[**Introduction to Mobile App Dev**](https://douglascollege.blackboard.com/webapps/blackboard/execute/courseMain?course_id=_95741_1)

**3175-070**

**FITGAIN**

**Group Project Final Report**

**Instructor:** Ivan Wong

**Group member (sort by letter) :**

Cheng-Yi Chen 300335751

Dong Nie 300340704

Minh Dat Tu 300341490

SungAh Kim 300341721 (Team Leader)

**Overview about the application**

FITGAIN is a mobile fitness app. It provides users with a variety of exercise plans and meal plans for busy modern people to use in a variety of ways. Users can register on this app by providing their personal information such as height, weight, age, and can keep updating body information. We provide different types of workout videos for users to watch and follow along. Users could also choose meals based on their aim. For the log function, users could track the workouts they have done and the meals they have had to help them examine their goals.

**Features and technical challenges**

**[Login Section]**

* In this feature, we used a simple layout that asks for user input: username and password. The difficulty is connecting between database and activity. After getting user input, we will pass values to an activity where it will connect to the database. In the database, it has a function that will get records based on username and password. Another problem is how can we connect one class to another class because we cannot ask users to input username and password every time they change the page. To solve this problem, we used shareprefernence which helps us to store current userID so even passing data between classes, we can still know who the current user is.
* Logout: in this feature, to do log out, we just redirect user back to the login page
* Profile: because we have userID in sharepreference, the only thing we need to do is get a record by userID and return the value to the view page.
* Update profile: same with the profile, we will use userID to get current user record information and display it on view. After user input, we will use the new information and update it in the database. The difficulty is passing data around and sometimes, we have lost data and it took lots of time to find the bug.
* Register: we only need to get user input and insert the information into the database.

**[workout Section]**

Using RecyclerView, Viewadapter and raw layout to handle each row, using “intent.putExtra” to send the data from video list activity to ActualWorkout activity, and using MediaPlayer to play the video, also add MediaController to play, pause and stop the video. Finally, using SQLite to save video information into the log table.

**[Meal Plan Section]**

Meal plan consists of a total of 4 meal plans. These plans are available to users with two options: "Gain Muscle" and "Loss Weight," and different layout designs apply depending on the option. In addition, when the user clicks on each menu, Spinner or Radio Button functions are provided to select a vegetarian meal and a regular meal.

* **Database**

**Features**

* + Making data list in csv files (food.csv, foodingredient.csv)
  + Read Data from CSV files and store these data in the List variable
  + Creating Fitgain.db database and in food table and foodingredient in LoginActivity
  + Inserting data in the Mealplan

**Technical challenges & applied**

* + Use getResources().openRawResource(R.raw.food) to read data in the csv file and insert it into a table created through the list.
  + Connecting Database to multiple Activities using openOrCreateDatabase()

* **Meal\_PlanActivity**

**Features**

* + **Listview showing the entire menu**: It offers four meals through the Listview, and when the user clicks on individual meals, it moves to another activity with each meal through the Intent.
  + **Gain Muscle Button:** After clicking the button, the title of the TextView changes to "Gain Muscle," and another Listview shows two meals according to the selected option. Also, a brief description of the meal is shown using the Typewriter effect through TextView.
  + **Loss Weight Button:** After clicking the Button, the title of the TextView changes to Loss Weight, and the contents of the two menus and contents of the TextView change according to the option selected by Button. The TextView contents are displayed with the TypeWriter effect.

**Technical challenges & applied**

* + Using three designs for one layout, the height of ListView with four items is the same as the height of ListView with two items, so the TextView below ListView with two items is cut off

There were many difficulties, but when moving to ListView with two items, the problem was solved by manually setting the height to change to coding.

*ViewGroup.LayoutParams params =*

*listViewMealPlan.getLayoutParams();*

*params.height = 700;*

*listViewMealPlan.setLayoutParams(params);*

*listViewMealPlan.requestLayout();*

* + After clicking Gain Muscle Button and Loss Weight Button, Listview and TextView with two visible items were static, so the Typewriter effect was applied to TextView to get user attention.
  + By inserting data into the table in Meal\_PlanActivity, when a user revisits this activity through the navigation bar, the data insert is executed again, and a warning message is an output to the log.

After attempting several methods, this problem was solved through checkDBData() method, which first checks whether data was stored in the database before insert() method. If the data does not exist in the database, the true value is returned to insert the data, and if the data already exists, false is returned to not insert data.

* **Meal\_GainMuscle\_Activity**

**Features**

* + As a goal for Gain Muscle, I offer two meal plans-"Double Chocolate Cherry Smoothie" and "Bibb Lettuce Burger" in the same layout design. In this activity, photos, total calories, and food ingredients are displayed along with the name of the meal.
  + **Double Chocolate Cherry Smoothie:** Milk and protein powder can be divided into regular and vegetarian meal plan through the Spinner function.
  + **Bibb Lettuce Burger:** Beef and mayonnaise can split normal and vegetarian meal plan through Spinner function.
  + **Calculate Calories Button:** Each food ingredient has a different calorie, and after clicking the button, all meal plans are recalculated according to Spinner's options and determined whether it is a vegetarian diet or not. The changed total calories and the type of diet are displayed on TextView. All changes are announced through Toast message.
  + **Save Button:** After clicking this button, the total calorie, food name, and meal plan option (vegetable diet or not) data according to the option selected by the user are stored in the meal\_log table.
* **Meal\_LossWeight\_Activity**

**Features**

* + Two meal plans of Overnight Oats with Berries and Salmon and Vegetatble are provided in the same layout design as a goal for Loss Weight. In this activity, photos, total calories, and food ingredients are displayed along with the menu name, and the total calories of the diet will vary depending on the food ingredients you select.
  + **Overnight Oats with Berries:** Milk and Geek Yogurt can split between regular and vegetarian meal plans through the Radio Button. According to the Radio Button selection, the TextView value displaying Milk and Geek Yogurt is changed to a regular or a vegetarian meal plan.
  + **Salmon and Vegetable:** Salmon and Teriyaki Sauce can divide the regular diet and vegetarian meal plans through the Radio Button. Also, the value of the TextView displaying Salmon and Teriyaki Sauce is changed to a regular or a vegetarian meal plan by Radio Button selection.
  + **Calculate Calories Button:** It has the same functionality as the Calculation Calories Button of Meal\_GainMuscle\_Activity. Still, the only difference is that the layout design is different, and Meal\_LossWeight\_Activity allows users to select options through Radio Buton.
  + **Save Button:** It is the same as the Save Buton function of Meal\_GainMuscle\_Activity, but Radio Buton makes the option value for the user's choice.
* **Meal\_GainMuscle\_Activity & Meal\_LossWeight\_Activity**

**Technical challenges & applied**

* + Save the image data as Integer and display it in the XML file image section by receiving resources using getResources().getIdentifier(columns[3],"drawable",getPackageName())
  + Use the data received from Database and hand over the value to another activity through Bundle: The primary key value of the database is transferred to another activity through the Bundle function according to the selection of ListView picked by the user. And through that primary key value, data is searched, the string values and images are displayed, calorie values are retrieved, and the values are stored in the database.
  + Display most string values imported from the database food table and foodingredient table for each part in XML file design.
  + Insert a user's data value to a table created by another group member.

**[Log Section]**

* **Features:**Log table of workout and meal show what workouts user has done and what user has eaten today.
* **Technical challenges and techniques applied**

**1. Display many records**

Since some of the workout sessions are very short, it is possible that users would do more than 10 workouts. It will be very difficult to hard code the log table. So, we use SQLite to store the data; and use the recyclerview to display the record. Users could scroll efficiently.

**2. Draw the table**

To draw a table in a recyclerview is also a challenge for us. To solve this issue, we define a text view frame in a xml file in the drawable fold. Key steps include setting the radius to “0dp” which will generate a non-round corner for the cell and setting appropriate paddings and the stroke color. Then we should set the background properties to this view frame in the item layout file.

**Installation guide**

If we publish this application, the Mobile FITGAIN application can be searched in google play. Install the FITGAIN on an Android Device.

* Android version requirement: 8.0+
* API : 26
* compile Sdk: 31

Follow these steps:

1. In the google play store. Search “FITGAIN”

2. In the application page tap the downloaded and it will automatically install on the Android device.

3. Double-click the application icon to open FITGAIN.

4. When the user first-time opens the app, the user must register an account.

5. After creating an account and inserting body information.

6. Log in with ID and password.

7. Enjoy the app.

**Functions and user manual**

**[Login Section]**

|  |  |  |  |
| --- | --- | --- | --- |
|  | The first thing that appears on the app is the login page so user will have to login in order to use all features. User will have to enter their username and password. |  | If user enters the wrong username and password, an alert will let the user know that their username and password are invalid. |
|  | If user does not have an account with our app, there is a register button that allows the user to create a new account. |  | If user leaves a field of information empty, an alert will let the user know that they need to enter that field of information. |
|  | User will need to enter all information in every field in order for the user to register successfully and it will redirect the user to the login page where the user will need to enter their username and password. |  | After user login successfully, user will get redirected to the profile page where it will display all user information. |
|  | On profile page, there is a button that is updated. It will allow user to edit their profile. After clicking the button, it will redirect the user to the edit page. This page will show all information of the current user. |  | User can click each field and edit. |
|  | After finishing editing, user will click submit to update their profile. It will also redirect user to the profile page with the new user information. |  | User can log out which will redirect user to the login page. |

|  |  |  |
| --- | --- | --- |
| Graphical user interface, application  Description automatically generated  Choose goal | Graphical user interface, application, website  Description automatically generated  Gain muscle | Graphical user interface, application  Description automatically generated  Lose Weight |
| In the workout goal activity the users can choose the “gain muscle” or the “lose weight”. | In these two video lists the users can review the video title and description. Tap on any item will open ActualWorkout activity. | |

**[Workout Section]**

| In this activity, the users can tap on the play button to play the video,tap on pause button to pause the video and after the video finishing tap on “Done and save to log” button will save video title and calories to the log. Oce tap the “Done and save to log” button it will become invisible to avoid the users re-tap it. | | | |
| --- | --- | --- | --- |
| play | Pause | save button | After Tapping save button |
|  |  |  |  |

**[Meal Plan Section]**

|  |  |  |
| --- | --- | --- |
| **[Meal\_Plan Activity]** | | |
| **1** | **2** | **2** |
|  |  |  |
| After the user logs in, click on the meal plan icon in the navigation bar below to go to Meal\_Plan Activity.  On the first screen, all meal plans are displayed as ListView, providing users with two options: "Gain Muscle" and "Lost weight."  According to the user's choice, click the button to go to another screen with information on the appropriate meal plan. | After the screen call, the two diets according to the user's choice are displayed as ListView, and a short description of the entire meal plan below is shown as the TypeWriter effect.  Users can choose their diet by their taste and click on it to move to the next screen.  Click on the meal plan navigation bar to go to the first screen, where all meal plans are displayed. | It consists of designs and effects such as the "Gain Muscle" option, and only the content is changed to a "Low Weight" meal plan.    The user sets a meal plan by their preference and clicks the list to go to an activity where detailed information on the diet is provided. |

|  |  |  |
| --- | --- | --- |
| **[Meal\_GainMuscle\_Activity]** | | |
| **Double Chocolate Cherry Smoothie** | | |
|  |  |  |
| **Bibb Lettuce Burger** | | |
|  |  |  |
| Each ingredient has different amounts and calories, and the total number of calories according to the Spinner selection.  By Spinner's selection options, it will be displayed in a TextView below the calorie information whether it is "Vegan meal Plan" or "No vegan meal plan".  It shows what material has been added through a Toast message. | When the "Calculate Calories" button is clicked, the total calories are calculated based on the user's Spinner option selection and the values are displayed in a TextView representing Total Calories.  A Toast message indicates that total calories have been changed. | When the "Save" button is clicked, the menu name, total calories and meal plan type (vegan or not) according to the option selected by the user are inserted into the meal\_log table of the database.  A Toast message indicates that the data value has been saved in the meal\_log table. |

|  |  |  |
| --- | --- | --- |
| **[Meal\_LossWeight\_Activity]** | | |
| **Overnight Oats with Berries** | | |
|  |  |  |
| **Salmon and Vegetable** | | |
|  |  |  |
| According to the selection, the total number of calories is also changed, and the value is displayed in the TextView representing Total Calories.  It shows through a toast message that the meal plan option has been changed. | When the "Calculate Calories" button is pressed, the total calories are calculated according to the user's selection of the Radio Button option and the total calories values are displayed in a TextView.  It also shows which material has been added as a Toast message. | When the "Save" button is clicked, the menu name, total calories and meal plan type (vegan or not) according to the option selected by the user are inserted into the meal\_log table of the database.  A Toast message indicates that the data value has been saved in the meal\_log table. |

Total calories depend on user's meal plan and the options user chooses in their meal. The default value of Total Calories is the total calorie value of "No vegan meal plan". Design is divided by meal plan options, and the same diet consists of the same layout and effect.

**[Log Section]**

**Log Activity**

**Functions**

The application will record every user’s workout after the user watches and follow the video she/he chose. After the user clicks the “Done” button on the workout page, this activity will be written into the table of workout log of the application’s database. When a user enters the log page, there are two choices: workout log and meal log. After the user clicks the workout log, it will switch to the workout log table screen. The table includes the number of the record, the name of the workout user chooses and the calories that burned by each of the workout. At the end, the application will calculate the total calories of those workouts. We track the user's workout daily, because it will be more meaningful to let the user know what he/she has done today. If the exercise is not enough, he/she could try to make up later in the day, or the user could also choose to eat less.

After the user clicks the meal log, it will switch to the meal log table screen. The table includes the number of the record, the name of the meal user chose, the type of meal (“Loss Weight” or “Gain Muscle”) and the calories of each meal. At the end, the application will calculate the total calories of those meals. We track the user's meal daily, because it will be more meaningful to let the user know what he/she has eaten today. If the user eats too much, he/she could eat less later in the day, or the user could also choose to do more exercises. However, if we could add some graphs to let users track the record of workout and meal of the past week or the past month, it will provide a better user experience.

**Step-by-Step Usage**

| **Choose Log** | **Workout Log** | **Meal Log** |
| --- | --- | --- |
|  |  |  |
| Click the “Log” button at the right-bottom of the screen, user could enter the following log activity. | Clicking the “WORKOUT LOG” button, it will show the following workout log page; users could scroll the table if there are too many records of workout. | Clicking the “MEAL PLAN LOG” button, it will show the following meal plan log page, user could scroll the table if there are too much of meal records: |

**Incomplete functions**

* **Countdown timer:**  We did not use the countdown timer function to set the workout time and rest time. The initial plan was using GIF pictures to demonstrate the exercise and calculate how many calories burned is based on how much time users spent on the workout. However we changed the way to demo the exercise and ended up using the video with a timer on it. This reduces the workload for us and we can focus on other features
* **Meal Plan Suggestion:** The initial idea of meal planning was to provide a total calorie diet per day accordingly when users set goals and to suggest users with a plan to make their own diets by recommending a daily diet or by providing food categories. However, we decided to reduce the scale of the project a little according to the advice. Thus, the right amount of diet was provided to users according to their options, and they were given the choice of a vegetarian and a general meal plan.
* **Calculate BMI:** We cannot implement this feature because we cannot store float values in the database. We get data input from the user interface which is a float data type, but we cannot put it into the database as a float number.

**Disclaimer**

We certify that all the coding (except the navigation bar) and layout design were our original works and not using any of our previous works.

This report is under Fair Use: Copyright Disclaimer Under Section 107 of the Copyright Act in 1976; Allowance is made for "Fair Use" for purposes such as criticism, comment, news reporting, teaching, scholarship, and research.

Fair use is a use permitted by copyright statute that might otherwise be infringing. Non-profit, educational or personal use tips the balance in favor of fair use.

All rights and credit go directly to its rightful owners. No copyright infringement intended.

**[Group Presentaion Link]**

[**https://1drv.ms/v/s!AnWYlzltMw95hNB2trGRQSa88H8A4g?e=rYTUqa**](https://1drv.ms/v/s!AnWYlzltMw95hNB2trGRQSa88H8A4g?e=rYTUqa)

**Reference**

**[All Activity]**

**<Menu Bar Navigation Code> :Bottom Navigation View for all activities**

<https://thumbb13555.pixnet.net/blog/post/318620330-%E3%80%8Eandroid-studio%E3%80%8Fbottomnavigationview%E7%9A%84%E4%BD%BF%E7%94%A8%E6%95%99%E5%AD%B8>

**> Main Icon Launcher by flaticon**

<https://www.flaticon.com/free-icon/healthy-lifestyle_4310163?term=healthy&page=1&position=70&page=1&position=70&related_id=4310163&origin=search>

**> Workout Icon by flaticon**

<https://www.flaticon.com/premium-icon/workout_3000588?term=workout&page=1&position=2&page=1&position=2&related_id=3000588&origin=search>

**> Body Icon by flaticon**

<https://www.flaticon.com/free-icon/user_3349798?term=body&page=1&position=14&page=1&position=14&related_id=3349798&origin=search>

**> Log Icon by flaticon**

<https://www.flaticon.com/free-icon/files_569837?term=log&page=1&position=14&page=1&position=14&related_id=569837&origin=search>

**> Meal Icon by flaticon**

<https://www.flaticon.com/free-icon/dinner_2648958?term=meal&page=1&position=9&page=1&position=9&related_id=2648958&origin=search>

**[Workout Secion]**

**> Videos and cover:**

Workout\_Abs<https://youtu.be/AqUO8npvBIU>

workout\_chest<https://youtu.be/_q_-adYzkh0>

workout\_legs<https://youtu.be/vUf6sKEHKi0>

workout\_standing\_abs<https://youtu.be/oQ--HZDG2ew>

fullbodystretch<https://youtu.be/g_tea8ZNk5A>

lean\_legs<https://youtu.be/-oxQEqzdFmc>

flat\_abs<https://youtu.be/-I9Rp0HMatM>

burn\_arms<https://youtu.be/XbkFLvMDaZc>

belly<https://youtu.be/ohgLmY19jNg>

Tabata <https://youtu.be/ZCcX2Egirp4>

**[Meal Plan]**

**<Changing listview height in code>**

<https://stackoverflow.com/questions/40861136/set-listview-height-programmatically>

**<TypeWriter Code>**

<https://www.youtube.com/watch?v=HpfOsl-0ZSU>

**> Meal Plan Contents**

Double Chocolate Cherry Smoothie and Bibb Lettuce Burger

<https://www.bodybuilding.com/content/meal-plan-for-every-guy.html>

Overnight Oats

<https://blog.fitbit.com/healthy-recipe-vanilla-overnight-oats-with-berries-lemon/>

Salmon and Vegetables

<https://blog.fitbit.com/meal-plan-weight-loss/>

**> Photo**

Double Chocolate Cherry Smoothie (Photo by Eva Elijas from Pexels)

<https://www.pexels.com/photo/photo-of-chocolate-smoothie-with-raspberry-on-top-6261272/>

Bibb Lettuce Burger (Photo by Rajesh TP from Pexels)

<https://www.pexels.com/photo/close-up-photo-of-a-cheese-burger-1633578/>

Overnight Oats (Photo by Micheile from Unsplash)

<https://unsplash.com/photos/YKrvaWAbBXQ>

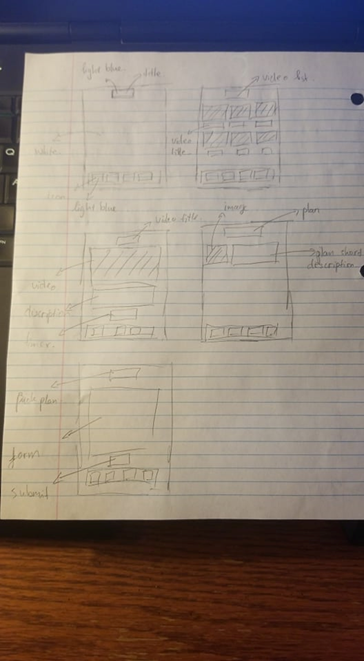
Salmon and Vegetables (Photo by Nadezhda Moryak from Pexels)

<https://www.pexels.com/photo/crop-guy-with-plate-of-tasty-fish-with-vegetables-in-teriyaki-sauce-4409306/>

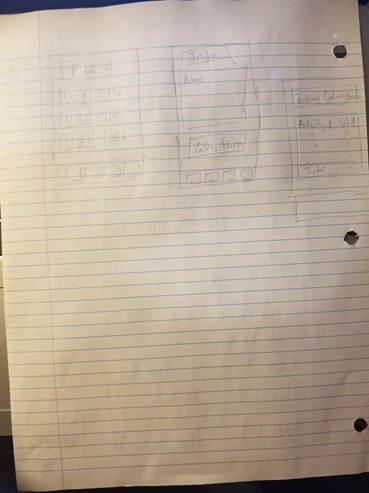
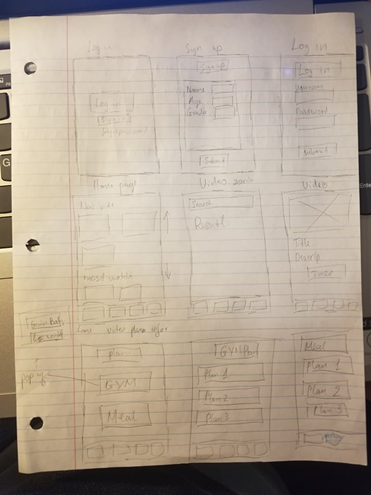
**[Individual Report]**

**300341490 Minh Dat Tu**

My first draft about the project interface:



Next design, I fixed some detail



In this project, my responsibility is the user management. So, I needed to create layouts for users to do login, logout, register, profile, and update profile. I created 4 activities which are:

|  |  |  |  |
| --- | --- | --- | --- |
| LoginActivity Layout | ProfileActivity Layout | UpdateProfileActivity Layout | RegisterActivity Layout |

In LoginActivity, I need to check if user exists in our database and display a message if user inputs the wrong username or password. check if user exist in database

In RegisterActivity, I need to check if user inputs all the fields which they need to enter. If they input missing information, I will need to display a message with corresponding error. Insert new record into database.

In ProfileActivity, I need to get user information from database and display that information to user

In UpdateProfileActivity, I need to get user information from the database and display that information to the user. Beside this, I also need to get new user input and update it in the database.

To make it able to connect with databases, I also need to create a user model, and a user database helper. In the user model, I store temporary information to publish it to the view, insert or update it to the database.

In database helper, I created a table called users which will store all users' information who complete register successfully. Function in table is doRegister, doLogin, doUpdate, and getUser

**[Individual Report]**

**300335751 Cheng-Yi Chen**

**The function I wrote in the program:**

* Use recyclerView in both video list activities.
* Use MediaPlayer in Actualworkout activity.
* Pass data from workout video list to Actualourkout activity.
* Save data of videos into workout log table.

**The design of the interface I did as below:**

* Workout goal activity and layout.
* Gain muscle video list activity and layout
* Lose weight video list activity and layout
* Actual Workout activity and layout.
* WorkoutAdapter class.

**Other:**

* Searching the sample of the navigation bar and altered it to suit our program then wrote a document to teach team members how to use it in every activity.
* Writing all the parts related to workout (except workout log activity) in the final report.
* Writing countdown timer part in Incomplete functions.
* Writing the Disclaimer
* Drafting Installation guide and complete it with SungAh.
* Found the icons on the Navigation bar.
* All the Pictures used in the workout part and searched Videos. (except edited the videos: cut a fragment from the videos. This part is completed by Dong

|  |  |  |  |
| --- | --- | --- | --- |
| Shape  Description automatically generated with low confidence | Shape  Description automatically generated with low confidence | Shape  Description automatically generated with low confidence | Shape  Description automatically generated with low confidence |

**[Individual Report]**

**300341721 Sung Ah Kim**

At the beginning of the project, our group suggested various ideas through active meetings and improved understanding of the functions of the application and the goals the group aims for. All members of our group contributed almost equally to all meetings and reports.

As the group leader, I managed the schedule, shared plans, and created and shared Google Doc files. I led out to help collect and submit data.

In the final report, I wrote meal planning sections in *function and technical tasks, function and user manuals, and incomplete functions*. I also partially wrote *the application overview, installation guide, and reference.* I proceeded with the editing of the final report too.

I was in charge of the meal plan section, so I worked on overall design and programming of this part. In addition, I created and added a Splash function by referring to the draft design that I worked on before.

Features implemented in programming.

* *SetOnItemClickListener():* Provide a recommended diet according to ListView selection.
* *SetOnClickListener():* "Gain Muscle" and "Loss Weight" button options are classified and explained with the Typewriter effect.
* After selection of options, vegetarian and general diet choices are provided through spinner's *setOnItemSelectedListener()* and radiobutton's *onCheckedChanged()*.
* Calculates final calories through Button's *setOnClickListener()* and stores user-selected data in the database by Save Button's *setOnClickListener().*
* After the eventListener is executed announcement of the changing state contents is displayed in the Toast message.

My final design is shared as a screenshot in *the function and user manuals* of the group report above.

I created the idea sketches. Suggestions for early design ideas.

**[Meal Plan Section]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A picture containing table, indoor, vegetable, items  Description automatically generated | A picture containing graphical user interface  Description automatically generated | Diagram  Description automatically generated | A picture containing text, businesscard, screenshot  Description automatically generated | A picture containing background pattern  Description automatically generated | A picture containing text, businesscard, screenshot  Description automatically generated | A picture containing text, food, vegetable, carrot  Description automatically generated |

**[Workout Section]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Text  Description automatically generated with medium confidence | Graphical user interface, text, application  Description automatically generated | Graphical user interface  Description automatically generated | Text  Description automatically generated with low confidence | A picture containing text, businesscard  Description automatically generated | Graphical user interface, text, application, chat or text message  Description automatically generated | A screenshot of a phone  Description automatically generated with low confidence |

**The Android file that I worked on**

|  |  |  |  |
| --- | --- | --- | --- |
| **Activity and class used in the meal plan** | | **The layout used in the meal plan** | |
| Meal\_PlanActivity  Meal\_GainMuscle\_Activity  Meal\_LossWeight\_Activity  Meal\_MenuAdapter  FoodIngredient  Food  Typewriter  Splash Activity | | activity\_meal\_gain\_muscle  activity\_meal\_loss\_weight  activity\_meal\_plan  meal\_menu\_layout  activity\_splash | |
| **Create Database and Table** | **Raw file** | | **Drawable** |
| food table  foodingredient table | food.csv  foodingredient.csv | | overnightoat1  salmon1  smoothie1  hamburger1  splashhome |

Other research to work on.

**[Existing application design research]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Graphical user interface, application  Description automatically generated | Graphical user interface, application, Teams  Description automatically generated | Graphical user interface, application, Teams  Description automatically generated | Graphical user interface, website  Description automatically generated | Graphical user interface, application  Description automatically generated |

**[Representative Icon Research & Suggestion]**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated | Icon  Description automatically generated |

**[ndividual Report]**

**300340704 Dong Nie**

**1. Functions:**

- **Database:** create mean log and workout table of the database and write interface for workout and meal activities to adding record to the database.

-  **Log table:** create recyclerview and the table of workout and meal table

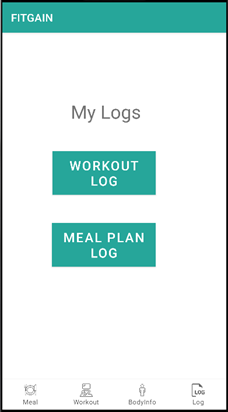
**- Display many records**

Since some of the workout sessions are very short, it is possible that user would do more than 10 workouts. It will be very difficult to hard code the log table. So, we use SQLite to store the data; and use the recyclerview to displays the record. User could scroll efficiently.

**- Draw the table**

To draw a table in a recyclerview is also a challenge for us. To solve this issue, we define a text view frame in a xml file in the drawable fold. Key steps include setting the radius to “0dp” which will generate a non-round corner for cell and setting appropriate paddings and the stroke color. Then we should set the background properties to this view frame in the item layout file.

**2. Design the interface of the log part of the application**

****

**3. Video editing**

Responsible for editing workout videos on YouTube provided by another team member to reasonable length.