**CS3354 Software Engineering**

**Final Project Deliverable 2**

**Pure Eats**

**By**

**Taha Arshad, Mahjoobah Billah, Bushra Hameed, Yongda Li, Anjali Prabhala, and Servando Luviano**

**GitHub**

https://github.com/ServandoL/CS3354-PureEats.git

**Deliverable 1:**

1. Delegation of tasks:
   1. Anjali Prabhala oversees addressing the feedback provided for our proposal and deciding which software process model is being employed in our project.
   2. Mahjoobah Billah oversees the lists of software engineering requirements.
   3. Taha Arshad will provide the use case diagram for our project.
   4. Yongda Li will provide the sequence diagram for each use case of our project.
   5. Bushra Hameed will provide the class diagram of our project.
   6. Servando Luviano will provide the architectural design of the project.
2. Project Deliverable 1 Content:
3. Address the feedback provided for your proposal by listing what you are doing to comply with the proposed changes/requests for additions to your project.  
     
   As mentioned in the feedback given to our group, we will discuss the differences between our application and any similar software on the market within our final report. A specific software application that we will be comparing our project to is the Scan Halal app. This application scans the barcode of a product and determines if there are any dietary restrictions applicable to that item. Our project enhances this idea as we widen our demographic for those with food allergies and other dietary restrictions including, but not limited to, kosher and veganism.
4. Which software process model is employed in the project and why?

The incremental software process model is employed within our project. This is because each phase can be referred to in order to update any changes made to the application. In addition, the cost is low and the process model is flexible to change. The incremental process model also promotes user involvement as well as maintainability with low risk involvement. Our application is created for users to be able to find alternative food items in the case that a specific product contains an ingredient that the user may be allergic to or may not be able to consume due to any dietary restrictions. The app is an interactive tool that requires flexibility related to change. For these reasons, we have decided to incorporate the incremental software process model.

1. Software Requirements

**Functional Requirements:**

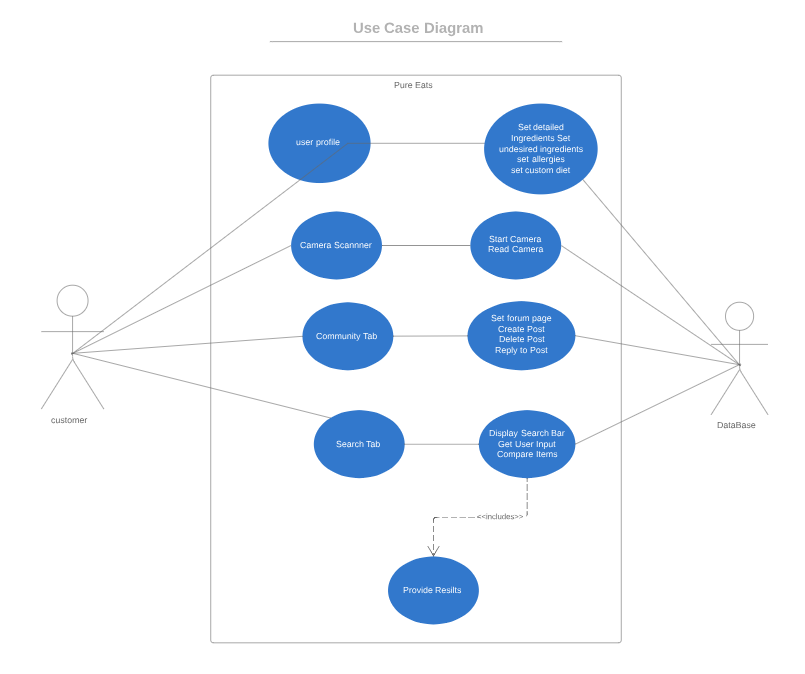
* User will be able to scan the barcode of an item using a phone camera
* User will be able to search products by name and category
* User will be able to search for an ingredient to see if it has different names
* User will be able to create a user profile where user can:
  + add an ingredient that user wants to avoid
  + add an ingredient that user wants more of
  + set up a preset diet with guidelines

**Non-functional Requirements:**

* **Product Requirements:**
  + **Usability requirements:** User must have: a working smartphone with a working camera and internet connection.
  + **Efficiency requirements:**
    - **Performance requirements (how fast the system must execute):** There should be no lagging
    - **Space requirements (how much memory it requires) :** About 50MB
  + **Dependability requirements:** The downtime shall not exceed 30 seconds
  + **Security requirements:** User will have a password to protect his or her personal information.
* **Organization requirements:**
  + **Environmental requirements (specify the operating environment of the system):** The system will be operated on a mobile app.
  + **Operational requirements (define how the system will be used):** The system will have the user login and set up their personal preferences for their diet. Then, when they need to use the app, they will go to the Scan Product area where the app will use their camera to scan the barcode of a product.
  + **Development requirements (specify the programming language):** The programming language will be Java
* **External requirements:**
  + **Regulatory requirements (set out what must be done for the system to be approved for use by a regulator):** Certain disclaimers will be disclosed such as: All information contained on the system, including nutrient information, ingredients, food allergens, and information relating to medical, religious, and health conditions is for informational purposes only and not provided as official medical, religious, and dietary advice. You should not use information on the services for the diagnosis or treatment of a health problem or diseases. You should always consult your healthcare provider before making any dietary or nutritional changes.
  + **Ethical requirements (ensure that the system will be acceptable to its users and the general public):**  The system will always stay up to date so that no information will be outdated.
  + **Legislative requirements  (ensure that the system operates within the law):** There will be a terms of service
    - **Accounting requirements:** All financial statements will adhere to GAAP (Generally Accepted Accounting Principles)
    - **Safety/security requirements:** Users’ personal information will be protected. There will be a privacy policy.

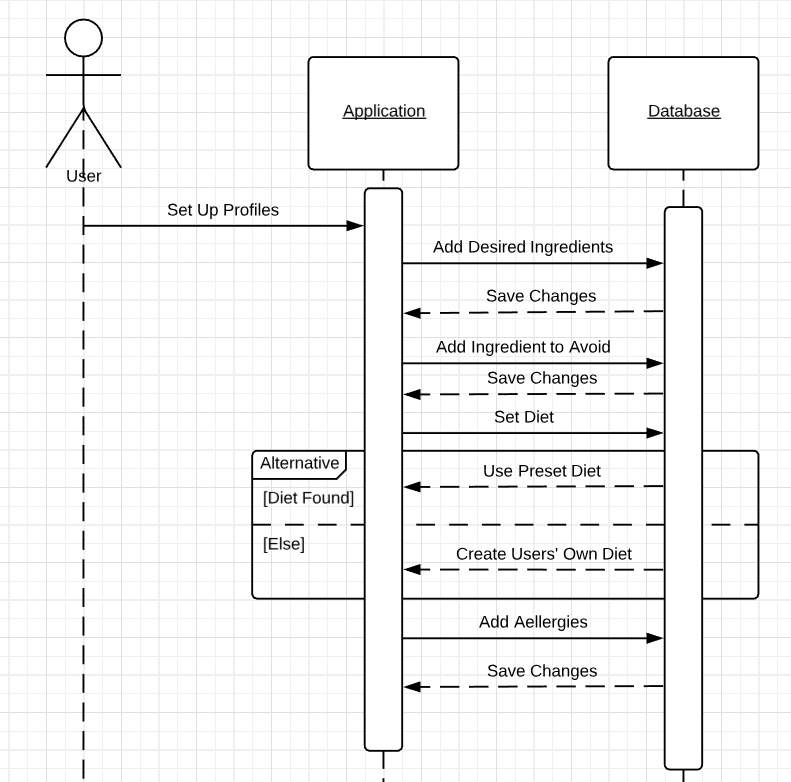
**Domain Requirements:**  
Entering the food allergies or dietary restrictions  
Entering the category of the food

1. Use Case Diagram

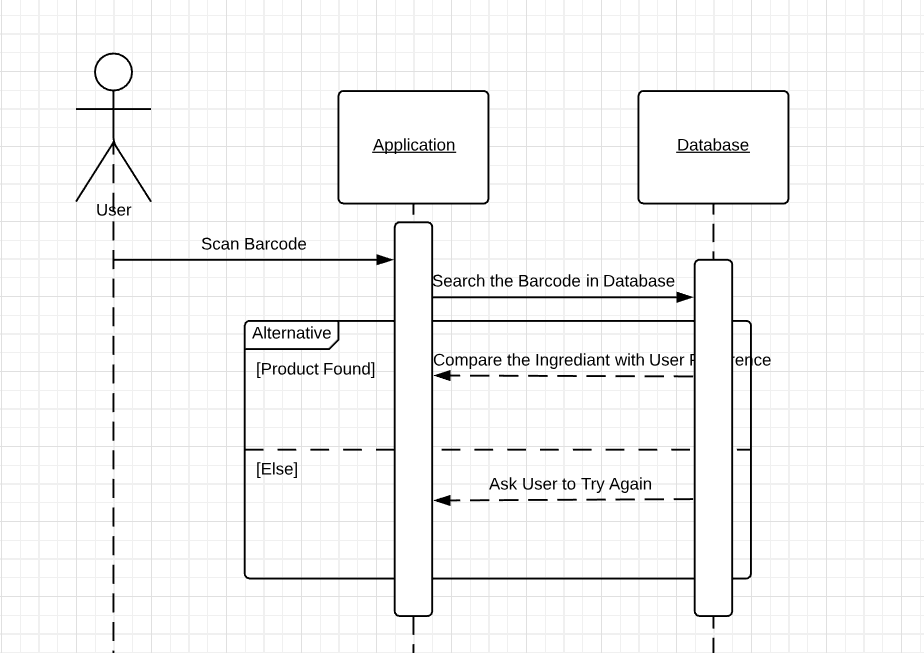


1. Sequence Diagram

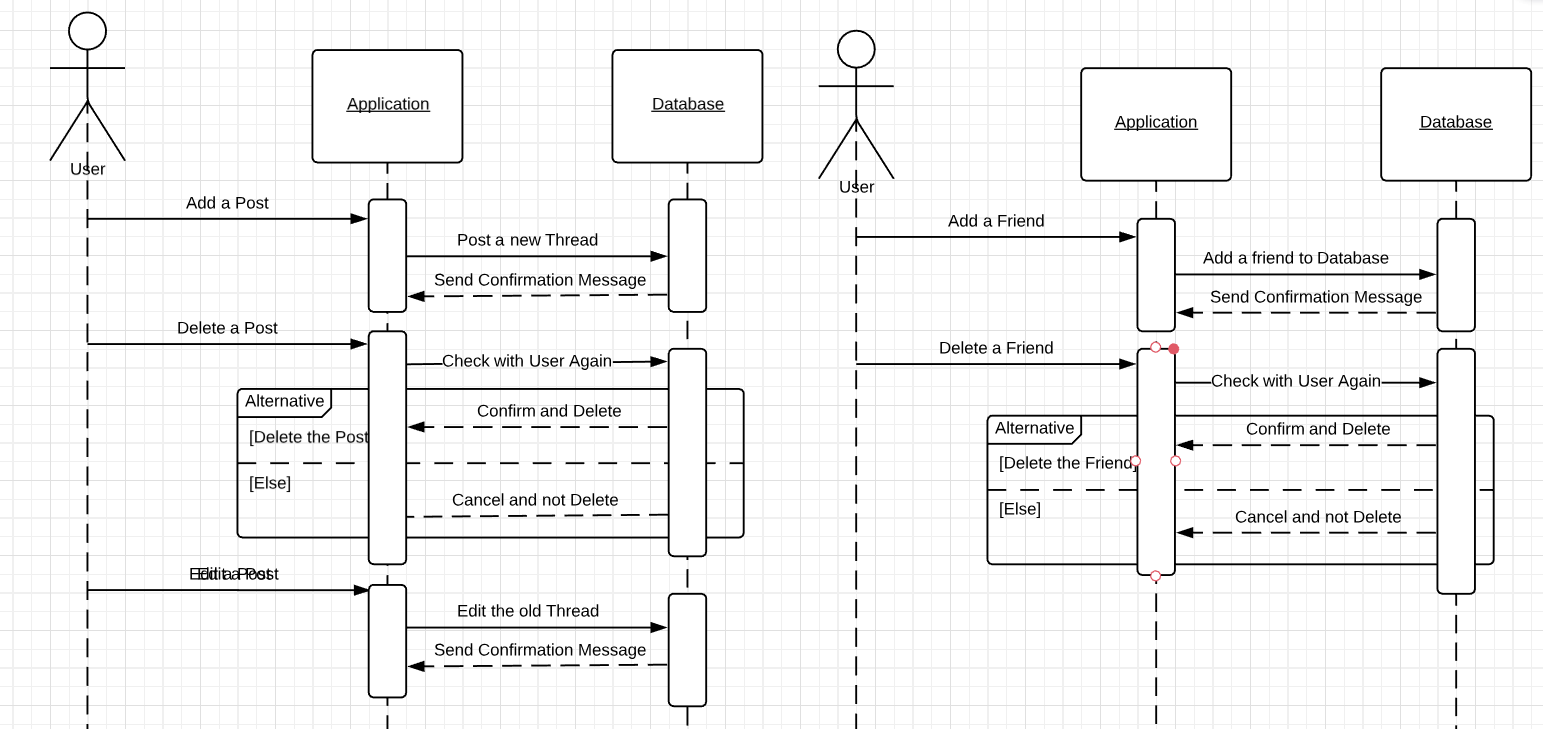
**PROFILE SETUP**



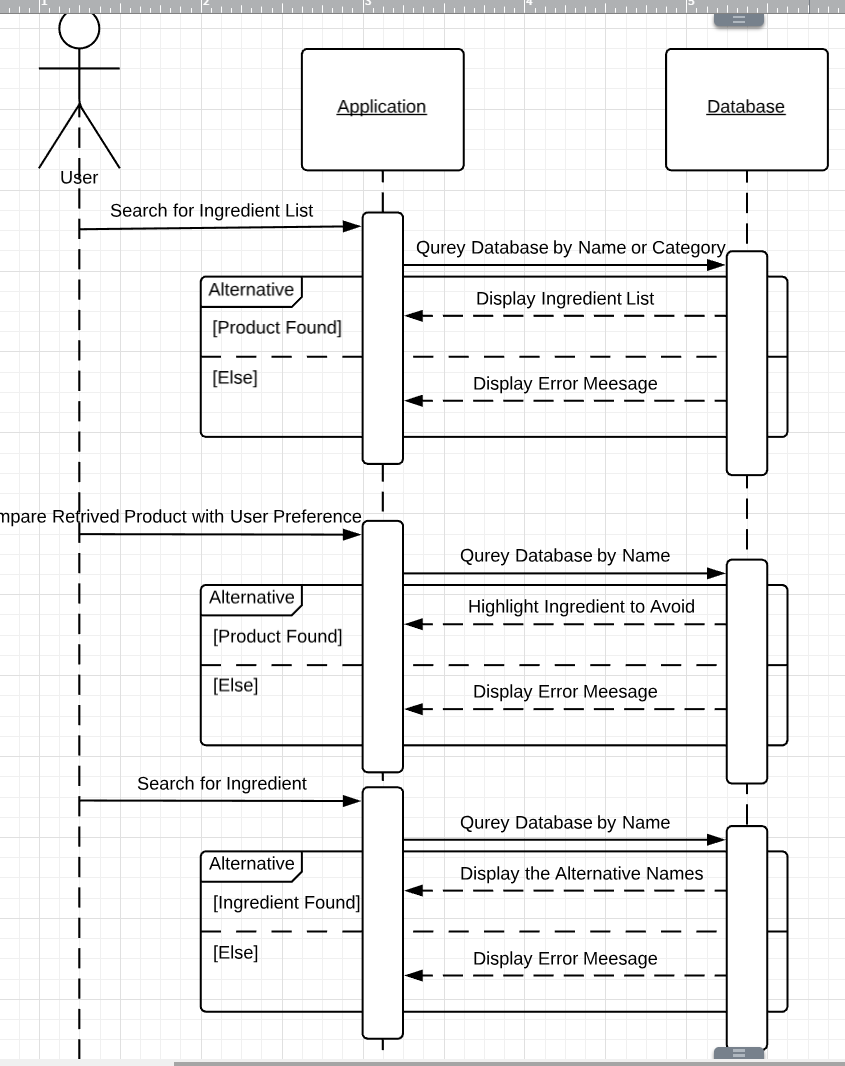
**CAMERA SCANNER**



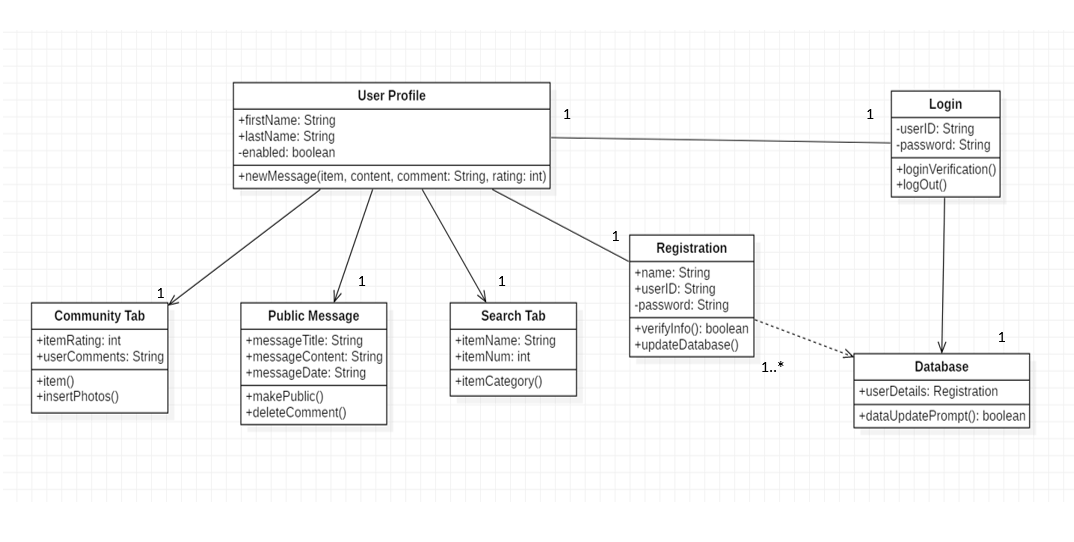
**COMMUNITY TAB**

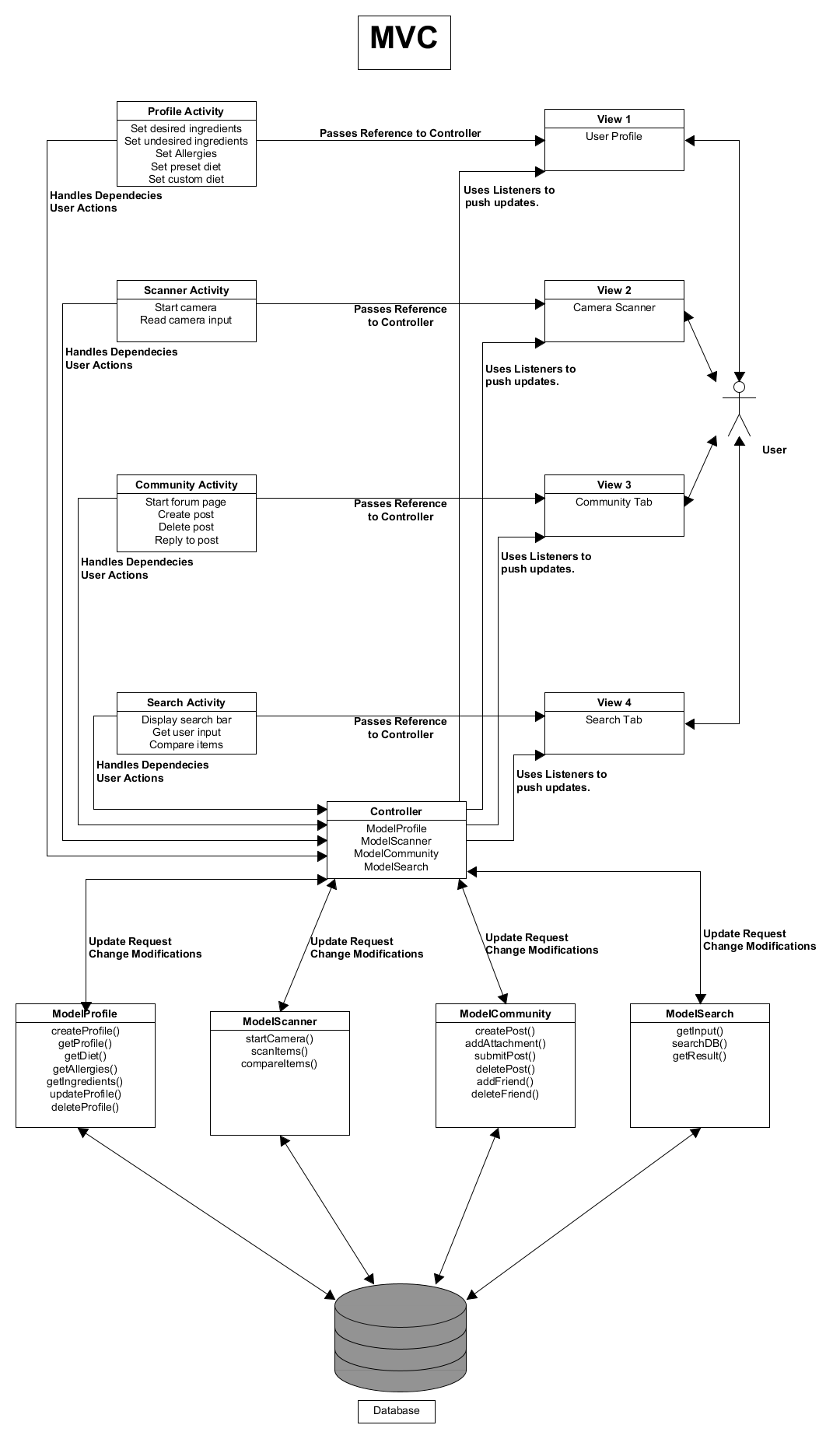


**SEARCH**



1. Class Diagram

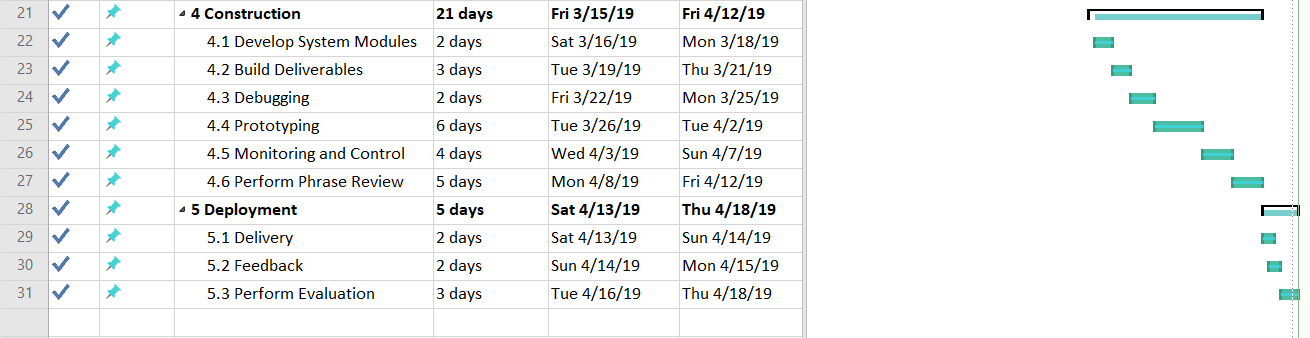


1. Architectural Design

# **Deliverable 2:**

# Project Scheduling:

# 



# Cost, and Pricing Estimation

Table – Software Effort using Function Point

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Function Category | Count | Complexity | | | | | Count \* Complexity |
| Simple | Average | | Complex | |
| 1 | Number of user input | 13 | **3** | | 4 | | 6 | 39 |
| 2 | Number of user output | 19 | **4** | | 5 | | 7 | 76 |
| 3 | Number of user queries | 4 | **3** | | 4 | | 6 | 12 |
| 4 | Number of data files and relational tables | 30 | **7** | | 10 | | 15 | 210 |
| 5 | Number of external interfaces | 4 | **5** | | 7 | | 10 | 20 |
|  |  |  |  | |  | | GFP | 357 |

**PROCESSING COMPLEXITY ADJUSTMENT**

PCA = 0.65 + 0.01(PC1 + PC2 + PC3 + PC4 + … + PC12 + PC13 + PC14)

PCA = 0.65 + 0.01(3 + 4 + 3 + 4 + 3 + 5 + 3 + 3 + 2 + 2 + 3 + 2 + 0 + 5)

PCA = 1.07

**FUNCTION POINTS**

FP = GFP \* PCA = 357 \* 1.07 = 381.99

**ESTIMATED EFFORT**

Assumption: 30 function points per person-week

E = FP/PRODUCTIVITY = 381.99 / 30 = 12.7 -> About 13 person-weeks

Estimated uncertainty: [4 – 52] person-weeks

**PROJECT DURATION**

Team size = 5 🡪 4 views total, 1 developer per view; 1 developer for database implementations

D = E / TEAM SIZE = 13 / 5 = 2.6 -> About 3 weeks

Estimated uncertainty: [1 – 12] weeks.

**SOURCE LINES OF CODE (SLOC)**

Language Factor (LF) for Java = 53

SLOC = FP \* LF = 381.99 \* 53 = 20,245 SLOC

**TOTAL ESTIMATE OF PROJECT**

Assumptions:

Average productivity = 225 LOC/pw

Labor rate = $2,000 per week.

Cost per line of code: approximately $9

LOC estimate: $9 \* 20,245 = $182,205

Total estimated cost: $182,205

* Our application will be developed using Android Studio which is a free software for Android app development and requires a small fee for publishing.
* The UPC database implemented into the UPC scanner is going to be from [www.upcdatabase.com](http://www.upcdatabase.com) and [www.upcdatabase.org](http://www.upcdatabase.org); both of these are free to use.
* The ingredients database that will be implemented into our project will be hosted by USDA Food Composition Database, which is free to use.

# Estimated cost of hardware products

Could Server: $100 per year

# Estimated cost of software products

UI Design Software: $50 per month

# Estimated cost of personnel

2 Front-end developer: $80000 per year \* 2 = 160000

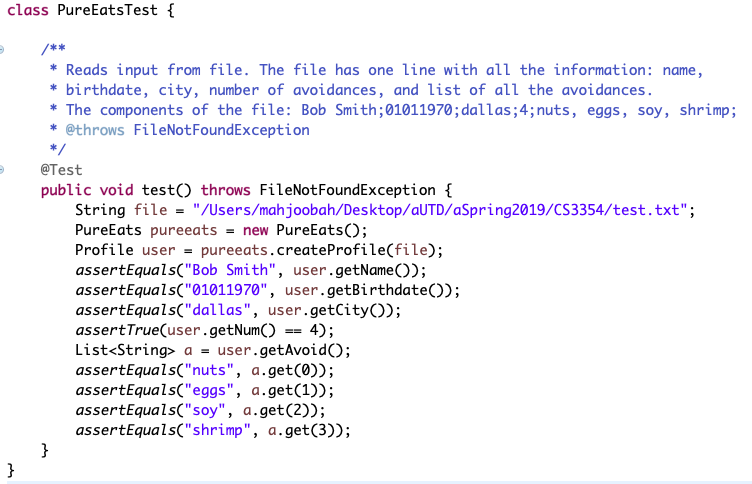
2 Back-end developer: $120000 per year \* 2 = 2400000

# Test plan for software

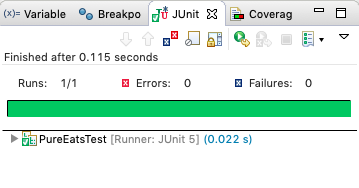
We created a method where a user profile is created. The user profile contains a username, birthday, city/location, and a list of foods or ingredients the user would like to avoid. For testing, we took input from a file which contained all the user information.

1. Test Plan



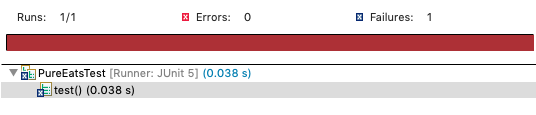


This is the correct test case



Test Case where it fails:





https://lh3.googleusercontent.com/EwAI8EgX7mfcvaQQuwm1WPy-TG6H4bpkPrxsebcTjYZ1eVrVgMo-lLsYQcF9Q7MfUVRH7iB8rxWB5sqQ_Q6O2ybst5W3rpuaYuHEf8zE6JKq7YyXeBdoqA2JNpdQSLUBa4HaxcdH

1. Comparison

The project domain of our app includes broad categories of food, allergies, ingredients, and health. Our app is targeted for the use of individuals who need to avoid certain ingredients i.e. food allergies and dietary restrictions: Ketogenic, Atkins, Vegan, Vegetarian, etc. The app is accessible for everyday life. When deciding the specific functions for our app, our team had the following thought process:

* Function: Scan ingredients with a barcode using the phone camera or image in the camera roll

Benefits: Ease of use and user-friendly. Assuming that most users have a smartphone, using the camera would not be a problem.

* Function: Search product by name and category

Benefits: Attain accurate results, which can also be customizable by the user as they can set more restrictions on diet and products. Also, it provides an allergy-oriented section that has allergens or any ingredients to avoid.

* Feature: Community tab

Benefit(s): Social network aspect to the app that creates a healthy living community. In addition, people are likely to stay online and spend more time on the app engaging in conversations or reading posts.

* Feature: Cross-platform app

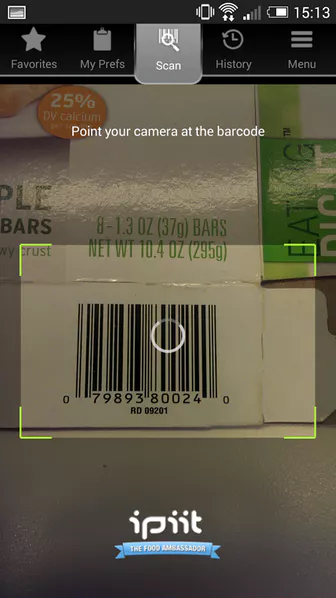
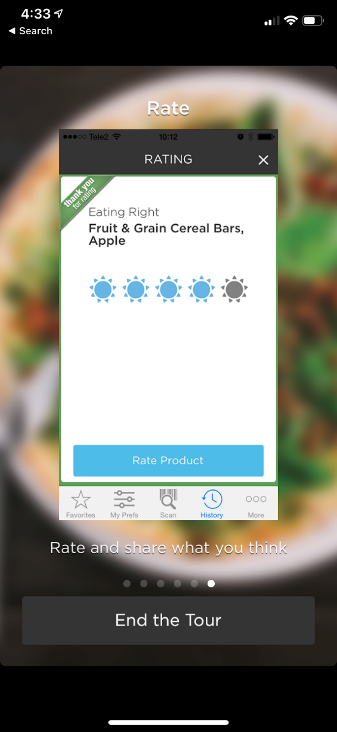
Benefit(s): Compatible with iOS (iPhone, iPad, Macbook, etc.) and Android (Phone, Tablet, Note, etc.)

Currently, there are many similar food allergy apps in the market. According to Forbes, the four most popular food allergy apps are [2]:

1. AllergyEats
2. Spokin
3. Ipiit
4. Allergy Food

AllergyEats is the most popular of these four options. It is an allergy app used to avoid food allergies in restaurant and is available in both the Apple Store and Google Store. Similar to AllergyEats, Spokin allows users to check for food allergies in nearby restaurants, hotels, summer camps, food products and more. In addition, it “curates a news feed for you based on your allergy preferences, location, and personal interests” [2]. This app also incorporates a social media aspect as users can interact with one another, follow each other’s accounts, and build relationships [2]. However, the one drawback of this app is everything is based on the location. This app stood out to be the best because it had multiple features and an interactive user interface. The app which was very different from ours was Allergy Food. Allergy Food allows users to create an account for every member in the family and helps to translate food allergies in French, German or Spanish when and if you’re traveling overseas [2].

The app with the closest functionality to our design is ipiit. It is available in the App Store, Google Play and through a website. The main functionality that is similar to this app is that it scans food products to check for allergens (as shown in Figure 1) [1]. It also has a section to blacklist specific ingredients and a tab to set food preferences. It has a few other interesting functions as it provides alternate ingredients and compares food products. It does not have a community tab but has a rating section to rate specific products. A unique feature included on the app is the “Try now” button, which enables users to understand how the ipiit works

Other apps, that were not mentioned in the article, but are similar to our project [3]:

1. CountChecked: helps users to avoid ingredients by scanning products. Also, available for iPhone and Android devices. The only con is that it is not free (costs $2.99).
2. FoodFacts: consists of all of the same features of our project with an additional feature of being able to add the entire family’s list of ingredients to avoid. Unfortunately, this app also costs $1.19.

Both of these apps did not have a community tab.

1. Conclusion

As mentioned in the Professor’s notes, our team did not need to make any changes to our project. Thus, we continued with our original idea as it had been approved earlier on within the semester. Seeing that there are similar applications available, we used those apps as an example and template for our project idea. Our app is different as it incorporates a social media aspect to it. Users can communicate and interact with one another in order to identify that there are other options available on the market. In the last we would like to say that doing this project was a great experience and we enjoyed doing it.

1. References:

[1] B. Toth, “ipiit - Scan food products to check for allergens,” *Product Hunt.* Available: <https://www.producthunt.com/posts/ipiit>.

[2] J. Jet, “Traveling With a Food Allergy? Check Out These Four Apps,” *Forbes*, 18-Jan-2018. Available:<https://www.forbes.com/sites/johnnyjet/2018/01/18/traveling-with-a-food-allergy-you-need-these-four-apps/#4f7e2de4c4c0>.

[3] J. Bradley, “6 Food Allergy Apps That Can Help You Shop, Eat Out ... or Find an ER,” Verywell Health. Available: <https://www.verywellhealth.com/food-allergy-apps-to-help-keep-you-safe-1324320>.