**Phood Buddy**

**SRS**

**<COP4331, Spring, 2016>**

Team Name: The Phoodies

Team Members:

* Timothy Flowers
* William Funk
* Evan Glazer
* Jorge Rodriguez
* Lyudmila Sandomirskaya

Modification history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| v0.0 | 02/26/2016 | W. Funk | Initial Requirements – Completed by section project manager, though all team members contributed to the document. |
| v1.0 | - | - | - |
| ... |  |  |  |

Contents of this Document

Introduction

* Software to be Produced
* Reference Documents
* Applicable Standards
* Definition, Acronyms, and Abbreviations

Product Overview

* Assumptions
* Stakeholders
* Event Table
* Use Case Diagram
* Use Case Descriptions

Specific Requirements

* Functional Requirements
* Interface Requirements
* Physical Environment Requirements
* Users and Human Factors Requirements
* Documentation Requirements
* Data Requirements
* Resource Requirements
* Security Requirements
* Quality Assurance Requirements

Supporting Material

**Introduction: Software to be Produced**

This feat of software engineering will provide the user with a solution to indecisive eating. Through random selection from a wide variety of recipes, while filtered down by health concerns and a taste profile unique to that person, the user will have a meal to cook at the click of the proverbial button. It will supply the user with a full lists of ingredients to make that recipe, and all of the directions it has available. To finish off the convenience that this application provides, it will offer a list of the best deals in the user’s area for the ingredients needed. Users can rate the recipes they liked or hated, and from this list of previously rated recipes they can search for something more specific.

**Introduction: Reference Documents**

For greater detail as to a more in-depth description, see the Concept of Operations document at: <http://www.williamrobertfunk.com/applications/phood-buddy/documentation/concept-of-operations-v0.html>

For specifics on how the software development will be carried out, see the Project Management Plan document at: <http://www.williamrobertfunk.com/applications/phood-buddy/documentation/project-management-plan-v0.html>

**Introduction: Applicable Standards**

For a more detailed list of intended standards, refer to the project management plan. The standards that rise above the others, however, are as follows:

* User Interface - this will not be measured by "size", but quality instead. The software will provide an easy to use, yet enjoyable, interface for the user to have maximum accessibility.
* Android Mobile – The application will have Android Mobile capabilities that reach maximum performance by industry standards.
* Windows Mobile - The application will have Windows Mobile capabilities that reach maximum performance by industry standards.
* Website – The application’s website portion will reach maximum performance by industry standards as it will be used by iOS devices, and desktop computers.
* Database - We will design a database that follows industry standards.

**Project Overview: Assumptions**

This software assumes the following third-party applications will remain in-service, and free for our use, during the intended life expectancy of this product. It’s expected to use:

* Firebase (NoSQL) as its main database.
* JSON as the syntax for communicating with the database.
* PHP as the language of choice for the backend data-access layer.
* Java as the language of choice for the Android Mobile portion of the application.
* C# as the language of choice for the Windows Mobile portion of the application.
* HTML/CSS/JavaScript as the language combination for the website portion of the application.
* AJAX as the means of communication between website front-end and PHP backend.
* FatSecret as a main API for the software’s access to large quantities of recipes.
* Fitbit API to access user’s specific health accomplishments.
* Walmart API to check best deals on ingredients needed.

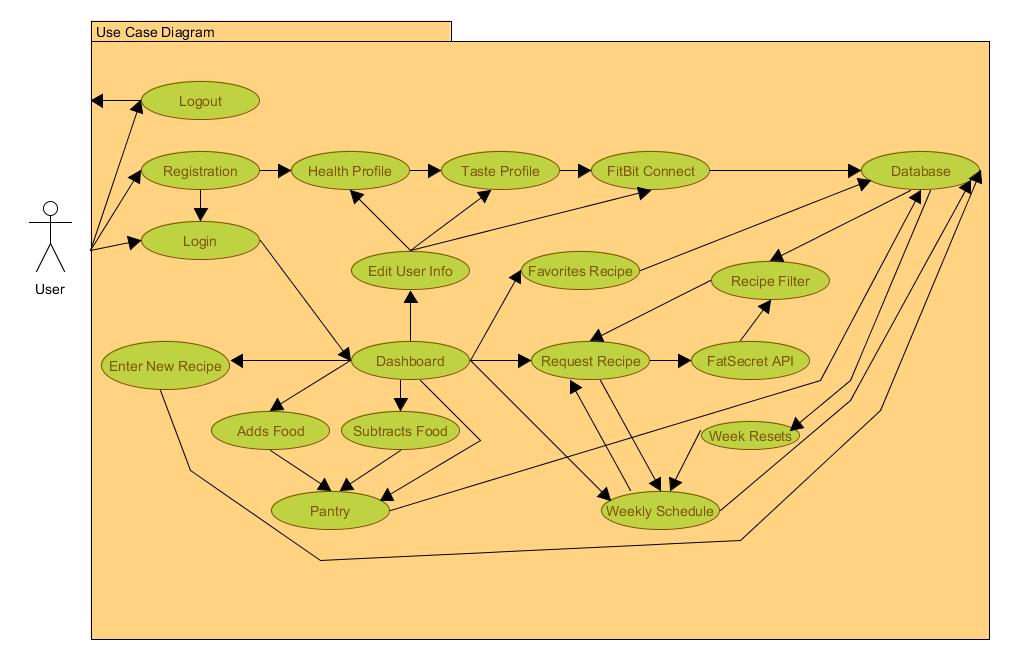
**Project Overview: Stakeholders**

* U.S. Food and Drug Administration: This regulatory agency would have concern if we were promoting unregulated foods (some are outlawed, others considered toxic) without the proper disclaimers to the end users. To address this concern, the software will provide adequate warnings all recommended recipes.
* U.S. Copyright Office: This agency would have concern over the unauthorized distribution of the recipes themselves. It could be argued the recipes were the intellectual property of those who posted them for consumption in various online resources or databases. This software doesn’t violate any of these laws as recipes are classified as exempt from copyright protection under *Publications Intl. v. Meredith, 88 F.3d 473 (7th Cir. 1996)*.
* American Dietetics Association: An organization determined to promote better health through better eating ([www.eatright.org](http://www.eatright.org)) would naturally have interest in an application that helps the common user to have easier access to a variety of healthier recipes. This application offers the ability to filter recipes by certain health criteria, and therefore, promote this similar mindset.
* The health-conscious user: Much like the ADA mentioned above, this user is concerned with achieving a healthier lifestyle and seeks resources to make that goal easier to achieve. This software will remove many of the barriers currently in place to this class of user. By offering detailed instructions and ingredients, alongside deals of those ingredients, we eliminate some of the more tedious tasks from the user’s plate.
* The foodie: This class of user is a lover of food. Their interest is in exploring the full range of the human palette and seeks out any resource that will give them access to more recipes. At its core, this software offers an almost infinite number of recipes; certainly more than any one person could try in their lifetime.
* The food-bored/indecisive user: This class of user is either tired of the monotony of choosing from the same list of meals they have in the past, or has difficulty making a choice from so many options available. This application has the ability to randomly select recipes for the user to put back the spice in their mealtime, and it doesn’t require any decision-making from the user beyond the decision to use the software.

**Project Overview: Event Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Event Name | External Stimuli | External Responses | Internal data and state |
| New User Registration | N/A | N/A | New user is entered into database. New rows added into respective database directories. |
| User Log in | N/A | N/A | N/A |
| User Logs out | N/A | N/A | N/A |
| Browsing Application | N/A | N/A | N/A |
| User Requests Recipe | N/A | FatSecret API returns data from their database. | API call collects large number of recipes based on category. Recipes are filtered based on health and taste before returning one recipe back to user. User is presented with image of meal, list of ingredients, and directions. |
| User Rates Recipe | N/A | N/A | Recipe is saved in database with user comments and five-star rating. |
| User Favorites a Recipe | N/A | N/A | Recipe is saved in database. |
| User Receives Deals for Ingredients | N/A | Various grocery store API returns with list of deals. | API call sends specific item requests for deals. |
| User Enters Food into Pantry | N/A | N/A | Food is stored in Pantry section of database. |
| User Subtracts Food from Pantry | N/A | N/A | Food is removed from Pantry section of database. |
| User Requests to See Pantry | N/A | N/A | Database returns user’s previously inputted content of their personal food inventory. |
| User Creates Weekly Recipe Schedule | N/A | N/A | Recipes based on Monday through Sunday (breakfast, lunch, and dinner) are stored in database. |
| User Edits Weekly Schedule | N/A | N/A | Database is updated with changes. |
| New Week Resets User’s Weekly Schedule | Time Passes | N/A | Database wipes old weekly schedule from database. |
| User Edits Personal Info | N/A | N/A | Database is updated. |
| User Enters New Recipe | N/A | N/A | New recipe is added to database. |

**Project Overview: Use Case Diagram**



**Project Overview: Use Case Descriptions**

* New User Registers: User creates a username and login, adding specific info such as email address. They complete a short medical questionnaire, or skip it. They complete a taste profile, or skip it. They provide access to their FitBit account, or pass. This info is entered into the database.
* User Logs in: After having registered, the user will be able to access the application with a username and password.
* User Logs out: Once logged in, the user can exit the application at any time by logging out.
* Browsing Application: Once logged in, the user can visit any of the sites areas that they've registered access accounts for.
* User Requests Recipe: User requests a recipe be presented to them. This is either a stand-alone request, or one associated with completing the Weekly schedule. User’s recipe specific data is collected from the database, and an API call is made to FatSecret for a host of recipes. The user’s data from the database is used to filter those results. From the remaining choices, a recipe is chosen at random and presented to user.
* User Rates Recipe: With a recipe already highlighted, the user clicks on a star between 1 and 5 for quality. They then again for the five taste categories: salty, sweet, bitter, sour, and spicy. This is used to better identify recipes to user later on. This information, and the recipe id associated to it by FatSecret is stored in the database.
* User Favorites a Recipe: With a recipe already highlighted, the user clicks on a heart (checkbox) to signify they want it favorited. The recipe id associated to it by FatSecret is stored in the database with an association to the user.
* User Receives Deals for Ingredients: Once a user has selected an offered recipe, an API call is made to Walmart (or related service), with the recipe’s ingredients as parameters, and the results are displayed to the user.
* User Enters Food into Pantry: User alerts application when they’ve acquired certain food items, either through the scan feature, or manually. This data is stored into the database with an association to that user.
* User Subtracts Food from Pantry: User alerts application when they’ve used/eliminated certain food items from their inventory. This data is stored into the database with an association to that user.
* User Views Pantry: User selects to view the current contents of their Pantry. This data is collected from the database and displayed to the user.
* User Creates Weekly Recipe Schedule: From the week schedule section, user selects from the breakfast, lunch, or dinner sections of a specific Monday-Sunday day. User is offered the chance to enter a recipe of their choice (either new or from favorited recipes), or to have one found for them. The user confirms their choice and that meal section is populated. When user is satisfied, they submit, and that data is stored in the database with an association to that user.
* User Edits Weekly Schedule: User visits their existing weekly schedule, and from there clicks on a field (either populated or blank). User is given the same options available when creating a new weekly schedule. After submission, the database is updated with the changes.
* New Week Resets User’s Weekly Schedule: After midnight on Sunday, the database purges the stored weekly schedule for all users.
* User Edits Personal Information: Sometime after conclusion of the registration process, a logged in user can edit their profile data, adding or removing access to various accounts (ie. FitBit). These changes are updated in the database.
* User Enters New Recipe: User elects to manually enter a recipe into the system. They upload an image of the meal, a title to the recipe, a list of ingredients with associated quantities, a list of directions to follow. They click on a star rating system from 1 to 5 for each of the five tastes (sweet, sour, bitter, salty, and spicy) to associate it to a taste for other users.

**Specific Requirements: Functional**

|  |
| --- |
| No: 1.1 |
| Statement: The Android system shall be written in XML, and native Java, languages |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Developer continuously checks to make sure the language used is the language required |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.2 |
| Statement: The Android system shall parse data from the recipe API |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.3 |
| Statement: The Android system shall be able to connect with FitBit API to collect user data |
| Source: Developer |
| Dependency: FitBit API |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.4 |
| Statement: The Android system shall be able to connect with application through Google, Facebook, and Twitter |
| Source: Developer |
| Dependency: Google, Facebook, and Twitter APIs |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.5 |
| Statement: The Android system shall be able to connect to database in 1000ms or less |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.6 |
| Statement: The database shall be constructed in such a way that all query calls return within 1000ms |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.7 |
| Statement: The database shall handle both one-way and two-way relationships through denormalization |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.8 |
| Statement: The database shall follow a “flattened hierarchy” design |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.9 |
| Statement: The Windows system shall be able to connect with application through Google, Facebook, and Twitter |
| Source: Developer |
| Dependency: Google, Facebook, and Twitter APIs |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Windows developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.10 |
| Statement: The system shall provide user with ability to add food type and quantity to their inventory via keyboard input, and the scanning of barcodes |
| Source: Developer |
| Dependency: Barcode translation API |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.11 |
| Statement: The system shall provide user with ability to remove food type and quantity from their inventory via keyboard input |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.12 |
| Statement: The Android system shall be able to connect to recipe repository API in 1000ms or less |
| Source: Developer |
| Dependency: Android SDK & FatSecret API |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.13 |
| Statement: The Android system shall support multiple languages |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.14 |
| Statement: The Android system shall be able to send and push notifications to the user from database |
| Source: Developer |
| Dependency: Android SDK & Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android and database developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.15 |
| Statement: The Android system shall use Gradle to build |
| Source: Developer |
| Dependency: Android SDK & Gradle |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 1.16 |
| Statement: The Android system shall auto login users after first login |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Interface**

|  |
| --- |
| No: 2.1 |
| Statement: The system shall have a PHP back-end layer capable of receiving data from the AJAX front-end for the website side of the application |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.2 |
| Statement: The system shall have a PHP back-end layer capable of sending and receiving data to and from the database |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.3 |
| Statement: The system shall have a PHP back-end layer that validates user input before sending user requests to database |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.4 |
| Statement: The system shall have a PHP back-end layer that implements error-handling for database connection and database retrieval/update exceptions |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.5 |
| Statement: The system shall have a PHP back-end layer that maintains the integrity of data stored in database |
| Source: Developer |
| Dependency: Firebase database |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.6 |
| Statement: The system shall have a PHP back-end layer that ensures no more than 5000 API calls are made to the database per day |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Back-end developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.7 |
| Statement: The system shall handle user authentication using stored data with in database |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.8 |
| Statement: The system shall handle user authentication from Facebook |
| Source: Developer |
| Dependency: Facebook API |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.9 |
| Statement: The system shall handle user authentication from Walmart |
| Source: Developer |
| Dependency: Walmart API |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.10 |
| Statement: The system’s database shall provide customized API calls for Android developer access |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all Android and database developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.11 |
| Statement: The system’s database shall provide customized API calls for Windows developer access |
| Source: Developer |
| Dependency: Windows SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all Windows and database developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.12 |
| Statement: The system shall allow a user to rate recipes and search for recipes by rating |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 2.13 |
| Statement: The system shall allow a user to post their own recipes |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Physical**

|  |
| --- |
| No: 3.1 |
| Statement: The system shall have capacity to run on the Android Mobile device |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.2 |
| Statement: The system shall have capacity to run on the Windows Mobile device |
| Source: Developer |
| Dependency: Windows SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Windows developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.3 |
| Statement: The system shall have capacity to run on the on the iOS Mobile (through the Safari browser) |
| Source: Developer |
| Dependency: iOS |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by web development team |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.4 |
| Statement: The system shall have capacity to run on the Google Chrome, Mozilla Firefox, and Microsoft Internet Explorer and Edge browsers for desktop and laptop computers |
| Source: Developer |
| Dependency: Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, and Microsoft Edge |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by web development team |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.5 |
| Statement: The Android system shall have access to the camera on mobile device |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.6 |
| Statement: The Windows system shall have access to the camera on mobile device |
| Source: Developer |
| Dependency: Windows SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Windows developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.7 |
| Statement: The Android system shall have access to the wifi and network state on mobile device |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 3.8 |
| Statement: The Windows system shall have access to the wifi and network state on mobile device |
| Source: Developer |
| Dependency: Windows SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Windows developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: User and Human Factor**

|  |
| --- |
| No: 4.1 |
| Statement: The application must have a custom health profile for user |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Figure 2.7 |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 4.2 |
| Statement: The application must have a custom login page for user (if they lack Google, Facebook, or Twitter accounts) |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Figure 2.7 |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 4.3 |
| Statement: The application must have a custom navigation drawer page |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Figure 2.7 |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 4.4 |
| Statement: The application must a custom profile setup page to include personalized taste profile |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Figure 2.7 |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 4.5 |
| Statement: The application shall be free to use on all available platforms |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Figure 2.7 |
| Evaluation Method: The requirements will be tested on the system by all five developers |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Documentation**

|  |
| --- |
| No: 5.1 |
| Statement: The project shall maintain an updated webpage Concept of Operations document. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: <http://www.williamrobertfunk.com/applications/phood-buddy/documentation/concept-of-operations-v0.html> |
| Evaluation Method: Version number and date of modification history on document |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 5.2 |
| Statement: The project shall maintain an updated webpage Project Management Plan document. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: <http://www.williamrobertfunk.com/applications/phood-buddy/documentation/project-management-plan-v0.html> |
| Evaluation Method: Version number and date of modification history on document |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 5.3 |
| Statement: The project shall maintain an updated webpage Software Requirements Specification document. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: http://www.williamrobertfunk.com/applications/phood-buddy/documentation/srs-v0.html |
| Evaluation Method: Version number and date of modification history on document |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 5.4 |
| Statement: The project shall maintain an updated webpage High-Level Design document. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: Yet to be created |
| Evaluation Method: Version number and date of modification history on document |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 5.5 |
| Statement: The project shall maintain an updated webpage Low-Level Design document. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: Yet to be created |
| Evaluation Method: Version number and date of modification history on document |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Data**

|  |
| --- |
| No: 6.1 |
| Statement: This system shall retain user’s taste profile |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.2 |
| Statement: This system shall retain user’s favorite recipes |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.3 |
| Statement: This system shall retain user’s FitBit data |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.4 |
| Statement: This system shall retain user’s Health information |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.5 |
| Statement: This system shall retain user’s submitted recipes |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.6 |
| Statement: This system shall retain user’s food inventory |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 6.7 |
| Statement: This Android system shall be able to store preference |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Resource**

|  |
| --- |
| No: 7.1 |
| Statement: The system shall have exactly five developers |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: See Team Members section of this document |
| Evaluation Method: All developers will have their sections to produce by their respective deadlines with these requirements as their requisites. The meeting of these deadlines and the satisfaction of those requirements will be their evaluation. |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Security**

|  |
| --- |
| No: 8.1 |
| Statement: The Android system shall have a 128 bit encryption for all server requests |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 8.2 |
| Statement: The system shall prevent unauthorized parties from accessing users’ private information |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 8.3 |
| Statement: The system’s database shall store roles for users to prevent unauthorized access |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by database developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Specific Requirements: Quality Assurance**

|  |
| --- |
| No: 9.1 |
| Statement: All developers shall continuously test the functionality of the application. |
| Source: Developer |
| Dependency: None |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: Android developer will test Android Mobile Windows developer will test Windows Mobile. Web development team will test all other forms. This will be accomplished through using the applications functions in the same manner that the user would—alpha testing. |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 9.2 |
| Statement: The Android system shall have crash analytics |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 9.3 |
| Statement: The Android system must be tested on AVD emulators to ensure responsiveness |
| Source: Developer |
| Dependency: Android SDK |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

|  |
| --- |
| No: 9.4 |
| Statement: The Android system shall meet screen density requirements |
| Source: Developer |
| Dependency: Android System Requirements |
| Conflicts: None |
| Supporting Materials: None |
| Evaluation Method: The requirements will be tested on the system by Android developer |
| Revision History: William Funk --- 02/26/2016 --- Initial Proposal |

**Supporting Material:**

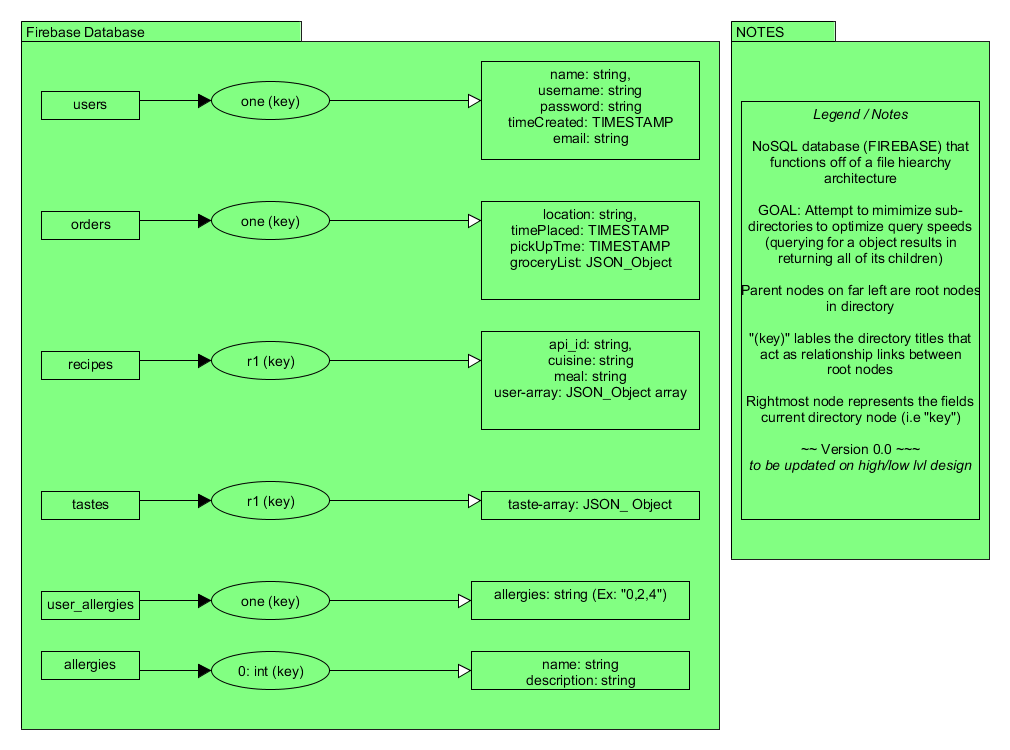
Figure 1: Preliminary Database Design 

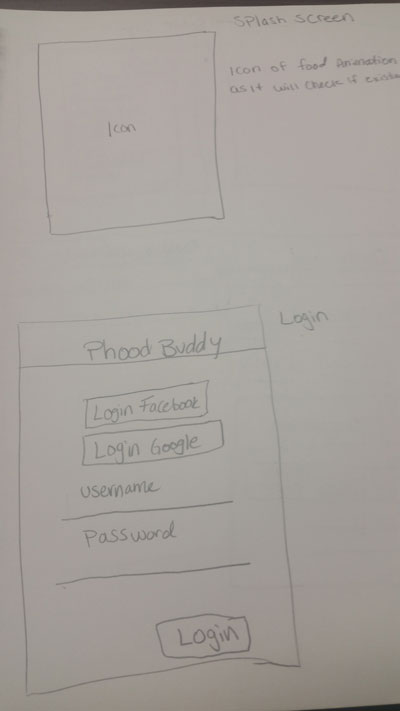
Figure 2.1: Preliminary Layout Design (Splash Screen & Login) 

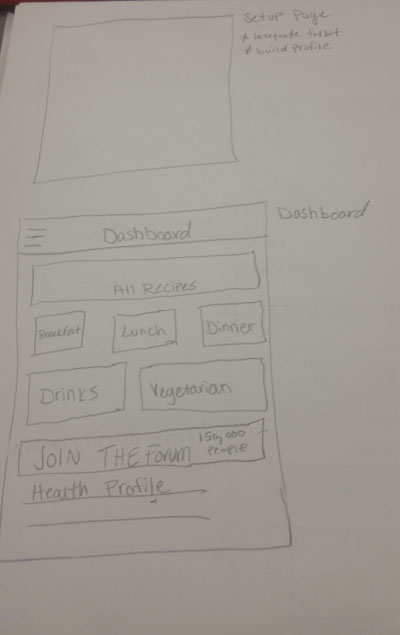
Figure 2.2: Preliminary Layout Design (Setup & Dashboard) 

Figure 2.3: Preliminary Layout Design (Registration)

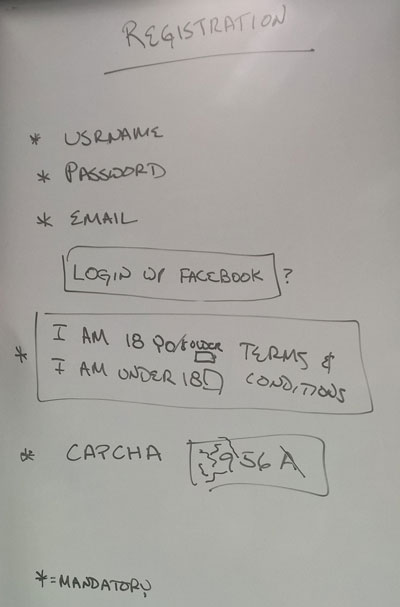


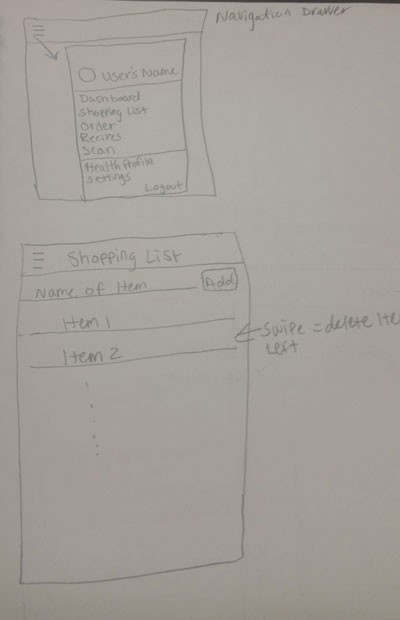
Figure 2.4: Preliminary Layout Design (Navigation Drawer & Shopping List) 

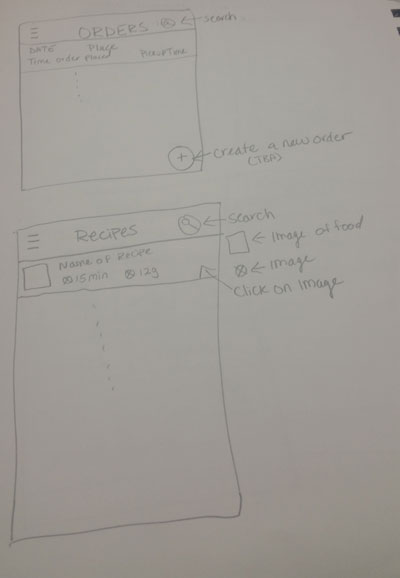
Figure 2.5: Preliminary Layout Design (Orders & Recipes 1) 

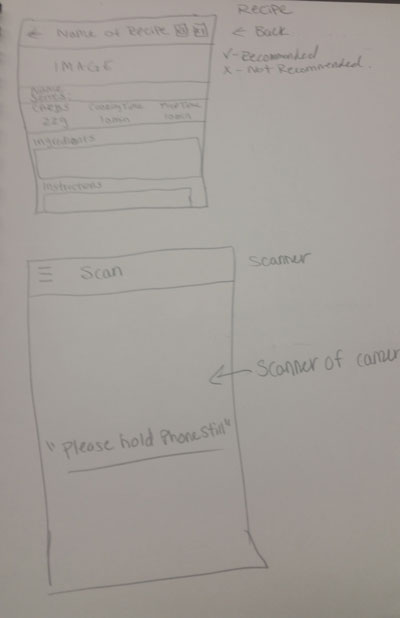
Figure 2.6: Preliminary Layout Design (Recipes 2 & Scanner) 

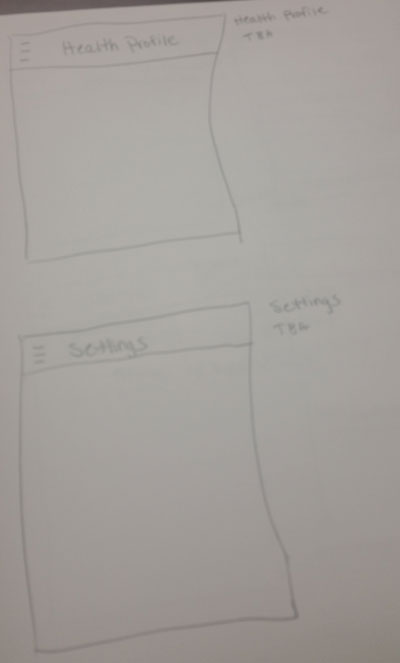
Figure 2.7: Preliminary Layout Design (Health Profile & Settings) 

Figure 2.7: Preliminary Feature Design (Recipe Slot Machine)



Figure 3.2: Preliminary Feature Design (Taste Profile)

