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.*

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 supposed 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* *Iteration 2: implemented the function*

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	*Users 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* *Iteration 2: implemented the layout*

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	*Iteration 1: UI implemented* *Iteration 2: implemented the layout, already know how to implement*

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, need to store on Firebase*

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 gets clicked, it the API send back an image object file to show on screen*

Acceptance tests	*The generated image file should be existed 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* *Will need to create cloud function then the Google Cloud Platform will send back an analysis image back to front end*

4. System Requirements

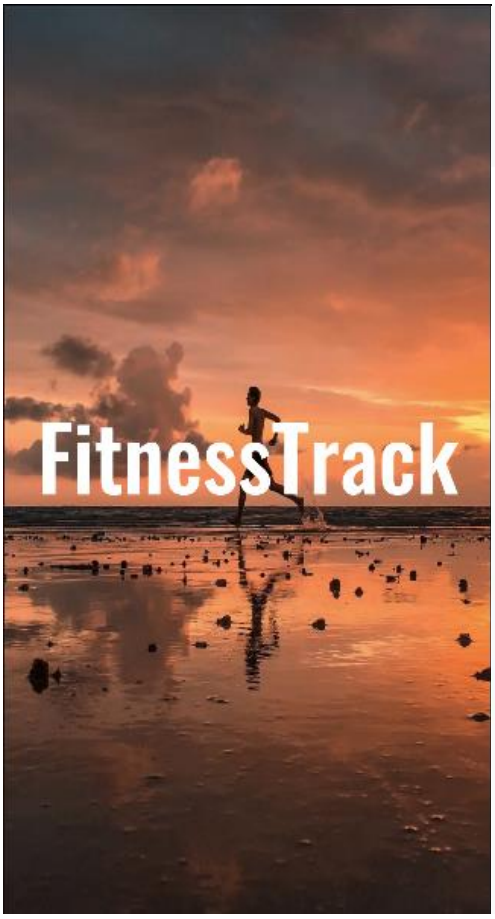
Fitness requires a modern Android Phone (we will be testing primarily on Macbook based Android studio - Intel chip and M1 chip) with internet access.

- Back-end: Google Firebase
    - Database: Cloud Firestore
    - Image Database: Cloud Storage
    - Logistic: Google Cloud Function
    - Authorization: Firebase Authorization
- Front-end: Kotlin and Android's modern toolkit
- Host: Firebase (back-end + front-end)
- Development tools: , Android Studio, Google accounts

5. Design and Implementation

- *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*



FitnessTrack

Let's Get started

Nice Fitness app?

SIGN IN

SIGN UP

SIGN UP

Enter your name, email id and password, to register with us.

Name

Email

Password

SIGN UP

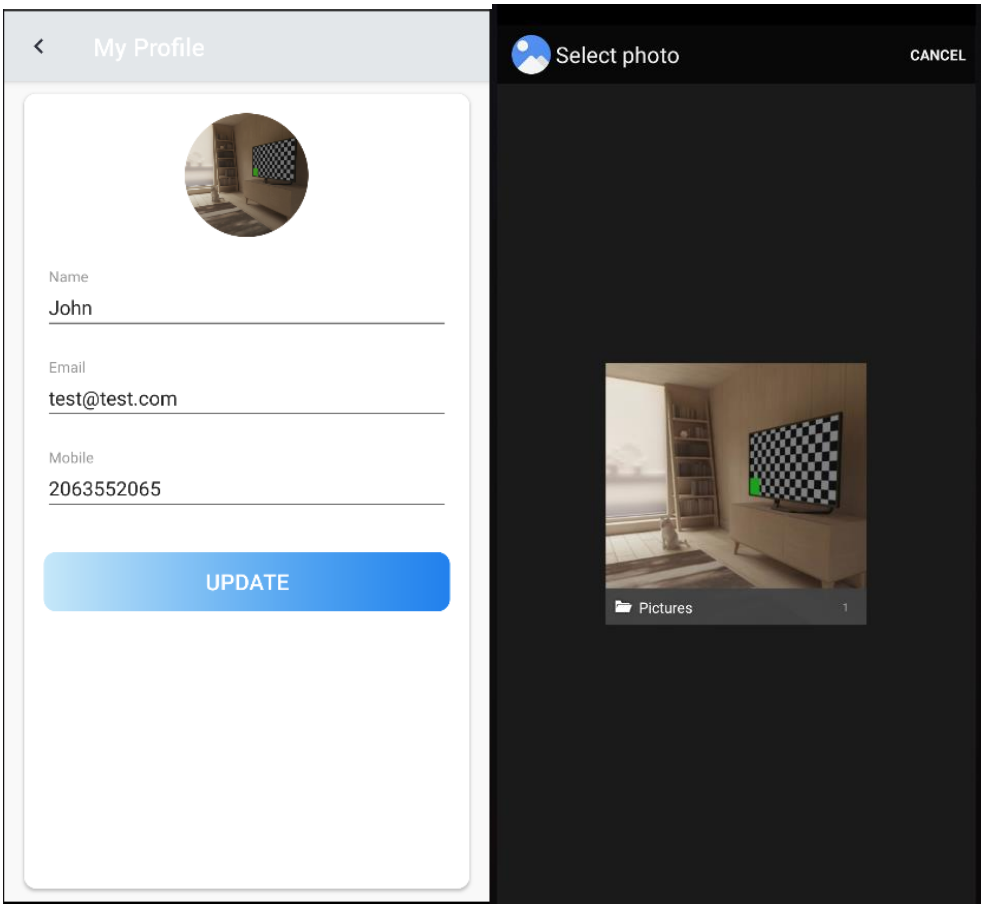
SIGN IN

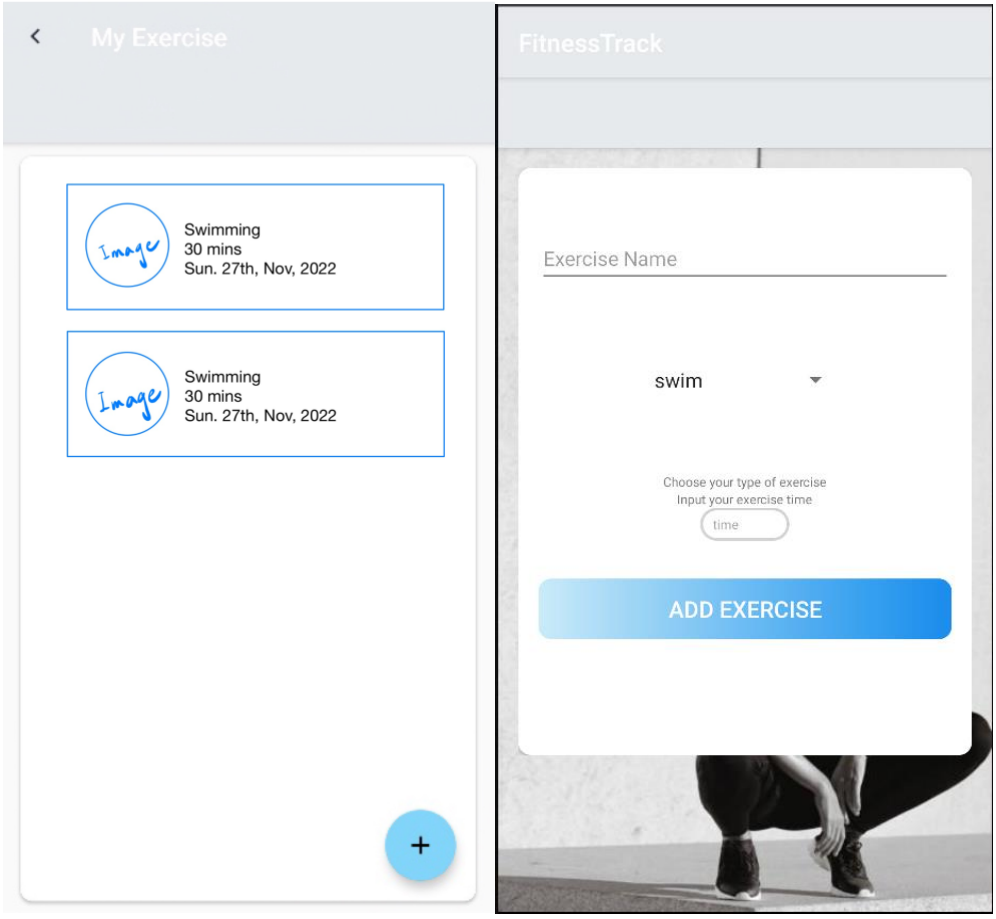
Enter your name, email id and password, to register with us.

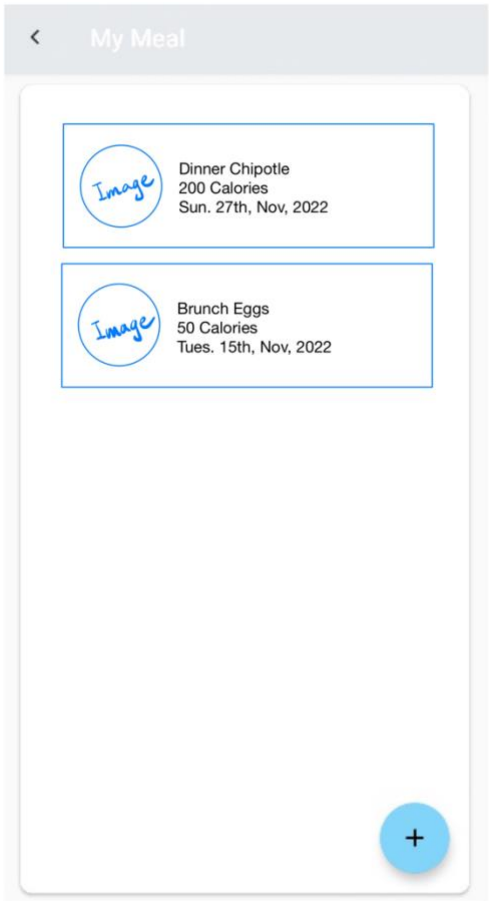
Email

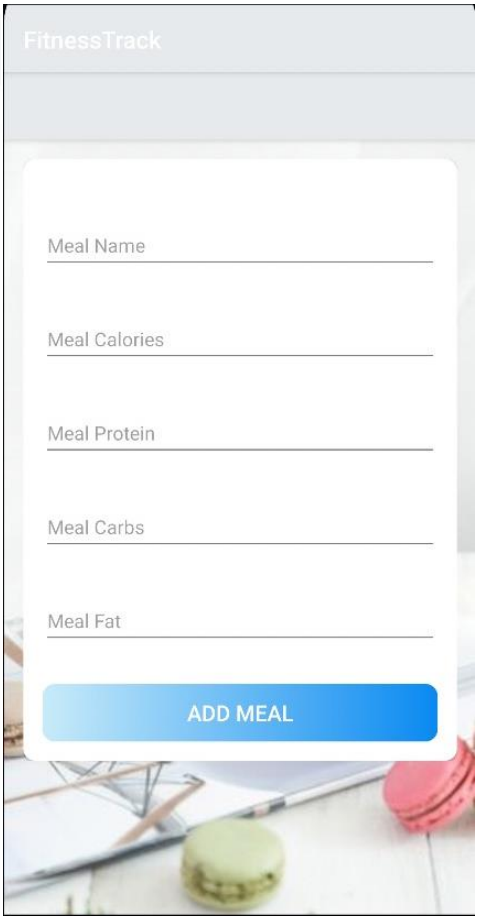
Password

SIGN IN


My Profile
Name
John
Email
test@test.com
Mobile
2063552065
UPDATE
Select photo
CANCEL
Pictures
1

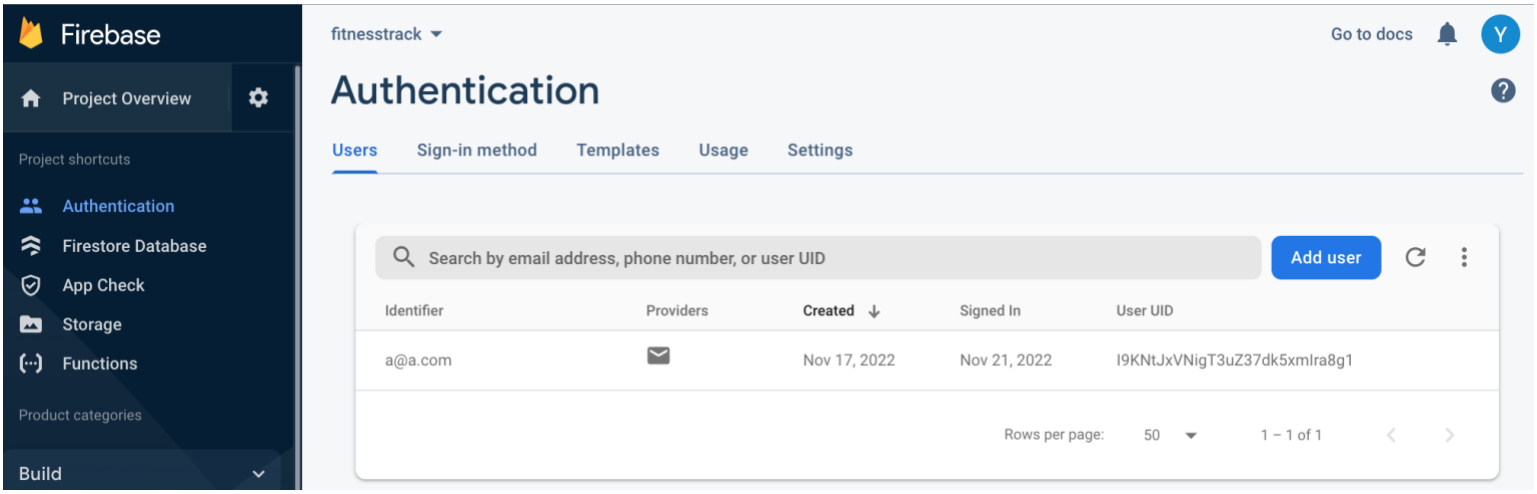

My Exercise
Image
Swimming
30 mins
Sun. 27th, Nov, 2022
Image
Swimming
30 mins
Sun. 27th, Nov, 2022
FitnessTrack
Exercise Name
swim
Choose your type of exercise
Input your exercise time
time
ADD EXERCISE

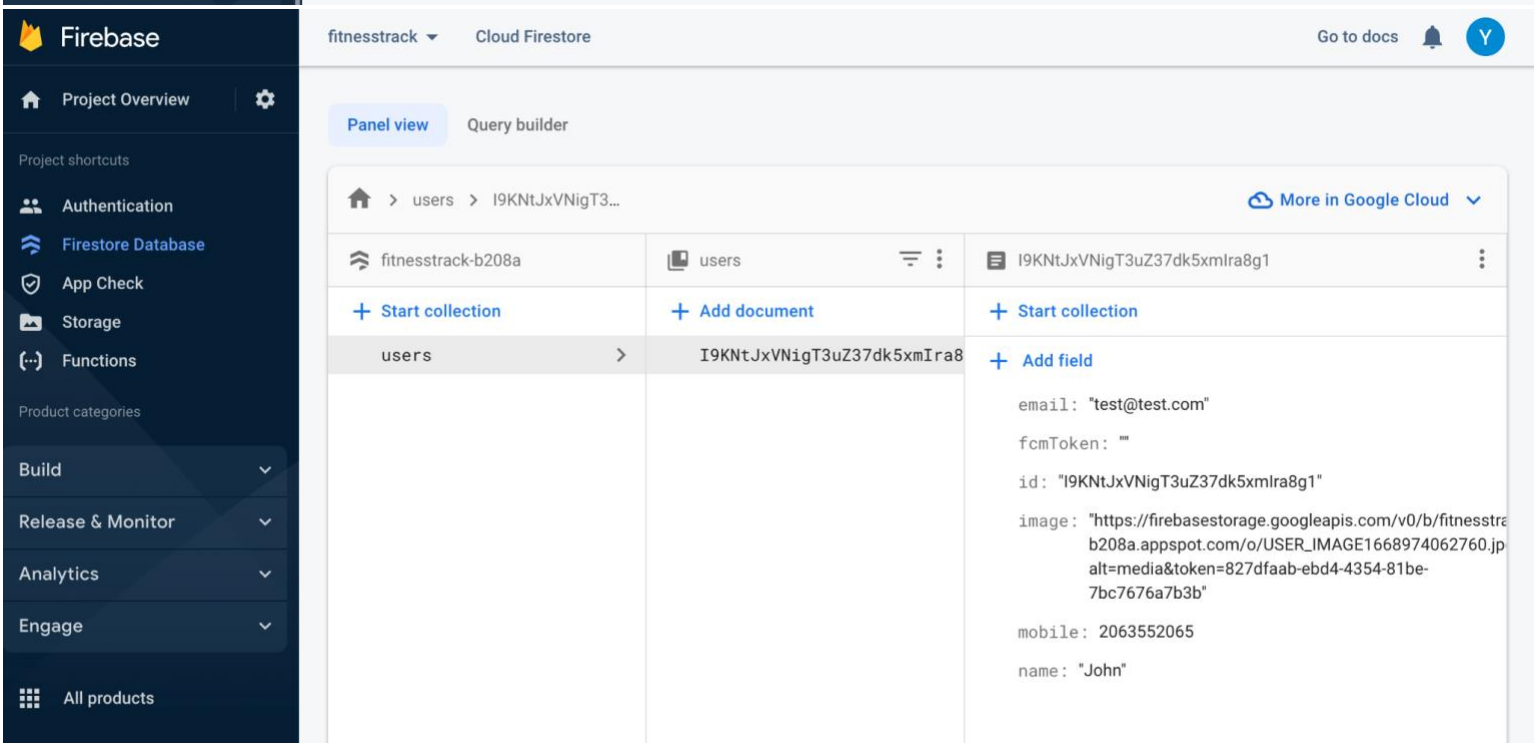

My Meal
Image
Dinner Chipotle
200 Calories
Sun. 27th, Nov, 2022
Image
Brunch Eggs
50 Calories
Tues. 15th, Nov, 2022
+

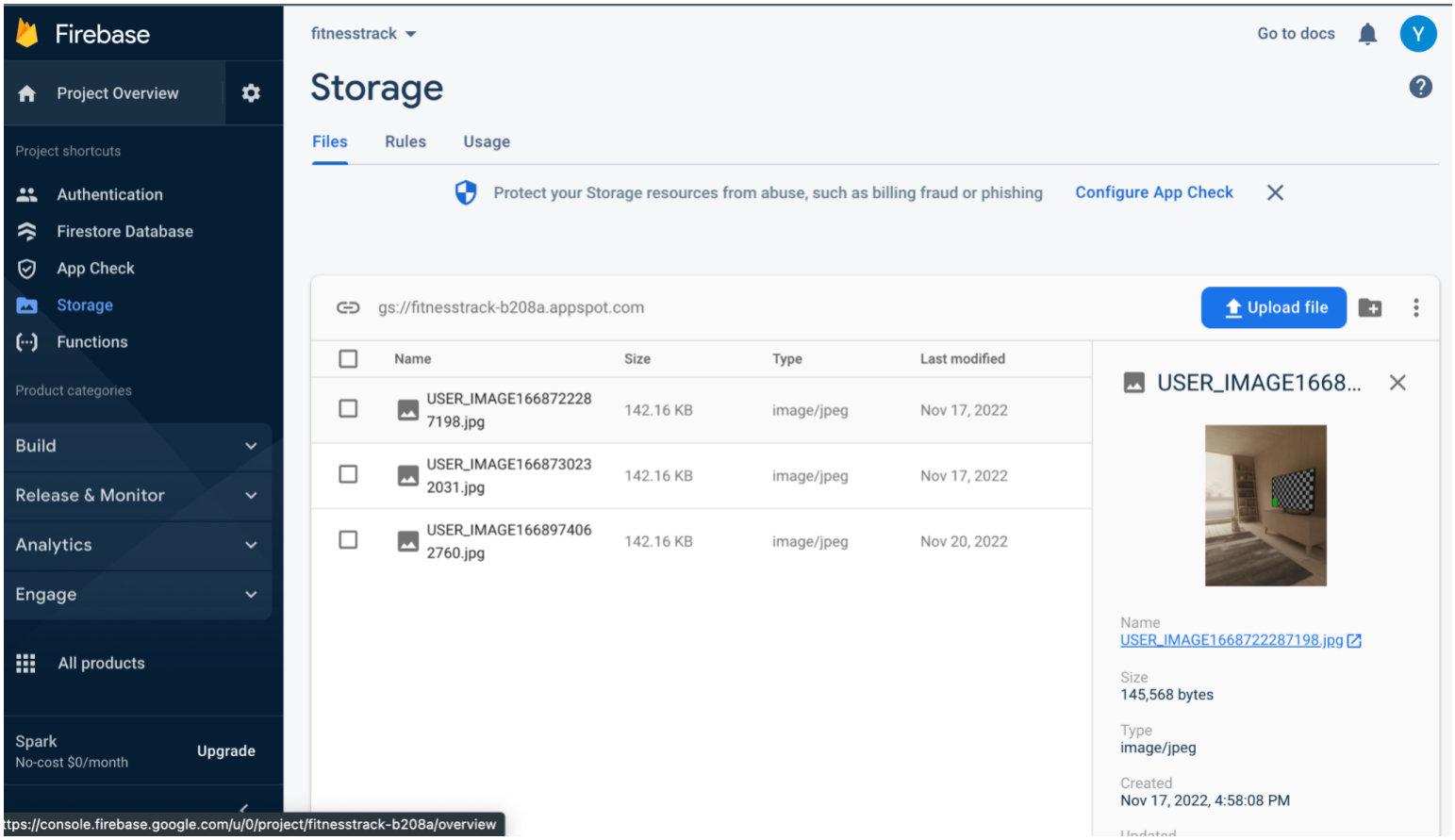

FitnessTrack
Meal Name
Meal Calories
Meal Protein
Meal Carbs
Meal Fat
ADD MEAL


FitnessTrack
Fitness Analysis
Activity
294 Avg Kcal/Day
6000 Avg Step/Day
2.82 Avg Miles/Day
BMI
18.5 25 28
Low Health Obese
BFP
6 14 18 25
Athletes Health Obese

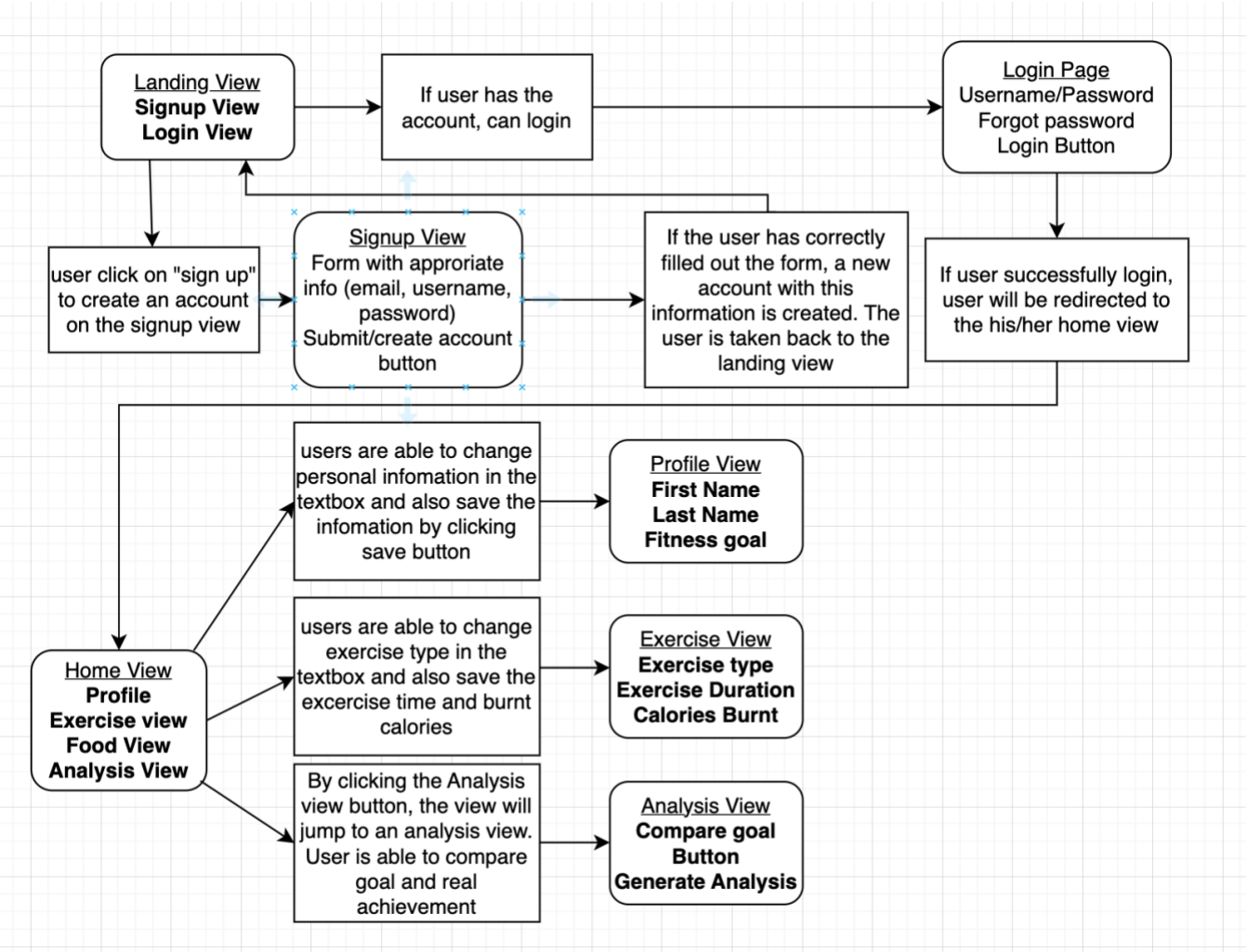
- *Other android features*
    - *Service, sensors, animations, etc (Optional)*
- *Third party APIs*
    - *Cloud function collects data from “My Meal” and “My Exercise” and generates the Analysis information, displaying on the “Analysis” view*
- *Data Design and implementation*
    - *Database schema, data storage*
    - *Firebase NoSQL data structure will be utilized*
    - *Each user has each object of variables*
    - *Whenever a user uploads image file data, it stores the file in “storage” and attached url in metadata, and in the database the collection, the url can be stored and be referenced.*







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.



6. Project Structure

- *AndroidManifest.xml*
  - An overview of the views of entire application, structures the application view flow redirecting to different views
- *CreateExerciseActivity*
  - It is a view after user clicking a button from ExerciseActivity, user can input data and store the value to store on UI and also to the firebase
- *CreateMealActivity*
  - It is a view after user clicking a button from MealActivity, user can input data and store the value to store on UI and also to the Firebase
- *ExerciseActivity*
  - In this view, it will list down the data stored on Firebase that user save from CreateExerciseActivity
- *FirstActivity*
  - The first view that pop up a few seconds before the the login/signup view appear
- *IntroActivity*
  - The view that shown to let user to login or sign up by redirecting in two button “SIGN UP” and “LOGIN”
- *MainActivity*

    - A drawer function-based view that user will swipe the tabs from left side of the app, showing the downloaded User Image from Firebase, Username, and 4 tabs - “My Profile”, “My Exercise”, “My Meal”, “Sign out”
- *MealActivity*
    - In this view, it will list down the data stored on Firebase that user save from CreateExerciseActivity
- *MyProfileActivity*
    - User can update his/her personal details in those text fields, and there is a ‘UPDATE’ button to save it to Firebase database
- *SigninActivity*
    - An activity to retrieve user data and validate the username and password from Firebase’s authentication system
- *SignUpActivity*
    - An activity to retrieve user data and validate the username and password from Firebase’s authentication system to create an account on the Firebase
- *Firestore*
    - Several Firebase function that other view activities will utilize including storing data to Firebase server and retrieving data from Firebase
- *User*
    - A object that bypassing from inputting and uploading to Firebase server, also able to download from NoSQL database from Firebase

7. 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	1. Completed the registration interface and login interface of the application 2. Completed the fragment.xml file for the two main	1. Created Home view and final generate analysis view 2. Draw the app diagram to show each step with function details 3. Adjusted Overview

		used	functions of the project, meal and exercise 3. used navigation to create the bottom navigation bar of the application	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.	We make a user account system in UI and connect to the backend Firebase system. Successfully uploading data to Firestore in NoSQL type data in *MyProfileActivity*	1. Set up the XML UIs and setup the view flow such as after clicking each button and sync with Firebase structure 2. Created Adding Exercise and Meal views. Created profile view for profile list to retrieve data from Firestore	1. Checking the syncing function in the Gradles files and create Firebase account and make cloud features work 2. Created the LOGIN and SIGNUP system on Firebase and able to store validation information on Firebase Authentication and Database Server

8. Future Work (Optional)

Finally, the Firebase Cloud function will be utilized on Google Cloud Platform. After gathering data from the user, the cloud function will be triggered and send an analysis image to the front end.

Fragments will be implemented under *CreateMealActivity*.kt and *CreateExerciseActivity.kt* by using *Recycleview* Adapter features.

As time permits, we will expand on the current functionality of the app. For example, we will allow users to record their own fitness time 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.

9. 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)

10. References

Google Fit: Activity Tracking

Omo: Fitness & Weight Loss

Strong Workout Tracker Gym Log

The Complete Android 12 & Kotlin Development Masterclass

CS 193a: Android App Development Winter 2019

Dartmouth CS 65/165 Smartphone Programming Professor Andrew T. Campbell

Glide

Google Firebase

Learn Android App Development from Zero to Hero - Build 60+ Apps from scratch - Become a real developer

Android Development Fundamental (Chinese)