CS683 Project Assignment

FitnessTrack
Haohan Guo

Yi-Yang Lin

1. Overview

With an increasing number of people struggling with obesity nowaday, people require more and more managing skills and auxiliary tools to help themself dealing with health.

In this application, we decided to bring a useful tool to let users track their daily food absorption and energy consumption. FitnessTrack is a health record listing application, allows users to record food items in each meal and also provides a health analysis function to provide some recommendations to users.

2. Related Work

A product with similar features on the market is a simple version of Google Fit: Activity Tracking, which has two main functions, "Journal" and "Profile" tabs. "Journal", for example, shows and tracks the route of running location information. The "Profile" will be able to update the user's information.

Some apps that are similar in the market -
Omo -
Lose weight and get fit with Omo, a fitness app for weight loss! It's like having a fitness coach, meal planner, calorie counter, and fasting tracker with you at all times. Discover healthy weight loss with Omo.

Strong Workout Tracker Gym Log -
Strong is the simplest and most intuitive workout tracker, designed to help you get better results from your workouts. Whether you want to gain strength or just stay healthy, join over 1.2 million people who have downloaded Strong to stay on track in the gym.

3. Requirement Analysis and Testing

(*This section should clearly describe all features/requirements that you plan to implement or have implemented for your application. You should separate them into three categories: essential, desirable and optional.*

For each requirement/feature, you should provide the following details:

Title	*Login or signup user account or enter as a guest (essential)*
Description	*Users are able to create their own account and store personal information* *Login page with queries: 1. Username 2. Password 3. Remember me 4. Forgot your password 5. Login button*
Mockups	*Checking if one user logins, it will correctly gain the data it suppose to get*
Acceptance tests	*The frontend account should match the same account in backend data*
Test Results	*If the user submitted username and password correspond to the data in the back-end part of the database.*
Status	*Iteration 1: implemented the project detail UI page*

Title (Essential/Desirable/Optional)	*Switching views by tapping the exercise view button, food view button, analysis view button (essential)*
Description	*At the bottom of layout of the app it appears tabs that can switching views to be redirecting to different functions*
Mockups	*The view can transition in correct order by following user's clicks*
Acceptance tests	*User click on each view and the screen should show home, exercise, meal view correspondingly what user clicks*
Test Results	*The clicked view should be the same as shown*

Status	*Layout of each was created*

Title	*Add Meal function (Essential)*
Description	*Allows the user to enter various data about the consumption of each meal, such as calories, protein. The app will count the data and display it.*
Mockups	*Ability to correctly count all data entered by the user.*
Acceptance tests	*The meal should be added after inputting data*
Test Results	*After inputting Meal, UI display the added Meal title* *The data should be found in cloud-based database*
Status	*Iteration 1: implemented the project detail UI page*

Title (Essential/Desirable/Optional)	*Add food in function under adding Meal function (Essential)*
Description	*A subset of adding meal, food data will be stored in cloud database and can be retrieved from API server*
Mockups	*The added data and retrieved data should always consistent and correctly stored*
Acceptance tests	*When click on the button of adding food, the data should be stored correctly*
Test Results	*The data should be found in cloud-based database*
Status	*UI implemented in iteration 1* *Function implemented in iteration 2* *In iteration 2 or 3, cloud-based database such as firebase will be utilized in this project*

Title (Essential/Desirable/Optional)	*Popup meal and add food item in the meal (Essential)*
Description	*Allows users to add their own custom foods and calculate calorie and protein data*
Mockups	*The popup meal sub-view should be correct and show the right data*

Acceptance tests	*When click on the button of adding food, the data should be stored in database correctly*
Test Results	*The data should be found in cloud-based database*
Status	*UI implemented in iteration 1* *Function implemented in iteration 2*

Title	*Add exercise type function (Essential)*
Description	*Enable users to select various exercises and help them calculate the corresponding calorie burn*
Mockups	*Drop down window to enable user to select movement*
Acceptance tests	*The selected drop down exercise should be displayed after selected and also updated to cloud-based database*
Test Results	*The data should be found in cloud-based database*
Status	*UI implemented in iteration 1* *Function implemented in iteration 2*

Title	*Popup exercise type and add exercise duration, calories in the current exercise type (Essential)*
Description	*Allows users to add their own custom exercise and calculate calorie burn*
Mockups	*The popup meal sub-view should be correct and show the right data*
Acceptance tests	*When click on the button of adding exercise, the data should be stored correctly*
Test Results	*The data should be found in cloud-based database*
Status	*UI implemented in iteration 1* *Function implemented in iteration 2*

Title	*Popup meal and add food item in the meal (Essential)*
Description	*User can create one’s own meal and can also add food to the current meal such as in a dinner meal, it contains meat, fruits*
Mockups	*The subset of adding Meal is adding food function*
Acceptance tests	*The added food items should under meal view page*
Test Results	*The data should be found in cloud-based database*

Status	*UI implemented in iteration 1* *Function implemented in iteration 2*

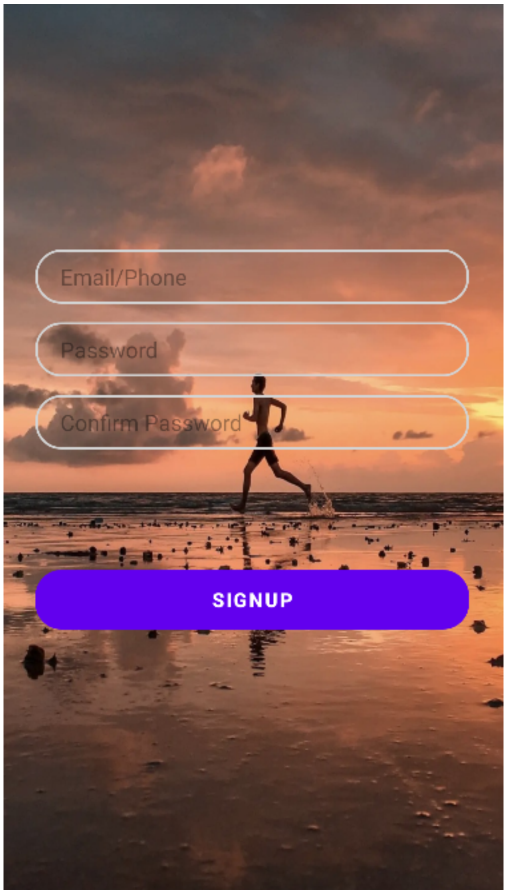
Title	*Generate Fitness Analysis function (Essential)*
Description	*In the analysis view, user can click the button of generating fitness analysis function to get the current fitness results from the cloud base API database server*
Mockups	*The analysis API are connected to this frontend app and when the function get clicked, it the API send back a image object file to show on screen*
Acceptance tests	*The generated image file should be exist and not a crash file*
Test Results	*The file displayed is an image file after built above the generate function button*
Status	*UI implemented in iteration 1* *Function implemented in iteration 2*

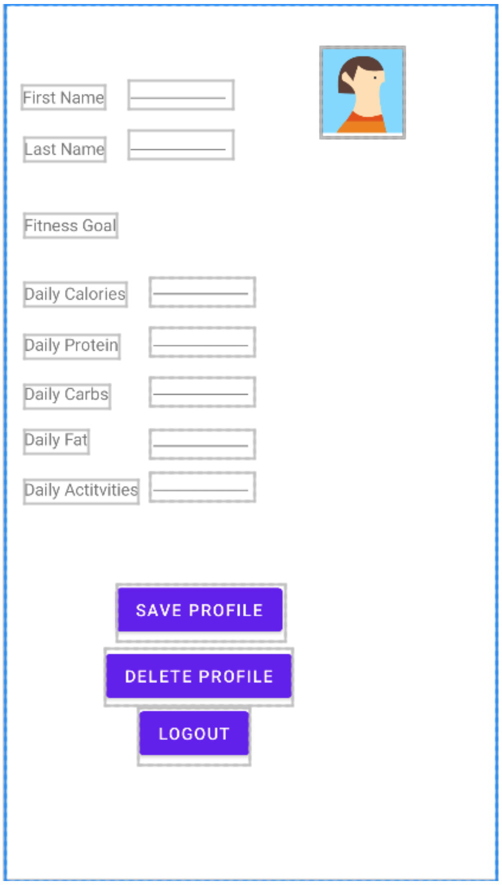
4. Design and Implementation

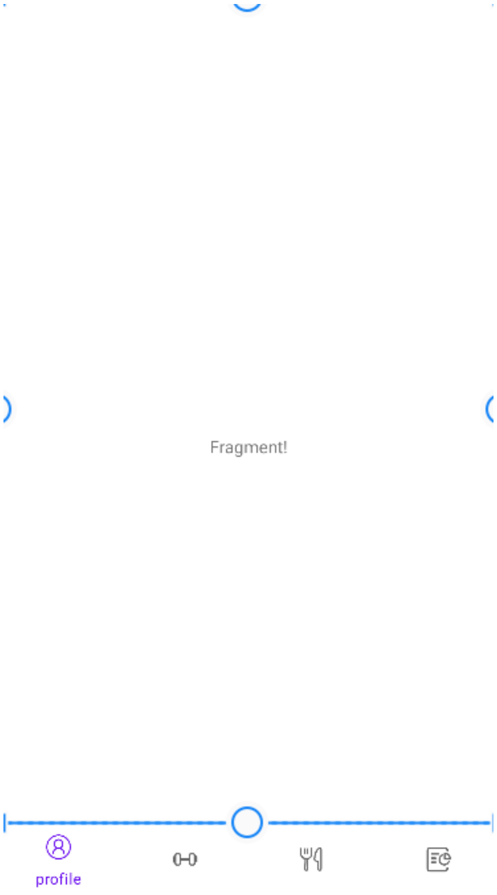
(*This section should describe the basic architecture (e.g. MVC, or MVVM) and your detailed design and implementation. This section may contain the following aspects:*

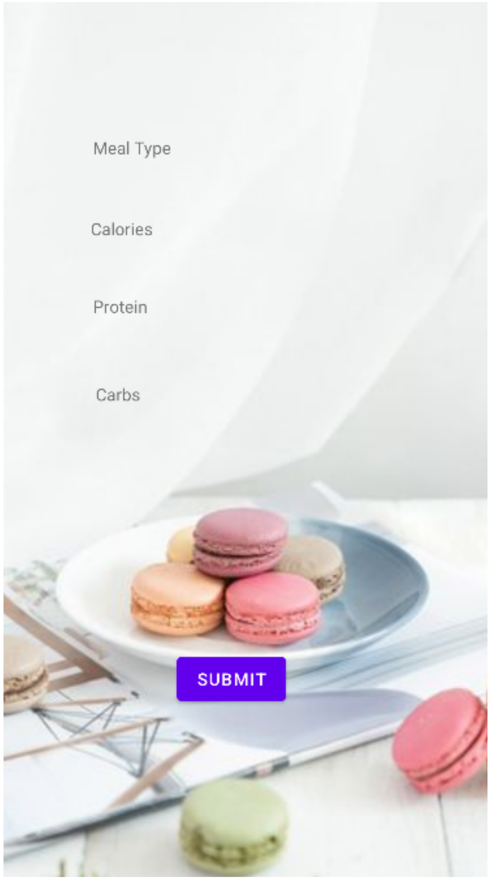
- *Basic architecture*
  The project will use the MVC architecture, which is a method of organizing the code by separating business logic, data, and interface display, gathering business logic into a single component, and improving and personalizing the interface and user interaction without rewriting the business logic. The view layer is separated from the business layer, which allows changes to the view layer code without recompiling the model and controller code. Similarly, changes to an application's business processes or business rules only require changes to the model layer of MVC. Because the model is separated from the controller and view, it is easy to change the data layer and business rules of the application. For this project, the functionality to be implemented is not very complex, so the MVC model makes the business logic all separated into controllers, which is highly modular. When the business logic is changed, it is not necessary to change the view and model, but only the controller, which facilitates the development of our program.

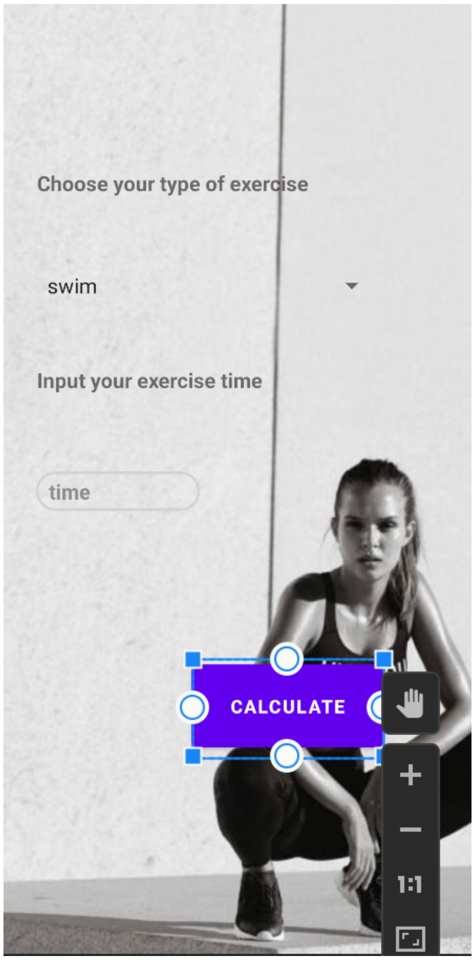
- *UI design and implementation*


Email/Phone
Password
Confirm Password
SIGNUP


First Name
Last Name
Fitness Goal
Daily Calories
Daily Protein
Daily Carbs
Daily Fat
Daily Actitvities
SAVE PROFILE
DELETE PROFILE
LOGOUT


Fragment!
profile


Meal Type
Calories
Protein
Carbs
SUBMIT

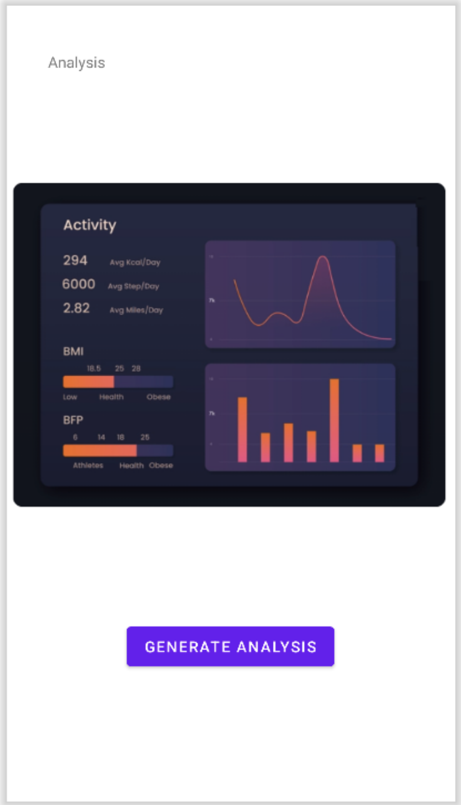


Choose your type of exercise

swim

Input your exercise time

time



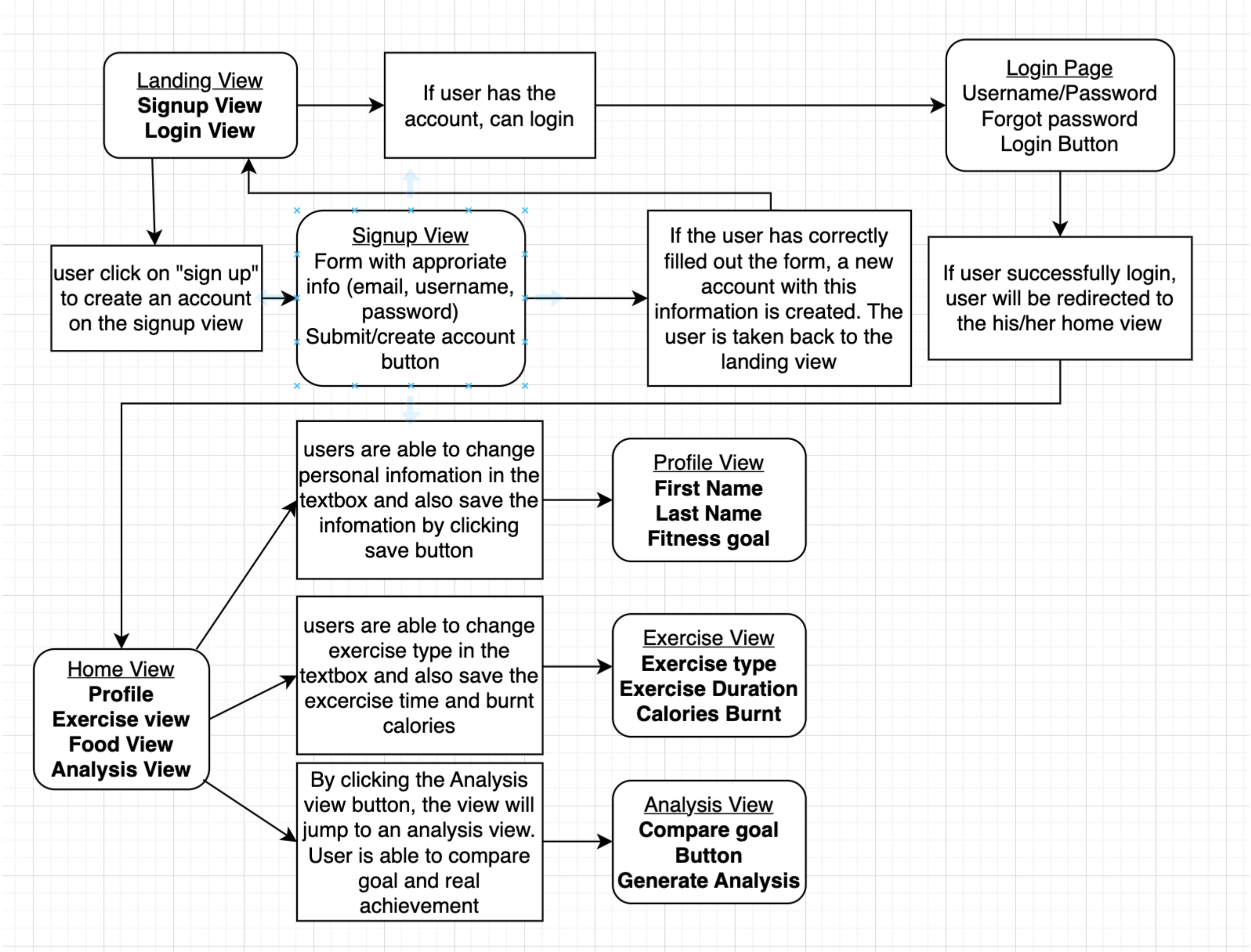
Analysis

OBJOBJOBJOBJOBJOBJOBJOBJ

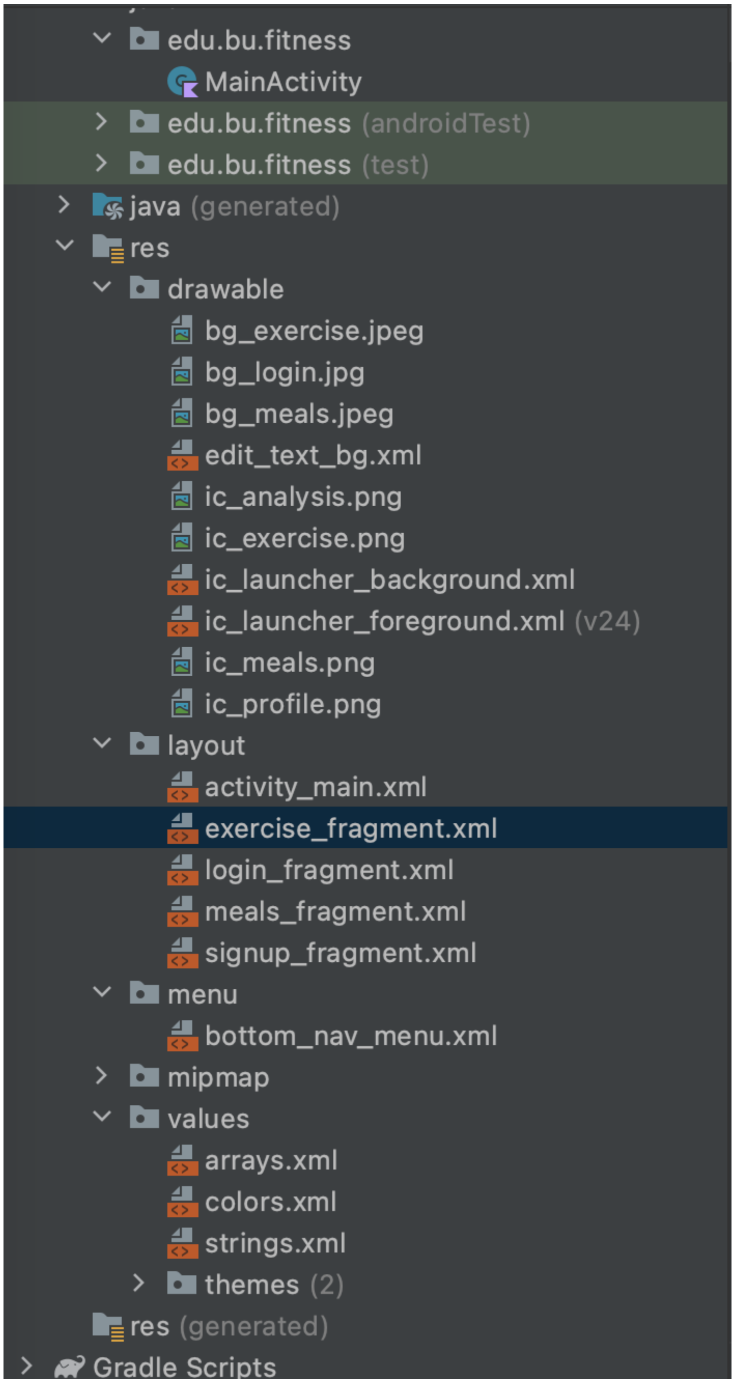
- *Other android features*
    - *Service, sensors, animations, etc*

- *Third party APIs*
- *Data Design and implementation*
    - *Database schema, data storage*
- *Algorithms*
- *…*

You can provide some brief description as well as supporting evidence, such as sample code, log info, or screenshots. You want to specify mapped requirements and files/classes in your project.


Landing View
Signup View
Login View
If user has the account, can login
Login Page
Username/Password
Forgot password
Login Button
user click on "sign up" to create an account on the signup view
Signup View
Form with appropriate info (email, username, password)
Submit/create account button
If the user has correctly filled out the form, a new account with this information is created. The user is taken back to the landing view
If user successfully login, user will be redirected to the his/her home view
users are able to change personal infomation in the textbox and also save the infomation by clicking save button
Profile View
First Name
Last Name
Fitness goal
users are able to change exercise type in the textbox and also save the excercise time and burnt calories
Exercise View
Exercise type
Exercise Duration
Calories Burnt
Home View
Profile
Exercise view
Food View
Analysis View
By clicking the Analysis view button, the view will jump to an analysis view. User is able to compare goal and real achievement
Analysis View
Compare goal
Button
Generate Analysis

5. Project Structure



```
        edu.bu.fitness
            MainActivity
        edu.bu.fitness (androidTest)
        edu.bu.fitness (test)
    java (generated)
    res
        drawable
            bg_exercise.jpeg
            bg_login.jpg
            bg_meals.jpeg
            edit_text_bg.xml
            ic_analysis.png
            ic_exercise.png
            ic_launcher_background.xml
            ic_launcher_foreground.xml (v24)
            ic_meals.png
            ic_profile.png
        layout
            activity_main.xml
            exercise_fragment.xml
            login_fragment.xml
            meals_fragment.xml
            signup_fragment.xml
        menu
            bottom_nav_menu.xml
        mipmap
        values
            arrays.xml
            colors.xml
            strings.xml
            themes (2)
    res (generated)
Gradle Scripts
```

6. Timeline

(*Please provide a summary of the requirements implemented and Android/third party components used in the past and current iterations, and the plan in the future iteration.*

Iteration	Application Requirements (Essential/Desirable/Optional)	Android Components and Features to be used	Member 1 contribution/ planned tasks	Member 2 contribution/ planned tasks

0	Listed app overview and potential features Functions are listed and defined requirementsListed app overview and potential features Functions are listed and defined requirements	none	Communicate and discuss to determine the topic, write iteration 0, and summarize	Communicate and discuss to determine the topic, write iteration 0, and summarize
1	Complete the essential part of the function, including user login, fitness detail section, food detail section, analysis section	We defined activity_main and four main fragment.xml files in which Spinner, Buttom, EditText, TextView are used	1. Completed the registration interface and login interface of the application 2. Completed the fragment.xml file for the two main functions of the project, meal and exercise 3. used navigation to create the bottom navigation bar of the application	1. Created Home view and final generate analysis view 2. Draw the app diagram to show each step with function details 3. Adjusted Overview description 4. Managed source code and pushed to Github
2	Implemented User click event.Improvements to the UI in Iteration0 based on subsequent requirements. Consider extending the application functionality depending on time.			

7. Future Work (Optional)

As time permits, we will expand on the current functionality of the app. For example, we will allow users to create their own fitness schedules and have alarms or pop-up reminders for users when it's time to get on the fitness schedule. We would also like to expand the functionality of the analytics interface to allow users to customize the time period for analysis, etc.

8. Project Demo Links

(*For on campus students, we will have project presentations in class. For online students, you are required to submit a video of your project presentation which includes a demo of your app. You can use Kaltura to make the video and then submit it on blackboard. Please check the following link for the details of using Kaltura to make and submit videos on blackboard. You can also use other video tools and upload your video to youtube if you like:* *https://onlinecampus.bu.edu/bbcswebdav/courses/00cwr_odeelements/metcs/cs_Kaltura.htm*)

9. References

Google Fit: Activity Tracking
https://www.google.com/fit/

Omo: Fitness & Weight Loss
https://play.google.com/store/apps/details?id=calorie.counter.lose.weight.track

Strong Workout Tracker Gym Log
https://apps.apple.com/us/app/strong-workout-tracker-gym-log/id464254577