Software Requirements Specification

for

Online Food Delivery System

Version 1.0 approved

Prepared by:

Shreeman R - 18BTRSE025
Srijan Kumar Srivastava - 18BTRSE028
T Aswin Barath - 18BTRSE031
Vyshnav Raj - 18BTRSE033

Jain University School of Engineering and Technology

07/03/2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Creation	06/03/2021	Creation of SRS document	1.0
Updation	07/03/2021	Adding new System features and requirements	1.1

1. Introduction

1.1 Purpose

The purpose of an online food delivery system(website/application) is to give a bug free, user friendly experience that ease the ordering system and simplify the overall process.

1.2 Document Conventions

Higher-level requirements use highest fonts and every other requirement statement has been subdivided to be inherited requirements. Each requirement uses numbered bullet points with bold typographical convention.

1.3 Intended Audience and Reading Suggestions

This project is for the online food delivery system and it is for our country premises. This has been implemented under the guidance of restaurant rules and payment regulation. This project is useful for the restaurants and as well as to the customers.

1.4 Product Scope

The purpose of this online food delivery system is to ease flight management and to create a convenient and easy-to-use application for customers who are trying to order food items from various restaurants and hotels. We will have a database server supporting hundreds of major cities and restaurants. Above all, we hope to provide a comfortable user experience along with the best pricing and offers will be available based on the variety of restaurants and the food items.

2. Overall Description

2.1 Product Perspective

The database system of the online food delivery system stores the following information such as:

- *Restaurant Details:*
  It includes the details of various restaurants, their food item list ,offers, and their payment options , pricing (includes GST).
- *Location Details:*
  It contains map and location details of restaurants and other information.
- *Customer details:*
  It includes customer personal details like name phone number, email address, location (optional) and order history and recommendation data.

2.2 Product Functions

The major features of Online Delivery System database system as shown in entity relationship diagram - refer **Appendix B: Analysis Models***.*

2.3 User Classes and Characteristics

The ***User class*** *of the system should be able to see restaurant information within a given city with the given time. The customer(user) should be able to do the following functions:*

1. *Make a new order*
2. *One user can take 3 orders at a time(In future we will remove the order restriction)*
3. *User can cancel the order with a valid reason*
4. *Will get a different mode of payment details like UPI,CREDIT CARD,DEBIT CARD, Cash on Delivery etc.*
5. *User can see the order history and personal recommendation*

The ***Employee class*** *should have following management functionalities:*

1. *Get all customers who have reserved on a given food item.*
2. *Get all information of food item in a given restaurant*
3. *View food available time schedule.*
4. *Calculate total sales for a given restaurant.*

The ***Administrator class*** *should have following functionalities:*

1. *Add/Delete a Food item*
2. *Add a new restaurant*
3. *Update fare for new food dishes*
4. *Update order place and arrival time*
5. *Update the restaurant pricing and offers*

2.4 Operating Environment

- *Distributed database systems*
- *REST API services (Client/Server System)*
- *Operating system: Windows.*
- *Database platform: PostgreSQL and MongoDB*
- *Development Platform and libraries: React.js , Node.js, Express.js, Ionic, Capacitor, NPM*

2.5 Design and Implementation Constraints

- *Compatibility issues on user devices*
- *Breaking changes due to new version releases*
- *Trade-off b/w High data usage over performance*
- *Security issues on Payment Gateways*
- *JavaScript requirement from client-side*
- *Regulatory policies on food delivery*
- *Certified sellers and licenced delivery man.*

2.6 Assumptions and Dependencies

Let us assume that this is a online food delivery system and it is used in the following application:

- *A request for ordering /cancellation of food items from any customer to any restaurant, connected.*
- *Calculation of high fliers (most frequent fliers) and calculating appropriate points for these fliers.*
- *We have to design a global schema and SQL commands for above queries/application.*
- *Then Implement the database at least using a centralized database management system.*

3. External Interface Requirements

3.1 User Interfaces

- *Front-end client software: React.js, Redux, React-DOM-router*
- *Back-end software: NPM, Node.js, Express.js, Ionic, Capacitor*

3.2 Hardware Interfaces

- Pentium Processor (or above)
- 60 MB of free hard-drive space
- 128 MB of RAM

3.3 Software Interfaces

- *The system will support these OS: Windows, Android, iOS, iMac, Linux flavours*
- *Databases: PostgreSQL and MongoDB*
- *Frameworks: React ecosystem libraries (JavaScript ES6 features based)*
- *Operating System: Windows (Vista/7 or above)*
- *Web Browser: IE 10 or above, Mozilla FF 31 and above or Google Chrome*
- *Drivers: Java Runtime Environment(JRE - for Android devices)*
- *Integrated Development Environment: JavaScript Compatibility*

3.4 Communications Interfaces

- *REST API to contact the client-side.*
- *Using SMTP protocols to send user transaction details as emails.*
- *FTP protocols to access file systems at backend servers.*
- *HTTPS protocols for sending secure client-side HTTP requests*
- *SSL protocols to verify and authorize social accounts.*

4. System Features

4.1 Priority Queue Feature

4.1.1 Description and Priority

The Online food delivery system maintains information on foods, category, restaurant, personal preferences, prices, and bookings based on priority of regular customers versus premium customers.

4.1.2 Stimulus/Response Sequences

a. *Search for food items in two different cities*
b. *Displays a detailed list of available food items or dishes and makes a "order" or book a favourite dish on a particular restaurant.*
c. *Cancel an existing order.*

4.1.3 Functional Requirements

Client/Server model:

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- *Some sites are client sites and others are server sites.*
- *All the data resides at the server sites.*
- *All applications execute at the client sites.*

5. Other Nonfunctional Requirements

5.1 Performance Requirements

> Client-side Requirements:

- *Minimum Memory of 2GB RAM is required*
  Medium to High data rates for downlink speed is required
- *Stable Internet Connection*
- *GPS feature is required*
- *Minimum of 100MB storage for mobile devices to download client-side apps*

> Server-side Requirements:

- *Capability of supporting no less than 200 concurrent connections from any combination of surface computers, tablets and displays, it shall provide no limit on how many devices are in the system.*
- *Supporting an arbitrary number of active meals/orders, that is, no meals/orders shall be lost under any circumstances and no payments shall be lost under any circumstances.*

5.2 Safety Requirements

This system shall log every state and state change of every surface computer, tablet and display to provision recovery from system failure. The system shall be capable of restoring itself to its previous state in the event of failure (e.g. a system crash or power loss).The system shall be able to display a menu at all times to facilitate manual order taking should the need arise. The system shall utilize periodic 30-second keep-alive messages between tablets and the server to monitor tablet operational status. The system shall flag tablets that fail to send timely keep-alive messages as non-operational and disassociate the assigned waiter from the tablet.

5.3 Security Requirements

Wireless communication throughout the system will be encrypted using SSLv3 at the application layer and WPA2-PSK at the data link layer. The WPA2-PSK password used for wireless communication must have a bit-strength of at least 80 bits. The WPA2-PSK password used for wireless communication must be changed every three months. A waiter password used for tablet login must have a bit-strength of at least 64 bits. A waiter password used for tablet login must be changed every three months. A waiter shall only be able to log into one tablet at any given instance of time. A waiter that attempts to log into a second tablet while already logged into another tablet shall be rejected and notified through both tablets.

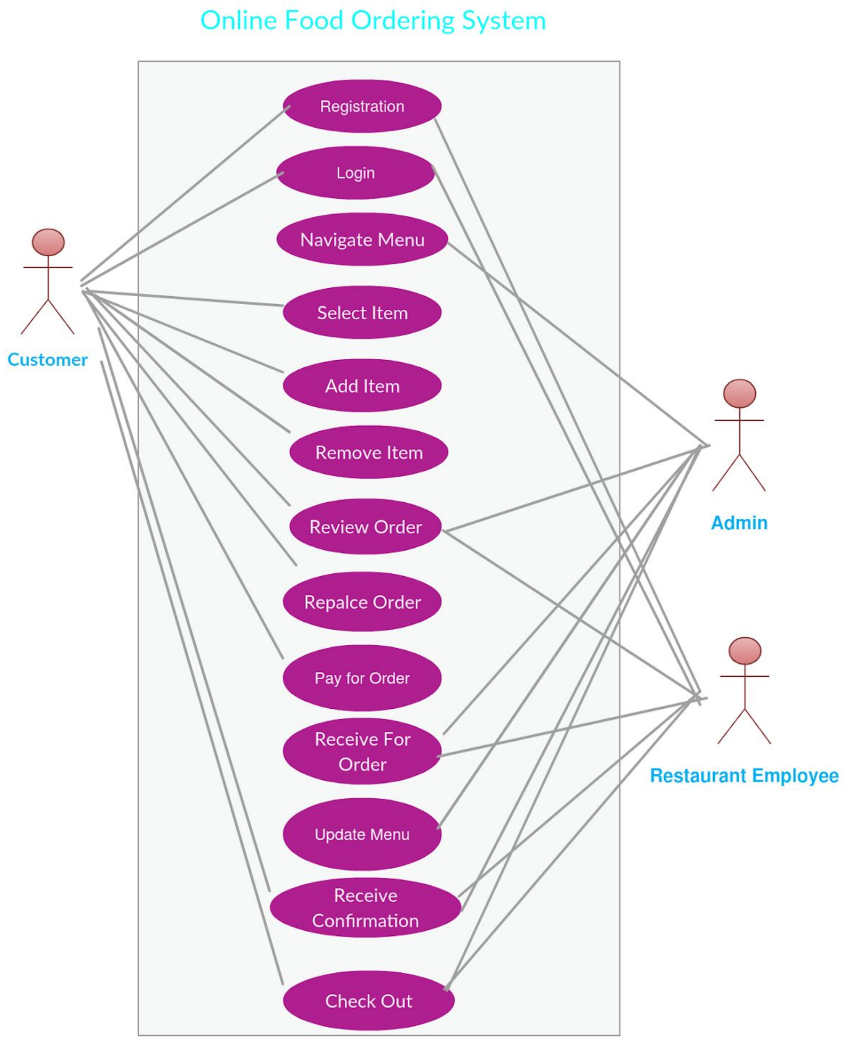
5.4 Software Quality Attributes

- *AVAILABILITY: The flight should be available on the specified date and specified time as many customers are doing advance reservations.*
- *CORRECTNESS: The flight should start from the correct start terminal and should reach the correct destination.*
- *MAINTAINABILITY: The administrators and flight in chargers should maintain correct schedules of flights.*
- *USABILITY: The flight schedules should satisfy a maximum number of customers.*

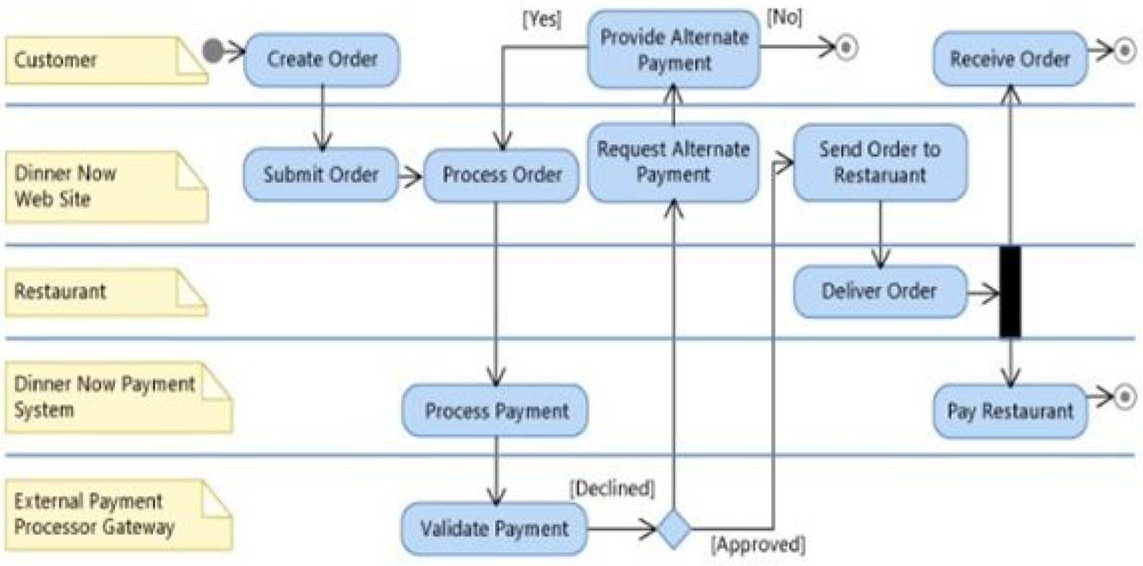
6. Other Requirements

Appendix A: Analysis Models

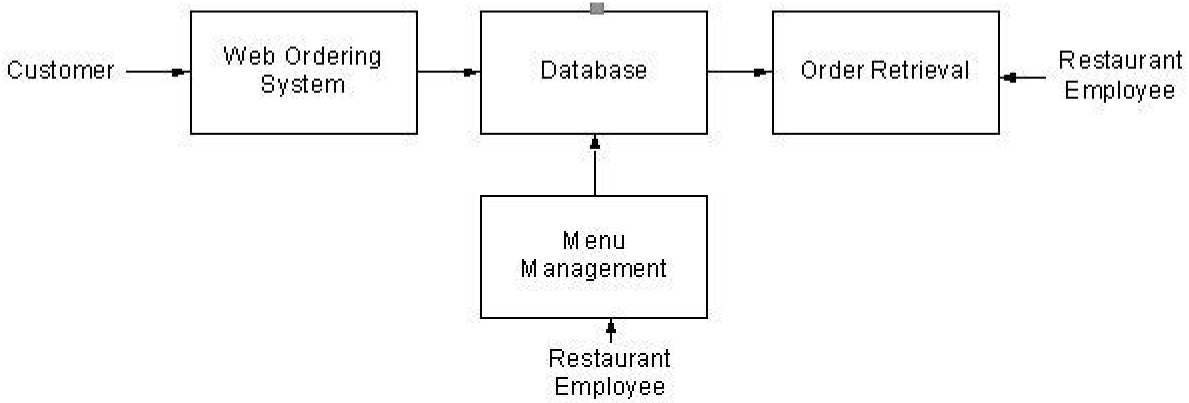
Entity - Relationship diagram:



Activity diagram:



System Model:


Customer
Web Ordering System
Database
Order Retrieval
Restaurant Employee
Menu Management
Restaurant Employee