Software Requirements Specification

Version 1.0

Prepared by

Georgia Gwinnett College

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this system is to improve upon the Satisfeed dashboard by implementing more sophisticated features that allow users to view school information on an interactive map in relation to hotspots of past customers.

1.2 Project Scope

The scope of this project includes improvements to the current dashboard map through the addition of school data and hotspot locations. It will also include implementation of an API that serves as a point of contact between the front-end system and database that holds customer / school data, retrieved from outside sources. This project will focus on parsing school data and storing said data to later be used on the front-end map. It will also include adding ways to store customer data in a way that is useful to the client in terms of locating hotspots around schools in need. This project will focus on improving the usability of data with a more robust API and database; it will not include redesigning of the overall dashboard UI nor refactoring of current dashboard logic unless necessary to meet project goals.

2. Overall Description

2.1 Product Features

The Satisfeed Dashboard is a Next.js web application that generates dynamic reports from Satisfeed's customer database. Satisfeed is a Gwinnett-based nonprofit food bank operating from a single distribution site. Guest demographic and service-usage data is collected during signups and stored in a structured database. The proposed version of this dashboard queries this database and renders reports in real time, providing stakeholders with up-to-date visibility into client demographics and program utilization.

2.2 Operating Environment

- **Hardware**: Any PC, Mac, or mobile device
- **Operating System**: Windows 10, macOS 10.15+
- **Software Components**:
    - React
    - Next.js
- **Development Environment:** IntelliJ, VSCode, Development Server
- **Production Environment:** Vercel

2.3 Design and Implementation Constraints

- Must use Next.js framework

- Must use Vercel for frontend deployment

2.4 Assumptions and Dependencies

- The project depends on the guest information database, hosted by Oasis, being accessible for integration. An API will be developed to receive data from this database and update the dashboard at regular intervals.
- The project assumes the guest sign up form, hosted by Oasis, will later add mandatory home address/school attendance input in order to populate the dashboard created in this system with relevant user data
- The project assumes that data needs to be stored within a database due to the scale of both customers and school information that needs to be reliably accessible.

2.5 Functional Requirements

REQ-1: Dashboard shall include a "Food Distribution by School Zone" option in the dropdown menu, alongside the existing "Food Distribution by County" and "Food Distribution by ZIP Code."
- **Priority:** High
- **Justification:** The inclusion of school zone data will allow Satisfeed administrators to partner with schools that have a high number of students in continual need of Satisfeed's services.

REQ-2: Dashboard shall include school zones for each elementary school in Gwinnett County
- **Priority:** High
- **Justification:** Provides school-level insights to allow for stronger community partnerships and resource allocation.

REQ-3: The dashboard map shall have a live icon to track the location of the currently logged in user, loading up the data for the location they're in.
- **Priority:** Medium
- **Justification:** Improves ease of use during mobile usage

REQ-4: School Data Import and Storage
REQ-5: **Priority:** High
REQ-6: **Justification:** System retrieves school data from external sources, parse it, and store it in the database for later use

REQ-7: Customer Data Management
REQ-8: Priority
REQ-9: System shall provide a way to store and update customer data relevant to locating hotspots
REQ-10: API for Data Access
REQ-11: Priority
REQ-12: system shall implement an API that allows the front-end to access school and customer data for visualization on the map
REQ-13: Data Filtering for Hotspots
REQ-14: Priority
REQ-15: system shall allow filtering and retrieval of schools near hotspot areas to support client analysis

3. Nonfunctional Requirements

3.1 Performance Requirements

4. Domain Requirements