Team 106
SW Engineering CSC648/848 Fall 2019

Milestone 1

19th September 2019

1. Executive Summary

Short description of the final product/application and its key advantages, novelty, value (up to 1 page). Make it as an executive summary – think of answering the question of why we should fund this project. We suggest you assign a name to your project for easier reference and good "marketing" (code name). This summary should be readable to a general manager/executive that is not a CS specialist and is used to explain and also to advertise/promote your project. Typical outline is: one paragraph on the motivation and importance of the application you are developing, followed by a paragraph on what your application will be doing and how it helps the users (high level only, no jargon) and optionally what is unique and special in your design. At the end say in one paragraph about your team (e.g. about your student startup team…).

Marketing Code Name Suggestion: Cloud Fridge: *storing things in your fridge and on the cloud.*

A user friendly application to store and keep track of items in your fridge in real time. The purpose of *Cloud Fridge* is to save our customers the time of rushing home to check what's in their fridge. With *Cloud Fridge* its an easy, quick way to check what food you have at home. An application to save you the trouble of repeatedly getting up to check what items you have in your fridge and keeps track of the nutritional value of your food, all at your fingertips.. Our app's suggestive features also tells you what you can make with the items in the fridge so there will never be a need for your food to go to waste.

This app is an inventory management application to make keeping track of your fridge much more simpler. Inventory management in general takes time and can be complicated depending on how efficient you want it to be. With Cloud fridge, each item in the fridge has all the necessary information such as expiration date, costs, recipe suggestions and more to take away unnecessary wasted time that the user might spend to look them up individually.

We are a group of students who are tired of getting up to check our fridge only to remember that we forgot to restock. No more will our time be wasted any longer and met with hopelessness. Our group range from juniors and seniors to graduate students with a variety of coding background and experience. To prevent such tragedy from happening to other people like us, we are making a change one fridge at a time.

2. 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 story you say how personas 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.

NOTE: avoid specific on HOW functions will be done and text resembling user manual: this is supposed to guide the design of the future product and is NOT a description of how the product will work (you don't know that yet) – see class slides for details.

1. Mary (32, F) is a mother of a 5-year-old child. She lives in an apartment with her son and spends most of her time either doing household chores or taking her son. Her sons name is David, named after her own father who she loved dearly. Her father did everything he could to make sure Mary had a good life by spending as much time as he could with her, and now Mary wishes the same for her son. Between being a mother and a full-time employee, Mary does not have much time or energy to spare.

   Every weekday, Mary drops her kid off at elementary school, goes to work, picks her kid back up, goes home to make them food, and then begins to do chores. After that, it is bedtime and the beginning of another day of much of the same thing. As much as Mary would love to spend some time on the weekends for herself, perhaps watching her favorite shows on Netflix (particularly Brooklyn 99), she must go grocery shopping and make sure they have enough food to survive the week. Mary likes to stay on top of things and is great at multitasking, but between juggling a job and a child, sometimes her memory fails her.

   With how little time Mary has, Mary needs to know if she has enough ingredients in the fridge to make dinner without standing in front of it. She does not want to waste money on buying ingredients she already has. Currently, she solves the issue by trying to remember or write a list before she goes out to shop. But at the end of the day, she does not want to have to try to remember or update a list of items she has in the fridge. Mary just wants a fast and simple way to make a list and access it through her mobile device.

2. Max (23, M) is fresh out of college working his first job at a tech company in San Francisco. He lives in a modest apartment in South San Francisco, as it is close to his workplace, but much cheaper than an apartment in San Francisco. He likes to learn and try new things in his free time. Max loves new technology, so instead of furnishing his place, it is littered with the next generation tech. In particular, he loves the convenience it brings and tries to use new technology in every chance he gets. When he has to pay for something, he always tries to use the mobile pay function on his Samsung phone before resorting to his credit cardW.

Max commutes to work by riding his electric skateboard to the Bart, and takes the train all the way to Montgomery, where his workplace is a few blocks away from. Max loves socializing with his coworkers, so when lunchtime comes around, he buys food with his coworkers from a nearby food truck and goes back to chat with them. After work, his coworkers tend to go out for dinner. Max wants to go with them but knows that if he buys food twice a day, he would go over his budget. He needs to save some money for emergencies, but also for when the latest technology drops.

As time goes on, Max notices one of his coworkers, John, has started to bring in his own lunch. Max asks him about his food, and John replies that he has discovered technology that has changed his life: a timed crockpot. John explains that he has been having trouble saving money and decided to make his own food. That way, he has food for lunch at a cheaper price, and can still socialize with his coworkers.

Max think this is a great idea. Currently Max solves the problem by compromising and not going out for dinner with his coworkers. But by following in John's footsteps, he can not only hang out with them during work, but after work as well. Without hesitation, Max goes home that night and orders the latest crockpot that allows him to cook with minimal effort, while letting him try out some new technology. Now he just needs a way to keep track of ingredients he buys in order to make the most of his money. Max knows that he can make a list on his phone by manually adding things he buys one by one, but he thinks it is a hassle. With all the new technology floating around, Max thinks there must be a better way to keep track of things he buys to store in the fridge.

3. Data Definitions

define the 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. In later milestones you will add more implementation details for each item. You will later expand this section with more details.

User Account - Takes the user’s name and email to make an account so that they can store a list of their food items

Food List - List of food items in the refrigerator

Food Item - Name of available item to be added to the refrigerator.

List Management/Inventory Management - Easily add or remove from food list

Tags/Filters - Categorize items based on food type or nutritional value (ex: highest sugar content, lowest sugar content)

4. Initial list of Functional Requirements

see class notes. This refers to high level functions you plan to develop to the best of your knowledge at this point. Focus on the WHAT and not HOW. Keep the user in mind. Develop these functions to be consistent with user stories and requirements above. Number each requirement with unique numeric value and use these numbers consistently from then on.

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

When you specify the functional requirement of adding a new grocery item to the refrigerator, the requirement should be validated of its technical feasibility with your prototype. Please show to the instructor during the team meeting session that your prototype could provide the specific functional requirement.

Functional Requirements listed in High-Level Project Description:

- Users input information on adding or removing an item; input could be text input, speech input, barcode scanning, etc.
- Users keep an inventory of what they have; recommended to access product information through external database
- Notifications of item expiration
- Ability to search for recipes based on current items
- Ability to create a shopping list for recipes that the user wants to try
- Regular report on food consumption

Potential Functional Requirements

1. **Customizable Inventory - High Priority**

   Allow user to add tags/filters, order the list of inventory, set text colors to more easily identify and categorize items

2. **Recipe Suggestions - High Priority**

   Offers easy recipes based on ingredients you already have in your fridge.

3. **Shopping List - Medium Priority**

   Ability to create a shopping list for regular grocery shopping and add that to inventory, removes tediousness of adding items one by one

4. **Consumption Report Suggestions - Low Priority**

   Food consumption report also gives suggestions on what to eat more or less of

5. **Customize Notifications - Low Priority**

   Allow user to customize food expiration or food report notifications, such as the minimum time limit before they are notified of food expiring or what they want to see on their food report (nutritional values, calories consumed, suggestions on food choices)

6. **Item Presets - Low Priority**

   Have item presets for commonly bought food items to reduce the need for manual input from user

5. 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, **so for Milestone 1 we recommend you simply copy them from high level document from iLearn. On top of these, if you prefer other non functional requirements, please add them.** Please observe and adhere to these non-functional requirements in your design and development from now on.

Non-Functional Requirements from High Level Project Description

- Application shall be developed, tested and deployed using tools and servers reviewed by Class TA (Nicholas Olegovich Stepanov) in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be reviewed by class TA).
- Application shall be optimized for mobile browsers.
- Data shall be stored in the team's chosen database technology on the team's deployment server.
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
- Application shall be very easy to use and intuitive.
- Pay functionality, if any (e.g. paying for goods and services) shall not be implemented.
- Site security: basic best practices shall be applied
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
- The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

Additional Non-Functional Requirements

-

6. Competitive Analysis

Find 3-4 competitive features against what is available as of now (including smarter FridgeCam). 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

	Samsung Family Hub 2.0	Smarter FridgeCam	Cloud Fridge
Mobile Accessibility	✓	✓	✓
Picture of Inside Fridge*	✓	✓	X
Expiration Date Tracker	✓	✓	✓
Create Item List with Receipt	X	X	✓
Cost	$2398	$164	X

*quality of images may vary

The planned advantage of the Cloud Fridge is its simplicity. Currently competitors try to have a camera in the refrigerator so users can see the contents from anywhere, but their main function is still creating and adding to a list one by one (due to the unreliability of the camera). Rather than manually adding items to a list, the Cloud Fridge's main function will be the ability to upload a picture of your grocery receipt and create a list of items from that picture. Our application plans to please users that prefer a faster and easier way of keeping track of what is in their refrigerator, and want to do away with the traditional list creation process. As a user, you want it to just work, and that is what our app aims to do; take a picture of receipt and done.

7. High-Level System Requirements

Briefly provide an 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 code/API/tool must be listed.

Digital Ocean

Nginx

NodeJS / Express

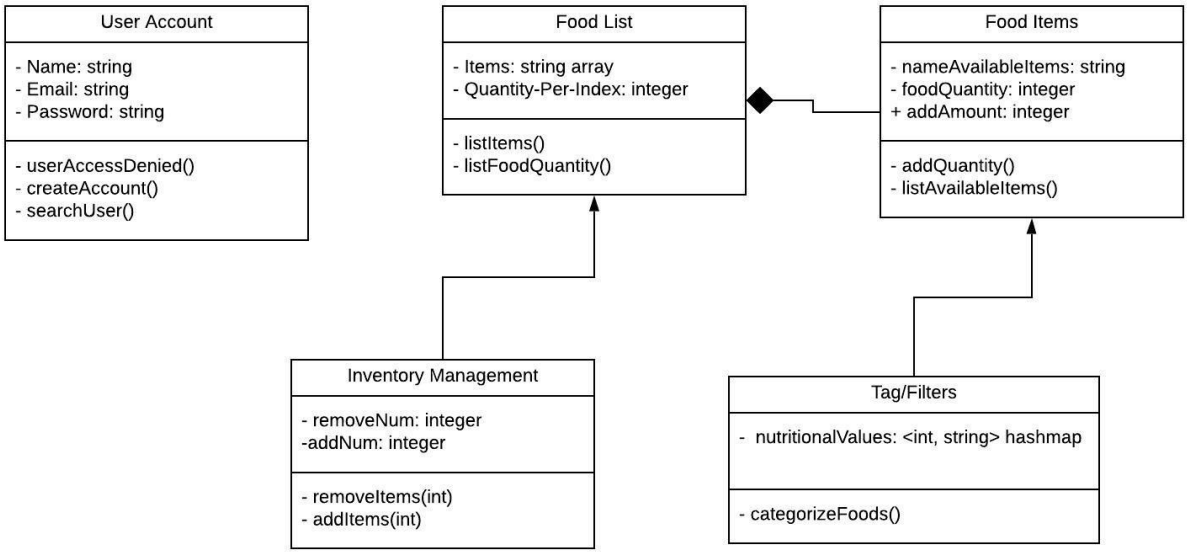
Handlebars (Templating Engine)

MySQL

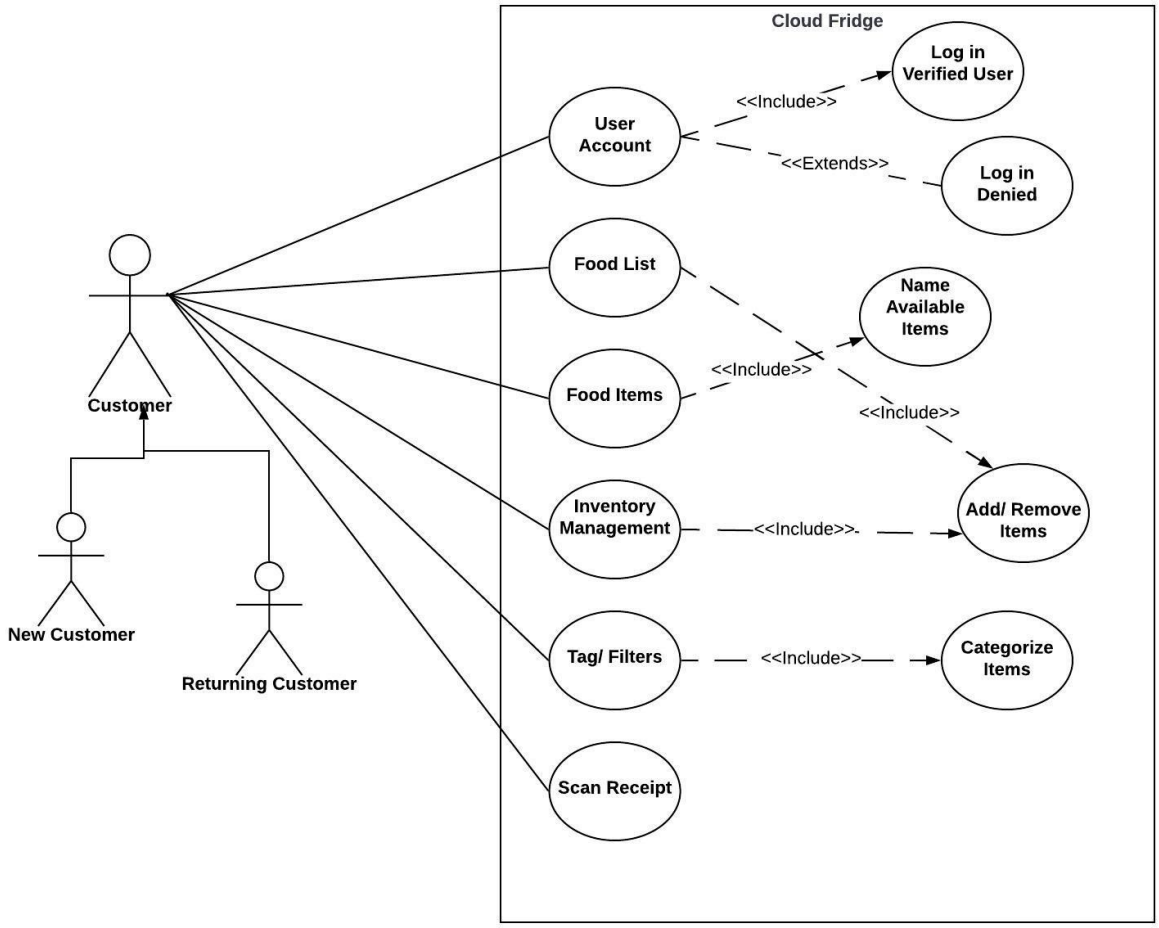
UML Diagrams

UML Class Diagram:

Team 106, SW Engineering CSC648 Fall 2019: UML Cloud Fridge



UML Use Case Diagram:



8. Team

list student names, name of the team leader, names of front and back team lead and roles for each member.

George Butler - Team Leader & Git Master

Sydni Starling - Scrum Master

Aung Hein - Front-End Lead

James Day - Back-End Lead

Raymond Gee - Front-End Development

Jian Xin Qi - Developer

Tommy Nguyen - Back-End Development

8. Checklist

For each item below you must answer with only one of the following: DONE; or ON TRACK (meaning it will be done on time, and no issues perceived); or ISSUE (you have some problems, and then define what is the problem with 1-3 lines)

- **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**

  ON TRACK

- **Team lead ensured that all team members read the final M1 and agree/understand it before submission**

  DONE