Architectural Design

Table of Contents

6.1 Introduction

This document presents the architecture and detailed design for the software for the Food Recovery Network (LMU chapter) website. The web application provides detailed route specifications given a start and end location from where the food is picked up to where it is dropped off. This software will also allow users to view the schedule for the current semester, sign up to volunteer, and track the amount of food that was picked up / delivered on a specific date.

6.1.1 System Objectives

The objective of this web application is to provide an intuitive and visually appealing user interface that helps the Food Recovery Network program at LMU run smoothly. By condensing details for this volunteer opportunity into a straightforward design, this website allows users to easily view scheduled deliveries, see their route, update how much food was donated, and learn how to get involved. Volunteers will also be able to receive SMS or email updates and reminders.

6.1.2 Hardware, Software, and Human Interfaces

In this section, I will detail the technical requirements for my senior project. This includes hardware, software, and human interfaces that will be implemented.

6.1.2.1 Server Requirements

Hosting this web application traditionally requires access to a cloud server. Given the scope and time-constraint of this project, I will not actually be hosting the website.

6.1.2.2 Network

The user of this application must have secure access to the internet. This is an assumed requirement and will not require any actual integration for this project. Internet access is required for API calls and access to the database.

6.1.2.3 Frontend Technologies

To manage and build the web interface, I will utilize HTML, CSS, and JavaScript. In addition, React will be used for dynamic rendering and building a responsive application. Finally, the map visualization will be integrated using Google Maps API. This API will present a map, including the route specifications.

6.1.2.4 Backend Technologies

To manage communication between the interface and server logic, JavaScript, specifically node.js, will be used. Node.js is chosen for its flexibility with the

software packages that I am integrating into this project. For route information, I will first gather users' information from Firebase. Then, any additional information that will be provided to the user will be generated using the Google Maps API. For users to receive SMS updates and reminders, I will utilize the Twilio API.

6.1.2.5 Database

To store user data, such as username and password, volunteer date preferences, and UI preferences, I will be using Firebase. Since this web app will not be hosted for the time being, I do not anticipate a large user base. Therefore, Firebase's free and simple design is optimal for this application.

6.2 Architectural Design

This section provides the overall design of my web application.

6.2.1 Major Software Components

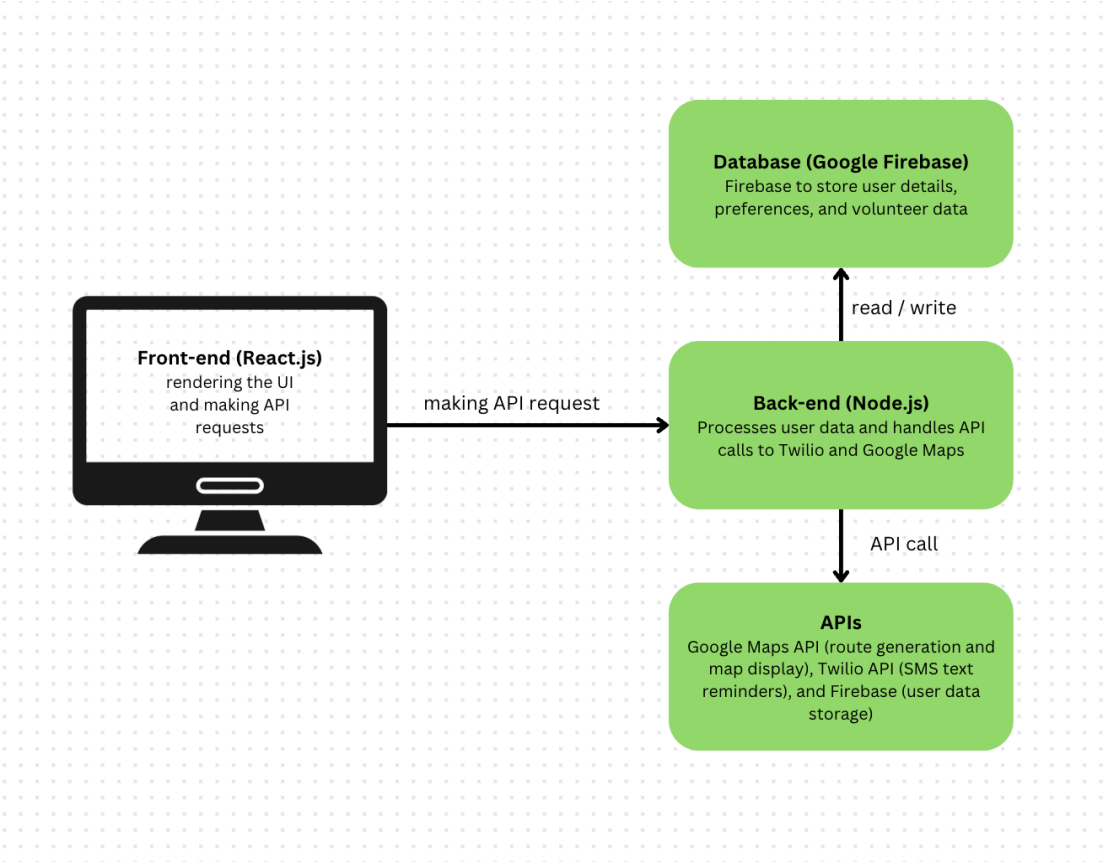
This web application will use a mix of both front-end and back-end software components. For the front-end, React.js will be used with the languages JavaScript, HTML, and CSS. For the back-end, the Node.js framework will be utilized. To store user data, the application will use Google Firebase. In addition, the application will make use of a variety of APIs to load data and generate appropriate information for the user, including generating delivery routes and sending SMS text updates.

6.2.2 Major Software Interactions

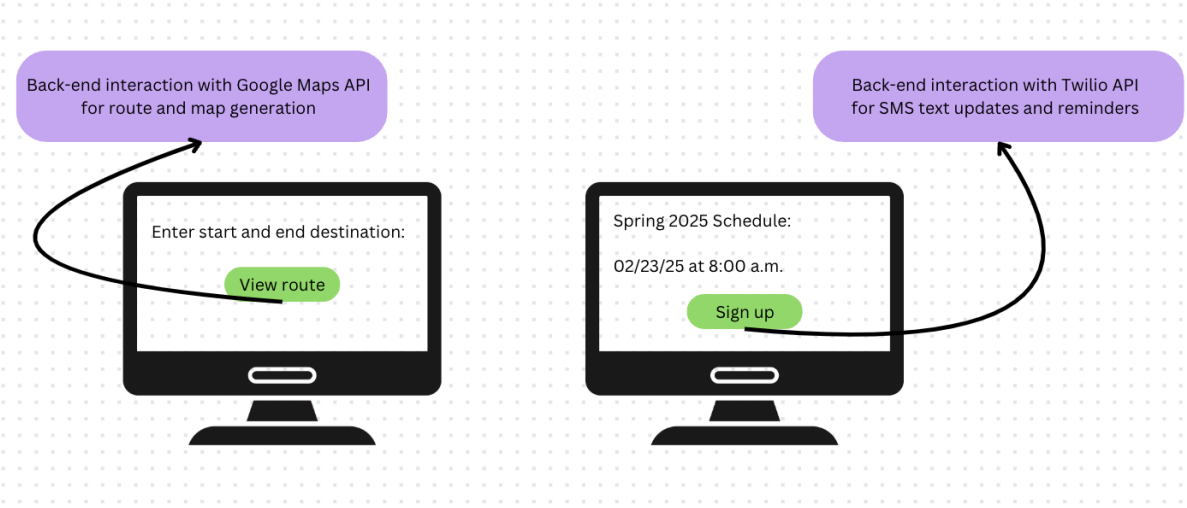
Some of the major software interactions for the LMU Food Recovery Network website include user interface interactions and front-end to back-end API calls. The user interface gets the user's food tracking details or sign up information, sends the request to the back-end, and stores the information while updating the front-end based on the user's inputs. Once the back-end receives the information from the front-end, it also makes an API call with the Google Maps API and outputs a map with the user's journey from their start location at LMU to their end destination at St. Joseph's Center. Another major software interaction involves handling errors and feedback where the back-end will log any errors for troubleshooting.

6.2.3 Architectural Design Diagrams

Component Diagram



Use Case Diagram



Sequence Diagram