SOFTWARE REQUIREMENTS SPECIFICATION

for

Food Delivery Service Management Software (FDSMS)

Version 1.0

Prepared by :
Apoorv Kumar (21CS10008)
Om Sadhwani (21CS10046)
Ronit Nanwani (21CS30043)

Indian Institute of Technology Kharagpur

March 26, 2023

Contents

1 Revision History 3

2 Introduction 4
2.1 Purpose . . . 4
2.2 Document Conventions . . . 4
2.3 Intended Audience and Reading Suggestions . . . 4
2.4 Product Scope . . . 4
2.5 References . . . 5

3 Overall Description 6
3.1 Product Perspective . . . 6
3.2 Product Functions . . . 6
3.3 User Classes and Characteristics . . . 6
3.4 Operating Environment . . . 7
3.5 Design and Implementation Constraints . . . 7
3.6 User Documentation . . . 7
3.7 Assumptions and Dependencies . . . 8

4 External Interface Requirements 9
4.1 User Interfaces . . . 9
4.1.1 HomeScreen . . . 9
4.1.2 Customer . . . 9
4.1.3 Restaurant . . . 10
4.1.4 Delivery Agent . . . 10
4.1.5 Management . . . 10
4.2 Hardware Interfaces . . . 11
4.3 Software Interfaces . . . 11
4.4 Communication Interface . . . 11

5 System Features 12
5.1 New Customer/Delivery Agent/Restaurant . . . 12
5.1.1 Description . . . 12
5.1.2 Stimulus/Response Sequences . . . 12
5.1.3 Functional Requirements . . . 12
5.2 Menu of Restaurant . . . 12
5.2.1 Description . . . 12
5.2.2 Stimulus/Response Sequences . . . 12
5.2.3 Functional Requirements . . . 12

5.3 Order by Customer . . . 13
5.3.1 Description . . . 13
5.3.2 Stimulus/Response Sequences . . . 13
5.3.3 Functional Requirements . . . 13
5.4 Promotional Offers . . . 13
5.4.1 Description . . . 13
5.4.2 Stimulus/Response Sequences . . . 13
5.4.3 Functional Requirements . . . 13
5.5 Feedback . . . 14
5.5.1 Description . . . 14
5.5.2 Stimulus/Response Sequences . . . 14
5.5.3 Functional Requirements . . . 14
5.6 Restaurant . . . 14
5.6.1 Description . . . 14
5.6.2 Stimulus/Response Sequences . . . 14
5.6.3 Functional Requirements . . . 14
5.7 Delivery Agent . . . 15
5.7.1 Description . . . 15
5.7.2 Stimulus/Response Sequences . . . 15
5.7.3 Functional Requirements . . . 15
5.8 Manegement . . . 15
5.8.1 Description . . . 15
5.8.2 Stimulus/Response Sequences . . . 15
5.8.3 Functional Requirements . . . 15

6 Use Case Diagram **16**

7 Class Diagram **17**

8 Other Nonfunctional Requirements **18**
8.1 Performance Requirements . . . 18
8.2 Safety Requirements . . . 18
8.3 Software Quality Attributes . . . 18
8.4 Business Rules . . . 18

1 Revision History

Name	**Date**	**Reason For Changes**	**Version**
Creation	26th March	Template Changes and Additions	1.0

2 Introduction

2.1 Purpose

The purpose of this document is to provide a detailed description of the Food Delivery Service Management Software. There are four main type of users of this Software: Customer, Restaurant, Delivery Agent, Management. The application is designed in such a way that the interface, functionality and constraints of the system will vary from user to user. For example, the Restaurant will be able to change their menu and assign delivery agents whereas the customer can choose from a number of Restaurants and place their order. Ratings of Restaurants, Delivery Agents and Customers will be maintained to improve user experience. This software will prove beneficial to all its users as it increases convenience of the customer, reach of the restaurants and employment of delivery agents.

2.2 Document Conventions

This Software Requirements Specification Document has been written using free online document preparation software LaTeX typed in Computer Modern Roman font for normal text as well as headings. The font size used is 10.95 pt for normal text and 20.74 pt for headings. All headings are highlighted appropriately in bold. The document is prepared using UK English convention.

2.3 Intended Audience and Reading Suggestions

The SRS Documentation covers all the technical and non-technical aspects of the software. It is intended to assist developers and other end users to understand the motivation behind the software and understand the implementation intricacies in it. Anybody who wants to use the software can read the appropriate parts of the document, a list of which is given in the Table of Contents of Page 1

2.4 Product Scope

FSDMS is a web based application intended for customers to order food online, for restaurants to provide services and local people to register as delivery agents. The application is based on a client-server model where the users can interact with the interface provided by the frontend and the data is managed by the backend server via an online database.

2.5 References

This document is based on the IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specification given by the IEEE Computer Society in 1998.

3 Overall Description

3.1 Product Perspective

This is a new self-contained product in development that was born out of the recognition that people sometimes find it inconvenient or impossible to visit restaurants due to factors like lack of time, discomfort, or lack of transportation. The project aims to address this issue by offering safe food delivery from popular restaurants, along with appealing discounts and low delivery fees.

3.2 Product Functions

The key features of FDSMS are as follows:

- Provide food delivery to customers directly from restaurants and generate a bill for online payment
- Employment opportunity for local people as food delivery partners
- Customer can see the estimated time of arrival of food as provided by the delivery agent and the restaurant
- Maintain a menu for all the restaurants
- Maintain ratings for customers, delivery agents and restaurants so as to improve user experience and provide offers to high rated customers

3.3 User Classes and Characteristics

- Customer: who orders food from various restaurants:
  - See menus of restaurants and order a subset of items.
  - View his / her past orders
  - See the status of his pending orders
  - Provide feedback about quality of restaurant or delivery agent
  - See promotional offers if any.
- Restaurant:
  - Should be able to create a menu

  - See the orders placed to this restaurant in the past.
  - See the pending orders, accept or reject them and also provide a time of service
  - See available nearby delivery agents, and send delivery requests for accepted orders.
  - Mark accepted orders as “out for delivery”.

- Delivery Agent:
  - Should be able to mark their location and see delivery requests from nearby restaurants
  - Should be able to accept delivery request, pick up the order, and deliver the order, while updating status for all the tasks.
  - Should be able to provide estimated pickup time for accepted order and estimate delivery time for picked up order.
  - Should be able to rate the customer

- Management:
  - Should be able to manage the lists of customers, Restaurants, and delivery agents
  - Maintain ratings for customers, Restaurants, and delivery agents
  - Provide restaurant and food recommendation for customers
  - Provide promotional offers for to good customers

3.4 Operating Environment

Since the users are interacting as clients on a GUI based web application, the software is independent of Operating System. The GUI is made with the help of React JS Framework. The backend and server side uses the flask framework in python which interacts with an online firebase database.

3.5 Design and Implementation Constraints

The use of firebase databases from google limits us from the usage policy of the database and changes in policy can affect the working and functionalities of the system.

3.6 User Documentation

The GUI interface makes the usage of the application pretty intuitive.

3.7 Assumptions and Dependencies

It is assumed that the user possesses both a functional internet connection and an email address in order to access the web-app. The dependencies for the project will be available in the requirements.txt file of the project.

4 External Interface Requirements

4.1 User Interfaces

The home page of the web application will consist of login and sign up options for all the users, namely customer, restaurant, delivery partners and management. Upon logging in users will be redirected to their respective dashboards.

4.1.1 HomeScreen

- About Us
  - Contain brief information of the website
- Login
  - Ask username and password
  - Redirect to respective dashboard
- Sign-Up
  - Register a new user of the website

4.1.2 Customer

- Access menus of restaurants
  - Order from a list of items provided by a restaurant
  - Proceed for payment after his choice of items
  - Can avail promotional offers, if any
- Provide Feedback
  - Provide feedback about restaurants and delivery agent
- Personal Details
  - See previous orders from their account
  - See status of his pending orders as provided by the restaurant and the delivery agent

4.1.3 Restaurant

- Create Menu
- Process Orders
  - Choose whether to accept or reject the orders placed by customers
  - Provide an estimate time for delivery if order has been accepted
  - Assign a delivery agent for the current order
- Personal Details
  - See all the orders placed to this restaurant previously

4.1.4 Delivery Agent

- Mark the current location
- Current Requests
  - See if there are any requests from restaurants to deliver food
  - Accept or reject the request
  - Provide an estimated time of arrival
- Feedback
  - Provide ratings to customers
- Personal Details
  - See all the previous deliveries

4.1.5 Management

- Customers List
  - Maintain ratings of customers
  - Provide recommendations for customers based on past orders
  - Provide promotional offers to good customers (based on customer ratings)
- Restaurant List
  - Maintain ratings of restaurants
- Delivery agents
  - Maintain ratings of delivery agents

4.2 Hardware Interfaces

There are no specific requirements for the hardware interface on the part of the users, however for best experience it is recommended to use a Desktop PC or a Laptop for better display of web contents.

4.3 Software Interfaces

As a final step, the application will be deployed as a web-app. For testing purposes it will be hosted locally. The web application is browser independent as well as Operating System independent making it convenient for users across various platforms. The Backend for the application is implemented using Python's Flask framework and the database is maintained by the firebase database. The client side GUI is implemented using the React JS framework.

4.4 Communication Interface

All communication will take place through a web browser using the standard HTTPS protocol.

5 System Features

5.1 New Customer/Delivery Agent/Restaurant

5.1.1 Description

A new user on the platform would be required to sign up to keep using the platform. Sign up for different profiles will have different features.

5.1.2 Stimulus/Response Sequences

The user will be redirected depending on the type of the user. There are different sign up pages for different users and all the relevant information pertaining to that type of user will be recorded and stored in the database.

5.1.3 Functional Requirements

At the home page there is sign up and login options for all types of users. If its the first time that the user is visiting the site and wants to sign up he can go to the respective sign up page. If the user has already been added to the database he can login using the credentials provided by him during sign up for further use of the website.

5.2 Menu of Restaurant

5.2.1 Description

The restaurant will be asked to create a menu during the signup. Then after signup it will be redirected to the restaurant dashboard page from where it can add item, delete item and make necessary changes. The customer can also view the menu of different restaurants from the customer dashboard page.

5.2.2 Stimulus/Response Sequences

The menu will be shown in a table form to the restaurant and there will be different buttons such as add, delete, edit to make the necessary changes. For customers the menu will be in the table form and there will be a single button of Add item to cart.

5.2.3 Functional Requirements

Some basic frontend Javascript would be required to add item to table and database to store the items

5.3 Order by Customer

5.3.1 Description

The customer can check the order status on the web-app. The different types of order status are Accepted/Rejected, Preparing food, Handed out for delivery, Delivery complete. After the customer has placed an order to a particular restaurant, the restaurant can Accept or Reject the order. After accepting an order the restaurant can update the status of the order from time to time. Once it is handed out for delivery then the delivery agent can update the status.

5.3.2 Stimulus/Response Sequences

The status will be presented as a timeline with checkpoints acting as the steps and paths/connections showing the typical time needed for the following step, which will be supplied by the restaurant or the delivery service.

5.3.3 Functional Requirements

The time of the updates will be shown by date time library and the status will be shown on front-end

5.4 Promotional Offers

5.4.1 Description

The customers will get promotional offers and discounts based on the ratings given by delivery Agents and Management. The customers can use these benefits once. If the management determines that the consumers have good ratings, they will make the offers to them. Customers may receive newly made offers or already created offers .

5.4.2 Stimulus/Response Sequences

These offers will be shown on the customer dashboard page under a section of offers. To use the offer the customer need to click on the offer and that offer will automatically get removed from the offers list. There will be a list of offers that the management can look at and add/delete offers in that list. They will give that offer to customers using Coupon Codes.

5.4.3 Functional Requirements

Each offer will be stored as a class and offerIds will be stored in the details of the user and the access for the offers will be done from the database.

5.5 Feedback

5.5.1 Description

The customer can give feedback to the restaurants and the delivery agents, and the delivery agents can give feedback to the customers according to the services and the behaviors.

5.5.2 Stimulus/Response Sequences

The feedback/rating stars are stored as floating-point numbers and will be averaged based on the previous rating. These will be given to the restaurants and the delivery agent respectively

5.5.3 Functional Requirements

The function will be requiring javascript system to take the feedback, it will be stored in the database.

5.6 Restaurant

5.6.1 Description

The restarant can accept or reject an order. After accepting an order it will update the status and put up a notification, the delivery agent will see the notification and will be able to choose the order. The restaurant will be able to see the list of orders and will update the status of the order based on that. Once the order is handed out for delivery that particular order will be added to the list of past orders of the restaurants

5.6.2 Stimulus/Response Sequences

When order is prepared, the restaurant can put up the order as available for pickup by clicking a button, which will add the order in the delivery agent's dashboard as a pickup order. A list of all current orders will be displayed, and the restaurateur can click on any of them to change their status. Using the orders button on the panel, you can access the list.

5.6.3 Functional Requirements

These functionalities can be implemented using Javascript , CSS , database and flask.

5.7 Delivery Agent

5.7.1 Description

Delivery Agents can mark their area. When the delivery requests are posted by the restaurants these show up in a list to all the nearby delivery Agents ans they can choose any request from this list. They will also provide the estimated time of arrival.

5.7.2 Stimulus/Response Sequences

The delivery agent will fill the estimated time in the text box according to their estimates while accepting the order. The time will be updated in the status bar.As the list of the orders is formed, there will be a button to accept only one at a time delivery request and then add estimated time to the restaurant and then to the customer.

5.7.3 Functional Requirements

This will require the use of Flask for updating and the marking location.

5.8 Manegement

5.8.1 Description

the management can recommend food and restaurants to all the customers. The list of customers, restaurants and delivery agents can be seen in different tabs and there they can see the details of the user. The management can also give offers to customers based on their rating.

5.8.2 Stimulus/Response Sequences

There will be options to push offers and recommend foods, by clicking the buttons or selecting the checklist buttons. The management can check the rating of the restaurants, and then select the restaurants and the food items they want to recommend. This will change the boolean value of the item and if the boolean is true, the system will know that it is recommended and hence will show it in the recommended list

5.8.3 Functional Requirements

The use of JavaScript and database will give the data, and CSS will help to display the data

6 Use Case Diagram

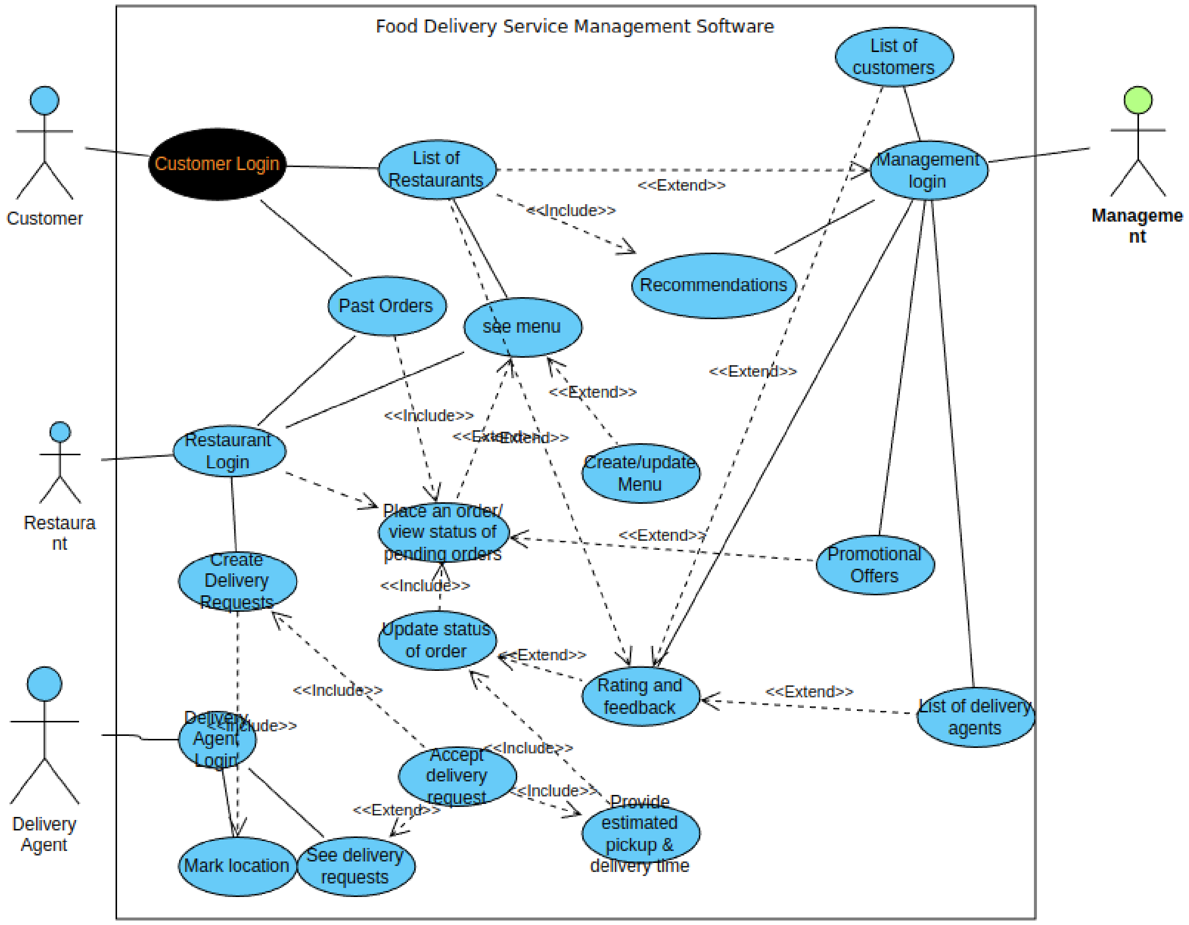

Figure 6.1: Use Case Diagram

7 Class Diagram

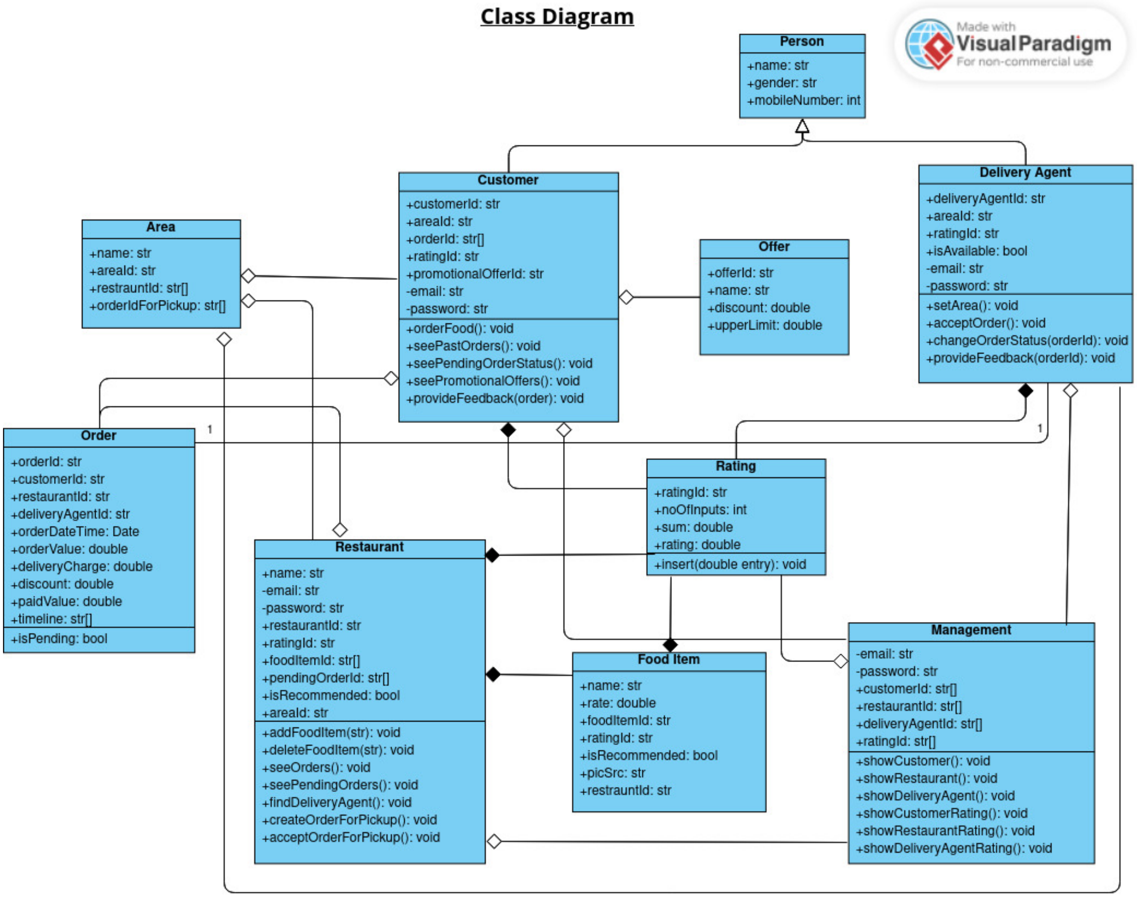

Figure 7.1: Class Diagram

8 Other Nonfunctional Requirements

8.1 Performance Requirements

To enable fast database queries and ensure relevant results, a suitable balance between speed and precision can be achieved through the use of Firebase.

8.2 Safety Requirements

Since the application is web-based, it poses minimal risk to the user's device. However, due to the significant amount of data reading and writing involved, potential harm to the server and data storage may occur only during periods of heavy usage.

8.3 Software Quality Attributes

- **Flexibility** : Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.
- **Maintainability** : The product's various iterations should be simple to maintain. An current system should be simple to add code to for development purposes, as well as simple to upgrade for occasionally new features and new technologies. It should be inexpensive and simple to maintain.
- **Usability** : The ease of usage can be used to gauge this. An application should be simple to use. Learning should be simple. It should be easy to navigate

8.4 Business Rules

The software will be free to use for all users and the source code will be publicly hosted for free use and modification.