

# SFWR 4G06A - System Requirements

Group 9

Gundeep Kanwal 400015267

Ivan Bauer 001418765

Yousaf Shaheen 400026476

Scott Williams 400031554

Lucas Shanks 40002994

October 28, 2019

# Contents

<b>1 Purpose</b>	<b>5</b>
<b>2 Scope</b>	<b>5</b>
<b>3 Context Diagram Showing Boundaries</b>	<b>6</b>
<b>4 Monitored and Controlled Variables</b>	<b>7</b>
4.1 Monitored Variables . . . . .	7
4.2 Controlled Variables . . . . .	8
<b>5 Constants</b>	<b>9</b>
<b>6 Diagrams Showing Functional Decomposition</b>	<b>9</b>
<b>7 Functional Requirements with Rationale</b>	<b>10</b>
<b>8 Non-Functional Requirements</b>	<b>11</b>
8.1 Performance Requirements . . . . .	11
8.1.1 NFR1 - Speed and Latency Requirements . . . . .	11
8.1.2 NFR2 - Scalability or Extensibility Requirements . . . . .	11
8.1.3 NFR3 - Workload Requirements . . . . .	11
8.1.4 NFR4 - Platform Requirements . . . . .	11
8.2 NFR5 - Security Requirements . . . . .	11
8.3 NFR6 - Look and Feel Requirements . . . . .	12
8.3.1 Appearance Requirements . . . . .	12
<b>9 Normal Operation</b>	<b>12</b>
9.1 Description/Behaviour . . . . .	12
9.2 Notation . . . . .	12
9.3 Normal Use Cases/Scenarios . . . . .	12
9.3.1 User Logs into NutriBud Account . . . . .	12
9.3.2 Scan Food Barcode . . . . .	12
9.3.3 Set Up GainRivals Friend Group . . . . .	14
9.3.4 View GainRivals Friend Leaderboard . . . . .	14
9.3.5 Manual Lookup Food Item . . . . .	14
9.3.6 Take Picture of Food . . . . .	14
9.3.7 Create User Account . . . . .	14
9.3.8 Log Weight Entry . . . . .	14
9.3.9 Log Food Diary Entry . . . . .	14
9.3.10 View Weight Entry . . . . .	14
9.3.11 View Daily Food Diary . . . . .	14
9.3.12 View Fitness Timeline . . . . .	14
9.3.13 View Nutrition Timeline . . . . .	14
<b>10 Undesired Event Handling</b>	<b>15</b>
10.1 Barcode not Detected in Camera's Boundary Box . . . . .	15
10.2 Barcode not Recognized . . . . .	15
10.3 Failure to Detect Food Item . . . . .	15
10.4 Failure to Identify Food Item . . . . .	15
10.5 Failure to Identify Portion Size of Food Item . . . . .	15
10.6 Failure to Detect Multiple Food Item Boundaries . . . . .	16
10.7 System Failure . . . . .	16
<b>11 List of Requirements that are Likely to Change</b>	<b>16</b>

<b>12 List of Requirements that are not Likely to Change</b>	<b>16</b>
--	-----------

## List of Tables

2	NutriBud Monitored Variables Table . . . . .	7
3	NutriBud Controlled Variables Table . . . . .	8
4	Functional Requirements Table . . . . .	10

## List of Figures

1	NutriBud Context Diagram . . . . .	6
2	NutriBud Functional Decomposition Diagram . . . . .	9
3	NutriBud Use Case Diagram . . . . .	13

## Revision History

Revision	Date	Author(s)	Description
1.0	10/28/2019	GK, IB, YS, SW, LS	Initial revision of the System Requirements document

# 1 Purpose

This project will involve developing and designing a mobile application that is capable of retrieving and effectively visualizing nutritional information of food to the user, enabling the user to make more health conscious decisions regarding their food selection. This information shall be used to construct a food diary which the user will use in order to track their food intake on a day-to-day basis. The application will provide a high level visualization of a user's fitness and nutritional habits through an interactive timeline. Ultimately, the overall objective is to help the user lead a healthier lifestyle.

# 2 Scope

This project shall primarily be centered on enabling the user to understand the nutritional content of the food that they are eating. The system shall be able to figure out the nutritional information of the food that the user captures with their picture, or scans with a bar-code. It shall be able to summarize a day's nutritional intake and visualize that information to the user.

In-scope items of functionality for the system include the following:

- Account system with various settings that can be personalised.
- Daily food diary which tracks nutritional information.
- Camera feature which takes pictures of a single food item.
- Ability to recognize different types of food items of a single sample.
- Ability to display nutritional content of the food from a picture.
- Transference of nutritional information from picture/bar-code to diary.
- Visualization of the fitness progress of a user.
- Visualization of the nutritional progress of a user.
- Game system that enables the user to compete with others.
- Camera feature that detects bar-code of food or drink.

Additionally, the following items are deemed to be out of scope:

- Determining nutritional contents of a drink through a picture of the item.
- Viewing Fitness Scores of users outside their immediate group.
- Determining every nutritional value in a given plate of food.

### 3 Context Diagram Showing Boundaries

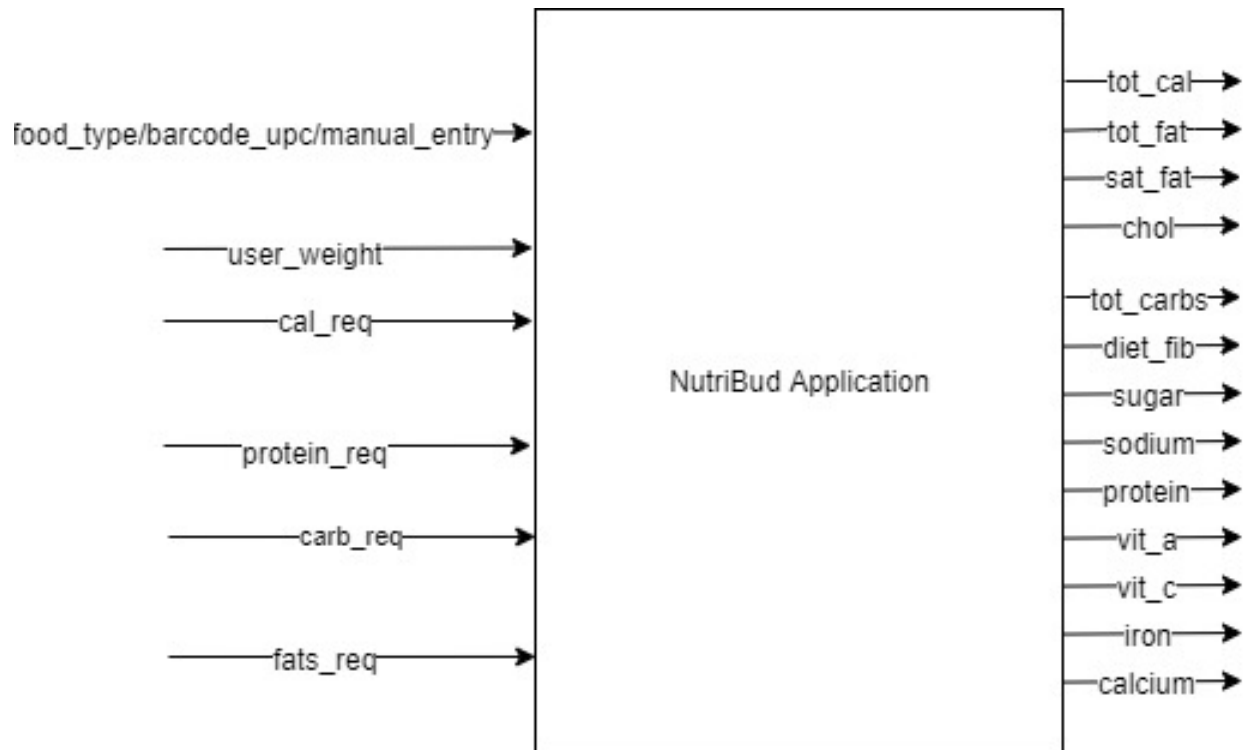


Figure 1: NutriBud Context Diagram

## 4 Monitored and Controlled Variables

### 4.1 Monitored Variables

Variable	Unit	Description
food_type	N/A	The food type that is being placed on the plate.
barcode_upc	integer	The UPC barcode of the food product.
manual_entry	word	The worded user input for the food type.
user_weight	lbs/kgs/st	The recorded weight of the user.
cal_req	Calories (kcal)	The recorded number of calories necessary for the user to achieve nutrition projections, as documented by the user.
protein_req	g	The recorded number of protein necessary for the user to achieve nutrition projections, as documented by the user.
carb_req	g	The recorded number of carbohydrates necessary for the user to achieve nutrition projections, as documented by the user.
fats_req	g	The recorded number of fat necessary for the user to achieve nutrition projections, as documented by the user.

Table 2: NutriBud Monitored Variables Table

## 4.2 Controlled Variables

Variable	Unit	Description
tot_cal	Calories (kcal)	Defined as the amount of energy needed to raise the temperature of 1 kilogram of water by 1 degree Celsius. Provided as an energy measurement for food.
tot_fat	g	The total amount of fat that can be found within the serving size of the dish.
sat_fat	g	The amount of saturated fats contained within the serving size of food.
trans_fat	g	The amount of trans fat in the food serving size.
chol	mg	The amount of cholesterol within .
tot_carbs	g	The total amount of carbohydrates found in grams.
diet_fib	g	The amount of dietary fiber found in the serving size, in grams.
sugar	g	The total amount of sugars found in the serving size, in grams.
sodium	g	The amount of sodium found in the serving size, in grams.
protien	g	The amount of protein found in the serving size, in grams.
vit_a	%	The percentage of Daily Value the serving size offers Vitamin A to the user.
vit_c	%	The Daily Value percentage of Vitamin C to the user according to their restrictions.
calcium	%	The Daily Value of Calcium held within the serving size to the user.
iron	%	The Daily Value of Iron according to the user, in percentage.

Table 3: NutriBud Controlled Variables Table



## 5 Constants

Due to the dynamic nature of the application, it has been determined that there will be no constants for the system.

## 6 Diagrams Showing Functional Decomposition

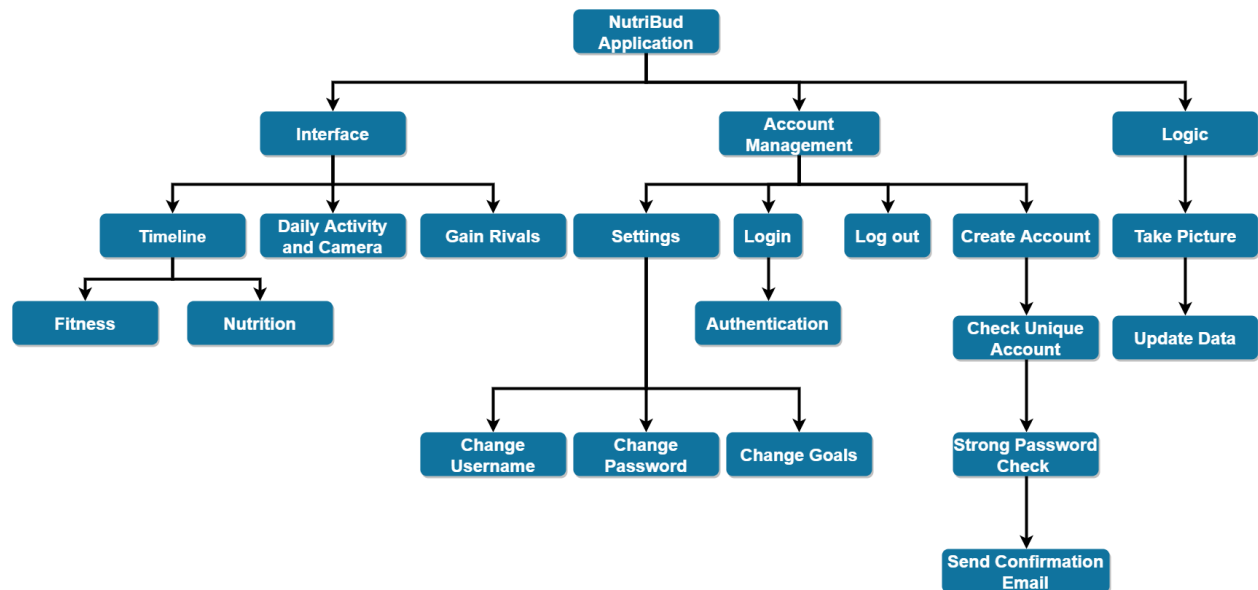


Figure 2: NutriBud Functional Decomposition Diagram

**NutriBud Application:** The overall goal is to create an application to determine and record the nutrition of meals had by the user

**Interface:** The user interface for interacting with the camera and viewing past statistics

**Timeline:** Interface to select between past fitness data or past nutrition data

**Fitness:** Graphical interface displaying fitness history (eg. Weight)

**Nutrition:** Graphical interface displaying nutrition history (eg. Calories, Fat, Protein, Carbohydrates)

**Daily Activity and Camera:** Interface where you see the nutrition statistics calculated for the current day and the ability to select the camera

**Gain Rivals:** Interface to show competing data between friends that are in the same group

**Logic:** Where the back-end functions will be run

**Take Picture:** After picture is taken calculations are run on food to determine characteristics

**Update Data:** Saves picture data from previous calculation into database

## 7 Functional Requirements with Rationale

Requirement ID	Functional Requirement	Rationale
FR1	The application shall enable the user to take a picture of a single food item that will be processed by the application	Allows the user to input a food item in a time efficient manner.
FR2	The system shall predict food items with a degree of certainty that shall be revealed to the user after image processing.	Allows the application to determine the food item captured by the user.
FR3	The application shall retrieve corresponding nutritional information to the most likely food item predicted.	Provides critical nutritional data for the application and the user.
FR4	The application shall keep a daily record of the nutritional information of meals consumed by the user	Provides a daily view of a user's food intake to help them make more informed nutritional decisions throughout their day.
FR5	The application shall recognize bar code images and retrieve corresponding food item nutritional data.	Provides nutritional data to the user for given product serving size.
FR6	The application shall allow the user to overwrite model predicted food quantity/nutritional values	Allows the user to amend values in case of incorrect model prediction.
FR7	The application shall allow the user to visualize their fitness progress on a timeline over user determined date ranges.	Allows the user to track fitness history.
FR8	The application shall allow the user to visualize their dietary progress on a timeline over user determined date ranges.	Allows the user to track food consumption history.
FR9	The application shall allow the user to create a NutriBud account using an email address and created password.	Provides a more personal user experience and enables storage of data on the cloud.
FR10	The application shall enable a user to select a picture from their phone's photo gallery. The picture will be processed by the NutriBud system and the predicted food items and corresponding nutritional information will be provided to the user.	The application must allow the user to provide a picture and predict corresponding nutritional information. If the user has taken a picture beforehand, they should be allowed to upload it for processing
FR11	The system shall offer users to create a leaderboard with a private friend group called GainRivals. The scoring metrics for GainRivals shall be customizable by the user according to their preferences.	Provides an engaging, competitive aspect to the application for the user's benefit
FR12	The NutriBud system shall detect and generate nutritional contents for multiple, distinct food items within a picture.	Provides a more time-efficient method of identifying food items for the user.

Table 4: Functional Requirements Table

## 8 Non-Functional Requirements

### 8.1 Performance Requirements

#### 8.1.1 NFR1 - Speed and Latency Requirements

1. The time to return a prediction of a food item and nutritional data given a picture shall take less than 8 seconds for 90% of cases
2. Program start-up shall take less than 5 seconds
3. Buttons/options shall respond in under 0.5 second to allow users flow of thoughts to remain uninterrupted

The above requirements are set to ensure the application will run and do tasks within reasonable speed. This is important because the user needs to be able to input and log data without issue. Having to use an application that is slow will degrade the user experience.

#### 8.1.2 NFR2 - Scalability or Extensibility Requirements

1. New features/ideas should be easy to incorporate into the existing application
2. The program shall be modular with the separation of concerns, allowing seamless addition of new modules
3. Software updates and new versions shall preserve the integrity of user data

The above requirements are implemented to ensure the future of the application is stable. Having an application that is easy to update and one that is modular will help in updating, patching, and overall upgrading the application to improve the user experience.

#### 8.1.3 NFR3 - Workload Requirements

1. The system's database shall be able to accommodate the data of 200 unique users
2. In standard workload, the CPU usage of the mobile device should be less than 50% to ensure available resources for other background tasks

The above requirements are set in order to ensure that multiple users can use the application and the system will be able to accommodate all of them with relative ease.

#### 8.1.4 NFR4 - Platform Requirements

1. This application shall run on mobile devices with Android OS with a minimal version of 7.0

This is to ensure users with older phones can also use the application, in the future this application may also be migrated to iOS.

### 8.2 NFR5 - Security Requirements

1. The application will store user submitted pictures in a secure setting.
2. The application will keep user profile information private.

This will keep personal data safe and provide restricted access to said data. To comply with privacy laws, the user data will need to be stored securely.

## 8.3 NFR6 - Look and Feel Requirements

### 8.3.1 Appearance Requirements

1. The function of buttons/options in the application must be obvious.
2. The application must have a simple GUI such that a new, inexperienced user to NutriBud can submit a single food item's nutritional contents to a food diary within 3 minutes.
3. The application must be intuitive and easy to learn.

This is to ensure the user can easily learn to use the application and finds using the application to be a pleasant experience.

## 9 Normal Operation

### 9.1 Description/Behaviour

The system must be able to, upon taking a picture of a food item on a plate, predict the particular type of food that is on the plate along with quantity either numerically or volumetrically. Afterwards, the system must use this data to accurately deliver the nutritional contents of the food item to the user on an haptic input screen that allows for manual override of the serving size. These nutritional contents are outlined through the context diagram on Figure 2.

Alternatively, the system must either allow the user to scan a UPC barcode through the phone's camera hardware or allow the user to interact with the phone's haptic touch screen to manually search for food products.

Additionally, the system will allow the user to track and visualize their fitness and nutritional progress. They shall be able to plot their progress over a timeline, which they can modify the size of in order to fit the view they wish to look at it with.

Finally, the system must offer users the ability to create an account through their email address and add friends through username lookup. Using these friend groups, the system must allow users to create private Friend Groups to play a game called GainRivals, a customizable leaderboard setup that indicates the number of calories consumed by each group member and ranks them.

As a whole, the system must perform all of these behaviours accordingly.

### 9.2 Notation

GainRivals = The game feature of the application, consisting of a leaderboard with a custom score options.

Macros = Macro nutritional content which the application will be tracking, including fats, proteins and carbohydrates.

### 9.3 Normal Use Cases/Scenarios

#### 9.3.1 User Logs into NutriBud Account

The application is logged onto by the user, and will subsequently display the current food diary for the given day they log on.

#### 9.3.2 Scan Food Barcode

The system will detect if a barcode in the camera view appears within a centered box boundary. Afterwards, the system will transfer the user to the logging screen for the daily food diary after retrieving the nutritional information according to the food.

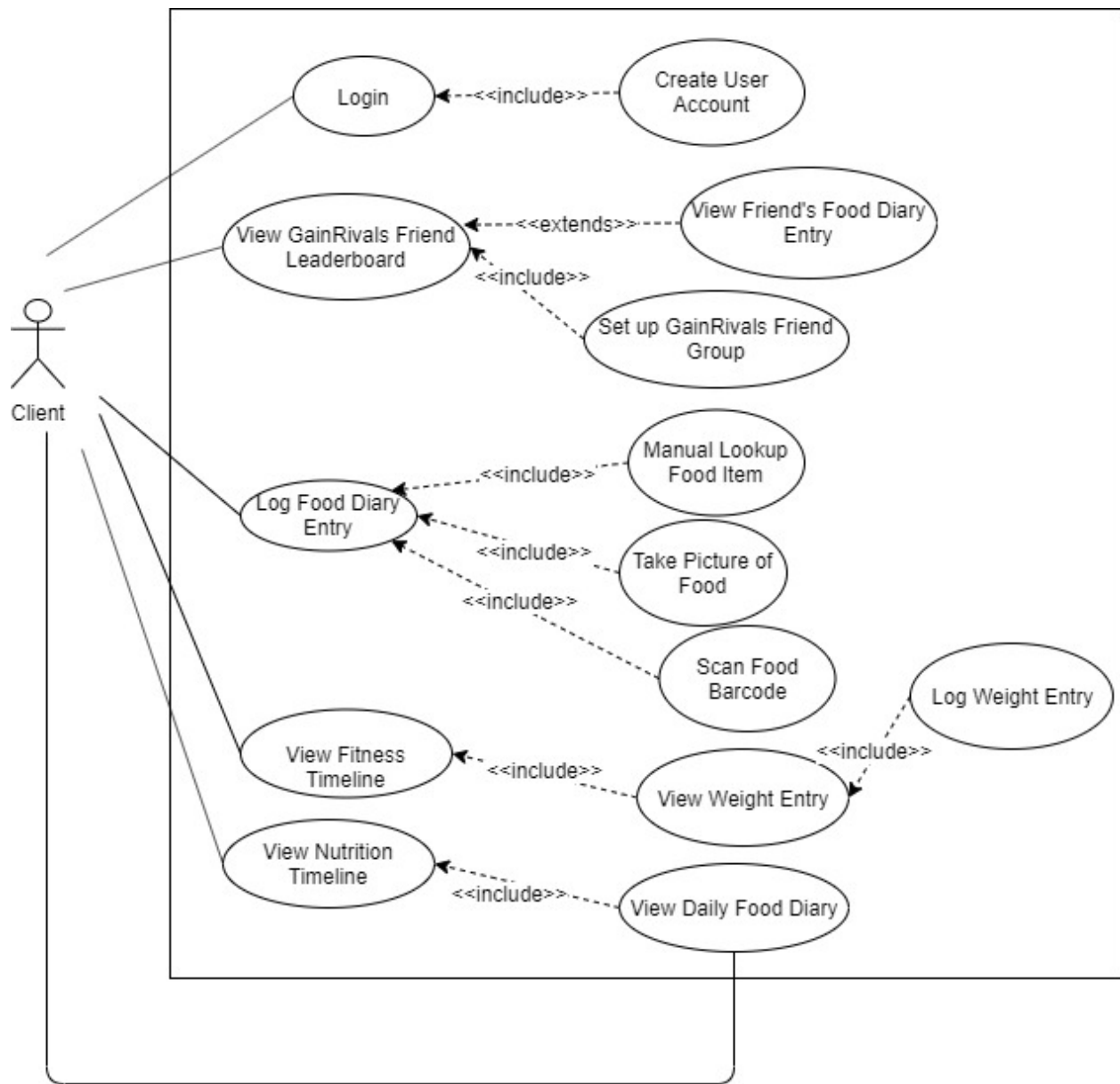


Figure 3: NutriBud Use Case Diagram

### **9.3.3 Set Up GainRivals Friend Group**

The system will have the ability to let users add friends to their account. Upon doing this, the system shall also allow users to create custom lobbies for the GameRivals game, which promotes competitive play by displaying a leaderboard for how closely they meet a user predefined scoring regimen.

### **9.3.4 View GainRivals Friend Leaderboard**

The system will allow the user to proceed to GainRivals by pressing a button from the main menu. The system will allow users to select which friend group they want to view for leaderboard statistics.

### **9.3.5 Manual Lookup Food Item**

The system will be able to, upon the user selecting a search button, to produce a list of food items according to a filter that is provided by the user and described by the control variable `manual_entry`.

### **9.3.6 Take Picture of Food**

The system must be able to open the phone's camera and allow the user to take a picture of the food for the system to recognize and process for nutritional content display.

### **9.3.7 Create User Account**

The system must allow for the user to create a NutriBud account using an email address and password combination.

### **9.3.8 Log Weight Entry**

The system must allow for the user to log their weight on any day and store this data.

### **9.3.9 Log Food Diary Entry**

The system must allow for the user to log their nutritional consumption on any day and store this data.

### **9.3.10 View Weight Entry**

The system must allow the user to view a logged weight entry from the fitness timeline by selecting a day.

### **9.3.11 View Daily Food Diary**

The system must allow the user to view a logged food diary on any current or past date showcasing a breakdown of nutrient consumption.

### **9.3.12 View Fitness Timeline**

The system must create a fitness timeline to be shown for the user to view a history of all logged weight entries. This can be viewed on a custom time scale to be selected by the user.

### **9.3.13 View Nutrition Timeline**

The system must create a nutrition timeline that showcases all logged food diary entries by the user. Upon touch input from the user, the system will then display the food diary entry for that day

## 10 Undesired Event Handling

The application should fail elegantly under critical software failure and handle errors or exceptions encountered during application use.

### 10.1 Barcode not Detected in Camera's Boundary Box

- Camera is too close to the barcode to detect the full barcode
- Camera is too far from the barcode to recognize the barcode
- Visual conditions too poor for the barcode to be recognized by the camera.

If the barcode cannot be detected in the camera's boundary box, the system will instruct the user to reposition the camera.

### 10.2 Barcode not Recognized

- Barcode is damaged
- Picture quality of barcode is too low
- Barcode is not valid
- Barcode is not able to be resolved

If the barcode cannot be determined by the system, the system will form a dialog box that is displayed to the user. This will ask the user whether they wish to continue using the camera functionality or provide an alternate solution of manually looking up the food item they wish to detect.

### 10.3 Failure to Detect Food Item

- Food items are too close or too far from the camera
- Lighting conditions are too poor for the camera to detect the food item

If the application cannot detect the food item upon a user's request, the application shall prompt the user to retake their picture with the food being 3 feet away.

### 10.4 Failure to Identify Food Item

- Distorted food items are present and cause application to misidentify
- Image recognition model fails to accurately identify the food item type

If the NutriBud system misidentifies a food item, then the percentage in confidence should be given to the user in order to highlight the inaccurate nature of the prediction. The system should advise the user to manually enter the name of the food item to ensure the item is properly logged.

### 10.5 Failure to Identify Portion Size of Food Item

- Image processing model overestimates portion size of the food item
- Image processing model underestimates portion size of the food item

If the NutriBud application fails to identify the portion size of the food item, it will prompt the user to manually enter their portion size of the food that they are eating.

## **10.6 Failure to Detect Multiple Food Item Boundaries**

- Food items are too close together and not separately distinguishable by the system

If the application is unable to detect multiple food items, it shall ask the user to retake the picture from being 3 feet away.

## **10.7 System Failure**

- A bug/glitch causes the NutriBud application to crash and shut down
- A bug/glitch that distorts the computed nutritional values
- The application does not complete a task correctly

The application will save any completed tasks and/or data it can. During the next time the application is launched, it will resume the same task it was performing before the crash if possible.

## **11 List of Requirements that are Likely to Change**

FR 11-12 (Refer to Section 7) Both of these functional requirements were deemed likely to change as we defined these functionalities to be stretch goals. There is a chance that these requirements won't be within our means to accomplish.

## **12 List of Requirements that are not Likely to Change**

FR 1-11, NFR 1-6 (Refer to Section 7 and 8) The remaining requirements are all requirements that we have deemed to be viable and meaningful for our application.