NYU | TANDON SCHOOL OF ENGINEERING

Computer Science and Engineering

Hunger Warrior

System Requirements Specification (SRS)

Version 4.0

Document Number: SRS-004

Project Team Number:	B14	
Project Team Members:	Gavin Senger (gjs374)	Sanidhya Sitaula (ss12081)
	Kori Vernon (ksv244)	Courtney Battieste (cbb389)

REVIEW AND APPROVALS

Team Members	Function	Date	Signature
Gavin Senger	Author	09/27/2021	Gavin Senger
Sanidhya Sitaula	Author	09/27/2021	Sanidhya Sitaula
Kori Vernon	Author	09/27/2021	Kori Vernon
Courtney Battieste	Author	09/27/2021	Courtney Battieste
Professor Strauss	Review/Approval		

REVISION LEVEL

Date	Revision Number	Purpose
03/09/2021	Version 1.0	Initial Release
03/18/2021	Version 2.0	Second Release
04/15/2021	Version 3.0	Third Release
09/27/2021	Version 4.0	Fourth Release

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this System Requirements Specification is to clearly define the system under development, namely Hunger Warrior. The intended audience of this document includes the professor who will be reviewing our project, the team members of the project, fellow students in the class, and the end users of Hunger Warrior. Other intended audiences include the development team and members of the developing organization.

2. SCOPE

Hunger Warrior is a project that aims to assist with the fight against food insecurity and food waste in the United States of America by reallocating food to those that are in need. This is done by helping shelters get connected with grocery stores, allowing the grocery stores to donate food to the shelters that would've otherwise been thrown out. Grocery stores will be incentivized to limit excess food waste with increased tax incentives, and homeless shelters will gain access to more affordable food and access to more efficient distribution. Our goal is to make Hunger Warrior easy to implement into local communities, gradually allowing for expansion in the future. There are other products in the market that are similar in goals such as minimizing food waste and making food more attainable, but none that are similar in methods.

Hunger Warrior aims to connect grocery stores with homeless shelters in order to efficiently organize a reallocation and distribution of food. Hunger Warrior is intended to provide procedures for shelters that will enable requests of food from grocery stores, and manage their inventory. Hunger Warrior also intends to provide procedures for grocery stores that will grant a tracking functionality so that the stores can see the food requests, as well as documentation of shipments and tax information within the website. Administrators will be able to add grocery stores and homeless shelters to the system, allowing them to send and receive food respectively. The system will include the following functionality:

- Dashboard interface for Grocery stores and Shelters
- Real time food requests by Shelters in the local area that can be fulfilled by Grocery Stores
- Visualization of tracking using a map
- Tax Incentive Documentation / Inventory Documentation

The goals of the system are:

- Connect Shelters to Grocery stores
- Reduce food waste when possible
- Provide benefits to participating Grocery stores and Shelters
- Be user friendly for employees and clients

However, the system will not be responsible for the specifics of the deliveries, nor will it be responsible for federally qualified food inspection.

2.1 Identification

Name: Hunger Warrior - System Requirements Specification

Number: SRS-004

Version 4.0

2.2 Bounds

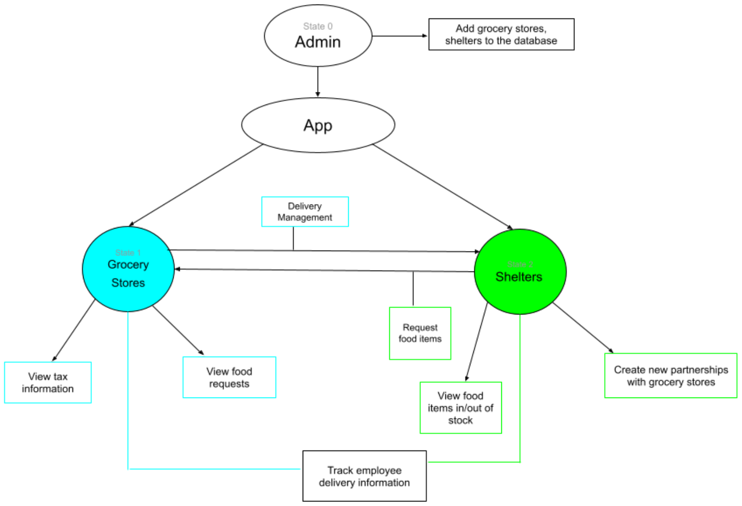
- Firestore to hold Grocery store and shelter information
- Browser that is connected to the internet to access web pages
- Hosted on a secure web page (HTTPS)
- The user interface will be beginner-friendly and intuitive
- Must be accessed by a browser (Chrome, Safari, Firefox)
- Adapted for tablet (landscape) and use on a personal computer
- Use of Google Maps API
- Food quality and inspection not facilitated by Hunger Warrior

2.3 Objectives

- Project Priorities:
  - Grocery store and shelter data accumulation
  - Efficient location and delivery matching
  - Seamless user interface
- Incremental Life Cycle
- Deliverables / Milestones:
  - Project Team Selection Form, September 20
  - Project Proposal, September 22
  - Software Requirements and Analysis Specification (SRS), September 27
  - Software Project Management Plan (SPMP), October 4
  - Project Description, October 6
  - Software Design Description (SDD) - Initial, October 12
  - Design Document Final, December 13
  - Presentation, Starting December 8

3. OVERALL SYSTEM OVERVIEW

3.1 Context Diagram


State 0
Admin
Add grocery stores, shelters to the database
App
Delivery Management
State 1
Grocery Stores
State 2
Shelters
Request food items
View tax information
View food requests
View food items in/out of stock
Create new partnerships with grocery stores
Track employee delivery information

3.2 Additional Descriptive Items

a. Product functions - A system that matches homeless shelters with grocery stores in order to reduce food waste through food reallocation. Shelters will be able to request food items and receive food deliveries from grocery stores, while grocery stores will be able to organize inventory, donations, and tax information.

b. User characteristics - The intended users can be separated into three categories: Grocery employee, Shelter employee, and delivery management. The grocery employee will be able to log inventory and prepare necessary items for their appropriate shelters. This training is provided already by grocery store policy but is necessary in order for deliveries to run smoothly. The shelter employee will be able to document shelter inventory, manage the requests that the shelter needs, and organize the deliveries that are brought to the shelter. This is necessary in order to ensure that the needs of shelters are met. Delivery management are responsible for loading and transporting food from the grocery store to its respective shelter location. This is necessary in order to ensure that the needs of grocery stores and shelters are met within a timely manner.

c. Constraints - For regulatory constraints, all food preparation and transportation must live up to federal standards in order to ensure the security of those residing at the homeless shelters. For interface to other application constraints, a way to be able to merge grocery store inventory and homeless shelter inventory into the inventory being kept by Hunger Warrior would greatly decrease the amount of work needed to manually manage inventory. For operations constraints, availability of delivery management could serve as a constraint and affect both grocery store inventory and shelter needs. For priority constraints, perishable foods are foods that have a high risk of being spoiled and must be delivered quickly or can be limited from being donated. For safety and security constraints, the quality of the food is always a risk, even after passing federal regulations.

d. Assumptions and dependencies - The factors that can affect the requirements are Federal Food Safety Laws, homeless shelter donation policies, and grocery store policies.

e. Requirements subsets - Basic or fundamental requirements that are needed for the website can be broken down into three categories: Grocery store, Homeless Shelter, and administrative. Requirements that are needed for grocery stores include: an interface where inventory and tax information can be stored and access to local shelter requests. Requirements that are needed for homeless shelters include: an interface where inventory and future donations can be stored and ability to request a need for certain items. Requirements that are needed for administrative purposes include: a base where both shelter and grocery store information can be stored, an algorithm that properly matches and organizes transfer of inventory between locations, and a team that manages day to day interactions in order to limit miscommunication. A fourth possible requirement subset that can be added in the future can be delivery management requirements. As of right now, delivery management involves tracking and time keeping but with further expansion it can include arranging employees and vehicles for delivery.

4. DOCUMENT OVERVIEW

The SRS will go over the reference documents that were used as reference for this document. It will then go over all of the requirements (business, specific, non-functional), and then describe the component architectures, provide class diagrams, explain state logic, and depict the overall behavior under the analysis section. The document will also detail the test plans, qualification provisions, appendices, and dictionaries. Overall, it describes the initial framework for the development of the project and creates a clear picture of the goals and obligations, which will also be updated throughout the course of the development period.

5. REFERENCE DOCUMENTS

A14, Hunger Warrior - Project Proposal BSSV, PP-001, Version 1.1, 2/24/2021

6. BUSINESS REQUIREMENTS

6.1 Technology

The technologies included are a database that stores information on grocery stores and shelters, an algorithm that can properly organize matches, inventory, and delivery times, and an application or website that can make this technology user accessible.

6.2 Economics

The goal is to get funding through donations and subsidies, leading to creating a sustainable non-profit organization.

6.3 Regulatory and Legal

The quality of the food must be inspected using Food Safety and Quality regulations provided by the Food and Agriculture Organization (FAO) and must meet U.S. Food and Drug Administration (FDA) standards . Furthermore, tax incentives must be examined for potential exploits.

6.4 Market Considerations

The primary market is those who are food insecure, and the goal is to grant ease for food redistribution. This benefits those who wish to be rid of food resources and those that seek them. We also wish to appeal to investors and the government, who may subsidize our costs and allow us to maintain a non-profit project.

6.5 Risks and Alternatives

Business Risk: The software is no longer needed
Description: There are already existing partnerships that exist between shelters and grocery stores
Probability: 35%
How discovered: There are some partnerships that already exist between grocery stores and shelters
Responsible Party Status: Inactive
Mitigation Plan: Focus on how to provide tax benefits for these already existing partnerships and how to further improve service within app

Operational Risk: Grocery store/Shelter pairing is not mutually beneficial
Description: The pairing between the grocery store and shelter is not mutually beneficial
Probability: 20%
How discovered: Reported by grocery store and/or shelter
Responsible Party Status: Active
Mitigation Plan: There will be a vetting process to determine whether or not a pairing will be mutually beneficial.

Technology Risk: Device incompatibility
Description: The web application works best on tablet and personal computer/desktop, so incompatibility with other devices might be an issue.
Probability: 20%
How discovered: Some websites do not work dynamically
Responsible Party Status: Active
Mitigation Plan: Have a disclaimer that the application is strictly a desktop and tablet application.

Economic Risk: Inaccurate tax benefits
Description: The tax benefits for the grocery store could be inaccurate-- either too much or too little.
Probability: 20%
How discovered: Reported by grocery store
Responsible Party Status: Active
Mitigation Plan: Have exact readings that are doubly checked by two grocery store personnel.

6.6 Human Resources and Training

Administration members are needed to be able to add stores and shelters to the database, as well as to deal with customer and IT support. Training is needed at both the store and shelter level to ensure safe and proper use of the website. Training is also needed for grocery store employees at the local level to efficiently prepare food for delivery. There must be a support team for Hunger Warrior in order to deal with malfunctions and miscommunication between grocery stores and shelters.

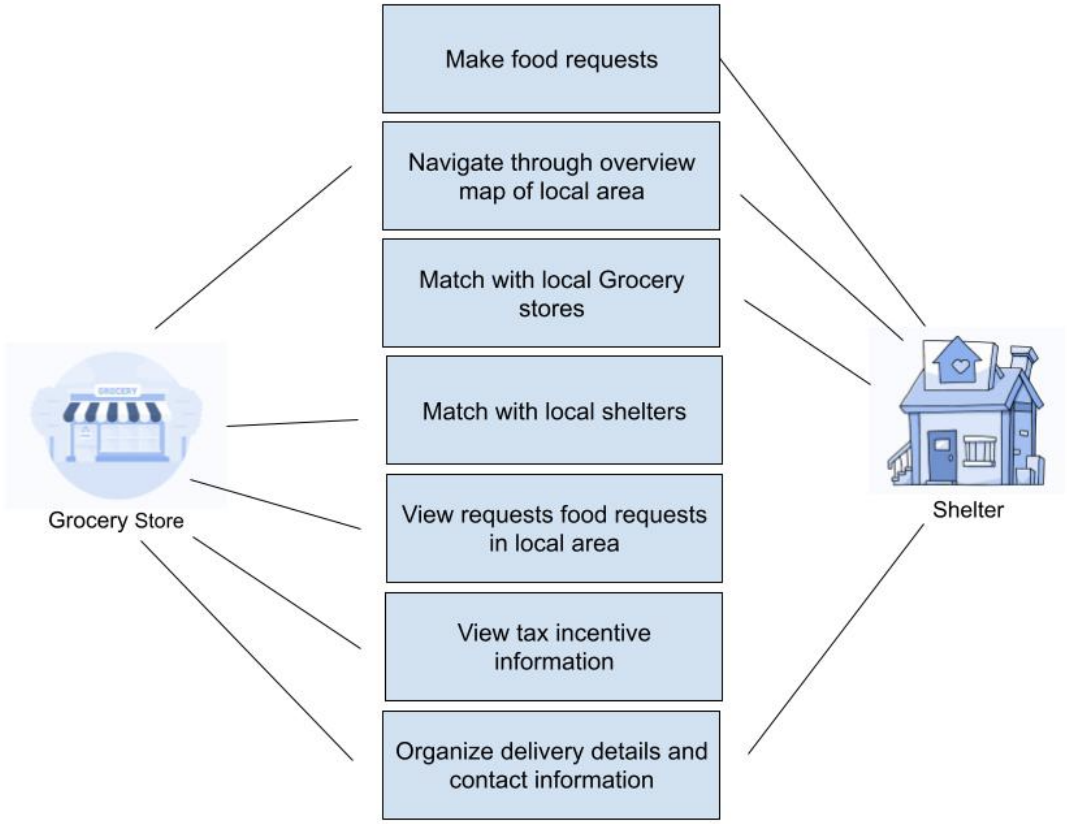
7. SPECIFIC REQUIREMENTS (DESCRIPTIVE FUNCTIONAL REQUIREMENTS)

7.1 Functional Descriptive Detailed Requirements

- Shelters shall be able to provide their location and get a list of grocery stores near them.
- Shelters shall be able to order food items to the grocery store of their choice and receive order updates from the respective grocery store.
- Shelters should be able to schedule future deliveries in anticipation of food shortages.
- Shelters shall be able to search for a grocery store if they do not find the store they need among the local stores provided.
- Shelters shall be able to communicate with a grocery store before they order food supplies.
- Grocery stores shall be able to edit their food inventory and view tax information.
- Delivery tracking information shall be shared with both grocery stores and shelters.
- Grocery stores and shelters shall be able to update and manage their information on the website.

7.2 Requirements Use Cases

7.2.1 Use Case Diagrams


Make food requests
Navigate through overview map of local area
Match with local Grocery stores
Match with local shelters
View requests food requests in local area
View tax incentive information
Organize delivery details and contact information
Grocery Store
Shelter

7.2.2 Use Case Descriptions

Make Food Requests		
Description	The Shelter user can request certain food items	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database, and a request form is generated.	
Flows	**Basic**	1. Press ‘Request food item’ button on Dashboard 2. Fill out desired quantity, food item, food type, delivery date, and other pertinent information 3. Confirm the request by entering password
Post Conditions	Once the food item request form is completed, send the information to the database, and populate the request via pin on the Google Maps API.	
Special Requirements	When necessary, the Google Maps API will be regularly updated to support the latest version of the interface.	
Extension Points	None	

Navigate through map overview of local area		
Description	Both the grocery store and shelter users can navigate the Google Maps API	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database, and must show a visualization of the local area map.	
Flows	**Basic**	1. Scroll vertically and horizontally through the local area map

		2. Drag map finger or computer mouse in any direction 3. Pins populate with requests in the local area, as well as pins of nearby shelters and grocery stores.
Post Conditions	The map will look different based on user changes.	
Special Requirements	When necessary, the map will regularly be updated to support the latest version of Google Maps API and bugs on the page will be fixed.	
Extension Points	None	

Match with local Grocery stores		
Description	Shelters can match with local grocery stores	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database.	
Flows	**Basic**	1. A list of grocery stores will pop up with a truncated list of available food items 2. The shelter user can view statistics and ratings about the grocery store 3. Shelters can request food items which can lead to a match with local grocery stores to supply said food items
Post Conditions	The shelter will see a “match found” flag on the stores that they are matched with and the request will be populated in Firestore.	
Special Requirements	When necessary, bugs on the page will be fixed.	
Extension Points	None	

Match with local shelters		
Description	Grocery stores can match with local shelters	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database.	
Flows	**Basic**	1. A list of shelters will pop up with a truncated list of requested food items 2. The grocery store user can view statistics and ratings about the shelter store 3. Grocery stores can request to match with local shelters to supply food items
Post Conditions	The Grocery store will see a “match requested” flag on the shelters they request matches with and the request will be populated in Firestore.	
Special Requirements	When necessary, bugs on the page will be fixed.	
Extension Points	None	

View food requests in local area		
Description	Grocery stores can view food requests in local area	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database, and Google Maps API	
Flows	**Basic**	1. Grocery stores can view food requests in area on local area map visualization 2. List will be on the right or left hand side of the screen with truncated list of food item requests for shelters
Post Conditions	The map will change based on screen movement, and the grocery store user can click on shelters and see if the request has been fulfilled.	

Special Requirements	When necessary, the map will regularly be updated to support the latest version of Google Maps API and bugs on the page will be fixed.
Extension Points	None

View tax information		
Description	Grocery stores can view their tax information and donation history	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database.	
Flows	**Basic**	1. Statistics will be generated for the Grocery stores user so they can view their tax information for the fiscal year.
Post Conditions	The user will be able to access and store their tax information documentation.	
Special Requirements	When necessary, bugs on the page will be fixed.	
Extension Points	None	

Organize delivery details and contact information		
Description	Both shelters and grocery stores will be able to track and keep contact information of delivery management	
Pre-Conditions	The web page must be loaded, connected to the internet, connected to the database.	
Flows	**Basic**	1. Food donation location and time of arrival can be tracked by both grocery stores and shelters. 2. Delivery management contact information

		is filled out upon arrival at the grocery store.
Post Conditions	The soonest possible delivery date and time match will be selected, and the grocery store and shelter will need to confirm.	
Special Requirements	When necessary, bugs on the page will be fixed.	
Extension Points	None	

8. NON-FUNCTIONAL DESCRIPTIVE DETAILED REQUIREMENTS

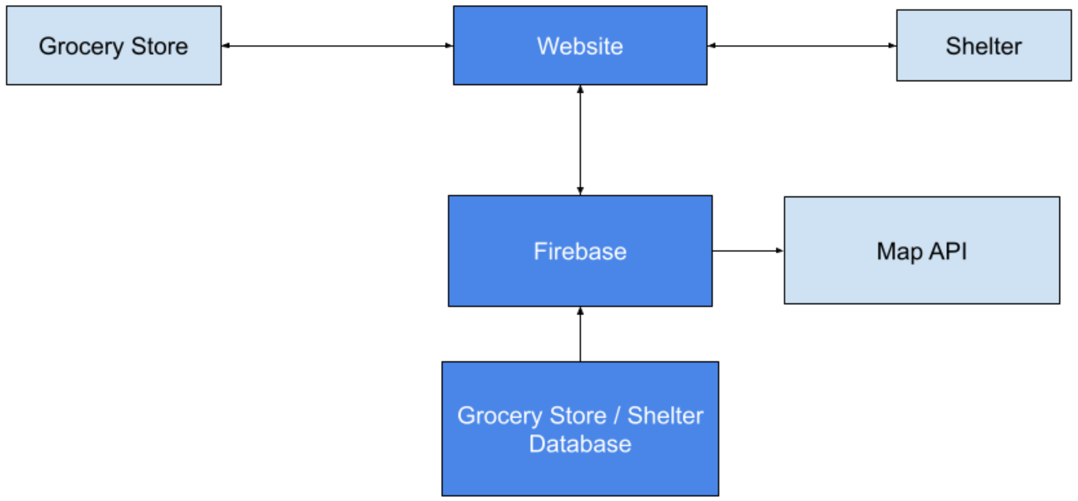
- Firebase, a cloud-based service, shall be used to store the information on grocery stores and shelters
- A map API shall be used in determining location and tracking deliveries
- Shelters and grocery stores shall be able to view our Terms and Conditions document at all times
- Both grocery stores and shelters shall be able to view or change their login information by getting in contact with the admin

9. ANALYSIS

9.1 Component Architecture

In order to create the web application for Hunger Warrior to function, we will be using a back-end of Javascript code attached to a front-end using HTML. The back-end of our software will be communicating with Google Firestore, which will act as the database for our project and store all relevant information to the project, such as locations, food items, and order history.

9.2 Component Architecture Diagram



9.3 Component Descriptions

Grocery Store - Able to transfer data to the website and receive data from the website

Shelter - Able to transfer data to the website and receive data from the website

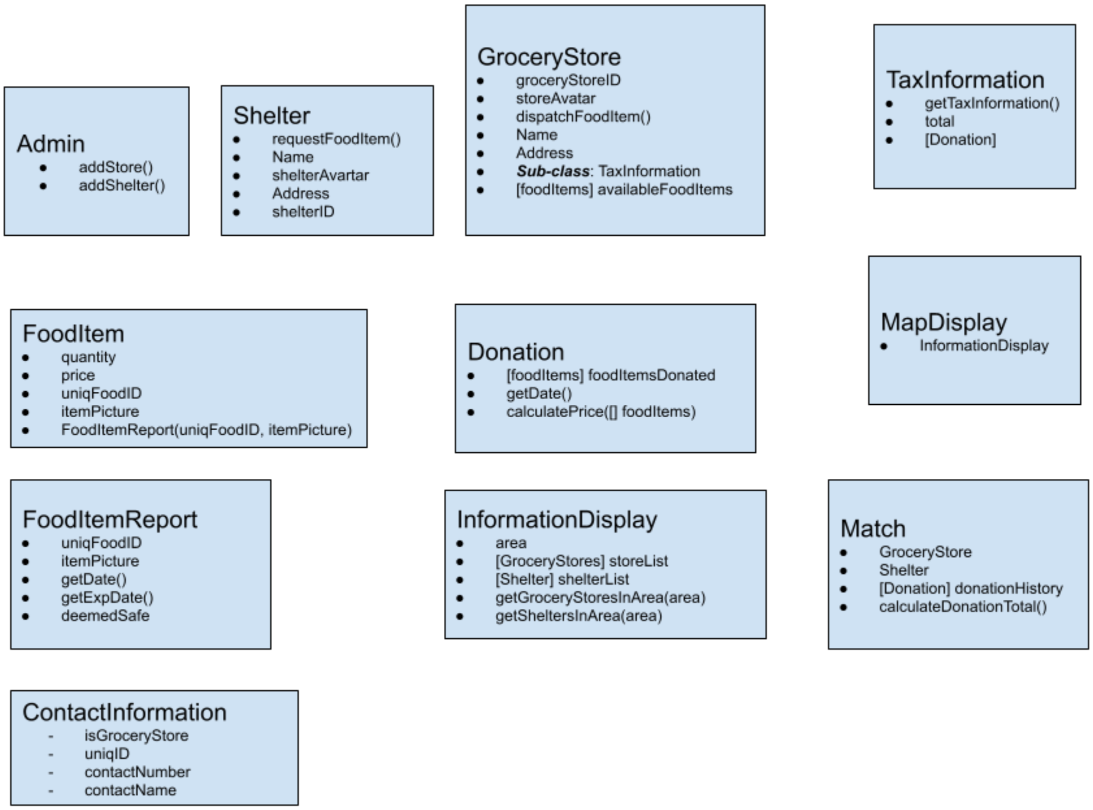
Firebase - Able to obtain transferred data from the website and update Map API.

Database - Able to obtain data from firebase in order to update and maintain data

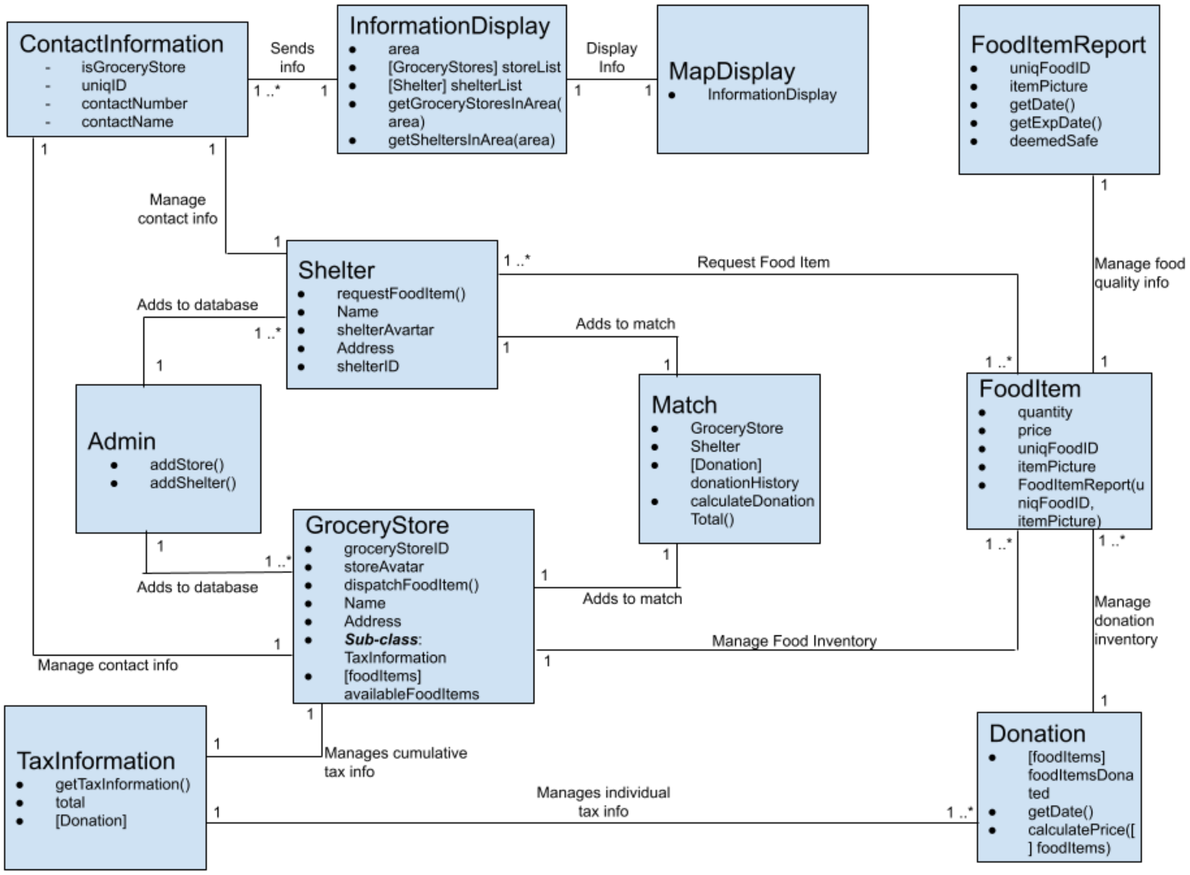
Refer to Dictionaries for further detailed information.

9.4 Class Diagrams

9.4.1 Individual Class Diagrams



9.4.2 Class Relationship/Interaction Diagrams



9.5 Events

The system responds to the following events:

1. Make Food Requests
2. Navigate through overview map of local area
3. Match with local Grocery Stores
4. Match with local Shelters
5. View food requests in the area
6. View tax information

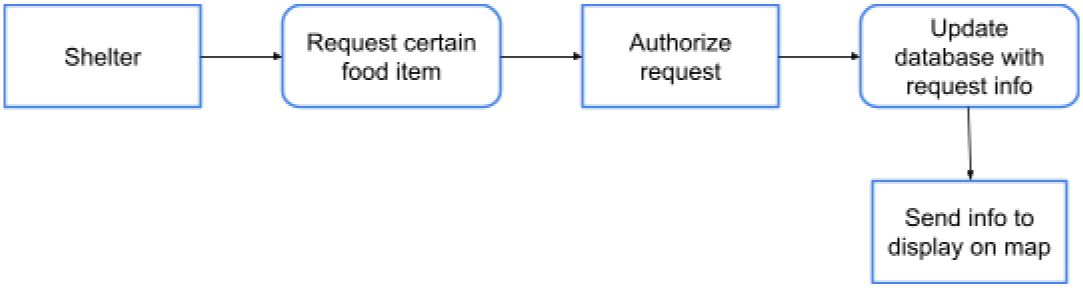
7. Organize Delivery Details and Contact Information

9.5.1 Motives

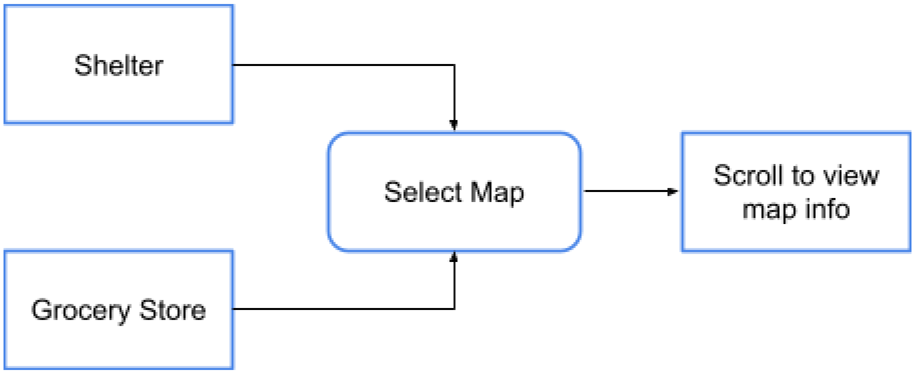
1. Make Food Requests - Shelters will be able to make requests for food that they would like to acquire.
2. Navigate through an overview map of the local area - Add interactivity for both grocery stores and shelters so that users can see different participating grocery stores in the area.
3. Match with local Grocery Stores - Give the option to match with Grocery stores so that shelters can develop a relationship and partnership
4. Match with local Shelters - Give the option to match with shelters so that they can develop a relationship and partnership.
5. View food requests in the area - View the available food requests in the area so that Grocery stores can decide how to distribute goods
6. View tax information - This is so the grocery stores can use the tax information from donations in Tax withholdings
7. Organize Delivery Details and Contact Information - This is to organize the delivery of goods as well as the contact information to connect Grocery Stores to shelters.

9.5.2 Event Diagrams

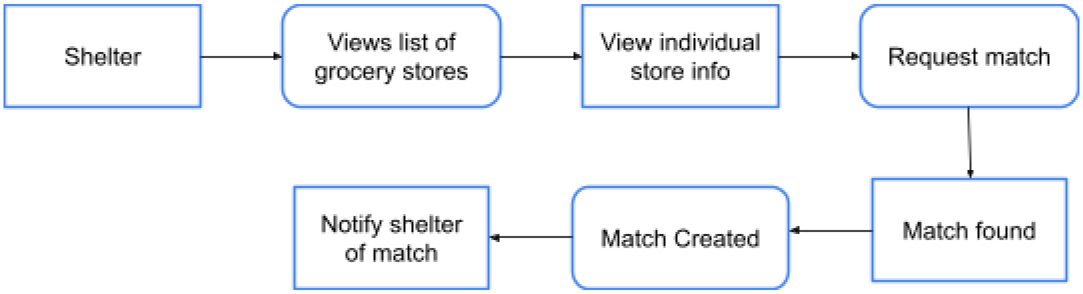
Make food requests


Shelter
Request certain food item
Authorize request
Update database with request info
Send info to display on map

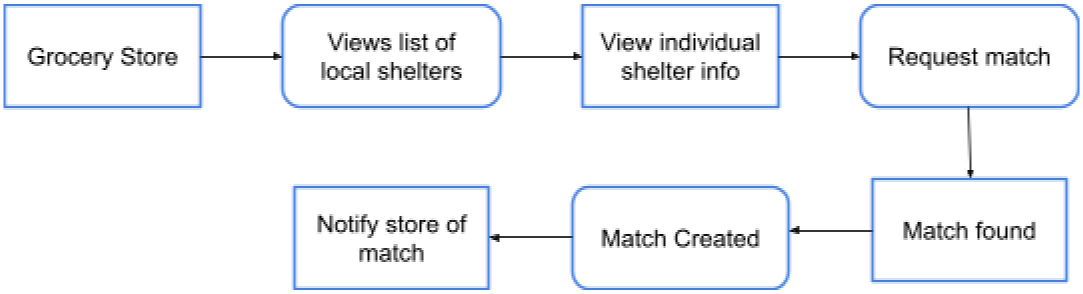
Navigate through overview map of local area



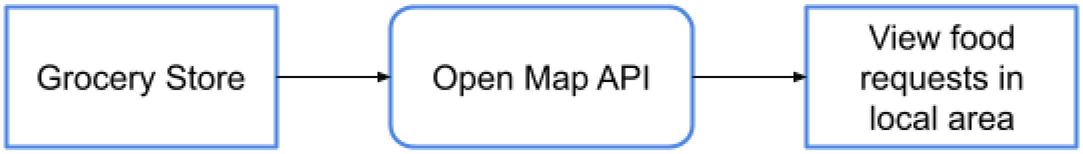
Match with local Grocery stores



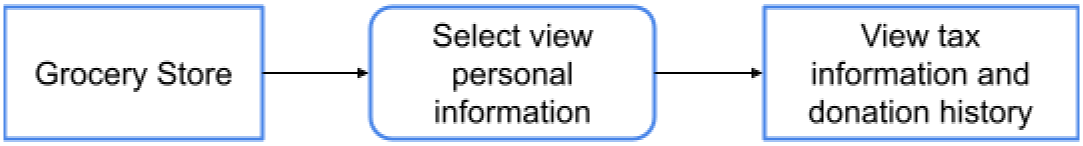
Match with local shelters



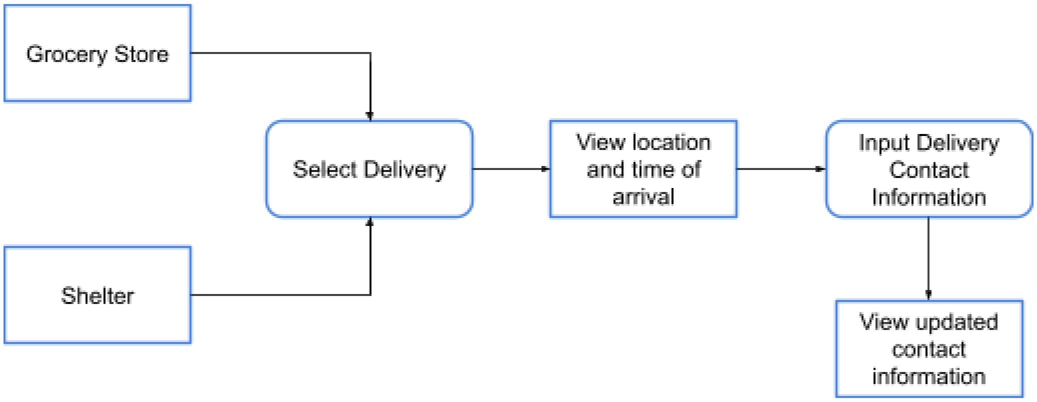
View food requests in local area



View tax information


Grocery Store
Select view personal information
View tax information and donation history

Organize delivery details and contact information


Grocery Store
Shelter
Select Delivery
View location and time of arrival
Input Delivery Contact Information
View updated contact information

9.6 Activity/State (Scenario) Sectional

The Activity/State Sectional will be provided at a later date.

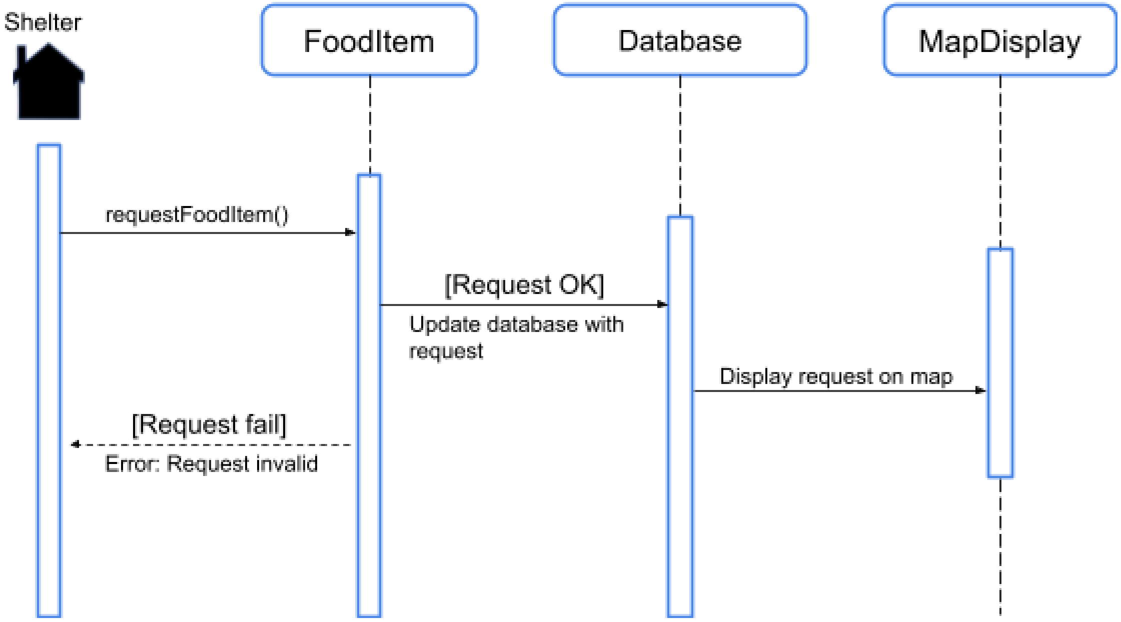
9.7 State Logic

The state logic will be provided at a later date.

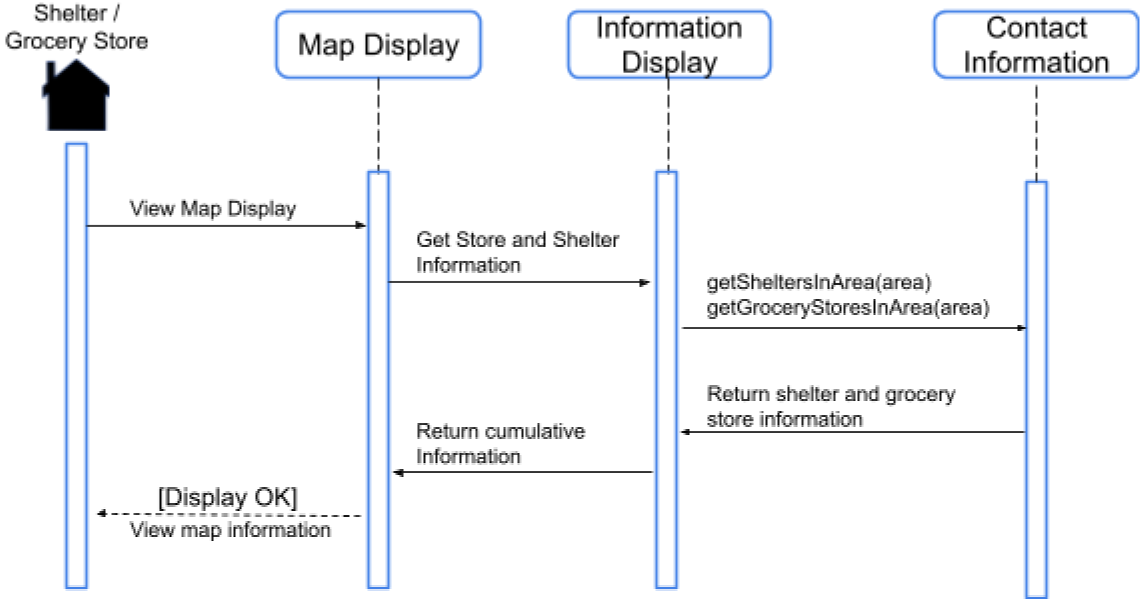
9.8 Behavior

9.8.1 Sequence Diagrams

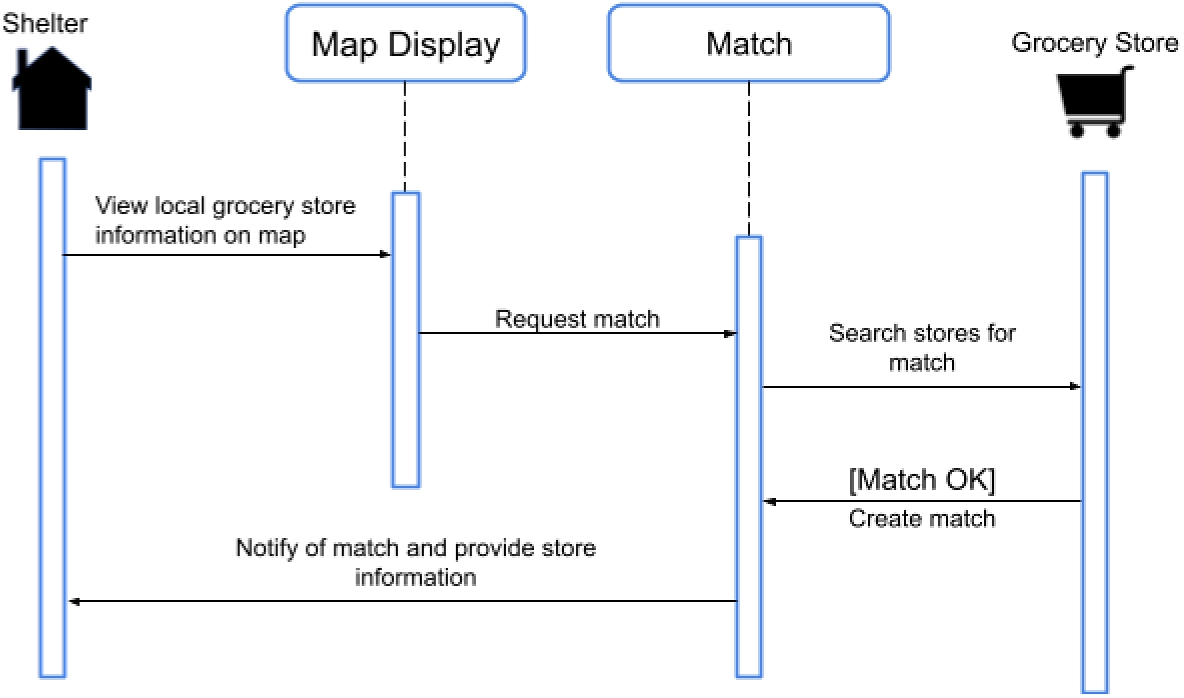
Make food requests



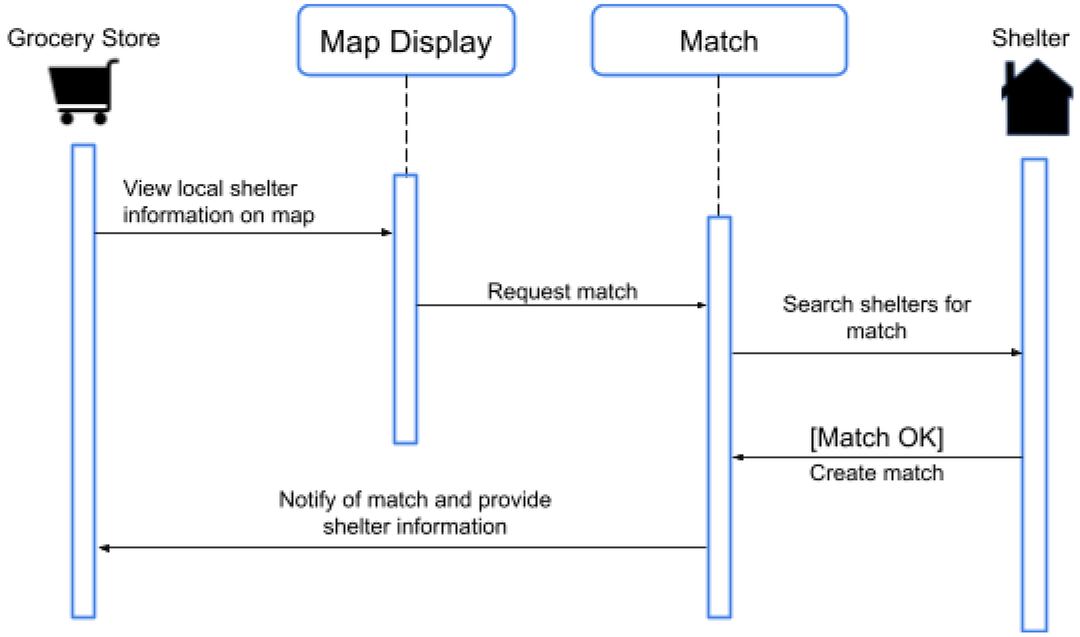
Navigate through overview map of local area



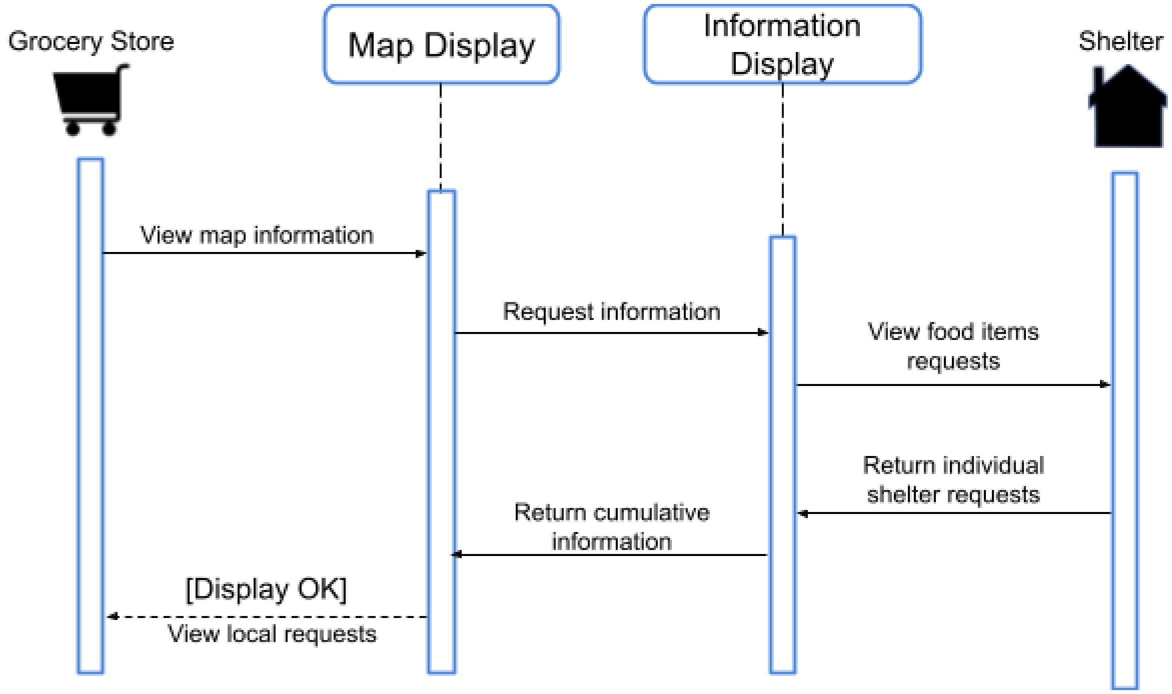
Match with local Grocery stores



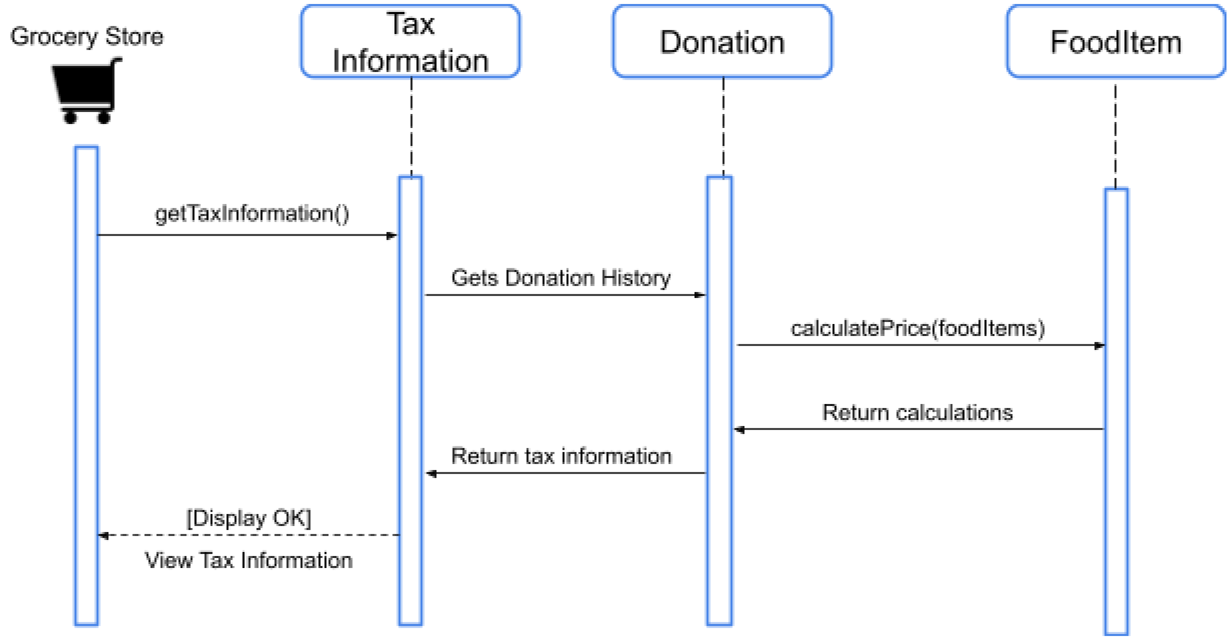
Match with local shelters


Grocery Store
Map Display
Match
Shelter
View local shelter information on map
Request match
Search shelters for match
[Match OK]
Create match
Notify of match and provide shelter information

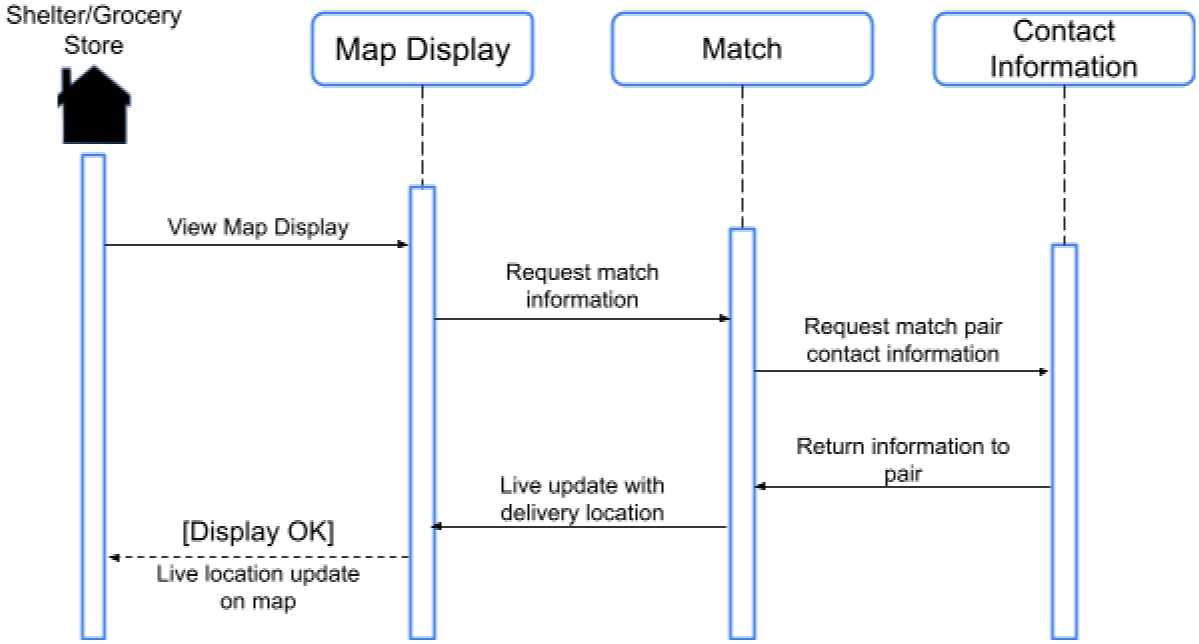
View food requests in local area



View tax information



Organize delivery details and contact information


Shelter/Grocery Store
Map Display
Match
Contact Information
View Map Display
Request match information
Request match pair contact information
Return information to pair
Live update with delivery location
[Display OK]
Live location update on map

9.8.2 Collaboration Diagrams

Collaboration Diagrams will be provided at a later date.

10. SYSTEM TEST PLAN REQUIREMENTS (UML)

Product Test

- A demo version of the website will be created for a few grocery stores and shelters. Feedback and errors will be reported.
- Shelters will test the log-in system, check if they are able to view their inventory, and verify that they are able to request food items.
- Grocery stores will test the log-in system, check if they are able to view their inventory, and verify that they are able to schedule deliveries.

Acceptance Testing

- Acceptance Criteria
  - Any reported bugs or system inconsistencies should be fixed
- Acceptance Tests
  - The tests below will be run with a demo version of the website. The demo will be released publicly if these tests run successfully.
  - Grocery stores and homeless shelters should be able to log in successfully with a token provided by the admin.
  - After logging in, shelters will be able to allow their location to be tracked and view the nearest grocery stores. Grocery stores will be able to view the nearest shelters, and a database of their employees who are in a position to deliver next.

  - Shelters should be able to request food items to grocery stores of their choice and respective grocery stores should be able to view the requests in real-time.
  - Grocery stores should be able to designate employees for delivery preparation and share tracking information with the shelters.

11. QUALIFICATION PROVISIONS

This document will be reviewed regularly by the team for correctness, ambiguity, completeness, consistency, modifiability, stability, verifiability, modifiability, and traceability. Every member will conduct an initial read-through of the document followed by a detailed review of the sections each member is responsible for. The final part of the review process will include every team member performing a peer-review of sections written by other members of the team. The initial read-through will consist of assessments for consistency, completeness, and general readability. If errors are noticed, they will be fixed immediately before the detailed inspection of the more specific items in the document. The detailed reviews that follow will focus on modifiability, stability, and verifiability. Team members will work together on peer-reviews so that errors are identified efficiently.

12. REQUIREMENTS TRACEABILITY

Requirement	Description	Artifact first present in	Creation Date	Modific-ation Date	Latest Artifa-ct
Functional Requirements	Services the system should provide, how the system should react to particular inputs and situations. Part of user requirements.	SRS 2.0	March 16, 2021	April 8, 2021	SRS 3.0
Non-functional Requirements	Constraints on the services or functions offered by the system.	SRS 2.0	March 16, 2021	April 8, 2021	SRS 3.0
Business Requirements	Business drivers for the system	SRS 2.0	March 16, 2021	April 8, 2021	SRS 3.0

13. EVOLUTION OF THE SRS

As the system development process progresses, this document will serve as a backbone. In the future, the SRS will be more flushed out because there will be more background information available on the various functionalities. The Hunger Warrior application will be accessible through the web and has the goal of connecting shelters with grocery stores to significantly minimize food waste. In the future, there may be a mobile application for tablets in order to aid in ease of use. Additional changes to the SRS will result from deficiencies, shortcomings, inaccuracies, or changes in the system environment.

14. RATIONALE

None

15. NOTES

None

16. APPENDICES

Schedule Tracking

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Domain	Gavin Senger	6 hours	5 hours	1 hour(s)
	Sanidhya Sitaula	6 hours	5 hours	1 hour(s)
	Kori Vernon	3 hours	5 hours	2 hour(s)
	Courtney Battieste	3 hours	3 hours	0 hour(s)
	Team Summary	18 hours	18 hours	

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Requirements	Gavin Senger	4 hours	3 hours	1 hour(s)
	Sanidhya Sitaula	4 hours	5 hours	1 hour(s)
	Kori Vernon	4 hours	5 hours	1 hour(s)
	Courtney Battieste	3 hours	3 hours	0 hour(s)
	Team Summary	15 hours	16 hours	

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Analysis	Gavin Senger	3 hours	3 hours	0 hour(s)
	Sanidhya Sitaula	3 hours	3 hours	0 hour(s)
	Kori Vernon	3 hours	3 hours	0 hour(s)
	Courtney Battieste	3 hours	3 hours	0 hour(s)
	Team Summary	12 hours	12 hours	0 hour(s)

	Who	Estimated	Actual	Difference
Cumulative	Gavin Senger	13 hours	11 hours	2 hours
	Sanidhya Sitaula	13 hours	13 hours	0 hours
	Kori Vernon	10 hours	13 hours	3 hours
	Courtney Battieste	9 hours	9 hours	0 hours
	Team Summary	45 hours	46 hours	1 hour

Defect Tracking

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Domain	Gavin Senger	2 hours	2 hours	0 hour
	Sanidhya Sitaula	2 hours	2 hours	0 hour
	Kori Vernon	2 hours	2 hours	0 hour
	Courtney Battieste	2 hours	2 hours	0 hour
	Team Summary	8 hours	8 hours	0 hour

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Requirements	Gavin Senger	1 hour	1 hour	0 hour
	Sanidhya Sitaula	1 hour	1 hour	0 hour
	Kori Vernon	1 hour	1 hour	0 hour
	Courtney Battieste	1 hour	1 hour	0 hour
	Team Summary	4 hours	4 hours	0 hour

Artifact or Deliverable	Who	Estimated	Actual	Difference
SRS - Analysis	Gavin Senger	2 hours	2 hours	0 hour
	Sanidhya Sitaula	2 hours	2 hours	0 hour
	Kori Vernon	2 hours	2 hours	0 hour
	Courtney Battieste	2 hours	2 hours	0 hour
	Team Summary	8 hours	8 hours	0 hour

	Who	Estimated	Actual	Difference

Cumulative	Gavin Senger	5 hours	5 hours	0 hour
	Sanidhya Sitaula	5 hours	5 hours	0 hour
	Kori Vernon	5 hours	5 hours	0 hour
	Courtney Battieste	5 hours	5 hours	0 hour
	Team Summary	20 hours	20 hours	0 hour

17. DICTIONARIES

Classes

Name	Description	Methods	Attributes
Admin	Adds stores and shelters to the database	addShelter(), addStore()	None
MapDisplay	Displays map	None	None
Shelter	Contains information about shelter	requestFoodItem()	shelterID, Name, Address, shelterAvatar, ContactInformation(False, shelterID)
GroceryStore	Contains information about grocery store	dispatchFoodItem()	groceryStoreID, Name, Address, storeAvatar, ContactInformation(True, groceryStoreID), availableFoodItems, donationHistory
InformationDisplay	Shows available stores and shelters in the area	getGroceryStoresInArea(), getSheltersInArea()	area, storeList, shelterList
Donation	Holds donation details	getDate(), calculatePrice()	foodItemsDonated
FoodItem	Details about the food item donated	None	uniqFoodID,price, quantity
TaxInformation	Shows tax information	None	shelterID
FoodItemReport	Shows additional information about a food item	getExpiryDate()	uniqFoodID, itemPicture, deemedSafe

Match	Matches shelter with nearest grocery store	calculateDonationTotal()	GroceryStore, Shelter, donationHistory
ContactInformation	Holds additional information about shelter/store	None	isGroceryStore, uniqID, contactNumber, contactName

Methods

Name	Description	Class	Arguments
addShelter()	Add shelter to the database	Admin	Shelter
addGroceryStore()	Add grocery store to the database	Admin	GroceryStore
requestFoodItem()	Request a food item from a store	Shelter	FoodItem
dispatchFoodItem()	Dispatch a food item for delivery	GroceryStore	FoodItem
getGroceryStoresInArea()	Get all grocery stores within an area	InformationDisplay	area
getSheltersInArea()	Get all shelters within an area	InformationDisplay	area
getDate()	Get the date of donation	Donation	None
calculatePrice()	Calculate the total price of donation	Donation	foodItemsDonated
getExpiryDate()	Get the expiry date of a food item	FoodItemReport	FoodItem

Attributes

Name	Description	C/S	Type	Size	Attributes
shelterID	Unique shelter ID	S	int	64 bits	None
groceryStoreID	Unique store ID	S	int	64 bits	None
Name	Name of shelter/store/fooditem	S	string	32 bits	None
Address	Address of shelter/store	S	string	32 bits	None
area	Current area	C	N/A	N/A	Longitude, Latitude
availableFoodItems	List of available food items	S	array	N/A	FoodItem
shelterAvatar, storeAvatar, itemAvatar	Avatars of shelter/store/food	S	Binary Large Object	N/A	None
price	Price of food item	S	int	64 bits	None
isGroceryStore	Indicate if the object is a grocery store	S	bool	1 bit	None
deemedSafe	Indicate if the food item is deemed safe	S	bool	1 bit	None

Relationship

Name	Description	From Class	To Class	Optional/Mandatory	Cardinality
Provides Information	Provide information from InformationDisplay to display the map	MapDisplay	InformationDisplay	Mandatory	1 to 1
Provides Information	Take shelter/grocery store information and transfer it to be mapped	Shelter/ GroceryStore	InformationDisplay	Mandatory	Many to 1
Provides Information	Provide donation information to calculate tax benefits	Donation	TaxInformatino	Mandatory	Many to 1
Manages Information	Add / remove grocery stores and shelters	Admin	Grocery Store/Shelter	Mandatory	1 to many
Provides Information	Take information about shelters and stores and match them	ContactInformation	Match	Mandatory	Many to 1
Provides Information	Provide tax information to the groceries	TaxInformation	GroceryStore	Mandatory	1 to 1