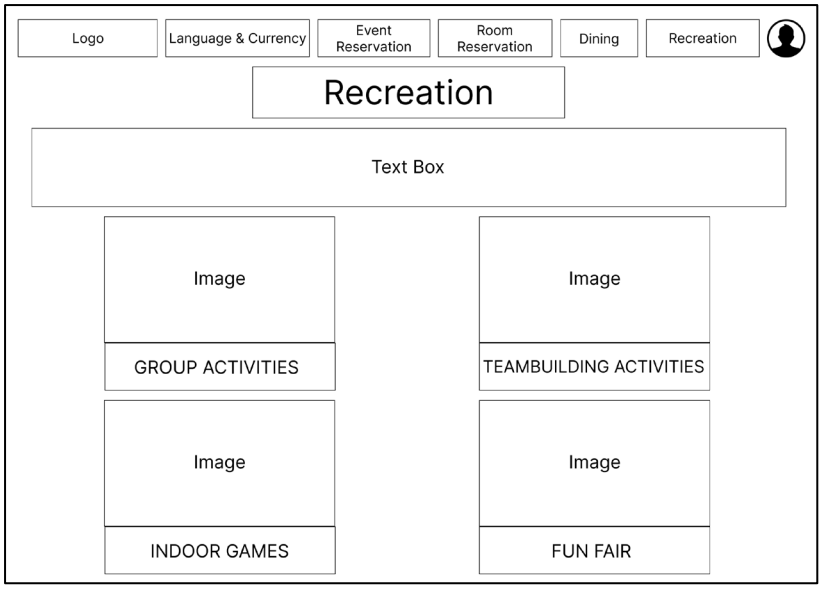


Figure 4.0_19 Recreation Home Page

This page displays the recreational activities and entertainment available at the hotel.

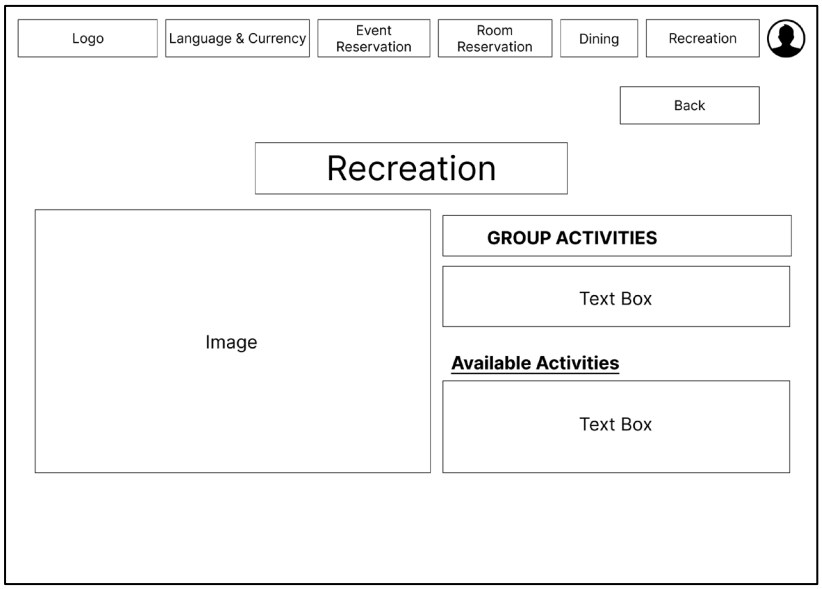


Figure 4.0_20 Recreation Details Page

This page provides detailed information about a specific recreation activity or entertainment option.

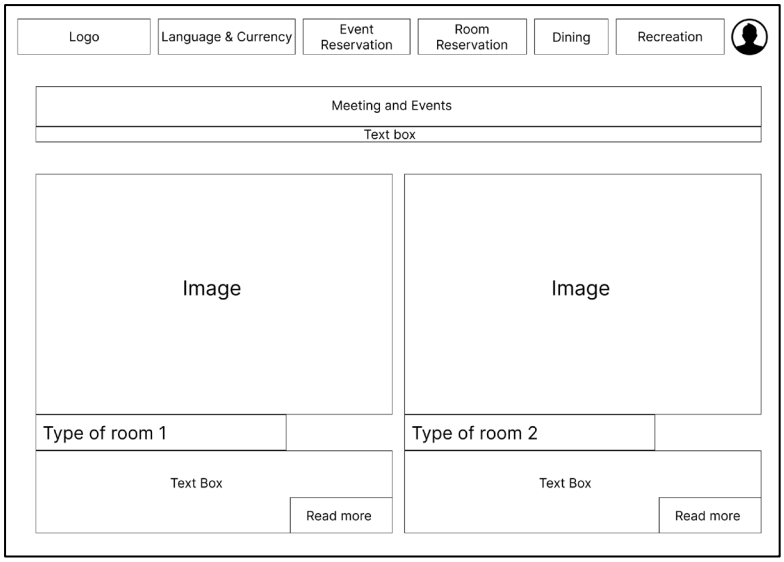


Figure 4.0_21 Meetings and Events Page

This page allows users to view the types of meeting rooms available at the hotel, with images and descriptions provided. A 'Read more' button offers detailed information about each specific room.

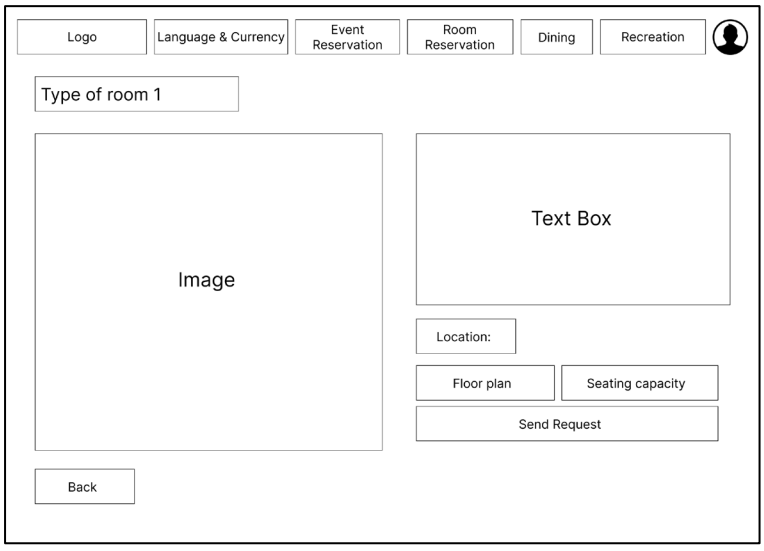


Figure 4.0_22 Details of Meeting Room

This page provides users with detailed information about each specific room, including location, floor plan, and seating capacity, to enhance their understanding. A 'Send Request' button allows users to apply for the meeting room.

Figure 4.0_23 Floor Plan

This page allows users to view the floor plans for event halls and meeting rooms.

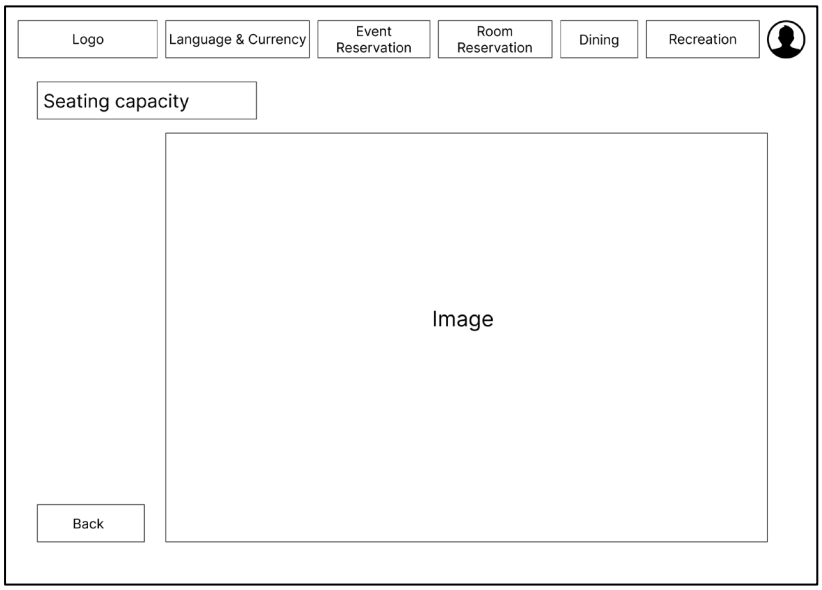


Figure 4.0_24 Seating Capacity

This page allows users to view the floor plans for event halls and meeting rooms.

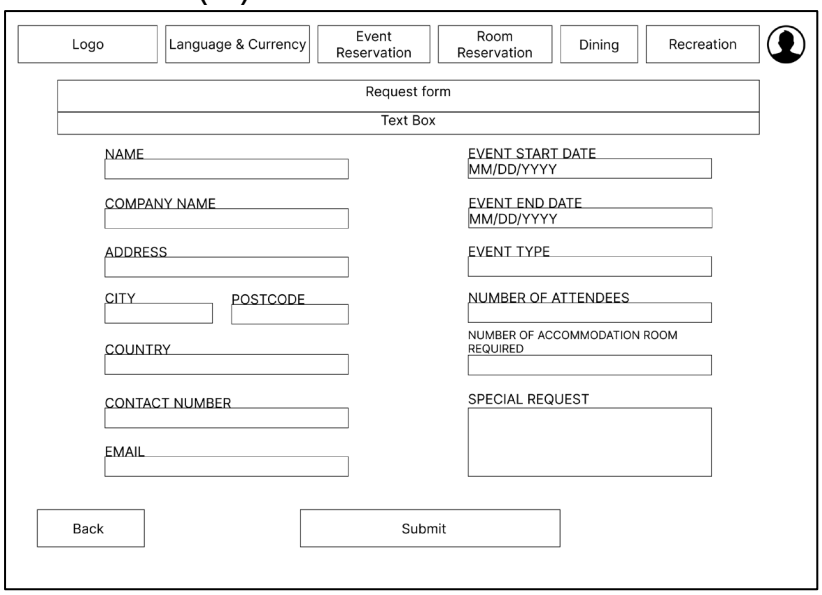


Figure 4.0_25 Request Form

This page lets users enter their personal and event details to apply for meeting rooms and event halls.

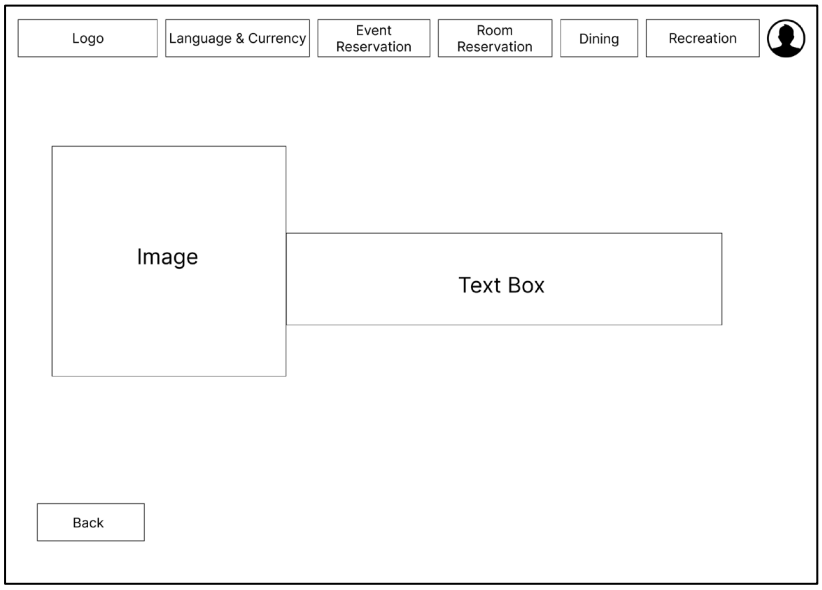


Figure 4.0_26 Done Submission Page

This page appears after users click the 'Submit' button on the request form, confirming that the form has been successfully submitted.

Admin action page

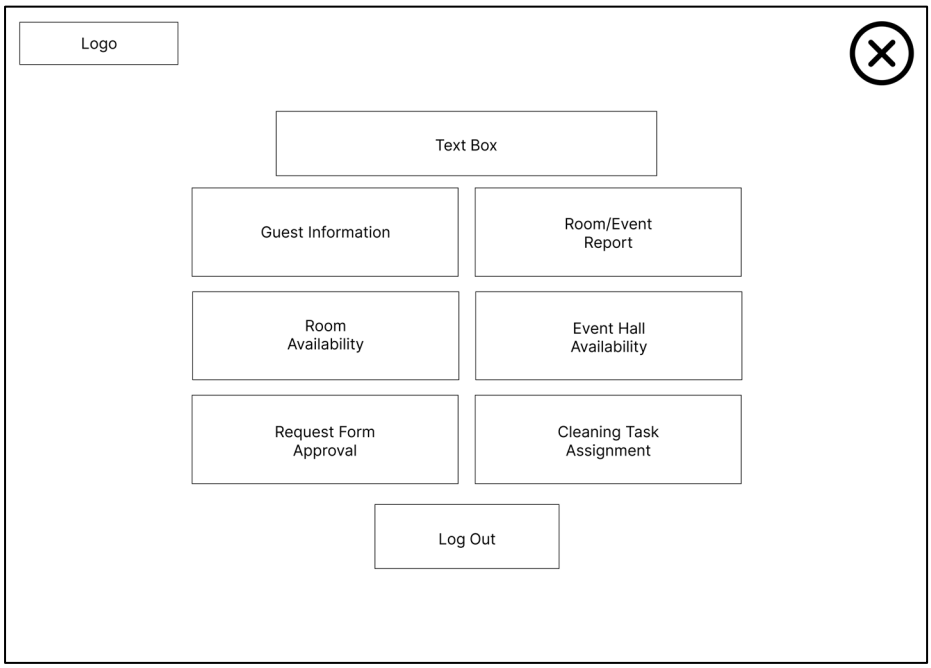


Figure 4.0_27 Admin Action Page

This page enables the admin to select which function is going to use.

Guest information page

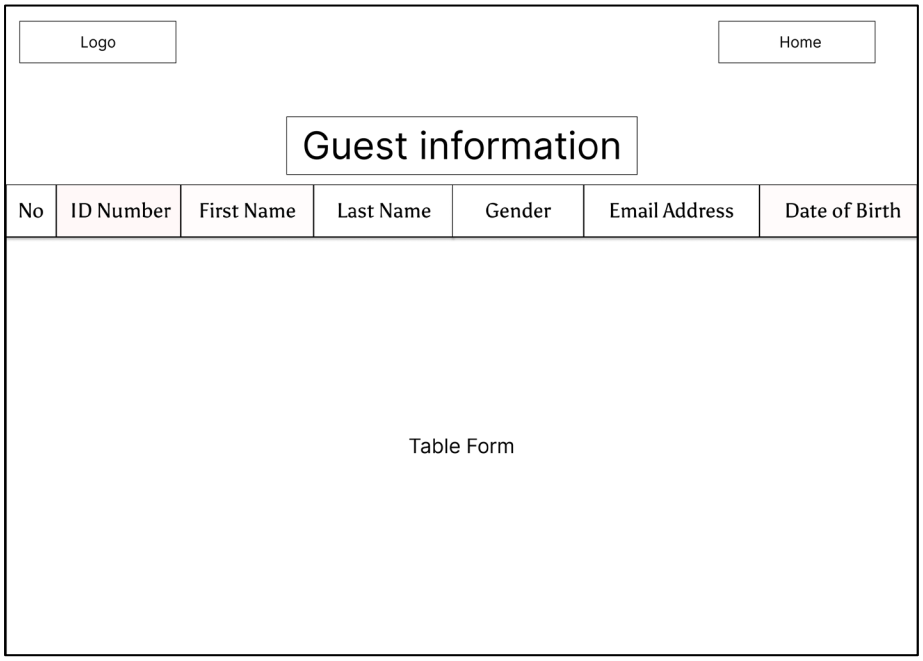


Figure 4.0_28 Guest Information Page

This page allows the admin to view the guest information.

Guest History page

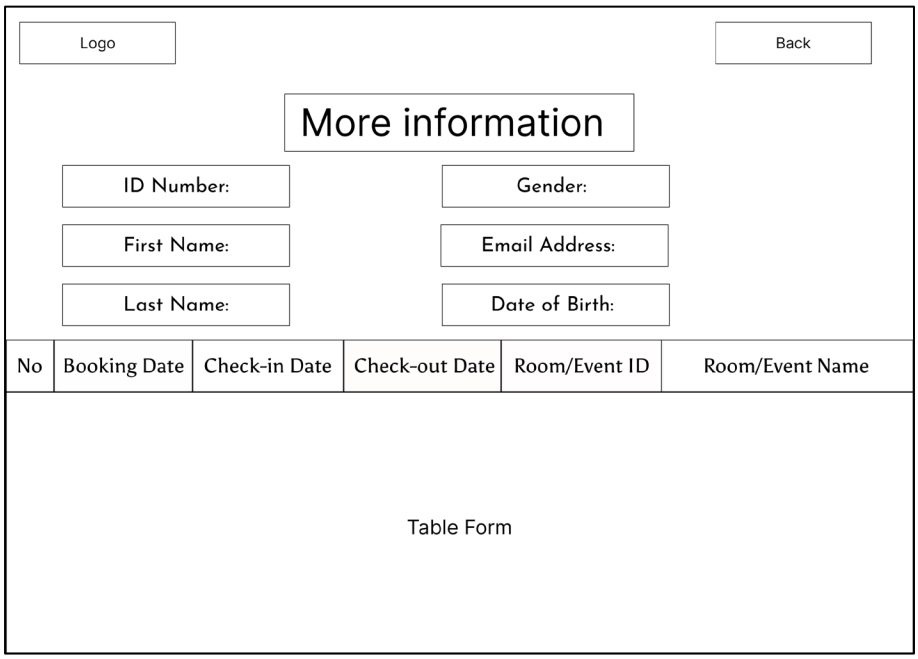


Figure 4.0_29 Guest History Page

This page allows the admin to view the guest's booking history.

Room/Event report generation page

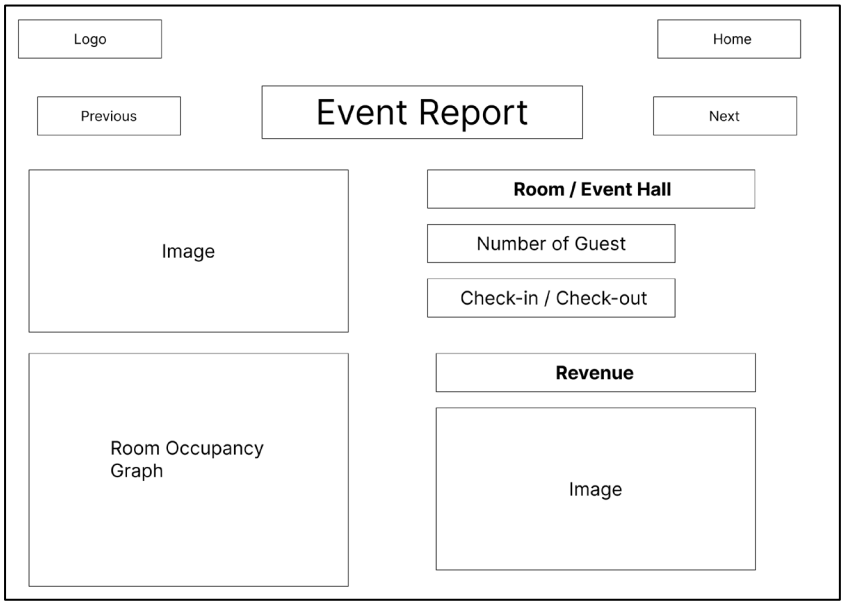


Figure 4.0_30 Room/Event Report Generation Page

This page allows the admin to view detailed information such as revenue, room occupancy, number of guests, and more.

Room availability

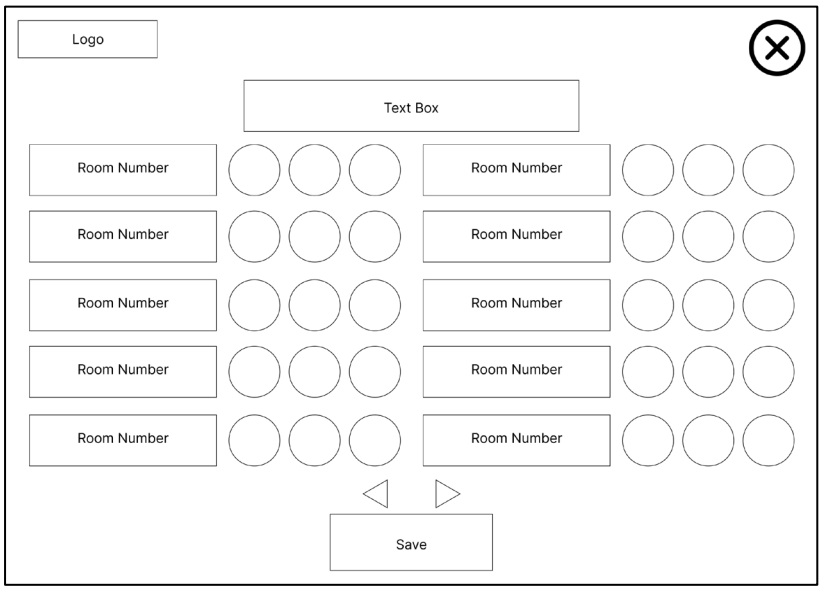


Figure 4.0_31 Room Availability

This page allows users to view room availability using color indicators: green for available, yellow for pending, and red for unavailable.

Event Hall availability

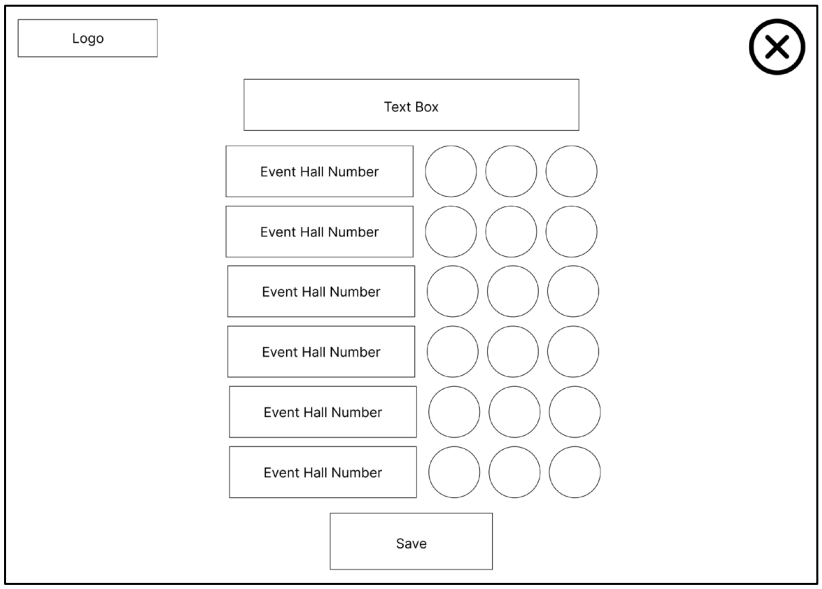


Figure 4.0_32 Event Hall Availability

This page allows users to view hall availability using color indicators: green for available, yellow for pending, and red for unavailable.

Request form approval

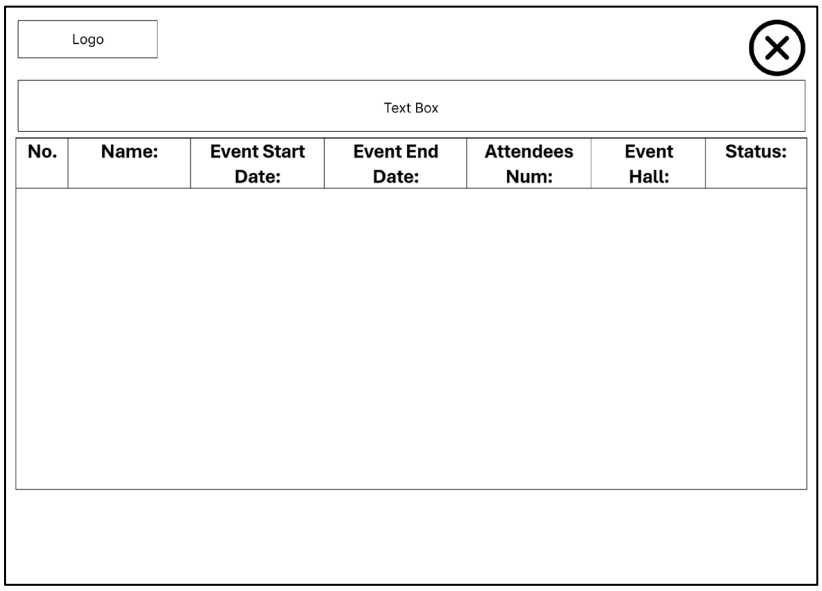


Figure 4.0_33 Request Form Approval

This page allows the admin to review event hall usage requests and related information.

Customer's request form

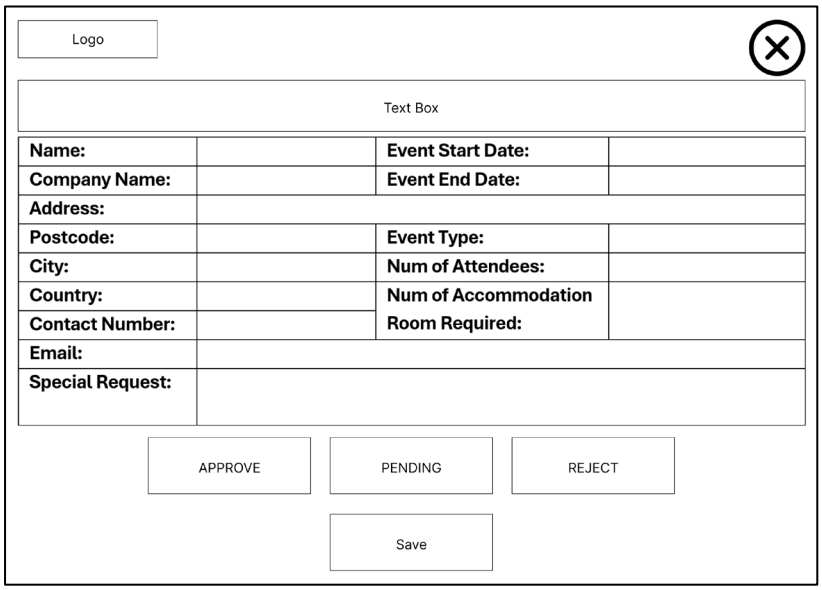


Figure 4.0_34 Customer's request form

Customers submit this form to apply for event hall usage. Admins can review the provided information and decide to approve, mark it as pending, or reject the application.

Cleaning task assignment page

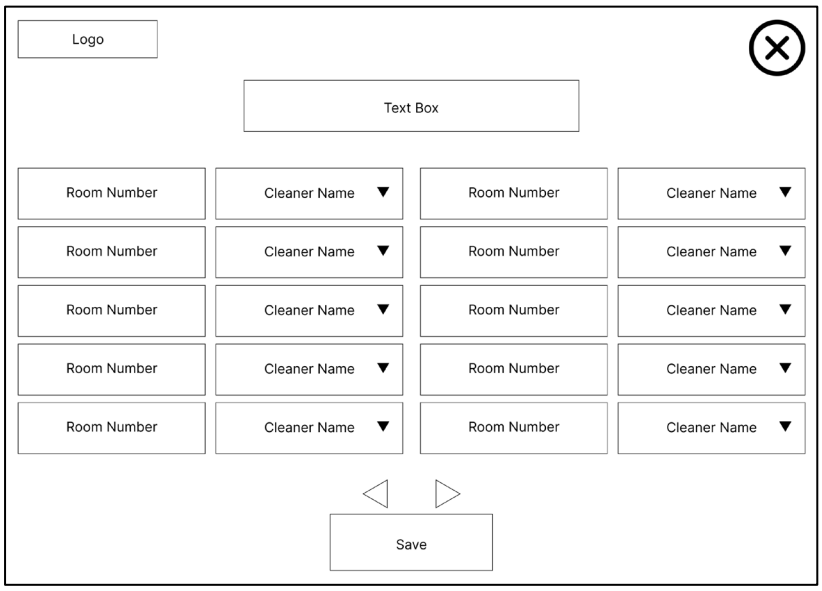


Figure 4.0_35 Cleaning Task Assignment Page

This page allows the admin to assign a cleaner to service the room.

Staff action page

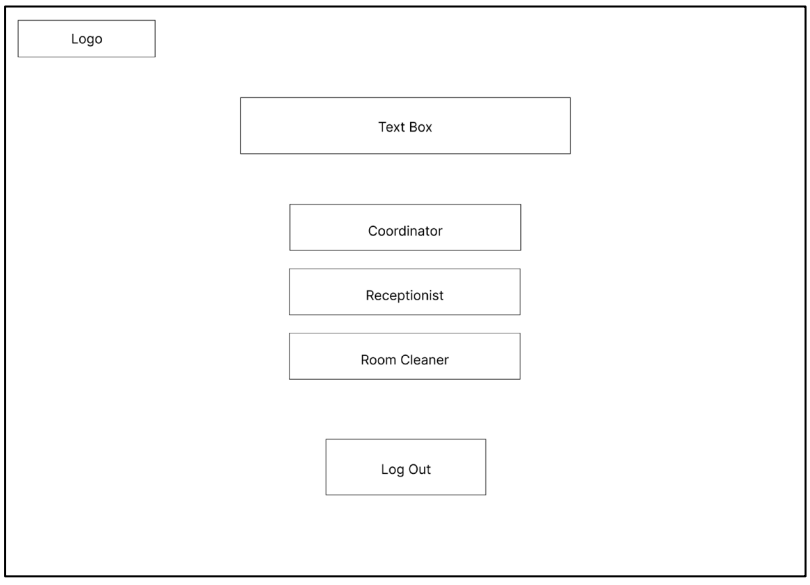


Figure 4.0_36 Staff Action Page

This page allows staff to choose their characters such as coordinator, receptionist, and room cleaner.

Staff booking management

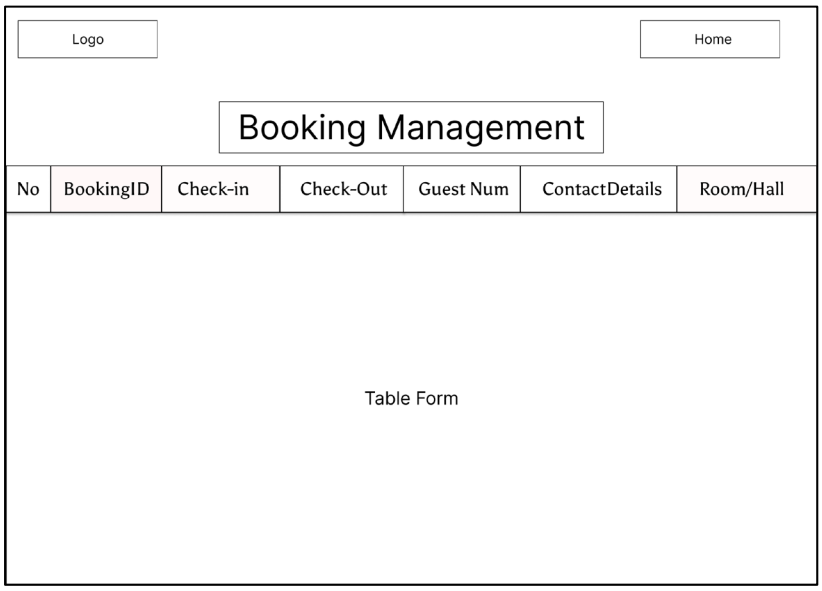


Figure 4.0_37 Staff Booking Management

This page allows the receptionist to view the booking information.

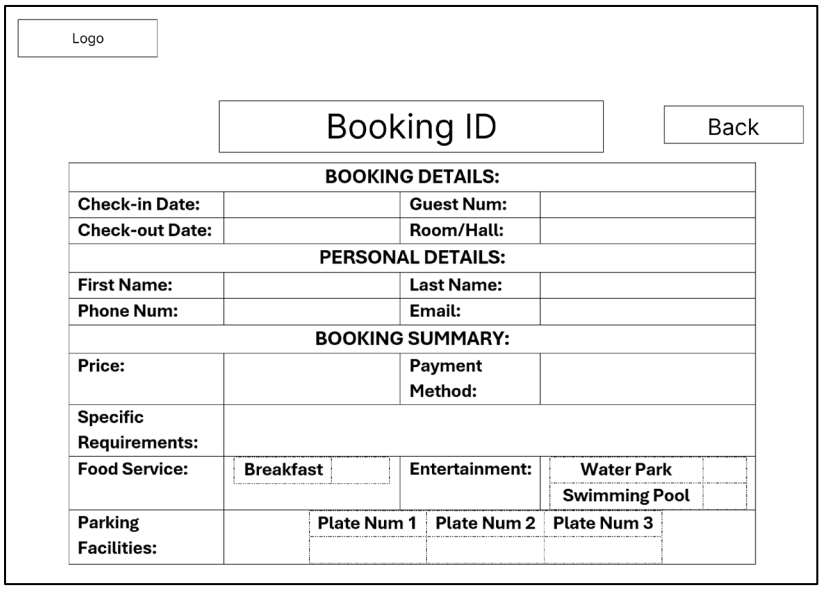


Figure 4.0_38 Booking Management Details

After selecting a specific booking ID on the staff booking management page, the receptionist can view detailed information about that booking.

Coordinator action page

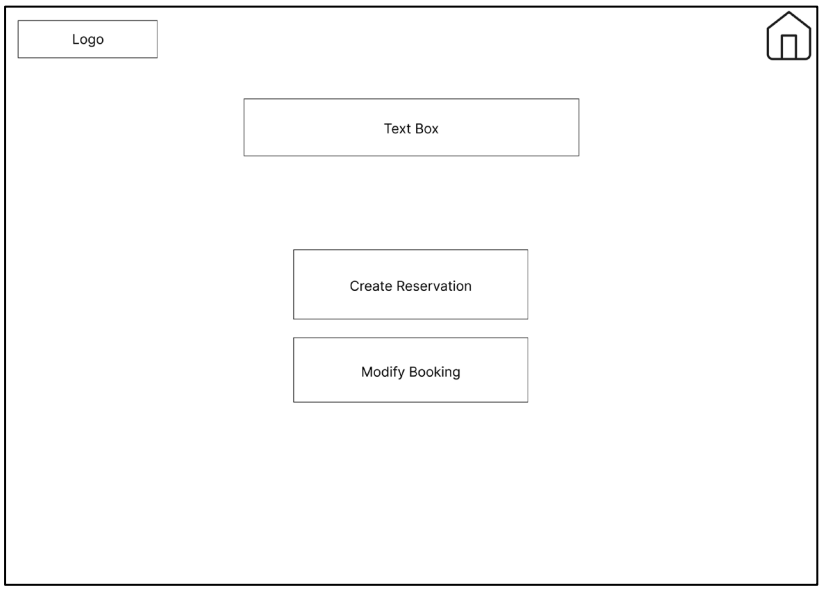


Figure 4.0_39 Coordinator Action Page

This page allows the receptionist to select specific functions to execute.

Modify booking page

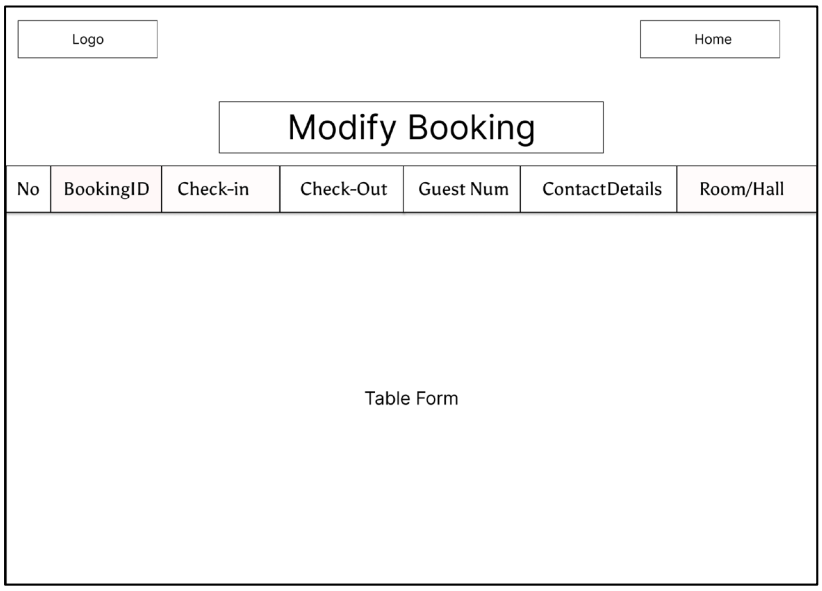


Figure 4.0_40 Modify Booking Page

This page allows the admin to modify booking information.

Figure 4.0_41 Receptionist Booking Page

This page allows staff to assist customers in booking a room.

Staff room cleaning page

Figure 4.0_42 Staff Room Cleaning Page

This page allows the room cleaner to view the rooms they need to clean and report any issues or damages.

5.0 Implementation

5.1 Hardware Requirements

Table 5.1 Hardware Requirements

Hardware	Purpose	Specification	Specific Model	Justification
Server	Database server to store/manage customer profiles, reservations, and non-database files	**Processor**: Intel Xeon Silver 4210 **RAM**: 32GB DDR4 **Storage**: 2TB SSD (Samsung 860 EVO) **OS**: Windows Server 2019 **Network**: Gigabit Ethernet	Dell PowerEdge T440	The Dell PowerEdge T440 with Intel Xeon Silver 4210 offers high reliability which is essential for handling large volumes of data transactions in hotels (Technology, 2020).
Backup Storage (NAS)	Regular backups to prevent data loss	**Storage**: 4TB with RAID support **Network**: Network-attached	Synology DiskStation DS920+	Synology DS920+ provides reliable data backup and supports RAID, ensuring data security. Its DSM software is more user-friendly compared to alternatives like QNAP, making it simpler for non-IT staff to configure scheduled backups (INC., 2020).
Point of Sale (POS) Terminals	Processing card payments at reception	**Display**: Touchscreen **Processor**: Intel Core i3 or higher	HP Engage One Prime	HP Engage One Prime provides an all-in-one POS solution. It is sleek and compact, which fits well in a hospitality environment and

		RAM: 4GB **Integrated Card Reader, Receipt Printer, Cash Drawer**		supports multiple payment methods, making it more versatile than other POS brands like Clover. (HP, 2024)
Barcode Scanner	Quickly scans customer IDs, room cards, and products	**Type**: Handheld **Technology**: Laser-based **Connectivity**: USB or wireless	Honeywell Voyager 1202g	The Honeywell Voyager 1202g is known for its high accuracy and speed in scanning barcodes, particularly in fast-paced settings like hotels. It outperforms other brands in range and battery life, making it ideal for busy front desks (1202g General Duty Scanner | Honeywell, 2021).
Card Payment Terminal	Process credit and debit card payments	**Features**: EMV Chip & PIN, Contactless payment (NFC), Integrated billing system	Ingenico Desk/5000	The Ingenico Desk/5000 is chosen for its high-level security compliance (PCI DSS), NFC support, and reliability. It supports multiple connectivity options and is known for compatibility, unlike Square terminals which are sometimes limited to specific integrations (Ingenico, 2022).
Printers	Calculating guest bills, processing payments, issuing receipts	**Type**: Multi-function laser **Features**: Network capability, Duplex printing	Brother MFC-L3770CDW	Brother MFC-L3770CDW offers network capability, efficient duplex printing, and reliability. Its multi-function capability

				(print/scan/copy/fax) outshines basic models, which lack versatility, making it a cost-effective choice for diverse printing needs in hotels (Harrel, 2018).
Workstation Computers	Managing room availability, booking reservations, updating systems	**Processor**: Intel Core i5 **RAM**: 8GB (16GB preferred) **Storage**: 256GB SSD **Display**: Full HD	Dell OptiPlex 7070	Dell OptiPlex 7070 combines performance, reliability, and security, which is essential for hotel front-office tasks. Unlike consumer PCs, it offers commercial-grade durability and features such as ProSupport, which ensures minimal downtime (Dell, 2024).
Router	Connecting devices to the network	**Type**: Dual-band **Speed**: 2.4 Gbps **Ethernet Ports**: Gigabit	Cisco RV340	The Cisco RV340 router supports robust, secure connectivity and excellent throughput, ideal for high-traffic environments like hotels. It provides better performance than consumer routers (e.g., Netgear Nighthawk) due to its commercial-grade security and VPN options (Cisco RV340, RV345, RV345P, and RV340W Dual WAN Security Router Data Sheet, 2020).
Automatic License	Integrated with License Plate Recognition	**Resolution**: 2MP to 9MP	Hikvision iDS-TCM403-A	This model is recognized for its superior accuracy in reading plates at various speeds,

Plate Recognition (ALPR)	cameras to automatically detect vehicle plate numbers at entry and exit points.	**Frame Rate**: Up to 60fps or 120fps **Form factors**: bullets, domes, PTZ, and multi-imager options		offering reliability that surpasses consumer-grade cameras (iDS-TCM403-A, 2023). This feature provides secure access and enables seamless check-in for guests with reserved parking, reducing wait times.

5.2 Software Requirements

Table 5.2 Software Requirements

Software	**Purpose**	**Specific Software**	**Justification**
Server OS	Running a hotel management system	Windows Server 2019	Windows Server 2019 is widely used for its stability, security, and ability to handle large databases efficiently, unlike Linux-based servers that may lack some enterprise-specific support in this context (Cascarano, 2023).
Client OS	Operating workstations	Windows 11 Pro	Windows 11 Pro offers enhanced security and user management, essential for front-office tasks. It is more user-friendly and secure than open-source options like Ubuntu in a non-IT environment (Muchmore, 2024).
Database Management	Storing data	MySQL Enterprise Edition	MySQL Enterprise Edition offers enhanced data security and performance monitoring, necessary for sensitive data in hospitality, surpassing the free edition in features (MySQL, 2019).
Hotel Management Software	Custom hotel operations	Oracle OPERA	Oracle OPERA is highly regarded in the hospitality industry for its comprehensive features, including reservation and billing integration, surpassing more basic options like Cloudbeds (OPERA Hotel Property Management Solutions (PMS), 2022) .
Billing Software	Generating bills/invoices	Square for Hospitality	Square for Hospitality integrates easily with POS and has strong reporting features, making it more efficient for billing needs than simpler billing tools like FreshBooks (Hospitality POS System | Hospitality Point of Sale Software | Square, 2024).

Antivirus & Security	Network protection	Kaspersky Endpoint Security	Kaspersky offers comprehensive endpoint protection with a firewall, encryption, and threat detection, preferred for its robust features compared to Norton or Avast (Kaspersky, n.d.).
Backup Software	Data backups for disaster recovery	Veeam Backup & Replication	Veeam is a reliable backup solution for servers with cloud compatibility, more versatile than Acronis for complex data environments like hotels (Inc, n.d.).
Event Management Software	Event booking and coordination	Oracle OPERA	OPERA's built-in event management module eliminates the need for third-party software, providing seamless integration with the hotel management system for event coordination.
Customer Feedback System	Collecting guest reviews	SurveyMonkey Integration	SurveyMonkey integrates with mobile apps for feedback collection, providing better data insights than simpler feedback mechanisms and integrating well with hotel CRMs (Team E. W., 2023).
Report Generator	Generating reports on hotel data	Microsoft Power BI	Microsoft Power BI's data visualization and report creation capabilities make it a top choice for hotel management analytics with more advanced features than simpler tools like Google Data Studio. (KnowledgeHut, 2019)
Multilingual support	Manage content in multiple languages.	WordPress Multilingual Plugin (WPML)	These plugins allow users to manage content in multiple languages. WPML also offers automatic translation services. (Jackie, 2016)

Multi-currency support	Provide real-time currency conversion rates	Fixer.io	This API provides real-time exchange rates from reputable financial sources that ensure users always have the most accurate information. (Martinbaldwin, 2024)
Card Payment Support	Secure card payments	iPay88	ipay88 supports credit/debit cards (Visa, MasterCard, AMEX) and integrates popular Malaysian e-wallets like Boost, GrabPay, Touch 'n Go eWallet, and ShopeePay (Malaysia Online Payment Gateway, 2024). It offers API integration and a ready-to-use payment page.
Chatbot support	Help users to solve their issues.	Zapier	Zapier are its powerful automation, easy-to-navigate dashboard, extensive task history, and flexible pricing (Zuckerman, 2019)
Data protection software	To ensure the security and confidentiality of customer and business data	Kaspersky Endpoint Security	It provides a multi-layered data protection approach, combining advanced threat detection, firewall features, and data encryption. It provides real-time monitoring and centralized management, making it very suitable for protecting sensitive customer data in the hotel industry. (Kaspersky Endpoint Security for Business (Enterprise Solutions) | Kaspersky, n.d.) .

6.0 Prototype

Refer to G10_VideoPrototypeModel.mp4

7.0 Conclusion

To conclude, the implementation of the Swiss Garden Hotel Management System is aimed at developing a system to streamline the operation of the hotel and change the process from manual to automatic. This system is also designed to improve the overall efficiency and productivity of hotel operations and improve customer satisfaction.

However, the Swiss Garden Hotel Management System has some limitations. Firstly, the system lacks advanced UI/UX elements, which makes it less visually appealing and less user-friendly to users. Besides that, the system only supports a few languages, which may not cater to international guests while interacting with the system, which leads to a loss of potential customers. Additionally, the chatbot's knowledge base may not be fully completed, potentially causing some bugs and errors while assisting the customer. (Chatsimple.ai, 2024). For example, failing to understand user requirements, providing unnecessary information, or failing to catch the keywords in customer questions (Khalid, 2024). Furthermore, despite the system including the automated booking process, it still requires checking in manually at the counter, which reduces convenience and lowers the efficiency for customers. Lastly, managing high volumes of data and customer transactions could lead to system lags or crashes, especially during peak periods.

Moreover, developing the system also came with challenges, including the high maintenance costs and time for the system to implement. Besides that, training staff is also one of the challenges faced, including the staff who might feel difficulty while using the newly developed system, especially those who are not tech-enabled (Khalid, 2024). The initial deployment might require additional staff training, which aims for a smooth acceptance of the new system. Furthermore, handling a growing number of customers' data might potentially risk system failure or data breaches.

For the future to continue to meet user needs, the development team had to address the limitations and challenges. For instance, they must fulfill the UI/UX element by implementing the design principles. To complete this, it requires gathering user requirements by using methods such as surveys, interviews, and observations. Besides that, to improve the user experience, the development team should introduce a system that can rearrange customers' bookings automatically if the pre-arranged rooms or event halls are under maintenance instead of manually checking and rearranging by the coordinator. In addition, an automated kiosk system is required to help customers check in automatically by entering their booking reference

number instead of going to the counter and checking in manually. Furthermore, improving the chatbot's knowledge base by implementing machine learning capabilities, making it able to identify keywords from user queries and provide an accurate response. Finally, adding personalized features, such as loyalty programs and offers, to grab customers' attention.

In conclusion, the Swiss Garden Hotel Management System provides a significant improvement in efficiency, productivity, and user satisfaction. While facing some limitations and challenges, addressing them with targeted enhancements to improve the system and meet the current needs.

8.0 Methodology (Individual Part 1):

Methodology is a set of processes or procedures followed in an organization to conduct all the steps necessary to analyze, design, implement, and maintain a system for day-to-day operation. It aims to provide a systematic approach to software development.

8.1 Spiral Model (VINCENT TEO KAI QI – TP075160)

Agile development is designed to reduce risk by developing software in shorter time frames known as iterations, which typically run one week to one month. (Sharma, 2020). The Spiral Model, one example of this methodology is a risk management technique that combines the iterative development process model with the Waterfall model parts. (Contributor, 2019). It was first proposed by Barry Boehm which provides support for risk handling and is especially suitable for large and complex development projects with high levels of uncertainty or risk. It emphasizes managing risks through multiple iterations.

According to information from GeeksforGeeks, this model provides a systematic and iterative approach to software development. It looks like a spiral with loops, where each loop represents a phase of development. The number of loops, which are called phases is determined by the project manager, depending on the project development risks. There are a total of five key phases of the spiral model which include objectives defined, analysis, engineering, evaluation, and planning. (GeeksforGeeks, What is Spiral Model in Software Engineering?, 2024).

8.1.1 Implementation of Spiral Model

First and foremost, the development team will **identify the primary objectives** for implementing this project. In this first phase of the spiral model, they need to clarify what the project aims to achieve, including functional and non-functional requirements. (GeeksforGeeks, What is Spiral Model in Software Engineering?, 2024). For instance, Dr. David, the owner of the Swiss Garden Hotel aims to develop a system to automate various manual processes of the hotel in order to streamline operations and enhance customer service. For functional requirements, the developed system must allow guests to book rooms and event halls online, including check-in and check-out dates, automatically check the availability and update the status in real-time, online payment handling via cards or e-wallet, and generate reports for admin. For non-functional requirements, they need to define the system's operational attributes, such as performance, security usability, and reliability which might enhance the system's quality and user satisfaction.

Secondly, during the **risk analysis** phase, the development team is required to identify potential risks related to implementing the new hotel management system. This phase is crucial for identifying uncertainties that may cause the system to fail or delay, especially in complex projects like the Swiss Garden Hotel Management System. (Singh, 2024). For example, there is a major risk that the staff who rely on the manual method may struggle to shift to the new digital information system. There could also be some other risks such as security, technical, and budget risks/ Some users might be concerned about protecting sensitive guest information such as address, contact number, and credit and debit card details. The risks associated with the solution are identified and resolved using the best feasible strategy. (GeeksforGeeks, What is Spiral Model in Software Engineering?, 2024).

Thirdly, in the **engineering** phase, the actual software development will begin based on the requirements gathered requirements and analysis. This step includes architectural design, module design, physical product design, and final design. (Singh, 2024). This phase involves creating modules for room and event hall reservations, event management, guest records, billing, and report generation. For example, a booking reservation system is implemented where guests can book rooms online, view real-time availability, and select check-in and check-out dates. The system has a real-time system to check availability. Thus, the development team needs to code, test, and deploy the system that fulfills all these key requirements to satisfy all the potential users including the manager, staff, and guests of the Swiss Garden Hotel.

Besides, **evaluation**, the next phase of the Spiral Model focuses on assessing whether the system genuinely improves efficiency compared to the previous manual processes. The software will be evaluated to determine if it meets the users; requirements gathered in the previous iteration. (GeeksforGeeks, What is Spiral Model in Software Engineering?, 2024). The team needs to check if the newly developed Swiss Garden Hotel Management System can improve accuracy, efficiency, and productivity and simultaneously reduce errors and bugs which might increase guests' experience and satisfaction. Not only that, but the usability of the system is also another critical factor. For instance, Mr. David and the hotel staff test the system by simulating real-world scenarios, such as booking rooms, managing events, and generating guest invoices. If any identified usability issues are discovered, the adjustments and changes must be addressed to ensure that it is intuitive and easy to use.

Last but not least, based on the evaluation results and feedback, the project enters the **planning phase for the next iteration**. Continuous enhancements might be made in the future to make sure the developed software system can always satisfy the users' needs. During the "spiral" phase, the team determines the order of events in the product's current state, followed by a "revision" or "reproduction" phase where the product is refined and prepared for production, making this a key part of the planning process. (Singh, 2024). For example, some additional features such as multi-language and currency support for international guests, chatbots, and parking management can be added to the system.

8.1.2 Advantages and Disadvantages of Spiral Model

Advantages

One of the advantages of using the Spiral Model for implementing the Swiss Garden Hotel Management System is its extensive focus on risk analysis, which enables early identification of potential risks and meanwhile can enhance the avoidance of risk. By analyzing and addressing the risks early, they can ensure that the system is reliable and secure. In addition, the system can run smoother and long-term success of the system.

Moreover, due to its flexibility, the Spiral Model is encouraged for development. Continuous changes can be made to the requirements after the development has started or already evaluated. (Contributor, 2019). For example, after evaluating the Swiss Garden Hotel Management System and receiving particular useful feedback from users, the development team might always improve the capabilities and functionalities based on the feedback gathered.

Not only that, but the Spiral Model is also suitable for large-scale projects, especially the implementation of the Swiss Garden Hotel Management System. (Joseph, 2021). This model allows for multiple iterations, which is crucial for a complex system like the hotel's, which needs a lot of key features like room and event hall booking, billing management, and report generation. The iterative approach ensures that the system evolves with continuous improvements based on feedback and testing.

Disadvantages

However, there are also some limitations while using the Spiral Model for this system development such as high cost and time consumption. The Spiral Model requires extensive risk analysis, planning, and multiple iterations, which can make the development process time-consuming and expensive. For a project like the hotel management system, the constant evaluations and updates may lead to higher costs and longer timelines compared to other models.

Furthermore, complexity in management for the implementation of the hotel management system might be another limitation. Managing the Spiral Model can be difficult due to its iterative approach and the need for frequent evaluations of risk and client feedback. To keep the process organized and guarantee that each phase is completed efficiently, this complexity may require highly qualified project managers and developers.

Lastly, the flexibility of the Spiral Model allows for continuous changes and refinements throughout the development process. However, this flexibility can lead to scope creep, where the project grows beyond its original scope due to new features or requirements being added over time. In the case of the Swiss Garden Hotel Management System, this could result in project delays, increased costs, and difficulties in maintaining focus on the core objectives if the scope is not carefully controlled.

8.1.3 Justification of Spiral Model

The Spiral Model is an ideal choice for implementing the Swiss Garden Hotel Management System due to its strong focus on risk management and iterative development, both of which are crucial for a large-scale and complex project. One key advantage of this methodology is its ability to perform a continuous risk analysis, allowing the development team to identify and mitigate potential risks early, ensuring that critical functions like room reservations, event management, and billing are secure and reliable. Additionally, the model's iterative approach allows for flexibility, enabling the system to evolve with ongoing feedback from hotel management and staff. This is particularly useful for accommodating future enhancements such as multi-language support or integration with third-party platforms, which may emerge during or after the development process. Furthermore, by developing and refining the system in stages, the Spiral Model enables the hotel to ensure that each feature, from guest bookings to real-time availability checks, is thoroughly tested and improved, making the system highly adaptable and suited to Swiss Garden Hotel's operational requirements.

8.2 Scrum Model (WANG JIM LOONG – TP075739)

Scrum is an agile framework that uses incremental and iterative software development approaches. (Sachdeva, 2016). Its incremental and iterative methods make it popular, and it is based on the Agile philosophy (Nikolaieva A. , 2024). Ken Schwaber and Jeff Sutherland developed this methodology in the early 1990s to help companies with sophisticated development projects (Scrum.org, n.d.). Scrum methodology is executed in sprints. Each sprint might last up to a month (Nikolaieva A. , 2024). This method helps the team quickly identify problems, output solutions, and gather feedback quickly (Nikolaieva A. , 2024).

Scrum methodology has 3 significant roles which are the product owner, scrum master, and the development team. The product owner acts as a mediator between the development team and stakeholders in the development process. (Sachdeva, 2016). A product owner is responsible for defining the features that are needed in the software. They gather feedback from the customer and ensure the development team is on track to meet customer satisfaction. (Nikolaieva A. , 2024). Then, a scrum master is responsible for ensuring the whole development team is following the scrum process and facilitating collaboration within the team. They need to solve obstacles/problems that might affect the team's work progress or slow the sprint's pace. (Sachdeva, 2016). A development team will be responsible for executing the development. (Nikolaieva A. , 2024). A team can be made up of developers, testers, UX designers, and so on…. They are the ones who deliver the product increment during each sprint. They turn the requirements in the product backlog into a shippable product increment.

3 artifacts are used in Scrum methodology. First, product backlog which is created by the product owner with a prioritized list of features and requirements for the product. The requirements for the product will be listed in a product backlog by using the user story format. (Sachdeva, 2016). Then, a sprint backlog is a list of selected tasks/requirements from the product backlog that the development team commits to complete in a specific sprint after a sprint planning meeting is conducted. Besides, burndown charts show the progress during a sprint on the completion of tasks in the sprint backlog. It is a graph that shows the amount of sprint backlog work remaining throughout the sprint. (Sachdeva, 2016).

After that, the Scrum methodology introduces 3 ceremonies which are sprint planning, daily scrum, and sprint review. Sprint planning is a meeting/discussion for determining what will be done during the upcoming sprint. It involves the scrum master, the development team, and the product owner. Daily scrum meetings discuss what the team has done since the last

meeting, what is going to be done before the next meeting, and what problems/obstacles met them. Lastly, the sprint review will be held at the end of the sprint. The team will demonstrate the functionality to the product owner and stakeholders, and then gather feedback from them to improve the product (Sachdeva, 2016).

8.2.1 Implementation of Scrum Model

Firstly, the product owner of Swiss Garden Hotel will represent the customer and list out features that can be built into the product such as room reservations, event booking, breakfast service, filter, payment methods, and so on …, into the product backlog based on descending priority. The user story format normally starts with "As a…", continues with "I need…", and ends with "so that…".

Number	User story	Priority
1	**As a** user, **I need** a filter feature **so that** I can eliminate unnecessary options.	High
2	**As a** user, **I want** a search bar **so that** I can look for the desired information more quickly.	High
3	**As a** manager, **I want** a report-generation feature **so that** I can easily make financial analyses.	Medium

Table 8.2.1 Example of product backlog (Wang Jim Loong, 2024)

Then, the development team, scrum master, and product owner will engage in the sprint planning meeting. They will select the requirements based on the priority list from the product backlog previously to create the sprint backlog. The items listed in the sprint backlog will need to be completed during the sprint. For example, the filter feature and search bar function are considered a high priority in the product backlog by the product owner of The Swiss Garden Hotel. Therefore, the team will select these 2 items and listed into the sprint backlog.

In addition, the team will then enroll in a 1- 3 week time box called sprint. The development team needs to implement the work that is listed on the sprint backlog during the period. During the sprint, the daily scrum occurs. The development team needs to hold a daily scrum every day during the sprint. It normally took 15 minutes for team members to discuss what they worked on the previous day, what they will work on today, and are there any challenges and problems they faced when developing the system. For example, a software

developer in the team- Ali, was implementing the filter feature yesterday. He will work on the search bar function today but the search bar's performance may slow down when querying a large database.

After the sprint, the team delivers a potentially shippable product. This is a product that can be ready to deploy if desired. However, the potentially shippable product needs to be reviewed by Mr. David and the stakeholders during the sprint review meeting. The development team needs to showcase their work to them. If they are satisfied with the product, it is considered complete. Conversely, if the product does not meet the satisfaction, the whole team needs to repeat the workflow for each sprint.

Once Mr. David and the stakeholders are satisfied with the product, they can decide to deploy the Swiss Garden Hotel's website and release it to the public.

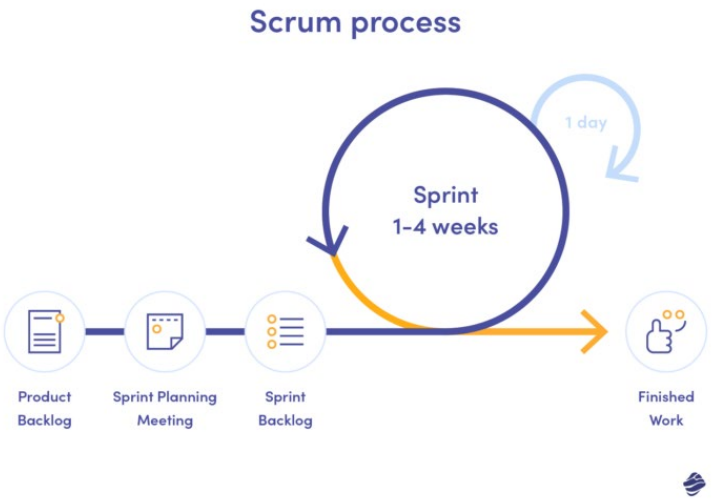


Figure 8.2.1 Scrum process (Dąbrowska, 2017)

8.2.2 Advantages and disadvantages of the methodology

Advantages

- Teams can complete the project quickly and efficiently (Chandana, 2024)
- Can easily make changes because it is running in short sprints and continuous feedback (Sharma K. , 2020)
- Short iterations make it solve new challenges quickly (Nikolaieva A. , 2024)
- Effective use of time and resources (Chandana, 2024)
- The contribution/ productivity of team members can be easily noticed (Nikolaieva A. , 2024)

Disadvantages

- The release of the product may be delayed due to no fixed end date (Nikolaieva A. , 2024)
- Team members will be frustrated because of daily meeting (Chandana, 2024)
- Difficult to implement with large teams (Chandana, 2024)
- Experienced team members are needed only (Sharma K. , 2020)
- Not suitable for huge projects (Nikolaieva A. , 2024)

8.2.3 Justification of Scrum Model:

The Scrum methodology is suitable for the Swiss Garden Hotel Management System as it enables the team to complete the project quickly and efficiently. By using Scrum, the team can develop a product that satisfies Mr. David and the stakeholders because Scrum emphasizes client feedback and allows for adjustments during the development process. Throughout each sprint, the team may encounter new challenges like real-time room availability and billing calculation errors. Therefore, a short iteration inherent in Scrum enables the team to solve these challenges quickly, ensuring effective integration of features like room reservation management while optimizing time and resource usage. Last, Scrum enhances individual contributions by allowing members to be easily monitored through daily sprint and sprint reviews which collectively improve the overall project progress and accountability.

8.3 Rational Unified Process (RUP) (WANG ZI JIE – TP076251)

Rational unified process (RUP) is a framework for software engineering and development processes cantered on using the unified modeling language (UML) to build and design the software (Team I. E., 2024). Besides that, it is developed by Rational Corporation and International Business Machine Corporation (IBM) to enable the user to personalize, design, and alter the unified process. (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). Furthermore, RUP has a total of five phases, including inception, elaboration, construction, transition, and production. (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). RUP is able to manage the risk and is flexible to change due to the use-case-driven and architecture-centric strategic (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). It is also using multiple iterations to gradually improve an efficient solution. (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). Furthermore, RUP is also good at decreasing development costs and deterring the waste of resources, which ensures the system architecture and design are validated early and guarantees the reliability and scalability of the system. (Team I. E., 2024).

8.3.1 Implementation of the Rational Unified Process (RUP):

The first phase of RUP is **inception**, which focuses on identifying the scope of the project and customer requirements (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). After identifying the project scope, the project team has to estimate the cost and time to build by using case modeling. (Team I. E., 2024). In the case study, the project team has to communicate and cooperate with the Swiss Garden Hotel Management to gather and identify the core functionality of the system, such as room reservation management, event booking, guest information management, and billing management.

The second is **elaboration**. The main goal of these phases is to develop a plan and model the project to reduce the risk. (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). Besides that, the use case model developed during this stage is used as a blueprint for the subsequent stages of the project (Team I. E., 2024). In the case study, the project team has to design the system architecture to analyze the expenses and resources required for the project's whole development, producing a software baseline architecture that is executable and actionable. (Team I. E., 2024).

The third phase is **construction**, which means the project is completed, the system code is finished, and the testing is completed (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). In the case study, the project team has to focus on developing the actual system based on the architecture designed in the previous phase, especially focusing on important features such as room reservations, event booking, and billing. After that, each feature must be tested to ensure that it meets the functional requirements.

The fourth phase is **transition**, which means the final project is deployed to the public and the beta test is carried out (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). In the case study, the Swiss Garden Hotel Management system is released and ready to use. All the staff are required to be trained to use the system. Besides that, the system often carried bugs and errors, which need to be fixed on daily usage (Team I. E., 2024).

The last phase is **production**, which is the last phase of the model, and the project requires updating and maintaining up-to-date (GeeksforGeeks, What is RUP(Rational Unified Process) and its Phases?, 2024). In the case study, the system has moved into long-term maintenance. The system is fully operational and planning to add new features such as customer loyalty, chatbots, and more.

8.3.2 Advantages and Disadvantages of the Rational Unified Process (RUP)

Advantages

The Rational Unified Process (RUP) offers several advantages that make it suitable for complex projects like the Swiss Garden Hotel Management System. Firstly. It offers detailed documentation. The RUP process strictly records each step while creating the system, which significantly helps in collaborative projects (Team I. E., 2024). Besides that, RUP is also able to improve procedures for risk management by breaking the development process into clearly defined phases. RUP makes project management more structured and manageable, allows the adaptive capability to deal with changing requirements, and improves risk management and troubleshooting efforts (Team I. E., 2024).

Disadvantages

However, RUP also has some disadvantages. The methodology requires significant project management efforts, which are highly consuming time, and manpower. Furthermore, the complexity of the process is also a fatal flaw in this methodology (Team I. E., 2024). Due to the complexity of the process, it heavily relies on expert team members to successfully perform this methodology (Team I. E., 2024). Lastly, RUP required sufficient time and cost to complete the system (Team I. E., 2024). RUP can be expensive and time-consuming because of the detailed steps and requirements (Team I. E., 2024).

8.3.3 Justification of the Rational Unified Process (RUP)

The RUP methodology is highly suitable for the Swiss Garden Hotel Management System because it is able to address complex requirements such as room booking, event management, and billing. Strong risk management helps to handle critical issues such as bugs and data security. The iterative model also allows for continuous user feedback at various stages, ensuring the system can maintain and fulfill the business needs.

8.4 Waterfall Methodology (WONG JUN MING – TP076078)

The waterfall model is a linear project management technique that gathers the requirements of stakeholders and clients at the initiative of the project and then establishes a subsequent project plan to meet these requirements (ProjectManager, 2022). In the waterfall model, all phases of the workflow should be done before leading to the next phase. Although there are several kinds of project management methods, the waterfall model is very suitable for projects with clearly defined goals from the outset (Hoory & Bottorff, 2022).

8.4.1 Implementation of the waterfall model

The requirements **analysis** phase involves collecting needs from stakeholders and analyzing them to identify the project's scope and objectives (GeeksforGeeks, Waterfall Model - Software Engineering, 2018). In this phase, collect system requirements by discussing the current challenges faced by the Swiss Garden Hotel such as manual booking, event management, billing, and others. The minimum requirements include specified functions like room booking, event booking, customer records, and others, and three additional functions such as a chatbot, parking management system, and multi-language and currency.

When the requirements are identified, the **design** phase will start. This involves creating a described design document that sketches out the software architecture, user interface (UI), and system components (GeeksforGeeks, Waterfall Model - Software Engineering, 2018). For example, the architecture of the hotel management system will be created. This includes creating a user interface for room and event bookings, a database for guest records and billing, and real-time integration for report generation. For Mr. David, a dashboard will be designed to provide instantaneous data and analytics for appropriate decision-making.

The **implementation** and **unit testing** phase involves writing software code according to design conditions. This stage also includes unit testing to make sure that each part of the software works as anticipated (GeeksforGeeks, Waterfall Model - Software Engineering, 2018). For instance, this phase involves coding the system components using appropriate technologies such as room booking, event booking, and payment processing. Additional features like a chatbot could be implemented with platforms like Zapier or Botpress.

The **integration** and system **test** phase focuses on verifying the interactions between different system modules. This involves combining individual units and testing them to identify interface defects between integrated components, ensuring they work together seamlessly (Nair, 2024). Each system component will be tested individually by using unit testing and then

integration testing to ensure proper functionality which includes payment processing and report generation. User Acceptance Testing (UAT) will be conducted with hotel staff to validate the system.

After successful testing, the system will be deployed. The last phase of the waterfall model is operation and maintenance, which involves resolving any problems that occur after software deployment and making sure it continues to meet needs over time (GeeksforGeeks, Waterfall Model - Software Engineering, 2018). The system is fully operational, and hotel staff have begun using it to handle room bookings, event bookings, billing, payment processing, and generating reports. If there are problems, the problems will be solved and fixed in this phase. For example, fixes bugs and issues like billing errors or room availability problems that occur during daily use.

8.4.2 Advantages and Disadvantages of the waterfall model

Advantages

The waterfall model applies a transparent and identified set of phases to develop when contrasted with other methodologies (Gaille, 2020). This is because each single phase is well-defined which makes it easy to know what is expected at each phase. That includes collecting needs and documentation, system design, implementation, testing, delivery, and maintenance. For example, after completing each phase, progress is checked and approved by management before moving forward. This helps the hotel's owner to see how the project is progressing at each phase.

The waterfall model is easy to understand (GeeksforGeeks, Waterfall Model - Software Engineering, 2018). The linear and sequential structure makes it simple for teams to follow. Each phase is well-
documented which is crucial for understanding the project's evolution and maintaining it long -term. It helps onboard new team members easily. The development team just needs to follow a simple process such as gathering requirements, designing the system, developing modules, testing, and deploying. This structure allows everyone to easily understand what to do next.

The waterfall model reinforces good testing habits (Gaille, 2020). Due to the waterfall model's use of a clear planning approach at each stage, testing steps are pre-planned before implementation. This advantage lets the team structure their best methods into the solution before every work begins. For instance, during the system design, the team anticipates needing to test various functions such as room availability, booking confirmation, and cancellation processes. Test cases are planned and documented in advance. They might be planned to test if the guest books a room for available dates.

Disadvantages

However, the waterfall model does not establish adjustments (Gaille, 2020). The waterfall model follows a set of steps, always making the team forward continuously. When developers use the old method of this approach, there is nearly no way for changes if unexpected situations occur during the project. A team can faithfully follow every step to the near end of a project, but then they might meet an unexpected barrier. If changes to the goals of work are required, making the required adjustments to move forward is almost impossible. For example, during the design phase, a test case will be prepared to test that room rates, taxes, and additional services are calculated correctly.

The waterfall model can invalidate the work previously accomplished (Gaille, 2020). When the developer needs to adjust after finishing part of the project, the waterfall model will not allow it to continue. The only way to solve this disadvantage is to find a method to finish the work in the current step. If that outcome is impossible, then the developer has to return to the starting point of the process and start processing the latest information again. For instance, the team must invest additional time and resources to repeat the design and development work already done. This could lead to delays in the project timeline and increased costs.

The waterfall model postponed testing until after the end of the project (Gaille, 2020). The waterfall model postpones the testing phase of every project until the last of the work. The old method creates the quarter step in the six-step process. Since the project may take quite a long time to accomplish at this stage, several critical outcomes may require significant revisions. Ignoring the empirical data in the anticipated value suggestion until developers start considering competing in the market could lead to significant user issues that are necessary to be operated.

8.4.3 Justification of Waterfall Model

The waterfall model is suitable for the Swiss Garden Hotel Management System because the project's requirements are clearly defined and are unlikely to change during the development process. For example, the hotel’s requirements like room reservations, event booking, billing, parking management, and payment management. The linear approach ensures that each stage is thoroughly completed before moving on to the next, thereby minimizing the risk of overlooking important details. Furthermore, Mr. David focuses on integrating real-time data and analytics, which means that a well-planned system from the outset will provide him with the reliable performance metrics needed to make business decisions.

8.5 Final Methodology Chosen

After discussion, the most suitable methodology chosen by us for the Swiss Garden Hotel Management System is the Spiral Model. The spiral model has a strong emphasis on risk management which is crucial for complex projects like the Swiss Garden Hotel Management System since the system requires handling large amounts of sensitive data such as guest information, and payment methods details making risk analysis a key consideration.

Moreover, due to the Spiral Model supports the iterative approach, it allows for incremental enhancements and improvements in the future which is beneficial for fulfilling and satisfying the system's for including a variety of features. For instance, more advanced room reservations, event booking, billing systems, and automated room cleaning assignments. This model supports continuous feedback and refinement, ensuring that the new features can be implemented based on the feedback given.

Furthermore, flexibility and scalability are essential as the hotel's requirements may evolve, especially when new features are developed such as multilingual and multi-currency support, chatbot, and so on. Unlike the waterfall model, its linear approach lacks the flexibility needed for handling potential changes and iterative feedback in the system development. However, the selected model, the spiral model offers the flexibility to revisit and refine earlier development stages based on the results.

Last but not least, the Swiss Garden Hotel Management System is a comprehensive, large-scale project that requires a systematic approach to manage its multiple components. The repeated phases (iterative process) of the Spiral Model are well-suited to handle such complexity. The scrum model may not be ideal for the huge system development because it requires a large team and frequent meetings for continuous feedback given by the users. Not only that, the RUP model is also unsuitable for this project since it requires more expert team members to develop it and causes cost and time consumption.

9.0 Activity Diagram (Individual Part 2)

9.1 Vincent Teo Kai Qi (TP075160)

9.1.1 Introduction

Use case: Book event hall

Actor: Customer

Description: Customer can book the event hall

Use case: Make approval

Actor: Admin, Customer

Description: Admin approves requests such as event hall.

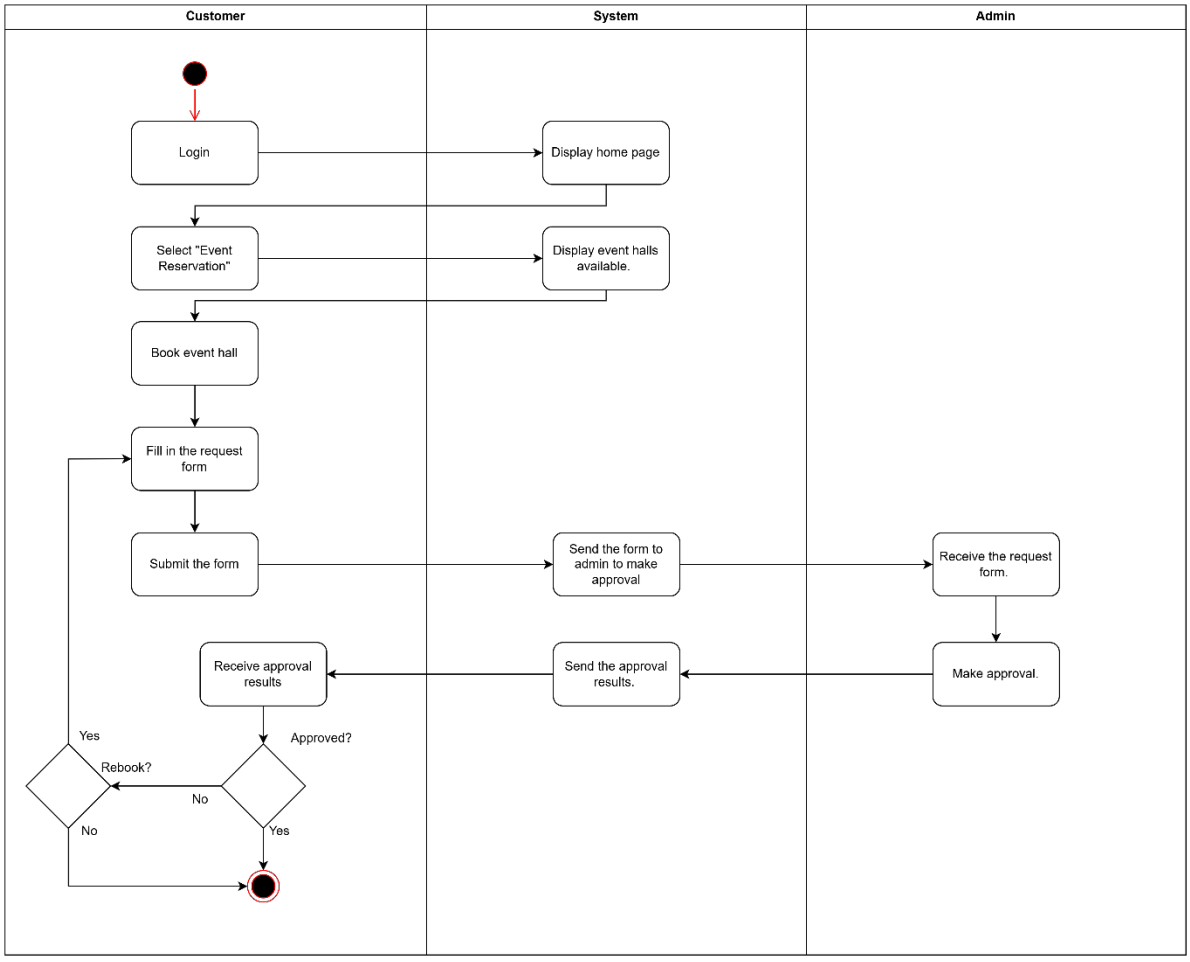


Figure 9.1.1 Vincent's Activity Diagram

The activity diagram shows the workflow involved in booking the event hall and making approval of the bookings. The key components in this diagram include booking the event hall, submitting a request form, and making approval.

Below are the steps when the users (customers and admins) use the system:

1. The customer is required to login to the system to access the service available of the website.
2. The customer selects the “Event Reservation” selection to view the event hall available at the hotel.
3. The customer selects the interested event hall, reads the details, then fills in the request form and submits it.
4. The system will send the request form to the admin and let the admin approve.
5. The admin will review the request form, and then decide whether want to approve or reject the submission.
6. After making approval, the system will send the result received to the customer and let the customer know about it.
7. If it is rejected, the customer might have the selection to rebook again or terminate the session. If it is approved, the booking session will be terminated.

9.1.2 Implementation

PROCESS FLOW: Identify primary objectives, risk analysis, engineering, and evaluation

In the first phase, the development team need to **identify the primary objectives** of these processes. The event hall booking process aims to let the customer request for rent space in the event hall by submitting a form and let the admin make approval. This process can highly streamline the whole booking process since just online instead of calling the staff and booking manually. The admin can review the form submitted by the customers, and then decide whether to approve or reject the request.

In the second phase, the development team needs to do a **risk analysis** by identifying the potential risk. For example, maybe some of the customers will submit incomplete or incorrect request forms, making the administrator face difficulties and conflicts while understanding and approving the requests received. Not only that, double bookings or unavailable time slots are counted as another part of the risks. Therefore, real-time form validation and live availability tracking should be implemented as soon as possible to reduce both admin and customer workload.

In the third phase, which is the **engineering** phase, the team should focus on both front-end and back-end development. A user-friendly event hall request form for customers and a dashboard of requests received for admin must be created. In addition, the team should integrate the system with a database for availability tracking and customer records. Besides, a notification functionality for both admins and customers is encouraged to notify them if the admin receives any booking requests and the customer receives updates upon their requests.

Lastly, usability testing with the potential users (customers and admins) should be conducted in the **evaluation and testing** phase. They need to try potential risks that could happen and test whether they are solved successfully or not. By gathering feedback from the users, the development team can make adjustments and enhancements to make the system become better.

9.2 Wang Jim Loong (TP075739)

9.2.1 Introduction

Use case: Update cleaning status (UC13)

Actor: Room Cleaner

Description: Room cleaner updates the cleaning status of a room.

Use case: Report issue

Extends: Update Cleaning Status

Actor: Room Cleaner, Admin

Description: Room cleaner reports any issue or damage in rooms or facilities.

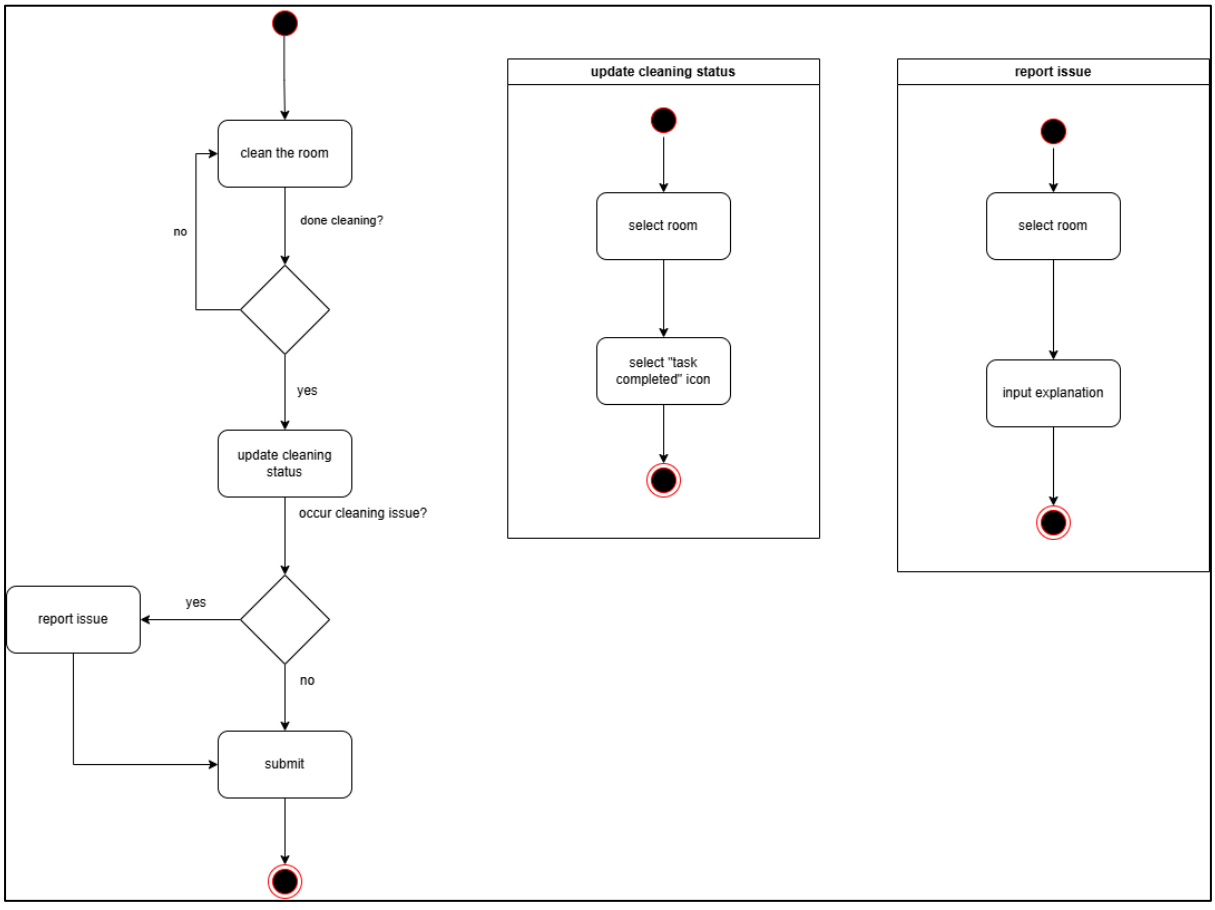


Figure 9.2.1 Wang's Activity Diagram

After cleaning a room, the room cleaner navigates to the *Room Status* section in the system. The system displays the room details assigned to the specific cleaner. To update the cleaning status, the cleaner selects the room number and reflects the current cleaning progress. If any issues or damages are found during cleaning, the cleaner selects the room number again, enters

the issue details in the provided field, and clicks the "Submit" button. This action logs the issue in the system and notifies the relevant personnel for further action.

9.2.2 Implementation

Objectives: The primary objectives are **to ensure that the room cleaners clean the rooms correctly and that the system appropriately updates the cleaning status**. Additionally, the system provides an **efficient mechanism for reporting any issues that arise** during the cleaning process, allowing the management team to take action accordingly.

Risk analysis: The room cleaner may sometimes **forget to update the cleaning status** after completing a task. This can lead to data inconsistencies and inaccurate records. In addition, this may disrupt the workflow and cause scheduling conflicts. Additionally**, software glitches** or **connectivity issues** might cause failures in log updates or issue reporting. Furthermore, **unclear or incomplete explanations** in issue reports will cause misunderstandings for the management team which may lead to incorrect or delayed actions to address the problem.

Engineering: The engineering focuses on creating a user-friendly system that allows room cleaners to easily navigate and use its functions. A **visible and functional "task completed" icon** must be implemented to expedite status updates. **The issue reporting feature should include a hotel room number selection option** and a **text box for users to input detailed explanations**. To ensure dependable data management and retrieval, a database is also needed to store the room cleaning statuses and issues reported.

Evaluation: The evaluation will focus on key performance metrics like the **accuracy of cleaning status updates**, **efficiency of issue reporting**, and the **resolution time for reported problems**. Other than that, **User feedback** will be gathered to identify areas for improvement. The system's capability to reduce operational delays and enhance communication between room cleaners and management will be assessed to ensure it effectively achieves its intended objectives.

9.3 Wang Zi Jie (TP076251)

9.3.1 Introduction

Use case: Arrange booking (UC25)

Actor: Coordinator, Admin

Description: Coordinator arranges a booking for the customer, such as for rooms, or event halls.

Use case: Contact customer (UC26)

Extends: Arrange booking

Actor: Coordinator

Description: Coordinator contacts the customer if they face any issues while processing the customer's booking.

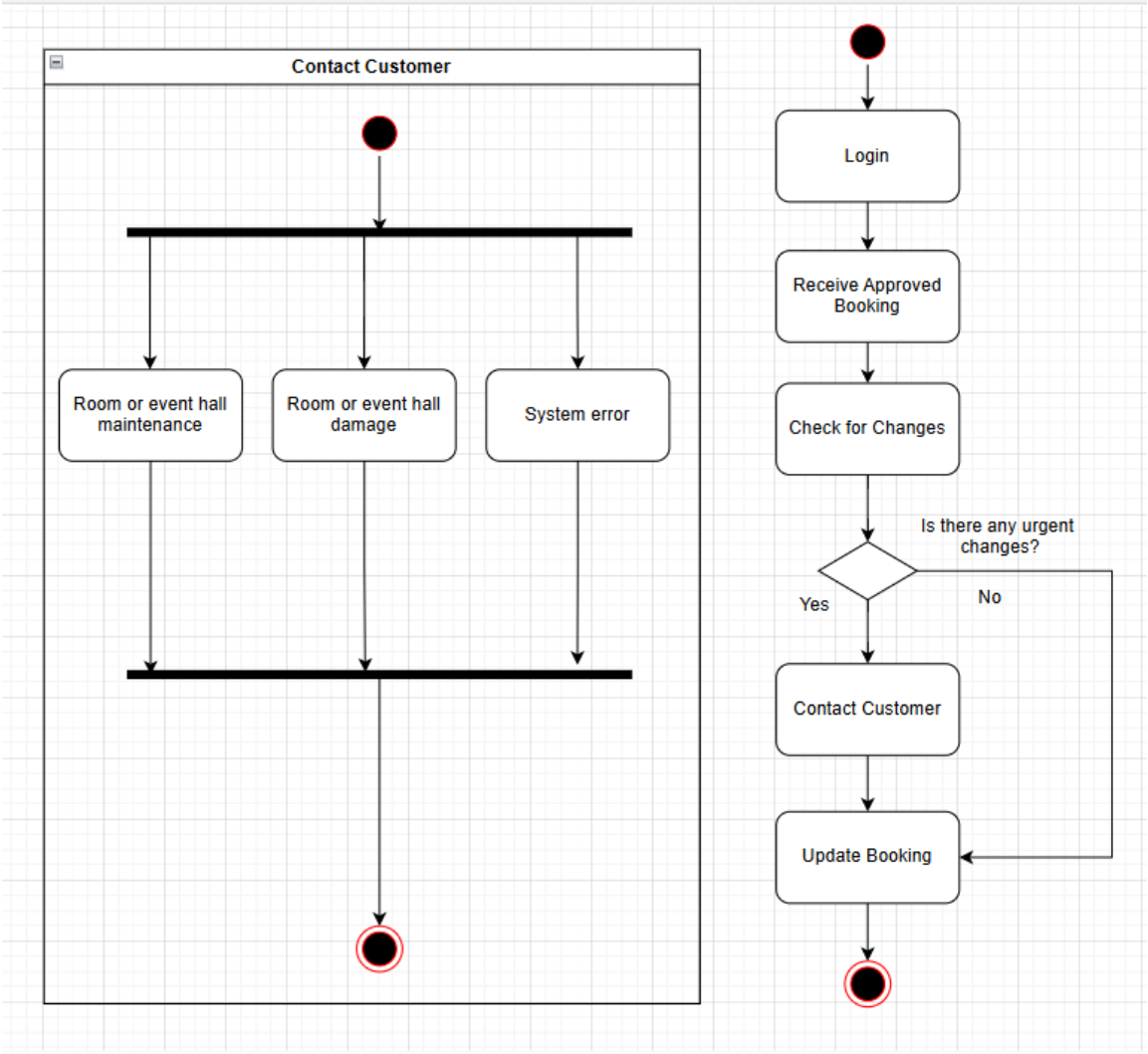


Figure 9.3.1 Wang's Activity Diagram

The activity diagram shows a detailed workflow about how the coordinator manages the approved booking. The primary focus is to communicate with the customer by phone when facing some changes due to unforeseen issues, such as room or event hall maintenance, damages, or system errors.

Below are the steps when the coordinator uses the system:

1. The coordinator must log in to the system before starting the process.
2. After login, the coordinator will receive an approved booking from the system.
3. The coordinator reviews the booking for any urgent changes, such as room or event hall maintenance, damage, or system errors.
4. Contact the customer by phone if changes affect their booking.
5. Updating the booking information in the system.

This process ensures the customer can receive the booking information timely and accurately, enhancing user experience and operational efficiency.

9.3.2 Implementation

There will be 4 phases that will be implemented in the arranged booking system, which include objectives defined, risk analysis, engineering, and evaluation.

The first phase is **objectives defined**, which need to clarify the goal of this arranged booking system. In this activity diagram, the system must allow the coordinator to log in securely, receive approved bookings, detect and address changes such as maintenance, damages, or system errors, notify customers promptly via phone, and update booking details.

Moreover, the second phase will be the **risk analysis**. It points out the potential risk in this system, which might cause the system failure. For example, system downtime, which is when the system fails to notify the approved bookings to the coordinator, and communication delays, which are when there are some problems while contacting the customer.

Furthermore, the third phase is **engineering**, which will start to implement the system step by step. The start will be creating a login module for the coordinator to access the system, a feature that can receive the approved booking, a feature that can detect urgent changes, a notification module that can communicate with the customers, and a booking update module that the coordinator can use to update the newest information to the system.

The last phase is **evaluation**, which determines if the system meets the user requirements by testing it. For example, the user can test some scenarios, such as notification and urgent changes features, to address system bugs.

In conclusion, the Spiral Model is an effective approach for developing the system, as it determines risks like system downtime and communication delays. Besides that, the activity diagram provides a clear workflow, helping ensure the system is developed and refined to meet user needs efficiently.

9.4 Wong Jun Ming (TP076078)

9.4.1 Introduction

Use case: Update cleaning status (UC17)

Actor: Admin

Description: Admin generates different types of reports such as financial, customer, and room/event hall reports.

Use case: Report issue

Actor: Room Cleaner, Admin

Description: Room cleaner reports any issue or damage in rooms or facilities.

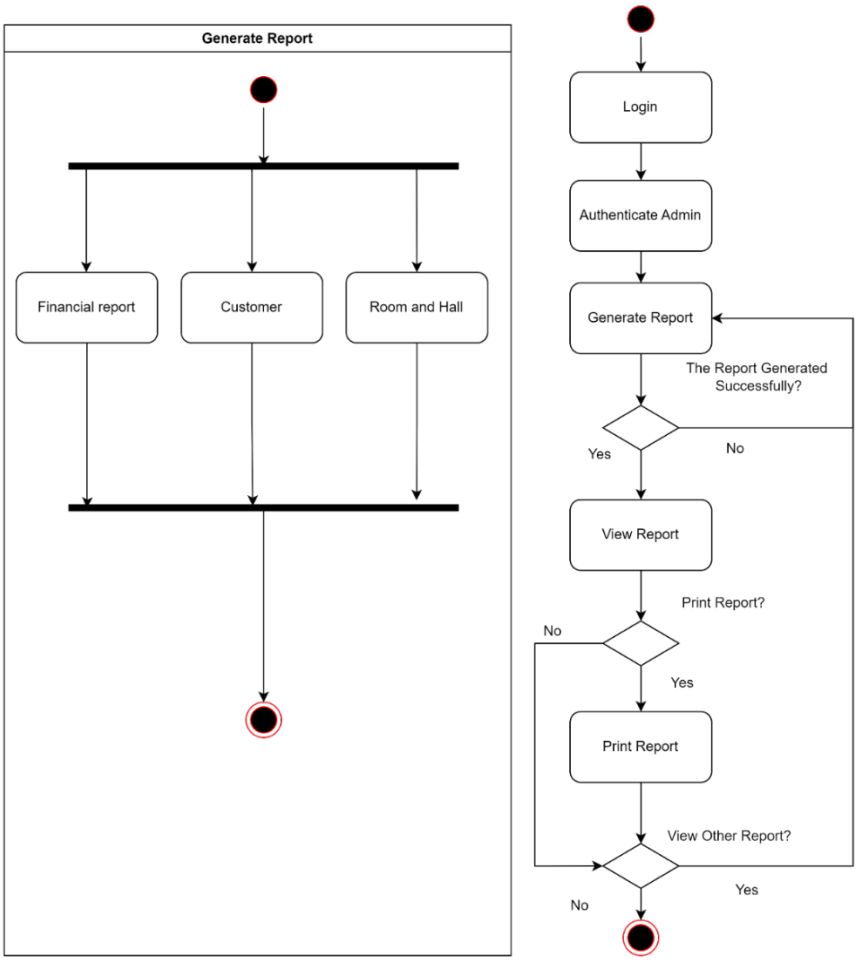


Figure 9.4.1 Wong's Activity Diagram

The activity diagram shows the workflow for generating and printing the different types of reports with the administrative system. The key components in this diagram include Login, Authenticate Admin, Generate Report, View Report, and Print Report.

Below are the steps when the admin uses the system:

1. The admin starts by logging into the system to initiate the process.
2. The system verifies the admin's credentials to ensure they have the required permissions.
3. After authentication, the admin selects the Generate Report function.
4. The generate report function can generate three types of reports that are Financial Report, Customer Report, and Room and Hall Report.
5. Once the report is generated, the admin can view the report on the system.
6. If the report needs a hard copy, the admin can select to print the report.
7. The process ends once the admin has either viewed or printed the report.

9.4.2 Implementation

The spiral model is an iterative software development process that integrates elements of both design and prototyping. It is especially suited for managing risk effectively during complex system development which supports well with the report generation system in this activity diagram.

In the planning phase, the primary objective is to create a user-friendly and efficient administrative system for report generation and management. For example, developers are identifying specific requirements such as generating, viewing, and printing different types of reports such as customer reports, financial reports, and room and event hall reports.

In the risk analysis phase, developers need to list out the risks like security risks, scalability risks, and usability. For instance, developers should ensure that the user authentication is secure to prevent unauthorized access. Then, the system can handle an increase in data. This is due to ensuring that the generating of complex reports does not impact the system's performance. The user interface of the system must be intuitive. So, the non-technical users can easily generate, view, and print the reports.

In the engineering phase, the developers start by building a basic version of the prototype that allows to log in, select the type of report, and view the report. For instance, implement login and authentication first and then followed by generating reports, and viewing the report.

In the evaluation phase, the system will be evaluated based on user feedback and technical tests. For example, after the authentication module is completed, it will be tested for security risks and usability issues.

In conclusion, the spiral model helps address key risks early in the development of the report generation system. This is due to ensuring it evolves into a consistent and secure system. The activity diagram clearly shows how users interact with the system at each step while the spiral model ensures that this interaction is refined through continuous iteration and feedback.

10.0 Testing (Individual Part 3)

10.1 Vincent Teo Kai Qi (TP075160)

10.1.1 Testing Method

Based on the use cases selected which are “Book Event Hall” and “Make Approval”, two actors will be involved in these cases. The system will allow the customers to fill in the request form for the event hall and the admins can review the request form received by making approval then the system will send back the results to the customers. For these cases, black box testing is chosen since it is the most suitable testing method to gather the users' feedback and satisfaction.

Black box testing is a technique in which testers examine a system's operation without knowing its internal workings by providing inputs and observing outputs to determine its reactions to various user activities, response time, usability, and reliability (Sharadin, 2023). The testing focuses more on the front end, which includes the functional and non-functional requirements, instead of the back-end coding and data management. The potential system users will become the testers of the developed system which is not required to have programming knowledge in this testing. It aims to verify that the system behaves as expected, ensure that all the functional requirements are met, detect errors or bugs by viewing the outputs, validate the accuracy of the data, and test the user-friendliness of the system developed without requiring knowledge of programming language.

For instance, the potential users of the “Book Event Hall” are the customers who would like to book the event hall for various purposes such as weddings, meetings, events, company meetings, and other special events, and need a simple platform to check availability, make reservations, and customize the event space to their specifications. The testers need to predict them as the customers and use the system, then give feedback based on the output given by the system. In the "Make Approval" use case, the admin role is tested by verifying that all inputs and outputs align with the intended functionality and meet satisfaction criteria, ensuring the system is easy to use, intuitive, and effective for administrative purposes.

One of the most important advantages of using black box testing is the testers do not need to learn implementation details including the coding techniques for the system (Sharadin, 2023). Testers just need to understand the system's functional requirements, give several inputs, and analyze the outputs to decide whether the system functions as expected and meets the expectations of the users. However, black box testing requires test case prioritization, as

evaluating all feasible user pathways is often difficult due to the complexity and variety of situations that the system might face.

In conclusion, the development team should prioritize functional needs by conducting black-box testing and gathering user feedback to evaluate the system's effectiveness. Based on client feedback and satisfaction, they must determine whether the system meets expectations or needs to be improved. If the testers are dissatisfied, additional changes should be made to address the highlighted issues and improve the system's performance.

10.1.2 Test Script

Refer to G10_TestPlan_Sheet1

10.2 Wang Jim Loong (TP075739)

10.2.1 Testing Method

In the use cases “Update cleaning status” and “report issues” act as significant functions in the Swiss Garden Hotel’s system. The “Update cleaning status” use case allows the room cleaner to mark their status for each specific cleaning task given easily. In addition, the admin can assign the cleaning task to the room cleaner more effectively due to the visible status. Moreover, the “report issues” use case allows the room cleaner to report any issues met when cleaning tasks are conducted. For both use cases, black box testing is conducted to test the functionality.

Black box testing is a type of software testing technique that is used to identify the functionality of the application (Verma, Khatana, & Chaudhary, 2017). The black box testing tests the system without any prior knowledge of its internal workings (Imperva, 2024). Moreover, programming knowledge is not required when conducting the black box testing. For example, the “Update cleaning status” and “report issues” use cases are performed by the tester in the front end of the system. accordingly, conducted to ensure that the status can be updated accordingly and the report written by the tester can be fetched to the admin side for them to take any future actions if needed. By using black box testing, issues like incorrect outputs, interface errors, or misinterpretations of requirements can be found easily. Thus, this testing can ensure “Update cleaning status” and “report issues” use cases with good performance and usability due to it supports multiple types of testing (i.e., functional testing, non-functional testing, security testing, and so on) that can address different aspects of the software quality.

Furthermore, the advantage of black box testing is the tester can be a non-technical person because this testing does not require any implementation knowledge (Verma, Khatana, & Chaudhary, 2017). Then, the testing is conducted in alignment with the needs of the customer (Verma, Khatana, & Chaudhary, 2017).The testing can involve evaluating all possible inputs and outputs that a typical user might encounter. Moreover, black box testing is efficient for large code-based systems. It is because the tester does not need to understand the code that runs the system therefore the tester can start testing without any understanding of the complex code base (Bakharev, 2023).

10.2.2 Test Script

Refer to G10_TestPlan_Sheet2

10.3 Wang Zi Jie (TP076251)

10.3.1 Testing Method

For the use case “Arrange Booking” and “Contact Customer,” there were both functionalities of the system. The “Arrange Booking” use case represents the coordinator managing bookings by addressing urgent changes such as maintenance, damages, or system errors. Then, the “Contact Customer” use case allows the coordinator to inform the customers when changes are affecting their bookings. To validate these use cases, the chosen testing method will be black box testing.

Black box testing is a sort of software testing where the tester concentrates on verifying the functionality given on specifications and requirements or needs rather than the software’s implementation details or internal knowledge (GeeksforGeeks, Black Box Testing – Software Engineering, 2018). This makes it suitable for validating the functionalities such as "Arrange Booking" and "Contact Customer."

Besides that, for the two-use cases "Arrange Booking" and "Contact Customer,” black box testing ensures that the system will work correctly by handling valid bookings, addressing urgent changes, and notifying customers about those changes. Furthermore, this method allows testers to test the system without requiring programming knowledge, which improves the efficiency of testing functionalities such as the booking arrangement and notification delivery (Javatpoint, 2022). Additionally, black box testing can imitate real-world scenarios and help identify issues such as improper booking format, incorrect notification, and missing updates.

The advantages of this method are that the tester does not need expertise in programming skills or functional knowledge while testing the system, making it accessible and easier to handle the system (GeeksforGeeks, Black Box Testing – Software Engineering, 2018). Moreover, this method also allows for testing a wide range of inputs, including valid and invalid scenarios and unexpected user behaviors.

In conclusion, black box testing enhances the system’s reliability, usability, and functionality by ensuring that it meets user expectations and addresses their concerns (Javatpoint, 2022). Using black box testing for the “Arrange booking” use case ensures that the system processes booking correctly, and the “Contact Customer” use case can validate that notifications are delivered successfully.

10.3.2 Test Script

Refer to G10_TestPlan_Sheet3

10.4 Wong Jun Ming (TP076078)

10.4.1 Testing Method

Based on the selected use cases, the use cases of "Generate Report" and "View Report" represent significant functionalities. This is because the system allows the user to generate the report and view the report. The "Generate Report" use case allows the user to generate the reports. However, the "View Report" use case allows the user to view the generated reports in a user-friendly system. The black box testing is chosen to test the validating the functions of "Generate report" and "View report".

Black box testing is a type of testing that only focuses on the external structure of the implementation that needs to be tested (Jain, 2018). The black box testing is evaluating the functionality of the software without needing to understand its internal code structure or implementation details (Black Box Testing | What it is, Types , Techniques & Examples, 2023). For example, the use cases "Generate Report" and "View Report" involve user interface functionalities. This aims to ensure that the system generates the expected results for the user such as the function of generating a report showing a successfully generated message and displaying the report. Black box testing helps ensure that these functions work as expected from the end user's viewpoint. Moreover, testers conduct this type of testing by providing inputs and checking outputs to ensure that the software can work as expected meeting the requirements (Jain, 2018)In generating and viewing reports, black-box testing is very useful for verifying whether the system produces the expected output given specific inputs (Das, 2024). It focuses on determining whether the report meets the requirements and accurately reflects the provided data regardless of how the system processes this data internally. Since use cases focus on front-end-user interactions, black-box testing is more relevant.

The advantage of black box testing is the tester does not need any code knowledge (Davis, 2022). Black box testing does not need testers to understand the internal implementation of the system like coding or programming. The black box testing is suitable for validating the functions of generate report and view report. The focus is on testing the system's response by entering different inputs. This is due to ensuring the output is correct, and the tester not need to worry about how the system processes these inputs. Moreover, testers verify that new updates do not break the software’s user-visible functionality (Vijay, Black Box Testing: An In-depth Tutorial with Examples and Techniques, 2011). For generating and viewing reports, testers can focus on testing various inputs, conditions, and report type variants

to ensure that all scenarios function correctly. This helps identify defects related to input processing, usability, and expected outcomes.

To use black box testing for the "Generate Report" and "View Report" test cases, testers should verify the software's functionality without understanding the internal code structure. For the "Generate report" test case, testers should create the input combinations to make sure the system can generate the report. Then, the testers need to check the report to make sure that it does not have any error handling. For "View Report", the testers must ensure that the generated reports display the correct data. This approach ensures that the report generation and viewing functions meet user needs and provide a seamless experience.

10.4.2 Test Script

Refer to G10_TestPlan_Sheet4

References

1202g General Duty Scanner | Honeywell. (2021). Retrieved from https://automation.honeywell.com/us/en/products/productivity-solutions/barcode-scanners/general-purpose-handheld/voyager-1202g-general-duty-scanner

9 Key benefits of using the agile methodology. (06 July, 2024). Retrieved from kissflow: https://kissflow.com/project/agile/benefits-of-agile/

Abrol. (04 October, 2024). *12 Reasons "Why a Multi-Language Website Is Beneficial to Your Business?"*. Retrieved from Digittrix: https://www.digittrix.com/blogs/12-reasons-why-a-multi-language-website-is-beneficial-to-your-business#:~:text=It%20Improves%20Customer%20Confidence%20and,they%20can%20expect%20from%20it.

Ahmad, N. (24 March, 2023). *Acceptance Testing Tutorial: A Comprehensive Guide With Examples And Best Practices*. Retrieved from www.lambdatest.com: https://www.lambdatest.com/learning-hub/acceptance-testing

Anshika. (01 May, 2024). *What is component design?* Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/what-is-component-design/

Bakharev, N. (12 October, 2023). *Black Box Testing: Types, Techniques, Pros and Cons*. Retrieved from Bright: https://brightsec.com/blog/black-box-testing-types-techniques-pros-and-cons/

Black Box Testing | What it is, Types , Techniques & Examples. (28 November, 2023). Retrieved from Testsigma Blog: https://testsigma.com/guides/black-box-testing/

Booking plugins vs. Manual scheduling. (5 February, 2024). Retrieved from wp booking system: https://www.wpbookingsystem.com/blog/booking-plugins-vs-manual-scheduling/

Brush, & Silverthorne. (15 November, 2022). *Agile software development*. Retrieved from TechTarget: https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development

Candelario. (07 October, 2024). *Agile Design Process: Definition, Main Principles, and Benefits*. Retrieved from DesignRush: https://www.designrush.com/agency/ui-ux-design/trends/agile-design-process#:~:text=The%20steps%20to%20the%20agile,all%20while%20delivering%20quality%20results.

Cascarano, M. A. (22 June, 2023). *The advantages of Windows Server 2019 Standard - Licendi*. Retrieved from https://licendi.com/en/blog/the-advantages-of-windows-server-2019-standard/

Chandana. (24 Septemebr, 2024). *Scrum Project Management: Advantages and Disadvantages*. Retrieved from Simplilearn.com: https://www.simplilearn.com/scrum-project-management-article

Chatsimple.ai. (30 March, 2024). *7 Disadvantages of chatbots and how to fix them*. Retrieved from Chatsimple.ai: https://www.chatsimple.ai/blog/disadvantages-of-chatbots

Checkout.com. (n.d.). *Multi-currency payments: benefits and use cases | Checkout.com*. Retrieved from Checkout.com: https://www.checkout.com/blog/multi-currency-payments

Cisco RV340, RV345, RV345P, and RV340W Dual WAN Security Router Data Sheet. (22 December, 2020). Retrieved from https://www.cisco.com/c/en/us/products/collateral/routers/small-business-rv-series-routers/datasheet-c78-742350.html

Contributor, T. (15 August, 2019). *Spiral Model*. Retrieved from TechTarget Software Quality: https://www.techtarget.com/searchsoftwarequality/definition/spiral-model

Coursera. (2023, September 11). *What Is Machine Learning? Definition, Types, and Examples*. Retrieved from Coursera: https://www.coursera.org/articles/what-is-machine-learning

Dąbrowska, M. (20 June, 2017). *What is Scrum in software engineering projects?* Retrieved from Miquido: https://www.miquido.com/blog/foster-greatness-embrace-scrum/

Das, S. (10 August, 2024). *What is Black Box Testing: Types, Tools & Examples*. Retrieved from BrowserStack: https://www.browserstack.com/guide/black-box-testing

Davis, R. (9 November, 2022). *What is Black Box Testing: Advantages and Disadvantages*. Retrieved from www.invensis.net: https://www.invensis.net/blog/black-box-testing-advantages-disadvantages

Dell. (2024). *OptiPlex 7070 Tower and Small Form Factor Desktops | Dell USA*. Retrieved from https://www.dell.com/in-en/work/shop/desktop-and-all-in-one-pcs/optiplex-7070-desktop/spd/optiplex-7070-desktop

Dunse. (6 April, 2021). *The Most Popular Payment Methods in Malaysia*. Retrieved from Shuttle: https://www.shuttleglobal.com/blog/the-most-popular-payment-methods-in-malaysia/#:~:text=In%20Malaysia%2C%20alongside%20traditional%20payment,and%20Samsung%20Pay%20gaining%20prominence.

Engineering, S. E.–S. (03 January, 2024). *Software Evolution – Software Engineering*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/software-engineering-software-evolution/

Functional vs. Non Functional Requirements. (22 October, 2024). Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/functional-vs-non-functional-requirements/

Gaille, L. (2020). *15 Advantages and Disadvantages of a Waterfall Model*. Retrieved from VITTANA: https://vittana.org/15-advantages-and-disadvantages-of-a-waterfall-model

GeeksforGeeks. (25 May, 2018). *Black Box Testing – Software Engineering*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/software-engineering-black-box-testing/

GeeksforGeeks. (18 March, 2018). *Waterfall Model - Software Engineering*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/waterfall-model/

GeeksforGeeks. (9 August, 2019). *System Testing - GeeksforGeeks*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/system-testing/

GeeksforGeeks. (18 April, 2019). *Unit Testing | Software Testing*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/unit-testing-software-testing/

GeeksforGeeks. (04 October, 2024). *Non-Functional Requirements in Software Engineering*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/non-functional-requirements-in-software-engineering/

GeeksforGeeks. (23 May, 2024). *Software Processes in Software Engineering*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/software-processes-in-software-engineering/#what-are-software-processes

GeeksforGeeks. (22 May, 2024). *What is RUP(Rational Unified Process) and its Phases?* Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/rup-and-its-phases/

GeeksforGeeks. (30 July, 2024). *What is Spiral Model in Software Engineering?* Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/software-engineering-spiral-model/#what-is-the-spiral-model

Gotra, V. (29 July, 2021). *QASource's Comprehensive Guide to Chatbot Testing*. Retrieved from blog.qasource.com: https://blog.qasource.com/industry-insights/a-guide-to-chatbot-testing

Harrel, W. (20 September, 2018). *Brother MFC-L3770CDW Review*. Retrieved from PCMAG: https://www.pcmag.com/reviews/brother-mfc-l3770cdw

Hils, & T. (2024, March 27). *The top 10 benefits of Chatbots in customer service*. Retrieved from HelpScout: https://www.helpscout.com/blog/benefits-of-chatbots-in-customer-service/

Hoory, L., & Bottorff, C. (2022). *What Is Waterfall Methodology And How Do I Use It?* Retrieved from Forbes Advisor: https://www.forbes.com/advisor/business/what-is-waterfall-methodology/

Hospitality POS System | Hospitality Point of Sale Software | Square. (2024). Retrieved from Square: https://squareup.com/au/en/solutions/hospitality

HP. (2024). *HP Engage One Prime*. Retrieved from HP: https://www.hp.com/gb-en/pos-systems/engage-one-prime.html?msockid=1296c57ee1f2696c330fd6f1e06e68e1

iderawpadmin. (20 10, 2022). *The Purpose of Verification and Validation in Software Testing What Is Verification and Validation in Software Testing?* Retrieved from Ranorex: https://www.ranorex.com/blog/validation-in-software-testing/

iDS-TCM403-A. (12 September, 2023). Retrieved from https://www.hikvision.com/my/products/ITS-Products/Smart-Monitoring-Systems/Smart-Monitoring-Cameras/ids-tcm403-a/

Imperva. (6 December, 2024). *Black Box Testing*. Retrieved 6 December, 2024, from Imperva: https://www.imperva.com/learn/application-security/black-box-testing/

Inc, G. (n.d.). *Veeam Reviews, Ratings & Features 2022 | Gartner Peer Insights*. Retrieved from trustradius.com: https://www.gartner.com/reviews/market/enterprise-backup-and-recovery-software-solutions/vendor/veeam

INC., S. (2020). *DiskStation*. Retrieved from SYNOLOGY: https://global.download.synology.com/download/Document/Hardware/DataSheet/DiskStation/20-year/DS920+/enu/Synology_DS920_Plus_Data_Sheet_enu.pdf

Ingenico. (10, 2022). *DESK-5000*. Retrieved from www.ingenico.com: https://ingenico.com/sites/default/files/resource-document/2022-10/DESK-5000_datasheet_EN_OCT22.pdf

Insights, T. A. (22 October, 2022). *What Is Validation Testing In Software Development?* Retrieved from Indeed Career Guide: https://www.indeed.com/career-advice/starting-new-job/what-is-validation-testing

Jackie. (21 January, 2016). *6 Advantages of Using WPML Plugin in WordPress*. Retrieved from Noo Theme: https://nootheme.com/6-advantages-of-using-wpml-plugin-in-wordpress/

Jain, M. (20 December, 2018). *Differences between black box testing vs white box testing - geeksforgeeks*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/

Javatpoint. (2022). *Advantages and Disadvantages of Black-box Testing*. Retrieved from Javatpoint: https://www.javatpoint.com/advantages-and-dsadvantages-of-black-box-testing

Johnson. (25 March, 2024). *Top 7 Advantages of Offering Multiple Payment Options for eCommerce Businesses*. Retrieved from Invensis: https://www.invensis.net/blog/multiple-payment-options

Joseph. (29 March, 2021). *What are the main advantages of the spiral model?* Retrieved from QASource: https://blog.qasource.com/software-development-and-qa-tips/what-are-the-main-advantages-of-the-spiral-model

Kaspersky Endpoint Security for Business (Enterprise Solutions) | Kaspersky. (n.d.). Retrieved from https://usa.kaspersky.com/enterprise-security/endpoint

Kaspersky. (n.d.). *Kaspersky Endpoint Security for Windows | Kaspersky*. Retrieved from https://www.kaspersky.com/small-to-medium-business-security/endpoint-windows

Khalid, H. (19 February, 2024). *What are the limitations of Online Ordering System?* Retrieved from ENATEGA: https://enatega.com/limitations-of-online-ordering-system/

Khrupa, A. (11 July, 2022). *What Is Software Evolution and Maintenance*. Retrieved from QAREA: https://qarea.com/blog/what-is-software-evolution-and-maintenance

King. (n.d.). *5 Advantages of Automating Manual Processes In Your Business*. Retrieved from GrowthForce: https://www.growthforce.com/blog/5-advantages-of-automating-manual-processes-in-your-business#:~:text=Higher%20Efficiency%20and%20Productivity&text=Automation%20relieves%20mundane%20tasks%2C%20allowing,that%20propel%20the%20organization%20forward.

KnowledgeHut. (14 June, 2019). *What are the Top 15 Power BI Benefits & Advantages?* Retrieved from Knowledgehut.com: https://www.knowledgehut.com/blog/business-intelligence-and-visualization/power-bi-advantages-benefits

Kulak, O., Cebi, S., & Kahraman, C. (September, 2010). Applications of axiomatic design principles: A literature review. *Expert Systems with Applications*, 6705-6717. doi:10.1016/j.eswa.2010.03.061

Lavelle. (8 February, 2024). *Manual process vs automated process: Which is better?* Retrieved from Enable: https://www.enable.com/blog/manual-process-vs-automated-process-which-is-better

Malaysia Online Payment Gateway. (2024). Retrieved from iPay88 | Malaysia Online Payment Gateway: https://www.ipay88.com/

Martinbaldwin. (13 January, 2024). *Using Currency Data to Your Advantage: Examining Fixer io's Free Forex API*. Retrieved from Medium: https://medium.com/@martinbaldwin127/using-currency-data-to-your-advantage-examining-fixer-ios-free-forex-api-ef03acbb4af6

Microsoft. (2024). Retrieved from Microsoft Support: https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5

m-kauppinen. (7 June, 2024). *How to perform localization testing - Globalization*. Retrieved from Microsoft.com: https://learn.microsoft.com/en-us/globalization/testing/how-to-perform-localization-testing

Muchmore, M. (17 June, 2024). *Windows 11 Home vs. Pro: What Are the Differences?* Retrieved from PCMAG: https://www.pcmag.com/comparisons/windows-11-home-vs-pro

Mufti, Mittal, & Gupta. (11 March, 2021). *A Review on Amazon Web Service (AWS), Microsoft Azure & Google Cloud Platform (GCP) Services.* Retrieved from European Union Digital Library (EUDL): https://eudl.eu/doi/10.4108/eai.27-2-2020.2303255

MySQL. (2019). *MySQL :: MySQL Enterprise Edition*. Retrieved from mysql.com: https://www.mysql.com/products/enterprise/

Nair, H. S. (12 February, 2024). *System Integration Testing: A Comprehensive Guide With Examples And Best Practices*. Retrieved from www.lambdatest.com: https://www.lambdatest.com/learning-hub/system-integration-testing

Nikolaieva. (20 May, 2024). *10 Best Software Development Methodologies*. Retrieved from uptech: https://www.uptech.team/blog/software-development-methodologies

Nikolaieva, A. (20 May, 2024). *10 best software development methodologies*. Retrieved from Uptech: https://www.uptech.team/blog/software-development-methodologies

Nixon. (18 July, 2022). *The Benefits of PHP, MySQL, JavaScript, and CSS.* Retrieved from O'Reilly Online Learning: https://www.oreilly.com/library/view/learning-php-mysql/9781449337452/ch01s02.html

O'Donnell, C. (2 January, 2024). *FDA Software Validation - 2020 Guide, Checklist & Template*. Retrieved from www.datacor.com: https://www.datacor.com/the-datacor-blog/fda-software-validation

OPERA Hotel Property Management Solutions (PMS). (2022). Retrieved from Oracle.com: OPERA Hotel Property Management Solutions (PMS)

Pollock, L. (22 August, 2023). *What Is an NLP Chatbot — And How Do NLP-Powered Bots Work?* Retrieved from Ultimate.ai: https://www.ultimate.ai/blog/ai-automation/how-nlp-text-based-chatbots-work

ProjectManager. (2022). *Waterfall Methodology*. Retrieved from ProjectManager: https://www.projectmanager.com/guides/waterfall-methodology

Robertson. (20 December, 2023). *Role-Based Access Control (RBAC)*. Retrieved from imperva: https://www.imperva.com/learn/data-security/role-based-access-control-rbac/#:~:text=Role%2Dbased%20access%20control%20(RBAC)%2C%20also%20known%20as,enable%20access%20to%20authorized%20users.

Sachdeva, S. (2016). Scrum Methodology. *International Journal Of Engineering And Computer Science, v*(6), 16792-16799. doi: 10.18535/ijecs/v5i6.11

Sahoo, P. K. (24 May, 2024). *What is software implementation?* Retrieved 5 October, 2024, from GeeksforGeeks: https://www.geeksforgeeks.org/what-is-software-implementation/

Scrum.org. (n.d.). *Ken Schwaber*. Retrieved from Scrum.org: https://www.scrum.org/team/ken-schwaber

Sharadin. (21 December, 2023). *Black Box Testing.* Retrieved from imperva: https://www.imperva.com/learn/application-security/black-box-testing/

Sharma. (25 December, 2020). *Top 12 Software Development Methodologies*. Retrieved from TatvaSoft: https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/

Sharma, K. (25 December, 2020). *Top 12 software development methodologies*. Retrieved from TatvaSoft: https://www.tatvasoft.com/blog/top-12-software-development-methodologies-and-its-advantages-disadvantages/

Singh. (16 July, 2024). *Spiral model in Software Engineering*. Retrieved from Shiksha: https://www.shiksha.com/online-courses/articles/spiral-model-in-software-engineering/

Sommerville Software Engineering (10 ed). (n.d.). Retrieved from CS 410/510 - Software Engineering Architectural Design: https://cs.ccsu.edu/~stan/classes/CS410/Notes16/06-ArchitecturalDesign.html

Stefanowicz, B. (22 May, 2024). *NLP Chatbot Tutorial: How to Build a Chatbot Using Natural Language Processing*. Retrieved from Tidio: https://www.tidio.com/blog/nlp-chatbots/#benefits

Susnjara, & Smalley. (14 February, 2024). *What is cloud computing?* Retrieved from IBM: https://www.ibm.com/topics/cloud-computing

Team, E. W. (6 July, 2023). *Pros and Cons of Survey Monkey - EducationalWave*. Retrieved from https://www.educationalwave.com/pros-and-cons-of-survey-monkey/

Team, I. E. (16 August, 2024). *RUP: Definition, Phases, Advantages and Best Practices*. Retrieved from indeed: https://www.indeed.com/career-advice/career-development/rup

Technology, D. (2020). *Adapt and scale with an expandable, virtualization-ready platform*. Retrieved from https://mlink-dell.com/wp-content/uploads/2022/04/M-Link-Dell-Poweredge-Two-Socket-Tower-Servers-Technical-Spec-Sheet.pdf

Verma, A., Khatana, A., & Chaudhary, S. (31 December, 2017). A Comparative Study of Black Box Testing and White Box Testing. *International Journal of Computer Sciences and Engineering, 5*(12), 301-304. Retrieved from www.ijcseonline.org

Vijay. (15 September, 2011). *Black Box Testing: An In-depth Tutorial with Examples and Techniques*. Retrieved from Softwaretestinghelp.com: https://www.softwaretestinghelp.com/black-box-testing/

Vijay. (20 January, 2023). *What is Acceptance Testing (A Complete Guide)*. Retrieved from www.softwaretestinghelp.com: https://www.softwaretestinghelp.com/what-is-acceptance-testing/

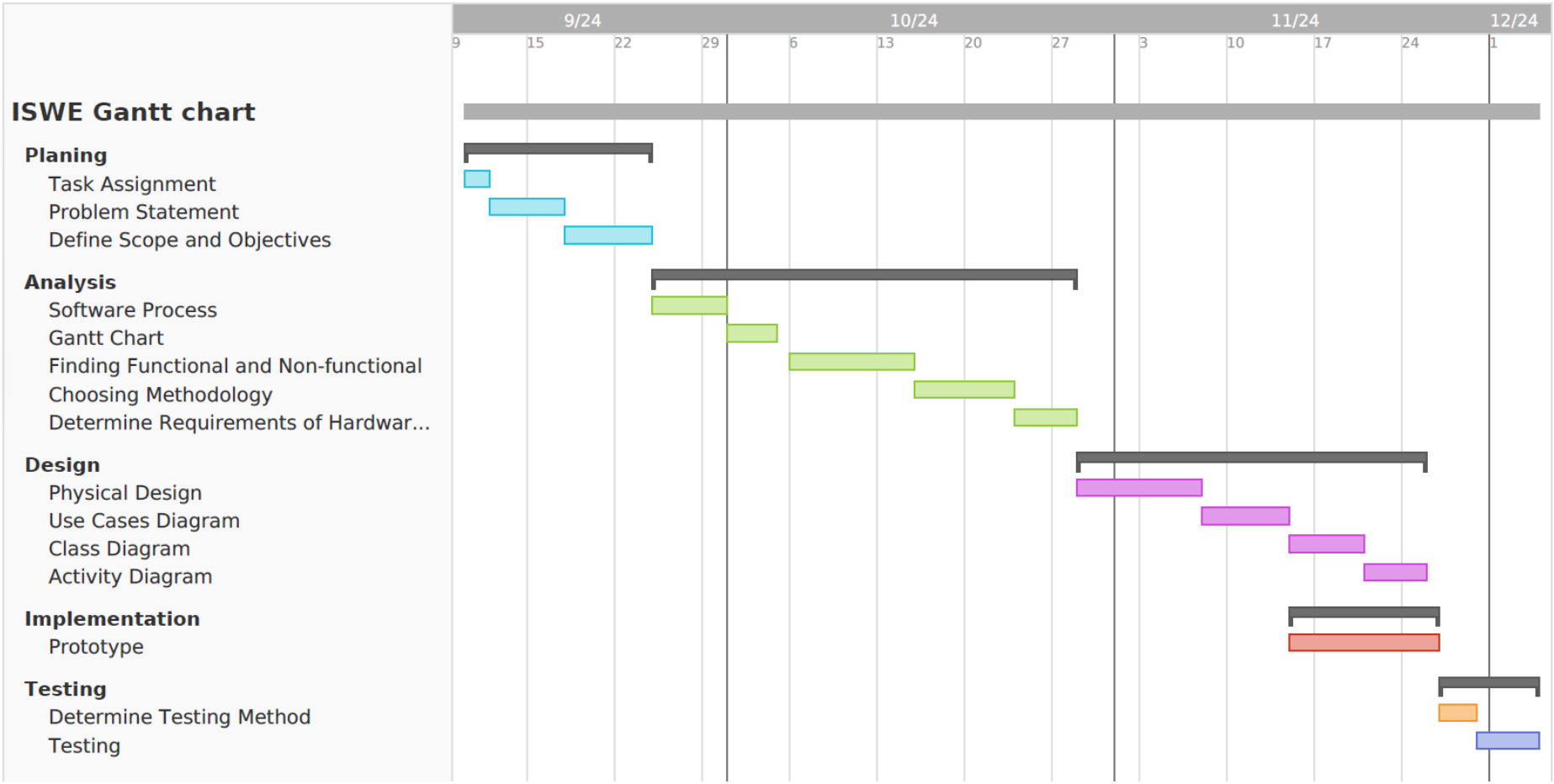
Why software design is important. (26 January, 2024). Retrieved 05 October, 2024, from IEEE COmputer Society: https://www.computer.org/resources/importance-of-software-design-is-important

Zhezherau. (16 November, 2022). *The Agile Software Development Life Cycle*. Retrieved from Wrike Agile Guide: https://www.wrike.com/agile-guide/agile-development-life-cycle/

Zuckerman, A. (20 February, 2019). *Zapier Review: Pricing, Pros, Cons & Features*. Retrieved from CompareCamp.com: https://comparecamp.com/zapier-review-pricing-pros-cons-features/

Appendix

Appendix 1: Gantt Chart

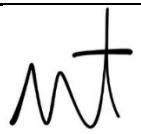


Figure_Appendix_1 Gantt Chart

Appendix 2: Workload Matrix

INTAKE: UCDF2307ICT(SE)	STUDENT NAME	VINCENT TEO KAI QI	WANG JIM LOONG	WANG ZI JIE	WONG JUN MING
System Name: SWISS GARDEN HOTEL MANAGEMENT SYSTEM	TP NO.	TP075160	TP075739	TP076251	TP076078

A. Group Component

	ASSIGNMENT COMPONENT	ALLOCATED MARKS	CONTRIBUTION PERCENTAGE	CONTRIBUTION PERCENTAGE	CONTRIBUTION PERCENTAGE	CONTRIBUTION PERCENTAGE	TOTAL %
Q1	Introduction	5	40.00	20.00	20.00	20.00	100
Q2	Planning and Requirements Analysis	5	25.00	25.00	25.00	25.00	100
Q3	Logical Design	10	20.00	30.00	20.00	30.00	100
Q4	Physical Design	10	30.00	30.00	20.00	20.00	100
Q5	Implementation	10	30.00	20.00	20.00	30.00	100
Q6	Prototype	10	40.00	20.00	20.00	20.00	100
Q7	Conclusion	5	20.00	20.00	40.00	20.00	100
Q4	Documentation & Report	5	25.00	25.00	25.00	25.00	100
	Total Marks and Contribution	60	29%	24%	24%	24%	
		Signature					

Table_Appendix_2 Workload Matrix

Appendix 3: Wireframe and Prototype

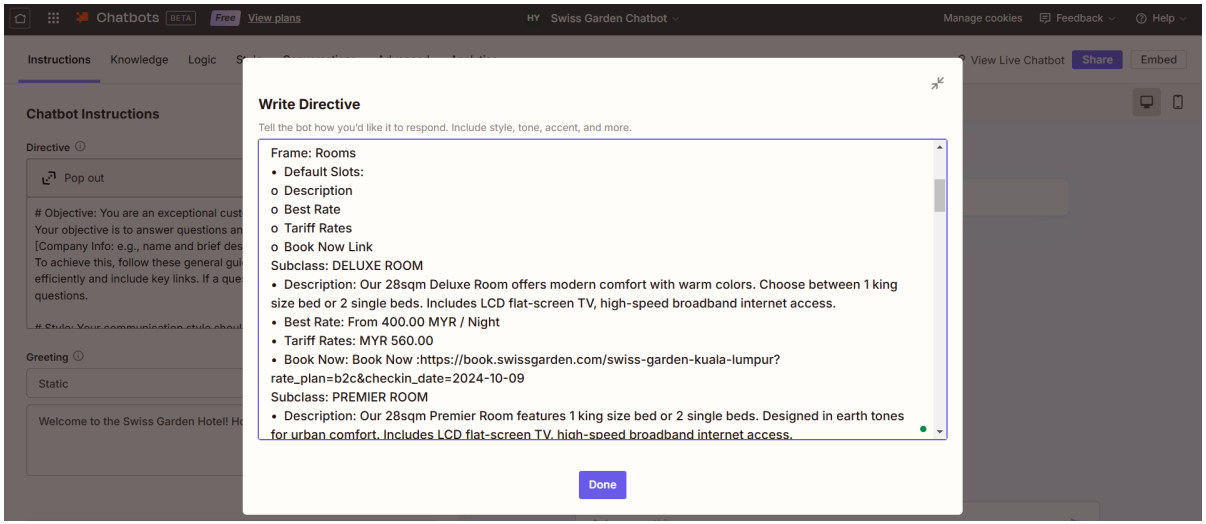
Figma Link: https://www.figma.com/design/8u2GDOaEm4nakLGoQTDrOl/ISWE-Assignment?node-id=0-1&t=ydESTf07DXhvbQe3-1

Our group decided to use Figma for designing both wireframes and prototypes (interface design).

Appendix 4: Additional Features – Chatbot (Zapier.com)

Zapier Link: https://swiss-garden-chatbot.zapier.app/

Our group decided to use Zapier.com as the tool for developing the AI-advanced chatbot for the Swiss Garden Hotel Management System since the system is already built-in with GPT-3.5 turbo, Natural Language Processing, and Machine Learning which can highly enhance user satisfaction by generating more accurate response.



Figure_Appendix_4 Gantt Chart

We use the frame representation as our knowledge base to represent knowledge about objects and events.