

# *Fitness and Nutrition Buddy*



*Prepared by*  
**Shreya Boyapati, Ayesha Quadri Syeda, Dat Huynh, Sarthak Patipati**  
**(Group 9)**  
**for use in CS 440**  
**at the**  
**University of Illinois Chicago**

**Spring 2023**

## Table of Contents

List of Tables	3
I Project Description	4
1 Project Overview	4
2 Project Domain	4
3 Relationship to Other Documents	4
4 Naming Conventions and Definitions	6
4a Definitions of Key Terms	6
4b UML and Other Notation Used in This Document	6
4c Data Dictionary for Any Included Models	6
II Project Deliverables	7
1 First Release	7
2 Second Release	8
3 Comparison with Original Project Design Document	8
III Testing	8
1 Items to be Tested	8
2 Test Specifications	8
3 Test Results	12
4 Regression Testing	14
IV Inspection	14
1 Items to be Inspected	14
2 Inspection Procedures	15
3 Inspection Results	15
V Recommendations and Conclusions	15
VI Project Issues	16
1 Open Issues	16
2 Waiting Room	16
3 Ideas for Solutions	16
4 Project Retrospective	16
VII Glossary	17
VIII References / Bibliography	18
IX Index	19

## **List of Tables**

**References / Bibliography**

19

# I Project Description

## 1 Project Overview

The Fitness and Nutrition Buddy application is an application focused on nutrition.. This application will use the users current location and will be able to locate all of the nearest restaurants around them as well as using a filter for certain restrictions/cravings that they desire (for example low carbs, low fat, etc). This will benefit the user by allowing them to intuitively live a healthy lifestyle, while not having the hassle of searching for nearby restaurants. They also will not have to search through all of the results/each individual menu. It will also be able to keep track of the user's steps, and calories burned (similar to the Fitness app on the Apple Watch)

## 2 Project Domain

The project focuses on the nutritional aspect of the application. The application prototype would have the ability to track their nutritional goal throughout the day and look for restaurants nearby that provide the food according to their dietary restrictions. The application would also be able to provide suggestions for meals from breakfast to lunch to dinner.

## 3 Relationship to Other Documents

As the Project Description document describes, the application would work to help people with their dietary needs by looking for restaurants around them with the specific dietary requirements. The application is usable for anyone with the ability to use a smartphone. The application is able to recommend meals from different restaurants that align with the dietary restrictions. But the application doesn't allow for reviews to adjust the recommendations. The application also has scope to accommodate both IOS and android. An API similar to the google maps API was utilized in the application. The user would be able to update their daily macronutrient intake and set goals as a limit, but won't be able to track calorie burn or steps taken throughout the day. The database for the application is restricted to the restaurants within Illinois, USA but has the scope to expand and include restaurants around the world.

As described in the Project requirements document, the application is able to meet most functional requirements and data requirements. The application is able to make customized meal plans, use location services to search for restaurants nearby, help users keep track of their macronutrient intake, and has a database for user information, restaurant information, and meal information. The application meets most performance and dependability requirements. All functions of the application take less than 5 seconds to complete. The meals don't check whether they are causing the user to be over the limit in calories. The application doesn't constantly download the map with the user's location but will show the area of the default user location. There are no instances of failure to load user, meal, or restaurant information and can run for the whole day. It can even run without any connection to the internet. The

application doesn't show any security conditions around the user's location because it doesn't constantly track the user's location.

The application meets most maintainability, supportability, and security requirements. It doesn't need to be regularly updated and maintained and will be able to provide customer service if required, though it shouldn't be necessary. The application is able to create new meal plans with the click of a button and is estimated to be able to sustain thousands of users. It is also estimated that the application should be able to work for three years or more. All the data from the application is stored in a database and can only be accessed by the programmers that have access to that specific code. The user data should be easily recovered in the case of memory loss through git repositories. The users aren't able to access any data without logging into the system. So, as long as the user has the correct account information, they are able to access their data. So far there are no options to purchase in the application so there are no security measures taken to fulfill the audit requirements. The application is not accessible online and so doesn't have the need to create a firewall.

The application meets most usability, humanity, look and feel requirements. It is easy to use and doesn't require any special skills to operate. A person should be able to work the application within ten minutes of being introduced. Although currently the application is only available in English because all locations provided are constrained to be in Illinois, USA. It also doesn't have any built-in tutorials to work the applications since the use is intuitive to the current generation. The application doesn't have functionality to explain how all the calculations are performed because of time constraints during the building process. Anyone without a visual disability should be able to use this application. The application looks modern and follows a mint cream and gray color pattern and should appeal to most audience types.

The application meets most operational, environmental, cultural and political requirements. It works without an internet connection and the location of the user has no influence on the different applications. The users aren't able to change to different versions of the application through the different updates but have the scope to add this functionality. The application doesn't focus on a particular world culture and currently would only be accessible in the US and countries that have similar technological advancements.

The application follows the project design document. It utilizes the model view controller software architecture. Though the application doesn't have different parts for server and database. The GUI of the application looks similar to the once documented but the color schemes are different than what was originally planned. The assignment also doesn't have any subsystems.

The project issues document describes the different problems the application could run into through the development and usage. A lot of the issues described in the project issues document are dealt with or are not an issue with the prototype application because of limited requirements of functionality. The current application doesn't face storage issues as the amount of data is limited and it doesn't get updated

with new restaurant discoveries. It also uses many common libraries to create common components of the application.

## 4 Naming Conventions and Definitions

### 4a Definitions of Key Terms

**IOS:** An iphone operating system designed for use with Apple's multi touch devices.

**Android:** A Linux-based mobile operating system that primarily runs on smartphones and tablets.

**Meal:** A combination of various menu items provided to the user after requesting a food recommendation from the app.

**Meal Calories:** How many calories an entire meal has

**Carbs:** One of the 3 macros the app will track, carbohydrates, refers to anything that falls under the carbohydrates section of a nutrition label

**Fat:** One of the 3 macros the app will track, fat, refers to anything that falls under the fat section of a nutrition label

**Protein:** One of the 3 macros the app will track, protein, refers to anything that falls under the protein section of a nutrition label

**Macros:** All three of carbs fat and protein together make up a users macros that they set

**Location:** Where the default location of the user is.

**User:** The person currently using the app to track their nutrition.

### 4b UML and Other Notation Used in This Document

This document generally follows the Version 2.0 OMG UML standard, as described in, M. Fowler, UML Distilled, Third Edition, Boston: Pearson Education, 2004. Exceptions are noted in their specific cases.

### 4c Data Dictionary for Any Included Models

**MealCalories:** Sum of calories from every item in a meal.

**Calories allowed:** Calories limit - Calories consumed.

**Carbohydrates allowed:** Carbohydrates limit - Carbohydrates consumed.

**Fat allowed:** Fat limit - Fat consumed.

**Protein allowed:** Protein limit - Protein consumed.

**Calories limit:** Goal set by the user to consume a set amount of calories.

**Calories Consumed:** The calories consumed by the user after a meal.

**Carbohydrates limit:** Goal set by the user to consume a set amount of carbohydrates.

**Carbohydrates Consumed:** The carbohydrates consumed by the user after a meal.

**Protein limit:** Goal set by the user to consume a set amount of Protein.

**Protein Consumed:** The Protein consumed by the user after a meal.

**Fat limit:** Goal set by the user to consume a set amount of Fat.

**Fat Consumed:** The Fat consumed by the user after a meal.

## II Project Deliverables

This semester, our group has produced an application that is capable of assisting users with their nutritional tracking and with creating plans to meet their nutritional needs. The application has functionalities for setting a user's nutritional goals, creating a detailed meal plan based on said goal, providing the user's with locations of healthy food to suit their nutritional needs, and tracking the user's daily caloric and macronutrient intake. These functions are easy-to-use and intuitive to users in order to maximize the effectiveness of the app. The application is a streamlined tool that will provide users with all they need to have a healthy, balanced diet that suits their particular nutritional needs.

### 1 First Release

The first release of the Fitness and Nutrition Buddy was on February 24. At this point the system had basic functionality - it would present the user with all the options that would be available upon the full completion of the application. The macro tracking option and the goal set/change option were fully functional. The meal plan, nutrition tracking, restaurant search, and map options were partially functional, with the GUI being fully implemented and some basic functionality being implemented. This release also contained the creation of databases that would be necessary for the full functionality of the application, such as a database of local restaurants, a database of nutritional information, and a database of sample users.

## **2 Second Release**

The second release of the Fitness and Nutrition Buddy was on March 31. At this point, the system had full functionality - it would lead the user through signing into their account and present them with the various functionalities the app could support. The macros tracking, goal set/change, and nutrition tracking options were all fully integrated to work together in painting a full picture of the user's nutritional intake. The restaurant search and meal plan options were made fully-functional and intuitive for the users. Altogether, the various options would allow the user to have full use of the app as was intended. In this release, a new database was created to hold information about various meals to be presented to the user and the previously created databases were updated. The frontend and backend were also fully integrated and every new functionality was integrated into the application as developed. Overall, the application was working as intended at this point.

## **3 Comparison with Original Project Design Document**

The application developed contains the options listed in the full project description. As noted in Figure 2 of the project description document, the nutrition tracking, meal plan, and restaurant search options are all present in the prototype. Section 4c of the project description enumerates the requirements for these options, which are met by the prototype. For example, the nutrition tracking option should allow the user to be "able to track their calories and nutrition data on a day-to-day basis" - this is present in the prototype. The meal plan option should pull data from nearby restaurants and create a plan that suits the user's nutritional needs - this is present in the prototype. The restaurant search option should allow "the user to find specific restaurants in their general vicinity" and to filter the results based on their preferences - this is present in the prototype. In this manner, we can conclude that the prototype produced meets the basic requirements set out by the previous group.

## **III Testing**

### **1 Items to be Tested**

1. ID#001 - Ability to track daily calorie and macronutrient intake
2. ID#002 - Ability to generate meal plans based on calorie goal and dietary restrictions
3. ID#003 - Ability to search for restaurants in the given location
4. ID#004 - Ability to display restaurants on map based on given location
5. ID#005 - Ability to change goals
6. ID#006 - Ability to sign-in/sign-up

### **2 Test Specifications**

#### **ID#001 - Nutrition Tracking Test**

**Description:** Tests whether users can track their calorie and nutrition data on a daily basis

**Items covered by this test:**

- Ability to track daily calorie intake
- Ability to track daily macronutrients
- Ability to reset calories
- Ability to reset macronutrients

**Requirements addressed by this test:** NA

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Navigate to the Nutrition Tracking screen
3. Add calories or macronutrients are required
4. Go back to the home screen and navigate to the Nutrition Tracking screen again
5. Check if it still displays data
6. Add more calories or macronutrients
7. Check if added correctly
8. Click on reset button
9. Check if reset all fields to 0
10. Continue adding calories or macronutrients

**Input Specification:** Integer values

**Output Specifications:** Daily calorie and macronutrient totals displayed on the screen.

**Pass/Fail Criteria:** Screen is functional and tracks daily calories and macronutrients accurately

**ID#002 - Meal Generation Test**

**Description:** Tests whether users can generate custom meal plans based on their calorie goals

**Items covered by this test:**

- Ability to generate custom meal plan with 3 meals

- Ability to filter meals based on dietary restrictions
- Ability to regenerate meal plans
- Ability to generate meal plan based on calorie goals

**Requirements addressed by this test:** ID#003, ID#004

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Navigate to the Change Goal screen and enter nutritional goals
3. Navigate to Meal Plan Screen and generate meals
4. Filter based on dietary restrictions and generate meals
5. Regenerate meals
6. Change calorie goal again and generate meals

**Input Specification:** Integer value

**Output Specifications:** Custom meal plan (breakfast, lunch, and dinner) that follow the user's calorie goal

**Pass/Fail Criteria:** Screen is functional and displays custom meal plans (breakfast, lunch, and dinner) based on user calorie goal and dietary restrictions

### **ID#003 - Restaurant Search Test**

**Description:** Tests whether users can search for restaurants in particular locations

**Items covered by this test:**

- Ability to find restaurants based on user's location

**Requirements addressed by this test:** ID#002

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Navigate to the Restaurant Search screen and enter/re-enter location

**Input Specification:** String value

**Output Specifications:** List of restaurants in the search area

**Pass/Fail Criteria:** Screen is functional and displays a list of restaurants based on user's desired location

#### **ID#004 - Restaurant Map Test**

**Description:** Tests whether restaurants are displayed on map based on user's location

**Items covered by this test:**

- Ability to find and display restaurants on map based on user location and on the screen

**Requirements addressed by this test:** ID#002

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Navigate to the Map screen and enter/re-enter location

**Input Specification:** String value

**Output Specifications:** List of restaurants in the search area displayed on the map and screen

**Pass/Fail Criteria:** Screen is functional and displays restaurants on a map and a list on the screen based on the user's desired location

#### **ID#005 - Change Goals Test**

**Description:** Tests whether user changes are reflected in other features

**Items covered by this test:**

- Ability to change goals
- Ability for the app to function with new goals

**Requirements addressed by this test:** ID#001

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Navigate to the Change Goals screen and change goals
3. Choose a random feature from the app and test

**Input Specification:** Integer value

**Output Specifications:** NA

**Pass/Fail Criteria:** Accurately record new changes which are reflected in other features such as meal plan generation

**ID#006 - Sign-In/Sign-Up Test**

**Description:** Tests users are able to sign-in or sign-up

**Items covered by this test:**

- Ability to sign-in and sign-up

**Requirements addressed by this test:** NA

**Environmental needs:** Mobile device running on iOS/Android

**Intercase Dependencies:** NA

**Test Procedures:**

1. Launch the Fitness and Nutrition Buddy application
2. Sign-in or sign-up as desired

**Input Specification:** String values

**Output Specifications:** Successfully allows the user to sign-in or sign-up and use the application

**Pass/Fail Criteria:** Allow the user to sign-in if already has an account or sign-up if new user and use the application

### 3 Test Results

**ID#001 - Nutrition Tracking Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Shreya Boyapati

**Expected Results:** Screen is functional and tracks daily calories and macronutrients accurately.

**Actual Results:** Screen is functional and tracks daily calories and macronutrients accurately.

**Test Status:** Pass

#### **ID#002 - Meal Generation Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Dat Huynh

**Expected Results:** Screen is functional and displays custom meal plans (breakfast, lunch, and dinner) based on user calorie goal and dietary restrictions

**Actual Results:** Screen is functional and displays custom meal plans (breakfast, lunch, and dinner) based on user calorie goal and dietary restrictions

**Test Status:** Pass

#### **ID#003 - Restaurant Search Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Sarthak Patipati

**Expected Results:** Screen is functional and displays a list of restaurants based on user's desired location

**Actual Results:** Screen is functional and displays a list of restaurants based on user's desired location

**Test Status:** Pass

#### **ID#004 - Restaurant Map Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Ayesha Quadri Syeda

**Expected Results:** Screen is functional and displays restaurants on a map and a list on the screen based on the user's desired location

**Actual Results:** Screen is functional and displays restaurants on a map and a list on the screen based on the user's desired location

**Test Status:** Pass

#### **ID#005 - Change Goals Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Dat Huynh

**Expected Results:** Accurately records new changes which are reflected in other features such as meal plan generation

**Actual Results:** Accurately records new changes which are reflected in other features such as meal plan generation

**Test Status:** Pass

#### **ID#006 - Sign-In/Sign-Up Test**

**Date(s) of Execution:** April 9, 2023

**Staff conducting tests:** Sarthak Patipati

**Expected Results:** Allows the user to sign-in if already has an account or sign-up if new user and use the application

**Actual Results:** Allows the user to sign-in if already has an account or sign-up if new user and use the application

**Test Status:** Pass

## **4 Regression Testing**

Does not apply.

## **IV Inspection**

Our group saw equal contributions from all members. All worked on significant pieces of code, with many pieces being collaboratively worked on or worked on by multiple members sequentially.

### **1 Items to be Inspected**

Item 1: Meal plan screen code - Meal.java, MealsInfo.java, mealPlanScreen() method in FitnessStart.java

Item 2: Restaurant search screen code - RestaurantInfo.java, restaurantSearch() and restaurantScreen() methods in FitnessStart.java

Item 3: Signup/signin screen code - UserInfo.java, signInButton(), signUpButton() and signUpScreen() methods in FitnessStart.java

Item 4: welcome and options screens code - start(), exit(), startButton() and optionScreen() methods in FitnessStart.java

## 2 Inspection Procedures

The primary inspection procedure was thorough testing of the code by ensuring that it met all the requirements for its purported functionality. Testers would use the app as if they were a new user and see if the code being tested could suit all potential user needs. The list of requirements was developed based on the project description document and kept as reference as the code was being tested. In this way, we were able to ensure that the code does what it is supposed to do. The team also discussed the results of testing in order to make sure that everyone was aware of any potential issues and/or feedback from the results of the testing. This occurred in the form of a team meeting following the second release of the application. Most of the work was done outside of group meetings, in a rolling manner throughout the last week of the second release. The final discussion was held as one large, in-person group meeting.

## 3 Inspection Results

All inspections were performed in the last week of March (3/27-3/31) by all three team members not responsible for that specific item of code. These inspections generally yielded positive results as the code performed as expected and met the requirements set for it. There were some issues that were uncovered by this testing, which shall be resolved in the following weeks. Inspection of item 1 found that the meals database did not have enough meals and should be expanded in volume. Inspection of item 2 found no issues - all requirements were met and edge cases accounted for. Inspection of item 3 found a minor issue in adding new users, which was immediately fixed by the inspector. Inspection of item 4 found no issues - all requirements were met and edge cases were accounted for.

## V Recommendations and Conclusions

All tests for the application eventually passed and the end product resulted as was expected. . Though there seems to be a problem with the amount of time it takes for the back button to process the action event and for the restaurant search screen to be generated. It is recommended that there be more testing and fixtures with the data structures being used in order to reduce the time cost of the project. In conclusion, since all parts of the project acted as expected it is concluded that the project was a success

## **VI Project Issues**

### **1 Open Issues**

The current situation that can cause issues to our product is how fast restaurants and meals change. At the moment, we don't have the optimal solution for restaurants to update their meals on our database. The more restaurants we have in the database, the more inaccuracy it will create. Furthermore, the type of meals change constantly. For example, if the application informs the user that a certain meal is keto, but the restaurant changes it to non-keto without updating the database. Therefore, the application will give false information, which can be a huge problem. We need to find a way for our application to update changes from the restaurants and add more if needed.

### **2 Waiting Room**

The possible features that can be added in the future are connecting to social media, competition and rewards, linked with a map application, and displaying recipes for certain meals. By connecting with users' social media, they can share their progress with friends. They can also see other people's achievement as a motivation. The application can have a competition section with friends and the rewards are restaurants' gift cards. Furthermore, linking with a map application can allow the product to receive updated information. With permission from restaurants, the application can display recipes for a certain food, which users can recreate at home. These features are designed to potentially provide a better service for the users.

### **3 Ideas for Solutions**

One of the possible solutions for the restaurants and meals update issue is creating another feature that allows only restaurants to modify their menus. However, this solution requires restaurant owners' cooperation. The product can provide more advertisement for restaurants as a reward if their menu is up to date. Another solution for this problem is to allow users to report menu errors. The first user to report any mistakes can receive a gift card from a certain restaurant. These solutions can help increase the accuracy of the output from the product.

## **4 Project Retrospective**

Worked well:

- The good collaboration skills between members of our team allow us to have good communication.
- Meeting every week to discuss completed and uncompleted tasks allows us to stay on track with the progress.
- Breaking problems into small tasks and creating our own deadlines for each member's task help us focus on one part at a time.

- Brainstorming ideas that all members agree on, which creates satisfaction for everyone.

Didn't work well:

- By dividing the work among members, each of us has to work by ourselves on the task without any assistance from other members. In a way, this defeats the purpose of collaboration.
- Consistency in coding style for our tasks is also a problem. Since we are working on our own part, it is hard to keep track of the coding style that other members use. It can cause inconsistencies in our codes and misunderstanding when we review other's code.

## VII Glossary

**Restrictions:** The limitation set by the user about their macronutrient intake.

**Track:** Saving specific past information about the user relative to their nutritional goals.

**Prototype:** A version of the original project that contains most of the functionality described but not all. It is a simpler version that can be easily modified and tested.

**Dietary Restriction:** Certain types of food chosen by the user to achieve their nutritional goals.

**Suggestions:** Recommendations from the application that don't need to be followed but would greatly benefit the user in achieving their nutritional goals.

**API:** Mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

**Functional Requirements:** Requirement for the application referring to the functionality.

**Data Requirements:** The data requirements to be met by the application.

**Performance and Dependability Requirements:** The requirements to be met by the application in regards to the performance, such as how long it takes for certain functions of the application to run, and dependability of the data, such as program failure.

**Maintainability and Supportability Requirements:** Refers to the bug fixes and updates the application needs in order to function properly.

**Security:** Refers to the security of the data taken from the user. Also refers to the application data security.

**Repositories:** A centralized digital storage that developers use to make and manage changes to an application's source code.

**Usability:** Refers to the measure of how easy it is to use the application for the first time and whether there is a learning curve.

**Humanity:** Refers to how accessible the application is to people with disabilities.

**Look and Feel:** Refers to how the finished application's look, color palettes. Also does the application align with the modern app styles.

**Operational:** Refers to which technological platform the application is best used on.

**Environmental and Cultural Requirements:** Refers to the physical environment the application is used in and any culture references it either uses or focuses on.

**GUI:** Graphical user interface; A graphics-based operating system interface that uses icons, menus and a mouse (to click on the icon or pull down the menus) to manage interaction with the system.

## VIII References / Bibliography

- [2] “JavaFx Overview.” *I JavaFX Overview (Release 8)*, [docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm](https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm).
- [8] Gartner\_Inc. (n.d.). *Definition of GUI (graphical user interface) - gartner information technology glossary*. Gartner. Retrieved April 22, 2023.
- [5] Hansana, A., Elliott, K., Weaver, Vincent., Crowley, Ryan. (2021). Fitness and Nutrition Buddy. University of Illinois at Chicago.
- [3] Mixon, E. (2023, February 23). *What is Android OS?: Definition from TechTarget*. Mobile Computing. Retrieved April 22, 2023.
- [4] Posey, B. (2023, February 23). *What is Apple Ios?: Definition from TechTarget*. Mobile Computing. Retrieved April 22, 2023.
- [1] Ted Kallmyer (Certified Macro. “The Formula to Calculate Your Macros and Ree Accurately.” *Healthy Eater*, 11 Apr. 2023, [healthyeater.com/how-to-calculate-your-macros](https://www.healthyeater.com/how-to-calculate-your-macros).

[6] The University. (1978). What Is An API (Application Programming Interface)? Amazon. Retrieved April 22, 2023.

[7] The University. (1978). *What is Repo?* Amazon. Retrieved April 22, 2023.

## IX Index

Project Description	5	Project Issues	17
Project Deliverables	8	Glossary	18
Testing	9	References / Bibliography	18
Inspection	15	Index	18
Recommendations and Conclusions	16		