

Unified Grocery App

Software Design Document

Travis Tran & Joseph Yu

Massachusetts Academy of Math and Science at WPI

Worcester, MA

Table of Contents

Table of Contents	2
Statement of Goals	3
Functional Description — Minimal Viable Product (MVP)	3
Technical Data and Feasibility	4
User Interface	5
Structural Diagram	6
Persistent Storage	6
Notes:	6

Statement of Goals

The primary goals for this product will be to make at-home cooking easier and more accessible. To do this, we will focus on creating a “virtual fridge” for each of our clients, such that the database knows what and how many ingredients are in the user’s fridge at a given time. With this knowledge, the virtual fridge will also be able to recommend recipes based on the current ingredients in the fridge—this will be the app’s core functionality. The virtual fridge can also incorporate expiration date information, favored recipes, and difficulty level. Additionally, we can work on enhancing data input into the virtual fridge with text-recognition algorithms that can scan a receipt and input data. A manual data entry method will also be available. Lastly, if there is still time, we can work on a grocery store price comparison that will make it easier for our users to purchase ingredients at the best possible price. Stores located in a specified radius will be compared for the user, rating them on the price and quality of the needed ingredients for the desired recipe. Currently, our target audience is 20-40 year old individuals (millennials) who wish to cook at home more often. Our goal is to make this process easier for them by recommending recipes that will not require them to buy (many) additional ingredients.

Functional Description — Minimal Viable Product (MVP)

At its core, our application will be a mobile app that can recommend recipes for users based on items in their fridge. A detailed list of base-level features is below:

1. An interface that allows users to manually enter ingredients and food items in their fridge/pantry.
2. An interface to display the items the user has listed as being in their fridge/pantry.

3. An interface that displays recipe recommendations based on the number of matching ingredients as in the fridge/pantry.

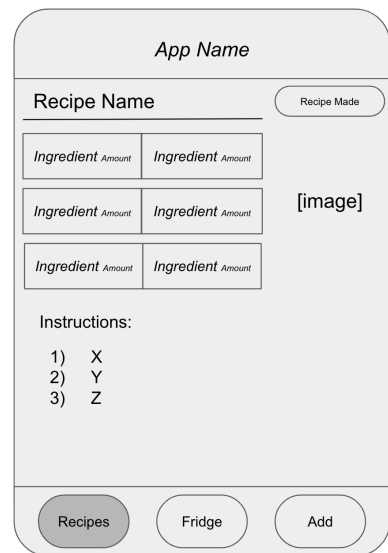
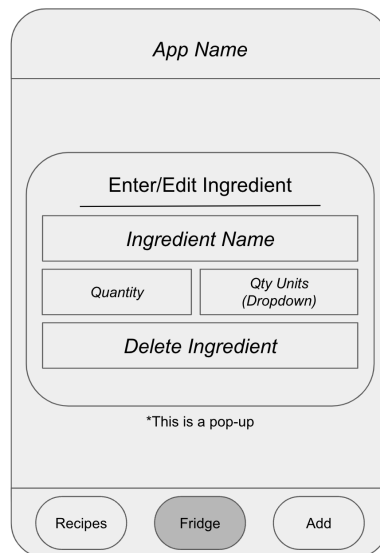
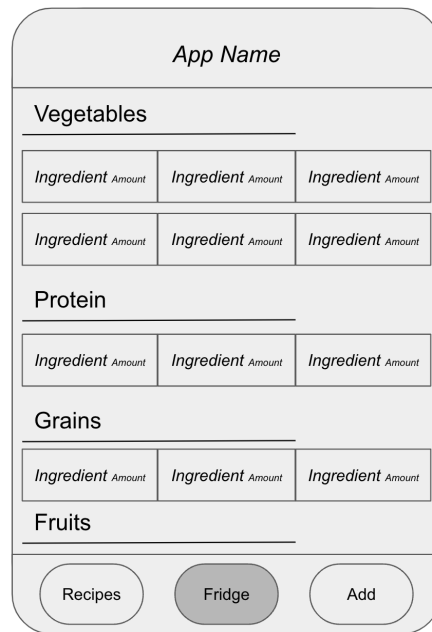
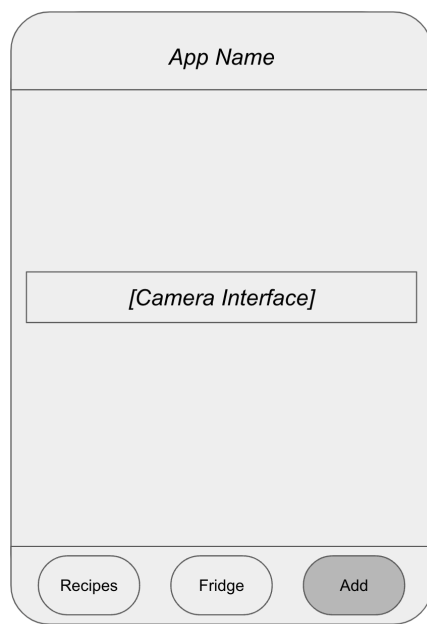
Any features added to the app other than the ones listed above will be considered additional and are not needed for this project to be considered a success. Some possible additional features, discussed in the statement of goals are: receipt scanning and location-based price comparisons.

Technical Data and Feasibility

Our application will require a list of recipes and their corresponding ingredients. We can source this data from online recipe websites or recipe databases (such as this one: <https://github.com/cweber/cookbook/blob/master/recipes.csv>). The application will also require the contents of the virtual fridge, which can be manually entered by the user. Our technology will depend on a reliable database (Firebase, Postgres, etc.) and being able to perform CRUD (create, read, update, delete) functions on it. It will also need an IDE for testing and development, which will likely be Xcode (the coding language will be Swift). Additionally, Google has a text recognition package for Swift that we can borrow (<https://developers.google.com/ml-kit/vision/text-recognition/ios>.) The above methods show a feasible way to implement the ideas of the app because the tools required do not have much of a learning curve and are attainable in the timeframe of this project.

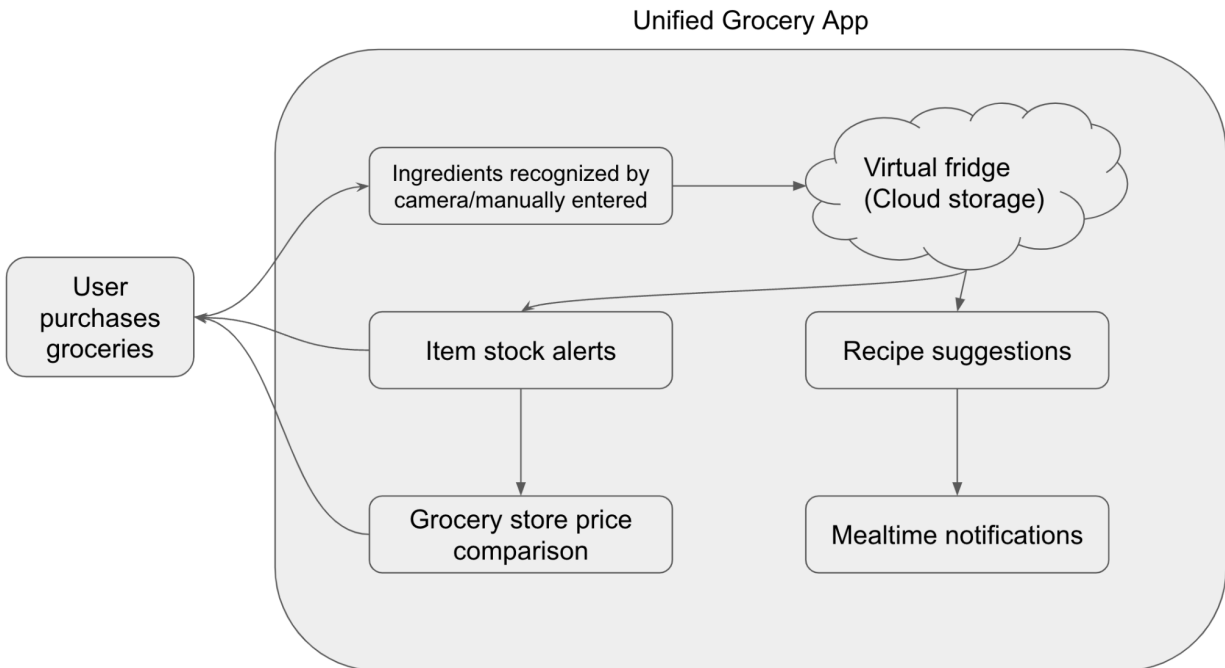
User Interface

Sketches of what the app will look like and the journey users will take as they click through the app are shown below. The current color palette to be implemented is: **F26979**, **B6B8D6**, and **F2EDEB**.



Structural Diagram

The diagram below shows how data will flow within the application.



Persistent Storage

The contents of the virtual fridge, as well as the master recipe/ingredient list need to be saved. There are two options for persistent storage, both of which use the cloud and are accessed via an account system. One option is to use an online pre-developed storage system such as Firebase. Another option is to host the data on our own server/computer, using a framework such as Postgres. Further cost/benefit analysis can be conducted to determine the best method of persistent storage.

Notes:

March 3 - V1 due, March 17 - V2 due