

Requirements Specification Document
Food Recovery Network at LMU
Food Justice Website for LMU's FRN Chapter

Table of Contents

Requirements Specification Document.....	1
LMU's FRN Website.....	1
1.0 Requirements Specification.....	1
1.1 Introduction.....	1
1.2 Functional Requirements by CSC.....	1
1.2.1 User Interface.....	1
1.2.2 Front-end Development.....	1
1.2.3 Back-end Development.....	1-2
1.2.4 Database.....	2
1.2.5 Data Processing.....	2
1.3 Performance Requirements by CSC.....	2
1.3.1 APIs Fetch Information in 3 - 5 seconds.....	2
1.3.2 Display Delivery Route Within 10 seconds.....	3
1.3.3 User Information and Preferences Retrieved at Log-in.....	3
1.4 Project Environment Requirements.....	3

1.0 Requirements Specification

1.1 Introduction:

This Software Requirements Specification (SRS) outlines the requirements, applications, and software required for LMU's FRN, a website to help anyone involved in the Food Recovery Network (FRN) program at Loyola Marymount University.

LMU's FRN website is a JavaScript- and React-based application that provides a user with information on what the FRN is, how it works at LMU, and how they can get involved. Upon entry into the app, users will be able to view the schedule for the current semester's weekly pickups and deliveries in addition to being able to input the amount of food recovered for a specific week. Users may also create an account where they can sign up to volunteer or find other ways to get involved with the program.

1.2 Functional Requirements by CSC:

1.2.1 User Interface (UI)

1.2.1.1 The UI subsystem shall be displayed in a web page.

1.2.1.2 The UI subsystem shall react to mouse clicks on all of the displayed buttons and tabs.

1.2.1.3 The UI subsystem shall incorporate a menu bar with the different pages at the top of the main window of the website.

1.2.1.4 The UI subsystem shall supply users text entry fields for which they may type values.

1.2.1.5 The UI subsystem shall check for errors on the text entry fields so that the user is provided with feedback in the case of an invalid entry.

1.2.2 Front-end Development

1.2.2.1 The front-end development subsystem shall utilize normal Hypertext Markup Language (HTML) for styling.

1.2.2.2 The front-end development subsystem shall utilize Cascading Style Sheets (CSS) for styling.

1.2.2.3 The front-end development subsystem shall utilize JavaScript (JS) to create the views of the website that users will interact with.

1.2.2.4 The front-end development subsystem shall utilize React.js for the functionality of the parts that the users will interact with.

1.2.3 Back-end Development

- 1.2.3.1 The back-end development subsystem shall use Node.js.
- 1.2.3.2 The back-end development subsystem shall utilize the API from Google Maps in order to depict the route from the user's start destination where they pick up the food and end destination where they deliver the recovered food.
- 1.2.3.3 The back-end development subsystem will update and show different routes if more placements are added in the future with the use of the Google Maps API.
- 1.2.3.4 The back-end development subsystem shall use the API from Twilio to send users SMS updates, notifications, and reminders with any important FRN information.

1.2.4 Database

- 1.2.4.1 The database shall be created and developed using Google's Firebase.
- 1.2.4.2 The database shall store user information, such as their email and password.
- 1.2.4.3 The database shall allow for dynamic manipulation of user information.
- 1.2.4.4 The database shall be created in a manner that protects sensitive user information.
- 1.2.4.5 The database shall store information regarding the user's history, such as previous volunteering dates or the amount of food tracked on a specific day.
- 1.2.4.6 The database shall store meta-data regarding app usage, allowing the developers to track user trends.

1.2.5 Data Processing

- 1.2.5.1 Data processing shall be able to calculate an efficient route and facilitate scheduling for food pickups and drop-offs.
- 1.2.5.2 Data processing shall be used for tracking and updating food donations, including the type of foods and the quantities.
- 1.2.5.3 Data processing shall be used to process date and time data in relation to optimize food pickup and delivery schedules.
- 1.2.5.4 Data processing shall handle constraints like volunteer availability and transportation capacity.

1.3 Performance Requirements by CSC:

1.3.1 APIs Fetch Information in 3 - 5 seconds

- For APIs used to track weekly food donations, schedule pickups, and coordinate volunteers, results shall be found and displayed within 5 seconds.
- Volunteer opportunities will be fetched within 3 seconds and sorted by proximity and urgency.

1.3.2 Display Delivery Route within 10 seconds

- The application should take 3 - 5 seconds to completely fetch all information from APIs, outlined in 1.4.1.
- After fetching and sorting all required information, the application will load and display images from APIs, display the calculated route, and display the volunteer schedules.
- This process is expected to take 10 or less seconds to complete.

1.3.3 User Information and Preferences Retrieved at Log-in

- Once a user is logged in to their account, their user information including email, name, role (volunteer, Sodexo worker St. Joseph's Center staff), and previous pickup/delivery history shall be retrieved automatically from the database.
- A user's preferences including preferred dates and notification settings shall be stored on the user's profile and retrieved when the user logs in.

1.4 Project Environment Requirements:

Category	Requirement
Hardware	<ul style="list-style-type: none">● LMU's FRN website shall not require any special computing hardware to operate.
Software	<ul style="list-style-type: none">● LMU's FRN website shall be able to run using any standard web browser.● LMU's FRN website shall be compatible with most recent versions of web browsers.