Super Shopping List

The Team

Do's 8

- Mica Haney – Team Lead; Web developer
- Brice Brosig – Web developer
- Mustafa Memon – Mobile Developer
- Nestor Molina – Database / Backend developer
- Tam Doan – Web developer
- Vandana Sinha – Database / Backend developer
- Nikhil Gaur – Mobile Developer
- Naveen Raja Yalagandula – ??

Project Description

Overview

The Super Shopping List is a solution to shopping that takes the ingredients and their amounts from your recipes and compiles a neat list of everything that you need based on what you want to eat. This project will let you make a meal plan for each day, and will let you know what you need to get based on how many days you are shopping for. This project is split into three parts. The web interface and the mobile interface both communicate with the same database to make creating and using the list very easy.

User Interface

This interface will be similar across both the website and the app in terms of the appearance. The website will be designed and rendered using HTML and CSS, perhaps using CSS frameworks like bootstrap to make aesthetics easy. For mobile, the user interface is built with XML. Design in terms of color, layout, and style should be consistent between the two platforms for seamless use.

The following are the different views of the user interface.

Recipe Book

A collection of recipes that the user can select recipes from. These recipes will include the ingredients and instructions. Users can create new recipes and save them in the recipe book or add recipes from a public database of recipes.

Meal Plan

These are the selected recipes for the user's shopping trip. For example, the user is planning their shopping for the week and they select 5 recipes from their recipe book that they plan to make that week. After selecting what they are added to the meal plan where they can be viewed and/or edited such as doubling or halving the servings. The meal plan can be displayed in some sort of tree hierarchy:

- Cake -- 8 servings
    - Eggs -- 2 -- unit
    - Milk -- 1 -- cup
    - Flour -- 12 -- oz
    - How to make cake...

Note: that the eggs have a unit of "unit" - this is a placeholder for some sort of denotation of an atomic item (you can't have half an egg or something, so the unit is… egg).

Shopping List

The shopping list is the compiled ingredients from the Meal plan. Any ingredients that were shared amongst different recipes are accumulated into one item. If recipes call for "double" or "half" then the ingredient amounts are adjusted accordingly.

It will be displayed as a list with 3 columns: ingredient, quantity, unit. Alongside this, there will be a checkbox that they interact with to indicate they picked up the item.

The shopping list will be filled based on ingredients that are insufficient or not in the user's inventory for a certain recipe. For example if you only have 1 cup of pizza sauce and intend to make cheese pizza, and the recipe requires 3 cups, the app will automatically put pizza sauce in your grocery list.

Calendar View

A user will be able to add what meals they want at mealtimes throughout a specific day, and any ingredients that are not present in the user's inventory will be added to the shopping list.

Ingredient Inventory

Users can keep track of what items they currently have and the amounts (when possible). This is a list of the items they have in their fridge, pantry, or spice cabinet. When they generate a shopping list, they will not see any items that they already have. If they do not have enough of some item, then it will put the difference that they need for their selected recipes in the list.

Public Recipes

The users can look through a list of public recipes that they might like. They can add them to their recipe book and then later their meal plan.

User Details

This page is just for a user to manage account information; name, email, password change, etc. User accounts will be necessary to handle association of the lists between the users' different devices.

Website

The website interface for the Super Shopping List allows the user to create, add, delete recipes from the recipe book and/or meal plan. These, in turn, add items to the shopping list. The user can set a modifier for how much of a recipe they want (such as double or half), and the system will populate in the shopping list with the quantities appropriate to the number of servings for the recipe. The app will also update the shopping list based on whether there are too few ingredients in the user’s inventory for a certain recipe about to be made on a certain day, or for a certain week.

This app will be built with Python and Flask for the backend and “business logic”, HTML/CSS for rendering pages, and will use Javascript should a website component need to be interactive (Python will be favored over Javascript when possible). CSS will likely use flexbox as we want this site to be device and screen agnostic. Use of Bootstrap and Sass can be employed to make life with CSS a bit easier.

Mobile App

We will use Android studio to build this app with Java and Kotlin. We plan to use Kotlin because it allows us to use multiple features that Java does not allow. Since Kotlin can be known as the more simplified version of Java, we opt to use it more than Java. For creating the User Interface for this application, we will be using XML.

Database

We will be using SQL to create the database tables and entities and this database will be hosted using SQLite. The database is a key component in that it will be what enables the user to create lists on whichever device they want and then use that same list on the device they bring to the store with them. This means we will keep a table of user accounts, their recipe book, and their meal plan / shopping list.

Risk Management

Team members get COVID19 and are unable to work effectively or drop the class.

- Probability: Medium
- Impact: Serious
- Monitoring: Hold regular meetings and keep asking if everyone feels fine.
- Mitigation: Reducing the size and/or scope of the project. Change the requirements to accommodate fewer hands. Drop app if too few people, have it purely browser-based.

Incompatible designs and/or implementations between members; i.e. miscommunication leads to two people implementing two components that will not communicate and/or interact properly

- Probability: Medium
- Impact: Serious
- Monitoring: Weekly meetings will allow for and encourage discussion of implementation details.
- Mitigation strategy: We will have an exhaustive discussion on the design of each piece of the project. These designs will be well-defined and well-documented so that referring to design will guide implementation.

Unable to host a live database, website, and/or app.

- Probability: Low
- Impact: Mild

- Monitoring: Address early on if there is some monetary cost associated with hosting or if there is a free way to do so.
- Mitigation: If it ends up not being possible or is too much of a hassle we can easily fall back on running the site on a local host and hosting a local instance of the database alongside.

Project Schedule

Milestones

Come to consensus on project details. Deadline: 9/27
Delegate Deadline: 9/28
Document the UI design, have mock-ups Deadline: 9/29
Have mock-ups of website and mobile application Deadline: 9/30
Database

1. Database - Collect initial recipes Deadline: 10/12
2. Database - Create tables Deadline: 10/13
3. Database - Host (SQLite) Deadline: 10/14

App

- Back end core functionality Deadline: 10/14
- Front end core functionality Deadline: 10/14

Website

- Front end core functionality Deadline: 10/14
    - All screens (html pages) complete
- Back end core functionality Deadline: 10/14
    - Screen transitions complete (links between pages / views work)

Testing Deadline: 10/14
Iterate Twice:

App

- Back end extra functionality Deadline 1: 10/28, Deadline 2: 11/18
- Front end extra functionality Deadline 1: 10/28, Deadline 2: 11/18

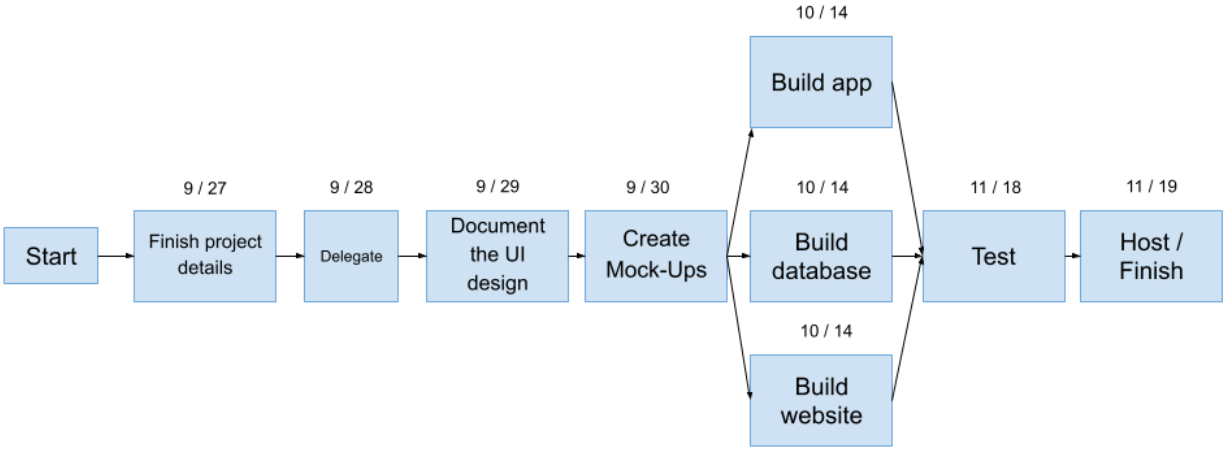
Website

- Front end extra functionality Deadline 1: 10/28, Deadline 2: 11/18
    - Add CSS / bootstrap
    - Adding interactive components to pages (if needed)
- Back end extra functionality Deadline 1: 10/28, Deadline 2: 11/18

Testing Deadline 1: 10/29, Deadline 2: 11/18

Testing Deadline 11/19
Host Deadline 11/19

PERT Chart



Gantt Chart

Team Contributions

Member name	Contribution Description	Overall Contribution (%)	Notes

Mica Haney	Schedule, Risk Management	14.5%	
Brice Brosig	Project Description, Risk Management	15%	
Mustafa Memon	Project Description, Schedule, Risk Management	14.5%	
Nestor Molina	Schedule, Risk Management	14%	
Tam doan	Schedule, Risk Management	14%	
Nikhil Gaur	Project Description, Schedule	14%	
Vandana Sinha	Schedule, Risk Management	14%	
Naveen Raja Yalagandula		0%	Joined late.