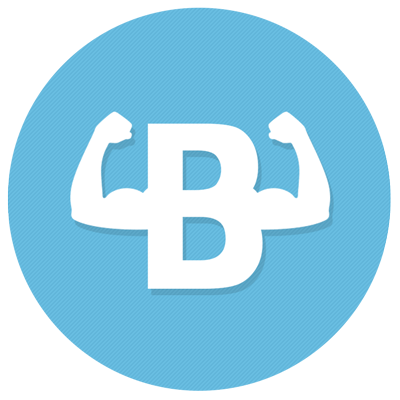
**CS4704 Software Engineering Capstone**

Spring 2016



**BetterU**

“Be you, but better”

|  |  |  |
| --- | --- | --- |
| Julia M. Binger  Mykhaylo D. Bulgakov  Allan F. Chua  Duke J. Forsyth  Filip D. Gouglev | Amanda J. Kahn  Mukund A. Katti  Travis M. Lu  Corey S. McQuay  Ojas D. Mhetar | Ryan T. Munz  Benjamin F. Robohn  Jared J. Schwalbe  Timothy T. Street  Hung T. Vu |

Department of Computer Science

Virginia Tech

Blacksburg, VA 24061

Date: May 1, 2016

Team Number: 1

Submitted to: Prof. Osman Balci

EXECUTIVE SUMMARY

TBD

<< Provide a structured summary of the content of the document, by emphasizing your important contributions to catch the attention of a very busy executive

Provide a numbered list of cloud software features, mobile software features, and other technology features you implemented in your cloud/mobile software-based solution.

>>

**TABLE OF CONTENTS**

EXECUTIVE SUMMARY ii

1. SOFTWARE LIFE CYCLE 1

2. PROBLEM SPECIFICATION 2

3. REQUIREMENTS SPECIFICATION 2

4. ARCHITECTURE SPECIFICATION 2

5. DESIGN SPECIFICATION 2

6. DELIVERED SOFTWARE FUNCTIONALITY 3

7. HOW TO BUILD AND RUN THE DELIVERED SOFTWARE 3

8. CONCLUSIONS 3

REFERENCES 3

TEAM MEMBER CONTRIBUTIONS 4

PROJECT GRADING 5

# SOFTWARE LIFE CYCLE

A **good software engineer** develops software by following the software life cycle shown below.



A **programmer (hacker or ad-hoc developer)** develops software by looking at the problem and directly coding in an IDE. This approach is known as the **Build-and-Fix Approach**, which must never be used!

# PROBLEM SPECIFICATION

TBD

# REQUIREMENTS SPECIFICATION

The next step after creating a problem specification is to do the requirements engineering for BetterU. The team completed requirements engineering by examining other nutritional and fitness applications such as My Fitness Pal and asked some other students what they would like to see in a nutritional and fitness application. Once the team gathered all the raw information, the information was then broken down and user stories were created. From the user stories, the team was able to create well-defined requirements that BetterU would have to meet. The user stories and their corresponding requirements are listed below. Some requirements are repeated across different user stories.

## Enter/Change Goals

The user can input fitness goals to personalize their experience within the app. This will allow the user to stay focused during their fitness journey and motivate them to continue working out in order to achieve a better them.

The user will be able to change the goals they set. The reason for doing this could be a user surpassing their original goal, or a user misjudging how much time they will have to work out during the average week. Changing the goals will help users stay on track with realistic improvements.

## Set reminders for workouts, food, goals

The user should be able to set personal goals since each person's goals will vary. Some users will be want to lose weight while other will want to gain muscle effectively gaining weight. Later users will be able to get points for each goal that they make that scale in amount based on the type of goal.

## Be reminded of goals (via push notifications)

Since the user will be able to set the goals for their account they must be able to receive notifications for the goals. They could also set an alert should a long achieving goal hits a certain threshold you would be reminded. An example would be if a user had a achievement of losing 50 pounds and they lost 25 they should get a reminder.

## Get points for meeting goals

Users will receive points for completing goals. The larger the goal, the more points rewarded. Then they can compete with their friends to see who can score more points each week/month. We’re confident that this gamification will increase motivation for maintaining and completing goals.

## View Recommended Foods

As a user is about to decide what they should eat today, BetterU will be able to recommend a food based on the user’s goals and current nutritional and caloric intake of the day.

Recommended foods will also consider the current meal of the day, and if requested, can offer popular foods for that meal.

## Select Recommended Foods

Once the users selects a food recommended to them, the app will take the food entry in our database and retrieve all nutritional information associated with it. The information will then be counted into the user’s daily intake.

## Track calories, grams of fat, sodium etc. daily

After the user inputs the data of the food they have eaten, the app will be able to aggregate key metrics such as macronutrients (protein, carbohydrates, and fats), as well as sodium, vitamins, etc. These will be broken down on a daily chart so the user has a summary of the nutrition they eat each day.

## Enter a specific dish from a restaurant

A user can browse the restaurant’s menu in the app and look at the nutritional information of each dish. This decreases much of the hassle presented by entering nutritional information, and helps the user make healthy choices.

## Enter Food Consumed To Log Calories

There is a user that wants to enter food items into the application to log calories consumed. The server-side application will search the USDA’s National Nutrient Database based on user’s food search. Once the food item is found and selected, the app will update the number of calories consumed on that day for that user. The new consumption data will also be appended to the Progression Database.

## Enter food item and receive nutritional info

The user should be able to search for food items on our application and then receive nutritional facts such calories, ingredients, sugars, proteins, etc. If this information is quick and easy to access then the user will be more inclined to research the facts about what they eat before they put it in their body.

## User Account Creation

There is a Virginia Tech student who wants to lose weight, but he is having a hard time keeping track of his progress. The student then decides to download BetterU from the Apple Store, hoping that he can finally organize his workout routines. He opens the application and decides to create an account. BetterU then prompts the student to enter his username, password, height, weight, age, and weight goals.

## User Account Deactivation

There is a Virginia Tech student who has lost 100 pounds and became a better version of himself. He is the embodiment of “Be U, but Better” so he has decided to delete his account. All he has to do is select ‘Delete Account’ in the menu and it will be deactivated.

## Edit Account Settings and Preferences

Every user is unique and will want the app the function differently for him/her. So the user will be able to change how frequent the push notifications appear and what he/she is notified about (Updates about friends, Goal Reminders, Enter Information Reminders), what the app emails the user about, reset user data, change profile information (Height, Weight Age, Sex, and Weight Goals), Set regiment type (Lose Weight, Maintain Weight, Build Muscle), and Connect/Disconnect from Facebook.

## Change Password

There is a user that has decided to change his/her password either because they want a new password or they have forgot their password and need to reset the password. The user will have to enter their current password once and the type in the new password twice to confirm it was typed in correctly. If user forgot their password, then they will hit the reset password button and an email will be sent with a temporary password to the email in the user’s account information. The user will then proceed as before, but use the temporary password instead.

## Storing the User’s progression throughout a period of time

The server side of the application will have a progression database which will store the User’s progression of calories consumed when logging in. This is needed so that that application can show charts and graphs of the user’s progression by pulling the data off of this database, analyze this data to see if the user is meeting their goals, and so on.

## Construct Graphs from User Account Information

The server-side application will retrieve information regarding calories consumed, minutes active, etc. from user records in the Progression Database. From this data, the application will generate graphs giving users a visual representation of their progress. Data will be displayed via line/bar graphs, where previous dates are along the x-axis.

## Log in/out

The user logs into their account by supplying a username and password that is associated with their account preferences and data stored in our users database. The user will be prompted to log in when they first open the application. They can either log out manually at any time, but will remain logged in if they close the app without logging out. If the user wishes, they may toggle their settings to log out automatically every time the application is fully closed.

## Review Graphs

The user logs their calories in/out and active minutes over time and is able to go to the data section of the app to look at how these have changed over time. The graphs are generated based on the user data stored in their associated account. The user is able to easily interpret and see trends in the data from the home screen of the application. To see how trends in the data, they can adjust the scope.

## Share accomplishments on social media

Imagine you were rewarded for eating healthy, exercising, and meeting your goals. We would like to create an environment where you earn points, trophies, and other bonus features for taking care of yourself. For this reason, a user who embraces this feature may want to share the progress they’ve made with friends and family on Facebook, Twitter, Tumblr, etc. Those who use social media regularly can find this as an incentive.

## Add/Recommend Friends

The Virginia Tech student notices that some of his senior friends have the application downloaded. Thus, as what any normal socially active person would do, he decides to add every one of his friends connected through Facebook. For everyone else, he decided to spam them with recommendations to join BetterU.

## Remove Friends

Because the Virginia Tech student does not want that many people spamming their personal goals and records to him, he decides to remove some of them. By clicking on settings, the student is able to search for specific people to remove from his friends list. He even blocked some of them from trying to add him again.

## Challenge friends across social media

Imagine an environment where users are earning points, trophies for eating and exercising, users can also challenge each other on Facebook, Twitter, Tumblr, etc to who can meet a goal first to earn more points. Everyone can pursue their goals at their own pace so the focus is on who can meet their goal first. Those who want motivation to meet goals can find this feature useful.

## Deleting Social Media

A Virginia Tech student has recently lost 100 pounds, but also put back on 200 pounds. He wants to keep his fitness goals private, so he wants to remove social media from his account. Through the settings menu he can remove social media accounts and keep everything private.

## Enter calories burned

Users will be able to keep track of the calories burned during their workouts. This will allow them to chart their progress. The information will be used with the user’s nutritional intake so that they can manage their fitness goals.

## Enter workout intensity and duration

Users will be able to accurately measure how many calories are burned based on the exercise and the time interval. The more intense the exercise, or the longer you exercise, the more calories are burned.

## View example exercises

The user will be able to browse through a catalog of exercises to gauge the difficulty and purpose of the exercise. The user will also be able to determine how the movement is performed.

## Get recommended exercises

The user will be able to see recommended exercises based on the user’s specific fitness goals. The exercises will vary in difficulty and provide the user with options that they feel most comfortable with.

## Track Running Statistics

Users who run or jog as an exercise will be able to track the time and distance they spend for each session as well as the route they have taken shown on a map.

## Non-Functional Requirements

TBD

<< follow the use cases-based requirements engineering >>

Specify a use case name and list the Functional requirements associated with that use case.

List the Non-Functional Requirements separately.

# ARCHITECTURE SPECIFICATION

TBD

Provide a graphical description of the architecture.

Specify the Java EE technologies used under the Client-Server Architecture together with the Service-Oriented Architecture if calling upon web services provided by other cloud software applications and APIs.

# DESIGN SPECIFICATION

TBD

Provide a graphical description (e.g., **storyboard**, **images, diagrams, drawings**, etc.) of the functionality of your cloud software application’s design.

Include UML diagrams, specifically class diagrams.

# DELIVERED SOFTWARE FUNCTIONALITY

TBD

Describe the functionality of your deployed cloud/mobile software application by using screenshots of user interfaces. Start with the URL for accessing your software. Then, screenshot by screenshot describe how your software is used. The entire software functionality must be described in sufficient detail.

# HOW TO BUILD AND RUN THE DELIVERED SOFTWARE

TBD (This section will contain item 1 below)

1. Develop step-by-step instructions on how to build and run your cloud and mobile applications on Dr. Balci’s iMac computer.
2. Place all your well-documented project files and associated resources on a flash drive.
3. On the day of delivery, come to Dr. Balci’s office with the flash drive and follow the written instructions in this section and deploy your cloud and mobile software applications to run on Dr. Balci’s iMac computer.

If your instructions are incomplete and/or inaccurate and I fail to rebuild and run your application, I cannot grade it.

# CONCLUSIONS

TBD

REFERENCES

TBD

Balci, O. (2016), “CS4984 Cloud Software Development Course Website,” <http://manta.cs.vt.edu/cs4984>

Oracle Corporation (2016), “NetBeans,” <https://netbeans.org/>

<< Include other references used >>

TEAM MEMBER CONTRIBUTIONS

Each team member provides a numbered list of his/her accomplishments in a concise and clear manner below.

|  |
| --- |
| Julia M. Binger |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Mykhaylo D. Bulgakov |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Allan F. Chua |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Duke J. Forsyth |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Filip D. Gouglev |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Amanda J. Kahn |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Mukund A. Katti |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Travis M. Lu |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Corey S. McQuay |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Ojas D. Mhetar |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Ryan T. Munz |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Benjamin F. Robohn |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Jared J. Schwalbe |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Timothy T. Street |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |
| Hung T. Vu |
| 1. Contribution 1 2. Contribution 2 3. Contribution 3   << continue >> |

PROJECT GRADING

* Quality of this Report 20%
* Quality of Code Documentation 10%
* Quality of Technical Work 70%