**EUROPEAN UNIVERSITY OF LEFKE**

FACULTY OF ENGINEERING

Graduation Project 2

Fitness Tracking application

### ANTHONY OLORI

### 194234

**This project is an application designed around the idea of fitness and running, the goal is to enable the user to track their runs as well as compare their activity with friends.**

**Supervisor**

Vesile Evrim

Publish Date

12-6-2023

**Table Of Contents**

[Your Name Surname i](#_gjdgxs)

[Your Student Number i](#_30j0zll)

[Your Supervisor Name i](#_1fob9te)

[Publish Date i](#_2et92p0)

[1.Introduction 1](#_3dy6vkm)

[1.1 Problem definition 1](#_1t3h5sf)

[1.2](#_4d34og8) Goals 1

[2. Literature Survey 2](#_2s8eyo1)

[3. Background Information 3](#_17dp8vu)

[3.1 Required & Used software 3](#_3rdcrjn)

[3.2 Other software 3](#_26in1rg)

[3.3 Hardware 3](#_lnxbz9)

[4.](#_35nkun2) Design Documents 4

[4.1](#_1ksv4uv) Data flow diagram 4

[4.2](#_44sinio) Your Context Diagram 4

[5.](#_2jxsxqh) Methodology 5

[6.](#_z337ya) Conclusion 6

[6.1](#_3j2qqm3) Benefits 6

[a.](#_1y810tw) Benefits to users : 6

[b.](#_4i7ojhp) Benefits to me : 6

[6.2](#_2xcytpi) Ethics 7

[6.3](#_1ci93xb) Future Works 8

[7.](#_3whwml4) References 9

# 

# 1.Introduction

## 1.1 Problem definition

Users interested in fitness can sometimes find it hard to know what exactly they do or how much they are doing. By tracking users' daily/weekly habits, users can feel rewarded or get encouraged to be consistent.

Example-Problems :

* Nancy, age 19. A Student of a university, finds out her friends have been using an application to track their runs, the application is also able to upload their runs online, she takes an interest and is excited to share her progress.
* John. A health conscious individual, wants to monitor his calorie intake and expenditure. downloads the application and uses it to monitor his activity and meet his daily goals. By Utilising the graphs and going on run sessions John gains insights into his fitness journey and can make informed decisions to achieve his health goals.

## Goals

* Calorie Calculation.
* Run distance and location tracking
* A social sharing feature .

# 2. Literature Survey

The proposed application is an application centred around fitness and user health tracking, similar applications exist, examples are samsung health,

**Samsung Health :** Like its name implies, this is an application helps track the users sleep, weight,diet and physical activity, it then serves insights as graphs into the users daily habits, it is a free application that comes bundled with newer versions of samsung phones, its core features are:

* Sport session tracking
* A pedometer.
* calorie tracking.
* dietary tracking

**Strava:** Strava like samsung health, is an application that centers around user health, unlike samsung health, it requires users to pay to access certain features. but it focuses more on physical activity and motivates the user more in that direction. it is built to encourage a community of runners, cyclists and swimmers to share their runs and it allows users to see the activity sessions and top distances of their friends, its features are:

* Sport session tracking
* A pedometer.
* activity sharing with photos and videos

It allows users to pair with a gps watch and head sensors to get a wider range of data.

In summary, these are the applications that are the basis for comparison, Fitwork will differentiate itself by being a daily log for user activity. By boiling down the unnecessary features it will be lightweight and feature slim, meaning new users will not be overwhelmed by an excess of features and will be able to know what the application allows them to do with a single glance.

**3. Background Information**

**3.1 Required software**

* **Visual Studio Code :**

Editor of choice.

* **Sqlite :**

Sqlite would be the storage for the local database.

* **Flutter :**

Application will be developed using the flutter Graphical User Interface library.

* **Firebase :**

used for storing.

**3.2 Required Hardware**

* **An android phone:**

This will be used for testing and deployment, as the application will be an android application.

**3.3 Other Components**

* **Google Maps API :**

Used for tracking location and displaying the data in a map.

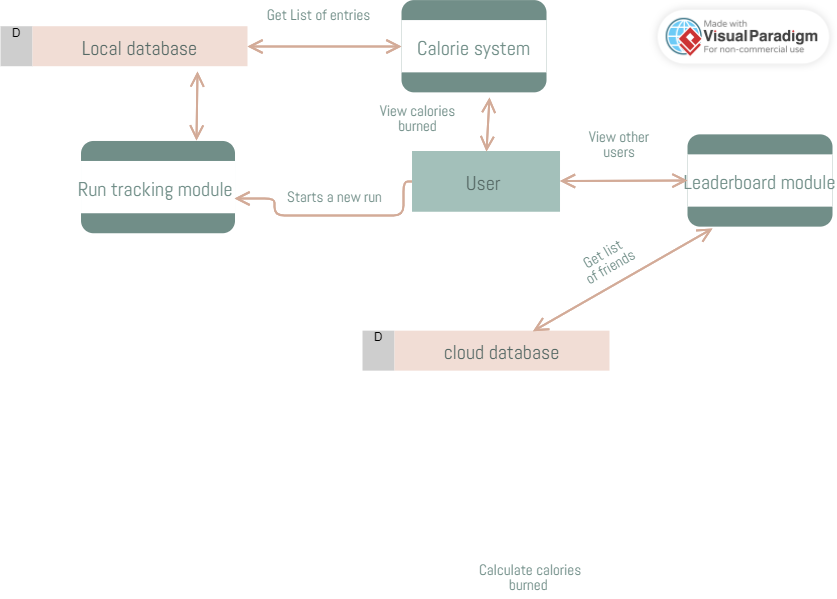
* **Firebase firestore:**

used for storing user data in the cloud

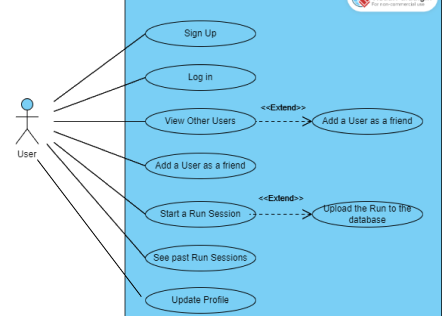
# Design Documents

If you have documents you must put here, your required document ( uml , class , dfd diagram etc…) do not forget to explain them shortly. If you have hardware you can add hardware architecture to. You can see below example Do not Forget Delete Examples !!

## Data flow diagram



## Use case diagram



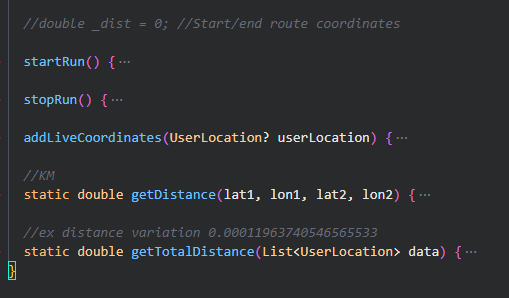
This outlines the features that the ways that the user is able to interact with the application.

# Methodology

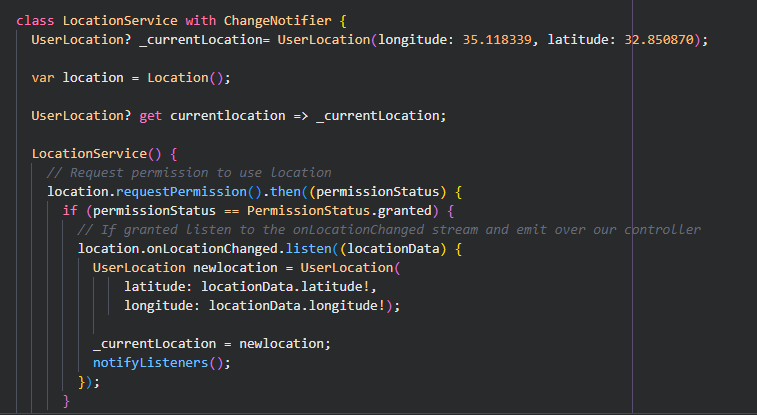
This application splits

**General architecture**:

I followed a design pattern in flutter that separates logic from UI. As much as possible, calculations, and things of that manner are handled by “helper” classes. These classes are responsible for one job usually, and are named as such, usually in a separate file as well. this makes it easy to track changes and tell at a glance what each piece of code is for. some parts of my application include;

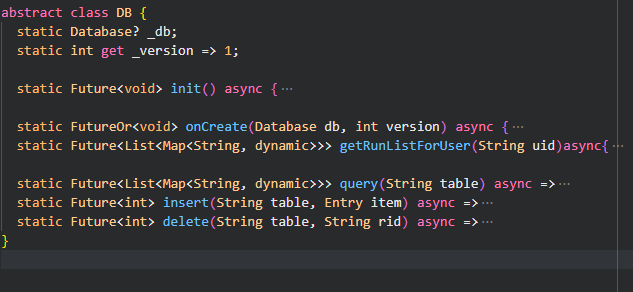
Run Helper:   
This class is responsible for providing the stop run and start run functions to the necessary pages, by setting up a stopwatch and other variables userful for tracking, they would continue to track up until the stop run button is clicked, from where the data captured can then be stored.

Location service:



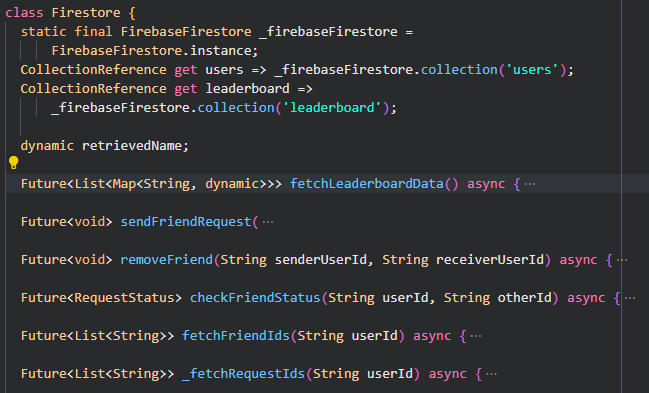
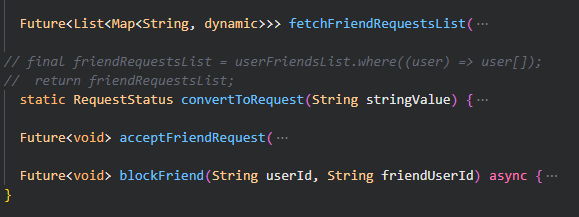
This class is responsible for tracking the users location in real time. Using another custom class called userlocation which houses just the latLng values for each location, it stores the new location on every update and notifies whatever processes that are currently subscribed to the stream that the location has changed. those classes can then implement a callback function that would handle a specific task

run\_database helper:



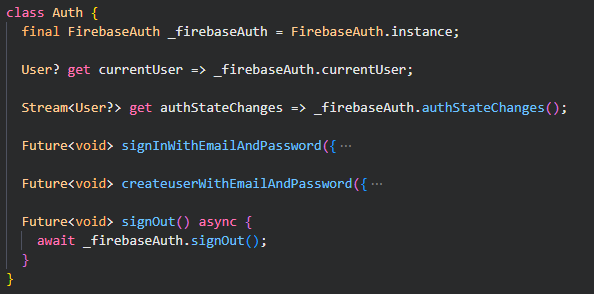
This class is responsible for CRUD operations, it allows users of the class to call relevant functions without initializing an instance member. All other developers have to do is call the innit function then use a “.then” callback function that would execute the after the abstract db has initialised its database as well as variables related to the path of the database.

firestore database helper:

  
  
This class is responsible for all functions related to the firebase firestore web service, it handles pathing, by providing variables that point to collections. if further down the line i want to change the name of the path i only need to change it once here and the change will have no impact on other code.

This class also provides functionality to send and receive friend request, fetch friend ids, which returns a list of unique ids generated by the firebase service as of the time of the user creating their account.  
fetchFriendReqList is responsible for querying the cloud database and filtering the results before rerutning the value. the filter checks for users who have sent a friend request before reutnring their profiles. this can be implemented to show the users data on demand.   
AcceptFriendRequest takes in the users id, the senders id and updates the firestore database for the user and the friend and adds them both to each others friends list respectively. blockFriend works the same way except instead of updating the status to show as friends it updates it to show as blocked.

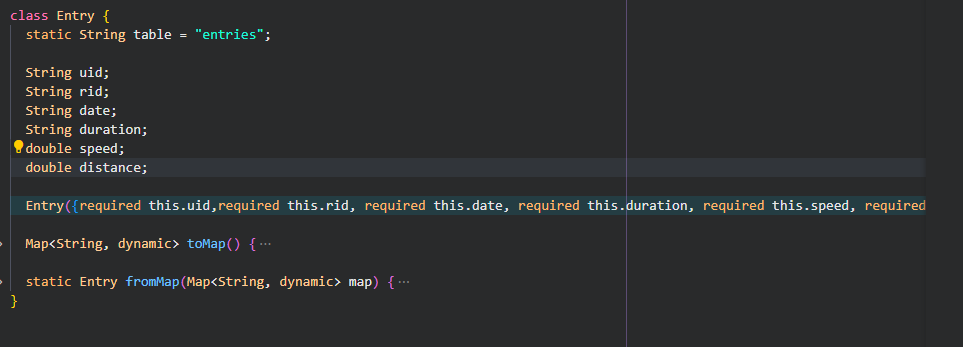
Firebase Auth:



This is responsible for user authentication, it holds the basic number of functions to help with that. because of the nature of the class it is used a lot. signInWithEmailAndPassword takes two strings, email and password and calls the instance of FirebaseAuth library which would then perform the necessary signIn or create user actions. By adding a layer of abstraction it yields a result that the pages on the front end are able to work by just calling a function and receiving an output, instead of getting bogged down with checks, try catches or error handling.

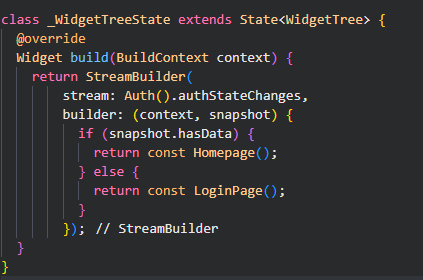
## Custom Types(Models):

Entry:



This refers to a run entry, this class is responsible for grouping information related to the users runs, when a run is created/recorded it is through this data type so i can ensure that the required information is being passed correctly, uid is the user id, while rid is the run id. the run id is usually a randomly generated string while the uid is the user id gotten from firebase Authentication package assigned to each user when they first create an account.

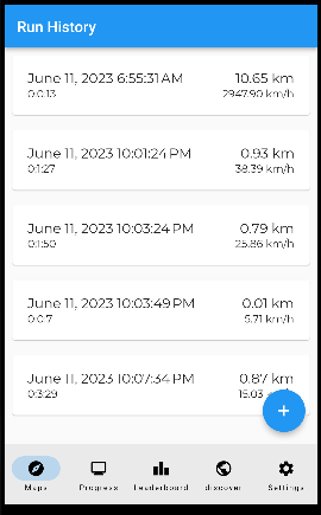
## Widget Tree:



In flutter, every object seen on screen is a widget, This class is named widget Tree because it controls what appears on screen, using a simple logic of, ‘if the user is logged in then return the homepage’, ‘else return the login page’, i am able to make it so that the application can be signed in and out from any page, all that page has to implement is a button that calls the sign out function in the Auth class and as soon as the state changes to ‘signed out’ the widget tree displays a login page, on successful login it directs the user to the homepage from where they are free to navigate the rest of the application, as this stream would only interfere when a change happens.

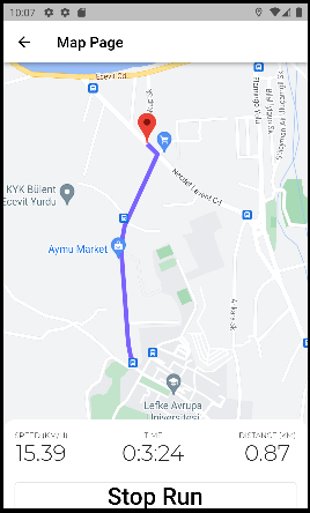
## Pages:

Run history Page:



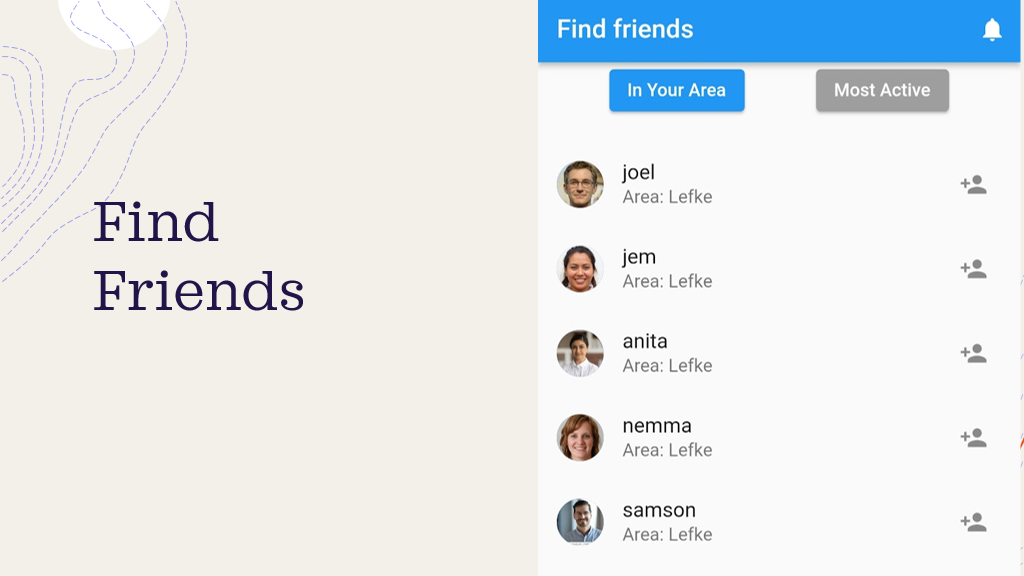
This page is the first page a user sees when they log into the application, it holds a history of past runs and shows information on them, the data is stored in the backend on the local database,

Map Page:



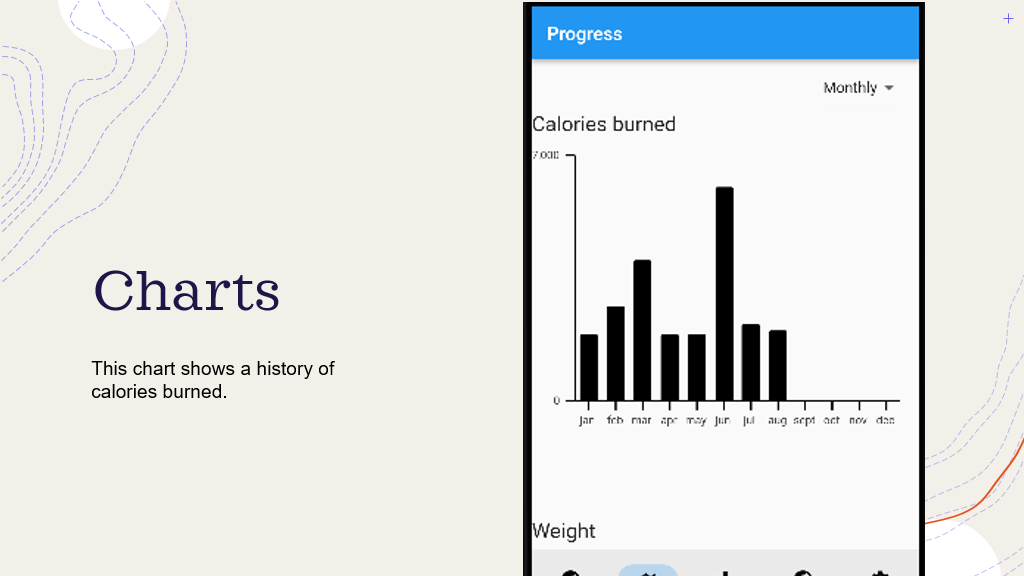
This is where i spent a lot of time writing and reviewing the code, as the main feature of the application it needed to be robust and account for a lot of cases, it uses google maps api to create the user interface but the api does not help with much else. tracking of the user’s time as well as distance and speed was handled in the back end and then displayed to the user. the purple line shows the path of the user from when the start run button was clicked. To account for if the user does not walk along an actual trail, i did not use an api to calculate the shortest route between two points but instead saved points at regular intervals and drew a polyline between them. Another example of a problem i had to account for was the possible ways a user might want to interact with the screen, when they are driving. They would expect that as they move the icon representing their location would stay centred on their location. but if they open the application while their hands are free then they might expect to be able to navigate and move their screen around even if their location is not changing. finding creative solutions to these kind of problems was a big part of the experience gained in this project.

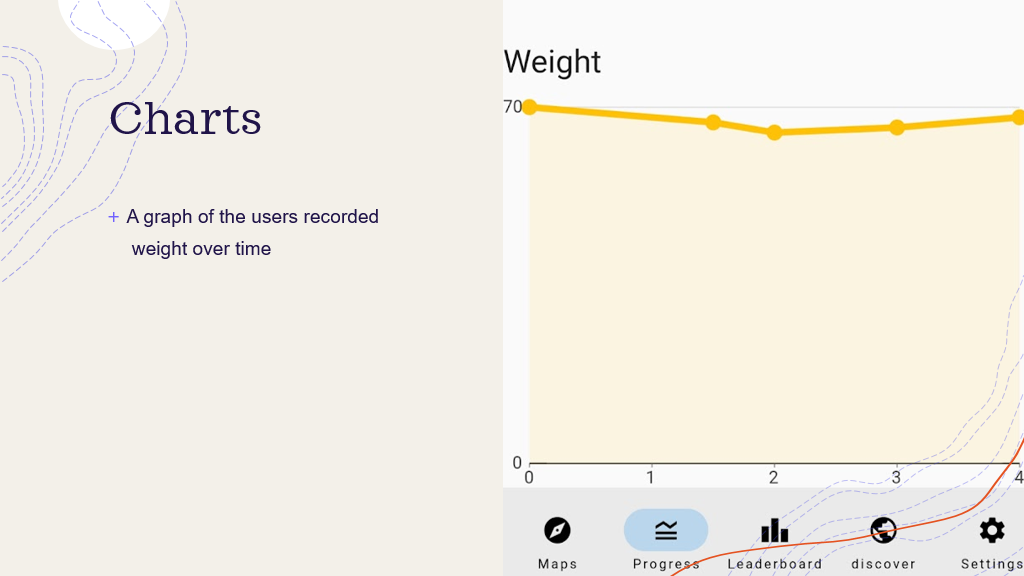
Discover Page:



This page is responsible for showing the user other users that can be added as well as a list of friend requests, if there are any for the current user to accept. on initial load it fetches the friends list from firestore as well as a list of friend requests from the firestore module.

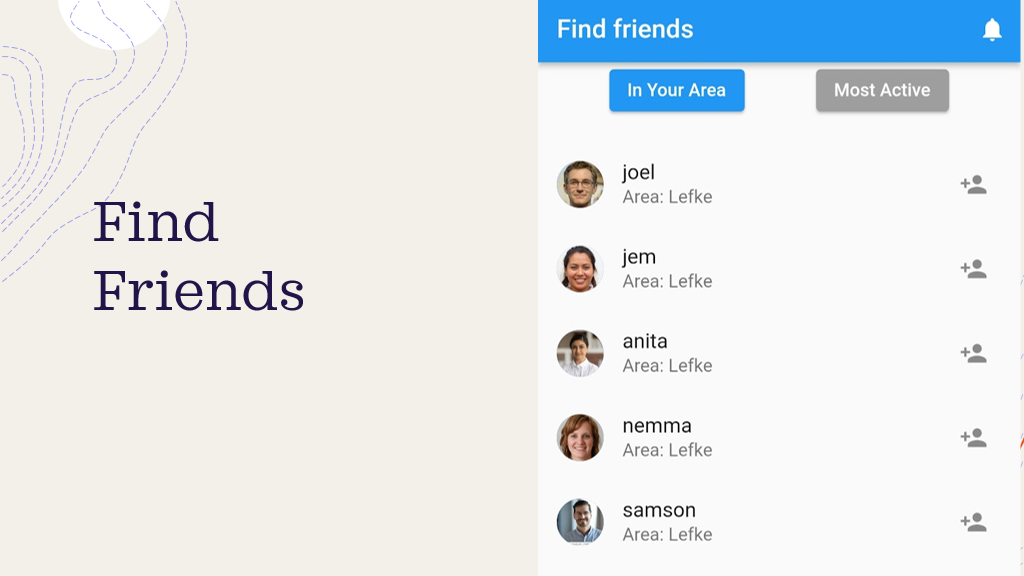
Chart Page:





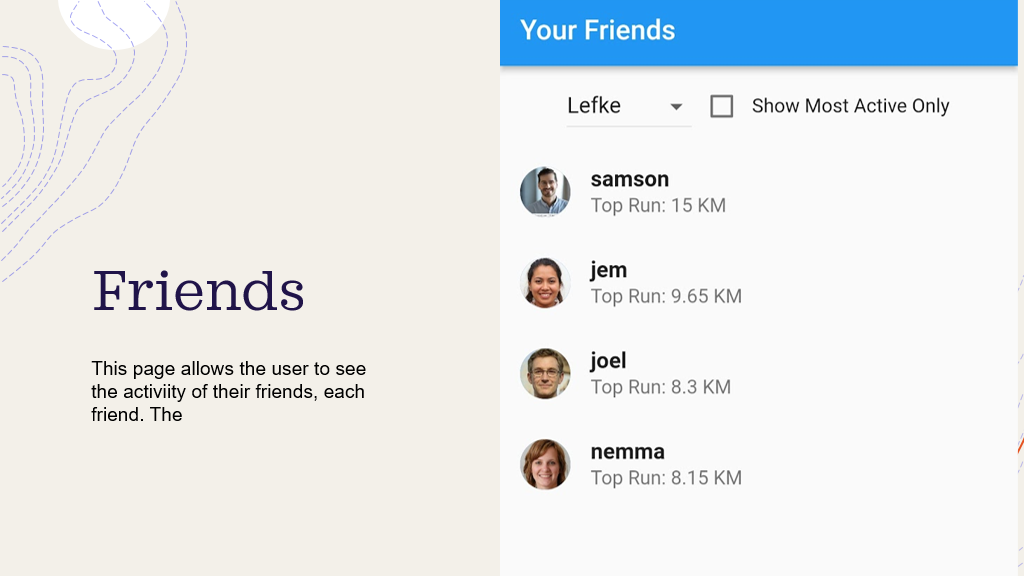
This page features two charts, "calories burned” chart, which uses a function that takes data from the “calories burned” table and converts it to a list of values that would be used in the graph. The next is a graph that displays the users weight and how it has changed over time. The screen has a button that allows the user to switch the timeframe between monthly and a weekly timeframe.

Find Friends Page:



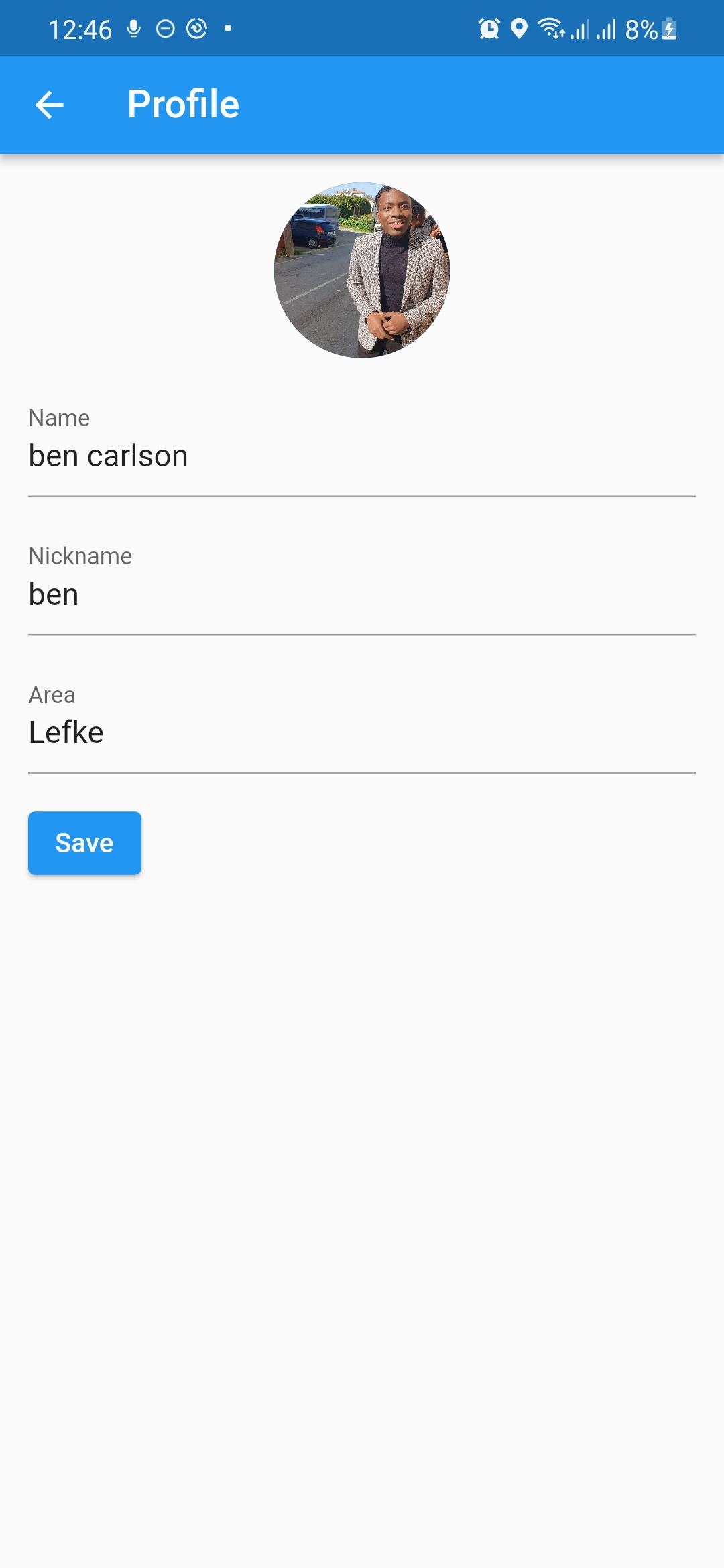
This page is where the user can add new fiends. On load the page fetches the data from firebase and then converts it to a form that can be then be easily displayed in the form of a list of users, where the user can then add choose to add a user or not to. from the notification bell icon it is possible to also see friend requests and accept or delete them.

Leaderboard Page:

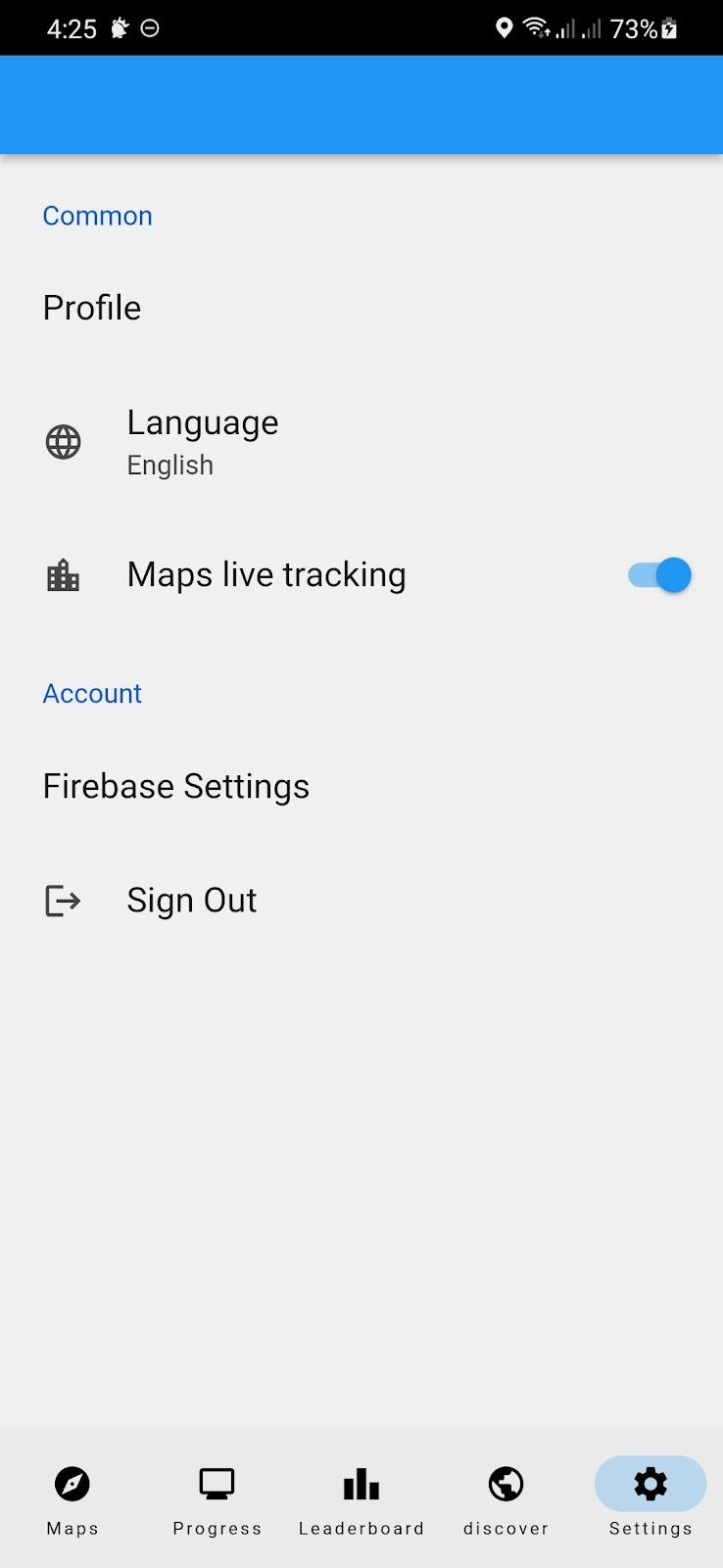


This page shows a list of users added a friend and their top runs, this is made with the intention that the user can view the activity of their friend group and compare against other friends.

Profile Page:



Here the user is able to change their information , after that has happened the application would send that data to the server to update it, including the profile picture. After the profile picture is uploaded for the first time, it is stored in the database using the Firebase Firestore service, and from then on it can be referenced by accessing the link to the image.



This is a page for a few settings that can be changed, the live tracking toggle is related to the feature where the google maps page centres the screen on the users location, it is useful when running or driving an you can't afford to keep scrolling as you move, but if you want to scroll around it can be inconvenient as it auto centres on the users location so a button was added to turn that off when necessary.

# Conclusion

This application is built to help users log their physical activity to a greater extent, its defining features are that it is built to be functional and by sending data between friends in a community more activity is encouraged.

* 1. **Benefits**

1. **Benefits to users :**

1. Step tracking: Users will have their step activity recorded daily and insights will be generated based on data, users will be able to reflect on

2. Health tracking: by recording weight and user activity, the application will be able to give the user insights into their health and activity over time.

3. It promotes walking: users being aware of their activity during the day will be incentivised to increase their step count for a higher number and a sense of satisfaction at the thought of meeting a goal.

1. **Benefits to me :**

Through this project i will learn about the following topics:

1. I have learnt about the object oriented programming language DART.
2. I have learnt more about mobile application development, specifically the use of design patterns

3. I have learnt about asynchronous programming.

**Why did I choose this project?**

I chose to build a mobile application because I would like to add this to my portfolio. A track record of past development means a better start when searching for a related job. After graduation I would like to pursue working as a developer so this application and one other will be the basis of my CV. I chose to build this project because it is practical and pertinent to the real world. These are services that exist and by developing them I have gained a better understanding on how they work and how to work with them.

## Ethics

Developing an application that tracks user location and activity does not come without its own set of concerns and issues, a case study being an incident in which a feature on the strava fitness application led to users accidentally revealing confidential information regarding the location of infrastructure critical to national security [2]. An ethical framework based on all these concerns is listed below:

* Users will have full control over their data and how long it would be stored, they will also have full visibility on what has been collected.
* User consent would be required when sending data online.
* Location tracking only happens when the user is expecting it, when the user is using the application .
* It is clear when data about the user is being stored.

## Future Works

This application is a demo, I would like to continue to build applications in the future and would likely continue to improve on it. Possible feature extensions include:

* Step tracking.
* Workout plans.
* Shared runs by friends.

Aside from feature extensions, one part of my application that I would like to work on in the future is the user interface. As my ability improves I would come back to make the application interface more user friendly and appealing to look at, by studying user interface design i would be able to come back and improve on the look and feel of the application and its features.

# References

**[1]** *How to Assess the Risk of Your Application*. (n.d.). Intel. Retrieved December 27, 2022, from <https://www.intel.com/content/www/us/en/developer/articles/training/software-security-guidance/secure-coding/how-assess-risk-your-application.html>

**[2]** Nebeker, C. (2018, March 3). *The Strava Heat Map: How a Social Network for Athletes Turned into a National Security Threat*. ReCODE Health. Retrieved December 27, 2022, from https://recode.health/2018/03/03/strava-heat-map-social-network-athletes-turned-national-security-threat/