Software Requirements Specification

for

Smart Restaurant System

Version 1.0

Prepared by

Group:

Anushka Pandit	60004210119
Varenya Uchil	60004210121
Nirzari Parikh	60004210156

Instructor: Prof. Meera Narvekar

Course: Computer Engineering

Lab Section: C2-2

Date: 25/03/2024

Contents

1 Introduction

1.1 Document Purpose

The Software Requirements Specification (SRS) document for a smart restaurant serves the following purposes:

- Requirements Clarification: The SRS document provides a clear and detailed description of the functionalities and features expected from the smart restaurant software. It helps to clarify the scope of the project, ensuring that all stakeholders have a shared understanding of what needs to be developed.
- Basis for Development: The SRS document serves as the foundation for the development process. It outlines the specific requirements that developers need to implement in the software system, guiding them in the design, coding, and testing phases.
- Communication Tool: The SRS document acts as a communication tool between different stakeholders involved in the project, including clients, developers, project managers, and quality assurance teams. It ensures that everyone is aligned with the project goals and requirements.
- Risk Management: By clearly defining the requirements and expectations upfront, the SRS document helps to identify potential risks and challenges early in the project lifecycle. This allows stakeholders to address issues proactively and minimize the likelihood of project delays or failures.
- Basis for Evaluation: The SRS document provides a benchmark for evaluating the completed software system. Once the development is complete, stakeholders can refer back to the SRS document to ensure that all requirements have been met and that the software functions as intended.
- Legal and Regulatory Compliance: In industries like food service, there may be legal and regulatory requirements that the smart restaurant software must comply with, such as data privacy laws or food safety regulations. The SRS document helps to ensure that these requirements are documented and addressed during the development process.

Overall, the SRS document plays a crucial role in the successful development and implementation of a smart restaurant software system by providing a clear roadmap of requirements and expectations for all stakeholders involved.

1.2 Product Scope

- **Functional Scope**:
  - Table management: Reservation system, table allocation, and tracking.
  - Menu management: Digital menu display, item categorization, and customization.
  - Order management: Online ordering, tableside ordering via tablets, order tracking, and status updates.
  - Payment processing: Integration with payment gateways for seamless transactions.
  - Customer management: Loyalty programs, customer feedback collection, and customer relationship management.
  - Inventory management: Stock tracking, ingredient sourcing, and automated reordering.
  - Reporting and analytics: Sales reports, inventory analysis, and customer behaviour insights.
- **Non-Functional Scope:**
  - Performance: Response time, throughput, and system availability.
  - Security: User authentication, data encryption, and access control.
  - Scalability: Ability to handle increased load during peak hours or expansion of the restaurant.
  - Compatibility: Support for various devices (desktop, tablets, smartphones) and operating systems.

1.3 Intended Audience and Document Overview

Intended Audience:

- Restaurant Owners/Managers: They would utilize smart restaurant products to streamline operations, enhance customer experiences, and optimize resource management to improve profitability.
- Customers: Diners benefit from smart restaurant products by experiencing seamless ordering and payment processes, personalized recommendations, and enhanced dining experiences through interactive menus and ambiance control.
- Technology Providers: Companies and technicians developing and providing smart restaurant solutions would be users of the SRS document, offering their products and services to restaurants to meet their technological needs.
- Food Suppliers: Suppliers could leverage data from smart restaurant systems to anticipate demand, manage inventory, and ensure timely deliveries to restaurants.

The SRS document provides a clear and detailed description of the functionalities and features expected from the smart restaurant software. It helps to clarify the scope of the project, ensuring that all stakeholders have a shared understanding of what needs to be developed. It serves as the foundation for the development process and outlines the specific requirements that developers need to implement in the software system, guiding them in the design, coding, and testing phases.

1.4 Acronyms and Abbreviations

1) SRS: Software Requirement Specification
2) FP: Functional Point
3) SRMS: Smart Restaurant Management System
4) CRM: customer relationship management
5) POS: Point-of-scale
6) SMTP: Simple Mail Transfer Protocol
7) RESTful APIs: Representational State Transfer APIs
8) HTTP: Hyper Text Transfer Protocol
9) MQTT: Message Queuing Telemetry Transport
10) SMTP: Simple Mail Transfer Protocol

1.5 Document Conventions

Heading Font size – Arial 14
Normal Font size – Arial 11
Text Alignment – Justified
Space – Single spaced
Margin – 1”

2 Overall Description

2.1 Product Perspective

The Smart Restaurant Management System is a comprehensive solution designed to streamline and optimize restaurant operations. With its intuitive interface and robust features, the Smart Restaurant Management System aims to revolutionize the way restaurants manage their daily tasks and serve their customers. This system caters to a wide range of users, including restaurant managers, staff members, and customers, providing each with tailored functionalities to enhance their experience. From inventory management and order processing to table reservations and customer feedback, the SRMS offers a holistic approach to restaurant management. Built on a modern architecture, the system leverages cutting-edge technologies such as cloud computing, mobile applications, and data analytics to deliver real-time insights and actionable information. Compliance with regulatory standards and data security measures is paramount, ensuring that the Smart Restaurant Management System meets industry requirements and safeguards sensitive information. While the initial release of the Smart Restaurant Management System focuses on core functionalities, future enhancements are planned to further enrich the system and adapt to evolving needs and trends in the restaurant industry. Overall, the Smart Restaurant Management System represents a pivotal tool for restaurants seeking to optimize efficiency, improve customer satisfaction, and stay ahead in today's competitive market landscape.

2.2 Product Functionality

The Smart Restaurant Management System (SRMS) offers a wide array of essential functionalities to streamline restaurant operations and enhance customer experiences.

- Firstly, it facilitates efficient table management, allowing staff to allocate, reserve, and track tables in real-time, optimizing seating arrangements and reducing wait times.
- Secondly, the SRMS automates the order processing workflow, from order placement by customers to fulfillment by kitchen staff, ensuring accuracy and timeliness.
- The system also incorporates robust reporting and analytics features, providing managers with valuable insights into sales trends, customer preferences, and operational performance to inform strategic decision-making.
- Moreover, the SRMS supports seamless integration with payment gateways, enabling secure and convenient payment processing for customers through various payment methods.
- Furthermore, it offers customer relationship management (CRM) tools to capture and manage customer information, preferences, and feedback, fostering personalized interactions and loyalty programs.
- The SRMS includes features for menu management, enabling dynamic menu updates, customization, and promotion management to cater to changing customer preferences and seasonal offerings.

Overall, the SRMS delivers a comprehensive suite of functionalities to empower restaurants to operate efficiently, deliver exceptional service, and thrive in today's competitive hospitality industry landscape.

2.3 Users and Characteristics

- The Smart Restaurant Management System (SRMS) caters to a diverse range of users, each with specific roles and characteristics tailored to their needs within the restaurant environment.
- Restaurant managers constitute a primary user group, requiring access to comprehensive administrative tools and reporting functionalities to oversee operations, analyze performance metrics, and make informed strategic decisions.
- Frontline staff members, including waitstaff, chefs, and kitchen personnel, rely on the SRMS for efficient task management, order processing, and communication to streamline their workflow and deliver prompt service to customers.
- Customers themselves are users of the system, interacting with user-friendly interfaces to place orders, make reservations, and provide feedback, requiring intuitive design and seamless functionality for a satisfying experience.
- Suppliers play a crucial role as users, accessing the SRMS to manage inventory orders, deliveries, and invoices, necessitating clear communication channels and integration capabilities.
- System administrators and IT personnel are responsible for maintaining and configuring the SRMS, requiring robust security features, system scalability, and reliable technical support.
- The system must accommodate users with varying levels of technological proficiency, offering intuitive interfaces, clear instructions, and training materials to ensure accessibility and adoption.
- Managers may require remote access to the system via mobile applications or web portals, necessitating cross-platform compatibility and responsive design.
- The system should support multiple user roles and permissions, enabling granular access control to safeguard sensitive data and restrict unauthorized actions.
- The SRMS caters to a diverse user base with distinct characteristics and requirements, aiming to provide tailored functionalities and a seamless experience for all stakeholders involved in restaurant operations

2.4 Operating Environment

The operating environment for a smart restaurant management system encompasses a combination of hardware and software elements tailored to streamline various managerial tasks. This includes point-of-sale terminals for order processing, kitchen display systems to coordinate food preparation, and backend servers or cloud infrastructure to host the software application and manage data. Essential software components comprise the application itself, a database management system, security measures, and integration tools for seamless interaction with external systems. Peripheral devices such as printers, barcode scanners, and card readers support operational efficiency. Supporting infrastructure, including reliable power sources and backup systems, ensures uninterrupted functionality. Additionally, staff training, user-friendly interfaces, and ongoing technical support contribute to a smooth user experience. Overall, the operating environment must be robust, secure, and scalable to meet the evolving needs of the restaurant.

2.5 Design and Implementation Constraints

The design and implementation of the Smart Restaurant Management System (SRMS) are subject to various constraints that influence its development and deployment. Firstly, compatibility with existing hardware and software infrastructure within restaurants may pose constraints, requiring careful consideration of system requirements and integration capabilities to ensure seamless operation. Secondly, budgetary constraints may limit the allocation of resources for system development and implementation, necessitating efficient use of available funds and prioritization of essential features. Additionally, time constraints may impact the project timeline, requiring adherence to strict deadlines and efficient project management practices to deliver the SRMS within specified timeframes. Furthermore, regulatory compliance requirements, such as data privacy laws and food safety regulations, impose constraints on system design and functionality, necessitating thorough compliance checks and documentation throughout the development process. Moreover, scalability constraints must be addressed to accommodate future growth and increased user demand, requiring flexible architecture and modular design principles. Additionally, usability constraints may arise due to the diverse user base of the SRMS, requiring intuitive interfaces, comprehensive user training, and ongoing usability testing to ensure user satisfaction. Furthermore, security constraints are paramount to protect sensitive data and prevent unauthorized access or breaches, necessitating robust encryption measures, access controls, and regular security audits. Lastly, resource constraints, including limited manpower and technical expertise, may impact system development and maintenance efforts, requiring efficient resource allocation and strategic outsourcing of tasks when necessary. Overall, the SRMS must navigate various design and implementation constraints to deliver a robust, secure, and user-friendly solution that meets the needs of restaurant stakeholders while adhering to industry standards and regulations.

2.6 User Documentation

User documentation for the Smart Restaurant Management System (SRMS) comprises user manuals, FAQs, and online tutorials. User manuals offer step-by-step guidance on system navigation and common tasks. FAQs address common queries and troubleshooting. Online tutorials on the website provide additional support and training. These resources ensure users can effectively utilize the SRMS, fostering efficient restaurant operations and enhancing user satisfaction.

2.7 Assumptions and Dependencies

- It's assumed that restaurants using the SRMS have reliable internet connectivity to access the system and perform online transactions seamlessly.
- The system assumes that restaurant staff are adequately trained to use the SRMS effectively, necessitating comprehensive training programs to ensure proficiency.
- The SRMS depends on third-party APIs for functionalities such as payment processing and integration with external systems, requiring stable API connections and adherence to service level agreements.
- The system assumes compliance with relevant regulations and standards governing data privacy, food safety, and payment processing to ensure legal and ethical operation.
- Dependencies on hardware infrastructure, such as POS terminals and mobile devices, require compatibility and reliability for smooth system operation. Overall, these assumptions and dependencies shape the operational environment of the SRMS, influencing its design, functionality, and reliability in supporting restaurant operations.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- Customer-Facing Interfaces:
  - Mobile App: The mobile app should have a user-friendly interface allowing customers to browse menus, place orders, make reservations, and access loyalty programs. It should feature visually appealing menu layouts, high-quality images of dishes, and easy navigation.
  - Self-Service Kiosks: Kiosks should have a touchscreen interface with clear menu categories, item descriptions, and customization options. The interface should guide customers through the ordering process step-by-step and offer multiple payment methods.
  - Digital Menu Boards: These boards should display menu items, prices, and promotions in an attractive and easy-to-read format. They may include images or videos showcasing featured dishes and special offers.
  - Online Ordering Platform: The online ordering interface should be accessible through a web browser and feature a simple layout with search functionality, filters, and a seamless checkout process.
- Staff-Facing Interfaces:
  - POS System: The POS interface should have a clean layout with buttons for common functions such as adding items to orders, applying discounts, and processing payments. It should also provide easy access to order history, table status, and customer information.
  - Inventory Management System: The inventory interface should provide a comprehensive view of current stock levels, ingredient details, and supplier information. It should include features for updating inventory counts, setting reorder thresholds, and generating reports.
- General Interface Standards:
  - Consistent Design: All interfaces should follow a consistent design language with uniform color schemes, typography, and navigation patterns.
  - Standard Buttons and Functions: Common functions such as "Order", "Cancel", "Add to Cart", "Checkout", and "Confirm" should be clearly labeled and accessible across all interfaces.
  - Error Message Display: Error messages should be displayed prominently and provide clear instructions for resolving issues, such as missing information or payment errors.
  - Accessibility: Interfaces should comply with accessibility standards to ensure usability for all users, including those with disabilities.

3.1.2 Hardware Interfaces

- Point-of-Sale (POS) System:
  - Supported Devices: The software should interface with POS terminals, which may include touchscreen terminals, or mobile devices.
  - Data Interactions: The software communicates with the POS system to process orders, payments, and manage inventory. It sends transaction data such as order details, customer information, and payment amounts to the POS for processing.
- Mobile Devices (Tablets, Smartphones):
  - Supported Devices: The software may run on various mobile devices used by customers, or management for tasks such as order placement, table management, and inventory tracking.

  - Data Interactions: The software communicates with mobile devices to send and receive orders, update menu items and pricing, and provide real-time notifications and alerts.
- Self-Service Kiosks:
  - Supported Devices: The software interfaces with self-service kiosks installed in the restaurant for customer use.
  - Data Interactions: The software controls the user interface displayed on the kiosk screen, guiding customers through the ordering process and transmitting order details to the POS system for processing.
- IoT Devices (Sensors, Controllers):
  - Supported Devices: The software may interface with various IoT devices installed in the restaurant, such as temperature sensors, motion detectors, and smart appliances.
  - Data Interactions: The software receives data from IoT devices to monitor environmental conditions, track inventory levels, and automate processes such as HVAC control and equipment maintenance.
- Payment Processing Devices:
  - Supported Devices: The software interfaces with payment terminals, card readers, and mobile payment devices used for processing customer payments.
  - Data Interactions: The software sends payment authorization requests to payment devices and receives transaction confirmations, updating order status and closing out transactions.

3.1.3 Software Interfaces

- Operating System Interface:
  - Operating System: The smart restaurant system may be designed to run on various operating systems such as Windows, Linux, or macOS.
  - Connection Purpose: The software interfaces with the operating system to manage system resources, access hardware devices, and execute application processes.
  - Data Items or Messages: Data items or messages exchanged with the operating system may include system calls, file I/O operations, and process management commands.
  - Services Needed: The software may require services provided by the operating system such as memory management, process scheduling, and device drivers.
  - Communication Nature: Communication with the operating system occurs through system calls and API functions provided by the operating system's kernel.
- Shared Data: Data shared across software components may include configuration settings, log files, and temporary storage files managed by the operating system.
- Implementation Constraint:

The smart restaurant system must be designed to be platform-independent or compatible with the targeted operating system(s) to ensure portability and interoperability across different environments.

3.1.4 Communications Interfaces

- HTTP/HTTPS: The Hypertext Transfer Protocol (HTTP) or its secure counterpart (HTTPS) is commonly used for communication between web-based interfaces such as online ordering platforms, reservation systems, and mobile apps. HTTP(S) enables the exchange of data between clients and servers, supporting features such as menu browsing, order placement, and payment processing.
- RESTful APIs: Representational State Transfer (REST) APIs provide a standardized approach for communication between software components over the internet. RESTful APIs are used for integrating different services within the smart restaurant system, allowing seamless interaction between modules such as the POS system, inventory management, and customer loyalty programs.

- MQTT (Message Queuing Telemetry Transport): MQTT is a lightweight messaging protocol ideal for IoT applications, commonly used for communication between IoT devices and backend servers. In smart restaurants, MQTT may be utilized for connecting IoT sensors and controllers to the central management system, enabling real-time monitoring of environmental conditions, equipment status, and inventory levels.
- SMTP (Simple Mail Transfer Protocol): SMTP is used for sending and receiving emails, which may be utilized for communication between the restaurant management system and staff members, suppliers, or customers. Email notifications may include order confirmations, reservation reminders, and promotional offers.
- Encryption and Data Security: To ensure the security of sensitive data such as customer information, payment details, and business transactions, communication interfaces may implement encryption standards such as SSL/TLS for securing data transmitted over HTTP(S) connections. Additionally, secure authentication mechanisms such as OAuth may be used to verify the identity of users and devices accessing the system.

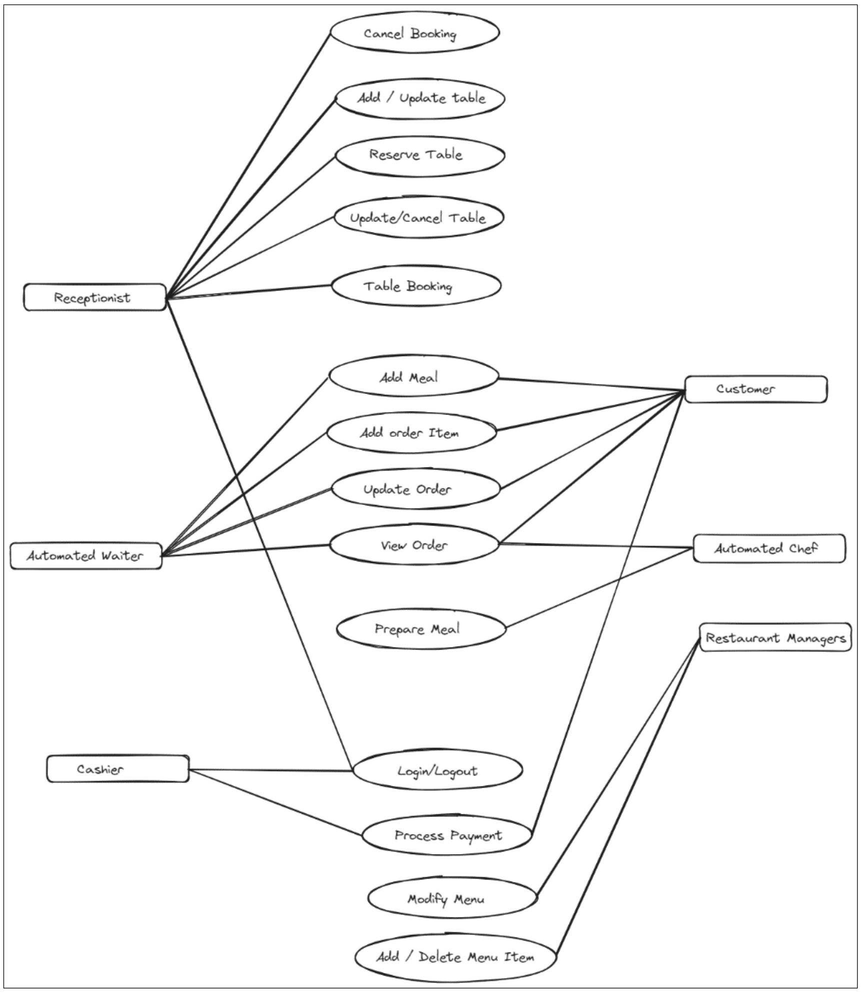
3.2 Functional Requirements

- Order Management
  - Online Ordering: Allow customers to place orders through a mobile app or website.
  - In-House Ordering: Enable customers to place orders via self-service kiosks or with waitstaff using a POS system.
  - Order Customization: Provide options for customers to customize their orders, such as selecting toppings, specifying cooking preferences, or adding special instructions.
  - Order Tracking: Allow customers to track the status of their orders in real-time, from preparation to delivery or pickup.
- Reservation Management
  - Online Reservations: Allow customers to make table reservations through the restaurant's website or mobile app.
  - Table Management: Provide staff with tools to manage table availability, assign reservations, and optimize seating arrangements.
  - Reservation Reminders: Send automated reminders to customers via email or SMS to confirm their reservations and reduce no-shows.
- Menu Management
  - Menu Display: Display the restaurant menu on digital menu boards, online ordering platforms, and self-service kiosks.
  - Menu Updates: Allow restaurant staff to easily update menu items, descriptions, prices, and availability in real-time.
  - Menu Personalization: Provide recommendations or customized menu views based on customer preferences, dietary restrictions, or past orders.
- Payment Processing
  - Secure Payment Handling: Ensure secure handling of payment information through encryption and compliance with PCI DSS standards.
  - Multiple Payment Options: Support various payment methods, including credit/debit cards, mobile wallets, and cash.
  - Split Payments: Allow customers to split bills, either by item or by individual, and process payments accordingly.
- Inventory Management

  - Real-time Inventory Tracking: Monitor inventory levels of ingredients and supplies in real-time to prevent stockouts and optimize purchasing decisions.
  - Automated Reordering: Set up automated alerts and reorder triggers to replenish inventory when levels fall below specified thresholds.
  - Ingredient Management: Track expiration dates, batch numbers, and supplier information for ingredients to ensure freshness and quality control.
- Customer Engagement
  - Loyalty Programs: Implement loyalty programs to reward repeat customers with points, discounts, or special offers.
  - Feedback Collection: Gather customer feedback through surveys, ratings, and reviews to identify areas for improvement and maintain customer satisfaction.
  - Promotional Campaigns: Send targeted promotions and marketing campaigns via email, SMS, or push notifications to attract new customers and drive repeat business.
- Reporting and Analytics
  - Sales Reporting: Generate daily, weekly, and monthly sales reports to track revenue, analyze trends, and make data-driven decisions.
  - Performance Metrics: Monitor key performance indicators (KPIs) such as average order value, table turnover rate, and customer satisfaction scores.
  - Business Insights: Provide actionable insights and recommendations based on data analysis to optimize operations and increase profitability.
- Staff Management
  - Shift Scheduling: Create and manage employee schedules, including shift assignments, time-off requests, and availability.
  - Task Management: Assign tasks and responsibilities to staff members, track task completion, and ensure efficient workflow management.
  - Training and Onboarding: Provide training materials and onboarding resources for new hires to ensure consistent service standards and operational excellence.

3.3 Behaviour Requirements

3.3.1 Use Case View



- Actors:
  - Customer: Interacts with the system to place orders, make reservations, provide feedback, and receive notifications.
  - Staff: Employees of the restaurant who interact with the system to manage reservations, process orders, update menus, manage inventory, handle payments, and receive notifications.
  - External Delivery: Represents third-party delivery services that deliver orders to customers.
  - Kitchen Display System: Displays incoming orders to the kitchen staff for preparation.
  - Delivery Personnel: Represents the delivery staff who pick up orders from the restaurant and deliver them to customers.
  - Chef: Prepares food orders in the kitchen based on the received orders.
- Use Cases:
  - Order Food: Customer places an order for food.
  - Make Reservation: Customer makes a reservation for a table.
  - Provide Feedback: Customer provides feedback on their dining experience.
  - Manage Reservations: Staff manages table reservations.
  - Process Orders: Staff processes customer orders for food.
  - Update Menu: Staff updates the restaurant's menu items and prices.

- Manage Inventory: Staff manages the inventory of ingredients and supplies.
- Handle Payments: Staff handles customer payments for orders.
- External Delivery: Third-party delivery service delivers orders to customers.
- Kitchen Display System: Displays incoming orders for kitchen staff.
- Delivery Personnel: Delivery staff pick up and deliver orders to customers.
- Chef: Prepares food orders in the kitchen.

4 Other Non-functional Requirements

4.1 Performance Requirements

1. Order Management System: A system for placing and managing orders, including features for both customers and staff to input orders, modify them, and track their status.
2. Table Management: Functionality to manage table availability, reservations, and seating arrangements, allowing staff to efficiently assign tables and optimize seating capacity.
3. Menu Management: Capability to easily update and customize menus, including item descriptions, prices, and availability, with options for seasonal variations and special promotions.
4. Inventory Management: Tools for tracking and managing inventory levels in real-time, automating reordering processes, and generating alerts for low stock items to ensure seamless operation.
5. Payment Processing: Integration with various payment methods, such as credit/debit cards, mobile payments, and possibly cryptocurrencies, ensuring secure and convenient transactions for customers.
6. Customer Relationship Management (CRM): Features for collecting and analyzing customer data, managing loyalty programs, and facilitating personalized experiences through targeted marketing and communication.
7. Staff Management: Tools for scheduling shifts, managing employee roles and permissions, tracking performance metrics, and facilitating communication among team members.
8. Integration and Scalability: Flexibility to integrate with other systems such as accounting software, POS systems, and online delivery platforms, as well as scalability to accommodate growth and expansion of the restaurant business.

4.2 Safety and Security Requirements

1. Data Security: Implement robust measures to protect sensitive data, including customer information, payment details, and business operations data. This involves encryption protocols, secure authentication methods, and regular security audits to identify and address vulnerabilities.
2. Backup and Recovery: Implement automated backup procedures to regularly backup critical data, such as transaction records, customer profiles, and inventory information. Additionally, develop a comprehensive disaster recovery plan to swiftly restore system functionality in the event of a data loss or system failure.
3. Compliance with Regulations: Ensure that the smart restaurant system complies with relevant data protection regulations, such as GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act), depending on the nature of the data collected and processed. This involves implementing

necessary safeguards, obtaining consent for data collection and processing, and providing mechanisms for data subjects to exercise their rights.

4. Physical Security: Secure physical access to servers, terminals, and other hardware components of the system to prevent unauthorized tampering or theft. This may include installing security cameras, employing access control measures, and implementing procedures for monitoring and reporting suspicious activities.

4.3 Software Quality Attributes

The given given quality attributes are essential for user and manager:

Usability: Ensuring that the interface is intuitive and easy to navigate, with features like touchscreen capabilities and straightforward menu structures. Quantitative measures could include average time taken for staff to complete tasks and customer feedback on user experience.

Reliability: The system would operate consistently without failures or errors. A quantifiable metric could be the system uptime percentage, indicating how often the system is available for use without disruptions.

Availability: This refers to the system's readiness for operation when needed. A verifiable measure could be the mean time between failures (MTBF) or the percentage of scheduled operational hours during which the system is functioning correctly.

Maintainability: The ease with which the system can be maintained and updated. Quantitatively, this could be assessed by measuring the average time required to implement software updates or the frequency of system maintenance tasks.

Flexibility: The system's ability to adapt to changing requirements or environments. A quantitative measure might involve tracking the time and effort needed to incorporate new features or modify existing ones.

Interoperability: It is the extent to which the system can interact with other hardware or software components. This could be evaluated by measuring the number of integrations with external systems or the percentage of successful data exchanges.

Robustness: The system's ability to handle unexpected inputs or errors gracefully. Quantitatively, this could be assessed by measuring the system's response time to errors or the percentage of transactions successfully completed despite errors.

Testability: The ease with which the system can be tested to ensure its functionality. This could be measured by the percentage of code coverage achieved by automated tests or the time required to execute a comprehensive test suite.

Portability: The ease with which the system can be transferred or adapted to different hardware or software environments. Quantitatively, this could be evaluated by measuring the time and effort required to deploy the system on different platforms.

Correctness: The accuracy and precision of the system's outputs. This could be assessed by comparing the system's outputs to expected results in a controlled testing environment and measuring the percentage of correct outputs.

Appendix A – Data Dictionary

Item	Description	Operations/Requirements
Constants	- MAX_TABLE_CAPACITY: Maximum capacity of a table	- Ensure party size does not exceed MAX_TABLE_CAPACITY when making a reservation
State Variables	- Table Status: Current status of each table (Empty, Reserved, Occupied)	- Set status when table is booked, reserved, or occupied
Inputs	- Reservation Request: Details of a reservation request (Table ID, Time, Party Size)	- Validate table availability and capacity
Outputs	- Reservation Confirmation: Confirmation of a successful reservation (Table ID, Time)	- Notify user of successful reservation
	- Table Assignment: Assigning a table to a reservation (Table ID, Reservation ID)	- Update table status to 'Reserved' or 'Occupied'
	- Order Request: Details of an order request (Table ID, Items, Quantity)	- Add items to the table's order list, update table status to 'Occupied' if not already
	- Payment Request: Request for payment of the bill (Table ID, Total Amount)	- Calculate total bill amount, notify user of the amount due
State Transitions	- Table Status Changes: Changing the status of a table based on actions (Booking, Order)	- Update table status from 'Empty' to 'Reserved' or 'Occupied' based on reservation or order
Constraints	- Table Availability: Tables should only be reserved or occupied when available	- Ensure table is not already reserved or occupied before assigning it to a reservation or order
	- Party Size Limit: Party size should not exceed the maximum capacity of a table	- Validate party size against MAX_TABLE_CAPACITY before accepting a reservation request
Data Requirements	- Table Information: Information about each table (ID, Capacity, Location)	- Maintain a database of tables with their capacities and locations to manage reservations and seating
	- Menu Items: List of items available on the menu (Name, Description, Price)	- Maintain a menu with details of each item including name, description, and price for placing orders

Appendix B - Group Log

Meeting Minutes:

- Meeting Date: March 21, 2024
    - Attendees: Anushka, Varenya, Nirzari
    - Discussion: Discussed project requirements and divided tasks.
    - Action Items: Alice to research reservation systems, Bob to work on menu management, Charlie to explore payment processing.
- Meeting Date: March 28, 2024
    - Attendees: Anushka, Varenya, Nirzari
    - Discussion: Reviewed research findings and discussed system architecture.
    - Action Items: Alice to create wireframes for reservation system, Bob to design menu interface, Charlie to investigate payment APIs.
- Meeting Date: Aril 6, 2024
    - Attendees: Anushka, Varenya, Nirzari
    - Discussion: Presented individual progress and discussed integration.
    - Action Items: Alice to finalize reservation workflow, Bob to implement menu CRUD operations, Charlie to test payment integration.
- Group Activities:
    - Research: Each member conducted research on their assigned topic (reservation, menu, payment) and shared findings with the group.
    - Design: Collaboratively designed the system architecture and UML diagrams for the restaurant system.
    - Documentation: Documented the project requirements, design decisions, and implementation details for future reference.
- Effort Put Forth:
    - Each member spent approximately 10-15 hours per week on research and documentation.
    - Weekly meetings were held to discuss progress, resolve issues, and plan next steps.
    - Regular communication via messaging apps and shared documents to coordinate tasks and share updates.