San Francisco State University

SW Engineering CSC648 / 848 Spring 2020

Team 103

Siddhita Chimote schimote@mail.sfsu.edu

Kevin Dizon-Cua

Calvin Tam

Sara Tama

Corey Russ

Matthew Ibrahim

Cassidy Mcskimming

Jose Pascual

Milestone 1

Revision 1 (05/11/2020)

Revisions

1. 03/04/2020
    - First submission
2. 05/11/2020
    - Updated Git approach

Executive Summary

“Stock up” is an application that provides a fast and easy tool to help keep track of the variety of storables stored inside your refrigerator. It offers several features that call to a wide range of customers, from tracking food, finding recipes with purchased goods, building shopping lists, to tracking food consumption. The goal for this project is to provide an easy to use interface so any type of user can interact with and generate these features.

The potential for this product is very promising as it stands alone. It provides numerous solutions to common household problems you see all over the world. Refrigerators remain in every single home you come across. Users will be able to be aware when they are running low in something before they even run out, eliminating unnecessary trips to the store for forgotten goods. Stock up will automatically filter out items that will be stored in the refrigerator by scanning the user’s receipt.This feature ensures that users have no difficulty using our product.

The demand for a product like Stock up will be timeless. There is no other application that offers such a combination of helpful tools to its users. Customers will build lifestyles around using the application to ensure organization, motivate change, and save time. Life after Stock up will be significantly different. The design and implementation was executed with expectations that the app will reach a diverse audience.

Our team has joined together at San Francisco State University to bring this project to life. We come from many different backgrounds all bringing something to the application that has made it both unique, and useful to all types of people. We all see great potential in this startup and are excited to see how it improves the lifestyles of all its customers.

Personas and User Stories

- Summarize several key personas (categories of users) for your application – their general characteristics, goals, skills, pain points related to the application you are developing.
- In user stories, you say how each persona will use your app (at high level)). Please number your user story, organize user stories into a category, and put your priority for each user story. Simple text format is OK.
- Focus on WHAT users do, their skill level, not on HOW is the SW implemented.

Scales

Skill level

1. Cannot cook
2. Beginner cooking level
3. Average cooking level
4. Intermediate cooking level
5. Chef

Persona 1

- *NAME:* Karen and John
- *CHARACTERISTICS:* tired, busy, interested in fast and easy options / opportunities to make things easier
- *ROLE:* Parents (caretaker of the household)

- *GOALS:*
    - cook nutritious meals for entire family
    - maintain organization of refrigerator
    - take care of the family on a daily basis (make sure everyone is fed / there is always food in the refrigerator)
- *SKILLS:* 4 (probably cooking often)
- *PAIN POINTS:*
    - most likely responsible for cooking for the family
    - require good time management, and good organization
    - does grocery shopping (often falls in the hands of the parents)
- *USER STORY:*
    - Parents are responsible for maintaining a healthy household, which involves cooking, feeding, stocking, cleaning, and keeping organized on a daily basis.
    - Parents could use an application like this to make their lives easier when it comes to grocery shopping and cooking every week.
    - We can assume parents will be using this app for creating shopping lists, deciding what to cook for dinner, taking care of expired items, monitoring diets for children, watching for allergies, etc.
    - We prioritize parents highly as we assume they will use many features of this application often and use it to its fullest potential.

Persona 2

- *NAME:* Mr. and Mrs. Davis
- *CHARACTERISTICS:* slow, tired, not so technologically advanced, maybe forgetful
- *ROLE:* Elderly couple
- *GOALS:*
    - having quick meals, low effort cooking
    - desire easy to use technology
- *SKILLS:* 4
- *PAIN POINTS:*
    - limited energy, meal plans
    - Interest in new recipes / new meals
- *USER STORY:*
    - Elderly couples who live on their own often are retired and have more time on their hands.We assume they would be very interested in browsing new recipes that they could use with the food they have in their homes.
    - They also can be forgetful and this application would help them keep track of their food, what they need, and what they don't before their trips to the store.
    - More often, older customers will not be as familiar with how to use newer technologies, so we assume they will not be testing new features or playing with complicated ones while using Stock up.
    - We prioritize the Elderly fairly but not with top priority.

Persona 3

- *NAME:* Danny, Pat, Kobe
- *CHARACTERISTICS:* each have unique lifestyles & characteristics, share the house and refrigerator with others
- *ROLE:* Housemates
- *GOALS:*
    - monitor their food (whether someone else is taking some or not)
    - maintain organization alongside housemates
- *SKILLS:* 3
- *PAIN POINTS:*
    - necessity to maintain organization
    - necessity to monitor expired items
    - shared refrigerator
- *USER STORY:*
    - Housemates share their household with others, including their refrigerator. They may be using the app to check repeatedly which of their items have been consumed recently, to check for stock of foods they want or like, or to check for what item has gone bad in the refrigerator that maybe doesn't belong to them.
    - These users range in skillset, characteristics, goals etc. In a broad sense we have built this app with intentions for it to reach a wide variety of customers.
    - We assume housemates would enjoy having an application that can help them keep track of the items that belong to them, and items that don’t.

Persona 4

- *NAME:* Stuart and Kelly
- *CHARACTERISTICS:* young, make accidents, still learning a lot
- *ROLE:* Children
- *GOALS:* eat, sleep, play
- *SKILLS:* 1
- *PAIN POINTS:*
    - needs food from parents
    - wants to know when low on a certain food product
- *USER STORY:*
    - Children live inside family homes where this application would be used. They likely will not require many of its features, however they may use it to check inventory of food they like.
    - Children describe people of ages 12 and under. Individuals of this age usually do not have much cooking skills, although a few may be skilled.
    - For the purpose of this project, we will assume the children are not very skilled at cooking. These children might want to cook simple snacks based off what's in the fridge.

Persona 5

- *NAME:* Sarah Jones
- *CHARACTERISTICS:* focused on ingredients, eats healthy, counts calories
- *ROLE:* Nutritionist
- *GOALS:*
    - stay healthy
    - try new recipes with healthy foods
    - create their own new recipes
- *SKILLS:* 4
- *PAIN POINTS:*
    - wants to eat healthy
    - wants to know when low on a certain food product
- *USER STORY:*
    - Nutritionists will be very consistent with their use in an application like ‘Stock Up’. We can assume a nutritionist might be very focused on their diet, what specific ingredients are inside specific food products, and what recipes they can use to cook meals with highly nutritious ingredients.
    - Nutritionists very much concern themselves with the food that they are eating which makes this application a great tool for this type of user.

Persona 6

- *NAME:* Jonathan
- *CHARACTERISTICS:* works a 9-5, busy life, wants to stay somewhat healthy
- *ROLE:* Working Professional
- *GOALS:*
    - keep an organized place to store things
    - wants fast and easy meals
- *SKILLS:* 4
- *PAIN POINTS:*
    - wants to eat healthy
    - wants to know when low on a certain food product
- *USER STORY:*
    - Working professionals often have busy lives, and so they do not really have time to cook meals that require a large amount of time.
    - They will probably want to just have a place to store stuff and to be able to create simple, quick and easy meals based on what's in the fridge since they are probably tired from work and do not want to create anything that requires a lot of effort.

Persona 7

- *NAME:* Brandon
- *CHARACTERISTICS:* an excellent chef, loves cooking and trying new recipes
- *ROLE:* Professional Cook
- *GOALS:*
    - gets items / ingredients for the new recipes
    - prepares top notch dishes
- *SKILLS:* 5
- *PAIN POINTS:*
    - wants to know when low on a certain food product
    - wants an organized kitchen
- *USER STORY:*
    - A professional cook wants to master his / her food storage
    - He / she will probably want to have every ingredient / item at their fingertips and be able to create new recipes based on what's in the fridge since they have keen eye in cooking.

Data Definitions

- Define main terms, data structures and “items” or “entities” at high or logical (not implementation) level (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document.
- Focus on key terms (main data elements used in your app, types of users and their privileges etc.) specific for this application and not on general, well known terms. These terms and their names must be used consistently from then on in all documents, user interface, in naming SE components and database elements etc.

Main terms

1. Storables
   DESCRIPTION: Items that can be stored in the refrigerator.
2. Non-Storables
   DESCRIPTION: Items that cannot be stored in the refrigerator.

Data structures

1. Recipes
   DESCRIPTION: A list of storables which are required to prepare a particular meal.
2. Refrigerator
   DESCRIPTION: Object that contains all the storable objects.
3. Shopping List
   DESCRIPTION: Object that contains all objects that the user needs to purchase.
4. Expired Storables List
   DESCRIPTION: Object that contains all objects that need to be removed from the refrigerator.

5. Low Stock List

   DESCRIPTION: Object that contains all the low stock objects.

Initial list of functional requirements

- Each requirement has a (reference) number, title, 1-3 line of description, owner / initiator (optional), priority, user story to be referenced.

Scales

Priority level

1. Urgent
2. High priority
3. Medium priority
4. Low priority

Function 1

- Users input information about adding an item to the refrigerator or removing an item from the freezer.
    - Input should be very convenient. As a main use case, it could be the receipt input out of the grocery using OCR API or barcode input (2D and QR codes). As the fall-back case when your choice of major input does not work, it should allow the manual input by website menu selection (e.g. “Add”)
    - It should provide a way to remove certain items (“Remove 2 apples”) when items were consumed or discarded.
    - As for clarity, the project considers the receipt (invoices) from “Safeway”.
- *PRIORITY:* 2
- *USER STORY:* Persona 1-7

- *USE CASE:*
    - A parent will want to be able to add an item to the refrigerator or remove an item from the freezer.
    - For example, maybe they bought ice cream, ate it day by day and finally finished what they had in the container.
    - Now, they must remove the container from the freezer. Our app will allow the user to indicate that they have removed something like ice cream from the freezer, making it easier to keep track of what they have in the fridge.

Function 2

- Users to keep an inventory of what they have.
    - It is highly recommended to access external product databases to look for product nutrition information for the given input item.
    - If this is not possible, you have to make databases on your own (enough to cover a variety of daily food items).
- *PRIORITY:* 2
- *USER STORY:* Persona 1-7
- *USE CASE:*
    - A working professional has a busy life. Sometimes they are not good at keeping track of certain things; they need some way to keep track of information such as what ingredients / foods they have at home.
    - Our app will allow the user to keep track of what kind of goods they have in their fridge so that it is possible to know what they have even with busy lives.

Function 3

- Users to be notified before items expire or low on stock level. May have to manually input the expiration date of the product when we add to our database.
- *PRIORITY:* 4
- *USER STORY:* Persona 4-6
- *USE CASE:*
    - A parent will often create meals for their kids from the fridge. Ideally, parents would like to feed their kids food that aren't expired and are of healthy quality.
    - Therefore, a useful feature to have would be to keep an expiration date recorded on the app, or when the fridge is running out of the kids' favorite foods.

Function 4

- Users to search for recipes based on their current items
    - Using a 3rd party database of recipes, use the current items to determine what recipes the user is able to make
- *PRIORITY:* 3
- *USER STORY:* Persona 5, 7
- *USE CASE:*
    - A professional cook may want to try a variety of recipes; he / she wants to branch out and try all kinds of recipes to expand his / her palette. To get inspiration, our app will allow him / her to search for recipes based on current itesm they have in the fridge / at hand.

Function 5

- Users to make a shopping list based on recipes they want to try
    - User inputs a recipe and the ingredients needed to make the dish, then a shopping list is generated upon entering all ingredients needed
- *PRIORITY:* 3
- *USER STORY:* Persona 1-7
- *USE CASE:*
    - A professional cook may want to buy ingredients to make a specific recipe. Our app will add all the required items to make the recipe to their shopping list.

Function 6

- Users to view the report regularly on their consumption of items
    - e.g. how much milk do they consume per month, etc
    - Highly recommended to report their nutrition consumption by nutrition category (calories, fat, protein, vitamin, …) It is advisable to access external product databases to look for product nutrition information.
    - Users really care about the accuracy of the report.
- *PRIORITY:* 4
- *USER STORY:* Persona 5
- *USE CASE:*
    - A professional cook may want to create a special dish on a certain night. To create this dish, he / she needs a variety of ingredients.

    - Our app will allow the user to create a shopping list based on the recipe he / she wants to try, an easy way for them to keep track of the list of ingredients.

Function 7

- Users to exclude ingredients in recipes based on allergens / intolerance
- *PRIORITY:* 4
- *USER STORY:* Persona 1-7
- *USE CASE:*
    - A person may be allergic / intolerant to certain ingredients (e.g. milk, peanuts) so suggested recipes should not include storables they are allergic / intolerant towards.

Handling technical challenges

All of our functional requirements are technologically feasible within our deadline. For example, we can read the input of our receipt by reading the barcode via a 3rd party API, or using Google Cloud's Vision AI (https://cloud.google.com/vision). We can also calculate the nutrient information of certain items simply by using another 3rd party API, such as spoonacular (https://spoonacular.com/food-api).

List of non-functional requirements

- (performance, expected load (the number of users), security requirements, storage, availability, fault tolerance…) Number each.
- Note that mandatory high level non-functional specs are given in high level document

Performance

1. Front-end renders React UI on the client side with minimum performance degradation.
2. Back-end responses to front-end requests on demands as it is running stateless containers.

Security

3. All data sent and received are using HTTPS protocol via API, and are protected by SSL / TLS.
4. Every device has to be registered before they can access the application. (Similar to WhatsApp Web.)

Infrastructure

5. Data shall be stored in the team's chosen database technology, which is MariaDB, on the team's deployment server, which is Google Cloud.
6. Application is deployed on Google Cloud Run on a Google Kubernetes Engine cluster, which can automatically scale up when demands increase with load balancer if enabled.
7. Application is possible to run on multi-cloud using Kubernetes to avoid cloud service downtime.

UX / UI

8. Application UI shall be easy to use and intuitive, especially adding and removing items.
9. Front-end should not allow to send duplicated requests to the back-end while back-end can distinguish a duplicated request and not to process it.

Accuracy

10. The inventory should be accurate.
11. The expiration date of storables should be accurately monitored.
12. The (nutrition) consumption should be accurate.

Coverage

13. Application should be able to process “Safeway” receipt when it contains non-edible items.

Project & code management

14. The codebase should be well maintained so newcomers can easily follow up with the code and add up their own functionalities.
15. There is a code style guide for formatting code in a consistent and organized way.
16. There are primary (master, develop), dedicated (sub)feature and (sub)patch branches in the Git repo.
17. The codebase has continuous integration to run unit tests.
18. All branches in the Git repo are merged by pull requests only after passing all unit tests.
    a. For subfeature and subpatch branches merging to their parent branches, we are using a squash and merge approach.

b. For feature and patch branches merging to their parent branches, we are using a squash and merge approach with mandatory code reviews by an assigned feature team member or leads.

c. For the develop branch merging to the master branch, we are using the standard merge approach with mandatory code reviews by the leads.

Competitive analysis

- Find 3-4 competitive features against what is available as of now (including Samsung Whisk, Chefling at Google Play).
- Present competitors' features vs. your planned ones.
    - First, create a table with key features of competitors vs. yours planned, only very high level, 5-6 entries max.
    - After the table, you must summarize in one paragraph what are the planned advantages or competitive relationship of your planned product to what is already available. In the table clearly mark your product, e.g. shade its column / data.

Key features comparison

Stock Up	**Samsung Whisk**	**Chefling at Google Play**
Turns recipes into shopping lists	Turn recipes into shopping lists	Recipes based on what you have
Monthly trackable nutrient intake	Share shopping lists	Smart, shareable shopping lists
Inventory management	Order groceries delivered to your door	Pantry & inventory management
Able to scan receipt and barcode to add items	Take Whisk Anywhere (Use and sync whisk across any devices)	Connect to smart appliances
Able to identify expiring storables		

Summary

Aside from the bigger competitors like Samsung Whisk and Chefling at Google Play that provides shareable shopping lists, grocery deliveries, and connection to smart appliances, Stock Up provides a more personal service in contrast to what is already available. StockUp provides a monthly trackable nutrient intake so you know what you are consuming to help you track any diets you have been, or want to be, following. Regardless if you're an expert, a beginner, or just want to get better at cooking, StockUp will cater to you.

High-level system requirements

- Briefly provide itemized list of all main SW components such as frameworks, tools and systems to be used, supported browsers and deployment platform (SW and server) to be used.
    - This list is to be the list of approved tools and systems from M0.
    - Any other external (open source) code / API / tool must be listed.

Server

- Cloud Provider
    - Google Cloud
- Cloud Deployment
    - Kubernetes (Google Kubernetes Engine)
        - Operation System
            - Container-Optimized OS (Release Channel - 1.157.7-gke.23)
- Local Environment
    - Docker Compose 1.25.4

Front-end

- Language
    - Javascript ES6 w/ JSX
    - CSS
- Web Server

    - Node.js 12.15.0
- Framework
    - Express 4.17.1
    - React 16.12.0
    - Atlaskit (https://atlaskit.atlassian.com/)
- Analytics & Tracking
    - Google Analytics

Back-end

- Language
    - Javascript ES6
- Web Server
    - Node.js 12.15.0
- Framework
    - Express 4.17.1
- Database
    - MariaDB 10.4.12

External APIs

- Google Cloud's Vision AI (https://cloud.google.com/vision)
- Edamam (https://developer.edamam.com/)
- Spoonacular (https://spoonacular.com/food-api)

Team

- Team Lead:
  Siddhita Chimote
- Front Lead:
  Calvin Tam
- Back-end Lead:
  Kevin Dizon-Cua
- Git Master:
  Matthew Ibrahim
- Scrum Master:
  Cassidy McSkimming
- Front-end/ Back-end team member:
  Corey Russ
- Front-end team member:
  Jose Pascual
- Front-end team member:
  Sara Tama

Checklist

- ❏ DONE - Team found a time slot to meet outside of the class
- ❏ DONE - Github master chosen
- ❏ DONE - Team decided and agreed together on using the listed SW tools and deployment server
- ❏ DONE - Team ready and able to use the chosen back and front end frameworks and those who need to learn and working on it, along with study schedule
- ❏ DONE - Team lead ensured that all team members read the final M1 and agree / understand it before submission