Software Requirements Specification

for

Food Delivery Service Management Software (FDSMS)

Version 1.0 approved

Prepared by:
Anish Sofat(19CS10011)
Aryan Agarwal(19CS30005)
Satvik Bansal(19CS10053)

IIT Kharagpur

20-Mar-2021

Table of Contents

Table of Contents 1

Revision History 2

Introduction 4

Purpose 4

Document Conventions 4

Intended Audience and Reading Suggestions 4

Product Scope 4

References 4

Overall Description 4

Product Perspective 4

Product Functions 5

User Classes and Characteristics 5

Operating Environment 5

Design and Implementation Constraints 5

User Documentation 5

Assumptions and Dependencies 6

External Interface Requirements 6

User Interfaces 6

Hardware Interfaces 7

Software Interfaces 7

Communications Interfaces 7

System Features 7

Register a Customer/Delivery Agent/Restaurant 7

Description 7

Stimulus/Response Sequences 7

Functional Requirements 7

Menu of a Restaurant 7

Creation and display of a restaurant menu 7

Description 7

Stimulus/Response Sequences 7

Functional Requirements 8

Order by a customer 8

See status of pending order 8

Description 8

Stimulus/Response Sequence 8

Functional Requirements 8

See previous completed orders 8

Description 8

Stimulus/Response Sequence 8

Functional Requirements 8

Promotional offers 8

Customer should be able to see promotional offers 8

Description 8

Stimulus/Response Sequence 9

Functional Requirements 9

Management should be able to provide promotional offer 9
Description 9
Stimulus/Response Sequence 9
Functional Requirements 9
Provide feedback 9
Description 9
Stimulus/Response Sequence 9
Functional Requirements 9
Restaurant 9
See available delivery agent 9
Description 9
Stimulus/Response Sequence 9
Functional Requirements 9
Set the available orders for pickup in an area 10
Description 10
Stimulus/Response Sequence 10
Functional Requirements 10
See pending orders 10
Description 10
Stimulus/Response Sequence 10
Functional Requirements 10
Delivery Agent 10
Mark their location 10
Description 10
Stimulus/Response Sequence 10
See Delivery Requests available in an area 10
Description 10
Stimulus/Response Sequence 10
Functional Requirements 10
Provide estimated pick-time from a restaurant and delivery time to a customer 11
Description 11
Stimulus/Response Sequence 11
Management 11
Provide Restaurant and Food Recommendations to Customers 11
Description 11
Stimulus/Response Sequence 11
Can manage list of customers, restaurants and delivery agents 11
Description 11
Stimulus/Response Sequence 11
Functional Requirements 11

Other Nonfunctional Requirements 11

Performance Requirements 11
Safety Requirements 12
Software Quality Attributes 12
Business Rules 12

Revision History

Name	Date	Reason For Changes	Version
Creation	20th March	Template changes and additions	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Food Delivery Service Management system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, and the constraints under which it must operate. The application is designed to take care of different types of users, which consists of restaurants, delivery agents and customers and provide them with services like selecting the menu items and location for delivery, location of the delivery agent and reject and accept orders for the restaurant. This also includes some quality of life features such as offers, and rating of customers, delivery agents and restaurants. It is useful to the people who cannot go out of their homes due to safety reasons or lack of resources and also a benefit for the restaurants to increase their reach.

1.2 Document Conventions

This Software Requirement Specification Document has been written using Free writing tools such as Google Docs typed in Arial font for normal text and Times Roman for the headings. The font size used is 11 for text and 14, 18 for headings. All headings are highlighted appropriately in bold. The document is prepared using UK English convention.

1.3 Intended Audience and Reading Suggestions

This document lists all 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 2

1.4 Product Scope

FDSMS is a web-based application that is intended to be used by individuals who want to order food online for home-delivery, restaurants for providing their services to the customers online, and registration of local people as delivery agents. This application will be based on the client-server model where the users can interact with the interface that is served by the frontend and the backend server manages the data via an online database and interacts with other clients for the data of locations and order contents and status.

1.5 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.

2. Overall Description

2.1 Product Perspective

It is a new self-contained developing product which originated from the idea that people are unable to visit restaurants because it is sometimes time consuming, out of their comfort-zone, and unavailability of a vehicle to visit one. This project tackles this program providing food to customers from famous restaurants in a safe manner, providing attractive offers and minimal delivery charges.

2.2 Product Functions

The key features of FDSMS are as follows:

- Provide food delivery to the customers directly from the restaurants.
- Delivery from a third party delivery person.
- Maintain the menu for the restaurants, and provide the final bill at the time of the order.
- Tell the estimated time for the delivery, from the restaurant as well as the delivery agent.
- Will provide the past information about the orders.
- Manage the rating for the customers, the restaurants and the delivery agents. And at the same time provide offers to some good customers.

2.3 User Classes and Characteristics

- Restaurant
    - Create the menu for all the items.
    - Accept/reject the status of the received orders, and provide the status of the accepted orders. Assign the estimated time of service.
    - Assign the delivery agent.
    - Mark the orders as “Out for Delivery”.
- Customer
    - Person ordering the food from the restaurants.
    - Check the menu of different restaurants, and order food if available.
    - The status of the order is updated based on the restaurant and the delivery agent.
    - Can rate the restaurant and the delivery person based on the quality of the service.
- Delivery Agent
    - They can mark their location for orders, and see the delivery requests from the restaurants.
    - Will be able to accept delivery requests, and set the estimated time for delivery, and update the order status.
- Management
    - Will manage the status for the customers, restaurants, and the delivery agents and can overlook their ratings.
    - Provide the recommendation of food and restaurants to the customers.
    - Provide offers to some of the good customers.

2.4 Operating Environment

The application is based on a client-server model with the users interacting as clients with a GUI web application. GUI is made with the help of HTML, CSS and JS with few free libraries from google for front-end. The backend and the server side are based on the flask framework of python which interacts with an online firebase database.

2.5 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.

2.6 User Documentation

The usage of the application is pretty intuitive, but there will be FAQ’s provided but still if there are any new questions, a form will be added for the queries. For developers and other technical users, all API’s, classes and methods will be documented and presented online.

2.7 Assumptions and Dependencies

We have assumed that the user has a working internet connection and an email-id for accessing the project.
For dependencies, we have used google's material.io for bootstrapping and google's firebase for an online database. Rest dependencies will be given in requirements.txt in the project.

3. External Interface Requirements

3.1 User Interfaces

The product will consist of a web-application with which the user will interact. The customer, restaurants, delivery agents and management have different dashboards based on their functionalities. And will be accessible by using their login credentials leading them to their user dashboards.

3.1.1 Homescreen
- Login
  - Username
  - Password
  - Redirection to respective dashboard
- Signup

3.1.2 Management
- Personal Details
- Lists of users(Customers, Restaurants and Delivery agent)
  - Details
  - Ratings
  - Recommendations(Restaurant List)
  - Promotions(Customer)

3.1.3 Delivery Agent
- Personal Details
- Mark the current Location
- Order lists available for pickup
- Details of the Selected Order
  - Update the status
  - Update the time required
  - Feedback(after delivery)

3.1.4 Restaurant
- Personal Details
- Menu
  - Create the menu
  - Menu details
- Present Orders
  - Update the status of the orders.
  - Send delivery requests
    - See available Delivery Agents
- Past Orders

3.1.5 Customer
- Personal Details
- Restaurant list
  - Recommended list
- Menu
  - Recommended items list
  - Create the order of items
  - Promotional Orders
- Present Orders
  - Status of Orders
  - Feedback

- Past Orders

3.2 Hardware Interfaces

The backend server will require a good processing unit to handle the requests. There are no specific hardware requirements for the user, but use on a desktop screen of around 13 inches or bigger is recommended. A decent connectivity to the internet is required for a good experience.

3.3 Software Interfaces

The project will connect to material.io for bootstrapping and firebase as a database. The project is a web app and thus just requires a working internet connection and a web browser and is operating system independent. The project uses python's flask library in the server side and simple javascript, html and css in the client side. All the data coming from the client will be stored in an online firebase database via the flask-server.

3.4 Communications Interfaces

All the communications will be done via the web-browser with the standard HTTPS protocol.

4. System Features

4.1 Register a Customer/Delivery Agent/Restaurant

4.1.1 Description

- Getting on the platform requires registering and logging on to the specific profile. Different profiles have different features. This task is essential as it provides the use class needed for all other activities.

4.1.2 Stimulus/Response Sequences

- The user will be first asked the profile that they want to register, and then specific information according to each profile will be asked from them, and then will be stored in their respective classes.

4.1.3 Functional Requirements

- Signup: As soon as the user enters they will see a general home page and there will be a link to go to the signup page. There the user will choose the profiles and then enter the details in the respective boxes. After the details are authenticated/stored the user will be directed to the respective user page.
- Login: Users will enter the credentials, created during the signup process and then after authentication, they will be redirected to the respective user page.

4.2 Menu of a Restaurant
Creation and display of a restaurant menu

4.2.1 Description

- At the time of Signup the restaurant will not be asked to create the menu but after the redirection. They will be allowed to add items, delete items and change the details of the menu item. The menu will be displayed by request from the restaurant or a customer.

4.2.2 Stimulus/Response Sequences

- The contents of the menu will be shown in table form.
- The restaurant can add/delete menu items and at the same time change the details of the current items by clicking on the add or delete button and the edit button in front of each item respectively.

- The customer can select the items from the menu and put them in the order list by selecting the add button. This will put the item in the order list.

4.2.3 Functional Requirements

- This requires the use of some common front end JavaScript to add the items as a table on the page, and using the database to store the orders.

4.3 Order by a customer
See status of pending order

4.3.1 Description

- The customers can check the order status for the order placed by them. The restaurant can either accept or reject that order and update the status from time to time. The delivery agent will be able to update the order.
- The status will contain the following steps: Accepted/Rejected, Preparing food, Food Prepared, Handed out for delivery, Delivery Complete.

4.3.2 Stimulus/Response Sequence

- The status will be updated by the restaurants and the delivery agent manually, and the updates will have time stamps.
- The status will appear as a timeline with checkpoints as the steps and the paths/connection showing the average time required for the next step which will be provided by the restaurant or the delivery agent.

4.3.3 Functional Requirements

- The restaurants and the delivery agent will be updating the statuses manually, and the estimated time will also be given by them. The time of update will be received using the datetime library and the updates will be shown on the frontend.

See previous completed orders

4.3.4 Description

- The orders will be shown to the customers that they have done in the past. Restaurants can also check all the past orders that have been done in their restaurants.

4.3.5 Stimulus/Response Sequence

- The customer can check the list of past orders by pressing the button on the page, and the orders will be shown in a list.
- The restaurant will have a similar section on their panel.

4.3.6 Functional Requirements

- The information about the order will be stored in the database with restaurantId and customerId to identify the customer that ordered that and from the restaurant. The respective user will have a list of orderId in the class to access the list easily. The list will be displayed using css and javascript.

4.4 Promotional offers
Customer should be able to see promotional offers

4.4.1 Description

- The customers will get promotional offers from management to get discounts on orders from the management if they have a good rating. They can apply that on their orders, but can be used only once.

4.4.2 Stimulus/Response Sequence

- There will be a section of offers to check different offers in the dashboard for the customer. To apply the offer the customer will choose the offer on the order page, the offer will be applied and removed from the list if the order is placed.

4.4.3 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.
- If the offer is used, it will be removed from the user by deleting that Id from the list.

Management should be able to provide promotional offer

4.4.4 Description

- The management will provide the offers to the customers if they find that they have a good amount of ratings. They can create new offers or give some old created offers to the customers.

4.4.5 Stimulus/Response Sequence

- 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 the offerId that will be

4.4.6 Functional Requirements

- Simple database in the backend and JavaScript in the front end can serve the purpose.

4.5 Provide feedback

4.5.1 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.

4.5.2 Stimulus/Response Sequence

- In each present order there will be a section to give the feedback, the feedback will consist of star rating. The feedback will be sent to the management and will be stored. A customer can give the feedback only once per order.
- 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.

4.5.3 Functional Requirements

- The function will be requiring a box containing stars and a css javascript system to take the feedback, it will be stored in the database.

4.6 Restaurant

See available delivery agent

4.6.1 Description

- As the delivery agent marks his/her availability for delivery, there will be a list that shows that data to the restaurant and then sends the delivery requests in the area.

4.6.2 Stimulus/Response Sequence

- While sending the request for delivery by the restaurant, it can see the delivery agents as a list and then send the request for the delivery.

4.6.3 Functional Requirements

- Same as “Set the available orders for pickup in an area” functionality.

Set the available orders for pickup in an area

4.6.4 Description

- When the order is ready or is about to be ready, the restaurant will update the status of that order and then put up a notification for pickup in the area. The delivery agent will be able to choose the order.

4.6.5 Stimulus/Response Sequence

- 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.

4.6.6 Functional Requirements

- The order selected to be made available will be added in the database as available and will be shown in the list of the delivery agents in that using the database and JavaScripts.

See pending orders

4.6.7 Description

- The restaurant will be able to see the orders list in the queue in the form of a table, it will show the status of the order and if the order is rejected, it will be placed in the past order list.

4.6.8 Stimulus/Response Sequence

- All the present orders will be shown in the form of a list and the restaurant can click on them to set their status. The list can be accessed by using the orders button in the panel.

4.6.9 Functional Requirements

- The function will be implemented using JavaScript and CSS for the frontend and the backend using the database to store the data and Flask to take care of the methods.

4.7 Delivery Agent

Mark their location

4.7.1 Description

- Delivery agents can set their current location as the area that they are in at the moment.[1]

4.7.2 Stimulus/Response Sequence

- The delivery agent can have a dropdown of the areas, and can select the area in which they want to work.

See Delivery Requests available in an area

4.7.3 Description

- When the delivery requests are posted by the restaurants, they show up in a list to all the delivery agents in the area and they can accept according to their will.

4.7.4 Stimulus/Response Sequence

- 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.

4.7.5 Functional Requirements

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

[1] The current area definition is broad and hence this functionality does not have much of a use. But in later development, various map api's can be used in order to get/set accurate information.

Provide estimated pick-time from a restaurant and delivery time to a customer

4.7.6 Description

- This will be estimated time according to the delivery agent and hence can be executed by using text boxes. The time will be updated in the order status and hence intimating the customer.

4.7.7 Stimulus/Response Sequence

- 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.

4.8 Management

Provide Restaurant and Food Recommendations to Customers

4.8.1 Description

- Based on the rating, the management can recommend food and restaurants to all the customers. The restaurants recommended will be shown to all and are not personalized to all.

4.8.2 Stimulus/Response Sequence

- 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.

Can manage list of customers, restaurants and delivery agents

4.8.3 Description

- The list of customers, restaurants and delivery agents can be seen in different tabs and there they can see the details of the user. Management can push offers to them, if they are customers, or recommend the restaurants and food items.

4.8.4 Stimulus/Response Sequence

- The listing functionality is the extension of the above used functionalities of showing the details. There will be options to push offers and recommend foods, by clicking the buttons or selecting the checklist buttons.

4.8.5 Functional Requirements

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

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The user should be able to query databases quickly and the results fetched must be appropriate. This can be done by finding the right balance between performance and accuracy by using Firebase.

5.2 Safety Requirements

The application runs on the web browser and hence harm to the user device is minimal, whereas a lot of data is to be read and written and hence the data storage and the server damage is possible but that too during heavy usage.

5.3 Software Quality Attributes

- Maintainability:
    - Different versions of the product should be easy to maintain. For development it should be easy to add code to an existing system, and it should be easy to upgrade for new features and new technologies from time to time. Maintenance should be cost effective and easy.
- Usability:
    - This can be measured in terms of ease of use. Application should be user friendly. Should be easy to learn. Navigation should be simple.
- 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.

5.4 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.