Food Forward Tracker

Team: LynxDB

Sabona Abubeker, Kainoa Borges, Angela McNeese

Software Requirements Specification Document

**Version: (1.0) Date: (09/21/2022)**

**Table of Contents**

1. [**Introduction** **5**](#_gjdgxs)
   1. [*Purpose* *5*](#_30j0zll)
   2. [*Scope* *5*](#_1fob9te)
   3. [*Definitions, Acronyms, and Abbreviations* *5*](#_3znysh7)
   4. [*References* *5*](#_2et92p0)
   5. [*Overview*](#_tyjcwt)

*5*

1. [**The Overall Description** **6**](#_3dy6vkm)
   1. [*Product Perspective* *6*](#_1t3h5sf)
      1. [System Interfaces 6](#_4d34og8)
      2. [Interfaces 7](#_2s8eyo1)
      3. [Hardware Interfaces 7](#_17dp8vu)
      4. [Software Interfaces 7](#_3rdcrjn)
      5. [Communications Interfaces 8](#_26in1rg)
      6. [Memory Constraints 8](#_lnxbz9)
      7. [Operations 8](#_35nkun2)
      8. [Site Adaptation Requirements 8](#_1ksv4uv)
   2. [*Product Functions*](#_44sinio)

2.2.1 - Households

2.2.2 - Menu Items

2.2.3 -Inventory

2.2.6 - Reports [*9*](#_44sinio)

* 1. [*User Characteristics* *9*](#_2jxsxqh)
  2. [*Constraints* *10*](#_z337ya)
  3. [*Assumptions and Dependencies* *10*](#_3j2qqm3)
  4. [*Apportioning of Requirements*](#_1y810tw)

2.6.1 - Security

2.6.2 - Jobs

2.6.3 - Payment

*10*

1. [**Specific Requirements** **10**](#_4i7ojhp)
   1. [*External interfaces* *12*](#_2xcytpi)

3.1.1 - Login Screen

3.1.1. 2 - Login Container

3.1.2 - Universal Navbar

3.1.3 - Account Page

3.1.3.1 - Admin Page

3.1.4 - Reportt Page

3.1.4.1 - Overview Report

3.1.4.2 - Meal Report

3.1.4.3 - Packaging Report

3.1.5 - Input Page

* 1. [*Functions*](#_1ci93xb)
     1. - “Entry” Definition
     2. - New Entry to SQL
     3. - Execute Query
     4. - Authenticate User
     5. - Ingredients Form Processing
     6. - Meal Form Processing
     7. - Inventory Form Processing
     8. - Household Form Processing
     9. - Unit Conversion
     10. - Calculate Servings Required
     11. - Calculate Quantity Required
     12. - Calculate Grand Total
     13. - Meal Plan Report
     14. - Report Generation  [*12*](#_1ci93xb)
  2. [*Performance Requirements* *13*](#_3whwml4)
  3. [*Logical Database Requirements* *13*](#_2bn6wsx)
  4. [*Design Constraints* *14*](#_qsh70q)
     1. [Standards Compliance 14](#_3as4poj)
  5. [*Software System Attributes* *14*](#_1pxezwc)
     1. [Reliability 14](#_49x2ik5)
     2. [Availability 14](#_2p2csry)
     3. [Security 15](#_147n2zr)
     4. [Maintainability 15](#_3o7alnk)
     5. [Portability 15](#_23ckvvd)
  6. [*Additional Comments*](#_4f1mdlm)

*14*

1. [**Change Management Process**](#_2u6wntf) **14**
2. [**Document Approvals**](#_19c6y18) **14**
3. [**Supporting Information**](#_3tbugp1)*15*
4. **Changes** [*1*](#_3tbugp1)*5*

# Introduction

Food Forward is a nonprofit organization working to assist people living in food deserts with access to nutritious food and meals. Still in their initial setup stages they would like an application produced which can help more efficiently track their costs and manage their inventory. This project will produce a working, finished application to serve their needs.

# Purpose

The purpose of this project is to build a database to assist the Food Forward initiative. It will provide a simple way for employees and volunteers of the nonprofit to track customers, expenses and inventory, minimizing their resource costs in order to maximize the number of households they are able to supply with nutritious meals and snacks.

# Scope

### This project will produce an application called the Food Forward Tracker.

### The Food Forward Tracker will provide an interface for a database that tracks inventory, costs and customers in order to streamline Food Forward’s efficiency and minimize their costs as they serve their customers.

### Food Forward will use this tracker to help minimize costs and streamline their process. Knowing at a glance how much of what items are in stock and having an instantly generated shopping list will save both time and expense for their business. In order to do this the application must track how many households they are serving and how much food per week will need to be prepared in addition to the costs of ingredients and packaging for each meal they provide. It also tracks the number of children in each household and snacks needed in each household. They have a goal to expand to serve up to 30,000 individuals known to live within food deserts in their region and the goal is to streamline operations enough to accomplish this with minimum costs, minimal time and as few resources as possible in order to be sustainable. The lower the costs involved, the more households they will be able to serve. In the future the non-profit also plans to include a mobile application for their customer households, further reducing manpower requirements on their end if enough customer buy-in can be achieved.

# Definitions, Acronyms, and Abbreviations.

API - Application Programming Interface

DB - Database

FF - Food Forward

GUI - Graphical User Interface

HS- Household size

I - Inventory

LBS - pounds

MPC - Meal Plan & Cost

Oz - ounces

PS - Portion size

Qrtz - quartz

R - Recipes

RAM - Random Access Memory

TBSP - tablespoon

TSP - teaspoon

UI - User interface

WM - Weekly meals

CPU - cost per unit

IN - inches

CM - centimeters

S - Snacks

# References

### RFC 9110 - Internet Standard <https://www.rfc-editor.org/rfc/rfc9110> (2022)

# Overview

The main body of the Food Forward application will work as a pillar to support other extensions, including the central database, an inventory management tool, seasonal menu forecast, household tracking, and raw goods cost estimates. It will be organized into preparing menus, pricings for each item in the inventory, and standardizing units for CPU between Meal plan, cost tracking, and shopping inventory tabs.   
 For a quick outline of how this will be achieved see section 2: *The Overall Description* provides an outline of the entirety of the project in a high level, non-technical way. For the more technical details see section 3: *Specific Requirements* where a more detailed, in depth description is provided.

# The Overall Description

Food Forward is a local initiative focusing on serving urban populations affected by food deserts where residents are unable to easily obtain fresh, healthy food. The Food Forward initiative is already underway and providing meals to a select group of households, but the goal is to develop a working model able to expand into serving the affected residents of the Duluth area, totalling approximately 25k to 30k people.   
 At present, Food Forward provides one pre-selected meal and 5 snacks per household per week. They require an easy to use tracking database to improve their efficiency and track their progress. The application will need to maintain an inventory of ingredients and supplies on hand, as well as what meals and snacks they offer to include portion sizes, costs, and other various details that go into ensuring this non-profit operation runs smoothly and sustainably.

# Product Perspective

# Currently the approach for this database is as an independent project with no larger system in mind for integration however, Food Forward does intend to branch out and is considering a customer facing mobile app in the future, therefore some consideration will go towards ensuring that transition is as simple as possible. Other inventory management solutions do exist but a cursory search did not find any that would satisfy this need. Existing free options are primarily retail based and restaurant focused solutions that would fit the food service nature of this project do not support the kind of rapid calculation of materials and costs required to serve a known-quantity of households, which is Food Forwards primary concern.

### System Interfaces The application has no external system requirements.

### Interfaces The application will use a graphical user interface (GUI) as it’s primary interface, if possible it will leave room to easily expand to a mobile application in the future. Users should be able to easily input and track menu items (recipes), snacks combinations, ingredients, and households and output custom reports as intuitively as possible. This product is targeted for employees of a non-profit expected to be familiar with average computer use. ADA interface is not a priority but it would be beneficial to include options like text-to-speech integration wherever possible.

* + - 1. Login Screen: The login screen allows the user to input or register their username and password to access the software.
      2. Username/Password Reset
      3. Menu Item Access: Provides an authorized user the ability to enter or edit recipes and create snack combinations.
* Provide authorized users the ability to enter or edit packaging supplies needed for a menu for single servings.
  + - 1. Menu Selection Screen: One quick overview where users can select the menu items which will be prepared that week.
      2. Ingredient Tracking: Provides authorized users that ability to quickly and simply add or edit ingredient cost, storage needs etc.
      3. Household Tracking: Provides authorized users a way to input or modify the necessary information regarding each household such as the number of contact information, number of adults, number of children, special dietary needs, etc.
      4. Report Generation: Allow the user to select date and data ranges to output to in a formatted, easy to read document.
      5. Expense/cost tracking: Provides the ability to captivate and track expenses corresponding to a meal

### Hardware Interfaces The system has no hardware interface requirements.

### Software Interfaces The system has no software interface requirements.

### Communications Interfaces

* + - 1. HTTP/HTTPS - Used to facilitate communication between a user’s web browser and the web server hosting the database. See [RFC 9110](https://www.rfc-editor.org/rfc/rfc9110) for more information.

### Memory Constraints The expected user-base and data to be stored are small enough that there are no specific memory constraints required.

### Operations

### Admin User - Password protected administrator role unlimited access and privileges. Able to input and edit data.

### Basic User - The default. Basic users will be able to view unprotected application data, but may not edit or input data.

### Reports - Weekly automated reports will be able to be set up generating user specified content on a repeating basis.

### Site Adaptation Requirement The Food Forward Tracker will not come pre-configured with data. Meals, ingredients, households and any other necessary information will need to be entered or verified (Such as in the case of any preloaded ‘costs’ or example data) before the application will be ready for use.

# Product Functions

## Households - Each Household created and entered into the database will include names, contact information (email, phone and address), number of adults, number of children, and a method to track dietary restrictions, delivery notes or other short notes.

## The number of meals and snacks allotted per week will be calculated based on the information provided.

## Menu Items - These can be created, edited or copied (such as to enter an alternate low-solidum version). Each Menu Item will include a list of ingredients and a list of packaging materials required for its production and delivery as well as the number of servings produced. A season selection may be included to enable faster item sorting.

## Ingredients Inventory - Ingredients will be able to be directly added into this inventory section of the database and will be automatically removed as weekly menu plans are finalized. The application will support manually adding, editing, or deleting items from the inventory as necessary. An inventory overview will provide a current look at the quantities of all items currently being tracked.

* + 1. **Packaging Inventory** - Each meal requires different types of packing and shipping materials. These can be added and tracked in a separate inventory. This includes but is not limited to items such as mason jars, packaging supplies, containers/tubawares, etc. The application will support manually adding, editing, or deleting items from the inventory as necessary and support type and size entries for each. Items currently out on loan to customer households will be automatically deducted from the total on hand and an inventory overview will provide the quantities available.
    2. **Conversions** - The application will support measurement conversions.
    3. **Reports** - The application will generate a report, based on current inventory data, detailing what items (ingredients or supplies) need to be purchased in order to fulfill the week’s meal plan.
       1. Reports will be able to be automated or generated manually with customizable time-frames, meals/households served etc.

# User Characteristics

* + 1. The expected users are employees and volunteers of a nonprofit currently being established by the students and faculty of a University level Cultural Entrepreneurship course. Generally, users are expected to have basic technical skills and are capable of logging in to a web server and working with a database using a web interface.
    2. Users are not expected to have much if any experience with managing databases through DBMS, SQL, or command line. This is why a GUI interface for Food Forward is necessary.
    3. They are expected to be accessing the application on mobile devices while shopping, delivering food, or working in food storage and production locations. Web interface will enable this mobile use, increasing accuracy as data can potentially be entered immediately from anywhere.
    4. Users have experience working with recipe additions/subtractions, as well as managing inventory changes and household information, so the user interfaces provided to them are streamlined for ease of use, but not overly simplified.

# Constraints

* + 1. Household information can contain sensitive information. This data will have to be secured.

# Assumptions and Dependencies

* + 1. The program assumes compatibility with standard web interfaces.

# Apportioning of Requirements.

* + 1. **Security Measures** - Aside from the household information which is private, this database won’t require advanced security. Once the project reaches the stage where payment information is involved, security methods will have to be revisited and more advanced protections established to meet legal requirements.
    2. **Public Facing Mobile App.** - Future expansion plans include the addition of interfaces for Food Forward’s customers, allowing households to sign up directly and to make payments. This will require more interfaces and infrastructure to be set up and secure payment methods will be established at that time.

# Specific Requirement

# External Interfaces

* + 1. **Login Screen** - A welcome screen that prompts users for account information and facilitates gaining access to Food Forward Tracker.
    2. **Login Container** - The segment of the Login Screen that contains the username and password text input fields and a submit button
    3. **Universal Navbar** - A persistent navigation bar at the top of the screen on all pages containing button links to the following views.
    4. **User Interfaces**
       1. Account Page - A page that displays account information such as:

1. Name
2. Username
3. Current privilege level
   * + 1. Admin Page - Entering account view using an admin account will instead lead to a special account view page that contains all items from previous account view with:
4. Scrollable list of users
5. A checkable box labeled as “privileged” next to each user in the list that determines whether the specified user is given edit privileges for the database.
   * 1. **Software Interfaces**
        1. Report Page - Displays reports on the information in the database and allows the user to export these reports in PDF format.
6. Selection of report views will be facilitated through a selectable and scrollable list of Screens
7. Overview Report will be loaded when entering the Output Page
8. Selecting a report from the list while another report is loaded will unload that report and load the new report instead.
9. The report selection list will remain above the displayed report and is accessible at the top of the page
   * + 1. **Meal Report** - The second view of the Report page, accessible from the Report selection list
10. A selectable list of meal names that are currently in the database.
11. If a meal is selected, a table of ingredients are displayed below the selectable list.
12. If a meal is unselected, the table is depopulated and another meal may be selected.
13. If a meal is selected while another meal ingredient table is displayed, the previous table will depopulate and repopulate with the new ingredients.
    * + 1. **Packaging Report** - The third view of the report page, accessible from the Report selection list
14. Displays a table of meals and how many of each type of package (labeled with quantity units) is needed for that meal this week
    * + 1. **Household Report** - The fourth view of the report page, accessible from the Report selection list
15. Displays a list of household entries:

- Each household entry contains

* + - Name of the head of household
    - Number of adults in the household
    - Number of children from in the household
    - A section for dietary restriction notes for the household
    1. **Inventory Management Page** - A page that allows users to input and edit inventory data entries in the database. Only accessible to Privileged Users

1. Displays a table of all ingredients entries currently stored in the database
   * Each ingredient entry consists of
     + Ingredient Name
     + Packaging Type
     + Storage Type
     + Flat Fees
     + Pack Size
     + Minimum Quantity
     + Minimum Quantity Units
     + Cost per Unit
     + Stock on hand
     + Preferred supplier
     + Current supplier
   * To the right of each ingredient entry is an edit button that allows each data field to be changed by the user. Also to the right is a delete entry button that removes the entry from the database.
2. Above the table is a separated row of labeled inputs allowing users to enter new ingredient entries to the database
3. A submit button to the right of the row of inputs enters inputted entries into the database and empties the fields
4. A cancel button to the right of the submit button will empty the fields
   * 1. **Recipe Management Page** - A page that allows users to input and edit recipe data entries in the database. It also allows cloning a recipe to make minimum changes to form a new recipe. Only accessible to Privileged Users
5. Displays a list of selectable meal entries currently in the database
6. For each new meal, it will allow users to add ingredients, create recipe/meal, and add to packaging.
7. The New Meal button above the list expands a form allowing the user to enter a new meal into the database. This form contains:
   * Input fields:
     + A labeled input field for the name of the meal
     + An optional labeled input field for the planned week which defaults to N/A
     + A labeled input field for the servings portions of the meal which only accepts integer numbers
   * A row of input fields:
     + A drop-down menu allows the selection of an ingredient that is already entered into the database
     + A labeled input field accepts the quantity of the ingredient
     + The units of the selected ingredient is displayed next to the quantity field
     + An Add Ingredient button appends the ingredient entry to the ingredient list and empties the input fields
   * Ingredient list - An initially empty list of ingredient entries
   * A submit button that checks that all non-optional fields are entered and enters the meal entry into the database
8. Selecting a meal from the meal list expands the entry into an indented meal information section

* Each meal information section contains
  + A number representing the week that this meal is planned for
  + A number of portions that this meal serves
  + A list of ingredients and quantities with units required to prepare the meal
  + An edit button that allows any of the fields to be changed and reveals a submit button that inputs any changes to the database
  + A copy button that creates a copy of the meal entry in the database and appends it to the end of the list
    1. **Household Management Page** - A page that allows users to input and edit household information entries in the database

1. Displays a list of household entries:
   * Each household entry contains
     + Name of `the head of household
     + Email
     + Phone number
     + Address
     + Delivery notes
     + Receiving SMS text messages (Y/N)
     + Adds flag/symbol for vegan/vegetarian
     + Number of adults in the household
     + Number of children in the household
     + A section for dietary restriction notes for the household
     + An edit button that allows any of the fields to be changed and reveals a submit button that inputs any changes to the database
2. An Add Household button above the list expands a form allowing the user to enter a new household into the database. This form contains:
   * Input fields:
     + Name of the head of household
     + Email
     + Phone number
     + Address
     + Delivery drop off location
     + Receiving SMS text messages (Y/N)
     + Adds flag/symbol for vegan/vegetarian/gluten-free/allergies
     + Number of adults in the household
     + Number of children from ages 6-17 in the household
     + Number of children from ages 1-5 in the household
     + A section for dietary restriction notes for the household
   * A submit button that checks that all non-optional input fields are entered and enters the meal entry into the database
     1. **Packaging Information** - A page that allows users to input and edit container information in the database
3. Displays a list of container entries stored in the database

* Each packaging entry contains:
* Name of the package type
* Size of the package
* Units of package size

1. Above the list is a row of input fields that allow users to add a new container entry to database

* Input Fields:
* Name of package type
* Size of package
* A submit button imports the new package to the database, appends the new container to the list and clears the input fields

# Functions

* + 1. **“Entry” Definition**:
       1. An “Entry” (Ingredient Entry, Meal Entry, etc.) as used below corresponds to the relevant form of input fields, as defined in the previous External Interfaces section.
    2. **New Entry to SQL** - Food Forward Tracker accepts input data from a submitted form and assembles a SQL query that imports the new entry to the database
       1. SQL queries are tailored to the form that submits them, with user inputted data being filtered for SQL-queries and entered into the SQL code.
    3. **Execute Query** - Food Forward Tracker takes SQL queries assembled from form data and executes them on the database server
    4. **Authenticate User** - Food Forward Tracker takes login form data from the user and retrieves the level of privilege for the user’s account, stored in the database
    5. **Ingredients Form Processing** - Food Forward Tracker takes the information from each input field of an Ingredient Entry form and returns a SQL query that creates appends a new Ingredient Entry to the Ingredient Table
    6. **Meal Form Processing** - Food Forward Tracker takes information from each input field of a Meal Entry form and returns a SQL query that imports a new Meal Entry to the Meal Table
    7. **Inventory Form Processing** - Food Forward Tracker takes information from each input field of an Inventory Ingredient Entry form and returns a SQL query that imports it to the Ingredient Table in the database
    8. **Household Form Processing** - Food Forward Tracker takes information from each input field of an Household Entry form and returns a SQL query that imports it to the Household Tab in the database.
    9. **Unit Conversion** - Food Forward Tracker converts a measurement from one unit to a different unit through a function run on the server.
    10. **Calculate Servings Required -** From meal servings, meal quantity, and household member information, Food Forward calculate the servings required for each household for a specific meal on its planned week.
    11. **Calculate Meal Quantity Required -** From meal servings per household required, Food Forward calculates the meal quantity required to meet those servings
    12. **Calculate Ingredient Grand Total -** From total meal quantity, list of ingredients per meal, quantity of each ingredient, and cost of each ingredient, Food Forward calculates a grand total cost of the meal for its planned week.
    13. **Meal Plan** - Food Forward Tracker takes tracks a variety of meals and the households they are planned for to ensure the proper meal plan is laid out for each household

# Performance Requirements

## Simultaneous User Access - Users will be able to access the database without conflicting with other users

# Logical Database Requirements

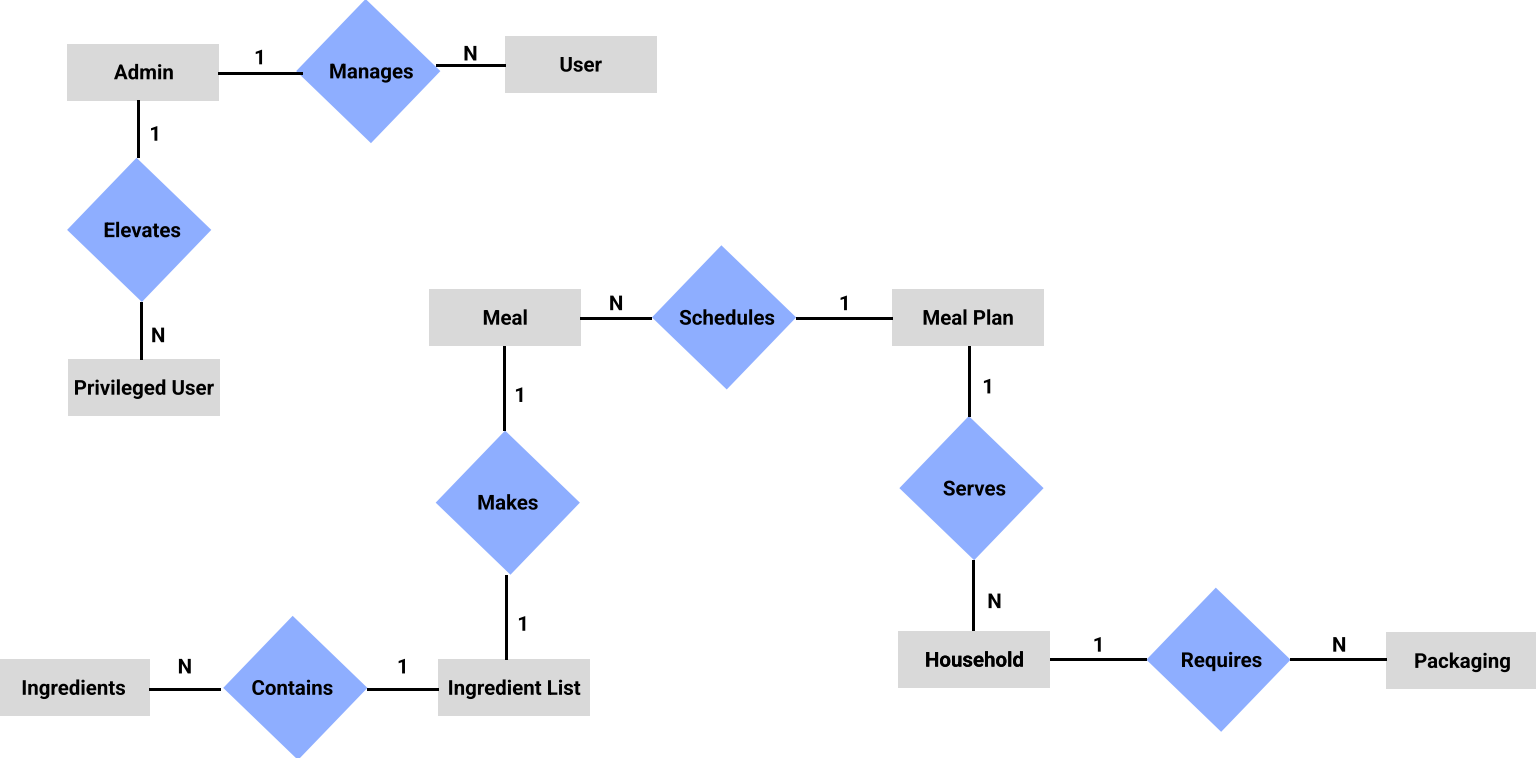


Figure 1: (OLD) Figma ER diagram

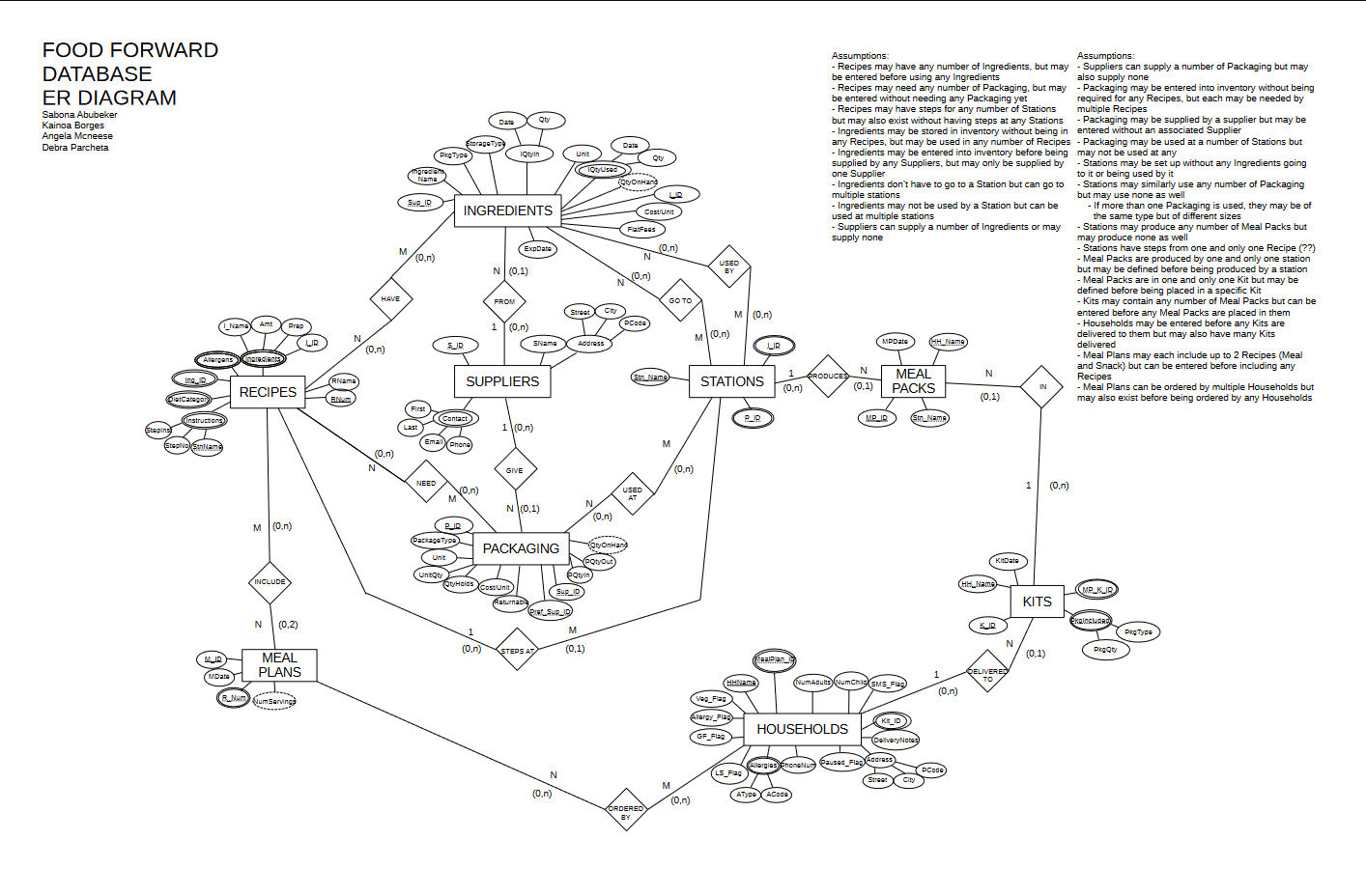


Figure 2: Updated ER diagram

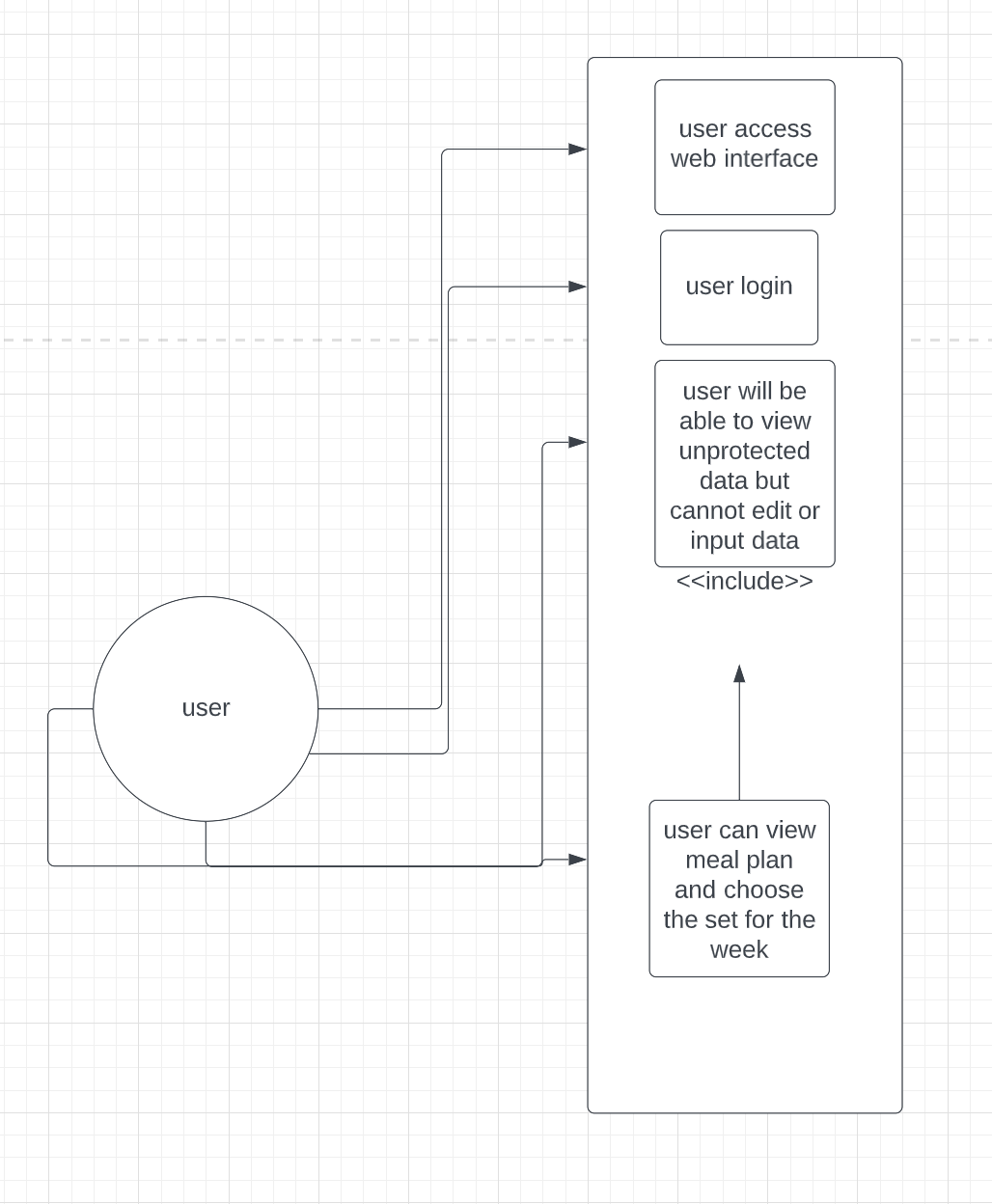


Figure 2: User Use Case Diagram

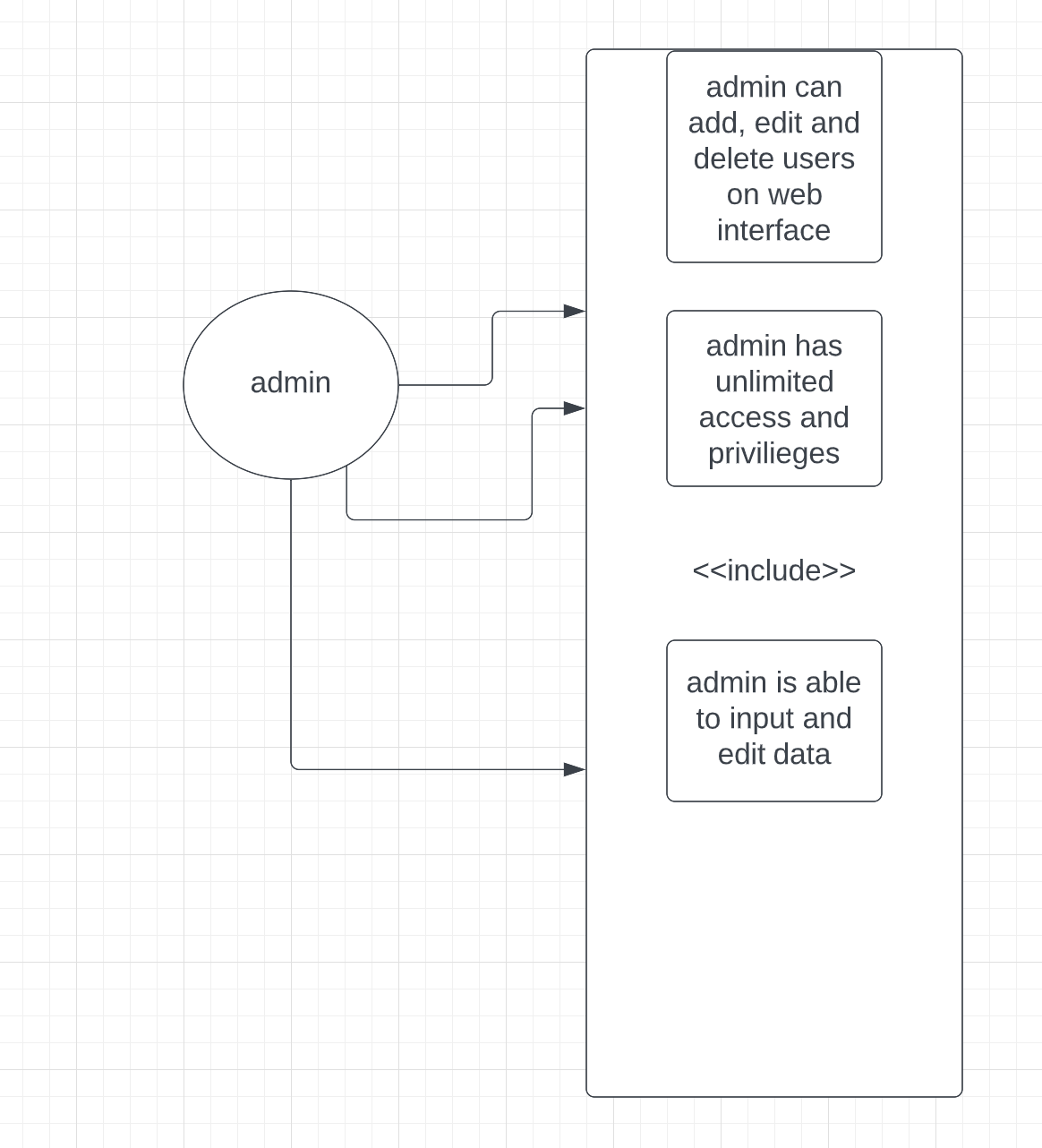


Figure 3: Admin User Use Case Diagram

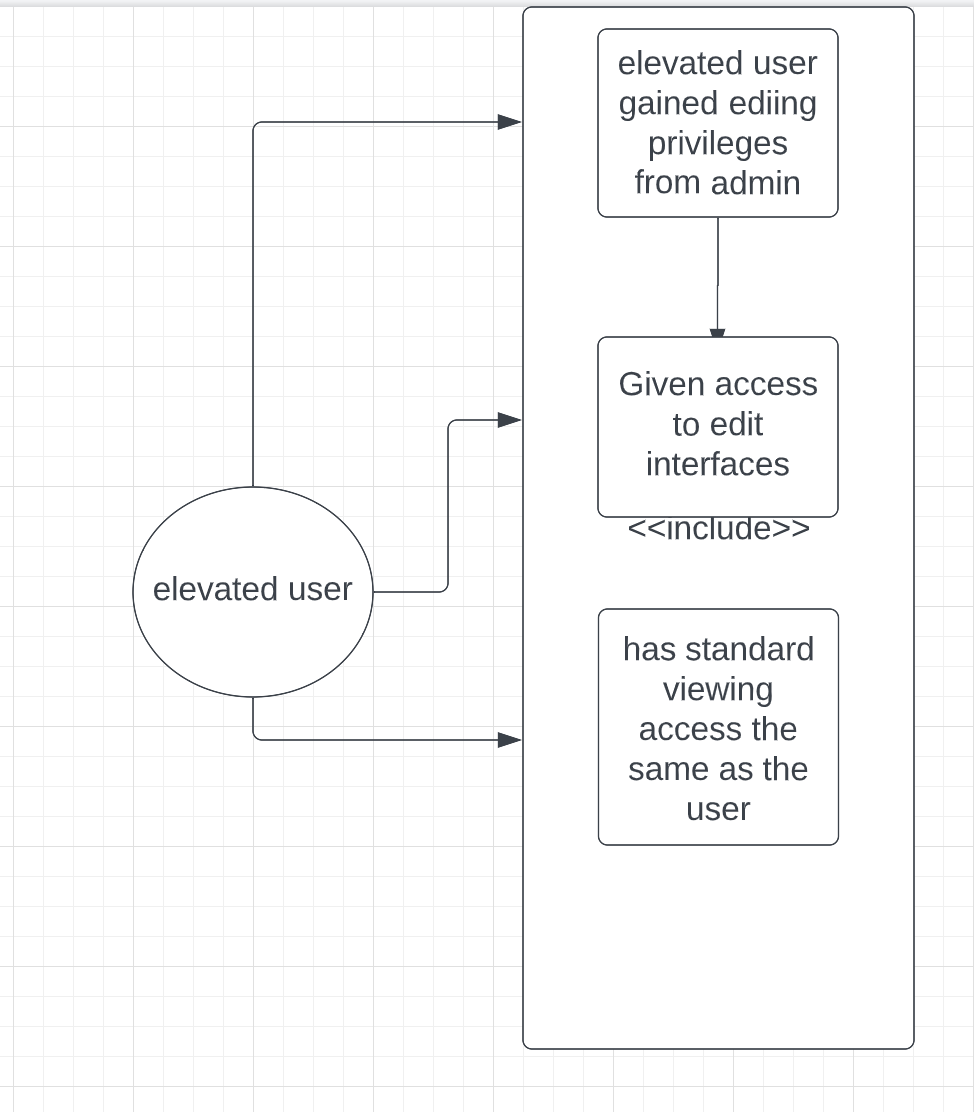
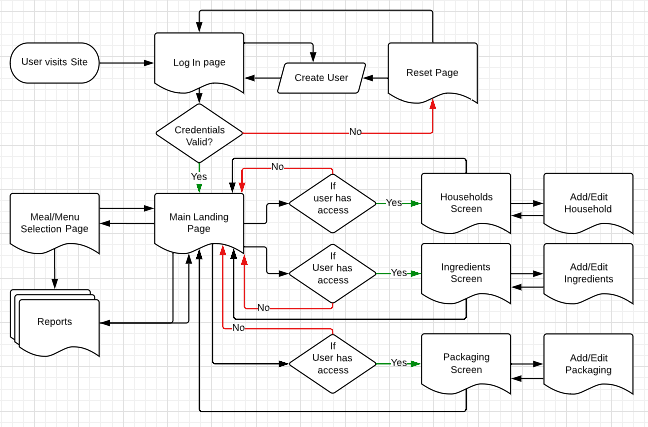


Figure 4: Elevated User Use Case Diagram

  
 Figure 5 Process Flow Diagram

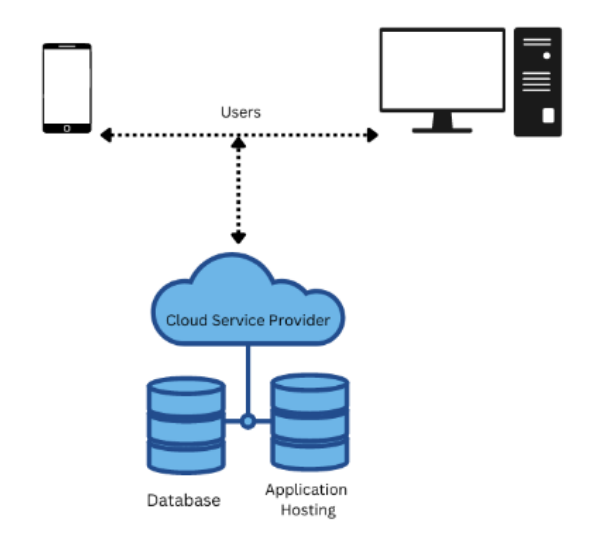


Figure 6. Architecture Diagram

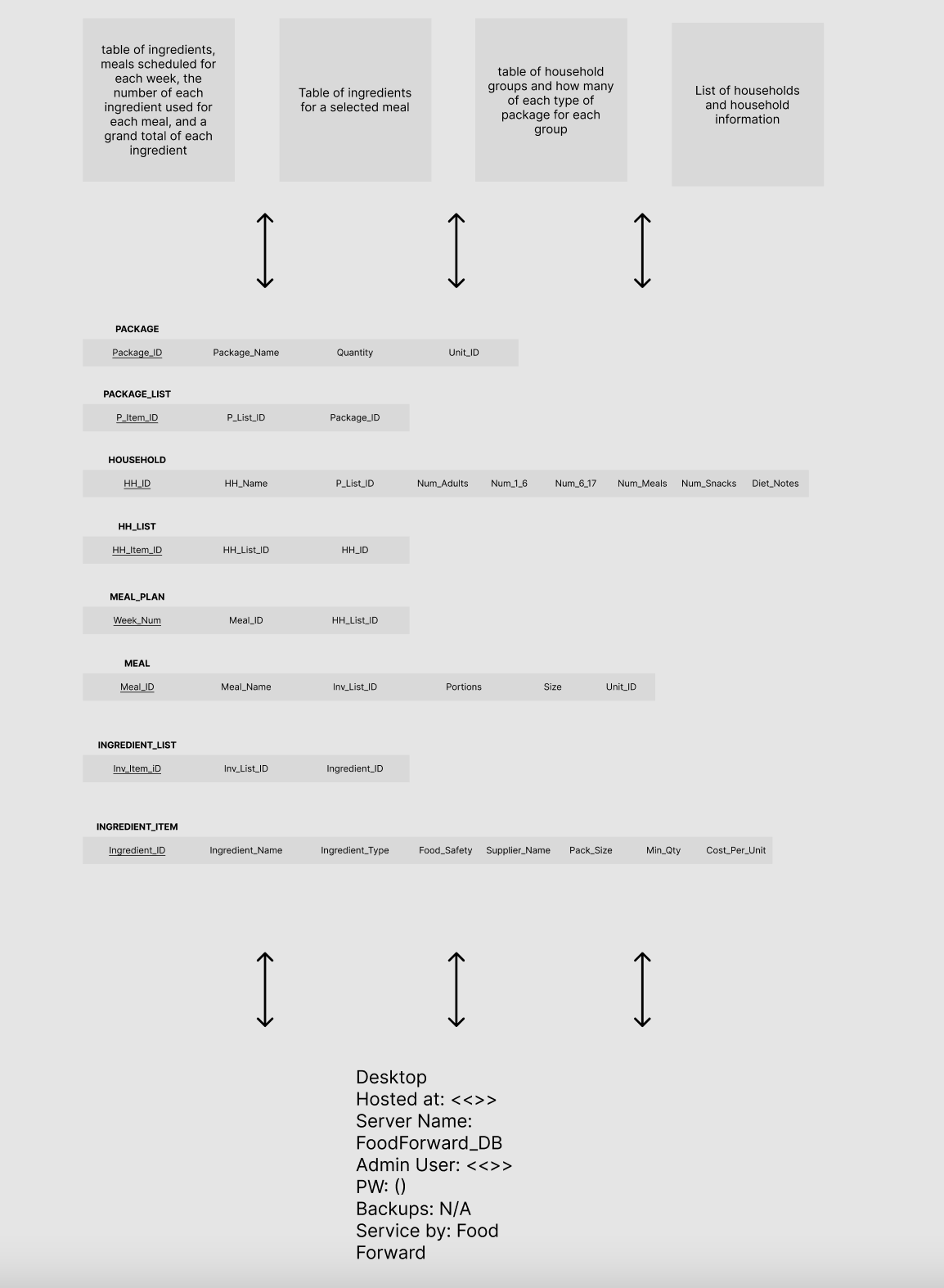


Figure 7. 3 Schema Architecture Diagram

# Design Constraints

## As a non-profit, cost constraints and simplicity are paramount.

## Minimizing costs should be prioritized

## Users are not expected to be technically adept so user interfaces must be as clear and unambiguous as possible.

# Software System Attributes

### Reliability

3.6.1.1. Database recovery will be handled by automatic, configurable backup log generation.

### Availability

* + - 1. The website and database need to be available during operational hours.
      2. Crashes should not result in the loss of any data already submitted.

### Security

* + - 1. Sensitive User and Customer data will be obfuscated or encrypted at rest and in transit.
      2. All user input must be sanitized and validated.
      3. System access and sensitive database changes will be logged and retained.
      4. Access to sensitive data will be restricted.

### Portability

* + - 1. **Use of a Web Interface** - A web-based user interface is used to allow Food Forward Tracker to be accessed on any platform
      2. **Responsive Interface Design** - Food Forward Tracker is designed with responsive output sizing to be readable and usable on mobile, tablet, and PC platforms

# Change Management Process

# Client Suggested Changes to Requirements

* + 1. The Client may suggest changes at any time through email or during weekly meetings with a follow up email detailing the proposed change.
    2. Changes will be considered by the team and will either be approved, denied, or suggestions will be made to make the changes as viable as possible.
       1. A change that requires other suggestions must then be agreed upon among both the team and the client for the change to become approved or denied
    3. Consensus must be reached within the team for a change to be approved or denied.

# Team Suggested Changes to Requirements

* + 1. Development team members may suggest changes to requirements to the client through email or during weekly meetings with a follow up email detailing the proposed change.
    2. Changes will be up to the client’s approval.

# Approved Changes

* + 1. Changes that have been approved by either the client or the team will be logged in the “Requirements Changes” section of the requirements document
       1. Each “Requirement Changes” entry must contain:
          1. A title for the change
          2. Date the change was approved
          3. A reason for which the change is necessary
          4. Alterations that will be made to the document to enact the change (including section numbers)

# Denied Changes

* + 1. Changes that have been suggested by the client but denied by the team may be logged to the “Future Changes” section by client request
    2. Each “Future Changes” entry must contain
       1. A title for the change
       2. Date the change was denied
       3. Reason for which the change is preferred
       4. Reason for denying the change
    3. Changes that have been denied by the client will not be logged in the “Future Changes” section except by client request.

# Document Approvals

*Identify the approvers of the SRS document. Approver name, signature, and date should be used.*

1.

2.

3.

4.

5.

# Supporting Information

# Food Forward Weekly Forecast - Aparna Katre (2022)

# Feasibility - Debra Parcheta (2022)

# Changes

## Approved Changes

* + 1. Oct 21, 2022  
       <If Time Permits>Add/modify/delete recipe card
       1. Add/modify/delete recipe card  
           - Meal preparation instructions with 6 steps at most (step number, image, 1-4 bullet points)  
           - What's in your crate  
           - What you will need  
           - Main recipe image
       2. Ability to print it
       3. Tips sheet containing (following types of information as paragraphs):  
           - Note (to be sent to the customers to aid the meal preparation)  
           - Ingredients and measurements for one serving  
           - Potential substitutes - description  
           - How to best save the food for future use
       4. Recipe categories (may add a few more categories)  
           - Comfort food   
           - Healthy food  
           - Gourmet meal  
           - One pot / pan style meal  
           - Needs baking - Y/N
       5. A video that matches the steps in the instructions

## Denied Changes

## Future Changes