**NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES**

**(KARACHI CAMPUS)**

**FAST School of Computing**

**Spring 2024**

**Course:** Software Engineering (CS3009)

**Course Instructor:** Ms. Javeria Farooq

**Project Name**: Restaurant Food Ordering App

**Project Milestone**: Software Requirement Specification Document

**Team Members:**

21K-3279 Insha Javed

21K-4503 Muhammad Tahir

21K-3317 Naqeeb Nadir

21K-4606 Sabika Shameel

**Software Requirement Specification**

**(Restaurant Food Ordering App)**

1. **Introduction**

In today's fast-paced world, the convenience of online ordering has become paramount, and our system aims to meet the evolving needs of both customers and restaurant administrators. With our user-friendly website, customers can browse a wide selection of delectable dishes, place orders effortlessly, and complete secure payments—all from the comfort of their homes. Meanwhile, for restaurant administrators, our RMS provides powerful tools for inventory management, allowing them to add new menu items, delete outdated dishes, and maintain a comprehensive overview of stock levels.

**1.1 Motivation**

Our project, the Restaurant Management System, is motivated by the need for a better way to handle food orders and inventory in restaurants. We want to make it easier for customers to order their favorite meals online and for restaurant owner to keep track of their inventory.

* To create a user-friendly system for food ordering
* To ensure security in payment
* To allow restaurant staff to view customer’s food order record

**1.2 Stakeholders**

* Muhmmad Tahir (Website Developer)
* Insha Javed (Technical Designer)
* Sabika Shameel (Business Analyst)
* Naqeeb Nadir (Tester)
* Ritesh Rajput (Product Client)

**1.3 Assumptions and Dependencies**

**` Assumptions:**

* Customers have access to the internet and can use the website to place orders.
* Restaurant staffs are trained to use the RMS for processing orders.

**Dependencies:**

* Availability of internet connectivity for both customers and restaurant staff.
* Access to accurate and up-to-date inventory information from suppliers.

1. **Functional Requirements**

**2.1 User Registration and Login:**

* Customers can create accounts and log in to place orders and view order history.
* Secure authentication ensures user privacy.

**2.2 Menu Browsing and Selection:**

* User-friendly interface for browsing the restaurant’s menu.
* Customers can view food items and select dishes they wish to order.

**2.3 Order Placement:**

* Customers can add selected food items to their cart and specify quantities.
* Secure order placement process.

**2.4 Payment Processing:**

* Multiple payment methods available for customer convenience.

**2.5 Customized Dishes:**

* Customers can provide preferences (e.g., spice level, dietary restrictions).
* Staff prepares meals accordingly.

**2.6 Discounts, Special Offers, and Packages:**

* Customers can avail special deals and discounts.
* Enhances customer loyalty.

**2.7 Order Status Updates:**

* Staff can view order status (processing, out for delivery, delivered).
* Real-time notifications for customers.

**2.8 Contact Details:**

* Customers can easily reach out to restaurant management for queries or complaints.

**2.9 Receipt Preview:**

* + Customers can review payment details before placing an order.
  1. **Cart Transections:**
  + Customers can add multiple items in the cart, as well as remove them.
  1. **Add Products:**
  + Customers can add an item on the menu along with its description and prices.
  1. **Digital Payment:**
  + Customers will have multiple options to get their order payment done. They will be able to pay via credit card as well as cash on delivery.

1. **Non-Functional Requirements**

**3.1 Performance**

The system should be responsive and handle multiple concurrent users without significant slowdowns or delays. Response times for actions such as order placement and menu browsing should be fast to provide a seamless user experience.

**3.2 Reliability**

The system should be always reliable and available for use, minimizing downtime and service disruptions. It should have backup and recovery mechanisms in place to ensure data integrity and availability in case of system failures or disasters.

**3.3 Security**

The system should ensure the security and confidentiality of user data, including personal information and payment details. It should enforce authentication and authorization mechanisms to prevent unauthorized access to sensitive information.

**3.4 Usability**

The system should be intuitive and easy to use for both customers and restaurant staff, requiring minimal training to navigate and perform tasks. User interfaces should be well-designed, with clear instructions and feedback to guide users through the ordering process effectively.

1. **Constraints**

**4.1 Technical Limitations**

Limited availability of certain technologies or programming languages may restrict the implementation of specific features or functionalities within the system.

**4.2 Time Constraints**

The project may be subject to deadlines or timeframes that restrict the amount of time available for development and testing phases, potentially affecting the scope and quality of deliverables.

**4.3 Resource Constraints**

Limited availability of human resources, such as developers, testers, or project managers, may impact the project's progress and execution.

1. **Architecture Design**

**5.1 Overview**

The Restaurant Management System (RMS) architecture consists of four main components: the Client Interface, Server-Side Application, Data base. Customers interact with the Client Interface to place orders and track them. The Server-Side Application processes these orders, handles payments, and updates inventory, communicating with the Database to store and retrieve data. These components interact seamlessly to provide a smooth and reliable ordering experience for customers while enabling efficient management of restaurant operations.

**5.2 Component Diagram**

**A screenshot of a computer

Description automatically generated**

**5.3 Deployment Diagram**

A screenshot of a computer

Description automatically generated

1. **Revision History**

| **Version** | **Date** | **Description** |
| --- | --- | --- |
| 1.0 | March 2, 2024 | Initial draft of the software requirement specification (SRS) document was created. |
| 1.1 | March 5, 2024 | Added functional requirements for user registration, login, menu browsing, and selection. |
| 1.2 | March 8, 2024 | Added functional requirements for order placement and payment processing. |
| 1.3 | March 11, 2024 | Added non-functional requirements addressing performance, reliability, security, and usability. |
| 1.4 | March 14, 2024 | Added constraints related to technical limitations, time constraints, and resource constraints. |
| 1.5 | March 17, 2024 | Added an overview of the architecture design, detailing the Client Interface, Server-Side Application, and Database components. |
| 2.0 | March 20, 2024 | Completed the front-end development of the application as per the outlined functional requirements. Made revisions to the SRS document to reflect the completion of this milestone. |
| 2.1 | March 23, 2024 | Conducted testing of the front-end components and updated the SRS document to detail the results and any identified issues. |
| 2.2 | March 26, 2024 | Implemented fixes and improvements based on front-end testing feedback. Updated the SRS document to reflect these changes. |
| 2.3 | March 30, 2024 | Finalized the front-end development, completing all necessary revisions. Updated the SRS document to reflect the completion of this stage. |