Super Shopping List Software Requirements Specification Document

1 Introduction

1.1 Purpose

This document outlines the specific requirements of the Super Shopping List. It identifies the features and requirements of those features, the requirements with regard to space, time, and data constraints, as well as design constraints like UML diagrams and UI designs.

This document is intended for developers of the app, so they may appropriately and completely implement features. Also for stakeholders of the app so that they may be in the loop as to what *exactly* will be implemented. Should there be questions as to an implementation / feature the developer or stakeholder should reference this document and then contact the party who may have an answer should this document fail to do so.

1.2 Scope

The scope of The Super Shopping List encompasses the user being able to view public recipes and create their own, select those recipes for shopping, view a shopping list, and then take all of that on the go with another device.

Things that are out of scope include any sort of transaction made through the app - users will not be able to purchase groceries on the app or share their shopping lists with others for them to shop for them.

1.3 Definitions

- **Ingredient** - a named item with an amount and unit.
- **Recipe** - a named collection of ingredients and a free text instruction list.
- **Meal plan** - selection of recipes.
- **Shopping list** - a list of ingredients partially generated from meal plan with the potential for extra items

1.4 References

IEEE 830-1993 - Recommended Practice for Software Requirements Specifications

Kaggle Recipe Dataset: https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions?select=RAW_recipes.csv

1.5 Overview of SRS

This document specifically identifies all of the requirements for the software. It is organized as follows:

- Section 2: specifies the design of the software
    - UML diagrams
    - Use case diagrams
    - User interface designs
- Section 3: specific requirements
    - Specifies hardware requirements
    - Specifies functional requirements by system feature
    - Specifies non-functional requirements
- Section 4: supporting information for this document like
    - Table of contents
    - Index
    - Incremental implementation plan
    - Contribution info

2 Overall Description

Creating a shopping list for the week can be a pretty big undertaking when thinking about what you want to eat, what each of those things require, what ingredients are common to both, and then checking what things you already have on hand. The Super Shopping List will do all of these tasks automatically making the creation of a shopping list as simple as selecting some recipes for a few days, a week, or even a month and then going to the store with a mobile device.

2.1 Product Perspective

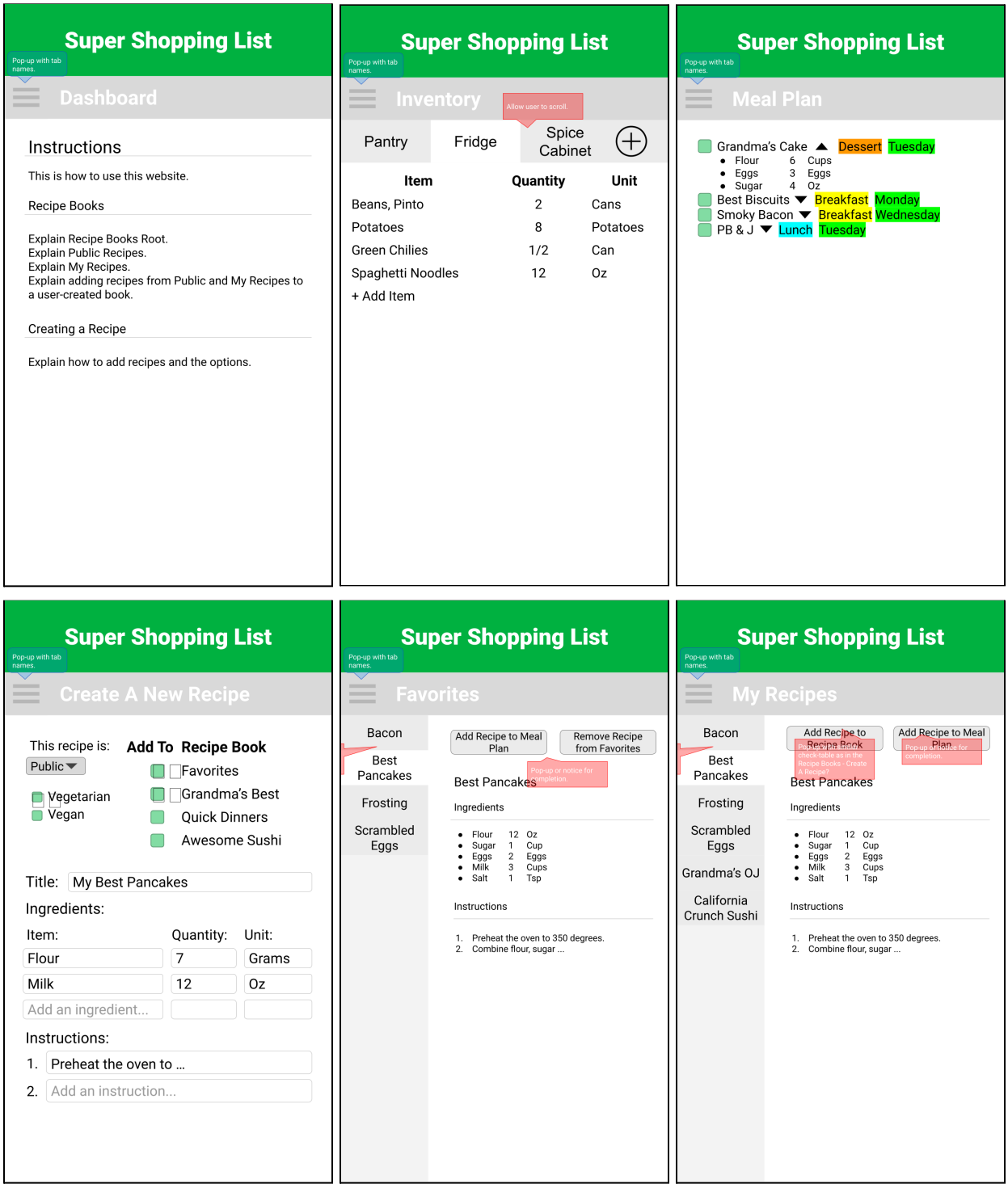
This product is being built for desktop and mobile devices and will utilize a SQL database for user's data. It is not connected to any existing software product outside of the tools that are used to build it. It doesn't depend on having another piece of software or an account with some other software.

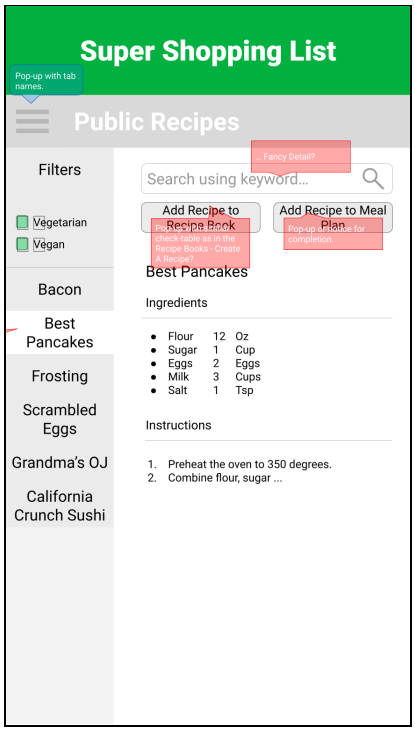
2.1.1 System Interfaces

The system interfaces are a windows, mac, or linux desktop for the desktop app. For mobile, the system interface is an android phone.

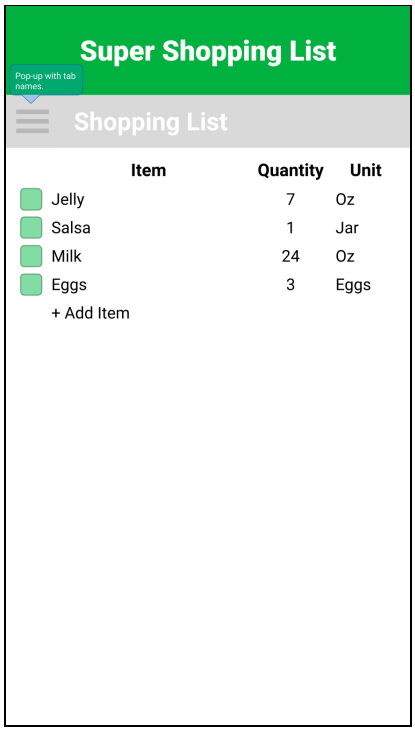
2.1.2 User Interfaces

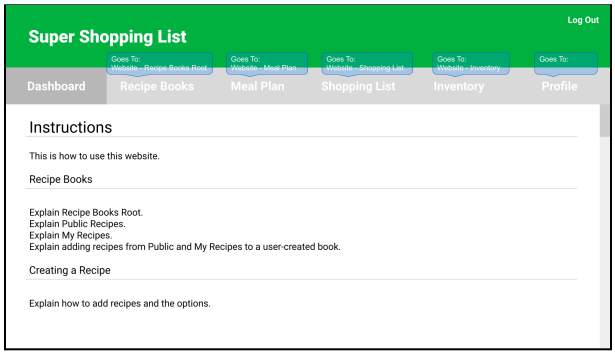
The full-sized files can be found in the GitHub repo, in the planning_documents > ui_sketches directory.

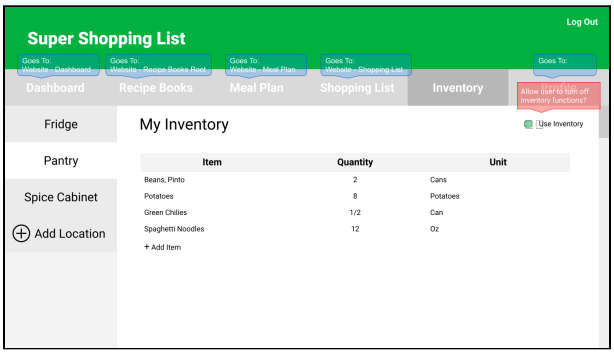


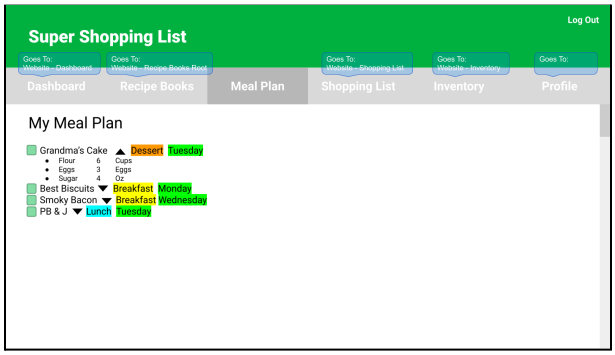

Super Shopping List
Pop-up with tab names.
Public Recipes
Filters
Vegetarian
Vegan
Bacon
Best Pancakes
Frosting
Scrambled Eggs
Grandma's OJ
California Crunch Sushi
Search using keyword...
Fancy Detail?
Add Recipe to Recipe Book
Add Recipe to Meal Plan
Pop-up to choose for completion.
Best Pancakes
Ingredients
Flour 12 Oz
Sugar 1 Cup
Eggs 2 Eggs
Milk 3 Cups
Salt 1 Tsp
Instructions
1. Preheat the oven to 350 degrees.
2. Combine flour, sugar ...

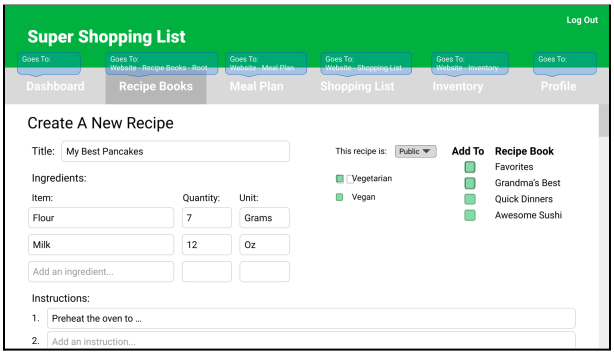

Super Shopping List
Pop-up with tab names.
My Recipe Books
Goes To: Website - Recipe Books Public Recipes
Public Recipes
Goes To: Website - Recipe Books My Recipes
My Recipes
Goes To: Website - Recipe Books - Favorites
Favorites
Grandma's Best
Quick Dinners
Awesome Sushi
Creates a new recipe book. Just create a new book of the name rather than a wole new page?
Add A Recipe Book
Goes To: Website - Recipe Books - Create A Recipe
Create A Recipe

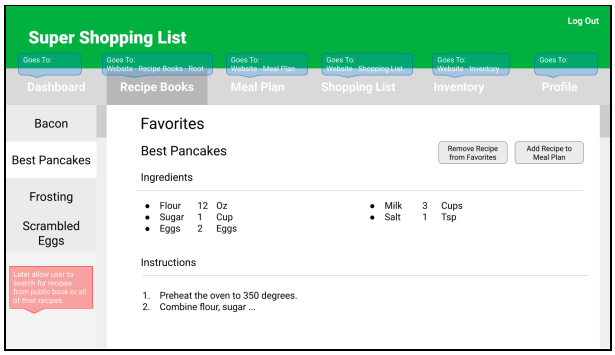

Super Shopping List
Pop-up with tab names.
Shopping List
Item
Quantity
Unit
Jelly 7 Oz
Salsa 1 Jar
Milk 24 Oz
Eggs 3 Eggs
+ Add Item

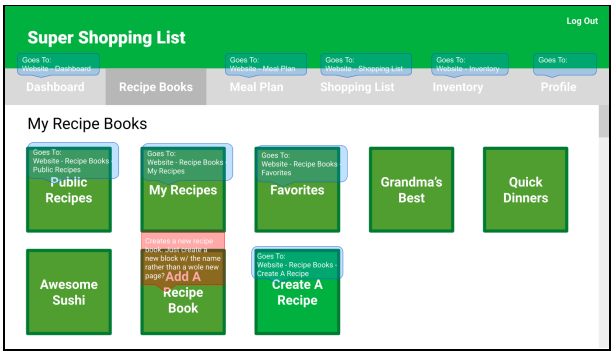

Super Shopping List
Log Out
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
Instructions
This is how to use this website.
Recipe Books
Explain Recipe Books Root.
Explain Public Recipes.
Explain My Recipes.
Explain adding recipes from Public and My Recipes to a user-created book.
Creating a Recipe
Explain how to add recipes and the options.

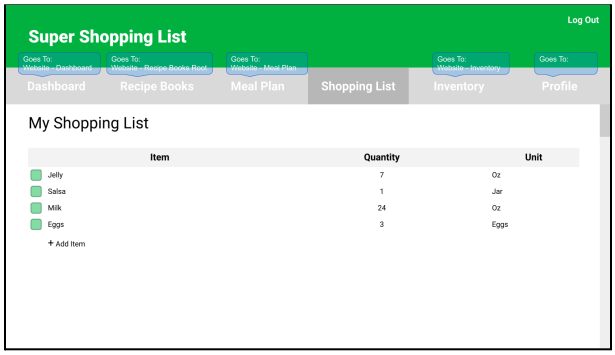

Super Shopping List
Log Out
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
Fridge
Pantry
Spice Cabinet
Add Location
My Inventory
Use Inventory
Item
Quantity
Unit
Beans, Pinto 2 Cans
Potatoes 8 Potatoes
Green Chilies 1/2 Can
Spaghetti Noodles 12 Oz
+ Add Item


Super Shopping List
Log Out
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
My Meal Plan
Grandma's Cake Dessert Tuesday
Flour 6 Cups
Eggs 3 Eggs
Sugar 4 Oz
Best Biscuits Breakfast Monday
Smoky Bacon Breakfast Wednesday
PB & J Lunch Tuesday

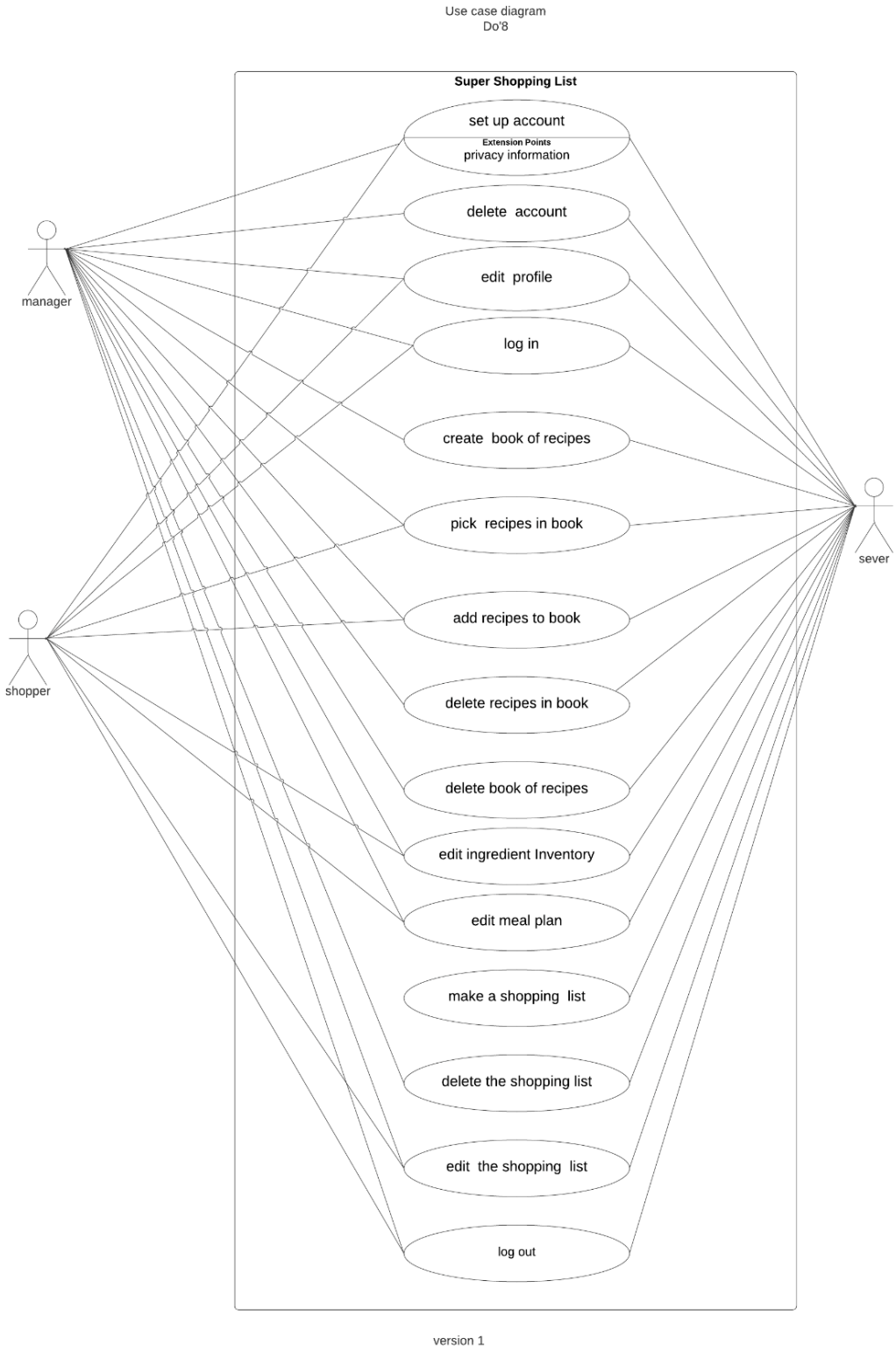

Super Shopping List
Log Out
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
Create A New Recipe
Title: My Best Pancakes
Ingredients:
Item:
Quantity:
Unit:
Flour 7 Grams
Milk 12 Oz
Add an ingredient...
This recipe is: Public
Vegetarian
Vegan
Add To
Recipe Book
Favorites
Grandma's Best
Quick Dinners
Awesome Sushi
Instructions:
1. Preheat the oven to ...
2. Add an instruction...


Super Shopping List
Log Out
Goes To:
Goes To: Website - Recipe Books - Root
Goes To: Website - Meal Plan
Goes To: Website - Shopping List
Goes To: Website - Inventory
Goes To:
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
Bacon
Best Pancakes
Frosting
Scrambled Eggs
later allow user to search for recipes from public book or all of their recipes.
Favorites
Best Pancakes
Remove Recipe from Favorites
Add Recipe to Meal Plan
Ingredients
Flour 12 Oz
Sugar 1 Cup
Eggs 2 Eggs
Milk 3 Cups
Salt 1 Tsp
Instructions
1. Preheat the oven to 350 degrees.
2. Combine flour, sugar ...


Super Shopping List
Log Out
Goes To: Website - Dashboard
Goes To: Website - Meal Plan
Goes To: Website - Shopping List
Goes To: Website - Inventory
Goes To:
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
My Recipe Books
Goes To: Website - Recipe Books - Public Recipes
Pubiic Recipes
Goes To: Website - Recipe Books - My Recipes
My Recipes
Goes To: Website - Recipe Books - Favorites
Favorites
Grandma's Best
Quick Dinners
Awesome Sushi
Creates a new recipe book. Just create a new block w/ the name rather than a wole new page?
Add A Recipe Book
Goes To: Website - Recipe Books - Create A Recipe
Create A Recipe


Super Shopping List
Log Out
Goes To: Website - Dashboard
Goes To: Website - Recipe Books Root
Goes To: Website - Meal Plan
Goes To: Website - Inventory
Goes To:
Dashboard
Recipe Books
Meal Plan
Shopping List
Inventory
Profile
My Shopping List
Item
Quantity
Unit
Jelly 7 Oz
Salsa 1 Jar
Milk 24 Oz
Eggs 3 Eggs
+ Add Item

2.1.2.1 Use Case Diagram



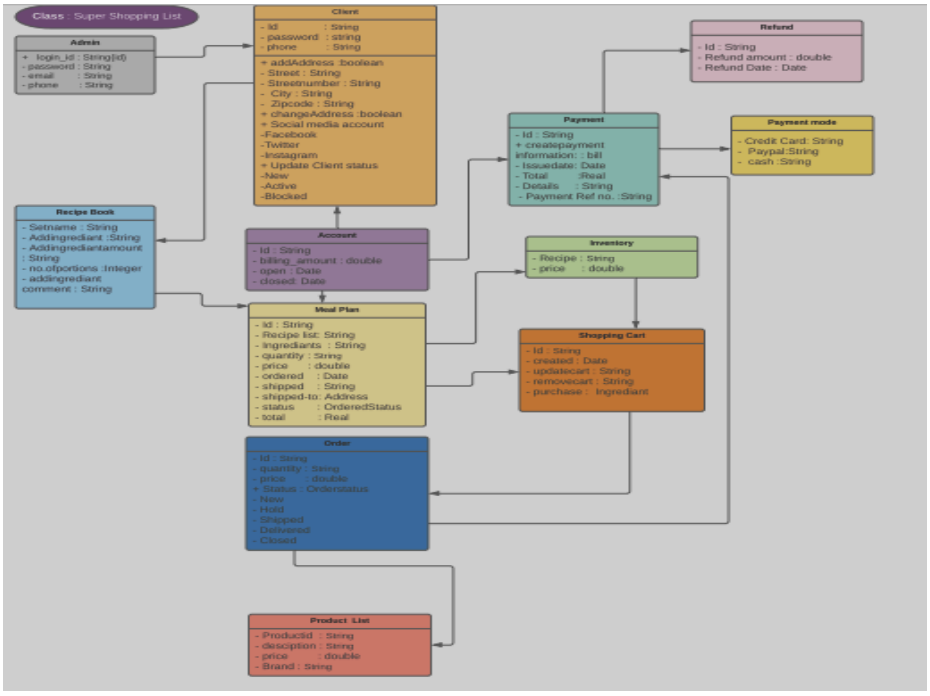
The system has three actors: a manager (or system administrator), the shoppers (or the users), and the system server (database). A shopper can set up an account, edit their profile, log in to their account, pick a recipe from a collection (book), edit their ingredient inventory, edit their meal plan, and log out of their account. A manager is considered a user with elevated privileges. Managers can do everything a user can but can also delete accounts, and delete recipes from the database. When a user creates an account they must input a unique username and password. This updates the database with a new user account once the username is verified as unique. A user can later edit their username and password. Edit either goes through the same verification process as creating an account. When logging in, a user must provide both a username and password. These are used to authenticate the user. If the user is authenticated, then the user is logged in. If the user is not authenticated, an login error message will display and the user will have to try again. When editing the ingredient inventory, a user may add or remove items. When editing a meal plan, a user may add or remove recipes. Whenever a change is made (e.g. profile change or recipe change) the server deals with the interaction to update the database if necessary.

2.1.3 Hardware Interfaces

No specific hardware interfaces required.

2.1.4 Software Interfaces

UML Class Diagram:



2.1.5 Communications Interfaces

No specific communication interface required. Communication with the server uses configured networking on the host.

2.1.6 Memory Constraints

The system should not leak any memory - since we are using a garbage collected language, this should be no issue.

2.1.7 Operations

Standard operating procedure is defined by way of the feature set in section 3.

2.1.8 Site Adaptation Requirements

No specific SARs for the system.

2.2 Product Functions

Put simply, the user will have a bank of recipes that they can browse and edit. From that, they can select some of the recipes that they want to shop for. From that selection, a shopping list will be automatically generated. They can then take this shopping list to the grocery store and shop with it.

The user will be able to plan meals on a calendar as well.

The user will be able to specify an inventory so that the shopping list that gets generated will not tell them to buy something that they already have.

2.3 User Characteristics

Intended users are those that go to grocery stores with shopping lists or those that want to start using shopping lists.

2.4 Constraints

2.4.1 No Budget

This project has to be done entirely for no monetary input on part of the developers (the only stakeholders). This means exclusively utilizing free tools and tools we can get access to with student emails.

2.5 Assumptions and Dependencies

No specific dependencies other than the specific hardware and operating systems we have chosen to support.

2.6 Apportioning of Requirements

All requirements in this document are expected to be implemented by the delivery date. Any additional requirements are beyond the scope of this project.

Should something catastrophic happen and all the requirements cannot be met, we will adjust this document and the project scope accordingly

3 Specific Requirements

3.1 External Interface Requirements

The external interface requirements describe the inputs into the system.

A. Keyboard
    a. The user shall interface with inputting text through keyboard whether it be mobile or desktop
    b. Source of input is keyboard, output is whatever selected form / input box the user chose.
    c. Valid text is any UTF-8 character

B. Monitor
    a. The user shall view all the user interfaces through a screen when using the desktop application. All the components of the system are communicated through a responsive graphical user interface.
    b. No input through monitor (on desktop).

C. Mobile Screen

a. The user shall interface with the UI by tapping and scrolling in the application on their mobile device.
b. User interface output will be through a responsive graphical user interface rendered on the mobile device.

3.2 Functional Requirements (by system feature)

3.2.1 Meal Plan

In the meal plan instance, the user has the ability to select multiple recipes from the recipe books to cook for the next selected days, which will be added to the meal plan of which the needed ingredients will be added to the shopping list. It will also have the ability to add tags to the meals - whether it is a breakfast, lunch or a dinner. Additionally, the system shall have a check off function to display whether the meal was cooked.

3.2.1.1 Purpose

The meal plan exists so that the user can select / add / delete entire recipes from their shopping list without interfacing with actual ingredients.

3.2.1.2 Add recipe From Recipe Book

The system shall add recipes that the user specifies in the recipes interface to the meal plan (see requirement 3.2.3.7).

3.2.1.3 Edit Serving Size

The system shall allow the user to change the amount of servings (like half or double) on a particular recipe and have the ingredients be modifiable.

3.2.1.4 Edit Recipes

The system shall allow recipes to be modified in the meal plan without modifying the recipe in the book - this amounts to a one-time change made to the recipe.

3.2.1.5 Delete Recipes

The system shall have a button that allows the user to remove a recipe from the meal plan.

3.2.1.6 Add Recipe from Meal Plan View

The system will allow the user to input a recipe directly into the meal plan via a form.

3.2.2 Shopping List

Our applications will also include a shopping list which are disparate for every user as they are created from their meals plans with having the current inventory of the user in consideration. An example being if a user has a cheesecake in their meal plan and has everything required in their inventory except the two needed eggs, the system will add two eggs to their shopping list. The user has the ability to add extra things to the shopping list which are totally unrelated to the actual recipe like paper towels. Once the user shops for the items in their shopping list and checks them off, the item is added to the inventory.

3.2.2.1 Purpose

The shopping list is what the user interfaces with while they do the actual shopping. This is a checkable list that is automatically generated from the meal plan ingredients, their amounts, and the amount of servings the user specifies.

3.2.2.2 Automatically Generated from Meal Plan

The system shall build the shopping list based off of the ingredients in the recipes of the meal plan. The amounts will be based on the servings for the recipe. Ingredients not currently present in the inventory, or with lower count than needed will be added to the shopping list.

3.2.2.3 Add Items to List

The system shall allow the user to add an item to the list directly.

3.2.2.4 Delete Items from List

The system shall allow the user to delete items from the list directly. Note that this functionality is different from requirement 3.2.2.6 in that this is not marking an item as bought (and thus it doesn't invoke requirement 3.2.4.3).

3.2.2.5 Modify Ingredients Quantities

The system shall allow the user to modify the ingredient quantities from the list view as well as through the edit recipe functionalities (requirements 3.2.3.2, 3, 4).

3.2.2.6 Checklist Functionality

The user can check off items when they acquire them at the grocery store. The system shall remove this item from the list and update the according data structures in the system and in the database (like in the inventory - see requirement 3.2.4.3).

3.2.3 Recipes

As recipes being one of the main features of our project, it is going to have multiple attributes. There are going to be two kinds of recipe books, first being the users' recipe book - which will have recipes posted by the users (each recipe has the option to be shared publicly) - and the second being the recipes uploaded by the administrators. Admins will have the ability to edit all recipes, either being from the public's or server's recipe books. A user that creates the recipes for the public book, has access to a separate tab where they can view the recipes they created and will have the authority to edit or delete them. Additionally, the users can add recipes to their meal plans.

3.2.3.1 Purpose

The system shouldn't burden the user with rewriting a recipe each time - they should simply select a recipe from some database of recipes (both public and private to the user). This makes the building of their shopping list for, say a week's worth, the shopping trip.

3.2.3.2 Add Ingredient

The user shall be able to add ingredients to the recipes.

3.2.3.3 Remove Ingredient

The user shall be able to remove ingredients to the recipes.

3.2.3.4 Create New Recipes

Users shall be able to create a recipe and store it in the recipe book. They can declare each recipe "public" or "private".

3.2.3.5 Public Recipes

The administrators of the app shall be able to add recipes to a public database where common recipes are available to anyone. This will also show recipes declared public by users along with their username.

3.2.3.6 Saving Recipes

The user shall be able to save recipes from the public database so that they show up in the recipes tab for selection. The system shall have a button attached to each recipe that indicates "save" or "saved".

3.2.3.7 “Add to Meal Plan” Button

The system shall have buttons attached to each recipe that indicate “add to meal plan” or “added to meal plan”.

3.2.4 Inventory

The inventory consists of the items a user has available at their home. The inventory starts off as an empty list when the user creates their account. The inventory can be manually edited. Once the user purchases any of the items from their shopping list, it is added to the inventory automatically. Additionally, when meals are cooked, the used ingredients are removed from the inventory.

3.2.4.1 Purpose

The user is likely to have a cabinet / pantry / refrigerator that holds items they don’t buy often and keep a lot of at once like spices, eggs, or milk. We don’t want to add things that they already have to the shopping list.

3.2.4.2 Add / Delete Inventory Items

The system shall allow the user to add items and remove items from the inventory list.

3.2.4.3 Automatic Checkoff Inventory Update

When users check off items from the shopping list, the amount/volume of items they got will automatically be added to inventory.

3.2.5 Calendar

The calendar view of the meal plan allows for the user to interface with their meal plan in a calendar based way. That is, the calendar will display days of the week in a grid with the planned meals on the calendar according to their set date. The user will be able to interface with this calendar to add meals to the meal plan rather than using the list interface. The calendar will update automatically when the meal plan is edited from the hierarchical list view and when the calendar is edited, it will message pass for the meal plan to be updated.

3.2.5.1 Purpose

A calendar view of the meal plan is very easy to parse and edit for the user. People plan out their week, fortnight, or even month at a time. Seeing the planned meals all in order on a calendar is convenient.

3.2.5.2 Allow Adding / Deleting to Meal Plan from calendar

From the calendar interface, the user shall be able to click a day and add a meal to the meal plan. They will be able to edit the meal plan in the same way they normally would and the date will auto-fill based on the day they select.

3.2.5.3 Moving of Dates

The user shall be able to pick up and drag a meal plan to a different day without having to edit the contents. The mouse should change icons to indicate this functionality to the user.

3.2.5.4 Up to Date with Meal Plan and Shopping List

The system shall treat the calendar as an interactive *view* of the meal plan - thus, the shopping list will update just the same as it would when the user interfaces with the hierarchical list view of the meal plan.

3.2.6 User Account

The applications allow users to create accounts, for which they have to provide a preferred username and a password. The mobile and desktop applications will share a database and hence the user can use their account for either of the applications. The information about their created recipes, inventory, etc will be stored in the database.

3.2.6.1 Purpose

The user will have their shopping list, meal plan, and recipes synced between mobile and desktop interface so that means we need to associate all their data between the two devices. The simplest way to implement this is to have them make an account and associate it with that.

3.2.6.2 Create Account or Log In

When a user opens the app, the system shall prompt them to log in or create a new account. Logging in option will take them through a typical username / password form. Creating an account will take them through a typical email / username / password / confirm password form.

3.2.6.3 Update Account

The system shall have an account page where the user can see their email, username, and password with the option to change their email and password.

3.2.7 Semi-Live Update

All data for a user will be available on all of their devices that they use the Super Shopping List on. Whatever they see on their desktop or laptop, they will see on their mobile device.

3.2.7.1 Purpose

The idea with this app is that the user can build the list on their desktop and then take their mobile phone to the store. This requires that the data displayed on the mobile device is up to date with what was displayed on the desktop version.

3.2.7.2 Update

The system shall send requests to the server to update the user’s shopping list upon changes made to the meal plan or shopping list. The system should wait say 5 seconds, before sending this request should the user make another change. The system, in that case, will send a batch of changes to be made to the server.

3.3 Non-Functional Requirements

3.3.1 Space Requirement

The system app must be under 250 Mb.

An exception could be made for the desktop version in the event that we need to package something along with our executable (for example, if it needs to ship with embeddable python, we will package that up with the app).

3.3.2 Time Requirements

Pages should load on the order of milliseconds.

Updates to local lists should be on the order of milliseconds - when a user updates the meal plan, the list should update nearly instantly.

Updates to the user's cloud should be on the order of seconds - we relax this requirement with respect to the above as we have to account for networking.

3.4 Logical Database Requirements

The database will be used by the meal plan, shopping list, and recipe features. The shopping list will access the database to update itself as well as push updates when the user makes changes. The meal plan behaves similarly.

The public recipes are shown to the user in the recipes tab. The recipes that the user creates and manages will push and pull updates from the database as well.

During a session, there will be many updates to make. The frequency of updates largely depends on how fast the user adds new items to their meal plan / list.

Data integrity is maintained by user authentication - only a user can edit their own meal plans, shopping lists, and recipe books (see requirement 3.6.3).

This system has no data retention requirements. None of the data stored is sensitive and there are no requirements for using the data for a separate system or selling the data to another company.

3.5 Design Constraints

There are no standards that we need to adhere to for the project. Nor are there any design constraints.

3.6 Software System Attributes

3.6.1 Reliability

System crashes should be infrequent (fewer than 5 reports per month). They should be handled gracefully such that the user stays informed about the problem as well as the developers so that we may fix the issue.

3.6.2 Availability

The system should support at least 8 users. The database should be easily scalable by utilizing cloud technology.

The system server will be run off of a machine belonging to a developer - the requirements are such that the online update service is only available during demos / tests.

3.6.3 Security

Only the user who owns the data can change the data in their database. Furthermore, while the data is in the database, it will not be lost or corrupted.

3.6.4 Maintainability

The system should be maintainable in terms of code and database management. The code should be modular enough that future work could be done to add features to it. The database system should be such that user and system data can be easily updated.

3.6.5 Portability

The desktop system should compile to linux, mac, and windows machines. The mobile system should support common versions of android later than kitkat.

4 Supporting Information

4.1 Table of Contents

4.2 index

4.3 Appendices

4.3.1 Incremental Development Plan

Feature	Phase	Task	Description	Technologies
Meal Plan	3	Add recipes to database	Go through Kaggle dataset and upload recipe and ingredients to database one by one.	Python, SQL
Meal Plan	2	Allow user to add custom recipe	Customizing the recipe from the existing recipes from Uploaded database one after other.	Python, Kotlin
Meal Plan	1	Allow user to edit recipe	Modifications from the selected recipe either adding a new ingredient	Python, Kotlin

			from existing recipes	
Meal Plan	1	Allow user to delete recipe	Deletes the selected recipe or item the list.	Python, Kotlin
Shopping List	1	Automatically generate shopping list (3.2.2.2)	Go through ingredients of a specific recipe, cross verify across inventory items to determine if user is running low on a specific item, if they are add that item to the shopping list.	Python, Kotlin, SQL
Shopping List	1	Add item to list	Insert directly to shopping list	Python, Kotlin,
Shopping List	2	Delete item to list	Delete directly from shopping list	Python, Kotlin,
Shopping List	3	Modify ingredient quantity	Modify amounts generated from meal plan	Python, kotlin
Shopping List	2	Checklist Functionality	Allow users to check off items when they are at the supermarket	Python, Kotlin
Recipe	3	User can declare recipe public or private	This will allow users to share or hide their recipes from all app users	Python, Kotlin
Recipe	1	Allow admins to add recipe to global recipe database	Admin users can add the recipe to the global database which reflects in the app globally to the end user	Python, Kotlin, SQL
Recipe	1	Add recipe to meal plan	This allows to add a new or modified meal plan to the recipe database, where end user can select the plan to their carts	Python, Kotlin

Recipe	2	Save Recipes	This allows to save the selected plans or can be favorited, that can be in their wish lists to review further	Python, Kotlin, SQL
Inventory	2	add/delete items from inventory	This allows to add or delete selected plans or items from the cart and can be modified at any instance	Python, Kotlin, SQL
Inventory	2	Automatic Inventory Update	When user checks off item from checklist while shopping, that items is added number of items in inventory	Python, Kotlin, SQL
User Account	2	Create User Account	End user can access the app using their personal information and can gain access to browse the app with secured credentials.	Python, Kotlin, SQL
User Account	2	Update User account	Give user option to change email and password	Python, Kotlin, SQL

4.3.1 Team Contributions

Member name	**Contribution description**	**Overall Contribution (%)**	**Note (if applicable)**
Mustafa Memon	Functional Requirements (Recipes, meal plan, shopping list, inventory, user account)	14	
Brice Brosig	Functional Requirements (calendar / some specifications), Non-functional Requirements, IEEE 830-1998 formatting	14	
Mica	UI Sketches, Functional Requirements (discussion), Non-functional Requirements (discussion)	13.5	

Tam Doan	Use Case Diagram	13	
Nestor Molina	Use Case Diagram description, Requirements discussion and reviewing	13	
Vandana Sinha	UML Class Diagram, User -Client and Admins, Recipes, Meal Plan, Shopping List, Inventory, Ingredients etc.	13.5	
Naveen	Filled some of the empty Incremental Plan Description columns	5	
Nikhil Gaur	Incremental Plan	14	