\

Software Requirements Specification

for

SSFDS (Social Serving Food Delivery System)

Version 1.0 approved

Prepared By

Tireless Whoopers

IIT Kharagpur

16.03.2022

Table of Contents

1. Introduction

1.1 Purpose

It is proposed to build a software, Social Serving Food Delivery System (Indicated as 'SSFDS') to keep tracks of leftover food from restaurants and hotels, which will be available for sale to NGOs and customers at lower prices. The purpose of this document is to present a detailed description of SSFDS. It will explain the purpose, features of this project and scopes of application. The interface and the details about interaction through the interface with database, data management strategies, constraints under which it will operate will also be there for an easy guide to the developers. This document is intended for understating requirements and specification in details.

1.2 Document Conventions

1. An asterisk after a word indicates more information about it in the glossary
2. To denote sub headers dot separated numbers are used, for ex 1.2.1 is sub header of 1.2
3. The priority of features is indicated as high, medium and low keywords in braces in the System Features section [4].

1.3 Intended Audience and Reading Suggestions

This document is intended for developers contributing towards the project, the client, users, and testers and anyone who is reviewing the project to easily view descriptions about functional and no-functional requirements and understand product scope.
This document contains six (6) sections:
a. An overall description, which includes a summary of the scope, assumptions and limitations of the project, types and characteristics of users, plan for implementation.
b. A brief description of external interfaces requirements.
c. A description of the system, which includes an explanation of the system's purpose, features and benefits, characteristics, and technical and operational background; and,
d. A list of nonfunctional requirements, which contains information regarding performance, safety, security and business rules.
e. A description of other uncategorized requirements.
f. A list of the appendices. (Definition of terms used in the document, analysis model UML and future extension possibilities) This is a suggested sequence to go through the SRS: -product scope and perspective -product function -user classes and characteristics -Appendix B -system features -other requirements

1.4 Product Scope,

The Social Serving Food Delivery System is being developed mainly to distribute foods among the poor and needy children through NGOs. Besides, this system will also help to reduce the food wastage. Here, restaurants will update information about leftover food on their respective accounts. These will reflect to accounts of all the nearby (less than 10Km*) customers and NGO authorities. NGOs can buy food to distribute among needy peoples from the portal itself at lower price. Moreover, customers can donate food to nearby NGOs through this system and can even buy food for themselves. So, users can login as Restaurant Management authority, NGO authority and Customer. Restaurant authorities can add product item (Food) in their account and can also update their profile information. The other users i.e. NGO and Customer can only see nearby restaurants and available foods from that restaurant. They can also update their profile information.

1.5 References

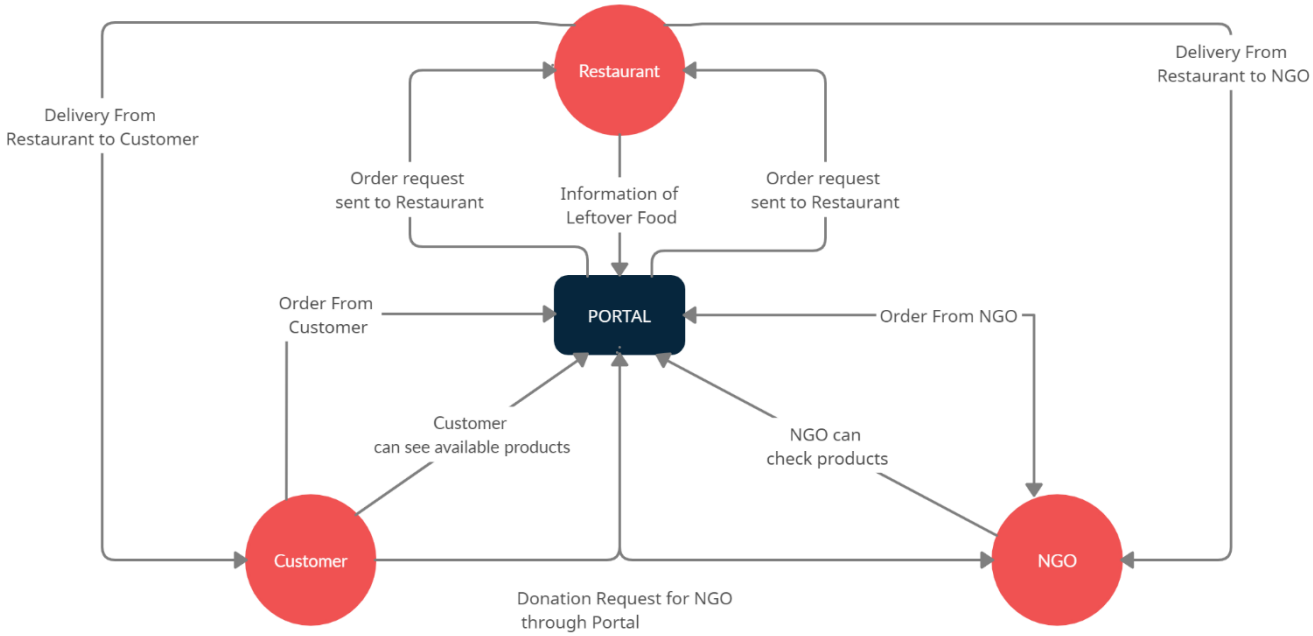
The material content of the following sites and YouTube lectures would be very helpful in the development process:

1. React Tutorial web
2. React Tutorial
3. Node Tutorial
4. MERN
5. MERN web
6. E-Commerce Website Guide

2. Overall Description

2.1 Product Perspective

It is a new self-contained product which aims to resolve the problems like hunger crisis, food wastage etc. in an organizing manner. Buying, selling of food items, donation to NGOs, everything can be done from the same system.



2.2 Product Functions

The Major Function of SSFDS are:

- **Food Tracking**
  - Creating or logging into account by the Restaurant
    - Name
    - Address (Using Google Map)
    - Contact Number
    - Email -id
    - Owner Name
    - Owner Email-id
    - Profile Photo
  - Enter the details of leftover food at a certain hour of the day
    - Name
    - Price per Plate

- Picture
- Amount of food left

- Reflection of the Food items
  - Restaurant Name
  - Restaurant Location
  - Food List
  - Food Details
  - Add to Cart
  - Remove from Cart
  - Place Order

➢ **Distribution or Sales**

- Distribution to NGO
  - NGO Sign In / Sign Up
  - Display of nearby restaurants' name
  - Discount upon placing order
  - Payment
  - Assigning delivery agent from NPO
- For Sales
  - Customer Sign In/ Sign Up
  - Display of nearby restaurants' name
  - Options :
    1. Donate to an NGO
       - Display nearby NGO names
       - Payment by customer
       - Assigning delivery agent from NPO
    2. Self-pick up
       - Only Prepaid
       - No Delivery Charge
    3. Home Delivery
       - Prepaid/ Cash on delivery

- Free delivery within certain distance
- Beyond limit a certain charge per Km
- Assigning delivery agent from NPO

2.3 User Classes and Characteristics

The various user classes using the product are:

I. Restaurants: Can create or delete account, edit profile, update information about leftover food.

II. NGO: Can create or delete account, edit profile, search nearby restaurants, place order.

III. Customer: Can create or delete account, edit profile, search nearby restaurants, place order, donate to an NGO.

2.4 Operating Environment

Updated Browsers and Internet connectivity is all that is required since this is a web-based project.

2.5 Design and Implementation Constraints

1) Since the application has to store all the past cases, if sufficient memory requirements are not available it might be a problem over a long period of time.
2) Login and password for user identification are used as means of security and authentication*.
3) The response time may vary slightly depending on the size of the database.
4) During online payment via portal, there might come issues
5) Problems can occur while using geolocation APIs to get user's location

2.6 User Documentation

User Documentation include:

1) A readme.txt file to set up the application and in brief, the using instructions.

2.7 Assumptions and Dependencies

1) It is hoped that the users enter correct details.
2) Delivery agents available in all user location and at all time
3) A good internet connection would ensure getting the information in a shorter time.
4) It is hoped that enough memory will be available to store all data.

3. External Interface Requirements

3.1 User Interfaces

The UI* will be an GUI* displayed in a browser have the following key functions:

1) A General Homepage
   I. NPO* details and a glimpse about their work
   II. Sign Up / Log in Page
      - Restaurant
        - Food item detailed list
        - Add item
        - Delete item
        - Pending Order Details
        - Edit Profile
        - Log out

      - NGO
        - Nearby restaurants list
        - Cart (Chosen food list)
        - Food list from a particular restaurant
        - Food details
        - Rebated price
        - Add to cart
        - Place order
        - Payment window
      - Customer
        - Nearby restaurants list
        - Cart (Chosen food list)
        - Food list from a particular restaurant
        - Food details
        - Rebated price
        - Add to cart
        - Place order
        - Donate to NGO
        - Payment window

   III. Developers Information
      - Contact
      - Email ID
      - GitHub
      - Facebook
      - Instagram
      - Twitter
      - LinkedIn

3.2 Hardware Interfaces

● Any modern PC or Laptop with Internet connection would suffice.
● Large database managing memory and resolution on server side.

3.3 Software Interfaces

Language Used:

Frontend: JavaScript (React JS)

Backend: Node.js Express.js Mongo DB (For database)

Operating System: Windows 10/11

IDE: VS Code

Database: Mongo DB

Server for hosting web page: TBD*

3.4 Communications Interfaces

Internet Protocol Used:

FTP and HTTP for uploading images, loading information from the database and for sending information for updates to the database.

Web Browser:

Google Chrome for Windows and all modern browsers where ES6 is supported

4. System Features

4.1 Restaurant Login

4.1.1 *Description and Priority*: (High Priority) Log in by providing correct username and password.

4.1.2 Stimulus/Response Sequences

Stimulus: Request to add new user
Response: System Provides a form to enter new user data

Stimulus: Request to edit Profile
Response: System provides an editable form alike interface

Stimulus: Request to add Food item
Response: System provide space to provide details of leftover Food items

Stimulus: Restaurant request to delete an item from list
Response: System deletes that item from the record list

Stimulus: Restaurant request to check pending orders
Response: System shows the list of pending orders

Stimulus: Request to show details of an order
Response: System provides the details of that particular order

Stimulus: Request to log out from the account
Response: System logs out and redirect to general homepage

4.1.3 Functional Requirements

Internet Connection:

Good Internet Connection required for fast services. It might take longer time to fetch data from database in slow connection.

Web Browser:

Recommended browser for this would be Google Chrome, as we will use google chrome during our development time. Other than Chrome, any browser which supports ES6 can run all these features.

4.2 Buyers Login

4.2.1 Login as NGO

4.2.1.1 *Description and Priority*: (Medium) Log in from an NGO authority by entering correct username and password

4.2.1.2 *Stimulus/Response Sequence*

Stimulus: Request to add new user
Response: System Provides a form to enter new user data

Stimulus: Request to edit Profile
Response: System provides an editable form alike interface

Stimulus: Request to show nearby Restaurants
Response: System shows list of nearby Restaurants

Stimulus: Request to show foods of nearby Restaurant
Response: System provides a detailed food list

Stimulus: Request to add food item to the cart
Response: System adds it along with calculating price

Stimulus: Request to increase the amount of the food item
Response: System increases it in a discrete manner

Stimulus: Request to see ordered food item list
Response: System shows the order list with total price

Stimulus: Request to place order

Response: System shows option for payment

Stimulus: Request to log out from the account
Response: System logs out and redirect to general homepage

4.2.1.3 *Functional Requirements*

Internet Connection:

Good Internet Connection required for fast services. It might take longer time to fetch data from database in slow connection.

Web Browser:

Recommended browser for this would be Google Chrome, as we will use google chrome during our development time. Other than Chrome, any browser which supports ES6 can run all these features.

4.2.2 Login as Customer

4.2.2.1 *Description and Priority*: (Medium) Login as a customer by entering correct username and password.

4.2.2.1 *Stimulus/Response Sequences*

Stimulus: Request to add new customer
Response: System Provides a form to enter new user data

Stimulus: Request to edit Profile
Response: System provides an editable form alike interface

Stimulus: Request to show nearby Restaurants
Response: System shows list of nearby Restaurants

Stimulus: Request to show foods of nearby Restaurant
Response: System provides a detailed food list

Stimulus: Request to add food item to the cart
Response: System adds it along with calculating price

Stimulus: Request to increase the amount of the food item
Response: System increases it in a discrete manner

Stimulus: Request to see ordered food item list
Response: System shows the order list with total price

Stimulus: Request to set the ordered food donate to an NGO

Response: System shows Online Payment option

Stimulus: Payment done
Response: System pops up a success message

Stimulus: Request to place order for own
Response: System shows option self-pickup/Home delivery

Stimulus: Customer chooses the option self-pickup
Response: System shows online payment window

Stimulus: Customer choose the option home deliver
Response: System shows pre/post paying option

Stimulus: Request to online payment done
Response: System pops up a success message

Stimulus: Request to log out from the account
Response: System logs out and redirect to general homepage

4.2.2.2 *Functional Requirements*

Internet Connection:

Good Internet Connection required for fast services. It might take longer time to fetch data from database in slow connection.

Web Browser:

Recommended browser for this would be Google Chrome, as we will use google chrome during our development time. Other than Chrome, any browser which supports ES6 can run all these features.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1) The system should be able to handle multiple logins (up to 800 to ensure negligible down-time)
2) The Database Management system should be public-domain
3) The server hosting the webpage should be efficient, and also the net connection needs to be good.

5.2 Safety Requirements

The data of past orders should be securely stored. All pending or being processed order information should not be lost. Account authentication details of user (Restaurants, NGO, Customer) should not change or be damaged unless done explicitly by them.

5.3 Security Requirements

1)The password of the registrar should be strong to prevent misuse of the System.

2) Any user should be authenticated to ensure no privacy breach.

3)The charge for searching cases should be private and well preserved to each lawyer account.

5.4 Software Quality Attributes

1.The software should be user-friendly.
2.The software should be reliable, testable, and maintainable.
3.The software should be efficient in terms of speed and security

5.5 Business Rule

Since this is a Non-Profit Organization, it would not charge any cost from the users. The major source of income of this organization is supposed to be donations from various philanthropies.

6. Other Requirements

Appendix A: Glossary

SSFDS: Social Serving Food Delivery System

UI: User Interface

GUI: Graphical User Interface

Authentication: The process of verifying if the user already has an account

NPO: Non-Profit Organization

NGO: Non-Governmental Organization

API: Application Programming Interface

TBD: To Be Determined

IDE: Integrated Development Environment

JS: Java Script

DB: Data Base

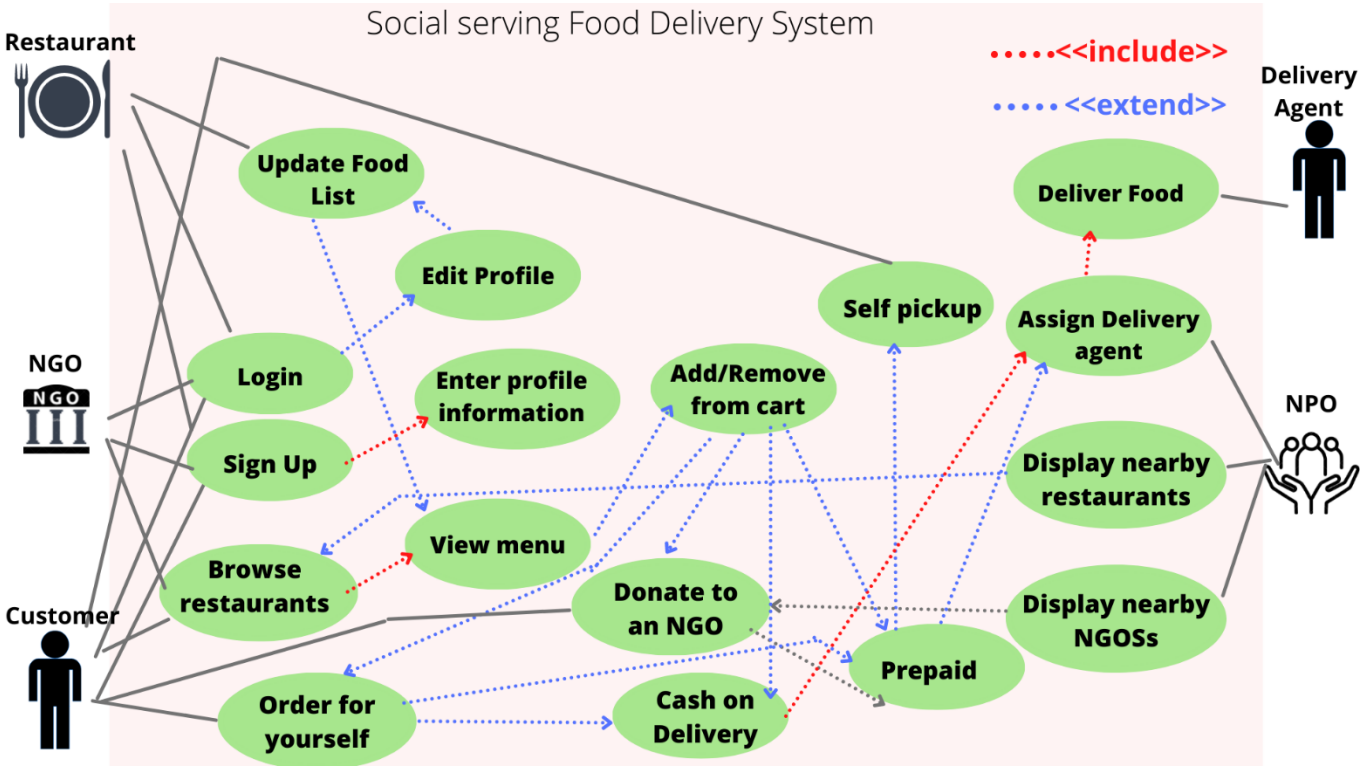
VS: Visual Studio

PC: Personal Computer

Km: Kilometer

Appendix B: Analysis Models

Use Case Diagram:



Appendix C: To Be Determined List

1. More details of User Interface
2. Hosting Server Details
3. Yet to decide whether to add forgot password facility