4. Software Requirements Specification

4.1 Introduction

4.1.1 Purpose

The purpose of the "QRSAY" project is to develop a user-centric application that revolutionizes the online food ordering experience, providing a seamless and convenient platform for users to explore menus, place orders, and enjoy diverse dining options. The project aims to address operational inefficiencies in the traditional restaurant landscape and introduce contactless dining solutions.

4.1.2 Product Scope

The scope of the project encompasses the development of a comprehensive application with features such as secure registration, diverse ordering options (delivery, take away, dining, scheduled dining, and party booking), AI-driven health and age-based food recommendations, and integration of health advisory guidelines. The project also includes the implementation of a responsive chatbot and personalized food recommendations using AIML technology.

4.2 Overall Description

4.2.1 Product Perspective

The "QRSAY" application exists as a standalone system, providing a user interface for customers and a backend system for managing restaurant data, menus, orders, and user profiles. It interfaces with external systems for payment processing and communication.

4.2.2 Product Features

The key features of the application include secure registration, comprehensive restaurant database, streamlined order placement, real-time tracking, secure payment processing, AI-driven chatbot assistance, and personalized food recommendations. Additional features include party booking, event management, and integration of health advisory guidelines.

4.2.3 User Classes and Characteristics

The application caters to two main user classes: customers and restaurant staff. Customers include individuals seeking to explore and order food, while restaurant staff encompasses those managing menus, orders, and restaurant operations.

4.2.4 Operating Environment

The application operates in a digital environment, with customers accessing it through web and mobile interfaces. The backend is hosted on server infrastructure supporting MongoDB, Express.js, Angular, and Node.js (MEAN stack).

4.2.5 Design and Implementation Constraints

The project employs MongoDB for data storage, Express.js and Node.js for backend development, Angular for frontend development, and AIML technology for chatbot assistance. These technology choices impose constraints related to the design and implementation of the system.

4.3 System Features

4.3.1 Feature: Based on User Preferences

4.3.1.1 Description and Priority

This feature focuses on providing personalized recommendations based on user preferences, including dietary restrictions and health goals. It is of high priority to enhance the overall user experience.

4.3.1.2 User Classes and Characteristics

Customers seeking personalized food recommendations and health-conscious choices.

4.3.1.3 Functional Requirements

- Implement AI algorithms for analyzing user preferences.
- Integrate health advisory guidelines into menu displays.

4.4 External Interface Requirements

4.4.1 User Interfaces

The application provides user interfaces for customer registration, menu browsing, order placement, and real-time tracking. A responsive and dynamic interface is developed using Angular.

4.4.2 Hardware Interfaces

Server infrastructure is required for hosting both the application and database servers.

4.4.3 Software Interfaces

The application interacts with MongoDB for data storage and utilizes Express.js and Node.js for backend development. AIML libraries are integrated for chatbot capabilities.

4.4.4 Communications Interfaces

The application communicates with external systems for payment processing and may integrate with third-party services for additional functionalities.

4.5 Nonfunctional Requirements

4.5.1 Performance Requirements

- Response time for user interactions should be within acceptable limits.
- The system should handle a large number of concurrent users.

4.5.2 Safety Requirements

- Ensure the secure storage of user data and payment information.

4.5.3 Security Requirements

- Implement secure authentication and authorization mechanisms.
- Encrypt communication between the application and external systems.

4.5.4 Software Quality Attributes

- Ensure a user-friendly and intuitive interface.
- Conduct rigorous testing, including unit, integration, user acceptance, performance, and security testing.

4.5.5 Other Requirements

- The system should be scalable to accommodate future enhancements.
- Ongoing maintenance and updates should be performed to address issues and introduce new features.

4.6 Conclusion

The "QRSAY" project aims to deliver a comprehensive and innovative solution for online food ordering, addressing current industry challenges and aligning with the evolving needs of both customers and restaurants. The outlined requirements provide a detailed roadmap for the successful development and deployment of the application, ensuring a seamless and enriching culinary experience for all users.