KEEP CYCLE

Software Design Specification

22.05.2022

İbrahim Hakkı CANDAN 150118061

Atila İlhan YATAĞAN 150118033

Serkan KOÇ 150118073

Ramazan KARKİN 150119512

Aykut BAŞYİĞİT 150115854

Prepared for

CSE3044 Software Engineering Term Project

Table of Contents

[Introduction](#_30j0zll) **3**

[Purpose](#_1fob9te) 3

[Statement of scope](#_3znysh7) 3

[Software context](#_2et92p0) 3

[Major constraints](#_tyjcwt) 3

[Definitions](#_3dy6vkm) 3

[Acronyms and Abbreviations](#_1t3h5sf) 3

[References](#_4d34og8) 3

[Design Consideration](#_2s8eyo1) **3**

[Design Assumptions and Dependencies](#_17dp8vu) 3

[General Constraints](#_26in1rg) 4

[System Environment](#_35nkun2) 4

[Development Methods](#_1ksv4uv) 4

[Architectural and component-level design](#_44sinio) **4**

[System Structure](#_2jxsxqh) 4

[Architecture diagram](#_z337ya) 6

[Description for Component n](#_3j2qqm3) 6

[Processing narrative (PSPEC) for component n](#_1y810tw) 6

[Component and interface description.](#_4i7ojhp) 6

[Component n processing detail](#_2xcytpi) 6

[Dynamic Behavior for Component n](#_1ci93xb) 6

[Interaction Diagrams](#_3whwml4) 6

[Restrictions, limitations, and constraints](#_2bn6wsx) **7**

[Conclusion](#_qsh70q) **7**

# Introduction

Tracking progress has always been necessary for fitness and healthy life, therefore we decided to build a mobile app for those who want to keep track of personal progress. Keep Cycle is a mobile app to make fitness life easier, especially for people who want to live a healthy life.

## Purpose

This is the Software Design Specification for the Keep Cycle fitness app. The document will outline in detail the software architecture and design for the Keep Cycle mobile app. It will inform the design pattern, the system's operation, expected input, and output. The general purpose of this application is to make user’s fitness life more enjoyable and easier to keep track of. Application is intended for everyone who wants to be fit except people with health issues such as heart diseases, old people, etc.

## Statement of scope

The scope of the Keep Cycle is to provide users with tracking health and fitness achievements.

Keep Cycle will keep the records of the diet, body fat, ideal weight and exercise routines of the user. And it will help the user to keep track (total calories taken, number of repetitions for each exercise etc.). This application also supports calorie calculator, body fat calculator, ideal weight calculator, plate calculator, 1RM calculator. There will be a graphic for tracking the weights day by day. Also you can watch the correct form of the exercises. If you are taking pills there will be a countdown for you.

**The Essential Requirements**

· Keep Cycle will keep the records of the weight, height, and exercise routines of the user.

· It should be able to calculate the ideal weight according to the body index of the user.

· This application also should support calorie calculator, body fat calculator, ideal weight calculator, plate calculator, 1RM calculator.

· The countdown timer can help you get adequate amounts of essential nutrients if you can not eat a nutritious variety of foods on time.

**The Desirable Requirements**

· There will be a graphic for tracking the weights day by day.

· The average amount of calories given according to the exercise movement and the amount of calories given according to the food taken.

**The Future Requirements**

· The app will include exercise programs based on individual users according to their needs.

## Software context

Tracking improvement in their body has always affected people's mental health. Especially during this pandemic, physical activities have been eliminated from people's lives. So, people want to exercise to maintain a good fitness level. They will be able to see how much and how to exercise using the application.

## Major constraints

* There is a time constraint for our project. We must finish before the deadline.
* Budget, since we do not have a mac computer operated by the macos we cannot publish Keep Cycle on IOS.
* Database, since the app budget is also minimal we cannot have a database operated by our computers. We are using the firebase cloud systems.
* Scope constraint that users can only be able to keep tracking body index and workout achievements.

## Definitions

**User:** A person who wants to keep track of his own training program ,diet , body measurements.

**Dart:** It is a programming language.

**Flutter:** Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase

**FireBase:** Firebase is a Backend-as-a-Service (BaaS) app development platform.

**Supplement:** A dietary supplement is a manufactured product intended to supplement one's diet by taking a [pill](https://en.wikipedia.org/wiki/Pill_(pharmacy)), [capsule](https://en.wikipedia.org/wiki/Capsule_(pharmacy)), [tablet](https://en.wikipedia.org/wiki/Tablet_(pharmacy)), powder, or liquid.

**Basal Metabolic Rate:** estimated daily calorie allowance that your body needs to perform its basic functions.

## Acronyms and Abbreviations

SW: Software

UCD: Use Case Diagram

UI: User Interface

BMI: Body Mass Index

BMR: Basal Metabolic Rate

## References

-<https://mimoza.marmara.edu.tr/~birol/Courses/CSE3044/lecture_notes.htm>

-- Ian Sommerville, Software Engineering, 8th ed. 2007

# Design Consideration

## Design Assumptions and Dependencies

**Assumptions:**

1. It is assumed that the application will not work with a large number of people to facilitate implementation.

2. All group members will do their responsibility.

**Dependencies:**

1. Skill: The application that the user has enough skill to use on the phone.

2. Operating System: Android system is required which is higher than Android 10.

3. Internet connection is needed.

## General Constraints

There is a resource constraint that although we write our mobile application in a flutter, which allows us to build apps for a multi-platform, we cannot make it work on iPhone devices because we do not have a MacBook.

There is a free stored data constraint that allows you to get started at no cost.

Our mobile app can typically have access to various types of personal/sensitive data (such as weight, age, photo) provided by users. Personal data can be stolen, which should be protected and this can cause unexpected privacy impacts.

There is a software platform constraint Keep Cycle is an android project, so we can do what this platform allows.

Keep Cycle mobile app project that supports the English language, but people who know a little bit of English can easily use the app.

Since our mobile application is not finished, we cannot say exactly how much size is needed, but it is 30MB on average.

## System Environment

The mobile application has been developed on Windows 10 operating system.

While developing the application, some of our friends used 8GB of RAM on their PC, while some of our friends used 16GB of RAM.

We used Android Studio and VSCODE as an IDE for developing the Mobile app.

We are going to keep our data in the Firebase cloud service.

We also used some dart packages in our project.

## Development Methods

After deciding what kind of application we want to develop. We chose the agile methodology that adapts easily to changes and can be getting faster results. We created a WhatsApp group to work effectively together. During the project, we make video calls on average 2 days a week. We exchange ideas and information daily on issues that we are stuck with and are not certain about. We divided the project into iterations and assigned tasks to each group member. In each iteration, the finished parts of the project were shown to the other project group members, and views were exchanged. While developing the application, we did not deal with the small mistakes that we often encountered. We tried to build the application as soon as possible.

# Architectural and component-level design

## System Structure

First of all, this is a mobile application. As you open the app the first thing you will see is the sign-in screen. If the user does not have an account he can not sign in before he signs up.

As he gets into the sign up screen he has to fill in the required information. After filling the text fields he presses the sign up button and the user is signed up. The data that the user has entered is sent to the firebase authentication system. A user is created at firebase authentication and the user's data is stored in firebase firestore database system(No-SQL). When a user signs up he gets a default profile picture from firebase storage. Firebase storage is the storage for the user's profile pictures. This is the end of user sign up scenario.

After a user signs up he has to sign in to get into the application. He has to fill in the required text fields for the sign in screen. Firebase authentication service checks whether user information is valid or not.

As the user gets into the application there is a home page. There are 4 options on this screen. First one is the user's profile screen. Second one is the calculators page. Third one is the exercise tracker and the last one is the countdown for supplements.

Lets say a user enters the user profile screen. User profile screen gets the users data from the firestore database and lets the user see his information. In this screen there are 4 options. First one is the edit profile screen. Here the user can change his height or weight and the database updates this data. Second one is the edit profile picture where the user can change his profile picture via gallery or taking a picture. Firebase storage is updated when a user changes his picture and the url of this picture is sent to the database in order to update the url of the user's picture. Third one is the go to main menu button where the user can return to the home page. The last one is the logout button where users log out.

In the home screen if the user enters the calculators page there are 4 options here. First one is Calorie calculator.Second one is Body Mass Index Calculator. Third one is the Plate Calculator and the last one is the Ideal Weight Calculator.

Calorie Calculator is a calculator where the user enters a weight, a body fat percentage and chooses the exercise level. One of the outputs is for basal metabolic rate and the other one is for the calculated calorie need for a day based on user data.

Body Mass Index calculator is where the user enters height and weight then the result shows his BMI.

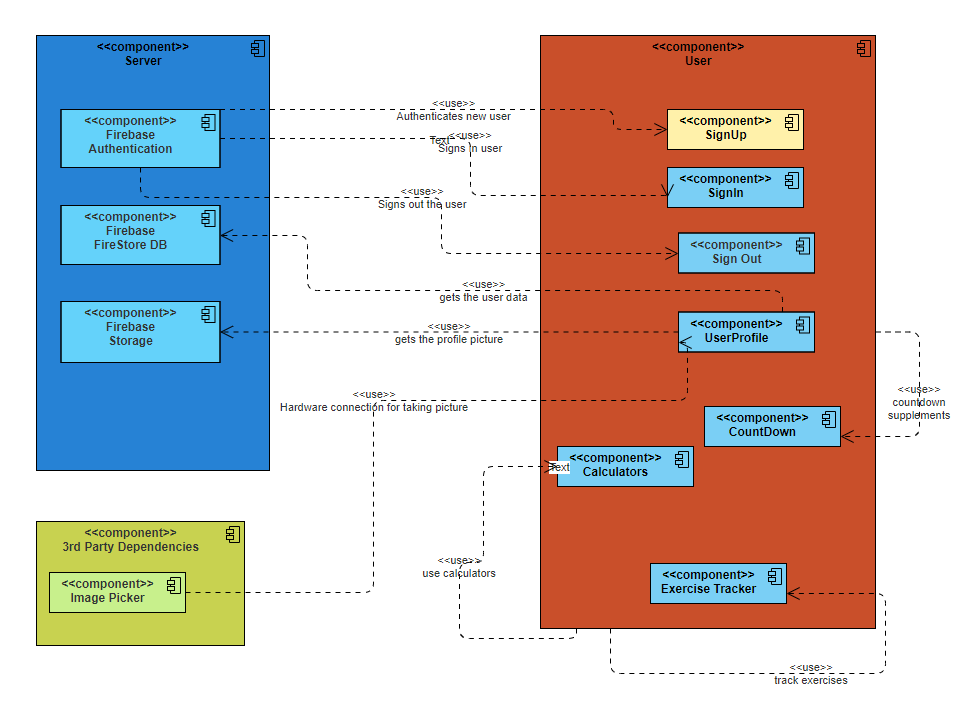
Plate Calculator requires only the weight of the plates that you want to lift. The calculator will tell you how many plates are needed to be put into each side of the bar.

Ideal Weight Calculator tells you what should be your ideal weight based on your height and gender.

Countdown feature is a basic countdown that you can create and edit after creation. It is a reminder that sends you a notification when you need to take your supplements. You can stop, restart, edit and delete the timer.

There will be a user interface where the user can enter the routine exercises to do. After the user has entered the exercises to be done, there will be a checkbox to be able to mark whether they complete these exercises. If there are exercises that he wants to delete, he will be able to delete them as well.

### Architecture diagram



## Description for Component n

3.2.1 Description of the Login Screen

| Identification | Login Screen |
| --- | --- |
| Type | Class/Form |
| Purpose | The login screen ensures that only users can access the system |
| Subordinates | This screen contains links to the following screens   * Main Menu Screen * Sign Up Screen * Reset Password Screen |
| Dependencies | The following screen links to this screen   * Main Menu Screen * User Profile Screen |
| Interfaces | The links are contained in the bottom half of the screen. |
| Resources | Firebase Authentication: enables the user to sign in to app |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to other screens using buttons in the half of the screen. Each button directs the user to a different screen that corresponds to the button that the user selects |
| Data | The data for the Login Screen is username and password entered by the user. It is validated with Firebase Authentication service. |

**3.2.2 Description of the SignUp Screen**

| Identification | Sign Up Screen |
| --- | --- |
| Type | Class/Form |
| Purpose | Sign Up screen allows new users to sign up to application |
| Subordinates | This screen contains links to the following screen:   * Login Screen |
| Dependencies | The following screen links to this screen   * Login Screen |
| Interfaces | The link is contained at the end of the screen. |
| Resources | Firebase Authentication: When the user enters required data to create a new account authentication system creates a new user.  Firebase Firestore Database: After the required data is taken from the user Firestore stores the data in the database. |
| Processing | The only type of processing required is inputting information into the text boxes and navigating to the Login screen via the button at the end of the screen. |
| Data | The data supplied by the user are name, email,password,height,weight,gender. After these data is taken from the user Firestore keeps these in the database. |

**3.2.3 Description of the Main Menu Screen**

| Identification | Main Menu Screen |
| --- | --- |
| Type | Class |
| Purpose | The main menu screen is a navigator of the application. |
| Subordinates | This screen contains links to the following screens:   * LogOut Screen * Calculators Screen * User Profile Screen * Exercise Tracker Screen * Supplement Countdown Screen |
| Dependencies | The following screens link to this screen:   * User Profile Screen * Edit Profile Picture Screen |
| Interfaces | The buttons are contained at the center of the screen. |
| Resources | None |
| Processing | Only processing type is that the user can select one of the buttons at the screen and link to other screens. |
| Data | No data for this screen. |

**3.2.4 Description of the Reset Password Screen**

| Identification | Reset Password Screen |
| --- | --- |
| Type | Class / Form |
| Purpose | This screen allows the user to generate a new password. |
| Subordinates | This screen contains link to the following screen:   * Login Screen |
| Dependencies | The following screen links to this screen:   * Login Screen |
| Interfaces | Text field for email and button for resetting password is at the top of the screen. |
| Resources | Firebase Authentication: When the user enters email and clicks the button firebase authentication sends an email to the user in order to reset the users password. |
| Processing | Users must enter email into the text field and click the reset button. The rest is handled by the Firebase Authentication. |
| Data | Only data required is the user's email saved in the database. |

**3.2.5 Description of the User Profile Screen**

| Identification | User Profile Screen |
| --- | --- |
| Type | Class |
| Purpose | This screen enables the user to see his information and edit his data. |
| Subordinates | This screen contains links to the following screen:   * Edit Profile Screen * Edit Profile Picture Screen * Main Menu Screen * Logout Screen |
| Dependencies | The following screens links to this screen:   * Main Menu Screen * Edit Profile Screen |
| Interfaces | User picture is at the top of the screen and user information is below the picture. The buttons for editing screens are at the end of the screen. |
| Resources | Firebase Firestore: Required data for the user is taken from the firestore database and if the user updates his data it is stored in the database. |
| Processing | Users can only select one of the buttons at the end of the screen. Users can edit profile data, profile picture, go to main menu or logout via buttons |
| Data | All the following data are stored in the firestore database and firestore storage.   * Profile picture * Name * Email * Gender * Height * Weight |

**3.2.6 Description of the Edit Profile Screen**

| Identification | Edit Profile Screen |
| --- | --- |
| Type | Class / Form |
| Purpose | The purpose of this screen is to update the height and weight of the user. |
| Subordinates | This screen contains link to the following screen:   * User Profile Screen |
| Dependencies | The following Screen links to this screen:   * User Profile Screen |
| Interfaces | There are text fields at the center of the screen where the user can enter his height and weight. There are two buttons for saving and resetting. |
| Resources | Firebase Firestore: When the user updates his height and weight firestore database is updated. |
| Processing | Users must enter height and weight. After that they can either save or reset the entered values. Firebase Firestore handles the rest of the process. |
| Data | Height and weight is a must here. |

**3.2.7 Description of the Edit Profile Picture Screen**

| Identification | Edit Profile Picture Screen |
| --- | --- |
| Type | Class |
| Purpose | This screen enables the user to change his profile photo. |
| Subordinates | This screen contains link to the following screen:   * Main Menu Screen |
| Dependencies | The following screen links to this screen:   * User Profile Screen |
| Interfaces | There are two buttons for selecting either uploading a photo via gallery or camera. |
| Resources | Firebase Storage: When a user changes his photo firebase storage keeps this photo and sends the url of the photo to the firebase firestore.  Firebase Firestore: The url sent to firestore from storage is stored in the database. |
| Processing | User can change his profile photo either with camera or gallery using the given buttons at the screens. |
| Data | A photo is required for this screen. |

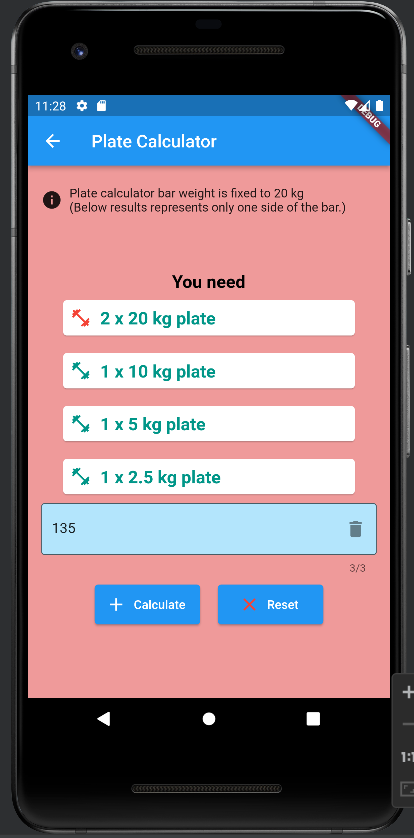
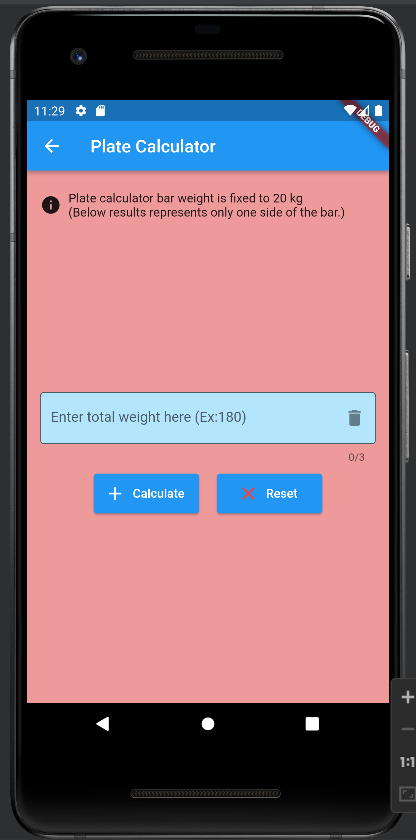
**3.2.8 Description of the LogOut Component**

| Identification | LogOut |
| --- | --- |
| Type | Class |
| Purpose | This component enables the user to logout |
| Subordinates | This component contains links to the following screen:   * Login Screen |
| Dependencies | The following screens links to this component:   * Main Menu Screen * User Profile Screen |
| Interfaces | There is no interface. Just a button for logging out. |
| Resources | No resource is required. |
| Processing | If the user wants to logout he can click the logout button. |
| Data | No data |

**3.2.9 Description of the Plate Calculator Component**

| Identification | Plate Calculator |
| --- | --- |
| Type | Class / Form |
| Purpose | This component helps the user to calculate the number of plates and their weights on each side of the weight lifting bar with specified total weight values. |
| Subordinates | This component contains links to the following screen:   * Calculator Selection Page |
| Dependencies | The following screens links to this component:   * Calculator Selection Page |
| Interfaces | This screen includes a text field and two buttons. Text field is for getting the user input. Two buttons are used for calculating the desired weight and resetting the screen. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | Plate calculator’s bar weight is fixed to 20 kg. Calculator outputs the numbers for plate weights based on user input. |
| Data | Total weight of the bar is provided by the user. |

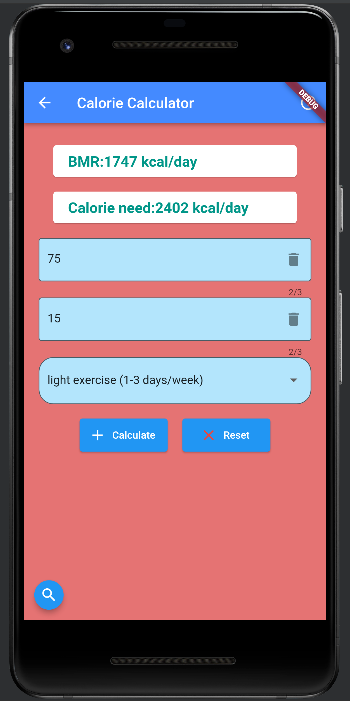
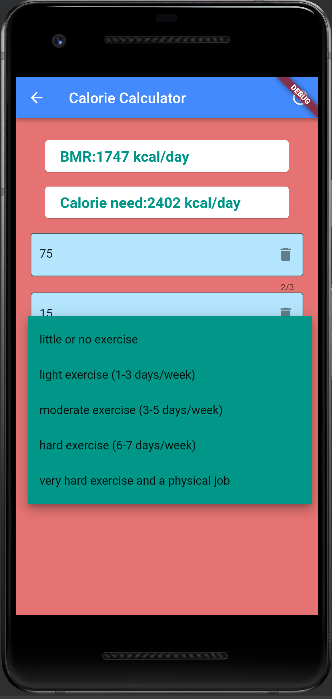
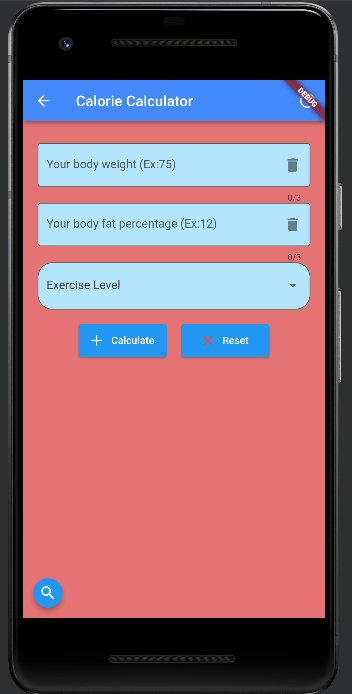
**Outputs of the Plate Calculator:**

****

**3.2.10 Description of the Calorie Calculator Component**

| Identification | Plate Calculator |
| --- | --- |
| Type | Class / Form |
| Purpose | This component helps the user to calculate the calorie need and basal metabolic rate of the user. |
| Subordinates | This component contains links to the following screen:   * Calculator Selection Page |
| Dependencies | The following screens links to this component:   * Calculator Selection Page |
| Interfaces | This screen includes two text fields, a dropdown menu and two buttons.    First text field is for getting the user input for the user's body weight.  Second text field is for getting the user input for the user's body fat percentage.  Dropdown menu is for getting the user input for the user's exercise level.  Two buttons are used for calculating the desired weight and resetting the screen. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | This calculator uses Katch-McArdle formula and allows you to calculate your Basal Metabolic Rate (BMR), which is the minimum amount of calories your body needs per day to keep functioning, assuming you were to do no exercise for that day. Based on BMR and exercise level it calculates the calorie needs of the user for a day.  Formulas:  LBM = (Weight[kg] \* (100 - Body Fat %)/100  BMR = 370 + (21.6 \* Lean Body Mass[kg])  Calorie/Day = little or no exercise  BMR \* 1.2  light exercise (1-3 days/week)  BMR \* 1.375  moderate exercise (3-5 days/week)  BMR \* 1.55  hard exercise (6-7 days/week)  BMR \* 1.725  very hard exercise and a physical job  BMR \* 1.9  Outputs : BMR and Calorie need per day (in kcal) |
| Data | Body weight, body fat percentage and exercise level of the user are provided by the user. |

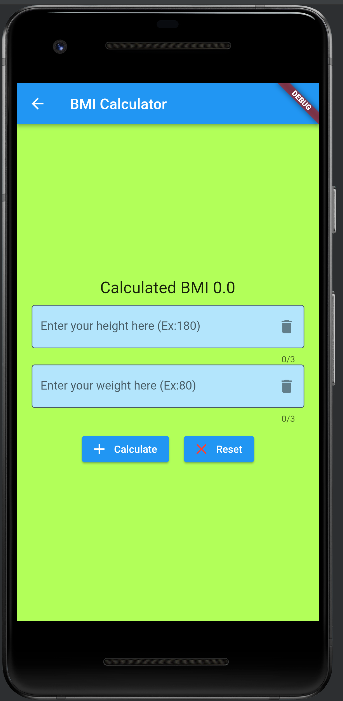
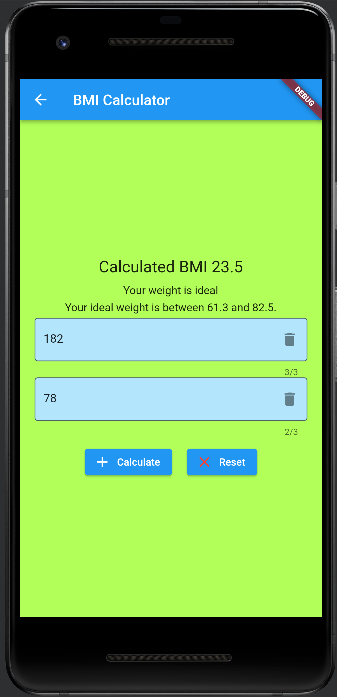
**Outputs of the Calorie Calculator:**

****

**3.2.11 Description of the BMI Calculator Component**

| Identification | BMI Calculator |
| --- | --- |
| Type | Class / Form |
| Purpose | This component helps the user to calculate the body mass index. |
| Subordinates | This component contains links to the following screen:   * Calculator Selection Page |
| Dependencies | The following screens links to this component:   * Calculator Selection Page |
| Interfaces | This screen includes two text fields and two buttons.  First text field is for getting the user input for the user's height.  Second text field is for getting the user input for the user's body weight.  Two buttons are used for calculating the BMI and resetting the screen. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | **BMI** = weight(KG) / (height(CM)^2 / 10000)  **idealWeightDownLimit** = height(CM)^2 \* (18.5 / 10000)  **idealWeightUpperLimit** = height(CM)^2 \* (24.9 / 10000) |
| Data | Body weight and height values are provided by the user. |

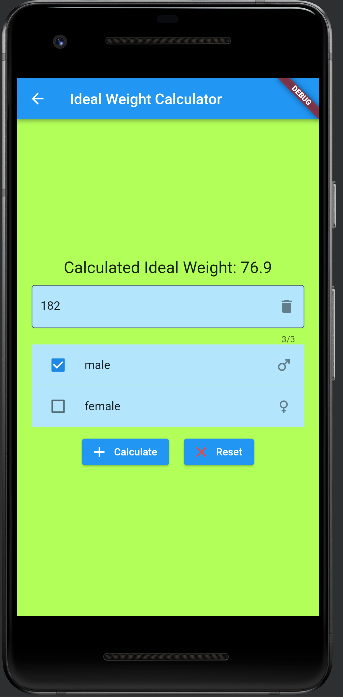
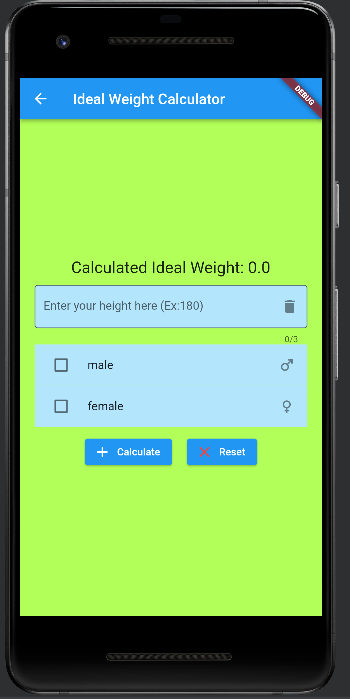
**Outputs of the BMI Calculator Component:**

****

**3.2.12 Description of the Ideal Weight Calculator Component**

| Identification | Ideal Weight Calculator |
| --- | --- |
| Type | Class / Form |
| Purpose | This component helps the user to calculate the ideal weight. |
| Subordinates | This component contains links to the following screen:   * Calculator Selection Page |
| Dependencies | The following screens links to this component:   * Calculator Selection Page |
| Interfaces | This screen includes one text field, two checkboxes and two buttons.  Text field is for getting the user input for the user's height.  Two checkboxes are for getting the user's gender.  Two buttons are used for calculating the ideal weight and resetting the screen. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | **Ideal Weight =**  (50 + (0.91) \* (height(CM) - 152.4)) -> male  (45.5 + (0.91) \* (height(CM) - 152.4)) -> female |
| Data | Height and gender info are provided by the user. |

**Outputs of the Ideal Weight Calculator Component:**

****

**3.2.13 Description of the Supplement Countdown Component**

| Identification | Supplement Countdown |
| --- | --- |
| Type | Class / Form |
| Purpose | This component reminds the user to take his/her supplements. |
| Subordinates | This component contains links to the following screen:   * Main Menu Screen * Alarm Edit Screen |
| Dependencies | The following screens links to this component:   * Main Menu Screen * Alarm Edit Screen |
| Interfaces | This screen includes one Container, three display Containers, five IconButtons and one FloatingActionButton.  First container includes all displays for the countdown.  Three containers contain hour, minute and second information.  Five IconButtons are used for Start, Stop, Reset, Edit and Delete operations.  The FloatingActionButton is for adding new alarms. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | A countdown is implemented in this component. It works separately for all alarms and gives a notification when the time runs out. |
| Data | Alarm name and frequency of supplement use in hours are provided by the user. |

**Outputs of the Supplement Countdown Component:**

**  **

**3.2.14 Description of the Exercise Routine Tracker Component**

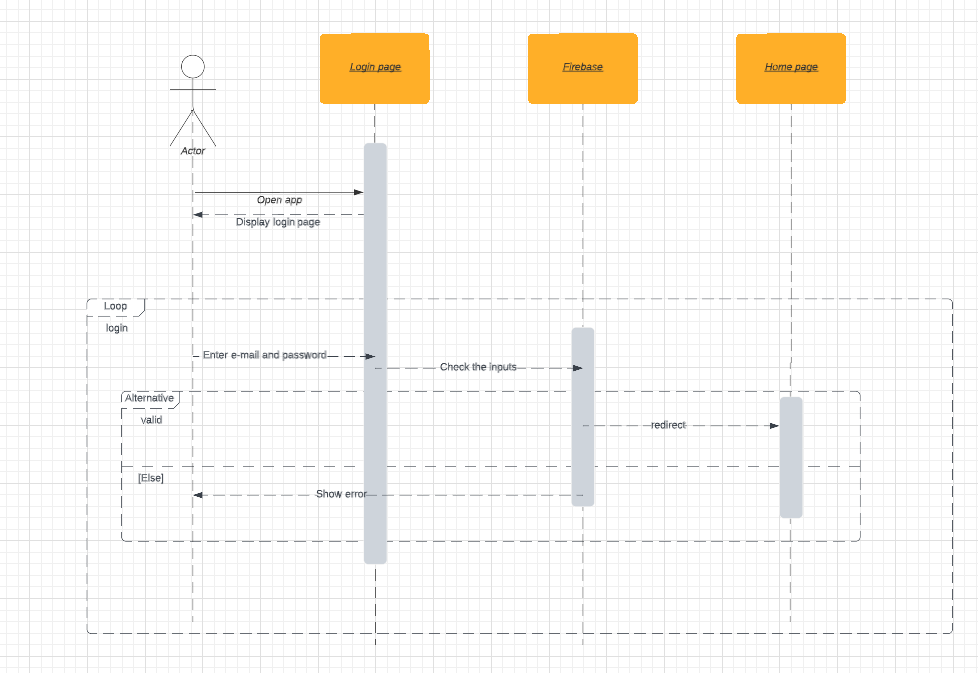
| Identification | Exercise Routine Tracker |
| --- | --- |
| Type | Class / Form |
| Purpose | User will be able to enter their exercise |
| Subordinates | This component contains links to the following screen:   * Main MenuScreen |
| Dependencies | This component contains links to the following screen:   * Main Menu Screen |
| Interfaces | This screen includes 3 buttons.  The first button is the FloatingActionButton required to open the second button.  There is one TextButton on the second screen so that the user can add exercises to do.  A checkbox will appear after the user enters the exercises to be done where the user can mark the exercises that the user has completed.  Those buttons are used for adding users' requirements. |
| Resources | No resource is required. All data is provided by the user. |
| Processing | After the user has entered the exercises to be done, they will be able to mark the checkbox when they complete these exercises. If there are exercises that he wants to delete, he will be able to delete them as well. |
| Data | The exercises to be done will be provided as an input to the TextButton by the user. |

## Dynamic Behavior for Component n

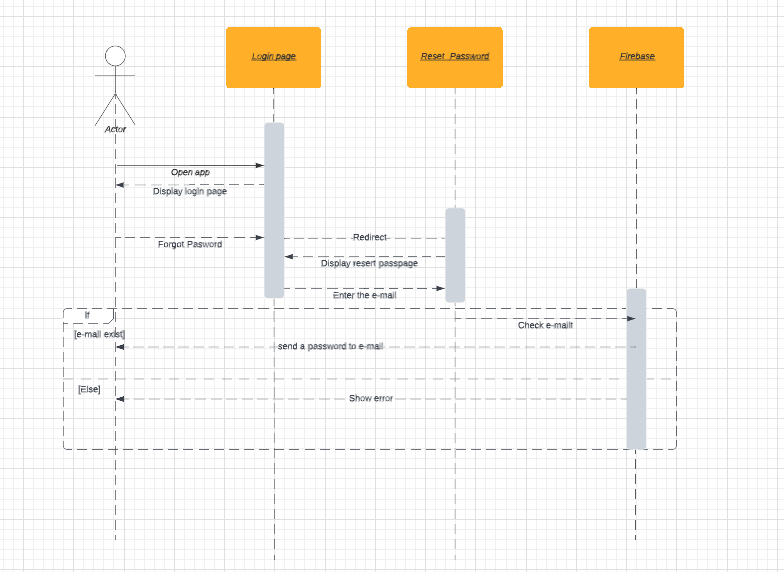
A description of the interaction of the classes is presented.

### Interaction Diagrams

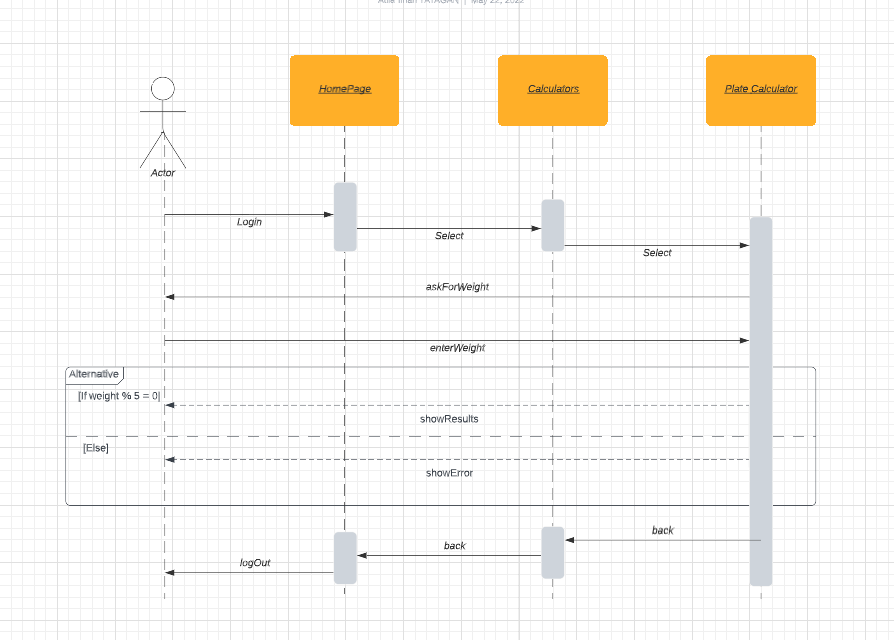
**Sequence Diagram for Login Page Component:**

****

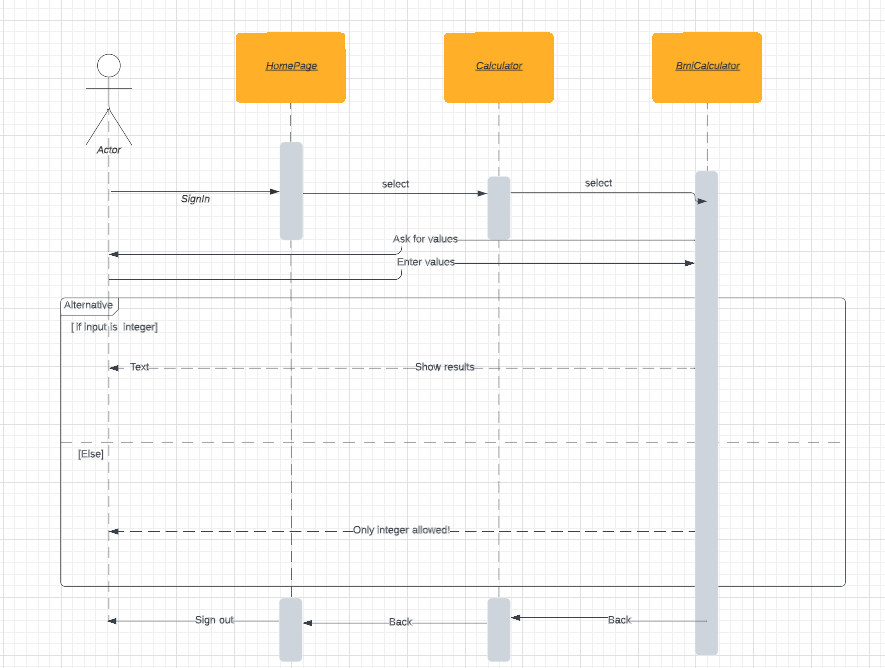
**Sequence Diagram for Forgot Password Page Component:**

****

**Sequence Diagram for Plate Calculator Component:**



**Sequence Diagram for Bmi Calculator Component:**

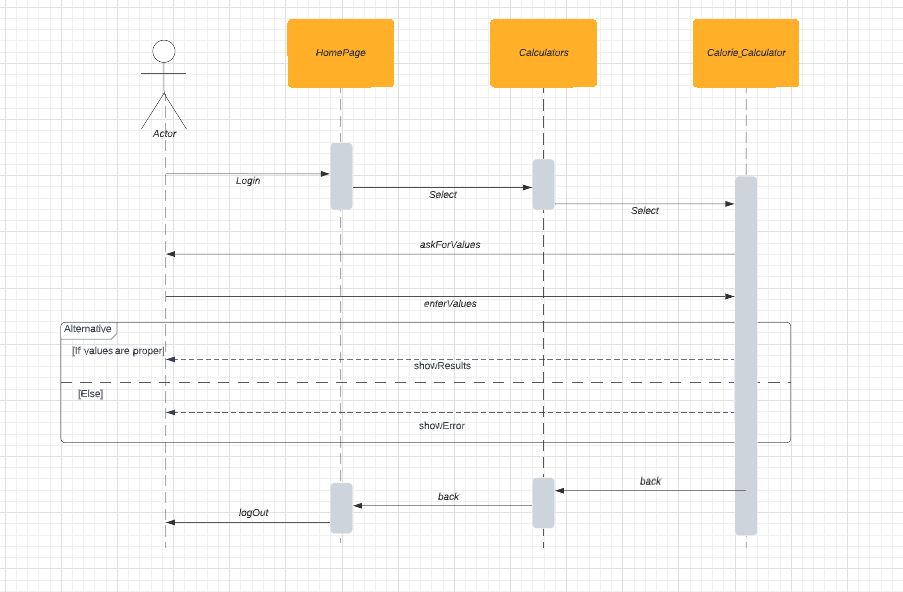
****

**Proper input values for the Bmi calculator component :**

**Height**: an integer must be provided. (leaving blank or anything else except numbers causes an error and shows a message.)

**Weight**: an integer must be provided. (leaving blank or anything else except numbers causes an error and shows a message.leaving blank or anything else except numbers causes an error and shows a message.)

**Sequence Diagram for Calorie Calculator Component:**



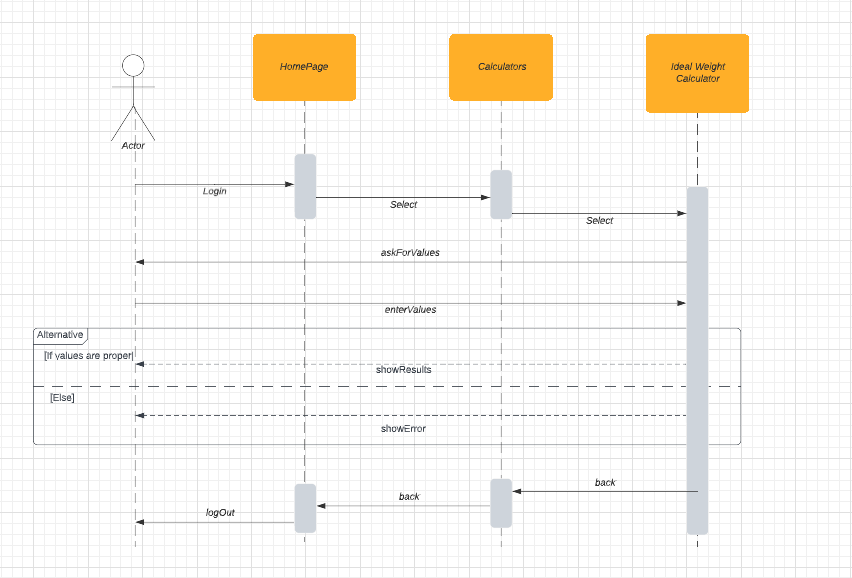
**Proper input values for the calorie calculator component :**

**Body Weight**: an integer must be provided. (leaving blank causes an error and shows a message.

**Body Fat:** an integer must be provided. (leaving blank causes an error and shows a message.

**Exercise Level:** any exercise level must be selected.

**Sequence Diagram for Ideal Weight Calculator Component:**

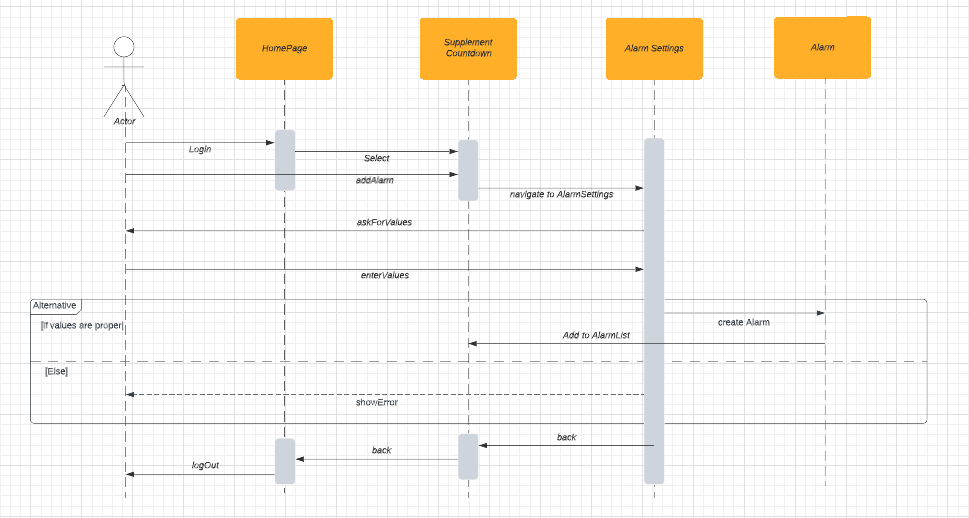


**Proper input values for the calorie calculator component :**

**Height:** an integer must be provided. (leaving blank causes an error and shows a message.

**Gender:** one of the two options must be selected (leaving blank causes an error and shows a message.

**Sequence Diagram for Supplement Countdown Component:**

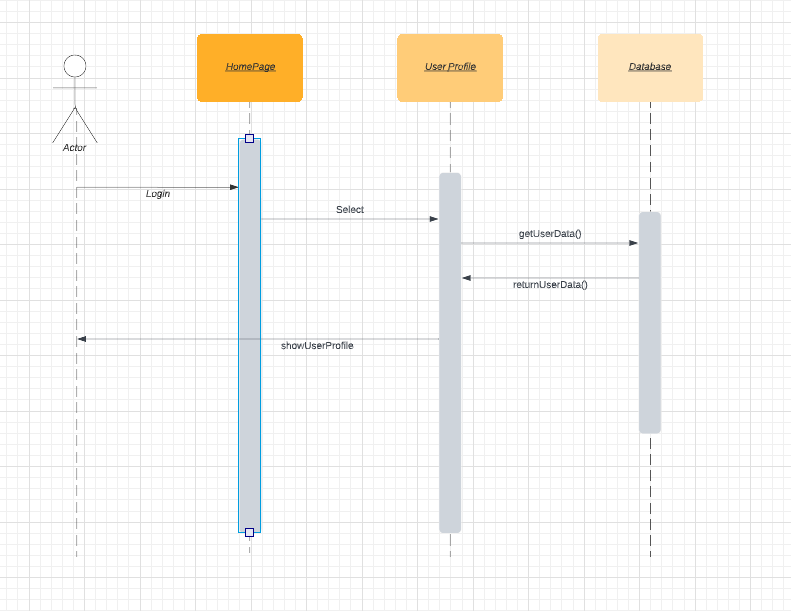


**Proper input values for the Supplement Countdown component :**

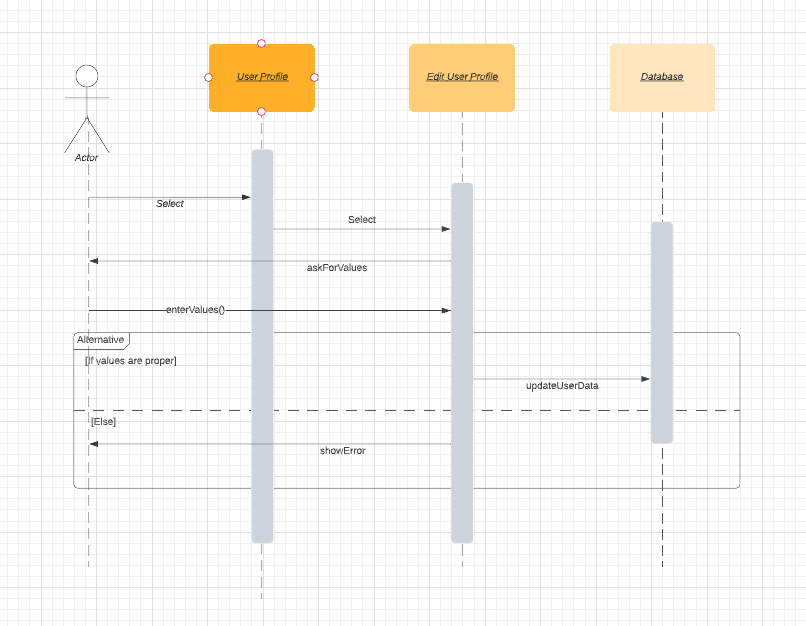
**Alarm Name:** a string must be provided. (leaving blank causes an error and shows an error message.

**Frequency of use:** an integer must be provided. (leaving blank causes an error and shows an error message.

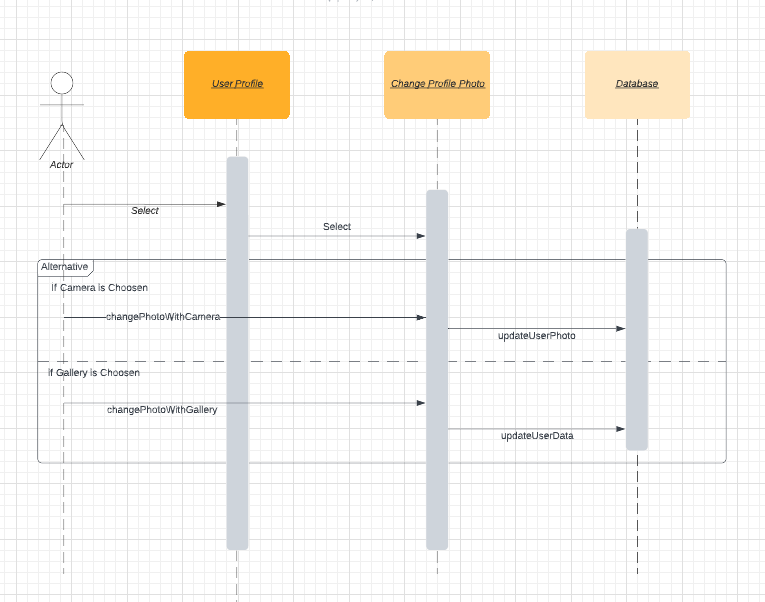
**Sequence Diagram For User Profile**



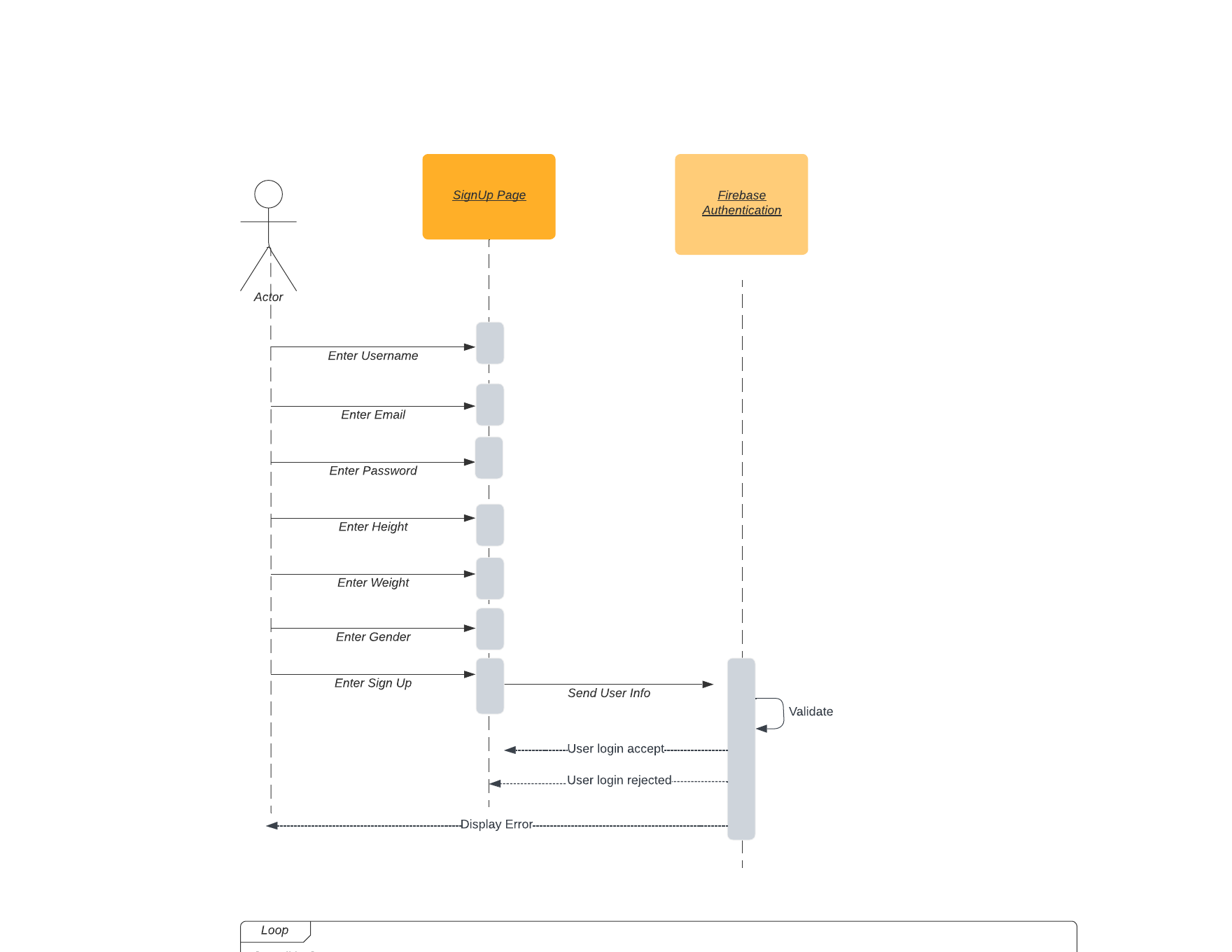
**Sequence Diagram For Edit User Profile**



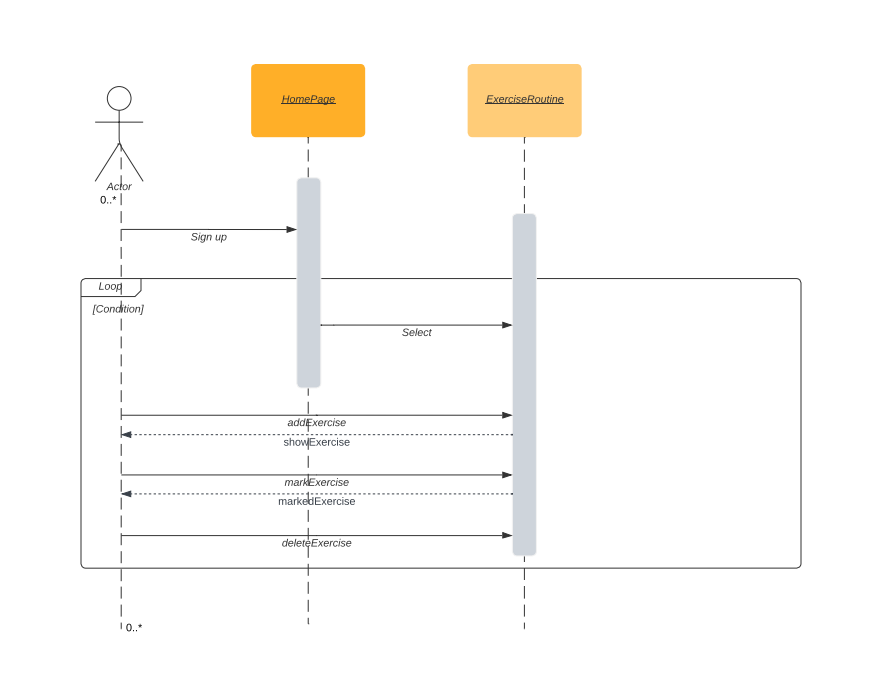
**Sequence Diagram For Change Profile Photo**



**Sequence Diagram For Sign Up**



**Sequence Diagram For Exercise Routine Tracker**



# Restrictions, limitations, and constraints

**4.1 Restriction**

* Time is the biggest restriction on a software development or lack of manpower if there is lack of developer time restrictions being more of a problem. In this project we also have a certain amount of time to finalize this Keep Cycle app.

**4.2 Limitation**

* In this case the biggest problem is the budget. Since we use Flutter to develop this application, we need to be able to use it on both platforms (Android and IOS).But we don't have a Mac operating system and we can't run Keep Cycle on IOS devices.

**4.3 Constraints**

* As constraints; memory, battery life, security and the computers can’t adapt to different screen sizes and orientations.

# Conclusion

In this document, design considerations for project Keep Cycle were dealt with. How our system works, how our system was decomposed, how these components work, their design architecture and connections, data design and flows were stated both by UML diagrams and by explanations. Moreover, user interactions were determined through user interfaces design. Libraries and tools which will be used during system development and operation were presented.