Software Requirements Specification

For

Restaurant Reservation System

Team Members:

- H.L.P.M.Shehara | 321425748
- C.S.Bandaragodage 721428828
- V.S.Ranasinghe | 421421850
- M.G.A.R. Deepthika | 121422462

Table of Contents

Introduction

1.1 Purpose

The purpose of this project is to develop a restaurant booking system that will allow customers to reserve tables at a restaurant easily.

The purpose of this document is to outline the requirements for the Restaurant Reservation System and seek approval for all requirements before proceeding to the design phase. The engineering team will use this document to create the system design. If any changes to requirements are requested after sign-off, they will be documented, and the effect on projectcosts, scope, and timelines will be analyzed and presented to the Projects in-charge Head of for approval.

1.2 Document Conventions

- Font: The font used in this document is Poppins, size 10 for descriptions and 12 for headings
- Heading Hierarchy: This document follows a heading hierarchy, where major sections are identified by a level one heading (e.g., "Introduction"), sub-sections by a level two heading (e.g., "Purpose"), and so on.
- Highlighting: Bold and italic text is used to highlight important terms, requirements, or notes throughout the document.
- Priority: Every requirement statement in this SRS is assigned its own priority level based on its importance and relevance to the system's functionality.
- Traceability: All requirements are traced back to the specific business needs and objectives identified in the project's scope statement.
- Glossary: A glossary of key terms and definitions is included at the end of the document for clarity and consistency.
- Use Case Diagrams: Use case diagrams are used to visually represent the various use cases and scenarios that the system will need to support.

1.3 Intended Audience and Reading Suggestions

- Developers: They will be responsible for designing, developing, and implementing the system. They need to understand the technical requirements and constraints of the system.

- Project Managers: They will be responsible for overseeing the project and ensuring that it is completed on time and within budget. They need to understand the project scope, objectives, and deliverables.

- Testers: They will be responsible for testing the system to ensure that it meets the requirements and functions as intended. They need to understand the functional and non-functional requirements of the system.

- Documentation Writers: They will be responsible for creating user manuals, technical documentation, and other support materials for the system. They need to understand the system's features and functionality in order to effectively document them.

- End-users: They will be using the system to make restaurant bookings, view reservations, and cancel bookings. They need to understand how to use the system and what features are available to them.

The SRS is organized as follows:

- Introduction: Provides an overview of the system's purpose, scope, and goals.

- Functional Requirements: Describes the functional requirements of the system, including use cases, user interfaces, and system behaviors.

- Non-functional Requirements: Describes the non-functional requirements of the system, including performance, security, and usability requirements.

- System Constraints: Describes any limitations or constraints that must be considered during the development of the system.

- Assumptions and Dependencies: Describes any assumptions or dependencies that are necessary for the system to function as intended.

- Glossary: Provides a list of key terms and definitions used throughout the document.

1.4 Suggestions for reading.

Introduction: This provides an overview of the system's purpose, scope, and goals, and is useful for all readers.

Functional Requirements: This section describes the system's functional requirements and is useful for developers, project managers, testers, and documentation writers.

Non-functional Requirements: This section describes the system's non-functional requirements and is useful for developers, project managers, testers, and documentation writers.

System Constraints: This section describes any limitations or constraints that must be considered during the development of the system and is useful for developers and project managers.

Assumptions and Dependencies: This section describes any assumptions or dependencies that are necessary for the system to function as intended and is useful for developers and project managers.

Glossary: This provides a list of key terms and definitions used throughout the document and is useful for all readers.

1.5 Product Scope

Purpose: The system is designed to allow customers to make reservations at a restaurant online. It should provide an easy and convenient way for customers to make reservations, view availability, and manage their bookings.

Features: The system will include the following features:

- Online booking: Customers will be able to book a table at the restaurant through the system, selecting the date, time, and number of guests.

- Availability display: The system will display the availability of tables in real-time, allowing customers to see what times are available and make their booking accordingly.

- Reservation management: Customers will be able to view, modify, or cancel their reservations through the system.

- Notification: The system will notify customers via email or SMS about their reservation status, confirmation, or cancellation.

- User account: Customers will be able to create and manage their user accounts, including their personal information and booking history.

- Exclusions: The following are excluded from the product scope:

- Payment processing: The system will not process payments for reservations. Payment will be handled separately, such as in-person at the restaurant.

- Table management: The system will not manage the physical layout of tables or the seating arrangements in the restaurant.

- Assumptions: The following assumptions have been made:

- The restaurant has a set number of tables available for reservations and is able to handle the demand.

- Customers have access to the internet and can use the system to make reservations.

- Customers will arrive on time for their reservations, and any delays or cancellations will be handled by the restaurant staff.

- The product scope of the restaurant booking system is focused on providing a user-friendly and efficient way for customers to make reservations at the restaurant. It is not intended to handle payment processing or table management, which will be handled separately. The system assumes that the restaurant has a set number of tables available and that customers will arrive on time for their reservations.

a. Summary

The restaurant booking system is an online platform that streamlines the process of reservinga table at a restaurant. Customers can easily search for available tables, make reservations,modify, or cancel them, all through a convenient online interface. This simplifies the booking process and makes it easier for customers to plan their visit to the restaurant.

In addition, the system provides restaurant owners with a powerful tool to manage their reservations efficiently. By using the system, restaurant owners can track their restaurant's performance, optimize their table occupancy, and manage their resources more effectively.This can lead to better overall restaurant operations and increased revenue.

The system's goal is to enhance the customer experience by providing an easy-to-use, hassle-free booking process. This can help build customer loyalty and lead to repeat visits, which can ultimately benefit the restaurant's bottom line. At the same time, the system provides valuable insights to restaurant owners, which can help them optimize their businessand improve their profitability.

Overall, the restaurant booking system is a win-win solution for both customers and restaurant owners. Customers can easily secure a table and plan their visit, while restaurantowners can optimize their table occupancy and resources. By leveraging the power of technology, the restaurant booking system can help improve the overall dining experience foreveryone involved.

b. Company Overview

Our company is Pixelpals a group of four undergraduate students from the Open University of Sri Lanka who have developed a restaurant booking system as our mini project. The system is designed to simplify the process of reserving tables at restaurants, with the aim of enhancing customer experience and optimizing restaurant operations.

c. Project Overview

Our restaurant booking system is a web-based platform designed to simplify the process of reserving tables at restaurants. The system will provide customers with an easy-to-use interface to search for available tables, make reservations, modify, or cancel them. The system will also provide restaurant owners with a tool to manage their reservations efficientlyand track their restaurant's performance.

The project will be developed by a team of four undergraduate students from The Open

University. The team will consist of a project manager, a front-end developer, a back-end developer, and a quality assurance specialist.

Each team member will have specific responsibilities and will work collaboratively to deliver a high-quality product.

The project will be developed using modern web development technologies such as React, Node.js, and MongoDB. The front-end developer will be responsible for designing and implementing the user interface, while the back-end developer will be responsible for building the system's logic and database management. The quality assurance specialist willensure that the system is thoroughly tested and meets the highest standards of quality.

The project will be divided into several phases, including requirements gathering, design, development, testing, and deployment. The team will use an agile development methodologyto ensure that the project is completed on time and within budget.

The project's success will be measured by the system's usability, reliability, and the satisfaction of both customers and restaurant owners. The team will continuously collectfeedback from users and make improvements to the system as needed.

In conclusion, our restaurant booking system has the potential to revolutionize the restaurant industry by enhancing the customer experience, optimizing restaurant operations, and increasing revenue. We are excited about the project's potential and look forward to delivering a high-quality product that meets the needs of both customers and restaurant owners.

Project Background

The proposed restaurant reservation system is specifically aimed at addressing the issues faced by restaurants with poor booking systems. These restaurants struggle to manage their table reservations and often lose potential customers due to a lack of availability or long waiting times. The new system will provide a much-needed upgrade, enabling restaurants tomanage their bookings more efficiently and offer a seamless dining experience to their customers. The user-friendly interface and online management tools will help these restaurants attract and retain customers, improving their overall business performance. The system will be tailored to meet the unique needs of these struggling restaurants, offeringthem the tools they need to compete in the highly competitive restaurant industry.

d. Assumptions, General Development, and deployment

Assumptions

- Customers will have access to the internet and will use a web browser to access thebooking system.
- Customers will be able to navigate the user interface of the booking system andcomplete the booking process without assistance.
- The restaurant staff will have access to the booking system and will be able to manage reservations, view seating availability, and communicate with customers.
- The booking system will accurately reflect the availability of tables and reservations inreal-time.
- Customers will provide accurate and complete information when making areservation.
- The payment processing system will be reliable and secure, and customers will beable to complete transactions without issues.
- The booking system will be able to handle multiple bookings simultaneously withoutconflicts.
- Customers will be able to modify or cancel their reservations as needed, within theguidelines set by the restaurant.
- The booking system will be compatible with different devices, operating systems, andweb browsers.
- The booking system will comply with relevant regulations and data protection laws.
- Customer will be able to rate the restaurant on different criteria.
- If there are any special requirements likewise Birthday parties, Anniversary parties, Get together will be able to added by the Customer.
- System will be provided navigation facilities through Google maps.

- Customer will be able to wait in the waiting list and the system will be provided anotification to Customers who are in the waiting list.

General Development

A restaurant booking system is a software application that allows customers to reserve tables at a restaurant. The system typically includes a front-end interface that customers can use to view available tables, select a date and time for their reservation, and provide their contact information. It also includes a back-end interface for restaurant staff to managereservations, view seating availability, and communicate with customers.

Developing a restaurant booking system requires careful planning, design, and developmentto create a robust, user-friendly, and secure platform that meets the needs of both customers and restaurant staff.

01) Define requirements:

Define the system requirements based on the needs of the restaurant and customers.This includes deciding on the features, functionality, and user interface of the system.

02) Choose a technology stack:

Select the programming language, framework, and database that best suit thesystem's requirements.

a. HTML, CSS, JavaScript, PHP, MySQL, and MongoDB.

03) Design the user interface:

a. Design a user-friendly interface for customers to view available tables, select a date and time and make reservations.
b. The interface should be easy to navigate.
c. The interface should be responsive and visually appealing.

04) Develop the back - end:

Develop the server-side functionality that enables the system to manage reservations, track availability, and communicate with customers.

a. Creating a database to store reservation data.
b. Developing an application programming interface (API) to communicate with the front-end and implementing server-side validation and security measures.

05) Implement payment processing: Integrate a payment gateway to allow customers to pay for their reservations online.
    a. Integrating with a third-party payment processor (Pay by Visa, Master, PayPal, Crypto Currency)
    b. Implementing security measures to protect customer data.

06) Test the system: Test the system thoroughly to ensure that it meets all requirements, functions correctly, and is user-friendly.

07) Deploy the system: Deploy the system to a production environment and make it available to customers.

08) Provide ongoing support and maintenance: Provide ongoing support and maintenance to ensure that the system continues to function correctly and meets the evolving needs of the restaurant and customers.

Deployment

01) Choose a hosting provider:
Select a hosting provider that can meet the requirements of the system. Its includes considerations such as performance, scalability, security, and cost.
Common hosting technologies
    - Cloud-based platforms
      Ex: Amazon Web Services
      Google Cloud Platform
      Microsoft Azure

    - Traditional web hosting services provided by companies.

02) Configure the server environment:

Set up the server environment with the necessary software and configuration settings.

- Installing the web server
- Database server
- Any other dependencies required by the system.

03) Deploy the application code:
Upload the application code and any necessary files to the server using a file transfer protocol (FTP) or a version control system like Git.

04) Configure the database:
Configure the database server with the necessary schema and tables for the application to function.

- Setting up user accounts and permissions
- Any necessary database backups or replication.

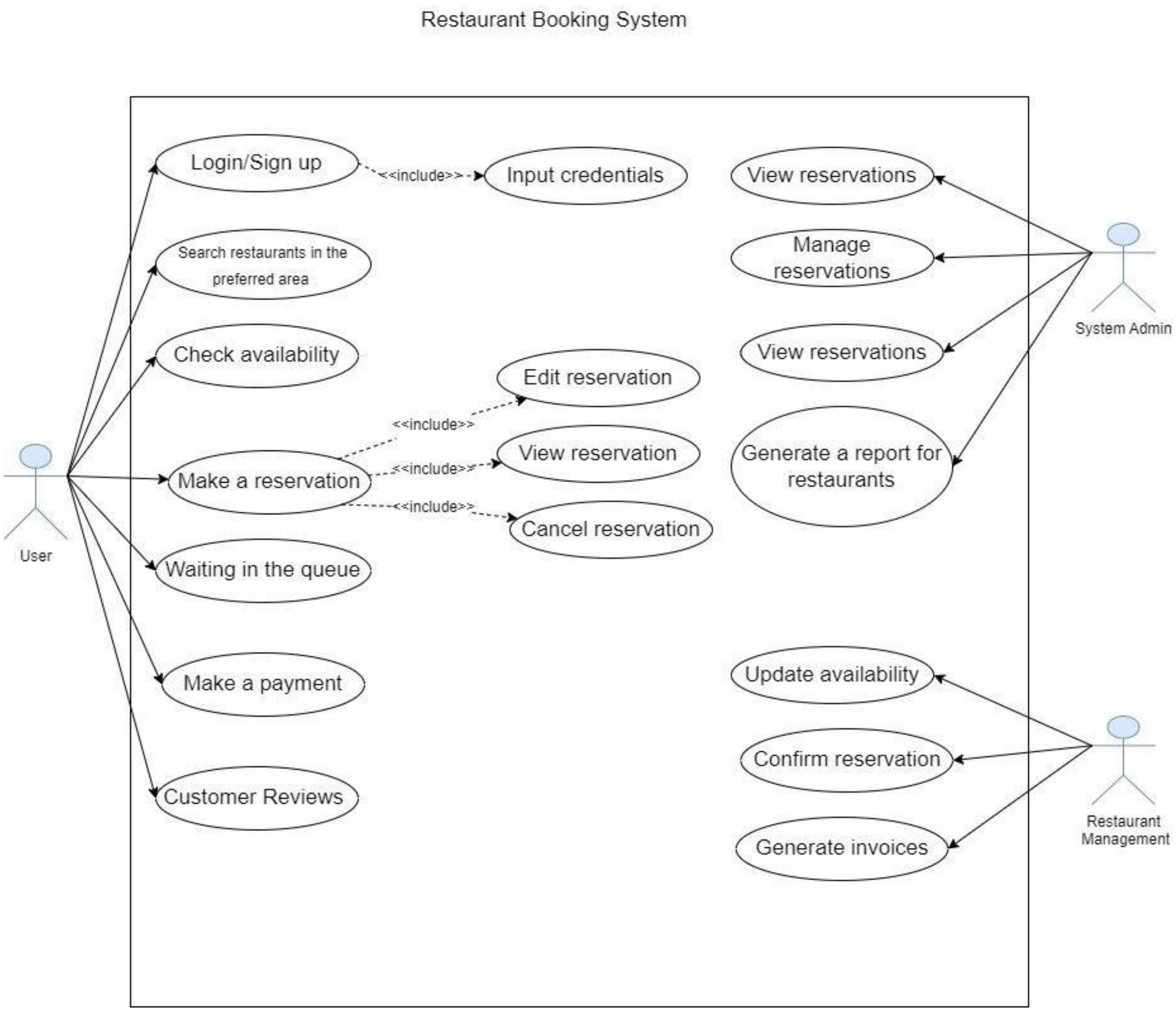
05) Test the system:
Test the system verify its functions correctly in the production environment. This will consider of testing the user interface, booking process, and payment processing.

06) Monitor the system:
Set up monitoring tools to track system performance, availability, and security. It includes monitoring server metrics like CPU usage, memory usage, and disk space, aswell as application-level metrics like response time, error rates, and traffic.

07) Provide ongoing maintenance and support:
Provide ongoing maintenance and support to ensure that the system remainsavailable, secure, and up to date.

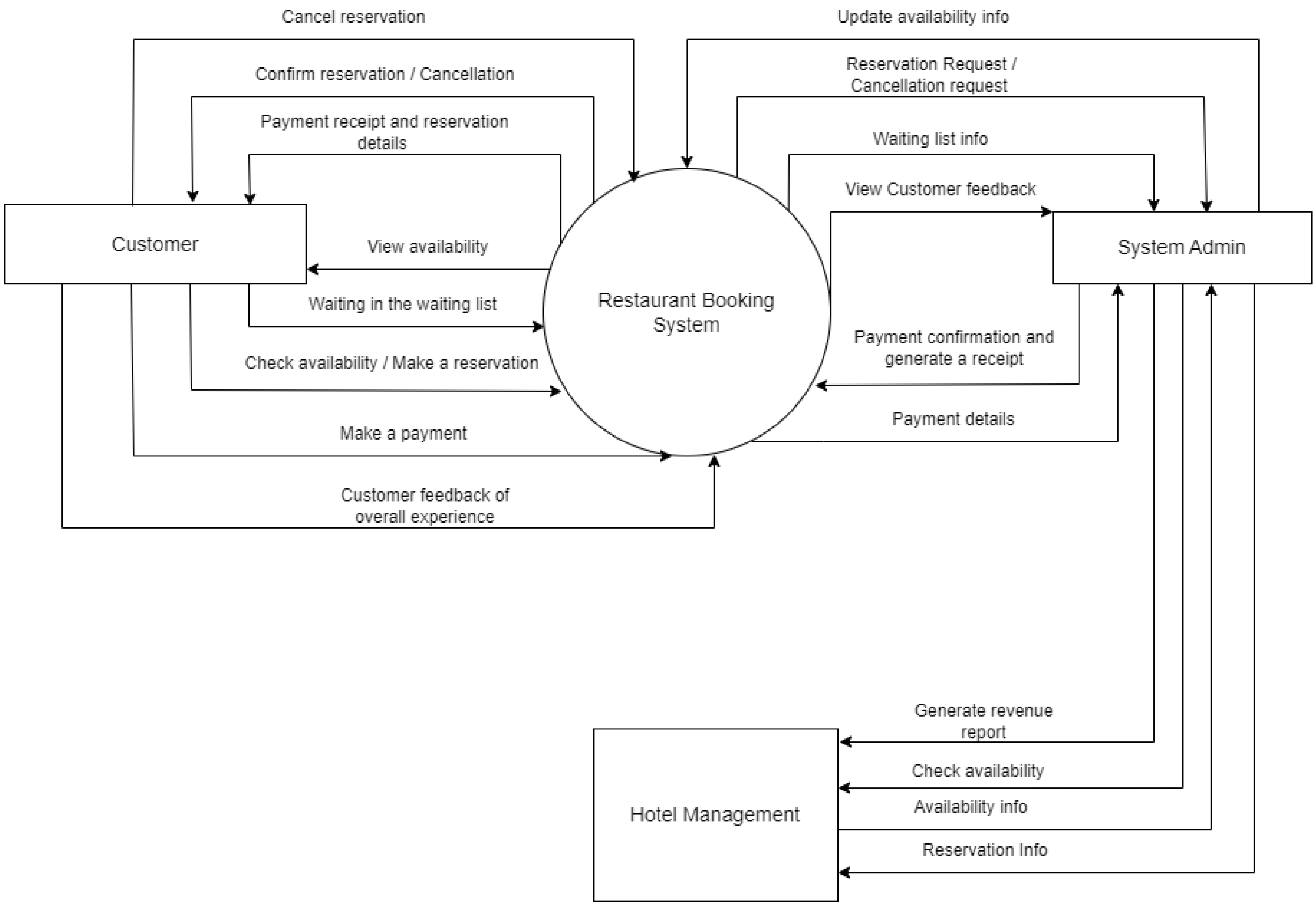
- Performing regular updates and patches
- monitoring for security vulnerabilities
- Addressing any issues that arise.

Structure of the system



Project Scope and Impact

Context Data Flow Diagram


Cancel reservation
Confirm reservation / Cancellation
Payment receipt and reservation details
Update availability info
Reservation Request / Cancellation request
Waiting list info
View Customer feedback
Customer
View availability
Waiting in the waiting list
Check availability / Make a reservation
Make a payment
Customer feedback of overall experience
Restaurant Booking System
System Admin
Payment confirmation and generate a receipt
Payment details
Generate revenue report
Check availability
Availability info
Reservation Info
Hotel Management

Scope Inclusions

In scope

- The system should enable customers to check table availability and reserve a table.
- The system should enable each online reservation to be connected to an account.
- The system should enable each user to have one account.
- The system should enable customers to join a waitlist when tables are unavailable.
- The system should enable customers to submit feedback on their dining experience.

- The system should enable customers to make secure online payments.
- The system should generate comprehensive reports on restaurant occupancy, revenue, and customer behavior.
- The system should enable customers to view and manage their reservations, including modification and cancellation.
- The system should be user-friendly.
- The system should handle high volumes of traffic and booking requests.

- The system should be secure and protect customer data.
- The system should be reliable and available 24/7.
- The system should be responsive and work seamlessly across multiple devices and platforms.
- The system should have fast loading times and minimal downtime.
- The system should comply with relevant industry standards and regulations.
- The system should incorporate backup and recovery mechanisms to prevent data loss.
- The system should manage table reservations of varying party sizes.
- The system should provide information on restaurant amenities, such as wheelchair accessibility and parking.

Page #	Page Name	Template	Components
			General:
			Site search
			Login
			Left navigation
			Top Menu
			Site Map
			Breadcrumb
			Master Page
			Navigation components
1	Home	Home Page	
			Enter Customer Details
2	Customer Login/Signup Menu		Related fields
3	Search Restaurants		Related fields

4	Restaurant List		Related details
5	Restaurant Details		Related details
6	Reservation Details		Related details
7	Profile		Related details
8	Dashboard		Related details
9	About Us	General Content	No New Components
10	Privacy Policy, Terms and Conditions	General Content	No New Components
11	Help	General Content	No New Components

- The system should be capable of handling table reservations for a variety of restaurant types, including fine dining, casual dining, and fast-food establishments.
- The system should support multiple languages to accommodate customers from various regions.
- The system should provide information on restaurant operating hours, location, and contact details.
- The system should allow restaurants to manage their staffing and shift schedules.
- The system should enable customers to filter restaurants based on dietary restrictions and preferences.
- The system should offer recommendations for types of cuisines based on a customer's search criteria and dining history.

Scope Exclusions

Out of scope

• Table management: The system will not include functionality to manage table assignments or table configurations.

• Menu management: The system will not include functionality for managing restaurant menus, such as adding or removing items, updating prices, or managing inventory.

• Order management: The system will not include functionality for placing orders, tracking orders, or managing payments for food and drinks.

• Delivery or takeout management: The system will not include functionality for managing delivery or takeout orders.

- Social media integration: The system will not include functionality for integrating with social media platforms, such as posting updates, reviews, or promotions on social media sites.

- Marketing and advertising: The system will not include functionality for managing marketing or advertising campaigns, such as email marketing, social media ads, or paid search campaigns.

- Employee management: The system will not include functionality for managing employee schedules, payroll, or HR functions.

- Food preparation or cooking: The system will not include functionality for managing food preparation or cooking, such as recipe management, ingredient tracking, or cooking instructions.

- POS integration: The system will not include functionality for integrating with point-of-sale (POS) systems, such as managing payments, discounts.

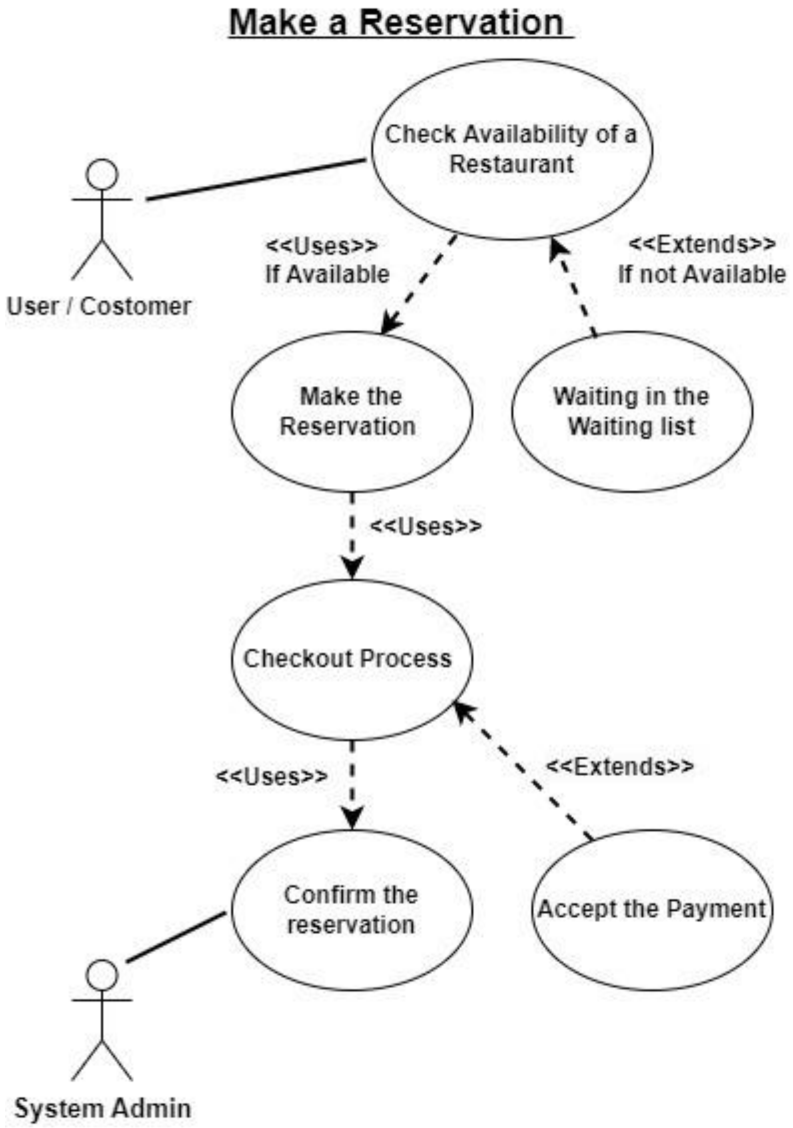
Functional Requirements

This section includes the functional requirements.

Use Case: Make a reservation.

Customer should make a reservation before visit a restaurant through the system.

Diagram:



Brief Description:

User will be able to check the availability of restaurants and make a reservation for available restaurants.

Step by step Description:

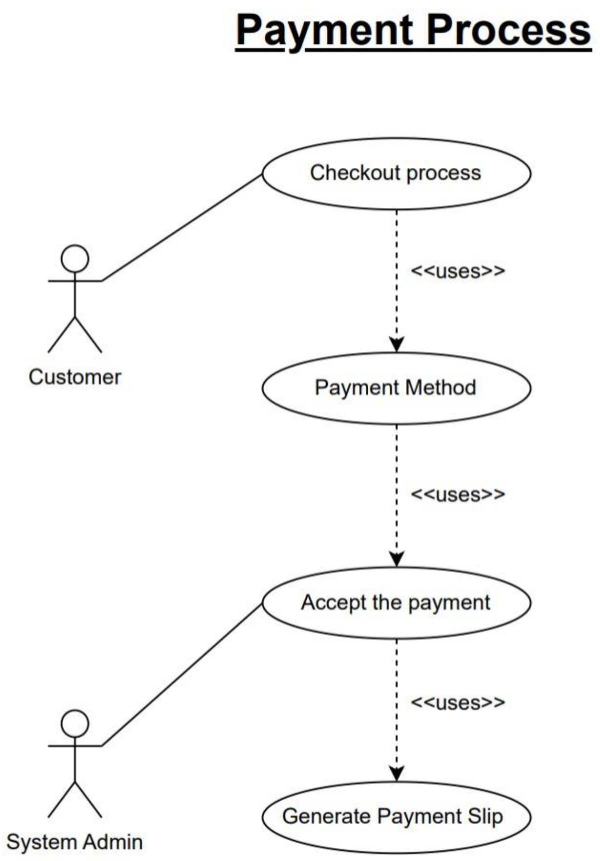
1) User searches for a restaurant and finds a convenient one for his requirements.
2) User makes the reservation for an available restaurant.
3) If there are no available restaurants user should wait in the waiting list.
4) After the reservation process user has to go to the checkout process and make the payment using any payment method given by the system.
5) Admin will accept the payment, send a payment receipt to Customer, and confirm the

reservation.

Use case: Payment Process

Customer should make the payment for reservation using a convenient method.

Diagram:



Brief Description:
Customer should make the payment for reservation using a convenient method.

Step by step Description:

1) User redirects to the checkout page before completing the reservation.
2) User will be able to choose most suitable payment method (Visa, Master, PayPal, Cryptocurrency etc.)

3) User Provides details of the Customer and continues the payment ahead.

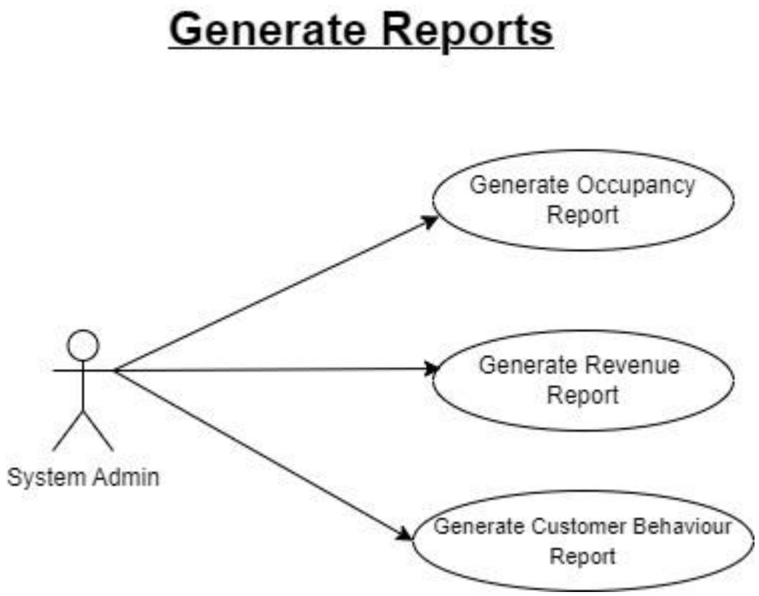
4) User confirms the payment.

5) System Admin generates a receipt for the customer and forwards it to the email address.

Use case:

System Admin generates reports based on the data in the system.

Diagram:



Brief Description:

System Admin generates Occupancy Report, Revenue Report and Customer Behavior System using information in the system.

Step by step Description:

1) System Admin Analyzes data in the System related to reservations, customers, and other data.
2) Generate revenue report using reservation data and share it with restaurant management.
3) Generate Occupancy Report
4) Generate customer behavior reports using user interaction and based on user’s information.

Non-Functional Requirements

Communication Interfaces:

The restaurant reservation system communicates with the restaurant's local server through a local area network. Each device requires a network interface card to enable communication. For the restaurant reservation system, we need to ensure that the system is compatible with the network interface card used by the restaurant's local server.

Security and Authentication Requirements:

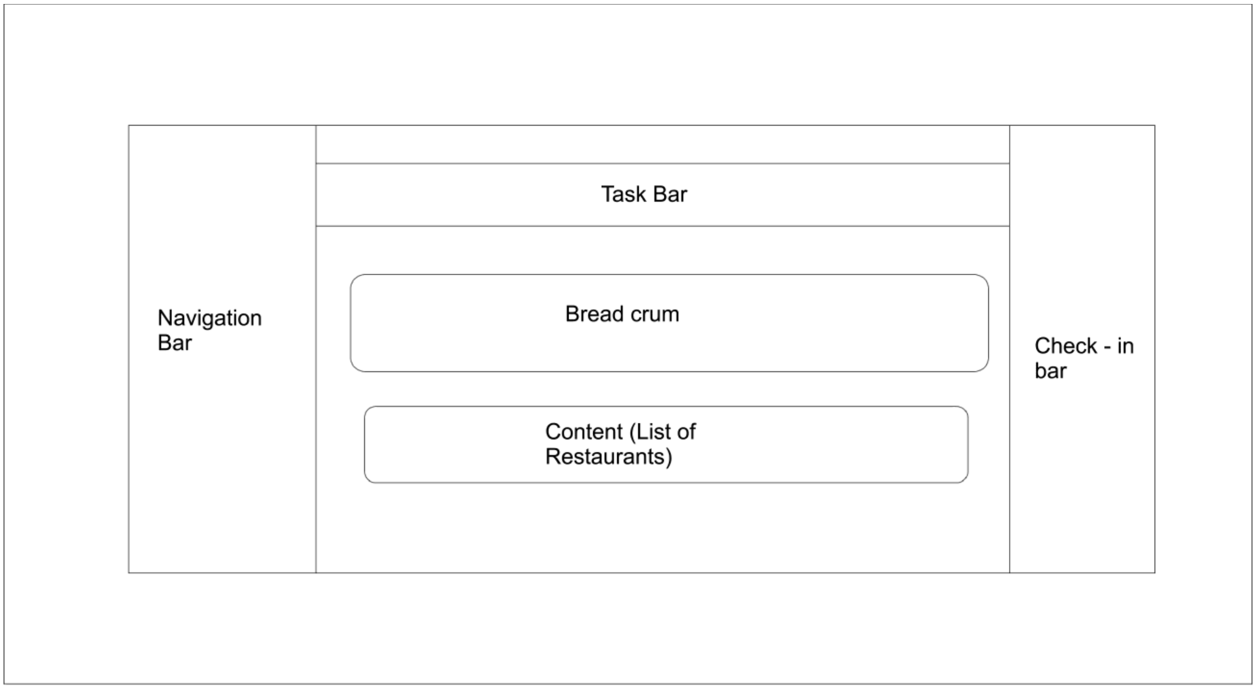
To ensure the security of the reservation system, all users must log in with their unique usernames and passwords before they can access the system. The system has different access levels for normal users, super users, and administrators. These usernames and passwords are assigned by the administrators to ensure that only authorized personnel can access the system.

Data Storage Security:

The restaurant reservation system uses a Microsoft SQL server to store its data. The server uses SQL server authentication to connect to the database, ensuring secure access. User passwords are also encrypted when stored in the database to ensure the security of the reservation system.

User Interfaces

Main Template: web-based layout



Quality Assurance Requirements

For our restaurant reservation system, we will leverage our experience and best practices in testing to align our QA services with the organizational and business requirements. Our QA team will follow these key phases during each testing cycle:

2Planning: We will plan the testing purpose and goals within the project scope and develop a business-oriented approach to testing.

Designing: Collaboratively, we will formulate effective and efficient test methods, processes, and techniques to carry out testing tasks in line with the devised strategy.

Execution: We will perform day-to-day testing operations while monitoring metrics to ensure optimal performance.

Reporting: Our team will measure and analyze results to identify areas for further improvement.

QA Scope

Types of Testing

1. Functionality testing: This testing type ensures that the restaurant reservation system is functioning as expected. It covers a range of test cases that verify the system's capabilities, such as booking a table, canceling a reservation, modifying a booking, etc.

2. Usability testing: This testing type checks how easy the system is to use. It covers aspects such as the system's interface, navigation, and user experience. Usability testing can help identify areas that may confuse or frustrate users and can help improve the system's usability.

3. Performance testing: This testing type evaluates the system's response time, stability, and scalability. It covers aspects such as the number of users that can access the system simultaneously, the system's ability to handle large volumes of data, and the response time for various operations.

4. Security testing: This testing type checks the system's ability to protect sensitive user information and prevent unauthorized access. It covers aspects such as data encryption, authentication, authorization, and access control.

5. Compatibility testing: This testing type checks the system's ability to work seamlessly with different browsers, operating systems, and devices. It covers aspects such as the system's compatibility with various browsers and devices, the system's ability to adapt to different screen sizes, and the system's ability to handle different network conditions.

6. Regression testing: This testing type checks the system's performance after changes or updates have been made to the system. It covers aspects such as the system's ability to maintain its functionality and usability after an update or change has been made.

7. Integration testing: This testing type checks the system's ability to work seamlessly with other systems or applications. It covers aspects such as the system's ability to integrate with payment gateways, email services, or social media platforms.

In the test execution phase of the restaurant reservation system, the QA team will use various testing types, including functionality, usability, performance, security, compatibility, integration, and regression testing. A smoke test will be performed to check the build's standard before proceeding with functional testing. The team will only accept builds that are 80% developed. Any defects found during testing will be logged and forwarded to the development team. Fixed code will be incorporated, and regression testing performed. Ultimately, the system will meet quality standards before public release.

Functional Requirements

- The following are the functional requirements.

  High Criticality:

- The system should enable customers to check table availability and reserve a table.
- The system should enable customers to make secure online payments.
- The system should generate comprehensive reports on restaurant occupancy, revenue, and customer behavior.
- The system should enable customers to view and manage their reservations, including modification and cancellation.

  Medium Criticality:

- The system should enable connect each online reservation to an account.
- The system should enable each user to have one account.
- The system should enable customers to join a waitlist when tables are unavailable.
- The system should enable customers to submit feedback on their dining experience.
- The system should manage table reservations of varying party sizes.
- The system should provide information on restaurant amenities, such as wheelchair accessibility and parking.
- The system should be capable of handling table reservations for a variety of restaurant types, including fine dining, casual dining, and fast-food establishments.
- The system should support multiple languages to accommodate customers from various regions.
- The system should provide information on restaurant operating hours, location, and contact details.

- Low Criticality:

- The system should allow restaurants to manage their staffing and shift schedules.
- The system should enable customers to filter restaurants based on dietary restrictions and preferences.
- The system should offer recommendations for types of cuisines based on a customer's search criteria and dining history.

Criticality Guidelines

HIGH: Testing critical business process flows, major non-functional requirements, security requirements, and external interface data validation.

MEDIUM: Testing important alternative business process flows and non-functional requirements that contribute to effective functioning but are not critical to the success of the business process.

LOW: Testing desirable but rarely used alternative flows or features that are not necessary for the basic functioning of the business process, such as non-essential UI requirements.

Performance Testing

To test the performance of the proposed system and identify areas for improvement, the following high-level tasks will be carried out:

- Identify load test scenarios
- Generate scripts for those scenarios
- Execute the scripts
- Analyze the results against provided benchmarks
- Provide a comprehensive report.

In Scope

PERFORMANCE TESTING

The performance testing will involve analyzing the base level performance of the application to ensure it meets the performance benchmarks and SLAs provided by the client. In case the client is unable to provide this information, we will propose the same based on our experience and consider industry performance boundaries and recommended response times by experts.

DEPLOYMENT

We will also provide a detailed deployment guide with step-by-step instructions to support the deployment stage of the proposed system.

ADMIN AND USER GUIDE

Additionally, an admin and user guide will be provided as a reference for system administrators and end-users, respectively.

Out of Scope

- Automation Testing: Automating tests involves the use of specialized tools and scripts to execute tests. While it can be beneficial for improving testing efficiency and accuracy, it is not a requirement for our restaurant reservation system. As such, we will not be dedicating resources towards automating tests for this project.

- User Acceptance Testing: User acceptance testing (UAT) is typically performed by end-users to ensure that the system meets their requirements and expectations. While our offshore QA team will be

involved in acceptance testing by identifying and executing test cases, the actual UAT phase will not be carried out as a separate phase. This means that end-users will not be directly involved in testing the system.

Development Requirements

Development Environments

Name: RestaurantReserveDev

Type: Cloud-based development environment.
The RestaurantReserveDev environment can be hosted on

- Cloud-based platforms: Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP).
- It can be set up to use development tools and frameworks such as Git for version control, Jenkins for continuous integration, and Docker for containerization.
- The environment will be configured to use programming languages such as Java or Python, web development frameworks such as Spring or Django, and database technologies such as MySQL or PostgreSQL.

Applicable Standards

- Legal and regulatory: The system may need to comply with legal and regulatory standards such as those set by the US Food and Drug Administration (FDA) and the Uniform Commercial Code (UCC).

- Communications standards: The system may need to comply with communication standards such as TCP/IP (Transmission Control Protocol/Internet Protocol) to ensure compatibility with the internet and other networks.

- Platform compliance standards: The system may need to comply with platform compliance standards such as Windows compatibility to ensure it runs smoothly on Windows operating systems.

- Quality and safety standards: The system may need to comply with quality and safety standards such as ISO (International Organization for Standardization) and CMM (Capability Maturity Model) to ensure the system meets quality and safety requirements and is developed and maintained efficiently.

Optional Functional Requirements

For the system, potential future requirements could include:
Integration with other systems: The system may need to be integrated with other systems such as a civil security network system and an island-wide criminal investigation system to share information and improve coordination.

Web-based system: To facilitate integration with other systems, the system may need to be converted into a web-based system. This would allow it to be accessed and used from any device with an internet connection and a web browser, making it more flexible and easier to integrate with other web-based systems.

It's important to note that these requirements are optional and may or may not be necessary for the specific restaurant reservation system being developed. However, considering potential future requirements can help ensure that the system is developed with flexibility and scalability in mind, and can be easily adapted to meet changing needs in the future.

END