Contents

Analysis

Problem identification....3
- Stakeholders....3
- Reasons for computational solution....3
- Functions....4

Research....4
- Existing similar solutions....4
- Parts that I can apply to my solution....5

Features of the proposed solution....6
- Initial concept of my solution considering this research....6
- Limitations to my solution....6

Interviews with potential users....7
- Conclusion from the interview process....8

Requirements....9
- Software and hardware requirements....9

Success Criteria....10

Design

User Interface Design....12
- First Page....13
- Second Page....15
- Third Page....15
- Fourth Page....17
- "Training Space" Page Design....19
  - Side Menu Navigation....19
    - Workouts Page....19
    - Routes Page....20
    - History Page....21
    - Achievements Page....21
    - Manual Entry Page....22
  - "Start Workout" Page....23
    - Running Stats Page....25

Stakeholder Input....27
Testing Methods....30
- Interface inputs and test tables....30

Development and Testing

Stage 1: The Questionnaire....33
Stage 1: Review....44
Stage 2: Saving of data into a text document....45
Stage 2: Review....52
Stage 3: Side Menu Navigation....53
Stage 3: Review....70
Stage 4: Working with Maps....71
Stage 4: Review....91

Stage 5: Final Testing……92
Stage 6: Stakeholder testing……97

Evaluation

Criteria Met……99
Success Criteria……99
Evidence……100
Limitations……105
Maintenance……107

Final Code

Final Code……108

Analysis

Problem Identification

According to the NHS website (https://www.nhs.uk/news/lifestyle-and-exercise/lack-of-exercise-as-deadly-as-smoking/), not exercising is as bad as smoking. Lack of exercise can lead to obesity, coronary heart disease, type 2 diabetes, mental health issues like depression and many more. There are a lot of different methods of exercising. However, today I am going to be focusing on one of the most popular and participated sports in the world – running. Even though it is considered to be quite a tough activity, running is regularly used to get in shape. However, a lot of people get demotivated to run – one of the major reasons being – not knowing any good routes, as a lot of people either really like a good scenery when they run, or want to know the exact distances when they run. For competitive runners that don't have a cutting edge technology, like a GPS watch, a way of tracking their progress is to run a route of known distance and time it. Not everyone can afford a GPS watch, however almost everyone owns a phone, which is the best way to spread different routes to users.

Stakeholders

The clients for this program would be users of smart phones. This application is aimed at a wide variety of people, ranging from absolute beginners trying to lose weight to competitive runners trying to take time off their personal bests. All levels can benefit from keeping track of paces, times and distances of their workouts, as well as discover new places to run, which could both benefit their fitness and make them look forward to their next workout. People who don't enjoy running with their phone have the ability to memorise the routes beforehand and time it themselves and enter their times manually. For people who take the phones on their runs, the app will automatically have the data from their run entered into the memory. Both types of people will be able to view the statistics from their previous runs.

The stakeholders that I chose to represent all the different types of people potentially using this app are:

1. Competitive runner Cameron Sale (ex Glenalmond Student) – takes running seriously; Runs 6 days per week, sometimes twice per day.
2. Thomas Kingan: Regular runner; Runs 3 times a week as games option in school.
3. Alice Lin: Hockey player; Runs for the benefits of fitness in hockey matches.
4. Elsa Ma: Occasional runner; Runs once a month

I believe that getting feedback from such a diverse set of people will really help me with finding the perfect solution for all.

Reasons for computational solution

Most people have access to a smartphone or a smart device, which have the ability to download applications and distribution of routes is much easier done that way. Also, an application is needed to keep track of distances and save workouts. The device would control main aspects such as distance and time tracking, however as mentioned previously, the user can enter this data as well. It's also a very desirable feature to store the records of all the runs in one place, therefore the app acts as a platform, which does that for the user.

Functions

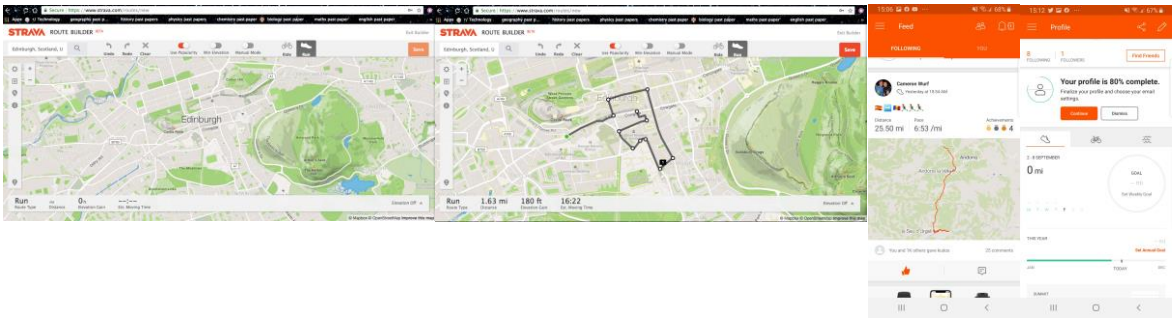
The app will have four main functions and one optional function. The main ones include: seeing good running routes around Edinburgh, tracking your workout as you run, saving your workouts (with distances, paces, times and routes) and workout suggestions to make you faster. At first, the app is going to ask for your aims in running (for example, to lose weight, training competitively, etc.). Depending on the answer, the app will suggest different workout to get to your goal faster.

The optional function is the idea of social networking in the way that users can interact by sharing their routes and times with each other over the app.

Research

Existing similar solutions:

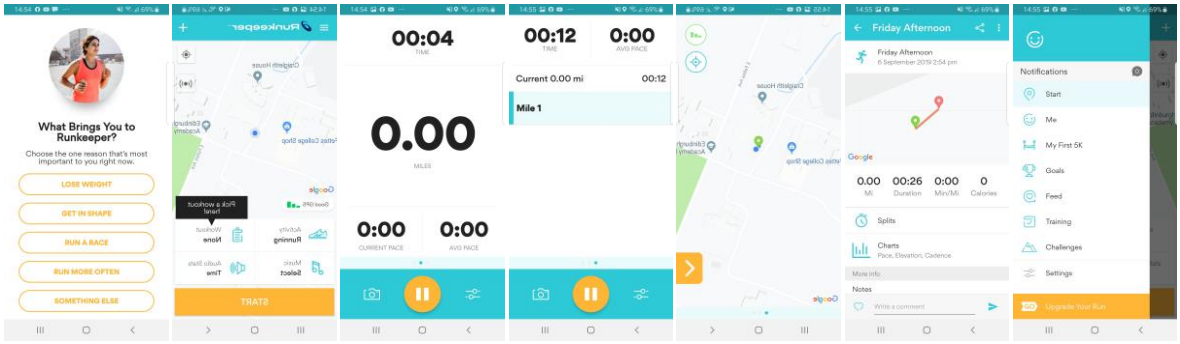
1. **Strava** – a very popular application for tracking workouts and sharing them with others.



Overview:

A lot of endurance professional athletes use this platform to share their workouts with their fans. In the "posts", it includes the map of the route that they ran/cycled, as well as distance/speed/time/power. The app has the ability to pair up with a GPS device, such as a watch and get the route from the run or bike session that way. People find that more convenient as it's much easier to run with a watch than a phone. One could also add "Manual Activity", which involves entering all the details manually, even including things like how you felt during the workout. To give the app a more "social" aspect, the function of "having followers" and "giving kudos" was added. Kudos have the same meaning as "likes" in social media like Instagram and Facebook. People can also enter and complete challenges for further entertainment, as well as find out near-by clubs to them. The function that I believe is missing is the ability to enter interval workouts manually, as well as in run mode, the app will not recognise that one is doing intervals.

2. **Runkeeper** – another very popular app that has similar features to Strava.



Overview:
Runkeeper is a very similar application to Strava, however has a much wider variety of sports available. Also function like connecting music, uploading workouts and having "Audio Stats" (computer saying every time you run a certain distance/time you enter) are also offered. Similar to Strava, one can set goals, enter challenges and save workouts they have done previously. Also similar to Strava, the app does not have the ability to enter interval workouts.

Parts that I can apply to my solution:

Both apps have a user-friendly user interface. It is easy to navigate through the apps, as each button is labelled and has a clear purpose. They achieved that by having a navigation panel either on the side or on the bottom of each page. That makes the usage of the app more enjoyable to the user.

Another good aspect of ***Runkeeper*** is that it says when you have ran a certain distance, as you are running (every one kilometre). That's a useful feature as – when running – checking the phone to see whether a certain distance has been ran is a hastle, as you have to stop, take your phone out of the pocket, unlock your phone, which slows down your average pace. Therefore it's easier to have it say every certain distance how much you have ran. I would also add to that feature to say what's the time every time it says the distance as well. That gives the user the indication of whether they are running too slow or too fast. Also, maybe a good idea to add a feature where the user can enter a target pace and every time they slow down and don't hit the target pace, the app tells the user to speed up. The downside to that is if the user accidently enters a pace that they cannot run at, for the whole run, the app is going to be repeatedly telling them to speed up, which can get annoying. So this feature is one to consider.

Another feature that, in my opinion, would be found useful by the user is highlighting the route as the user is running. If the user gets lost, they can retrace the route back by looking at the highlighted bit on the map and follow it back home. The route can also be saved and looked at later, which helps with the visualisation of the route.

Features of the proposed solution:

Initial concept of my solution considering this research:
My solution will be an application that when started will ask the user a few questions about their current personal bests, their goals and their preferred distance, which will all be saved to a text file on an internal storage. That, of course, involves permission from the user to save and create files on their phones. Once all the questions have been answered, it will go to a new page with either bottom navigation, but most likely a side slide navigation, like such:

The app will have suggested workouts for the distance that the user is training for. That distance can also be changed in the setting if you have decided to change focus of your training.

Limitations of my solution:
In my opinion, a good running app is one that can recognise interval training while it's happening. Interval training is where the user runs for some distance, then rests, and repeats that multiple times. Both *Runkeeper* and *Strava* don't have the feature, which detects interval workouts. The absence of this aspect in such popular and advanced apps suggests the complexity of the task. I think the way of detecting interval styled running is constantly monitoring the speed of the user and looking at massive variations in short period of time. I would definitely give this a try but I would not finish before the project deadline.

I have chosen to write the code in Android Studio. This limits the users as only Android operated phones will run the app. However, there is a way of solving this problem as there is software which can take any code for android phones and "translate" it to code that would be understood by other operation systems. I am also not going to attempt this right now, as that is time consuming.

One of the criteria of the app is to show good routes of where the user lives. The limitation with my program is that it will only show routes in Edinburgh, as that is where I live and know good routes. I would overcome this problem by making it possible for other users around the world to upload a route that they think is good and that way, maybe users that live in the same city can also enjoy good routes.

Interviews with potential users:
Interviewee #1: Competitive runner Cameron Sale (ex Glenalmond Student) – takes running seriously; Runs 6 days per week, sometimes twice per day.
Interviewee #2 Thomas Kingan: Regular runner; Runs 3 times a week as games option in school.
Interviewee #3 Alice Lin: Hockey player; Runs for the benefits of fitness in hockey matches.
Interviewee #4 Elsa Ma: Occasional runner; Runs once a month.

Questions asked by the interviewer:

1. "What's the reason for your running (eg Get in shape, lose fat, set a pb)?"
2. "What demotivates you the most about running?"
3. "Do you care about the locations you run at – do you like a good scenery?"
4. "Does how much you run in the workout – whether it's long run or intervals – matter to you?"
5. "Do you keep track of all your workouts? If so – how do you keep track of them?"

Key points from interview #1:

1. Get faster over the 5k and 10k distances.
2. Sometimes the toughness of the exercises, as well as the weather, can really influence his mood.
3. Cares that the route is good (ie surface, not overly populated with people) but doesn't really care about the scenery.
4. Yes, definitely. Distances and times are very important to see progress over time and do workouts dedicated exactly for him to get better
5. Yes – a word document

Key points from interview #2:

1. Likes running
2. Bad weather
3. Yea likes a gd route, clean air
4. Yea
5. Yes – uses strava

Key points from interview #3:

1. Getting in shape + set personal best
2. The initial part of actually going on a run and finding routes
3. Yes definitely; likes to change up routes
4. Yes obviously – otherwise, wouldn't be able to know her prs; likes to track her progress
5. No – but really wishes she could

Key points from interview #4:

1. Get in shape
2. Doesn't like running by herself
3. Yes – prefers scenery when runs; doesn't like a track or something like that
4. Yes
5. No

Conclusion from the interview process:
All interviewees seemed to care about the routes, whether it was for scenery or features like running a known distance, having good surface to run on, etc. Therefore, I think it is very important to really focus on that in my app, making as many good routes as possible.

Requirements

Software and hardware requirements:

Hardware:

- **A computer capable of running Android Studio 3.5**– Android Studio 3.5 is one of the latest versions and will have a lot of updates that I will use. One of those being permission setting in Maps – to ask for the user's permission to get their location.

- **An Android-operated smart phone** – to test the program is running properly, I will need an Android device. I'm going to be using my phone HUAWEI Mate 8. This is one of many phones that allow Android Studio apps to be run on.

Software:
(https://www.peerbits.com/blog/android-studio-3-5.html)

- **Android Studio 3.5** – the new Android Studio has multiple added benefits. It deals with User Interface freezes, which were present in the old versions. As well it allows minor changes instantly and test them. This platform was also chosen as it uses Java as its programming language, but the graphical user interface side is much easier to handle – no need to program every button and window, they can just be added manually and Android Studio writes the code automatically. This will allow for more time to spend on harder features, such as GPS navigation and storage of different routes. In previous encounters with the platform, the programming was made much easier by those simple functions.

- **Google Maps API key** – easily instructed process by the Android Studio website to get the API key in order to access the Google Maps Server.

- **Google Play services SDK** – also needed to run the app on an Android Device.

Success criteria:

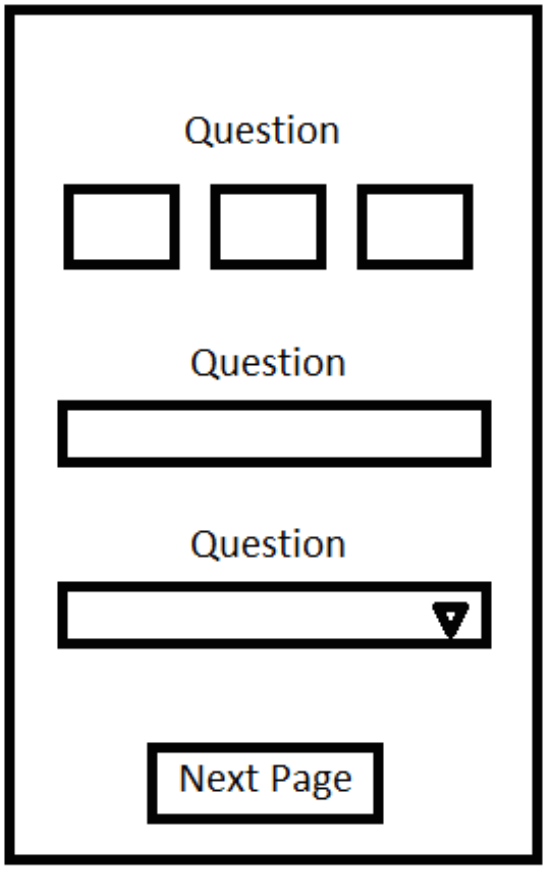
Criteria	**How to evidence**
Reasonably good and user-friendly graphical user interface.	Give to test to random people who have experience with usage of apps and get their opinion.
Easy and clear to answer questions to find out information about the user.	Clear method and space for answering straight after a question.
Have a page, which asks the user about their preferred distance to train and their personal bests for that distance.	
Saving the user's preferred distance to train and their personal bests to a text file.	Being able to read it from that file later in the program.
Being able to access the text file, which stores user's answer to the questions later in the program.	Saving a known text into the file and checking if it can output that same text from the text file.
Have a clear transition between pages. The user knows how to go both forward and back a page.	Giving the app to stakeholders and see if they think it's easy to navigate through pages.
Have a page, which asks the user whether they have a race coming up.	
Have a page, which asks the user about when their future races are.	
Saving the dates for user's future races.	Write an extra program in the background, which reads the file the races were saved to just to test it saved and deleting it after.
Have a page, which asks the user where they usually live, and train.	
Saving the destination of where the user usually lives and trains in a text document.	Being able to read it from that file later in the program.
Have side menu navigation at the main page with many other pages linked to it.	Have a typical 3-line sign in the top left corner, which when clicked, opens up to the right.
Have a page, which outputs the user's workout history.	Do a few fake workouts and screen shot the history page afterwards.
Have a page, which outputs user's achievements: longest distance/longest time/fastest average speed ran.	After the first run, since it's the only run, the achievements page must have the statistics from that run as achievements. Maybe go for a second (longer/faster) run after that and see if the achievements page changes.
Have a page, which allows manual entry of workouts.	Manual entry will automatically add a run to the workout history page.
Use GPS to track routes and work out distances ran during the workout.	Go on the same run twice, once with my program, and once with a tested program Strava/Runkeeper and compare results after.
Use time measurements to work out and display paces and times.	1. Go on the same run twice, once with my program, once with Strava/Runkeeper and compare results after.

	Look whether times and paces are displayed properly.
Clearly show good routes to the user.	2. Give the app to one of the interviewees and see whether they understand the routes suggested to them.
Suggest good workouts to the user to get to their goal faster	Screenshot of the workout suggestion window

Design

User Interface Design:

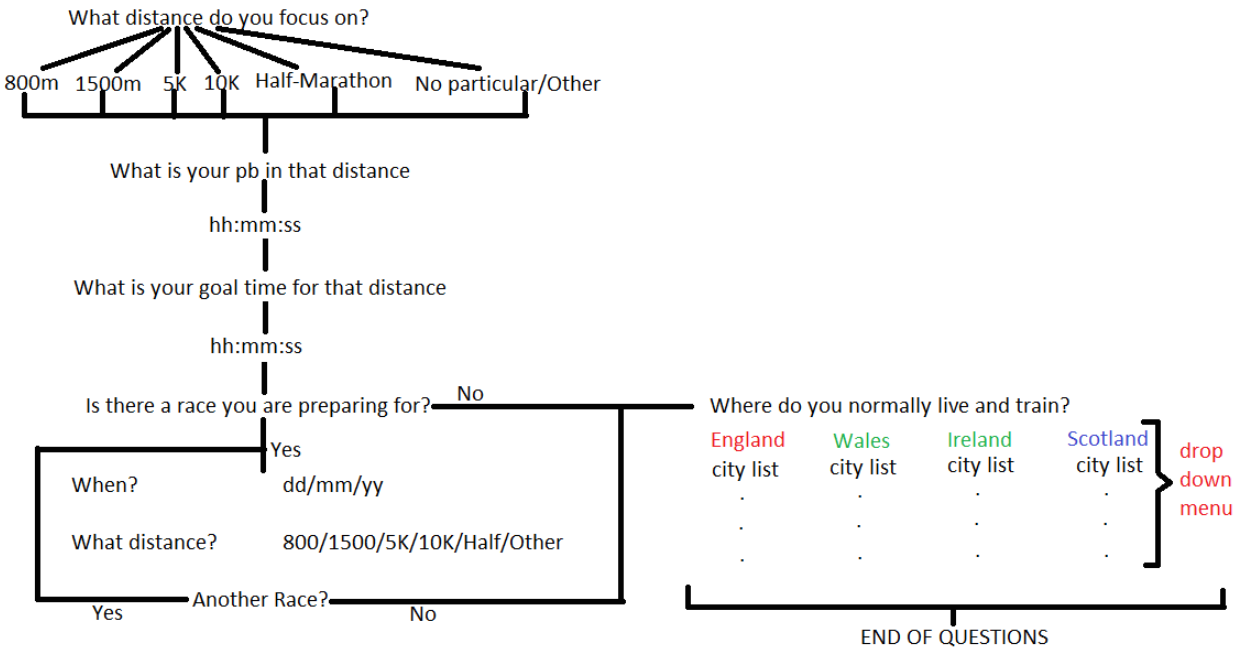
Upon opening, the app will have a few pages at the start, which ask the user about themselves: the distance they run, their personal best's, where they live and train and whether they have any races they are preparing for.



Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Saving the answer to those questions above to a text file.
- Reasonably good and user-friendly graphical user interface.

The questions that are to be asked about the user:



First Page:
First page will have the first three questions: about the distance, the best time and the goal time. For the distance question, I decided to do them as buttons. That avoids any entries that don't fit the criteria, which could be present in a "write down" answer.

Both the questions about the times will be a "write down" answer. I think that will be easier for the user to enter and it's not very hard for the program to check the validity of the times. The hours slot should not be bigger than 1 for the 800, 1500, 5km and 10km distances or bigger than 2 for the half-marathon. The minutes and seconds slots should not be bigger than 60. There is a possibility that the person doesn't yet have a personal best in that distance so no value will be accepted as well.

At the bottom of the page, there will be a "submit" button, which – when pressed – takes in all the values inserted, checks for errors and saves the values to a text file created. Then goes to the **second page**.

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Saving the user's preferred distance to train and their personal bests to a text file.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Reasonably good and user-friendly graphical user interface.

Possible Pseudocode Algorithm:

```
boolean eight;
boolean fifteen;
boolean fiveK;
boolean tenK;
boolean HALF;
boolean no;

FUNCTION eightPressed(){ ##if 800m button is pressed
     Eight=true;
     Fifteen=false;
     fiveK=false;
     tenK=false;
     HALF=false;
}

##do the same for the other distances

FUNCTION submitPressed(){ ##when submit button is pressed
     if (!eight AND !fifteen AND !fiveK AND !tenK AND !HALF AND
!no){
          print ("Gotta choose something");
     }
     else{
          if (eight){
               if (hoursEntered != 0 OR minutesEntered>60 OR
               secondsEntered>60)
               {
                    print ("Typed in something wrong");
               }
               else{
                    saveToFile("800", "questions.txt")
                    saveToFile(hoursEntered + minutesEntered +
                              secondsEntered, "questions.txt);
                    openNewPage(secondPage);
               }
          }

          ##do the same for other distances
     }
}
```

Key Variables	Data Type	How it is used
eight/fifteen/fiveK/tenK/ HALF/no	Boolean	Used to help the program understand which button the user has pressed and therefore know which distance the user is training and what to save to the text file.
hoursEntered/minutesEntered/secondsEntered	Integer	Holds the values for the user's personal best times over the distance chosen earlier.

Second Page:
Second page is only going to have one question: "Is there a race you are preparing for?" with two possible answers: Yes or No. Those will also be set up as buttons to avoid any confusion to the user. An alternative approach is to have a drop down menu, but in my opinion the buttons look better.

At the bottom there will once again be a "submit" button. If answered yes, pressing submit will take the user to the **third page** and if answered no, to the **fourth page**.

Links to the success criteria:

- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Reasonably good and user-friendly graphical user interface.
- Easy and clear to answer questions to find out information about the user.

Third Page (If answered yes):
Since in the **second page**, the user said that there is a race they are preparing for, this page will ask about the race: date and distance. The date will be entered manually into three different boxes: "day", "month" and "year". The entry for "day" should not exceed 31 and for "month" – 12. There isn't really a limit for the year, because the race could happen any year, but I'm thinking of putting a limit of 10 years into the future. The date obviously can't be before the present day's date as well. To additionally bullet prove the system, certain months can only have certain amounts of days (December: 1-31; April: 1-30; February: 1-29). To avoid any word written answers, the type-in field should be set to integer only (ie when clicked on it, only numbers appear on the keyboard of the phone).

Under the date, it will ask about what distance the race is. This will once again be buttons and will follow the same procedure as the buttons in the **first page**.

At the bottom of the page will be another question: "Is there another race you are preparing for?". This will have two possible answers: Yes or No, also in the button form. If the user presses yes, it will take them to the same page that they are on to enter the details of another race. If no was pressed, it goes to the **fourth page**. The pressing of any button, will save the details for the race to the same document created in the **first page**.

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Saving the dates for user's future races.

Pseudocode Algorithm:

##Use same code as in first page for the buttons

```
FUNCTION noPressed(){ ##If "NO" button is pressed
     if (!eight AND !fifteen AND !fiveK AND !tenK AND !HALF AND
     !no){
          print ("Gotta choose a distance");
     }
     else{
          if (dateEntered<todaysDate){ ##if date entered is
                                       before current date
               print ("This day is already behind us");
          }
          else if (dayEntered<28 OR dayEntered>31 OR
               monthEntered<1 OR monthEntered>12){
               print ("Wrong entry details");
          }
          else{
               if (eight){
                    saveToFile(date + "800m", "future_races.txt");
                    openNewPage(FourthPage);
               }
               ##do the same for other distances
          }
     }
}

##do the same for FUNCTION yesPressed(){…}
```

Key Variables	Data Type	How it is used
eight/fifteen/fiveK/tenK/ HALF/no	Boolean	Used to help the program understand which button the user has pressed and therefore know which distance the user has their race in. and knows what to save to the text file
dayEntered/monthEntered/yearEntered	Integer	Holds the values for the date the user has their race on.
currentDate	Date	Current Date – used in testing for validity of entry.

Fourth Page (If answered no):
In this page, the user is asked about the place they normally live and train in. This section is used for the route suggestion talked about in the success criteria. I am thinking of only having countries and cities across the UK. There will be two drop down menus: one for the country and one for the city. To avoid unreal entry information, I think the city drop down options have to be affected by what the user chooses for the "country" option. So for example, if the user clicks Scotland in the country drop down menu, then the city drop down menu will only have cities of Scotland and not England, Wales or Ireland. This avoids entries like "England, Edinburgh". This also saves time for the user having to scroll down through the whole list trying to find their city.

At the bottom of the page, there will once again be a "submit" button. Unlike the previous submit buttons, when pressing this one, there will be no checking done about the information entered, as that all would have been done before pressing the button. This page is the last page where the questions are asked so pressing the submit button will take the user to the "**Training Space" Page.**

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Saving the destination of where the user usually lives and trains in a text document.

```
list countries = ["England", "Scotland", "Ireland", "Wales"];
list cities = […];
list subcities = [];

dropDown1 = new dropDown(countries);
dropDown2 = new dropDown(cities);

if (dropDown1=="England"){
     for i in cities.length(){
          if (i<n){ ##n=number of cities in England that are on
                    the list. Note: it's only i<n cause English
                    cities are first in the cities list. If it
                    was Scotland, it would be: a<i<b, where a=n+1
                    and b=n+N (where N is number of cities in
                    Scotland) cause Scotland is second on the
                    list

               subcities[i]=cities[i];
          }
     }
     dropDown2.set(subcities);
}

##do the same for other countries
```

Key Variables	Data Type	How it is used
countries	List	Used to store the list of countries in the UK for the user to choose from in the drop down menu.
cities	List	Used to store the list of cities in the UK for the user to choose from in the drop down menu.
subcities	List	Used to store the list of cities which exist in the country the user has chosen from countries drop down menu.

“Training Space” Page Design:

The page will look like this:

Side Menu Navigation:
As talked about in the analysis, when looking at existing solutions, they made a good job of helping the user navigate through the app by using a navigation panel. I decided to go with the side menu navigation because there are a few extra pages that I have to put and that would be rather hard to fit them all in at the bottom. The side navigation will include the following pages: **Workouts, Routes, History, Achievements,** and **Manual Entry**. Upon pressing any of the links, it will take the user to a new page.

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Have side menu navigation at the main page with many other pages linked to it.

1. **Workouts Page:**
   This page is used to output suggested workouts to the user depending on the distance they said they are focusing on at the start of the “questionnaire”, when the app was first opened.

The program achieves that by reading through the text file, which was created at the start. The program then outputs the contents of a relevant text document according to the distance chosen by the user, which has suggested workouts to better improve at that distance. Those text files are pre prepared by me. In there, I've put about 4 different workouts, which – from research and personal experience – help athletes get better at that certain distance.

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Suggest good workouts to the user to get to their goal faster.

Pseudocode Algorithm:

```
FUNCTION workoutsSuggestions(){
     if (readFirstLine("questions.txt")=="800m"){
          outputDocument("800trainingPlan.txt");
          ##Those training plans are preprepared
     }

     ##do other else ifs for other distances as well
}


PROCEDURE String readFirstLine(File file){
     for i in line(){
          firstLine=readFile(file);
          return firstLine;
          break;
     }
}
```

2. **Routes Page:**
   This page will show the good routes to run on and train on to the user. As talked about in my analysis, the limitation to this is that it will only show routes in Edinburgh. So the first thing this page will do is read through the document where it was saved where the user lives and trains most of the time. If that place happens to be Edinburgh, then the page outputs the contents of another text document, where I once again pre prepared the pictures of the routes and a brief description so the user doesn't get lost.

   I'm not sure if android studio is fine with outputting pictures from a text document, but I will have to come back to this when I'm actually coding this out.

   Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Clearly show good routes to the user.

The algorithm for this will also be using the `readFirstLine(File file)` procedure from workouts page, but this time, it will be reading from the document where the location of the user's training and living was saved.

3. **History Page:**
   Every run will have stats about the run: things like distance ran, time ran, and average pace. Those stats will be saved in a text file after each run (I will talk about this later in this section), along with the date and then when the **history page** is opened, the stats of every run ever done will be outputted on the screen. This way the user can track their progress and this also adds a convenience factor, as all the workouts will be in the same place.

   Links to the success criteria:
   - Easy and clear to answer questions to find out information about the user.
   - Reasonably good and user-friendly graphical user interface.
   - Have a clear transition between pages. The user knows how to go both forward and back a page.
   - Have a page, which outputs the user's workout history.

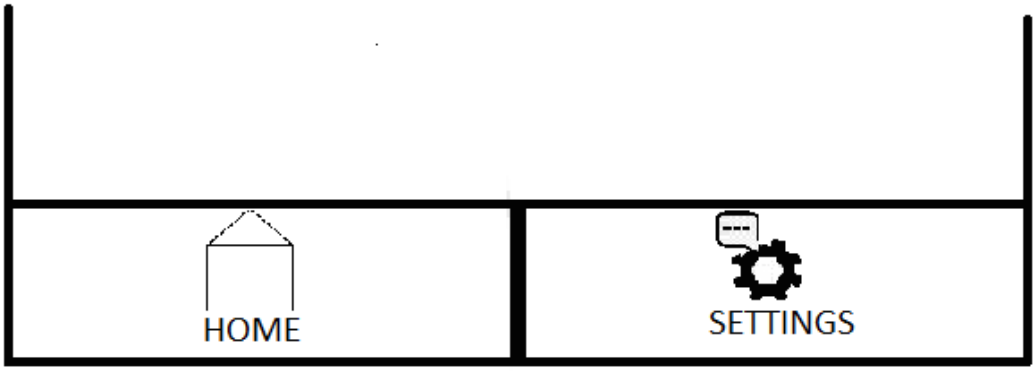
4. **Achievements Page:**
   This is quite closely linked to the things I was talking about in the **history page** section. After the run, along with saving the stats to the history text document, the program will also go through the achievements text document and check if one or more of the following categories have been bettered: furthest distance ran, longest time on feet, and fastest average pace. If the user has gotten a new achievement, the information gets updated. The user can see all of their achievements on the **achievements page**. This in a way promotes competition – against yourself and hopefully gets the user to train more.

   Links to the success criteria:
   - Easy and clear to answer questions to find out information about the user.
   - Reasonably good and user-friendly graphical user interface.
   - Have a clear transition between pages. The user knows how to go both forward and back a page.
   - Have a page, which outputs user's achievements: longest distance/longest time/fastest average speed ran.

5. **Manual Entry Page:**
   This page will allow the user to enter the details of the workout they've just done manually. This will ask the user for the distance ran and the time they ran it in and the program will calculate the average pace itself. Either of the entries can be left blank but not both (because the user might not know what distance they ran – only timed themselves). Any run entered in the **manual entry page** will automatically go to the **workout history page**. It will take a simple text entry for the distance and time. Distance will be asked in terms of kilometres and time will have a hh:mm:ss format. Any manual entry can also be put down as a new achievement if it beats any of the old achievements.

   Links to the success criteria:
   - Easy and clear to answer questions to find out information about the user.
   - Reasonably good and user-friendly graphical user interface.
   - Have a clear transition between pages. The user knows how to go both forward and back a page.
   - Have a page, which allows manual entry of workouts.

Each of those page will also have bottom navigation, which can get the user back to the **"Training Space" Page.** The bottom navigation looks like this:



Links to the success criteria:
- Easy and clear to answer questions to find out information about the user.
- Have a clear transition between pages. The user knows how to go both forward and back a page.

“Start Workout” Page:
When the user presses “START WORKOUT” on the **“Training Space” Page**, the following page will show up:

This is the page that calculates the running statistics (distance and time), as the user is running. The route that the user is running will be highlighted as they are running. This makes it easier to visualize the distance ran as well as helps the user in the event of getting lost to retrace the route back home.

Links to the success criteria:

- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Use GPS to track routes and work out distances ran during the workout.
- Use time measurements to work out and display paces and times.

Pseudocode Algorithm:

```
##global variables:
public Location currentLocation = getCurrentLocation();
public totalDistance = 0;
public totalTime;

FUNCTION onMapOpened(){
        startStopwatch(); ##starts the stopwatch needed to count time ran
        showOnMap(currentLocation); ##shows current location on the map

        repeatFunction(continuous, time: 2s); ##repeats function "continuous" every 2sec

}

FUNCTION continuous(){
        newLocation = getCurrentLocation();
        showOnMap(newLocation);
        totalDistance = totalDistance + distanceBetPoints(currentLocation, newLocation);
        ##function distanceBetPoints(Location location1, Location location2) calculates
distance between location1 and location2

        currentLocation = newLocation;

}

FUNCTION onStopWotkoutPressed(){ ##when "stop workout" is pressed
        stopStopwatch();
        totalTime = stopwatchValue();
}

PROCEDURE Double getTime(){
        return totalTime;
}

PROCEDURE Double getDistance(){
        return totalDistance;
}
```

The getTime() and getDistance() functions is something I have learnt from Object Orientated Programming. Because next page is going to be outputting the stats that the user has just ran, I need a public method which will allow me to access distance and time from other pages.

Key Variables	Data Type	How it is used
currentLocation	Location	The current location of the user. It is used in two ways: 1. Plot the current location of user on the map. 2. Use it to calculate distance to newLocation.
newLocation	Location	New Location of user is requested every 2 seconds. Used to update the location of user on the map, as well as calculate the distance from previous recorder location.
totalDistance	Double	Shows the user how long (distance wise) their run was.
totalTime	Double	Shows the user how long (time wise) their run was.

Running Stats Page (When "STOP WORKOUT" is clicked):
This will be a pretty simple page, which just summarizes the run in numbers. The distance and time ran are the things that were calculated in the **"Start Workout" Page** and the program uses the two to calculate average pace. All three will be displayed on the **Running Stats Page**.

As briefly talked about previously, two other processes are going to happen in the background. The first one is that the program is going to save those statistics, along with the date, to the workout history text document. This document is used to show the user's workout history in the **history page.**

The second process going on in the background is the checking of whether a new achievement has been accomplished. The program is going to go through the achievements text document and check whether the distance just ran by the user is longer than the longest distance saved in the document, or whether the time just ran by the user is longer than the longest time saved in the document, or whether the average pace just ran by the user is faster than fastest average pace saved in the document. If they are, the document is rewritten with the new values inserted. If not, the document stays the same. This document is used to show the user's achievements in the **achievements page.**

At the bottom of the page, there will be a button saying "RETURN". This button takes the user back to the **"Training Space" Page**.

Links to the success criteria:

- Easy and clear to answer questions to find out information about the user.
- Saving the user's preferred distance to train and their personal bests to a text file.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Use time measurements to work out and display paces and times.

```
FUNCTION outputStats(){
     time = startWorkout.getTime();
     distance = startWorkout.getDistance();

     speedInKPH = (distance/1000)/(time/3600); ##sped in km/h

     speedInMPK = 60/speedInKPH; ##speed in min/km
}
```

Stakeholder Input

To make sure the targeted audience is approving my approach to the solution, I have sent the following email.

"Dear **NAME,**

Hope you are well.

Please find attached the approach that I took to design the app I was interviewing you about. One of the main points that I was focusing on is easy navigation through the app so the user doesn't struggle finding any pages they desire. I also made sure the app has a way of remembering all the workouts done by the user and can be accessed at any point, as well as to add motivation to the user, the app has an achievements page, where the user's furthest distance, longest time and fastest average speed ran is shown.

Please let me know what you think about this design and any suggestions on improving it.

Thank you.

Kind regards,
Gleb"

Here are their replies:
Cameron Sale:
"Hey mate,
It's looking good to me. Seems like a hard coding job for you though ☺.
I'm not sure if you are going to do this but if I had to suggest one thing, I'd recommend adding some colour. Black and white seems a tad bit boring.

Good luck."

Conclusion:
Interesting point raised by Cameron. Black and white is definitely not the colours that attract the user to go on the app again. I think I'm going to mix in some red to make it look better.

Alice Lin:
"Dear Gleb Sokolovskyi,

Thank you I am well and I hope you are too.
The diagrams are easy to read and the navigation is simple, I also love the motivation features you plan to add.
I currently have no notes but please do keep me updated!

Best Wishes,
Alice
"

Conclusion:
Alice is the only one of my friends who can write emails properly.

Elsa Ma;
“ Think the design is really good and clear.
As I am not a very regular runner, I think that having a notification or pop up message to encourage people like me to increase the number of times they run a week.
Other than that I think it is great.
“

Conclusion:
That’s something to definitely consider have I had more time for this project. As the problem I’m trying to solve with this project is demotivation to exercise and regular pop up notifications could get the user to visit the app more and therefore exercise more.

Input	Process	Output
Pressing Buttons	Changing of colour of buttons and recognition of which button has been selected by the user.	The button that was pressed goes dark grey, the other – stay grey. And the Boolean variable associated with the button pressed becomes true while other become false.
Inputting Time	Editing text in a text box	Inputting time for hours, minutes and seconds as integers within a given range (eg 1-60 for minutes) into a type in box.
Inputting Date	Editing text in a text box	Inputting day/month/year as integers within a given range (eg. 1-31 for days) into a type in box.
Drop Down Menu	Multiple Choice Selection	Options given in the “cities” drop down menu are affected by the choice made in the “countries” drop down menu.
Change In Location	Location Change Algorithm	Every 2 seconds, the location of the user is updated along with the map showing the location of the user.

Testing methods

Throughout the whole coding stage, each function should be tested for functionality. To further test the robustness of the program, multiple pieces of data will be designed to fully test every possible input and output. The good thing about using Android Studio to create apps is that the outcome is pretty resilient to error. Try{…}catch{…} technique is popular among crash prevention. I'll be looking to using that a lot.

The testing during development process should include data input to the program (including testing of map) and the result, and if the result is what was intended before running. Any error messages should be shown.

Interface inputs and test tables:

Multiple Choice Buttons

Buttons are not very hard to test: they either work or they don't. Here is what I think functional and "successful" buttons should do:

1. When a button is pressed, it goes darker in colour to help the user understand that the software has picked up on the fact that they have pressed the button.
2. If another button is pressed after, that button goes darker in colour and the previously pressed button goes lighter (same colour as other non-pressed buttons around).
3. Signify the program the information that is related to this button getting pressed.

Submit Buttons

There are three main processes that those type of buttons undergo:

1. Checks for validity of the data entered by the user on the page. If there are any invalid entries, makes a message to the user to re-evaluate the data entered.
2. If required, saves that data to a text file.
3. Takes the user to the next page.

Type-in boxes: Time entry

These are where the user types in their personal best and their goal time for their preferred distance. There are multiple validity checks for this one:

1. No decimals for "hour" and "minute" boxes.
2. No text for any of the boxes – just numbers.
3. No symbols for any of the boxes (except "." For the "seconds" box) – only numbers.
4. All numbers have to within a valid range: "seconds" and "minutes" should be 0<=input<60.
5. Time inputted has to be within a realistic range (e.g. user's personal best for the 5k can't be 5minutes, nor can it be any longer than an hour).

Following those restrictions, I have created a set of values to input into the boxes to check for validity:

Input	**Accept/Reject**	**Result**
0**hours** 15**minutes** 50**seconds** (5k entry)	Accept	
0**hours** 1**minute** 59.89**seconds** (800m entry)	Accept	
Hello **hours**	Reject	
3.14 **hours** 15**minutes** 16**seconds**	Reject	
Paste in *If-* by Rudyard Kipling into **hours/minutes/seconds**	Reject	
Paste in the first 100 digits of pi into **hours/minutes/seconds**	Reject	

Type-in boxes: Date

This is where the user enters the date of their next race. Once again there are limitations as to what can be entered:

1. Acceptable Ranges: 27<day<32 && 0<month<13 && 2019<year<2029 (I gave it a 10 year expiry because it's unrealistic someone would know when their race is going to be in 10 years).
2. Data entered can't be before today's date.
3. Decimals can't be entered into any box.
4. Symbols can't be entered into any box.

Following those restrictions, I have created a set of values to input into the boxes to check for validity:

Input	**Accept/Reject**	**Result**
dd: 17; **mm:** 12; **yyyy:** 2021	Accept	
The day's date	Reject	
dd: 17; **mm:** 12; **yyyy:** 1969	Reject	
dd: 33	Reject	
Paste in *If-* by Rudyard Kipling into **dd/mm/yyyy**	Reject	
Paste in the first 100 digits of pi into **dd/mm/yyyy**	Reject	

GPS tracking

That's when the user does their workout and the app tracks their route, distance ran, time ran and average pace. There aren't any specific tests for this, except for the accuracy test: take 2 devices: one with my app and the other with strava; do the workout while recording on both devices; stop the workout at the same time; compare the accuracy of the app.

Action to Test	Does it work? Y/N
Check the functionality of each type in box – make sure it only takes in correct and reasonable values.	
Pressing “SUBMIT” button (or other buttons with similar functionality) checks the validity of the information entered by the user and takes the user to the next page.	
Saving the information that the user entered into a text file, including dates, distances and locations.	
Being able to access the text file, which stores user’s answer to the questions later in the program.	
Ability to go both forward and back a page.	
Have side menu navigation at the main page with many other pages linked to it.	
Pressing any option from the side menu navigation takes the user to different pages, specifically assigned to the user’s choice.	
Pressing “START WORKOUT” button takes the user to the GPS window.	
Use GPS to track routes and work out distances ran during the workout.	
Use time measurements to work out and display paces and times.	
Pressing “STOP WOTKOUT” button takes the user to a page, which outputs the statistics about the user’s run.	
Saving those statistics to the history archive text document, along with the date.	
Checking those statistics against the achievements and replace the achievements if the user has bettered their previous personal best.	

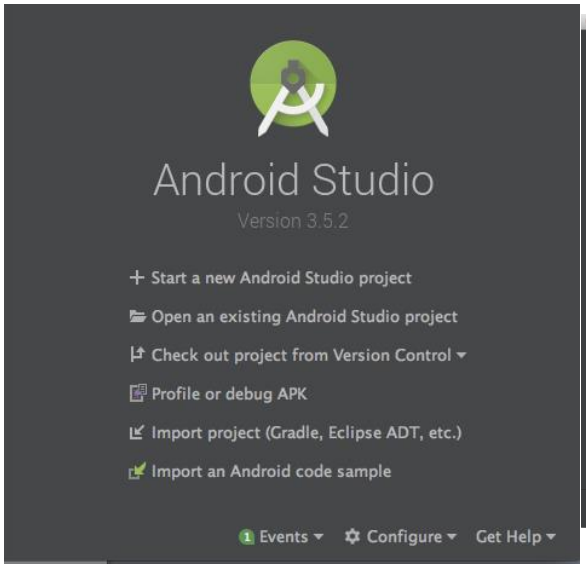
Development and Testing

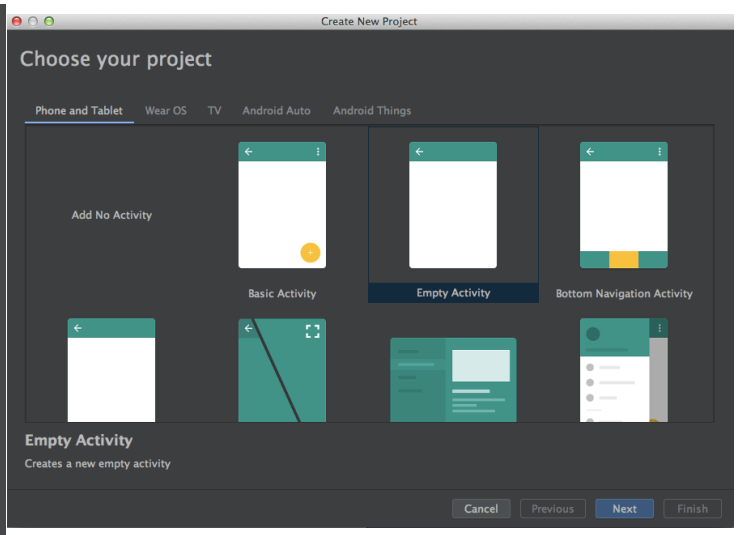
Stage 1: The Questionnaire

First Page:

As mentioned above, the first thing the user is going to see when they open the app is a set of questions, which helps the app get to know specific detail about the user. Since I've decided to use Android Studio, each page will have an "activity". Each activity has two (or more) files designed for it: the XML file and the JAVA file. There was a possibility to code in Kotlin instead of Java. I went with Java because I have previous experience with Java and feel more comfortable reading and writing code in it.

Upon opening Android Studio, all I had to do is start a new project and choose an empty activity:





The XML file was empty and the Java equivalent only had a command, which made the program open a blank page upon running it:

```
public class MainActivity extends AppCompatActivity {
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
}
```

First upon trying to add buttons to the page, I would drag the buttons onto the page and place them as shown in the diagrams in the design section. I got a sort of error saying that those positions are not fixed and when the program is going to run, they will all relocate to position (0,0) – top left hand corner I presume. Therefore I looked at approaches, which made those positions fixed. I didn't find one where I can just specify the location/coordinates and the button would go there. Most solutions included use of *layouts.* I used *Linear Layouts.* Those layouts can have different orientations: *vertical* and *horizontal*

– the names are pretty self-explanatory. There can be layouts inside different layouts and you can have as many layouts as you want. Having any component (including layouts, buttons, text fields, etc.) inside the layout and specifying the height and width, as “match_perent” will take up the whole layout. Therefore having more than one component inside the layout all with “match_parent” height and width will split the height and width into equal segments. Therefore having multiple components with different height or width inside the same layout would mean changing their “weight” – could be seen as their importance.

So, for the **first page/activity,** I started off by creating one big linear layout on the outside and inside of it were 5 linear layouts (because there are 3 questions on the **first page** + a welcome message + a submit button). The code for that:

```
<LinearLayout
   android:id="@+id/overall_layout"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_margin="10dp"
   android:layout_marginBottom="20dp"
   android:orientation="vertical"
   app:layout_constraintTop_toTopOf="parent"
   tools:layout_editor_absoluteX="1dp">

   <LinearLayout
      android:id="@+id/welcome_message_layout"
      android:layout_width="match_parent"
      android:layout_height="80dp"
      android:layout_margin="10dp"
      android:orientation="vertical"
      app:layout_constraintTop_toTopOf="parent"
      tools:layout_editor_absoluteX="1dp">
    </LinearLayout>

   <LinearLayout
      android:id="@+id/question1"
      android:layout_width="match_parent"
      android:layout_height="80dp"
      android:layout_margin="10dp"
      android:orientation="vertical"
      app:layout_constraintTop_toTopOf="parent"
      tools:layout_editor_absoluteX="1dp">
   </LinearLayout>

   <LinearLayout
      android:id="@+id/question2"
      android:layout_width="match_parent"
      android:layout_height="80dp"
      android:layout_margin="10dp"
      android:layout_marginTop="20dp"
      android:orientation="vertical">
   </LinearLayout>
```

```
    <LinearLayout
        android:id="@+id/question3"
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:layout_margin="10dp"
        android:layout_marginTop="20dp"
        android:orientation="vertical">
    </LinearLayout>

    <LinearLayout
        android:id="@+id/submit_button_layout"
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_margin="10dp"
        android:gravity="top|center_horizontal"
        android:orientation="vertical">
    </LinearLayout>

</LinearLayout>
```

This code creates invisible guide lines in the page. The next stage is to fill those boundaries with actual buttons and other components:

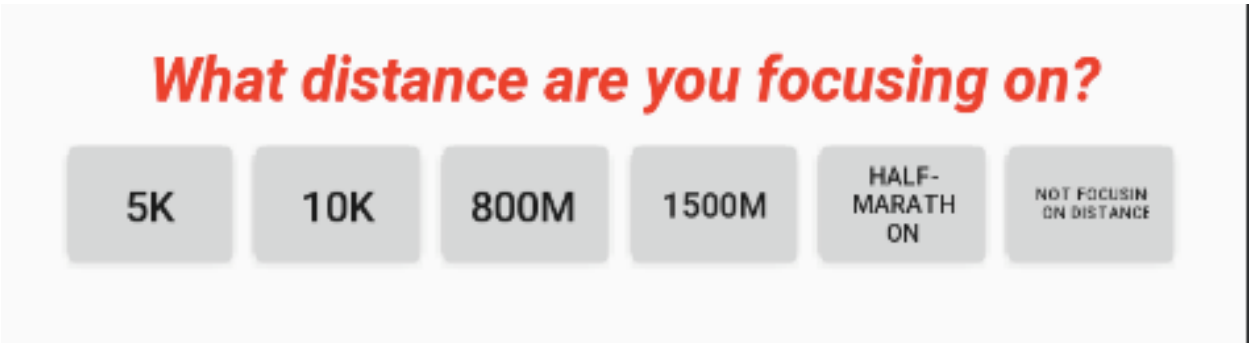
Here is the code for the first question, where the user is asked what distance they focus on:

```
<LinearLayout
    android:id="@+id/question1"
    android:layout_width="match_parent"
    android:layout_height="80dp"
    android:layout_margin="10dp"
    android:orientation="vertical"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteX="1dp">
    <TextView
        android:id="@+id/question1_text"
        android:layout_width="match_parent"
        android:layout_height="30dp"
        android:textAlignment="center"
        android:text="What distance are you focusing on?"
        android:textStyle="bold|italic"
        android:textColor="#f00"
        android:textSize="20dp" />

    <LinearLayout
        android:id="@+id/question1_buttons_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center_horizontal|center_vertical"
        android:orientation="horizontal">
```

```
    <Button
        android:id="@+id/question1_button1"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="5k"
        android:onClick="button1Pressed"/>

    <Button
        android:id="@+id/question1_button2"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="10k"
        android:onClick="button2Pressed"/>

    <Button
        android:id="@+id/question1_button3"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="800m"
        android:onClick="button3Pressed"/>

    <Button
        android:id="@+id/question1_button4"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="1500m"
        android:textSize="11dp"
        android:onClick="button4Pressed"/>

    <Button
        android:id="@+id/question1_button5"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="Half-Marathon"
        android:textSize="8dp"
        android:onClick="button5Pressed"/>

    <Button
        android:id="@+id/question1_button6"
        android:layout_width="50dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="Not focusing on distance"
        android:textSize="5dp"
        android:onClick="button6Pressed"/>

</LinearLayout>
```

```
</LinearLayout>
```

As can be seen from the code, the *question_1* layout is split into 2: a "TextView" and another layout. The TextView outputs the question to the user, and the other layout is used to display the buttons. The buttons are going to be displayed from left to right in horizontal fashion because I specified that the layout is horizontal. The buttons will all take up exactly the same width and height, as I have added a weight of 1 to all of them. Here is how this section looks in the visual design:



As can also be seen from the code each button has "android:onClick" function in them. This is used to tell the program what to do once the button has been pressed. This is dealt with in the Java section of the code.

In the onCreate section of the code, I've added the following code:

```
Button button = (Button) findViewById(R.id.question1_button1);
Button button2 = (Button) findViewById(R.id.question1_button2);
Button button3 = (Button) findViewById(R.id.question1_button3);
Button button4 = (Button) findViewById(R.id.question1_button4);
Button button5 = (Button) findViewById(R.id.question1_button5);
Button button6 = (Button) findViewById(R.id.question1_button6);
```

This identifies each button uniquely and I can now use those new variables created to make the page more user friendly. And the first thing I did to make it more user friendly was to make the user can see which button they have selected. I did in the following way: first I set each button to be grey:

```
button.setBackgroundColor(Color.GRAY);
button2.setBackgroundColor(Color.GRAY);
button3.setBackgroundColor(Color.GRAY);
button4.setBackgroundColor(Color.GRAY);
button5.setBackgroundColor(Color.GRAY);
button6.setBackgroundColor(Color.GRAY);
```

then whenever a button was pressed, that button would go dark grey and they others would stay grey:

```
public void button1Pressed(View v){
    button.setBackgroundColor(Color.DKGRAY);
    button2.setBackgroundColor(Color.GRAY);
    button3.setBackgroundColor(Color.GRAY);
```

```
    button4.setBackgroundColor(Color.GRAY);
    button5.setBackgroundColor(Color.GRAY);
    button6.setBackgroundColor(Color.GRAY);
}
```

The reason I didn't just make the button that was pressed, dark grey and not touch the other buttons is because when I ran it for the first time, I realized that if there was a button pressed before, the pressing of another button will make the new button dark grey but keep the old button dark grey as well, which is confusing to the user.

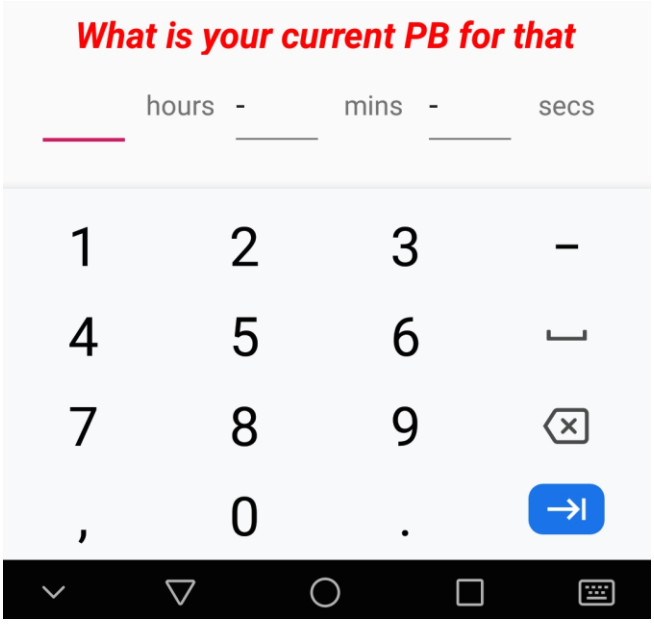
Another aspect that I added was four global (Boolean) variables: *fiveK, tenK, eight, fifteen, half* and *no.* I defined them all as "public static boolean" so they can be changed later. And they are changed once the relevant button has been pressed (ie the equivalent Boolean to the button pressed becomes true and all the others – false).

Those Booleans are all used for when the "submit" button at the bottom gets pressed and the equivalent distance to the Boolean that is true gets saved to the text file (saving of information in text files is talked about in detail in **Section 2: Saving User Inputs to Text Files**).

Other aspects of the **first page**, which are needed to be talked about, are questions 2 and 3. The questions are similar and involve the user to input their times: their personal best and their goal time for the distance they chose as buttons. They both follow the same structure as the first question, but now we introduce new component in the XML file: <EditText>. Once again – like buttons – they have to be defined in the Java file and they allow the user to type in whatever they want. However to add to the robustness of the program I added the following code:

```
        android:inputType="number"
```

This means that when, on user's device, they click on the field to start typing, only the numbers keyboard pops up and they can't change that to the letters keyboard. This is how it looks like when ran:



A useful feature of Android Studio is that because I specified that only numbers could be entered, the program prevents the user from pasting in any text. That saves me a lot of time to code that out myself.
When the submit button is pressed, the following steps are done: checks a button has been pressed, checks the times entered are appropriate, saves the details into a text document called “data.txt” and opens the **second page**. The program uses if statements as well as the global Boolean statements, talked about earlier, to figure out what to save to the text file.

The code below shows how I made sure the user presses on a button:

```
public void goToSecondPage(View v) { //when submit button is pressed
if (!fiveK && !tenK && !eight && !fifteen && !half && !no) {
   Toast.makeText(this, "CHOOSE A DISTANCE MA DUDEEE",
Toast.LENGTH_SHORT).show();
}
else {
            .
            .
            .
}
}
```

The code says that if not a signle Boolean talked about earlier is true, then it tells the user to press on one of the buttons.

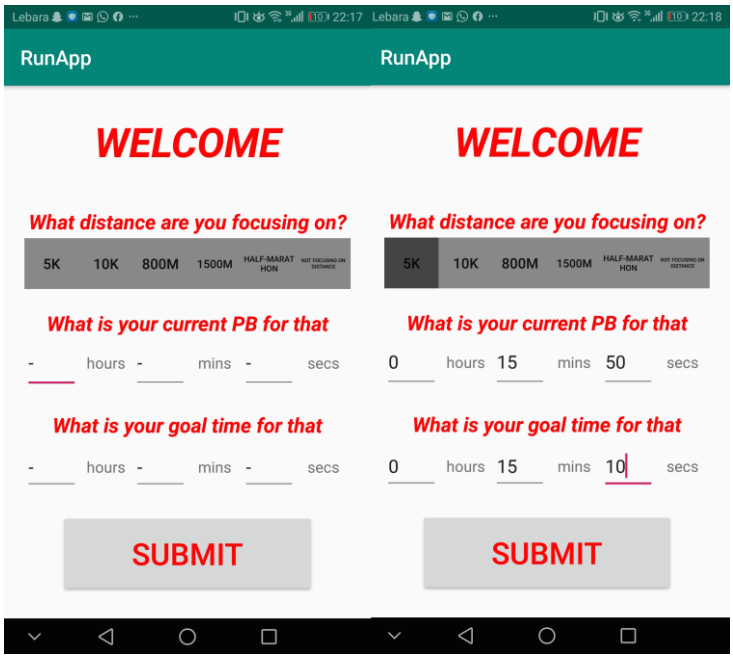
And here is the code that makes sure the times entered are of appropriate format:

```
if (fiveK) {
   if (Integer.parseInt(question2_time_hours.getText().toString()) > 1 ||
         Integer.parseInt(question2_time_minutes.getText().toString()) > 60 ||
         Integer.parseInt(question2_time_seconds.getText().toString()) > 60) {
      Toast.makeText(this, "THE DETAILS ENTERED ARE INCORREST",
Toast.LENGTH_LONG).show();
   } else {
      if (FileHelper.saveToFile("data.txt", "5K") && FileHelper.saveToFile("data.txt",
question2_time_hours.getText().toString()) && FileHelper.saveToFile("data.txt",
question2_time_minutes.getText().toString()) && FileHelper.saveToFile("data.txt",
question2_time_seconds.getText().toString())) {
         Intent intent = new Intent(this, secondPage.class);
         startActivity(intent);
         Toast.makeText(MainActivity.this, "Saved to file", Toast.LENGTH_SHORT).show();
      } else {
         Toast.makeText(MainActivity.this, "ERROR!!!", Toast.LENGTH_SHORT).show();
      }
   }
}
```

The code means that the program first checks that the hours slot was not entered above 1 (ie makes sure the user doesn’t claim that they run a 5k in 2 hours or more, because that’s unrealistically slow), as well as making sure both minutes and seconds slot has a value less than 60, otherwise it outputs “THE DETAILS ENTERED ARE INCORREST”. If however

the times were entered correctly, the program proceeds to saving those details to the text document.

So here is how the **first page** looks when ran on my phone (on the left) and what it looks like when a button is pressed and text is inputted (on the right).

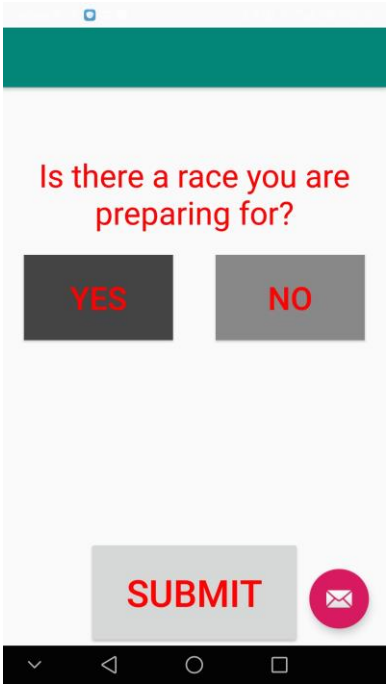


Another aspect that I've found out when running the program is that, the original layout coded out didn't fit my phone screen for some reason. For example when I ran the page above for the first time, submit button was not seen as I coded it to be at the bottom of the bottom layout but it seems that the parameters of the device in Android Studio is different to the ones of my phone. However another suggestion is because I defined the height of each layout as numbers, rather than using the "match_parent" and weight technique, so that could have been the reason.

Second Page:

Second Page is a simple yes or no question with buttons as answers. I decided to go for just two main layouts – one for the question and the "YES" and "NO" buttons, the other one for the submit button. The buttons will have the same feature as the buttons in the first page – the one that's pressed will go dark grey so the user can see which one they have chosen.

Here is how the page looks like when ran:



This time I defined each layout's height and width as "match_parent", so the submit button was in the right place the first time running.

Alignment of the text and buttons was something I had to experiment with. I saw that there were two different alignment command in XML: android:gravity and android:layout_gravity. I would try to get the "YES" and "NO" buttons to be center aligned withing their horizontal layout and the text in those buttons to be in the center so it looks better. And from playing around with it and trying out different command I've learnt to achieve this in the following way: first, when defying the layout the button is in you need to say layout_gravity="centre" for the "YES" and "NO" buttons and layout_gravity="centre_horizontal | bottom" for the submit button:

```
<LinearLayout
    android:id="@+id/second_page_submit_button_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="bottom|center_horizontal"
    android:layout_weight="1">
    .
    .
    .
</LinearLayout>
```

Then for the text inside, there are two ways: 1. Use android:textAlignment and set it to centre, or 2. Inside the button tag do android:gravity, set it to centre and the text will automatically be centred.

Fourth Page:

This is the page where the user is asked where they usually live and train. The tricky thing about this is not the actual drop down menu but the fact that decision of the first drop down menu affects the list in the second one. So if Scotland was selected in the countries drop down, then only Scottish cities should show up in the second drop down.

Firstly I did research on how to program a drop down menu. In Android Studio, they are called spinners and are defined in the XML file as such:

```
<Spinner
    android:id="@+id/page_4_country_drop_down"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:background="@android:drawable/btn_dropdown"
    android:spinnerMode="dropdown"
    />
```

In order to add values to the spinner, in the java file, I needed to firstly identify the spinner:

```
Spinner country = findViewById(R.id.page_4_country_drop_down);
```

Then I had to create a list of values, which would go into that spinner:

```
String[] countries = new String[]{"England", "Scotland", "Wales", "N Ireland"};
```

Then create an “adapter”, which is just an ArrayList and put the countries list into it:

```
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, countries);
```

And finally set the adapter of the spinner as the adapter I have just created:

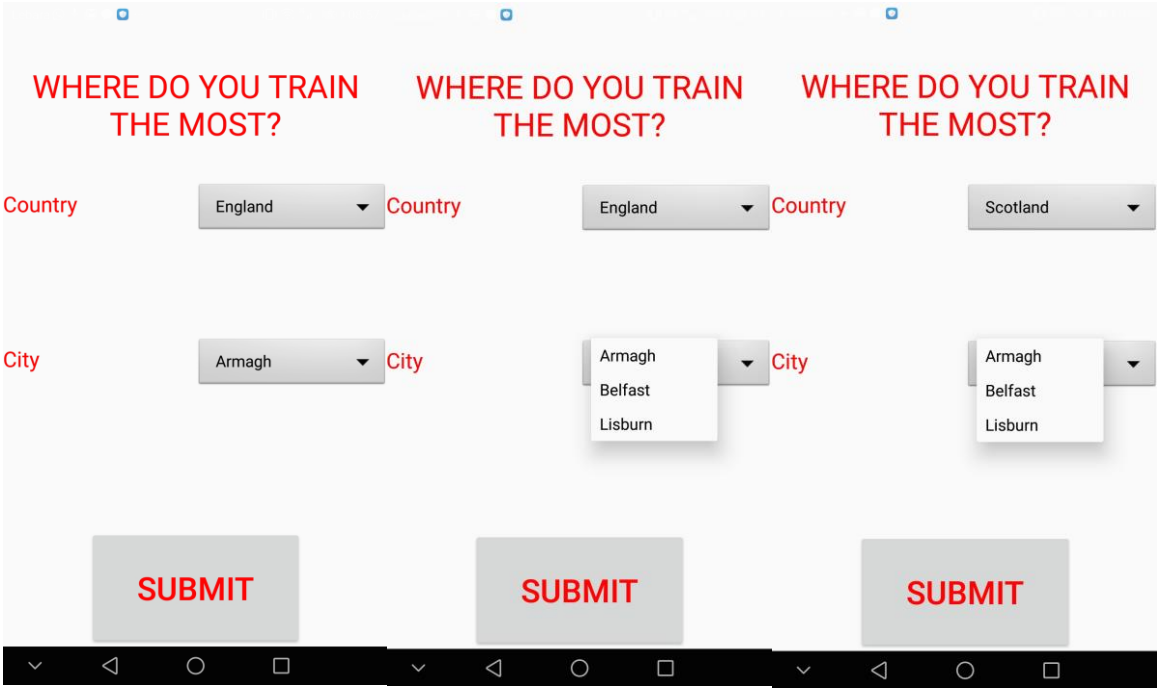
```
country.setAdapter(adapter);
```

Now in order to adjust the second spinner as specified above, I first went for this code:

```
if (country.toString().equals("England")){
    String[] cities = new String[]{"Bath", "Birmingham", "Bradford", "Brighton & Hove",
"Bristol", "Cambridge",
    "Canterbury", "Carlisle", " Chelmsford", "Chester", "Chichester", "Coventry", "Derby",
"Durham", "Ely", "Exeter",
    "Gloucester", "Hereford", "Kingston-upon-Hull", "Lancaster", "Leeds", "Leicester",
"Lichfield", "Lincoln", "Liverpool",
    "(City of) London", "Manchester", "Newcastle-upon-Tyne", "Norwich", "Nottingham",
"Oxford", "Peterborough", "Plymouth",
    "Portsmouth", "Preston", "Ripon", "Salford", "Salisbury", "Sheffield", "Southampton", "St
Albans", "Stoke-on-Trent", "Sunderland",
    "Truro", "Wakefield", "Wells", "(City of) Westminster", "Winchester", "Wolverhampton",
"Worcester", "York"};
    ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
    city.setAdapter(adapter2);
}else if (country.toString().equals("Scotland")){⋯}
```

```
else if (country.toString().equals("Wales")){···}
else{
    String[] cities = new String[]{"Armagh", "Belfast", "Lisburn"};
    ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
    city.setAdapter(adapter2);
}
```

When I ran it however, the following happened:



The leftmost screenshot is what appears when the page is first opened. To start off, it's already wrong because it is showing Irish cities when England is chosen. But I thought maybe if I choose a different country the second spinner will change, but as can be seen on the rightmost screenshot, choosing Scotland didn't change the cities. As can be seen from the code, the program is choosing the conditional statement else{..}. So if I were to change that to else if (country.toString().equals("Ireland")){...}, the second spinner would not show any city. So I looked on Stack Overflow (https://stackoverflow.com/questions/1947933/how-to-get-spinner-value) and realized that the proper way of accessing the Spinner value is: spinner.getSelectedItem().toString(). So I changed the code to that.

The submit button, just like in **first** and **third pages,** saves the details of the page into a text file.

Stage 1: Review

What has been done:

Stage 1 has mainly been about creating a graphical user interphase and asking questions about the user. Saving the answers to a text file is an important part of it but I haven't spoken about it in this stage, but I will in Stage 2. When coding this, it was important to me to have all working buttons, type in boxes, drop down menus, etc., as well as make it look fairly appealing. This was my first time properly coding in Android Studio so I tried to learn as much as possible about the structure of the platform and the basics.

How it has been tested:

Each page has been tested straight after it was programmed. Those tests however were mainly to make sure the app looks the way it should and is functional "on the surface". By that I mean that it works without me trying to break it.

Criteria met/being met:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.

Summary of the whole project as a prototype at this stage:

The program is shaping out to be the same as I imagined in the design section. This part is important to get the information about the user from the user, which can later be used in later stages of the project.

Stage 2: Saving of data into a text document

The feature and where it's seen:

Saving data into a text document is the way the program "remembers" information about the user. The program saves anything the user answers or does. This process is seen in the following activities: **first page, third page, fourth page, manual entry page and run stats page.** The program can also access information from the files and output them. This is seen in the following activities: **workouts page, routes page, history page, achievements page and run stats page.** As can be seen it appears in a lot of pages and is an important part of the functionality of the program.

Developing the feature:

I started off by creating a new folder in the source folder. I called it "instintcoder.readwritefile". And inside it I put a new java file called "FileHelper". The idea was to add public classes, which can be accessed from other activities. Mistakably, after that, I looked into reading and writing from and into files in Java. Here is the code that I used:

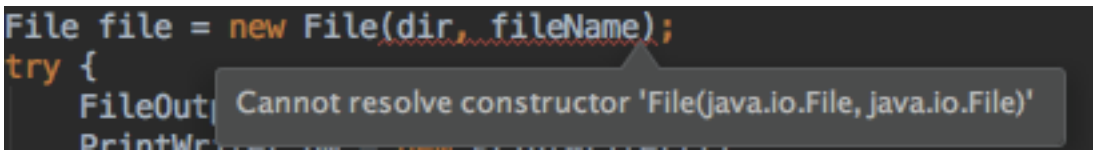
```
public static boolean saveToFile(String fileName, String data){


       boolean fileCreated = false;
try {
   new File(path).mkdir();
   File file = new File(path+ fileName);
   if (!file.exists()) {
      fileCreated = file.createNewFile("data");
   }
   FileOutputStream fileOutputStream = new FileOutputStream(file,true);
   fileOutputStream.write((data + System.getProperty("line.separator")).getBytes());

   return true;
}  catch(FileNotFoundException ex) {
   Log.d(TAG, ex.getMessage());
}  catch(IOException ex) {
   Log.d(TAG, ex.getMessage());
}
return  fileCreated;



   }
```

```
public void createNewFile(String name) {

    try {
        File myObj = new File(name + ".txt");
        if (myObj.createNewFile()) {
            System.out.println("File created: " + myObj.getName());
        } else {
            System.out.println("File already exists.");
        }
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }

}
```

The code above checks whether the file entered exists. If it doesn't, the program creates a new file. The code uses try{...} catch{...} structure to prevent the program from crashing when an error is spotted. The problem about this approach is the difference of Java and Android Studio. Java creates a new file on the computer, while Android Studio creates it on the user's phone, which is an external device, so I can't use the same code that I would use for Java in Android Studio. So the next page was not loading up and nothing was getting saved to the text document. I have changed the code to the following:

```
public void writeInternalStorage(File fileName, String textToWrite) {
    File root = android.os.Environment.getExternalStorageDirectory();
    File dir = new File (root.getAbsolutePath() + "/download");
    dir.mkdirs();
    File file = new File(dir, fileName);
    try {
        FileOutputStream f = new FileOutputStream(file);
        PrintWriter pw = new PrintWriter(f);
        pw.println(textToWrite);
        pw.flush();
        pw.close();
        f.close();
    } catch (FileNotFoundException e) {
        e.printStackTrace();
        Log.i(TAG, "******* File not found.");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

As can be seen, the purpose of this code was to write to the internal storage of the user's device. The code firstly gets the path for where the file is saved. Then uses "PrintWriter" to input the text into the file. The code once again makes use of try and catch structure. I had a problem with this code however, and that was that the line "File file = new File(die, fileName)" was getting highlighted suggesting an error. It was saying:



So I changed this line of code to the following:

```
File file = new File(dir.getAbsolutePath()+fileName);
```

That got rid of the error.

However in order to make sure the program works when ran in activities, I had to transform it to be error friendly (ie when I call FileHelper.writeInternalStorage(“”), it shouldn’t just crash. Instead it should just output on the scrren that the process was unsuccessful). So first of all I made the class from function to a procedure which return a boolean:

```
public boolean writeInternalStorage(File fileName, String textToWrite) {⋯}
```

And inside, I added a new Boolean “done” which changes to true if successful and false if unsuccessful.

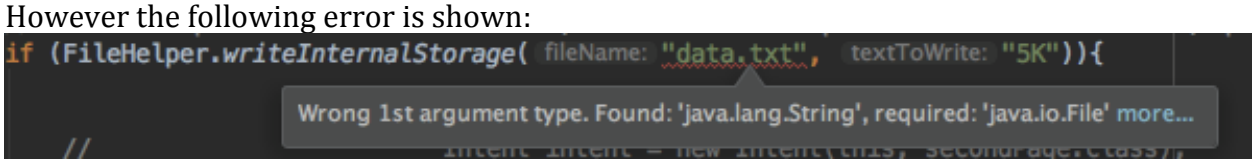
```
boolean done = false;
…
try {
    …
    done=true;
} catch (FileNotFoundException e) {
    …
    done=false;
} catch (IOException e) {
    …
    done = false;
}
return done;
```

Then I can use:

```
if (FileHelper.writeInternalStorage("data.txt", "5K")) {
    Intent intent = new Intent(this, secondPage.class);
     startActivity(intent);
    Toast.makeText(MainActivity.this, "Saved to file", Toast.LENGTH_SHORT).show();
} else {
    Toast.makeText(MainActivity.this, "ERROR!!", Toast.LENGTH_SHORT).show();
}
```

in the **first page** activity.

However the following error is shown:



So I’ve changed the code in FileHelper so that it takes in String fileN and put the following code inside the function:

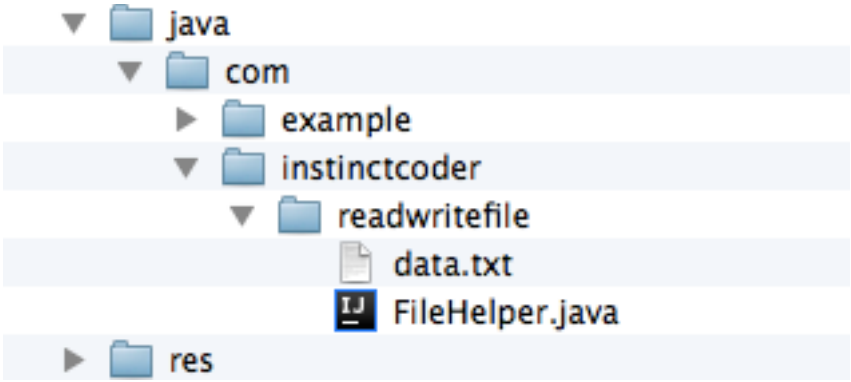
```
File fileName = new File(fileN);
```

When I ran the code, the following error message was shown:

```
W/System.err: java.io.FileNotFoundException: /storage/emulated/0/downloaddata.txt (Permission denied)
        at java.io.FileOutputStream.open0(Native Method)
        at java.io.FileOutputStream.open(FileOutputStream.java:287)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:223)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:171)
        at com.instinctcoder.readwritefile.FileHelper.writeInternalStorage(FileHelper.java:164)
        at com.example.runapp.MainActivity.goToSecondPage(MainActivity.java:71) <1 internal call>
        at androidx.appcompat.app.AppCompatViewInflater$DeclaredOnClickListener.onClick(AppCompatViewInflater.java:385)
        at android.view.View.performClick(View.java:6291)
        at android.view.View$PerformClick.run(View.java:24931)
        at android.os.Handler.handleCallback(Handler.java:808)
        at android.os.Handler.dispatchMessage(Handler.java:101)
        at android.os.Looper.loop(Looper.java:166)
        at android.app.ActivityThread.main(ActivityThread.java:7529) <1 internal call>
        at com.android.internal.os.Zygote$MethodAndArgsCaller.run(Zygote.java:245)
```

So it catches the FileNotFoundException. I've realized that I forgot to add the createNewFile() function for if the file doesn't exists. The file obviously doesn't exist when ran the program the first time. So I added the if (!file.exist()){ createNewFile(); }else{…} statements that I used in the first program. The result however was still negative.

Then I've thought of a way for the app to not create a new file on the phone. For some reason this is way complicated than anticipated. I thought that if I create an empty text file, call it "data.txt" and place it in my program's folder on my laptop. That way, when the program is ran, it doesn't have to create a file on the device and can just store it in an already existed file:



I ran the program again. Result:

```
W/System.err: java.io.FileNotFoundException: /storage/emulated/0/downloaddata.txt (Permission denied)
        at java.io.FileOutputStream.open0(Native Method)
        at java.io.FileOutputStream.open(FileOutputStream.java:287)
W/System.err:     at java.io.FileOutputStream.<init>(FileOutputStream.java:223)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:171)
        at com.instinctcoder.readwritefile.FileHelper.writeInternalStorage(FileHelper.java:164)
        at com.example.runapp.MainActivity.goToSecondPage(MainActivity.java:71) <1 internal call>
        at androidx.appcompat.app.AppCompatViewInflater$DeclaredOnClickListener.onClick(AppCompatViewInflater.java:385)
        at android.view.View.performClick(View.java:6291)
        at android.view.View$PerformClick.run(View.java:24931)
        at android.os.Handler.handleCallback(Handler.java:808)
        at android.os.Handler.dispatchMessage(Handler.java:101)
        at android.os.Looper.loop(Looper.java:166)
        at android.app.ActivityThread.main(ActivityThread.java:7529) <1 internal call>
        at com.android.internal.os.Zygote$MethodAndArgsCaller.run(Zygote.java:245)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:921)
```

It seems to be outputting the same error. Then I realised that right at the top it says that the program is looking for a file with directory:

/storage/emulated/0/downloaddata.txt

which is missing a slash in between download and data.txt. After adding the slash and running the app again, it was still not working – outputting the same message as before. I wondered whether the "(Permission Denied)" message in the first line was the problem. As the program is writing something into the user's device, I wondered whether the user had

to give permission for the actions. And I was correct. The following two lines had to be added to the “AndroidManifest” file:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

The same message still showed up. Then I thought I’d change the path of the file to the path that is in my laptop:

```
File file2 = new File("documents/RunApp/app/src/main/java/readwritefile/data.txt ");
```

The error message after running this code was:

```
W/System.err: java.io.FileNotFoundException: documents/RunApp/app/src/main/java/readwritefile/data.txt (No such file or directory)
        at java.io.FileOutputStream.open0(Native Method)
        at java.io.FileOutputStream.open(FileOutputStream.java:287)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:223)
        at java.io.FileOutputStream.<init>(FileOutputStream.java:171)
        at com.instinctcoder.readwritefile.FileHelper.writeInternalStorage(FileHelper.java:165)
        at com.example.runapp.MainActivity.goToSecondPage(MainActivity.java:71) <1 internal call>
        at androidx.appcompat.app.AppCompatViewInflater$DeclaredOnClickListener.onClick(AppCompatViewInflater.java:385)
        at android.view.View.performClick(View.java:6291)
        at android.view.View$PerformClick.run(View.java:24931)
        at android.os.Handler.handleCallback(Handler.java:808)
        at android.os.Handler.dispatchMessage(Handler.java:101)
        at android.os.Looper.loop(Looper.java:166)
        at android.app.ActivityThread.main(ActivityThread.java:7529) <1 internal call>
        at com.android.internal.os.Zygote$MethodAndArgsCaller.run(Zygote.java:245)
W/System.err:     at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:921)
```

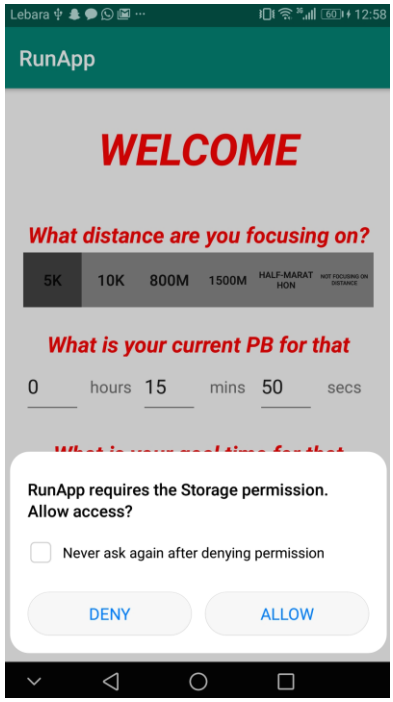
This tells me that I was on the right path before when I used the path of the user’s device. The problem there was that there was no permission from the user to write in the user’s device. I’ve looked into it. Apparently due to the newer version of Android Studio, permissions have to be asked during run time and therefore needs specific code for that, as compared to the older version where the permissions were asked during installation.

So the code I found (https://codinginflow.com/tutorials/android/run-time-permission-request) is:

```
private int STORAGE_PERMISSION_CODE = 1;
   .
   .
   .
public void goToSecondPage(View v) { //when submit button is pressed
      EditText question2_time_hours = (EditText) findViewById(R.id.question2_hours_type_in);
      EditText question2_time_minutes = (EditText)
findViewById(R.id.question2_minutes_type_in);
      EditText question2_time_seconds = (EditText)
findViewById(R.id.question2_seconds_type_in);

      if (ContextCompat.checkSelfPermission(MainActivity.this,
            Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED){
         .
         .
         .
      }else{
         requestStoragePermission();
      }
```

```
    }

private void requestStoragePermission() {
    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)){
        //if user denied the permission before but tries to access it again
        new AlertDialog.Builder(this)
                .setTitle("Permission needed")
                .setMessage("This permission is needed to save data into a text file")
                .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        ActivityCompat.requestPermissions(MainActivity.this, new String[]
{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                    }
                })
                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        dialog.dismiss();
                    }
                })
                .create().show();
    }else{
        //requesting permission
        ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
    }
}

@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    if (requestCode == STORAGE_PERMISSION_CODE){
        if (grantResults.length > 0 &&
grantResults[0]==PackageManager.PERMISSION_GRANTED){
            //Toast that permission is granted
        }else{
            //Toast that permission wasn't granted
        }
    }
}
```

When I ran the code, entered all the details and pressed the submit button, the following showed up:



I clicked “ALLOW” and it went to the next page while saying “Saved to File”, suggesting it has finally been written to the text file!

For reading from a file, the code is exactly the same. The only thing that changed is that it’s not “WRITE_EXTERNAL_STORAGE”, but “READ_EXTERNAL_STORAGE”.

Stage 2: Review

What has been done:

The code to save information to and read information from has been written. As well the code for getting the user's permission to write into and read from their device has been written. The program saves what the user has entered in the **first** and **third pages.**

How it has been tested:

A lot of testing has been going on as I was writing the code. Basically after every single prototype of a function that saves information to files has been written, it was immediately tested by entering information in the **first page** and pressing submit button. Code has been written which outputs "Error: Not saved" if the information was not saved and "Saved to File" if it was. That was the main factor of whether the code was successful or not.

How it meets the success criteria and user expectation:

The app now saves the information entered by the user in the opening stages of the app.

Criteria met/being met:

- Saving the user's preferred distance to train and their personal bests to a text file.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Saving the destination of where the user usually lives and trains in a text document.
- Saving the dates for user's future races.

Summary of the whole project as a prototype at this stage:

The app is getting to both look and function more like I described it in the design section. Combined with **Stage 1: the questionnaire,** the app seems to be looking very easily operative to the user (but this will be up to the stakeholders to decide) and it is functional, which is always a good thing.

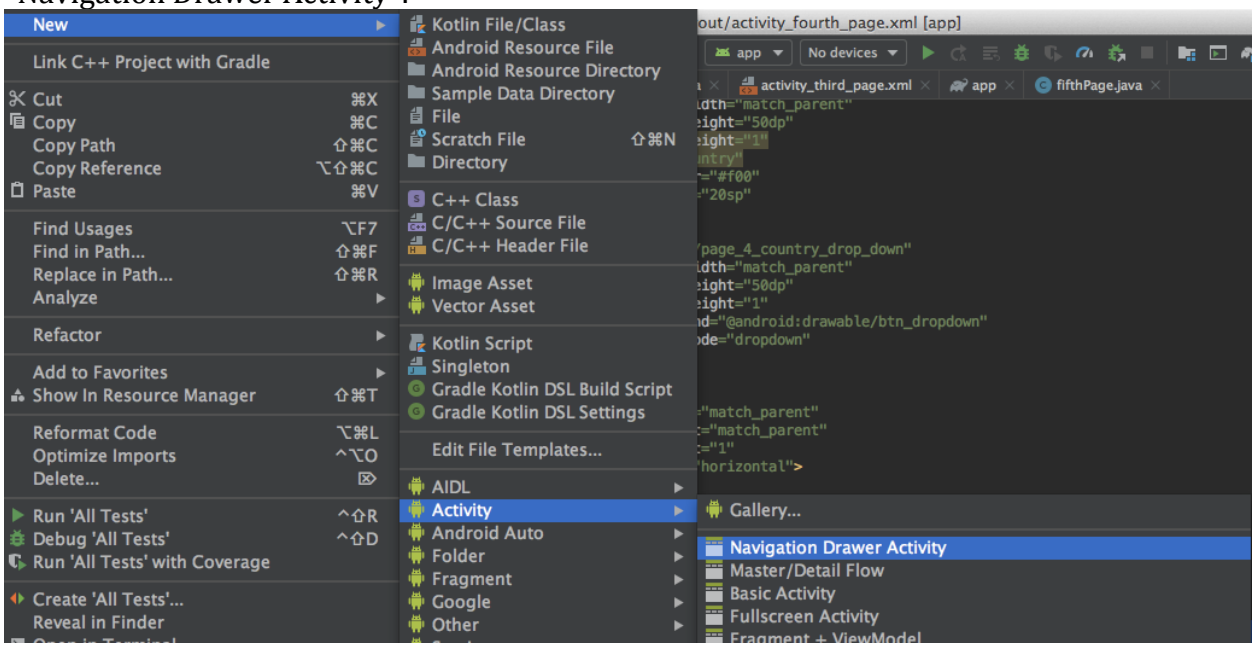
Stage 3: Side Menu Navigation

From the design section of the project, we know that side menu navigation is really important in helping the user navigate through the app. The side menu navigation has following pages linked: **workouts, routes, history, achievements and manual entry pages.**

Clicking on any of them opens a new activity with two bottom navigation options: home and settings. This will be talked about later.

Coding the side menu navigation:

Firstly I created a new activity, however this time instead of a blank activity I created a "Navigation Drawer Activity":



This is a natural feature in Android Studio.

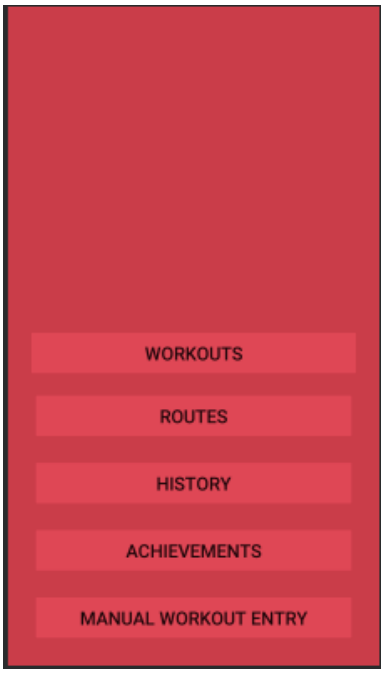
The confusing part about this activity is that it creates many XML pages, instead of just one like all the other activities, and each page is responsible for some part of the side menu navigation.

The activity generated the following code in the Java file upon opening:

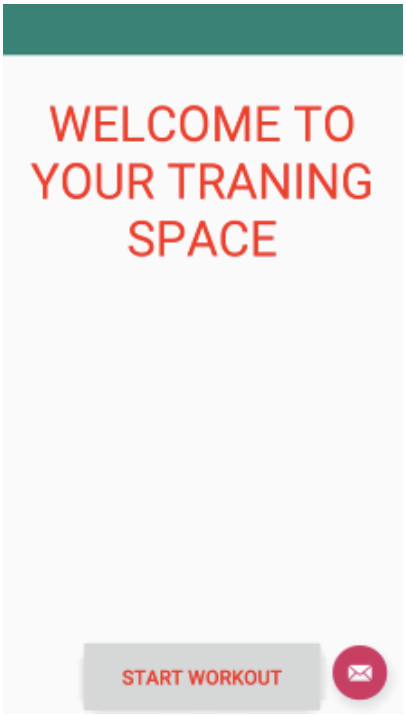
```
public class fifthPage extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fifth_page);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    NavigationView navigationView = findViewById(R.id.nav_view);
    // Passing each menu ID as a set of Ids because each
    // menu should be considered as top level destinations.
    mAppBarConfiguration = new AppBarConfiguration.Builder(
            R.id.nav_home, R.id.nav_gallery, R.id.nav_slideshow,
            R.id.nav_tools, R.id.nav_share, R.id.nav_send)
            .setDrawerLayout(drawer)
            .build();
    NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
    NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
    NavigationUI.setupWithNavController(navigationView, navController);
  }

}
```

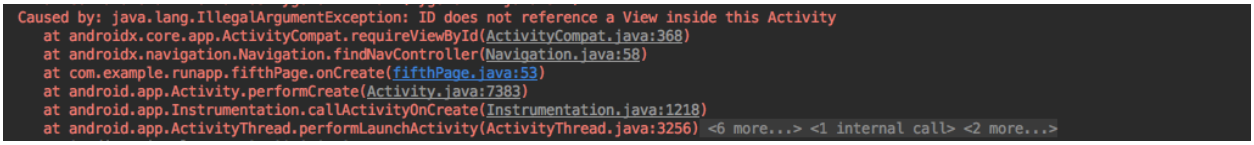
One of the “subpages” is “nav_header_fifth_page.xml”. This page is responsible for creating the actual buttons, which take the user to different pages. The coding is actually quite simple; it was finding which subpage relates to which page of the page I found to be the trickiest part. Like in *Stage 1: The Questionnaire*, I created a LinearLayout, which contained five buttons, each having an OnClick command, which opened a new activity. Here is how it looked:



However since side menu navigation is just a part of an actual page, I needed to look for a file, which was responsible for the main page. I first tried to code in the main fifth page activity xml, however that wasn't working. Finally I found that coding in "app_bar_fifth_page.xml" was the right thing. I added some text in the middle of the page and a "START WORKOUT" button at the bottom. Here is how it looks:



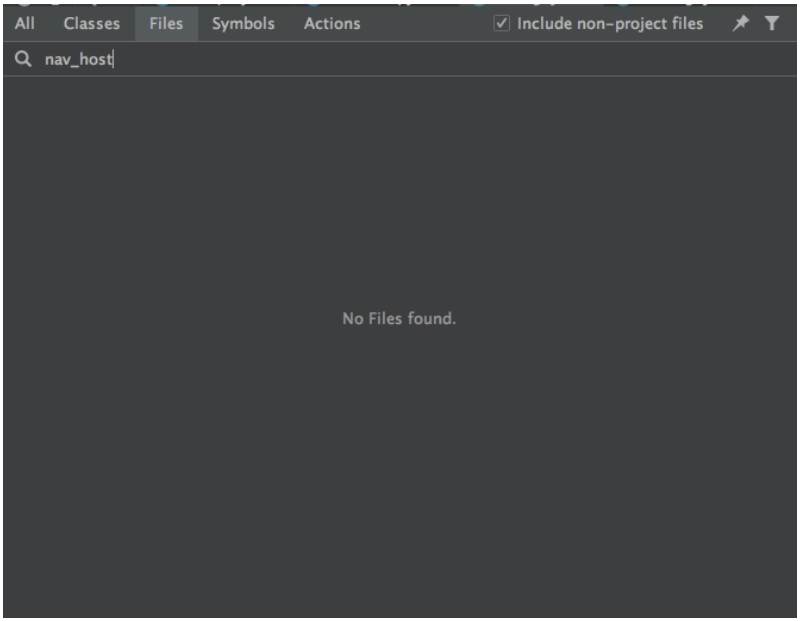
Upon trying to run the side menu navigation, the following error occurred:



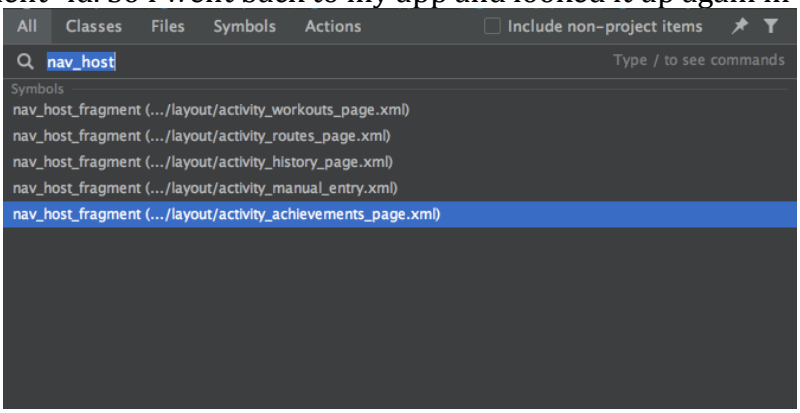
I looked at 53rd line of fifthPage.java and it was this:

```
NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
```

So what it is saying is that there isn't a file called "nav_host_fragment" and:



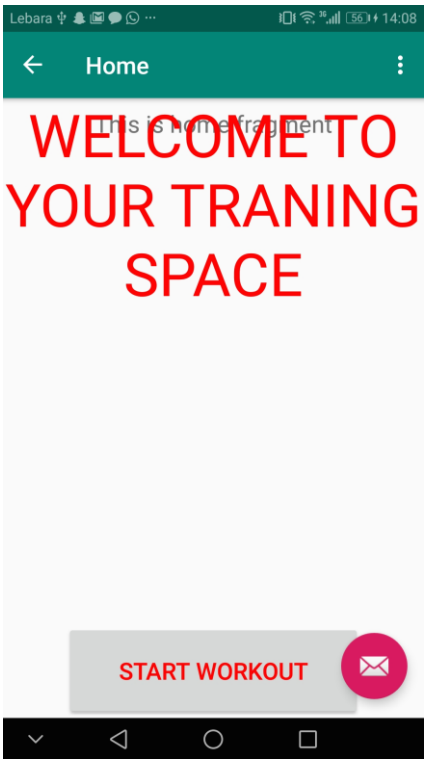
realised that no such file existed. I must have deleted it accidently. So to fix this, I created a new project and in there created a new "navigation drawer" activity. There I looked up "nav_host_fragment" but not in "Files" but in "All" and a path came up, I clicked on it and it took me to "content_fifth_page.xml". There was a <fragment> which had a "nav_host_fragment" id. So I went back to my app and looked it up again in "All" this time:



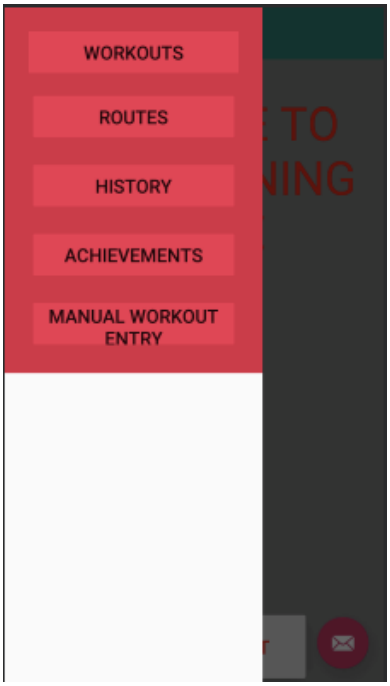
That made sense as to why the program was crashing – there isn't one for fifth page activity. So I went into "content_fifth_page.xml" and found this:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="com.google.android.material.appbar.AppBarLayout$Scrolli..."
    tools:showIn="@layout/app_bar_main">

    <!--<fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />-->
</androidx.constraintlayout.widget.ConstraintLayout>
```

For some reason I had it commented out. So I ran the code with the code uncommented out. The program didn't crash this time but it also didn't work. Here is how the page looked when it was opened:



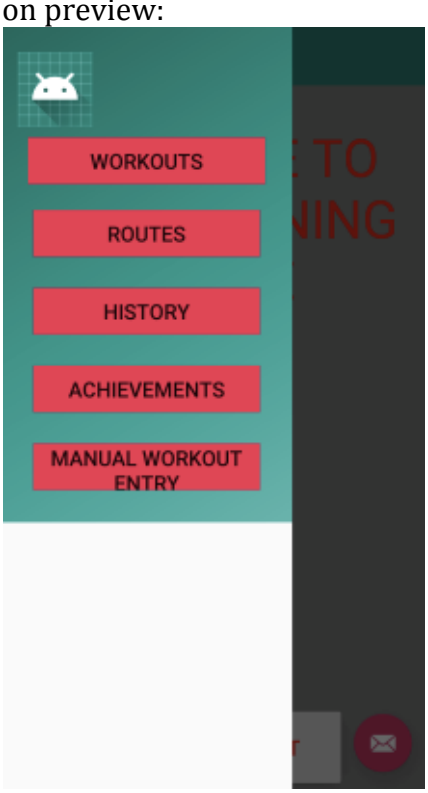
Clicking on the three dots in the top right corner only showed "Settings", which upon clicking does nothing. Clicking the back arrow doesn't work. And there is nowhere to view the side menu. This was unexpected as in Android Studio:



it looks functional. Looking through the terminal, no errors seemed to have showed up, so I'm not sure what the problem is here and how to solve it.

I've went back to "nav_header_fifth_page.xml" and changed it back to what it was. As I understand, the buttons I created above were in the header, so that could have been causing the problem. But the other problem is: that's the only place I could create buttons. "activity_fifth_page_drawer.xml" has <item>s in it, but they can't be given an onClick()

command and they don't work with buttons. So I commented out all the <item>s, increased the size of the header and added the buttons back. The idea behind this is that buttons were interfering with the <item>s, but now that they have been commented out, they shouldn't. This is how it looks on preview:



For some reason, it still doesn't work and I have no idea why because it is not outputting any error so I don't what to fix. Unfortunately I'll have to leave it like that.

Let's now talk about each page individually:

1. **Workouts Page:**
   This page outputs suggested workouts to the user according to what they chose as their main distance right at the start when they first opened the app. So the first thing, the program has to do is to check the contents of "data.txt", where the necessary information is stored. Since the distance is the first thing the program stores to the file, I wrote a code, which only reads the first line of the document and have placed it in the FileHelper file:

```
public static String readFirstLine(String file) {
    String text = null;
    try {
        File File = new File(file);
        BufferedReader br = new BufferedReader(new FileReader(File));
        text = br.readLine();
        br.close();
    } catch (FileNotFoundException ex) {
        Log.d(TAG, ex.getMessage());
    } catch (IOException ex) {
        Log.d(TAG, ex.getMessage());
    }
    return text;
}
```

   As can be seen from the code, the program uses BufferReader to read one line and then closes.
   This code can then be implemented to **workouts page** in such way:

```
TextView workout = findViewById(R.id.workouts_page_workouts);


if (FileHelper.readFirstLine("data.txt").equals("5K")){
    workout.setText(FileHelper.ReadFile("fiveK_training_plan.txt"));
}
else if (FileHelper.readFirstLine("data.txt").equals("10K")){
    workout.setText(FileHelper.ReadFile("fiveK_training_plan.txt"));
}
else if (FileHelper.readFirstLine("data.txt").equals("800")){
    workout.setText(FileHelper.ReadFile("800_training_plan.txt"));
}
else if (FileHelper.readFirstLine("data.txt").equals("1500")){
    workout.setText(FileHelper.ReadFile("800_training_plan.txt"));
}
else if (FileHelper.readFirstLine("data.txt").equals("HALF")){
    workout.setText(FileHelper.ReadFile("half_training_plan.txt"));
}
else if (FileHelper.readFirstLine("data.txt").equals("NONE")){
    workout.setText("THERE ARE NO WORKOUT SUGGESTIONS FOR YOU");
    workout.setTextSize(30);
}
```

   The program uses a series of if{…}else if{…} statements to cover every possible choice the user could have made back at the start. The ReadFile(String file){} function is coded like such:

```
public static String ReadFile(String file) {
        String line = "";
```

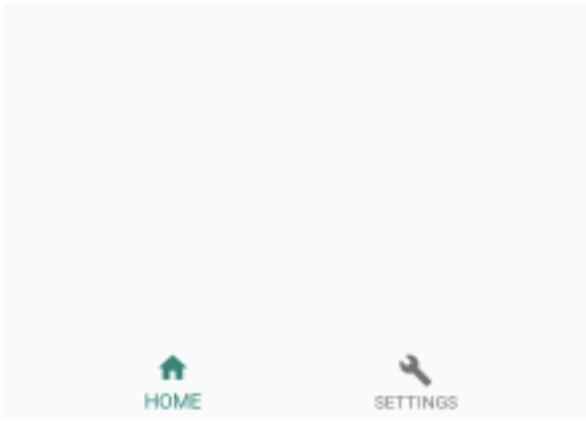
```
    String directory = System.getProperty("user.home");
    String absolutePath = directory + File.separator + file;

    try {
        BufferedReader bufferedReader = new BufferedReader(new FileReader(absolutePath));
        line = bufferedReader.readLine();
        while (line != null) {
            line = bufferedReader.readLine();

        }
    } catch (IOException e) {
        System.err.println();
    }
    return line;
}
```

The code is similar to readFirstLine() function (I used this code to write readFirstLine()), except here the program uses a while loop to keep outputting lines until the next line is blank. The "fiveK_training_plan.txt", "800_training_plan.txt" and "half_training_plan.txt" have been prewritten by me and placed into the "res" folder of the program.

I have also talked about those pages having bottom navigation with a home and a settings button. Each page is going to have the following code in "bottom_nav_menu.xml":

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">


    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="HOME" />

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"/>

</menu>
```

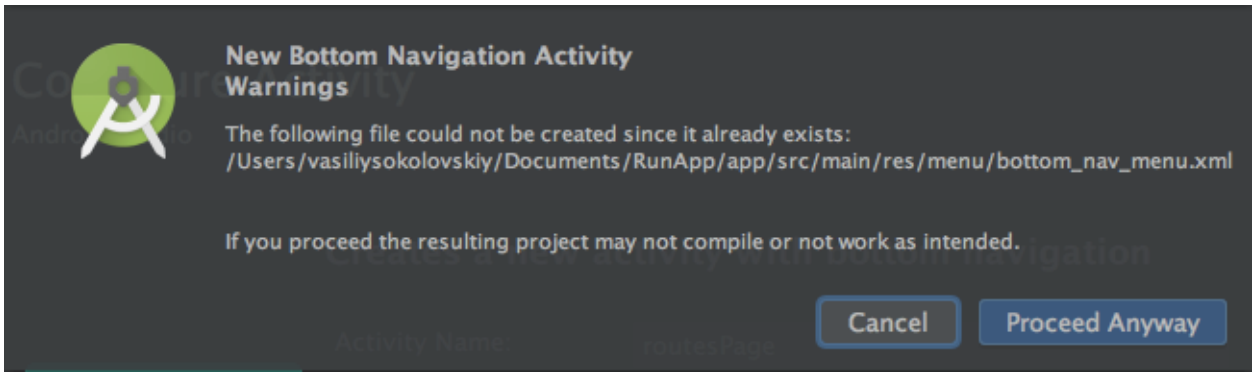
And it is going to look like this:



Obviously each is going to have an onClick command. When the home button is pressed, the program takes the user back to **"training space" page.** When the settings button is pressed, the user is taken to a settings page.

2. **History Page:**
   The history page is going to display all the workouts the user has ever done. Text document called "history.txt" is used to save all the user activity. It stores distance ran, time ran, average pace and date.

   When trying to create a new bottom navigation activity the following error occurred:



   I've realized later that because workouts page was also bottom navigation it created a file called "bottom_nav_menu.xml". So all I had to do was Refactor→Rename→"bottom_nav_menu_workouts_page.xml".

   So the code for the history page is pretty simple and looks like this:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_history_page);
    BottomNavigationView navView = findViewById(R.id.nav_view);
    // Passing each menu ID as a set of Ids because each
    // menu should be considered as top level destinations.
    AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
            R.id.navigation_home, R.id.navigation_dashboard, R.id.navigation_notifications)
            .build();
    NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
    NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
    NavigationUI.setupWithNavController(navView, navController);

    TextView history = findViewById(R.id.history_page_history);

    history.setText(FileHelper.ReadFile("history.txt"));
}
```

The process of saving workouts is going to be talked about in the **Run Stats Page (Stage 4: Working with Maps)** and **Manual Entry Page**.

3. **Routes Page:**
   This page is going to display to the user good routes in their area. As mentioned in the design section however, the limitation to this feature is that it will only display good routes if the user lives in Edinburgh.

   So the first step is to check where the user said they usually live and train:

```
if (FileHelper.ReadFile("residence.txt").equals("Edinburgh")){
    .
    .
    .
}
else{
    .
    .
    .
}
```

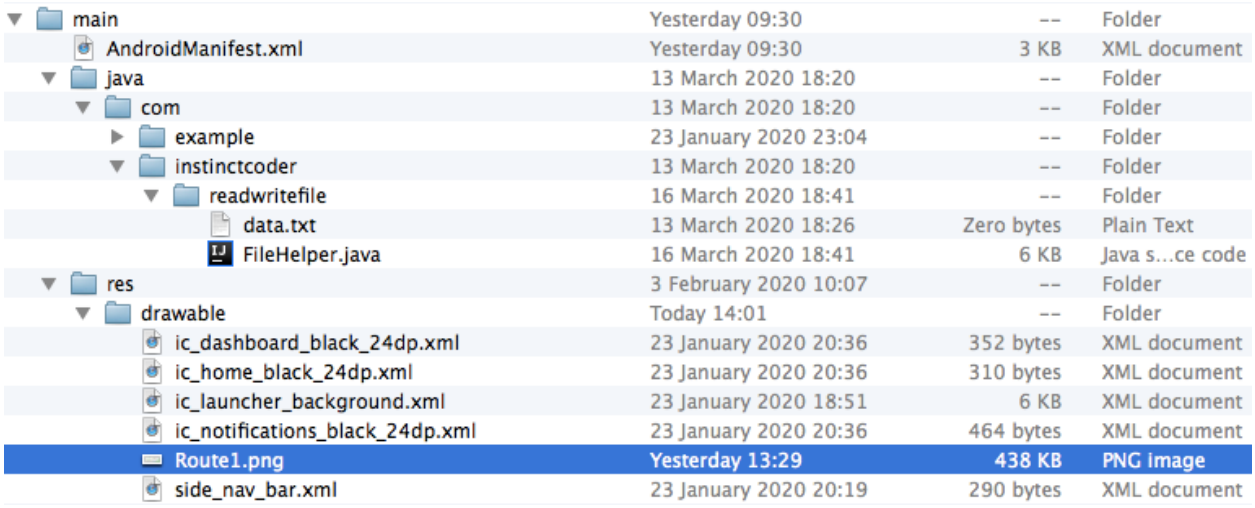
   In the xml document, I found a new property called <ImageView>, which I'm guessing is designed for outputting images to the user:

```
<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/routes_page_first_route"
    />
```

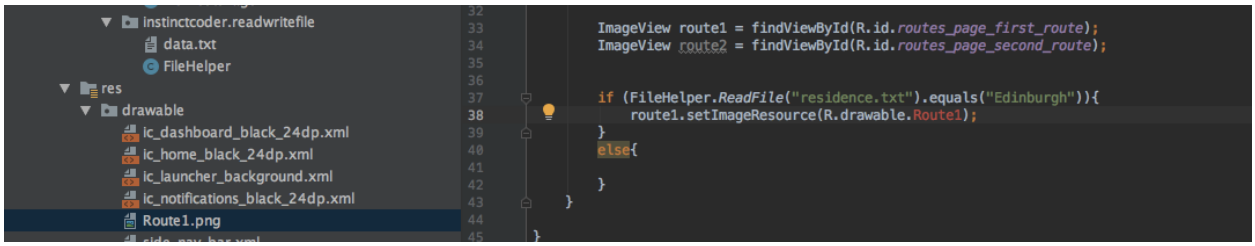
   The "<imageView" text got highlighted. Scrolling my mouse over it:

   According to https://www.deque.com/blog/android-imageviews-accessible-content-descriptions/, the contentDescription is not always needed so I ignored it.

According to this YouTube tutorial, https://www.youtube.com/watch?v=Y7JTkXoN8OE, the way to add an image through the Java file is: firstly add the file to the "drawable" folder in the app (RunApp→src→Main→res→drawable):

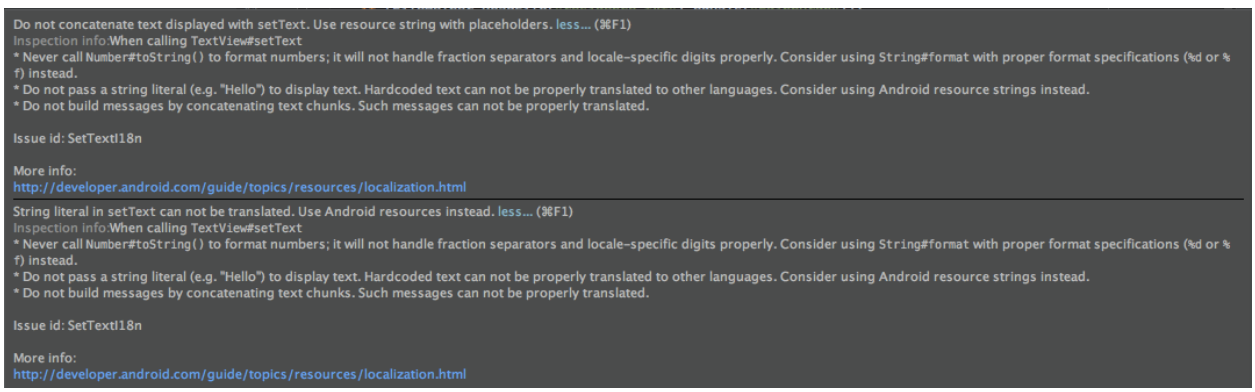


Then after defining the <ImageView> in Java, you can set the image using route1.setImageResource(). I've tried doing that:



As can be seen, the code is not recognising "Route1". That's weird. I've looked closer into the tutorial and realised that the person's image had JPEG extension, while mine has PNG. So I've changed it to JPEG. Added it to the "drawable" folder and tried the same code again. "Route1" was still getting highlighted. At this point, because I still had the PNG image in there with the same name, I thought the program is probably not differentiating the two images. So I renamed the JPEG one into "route1.jpg". And finally it worked.

After defining <TextView>, I added a description for the user so they know what to look out for when they run the course. Once I inputted it, the text got highlighted. I rolled my mouse over it. Here is what it says:



I believe the most important part of that was:

```
* Do not pass a string literal (e.g. "Hello") to display text. Hardcoded text can not be properly translated to other languages. Consider using Android resource strings instead.
* Do not build messages by concatenating text chunks. Such messages can not be properly translated.
```

StackOverflow saves the day: https://stackoverflow.com/questions/15151152/load-large-text-in-android. It said to create a new folder (I called it "raw") inside RunApp→src→Main→res. Place a text file with the description inside it (I called it route1description.txt). Then implement the following try{…}catch{…} code:

```
try{
    Resources res = getResources();
    InputStream in_s = res.openRawResource(R.raw.route1description);
    byte[] b = new byte[in_s.available()];
    in_s.read(b);
    route1description.setText(new String(b));
}catch (Exception e){
    route1description.setText("Error: Can't show description");
}
```

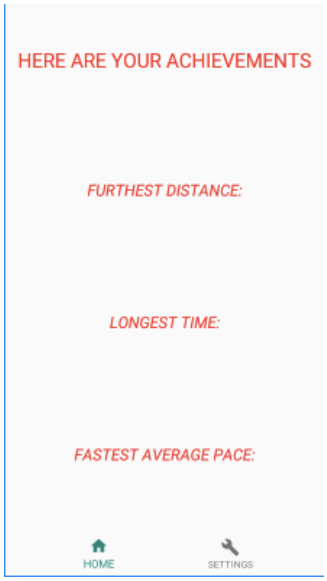
The same code was implemented for the second route.

4. **Achievements Page:**
   This page, similar to the **history page**, was not a lot of effort to code out. This page outputs the user's achievements. Those include: furthest distance ran, longest time on feet and fastest average pace. Those three values have been saved to a text document following a personal record-breaking run. There are two ways of making new values on the achievements page: 1. Do a workout using the app and if the user set a new pb, it automatically adds it to the document after the run, or 2. Enter the run the user did through the **manual entry page** and if the run included a pb, it also automatically adds it to the page. Both of those ways will be talked about later.

   This page's main goal is to output the contents of a text document called "achievements.txt". Pretty nice and simple:

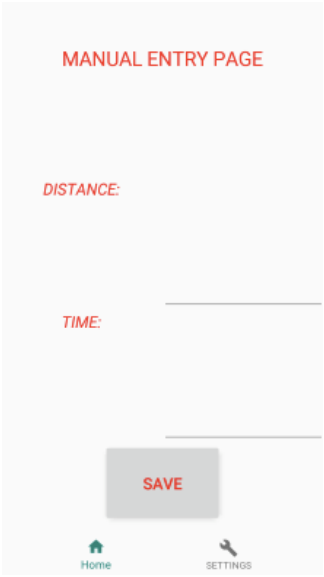
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_achievements_page);
    BottomNavigationView navView = findViewById(R.id.nav_view);
    // Passing each menu ID as a set of Ids because each
    // menu should be considered as top level destinations.
    AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
            R.id.navigation_home, R.id.navigation_dashboard, R.id.navigation_notifications)
            .build();
    NavController navController = Navigation.findNavController(this, R.id.nav_host_fragment);
    NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
    NavigationUI.setupWithNavController(navView, navController);

    TextView furthest_distance = findViewById(R.id.furthest_distance);
    TextView longest_time = findViewById(R.id.longest_time);
    TextView fastest_average = findViewById(R.id.fastest_average);

    furthest_distance.setText(FileHelper.ReadFile("distance_achievements.txt"));
    longest_time.setText(FileHelper.ReadFile("time_achievements.txt"));
    fastest_average.setText(FileHelper.ReadFile("pace_achievements.txt"));
}
```

   The page looks like this:



5. **Manual Entry Page:**
   This page allows the user to enter the details of the run they did without the app. The processes behind this page are pretty simple. There are three main processes:
   1. Entry of details
   2. Adding those details to the **history page.**
   3. Checking whether a new pb has been set.

   Number 1 is pretty simple. It includes setting up a new activity with icons and properties which allows the user to enter the details of their run:



   Next to "DISTANCE:" and "TIME:" there are boxes for text entry. For them I did inputType=numberDecimal.

   Process number 2 requires writing of data into the "history.txt" document:

```
public void save(View v){ //when "SAVE" is clicked
    EditText distance_entry = findViewById(R.id.distance_entry);
    EditText time_entry = findViewById(R.id.time_entry);

    Date today = Calendar.getInstance().getTime(); //gets the current date
    SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
    String formattedDate = curFormater.format(today);


    if (!distance_entry.getText().toString().equals("-") ||
!distance_entry.getText().toString().equals(" ")){
        if (!time_entry.getText().toString().equals("-") || !time_entry.getText().toString().equals("
")){
            double distance = Double.parseDouble(distance_entry.getText().toString());
            double time = Double.parseDouble(time_entry.getText().toString());
            double average_pace = time/distance; //PS I'm not dumb - this gives minutes per k
            //which is what runners tend to operate at

            String details = formattedDate + ": " + distance_entry.getText().toString() + "km; " +
time_entry.getText().toString() + "min; "
                    + Double.toString(average_pace) + "min/k;";
            FileHelper.writeInternalStorage("history.txt", details);
            Intent intent = new Intent(this, fifthPage.class);
            startActivity(intent);
        }
```

```
        else{
            String details = formattedDate + ": " + distance_entry.getText().toString();
            FileHelper.writeInternalStorage("history.txt", details);
            Intent intent = new Intent(this, fifthPage.class);
            startActivity(intent);
        }
    }
    else{
        if (!time_entry.getText().toString().equals("-") || !time_entry.getText().toString().equals("
")){
            String details = formattedDate + ": " + time_entry.getText().toString();
            FileHelper.writeInternalStorage("history.txt", details);
            Intent intent = new Intent(this, fifthPage.class);
            startActivity(intent);

        }
        else{
            Toast.makeText(this, "Nothing Entered", Toast.LENGTH_SHORT).show();
        }
    }
}
```

As can be seen from the code, the program goes through different possibilities of entries – because the use maybe not have entered either distance or time. And only when both are entered, the program calculated average pace. It saves to the "history.txt" document the following: date (dd/mm/yyyy); distance (if present) km; time (if present) min; pace (if both time and distance are present) min/k;

Final stage: check if the user has beaten any of their previous records. This was a tad bit trickier. The problem here is that if the user did get a new personal record, the file with the previous record had to be cleared and the new record had to be written over that. For that I wrote a new function in "FileHelper":

```
public static boolean writeToAchievementsFile(String fileN, String textToWrite){
    boolean done;
    File fileName = new File(fileN);
    File root = android.os.Environment.getExternalStorageDirectory();
    File dir = new File (root.getAbsolutePath() + "/download/");
    dir.mkdirs();
    File file = new File(dir.getAbsolutePath()+"/"+fileName);
    try {
        FileOutputStream f = new FileOutputStream(file);
        PrintWriter pw = new PrintWriter(f, false); //deletes the contents of the file
        pw.println(textToWrite);                    //before writing to it
        pw.flush();
        pw.close();
        f.close();
        done=true;
    } catch (FileNotFoundException e) {
        done=false;
        e.printStackTrace();
        Log.i(TAG, "******* File not found ");
    } catch (IOException e) {
```

```
        done = false;
        e.printStackTrace();
    }
    return done;
}
```

As can seen, the code doesn't differ much from "writeToInternalStorage" that I wrote earlier. But the important line in this code is:

```
PrintWriter pw = new PrintWriter(f, false);
```

The second statement (false), suggests that the document will firstly be flushed before writing into it (https://stackoverflow.com/questions/9927281/how-to-use-flush-for-printwriter).

So the new code added to the manual entry page was:

```
Double previous_distance_record =
Double.parseDouble(FileHelper.ReadFile("distance_achievements.txt").toString());
Double previous_time_record =
Double.parseDouble(FileHelper.ReadFile("time_achievements.txt"));
Double previous_pace_record =
Double.parseDouble(FileHelper.ReadFile("pace_achievements.txt"));

if (distance>previous_distance_record){
    String newRecord = distance_entry.getText() + "km (" + formattedDate + ")";
    FileHelper.writeToAchievementsFile("distance_achievements.txt", newRecord);
}
if (time>previous_time_record){
    String newRecord = time_entry.getText() + "min (" + formattedDate + ")";
    FileHelper.writeToAchievementsFile("time_achievements.txt", newRecord);
}
if (average_pace<previous_pace_record){
    String newRecord = Double.toString(average_pace) + "min/k (" + formattedDate + ")";
    FileHelper.writeToAchievementsFile("pace_achievements.txt", newRecord);
}
```

Stage 3: Review

What has been done:

The main **"training space" page** has been created. I made sure that it looked as friendly as possible. Also the page includes a side menu navigation, which upon clicking gives a choice of five pages. All of these pages have been coded. Each of these pages also have bottom navigation, where there is an option to either go back to the **"training space" page** or to the **settings page**. This has made it easier for the user to navigate across the app.

How it has been tested:

Once again, a lot of tests have been conducted during the development stage. However due to the non-functional side menu navigation, I had to create new projects and tests pages I wrote about above individually to make sure they work. It is a shame the side menu navigation doesn't work because that means the stakeholders won't be able to use the app properly when testing.

How it meets the success criteria and user expectation:

There are many functional pages, which display the user's workout history, achievements, suggested workouts and good routes to run on. The user also has the ability to manually enter their workout and the app will automatically save that workout (along with the date) into the workout archive, as well as checks whether new personal bests have been set.

The app unfortunately doesn't meet success criteria and user expectation for side menu navigation. For some reason it just won't work. It outputs perfectly functional side menu navigation in preview, however whenever I run it on my phone, it doesn't show up and most importantly doesn't output any sort of error.

Criteria met/being met:

- Easy and clear to answer questions to find out information about the user.
- Reasonably good and user-friendly graphical user interface.
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Have a page, which allows manual entry of workouts.
- Have a page, which outputs user's achievements: longest distance/longest time/fastest average speed ran.
- Have a page, which outputs the user's workout history
- Clearly show good routes to the user
- Suggest good workouts to the user to get to their goal faster.
- Have side menu navigation at the main page with many other pages linked to it.

Summary of the whole project as a prototype at this stage:

Hard to think positively at this stage, especially because the hard coding part is **Stage 4**. I shall maybe come back later and try to fix this error if I have time at the end.

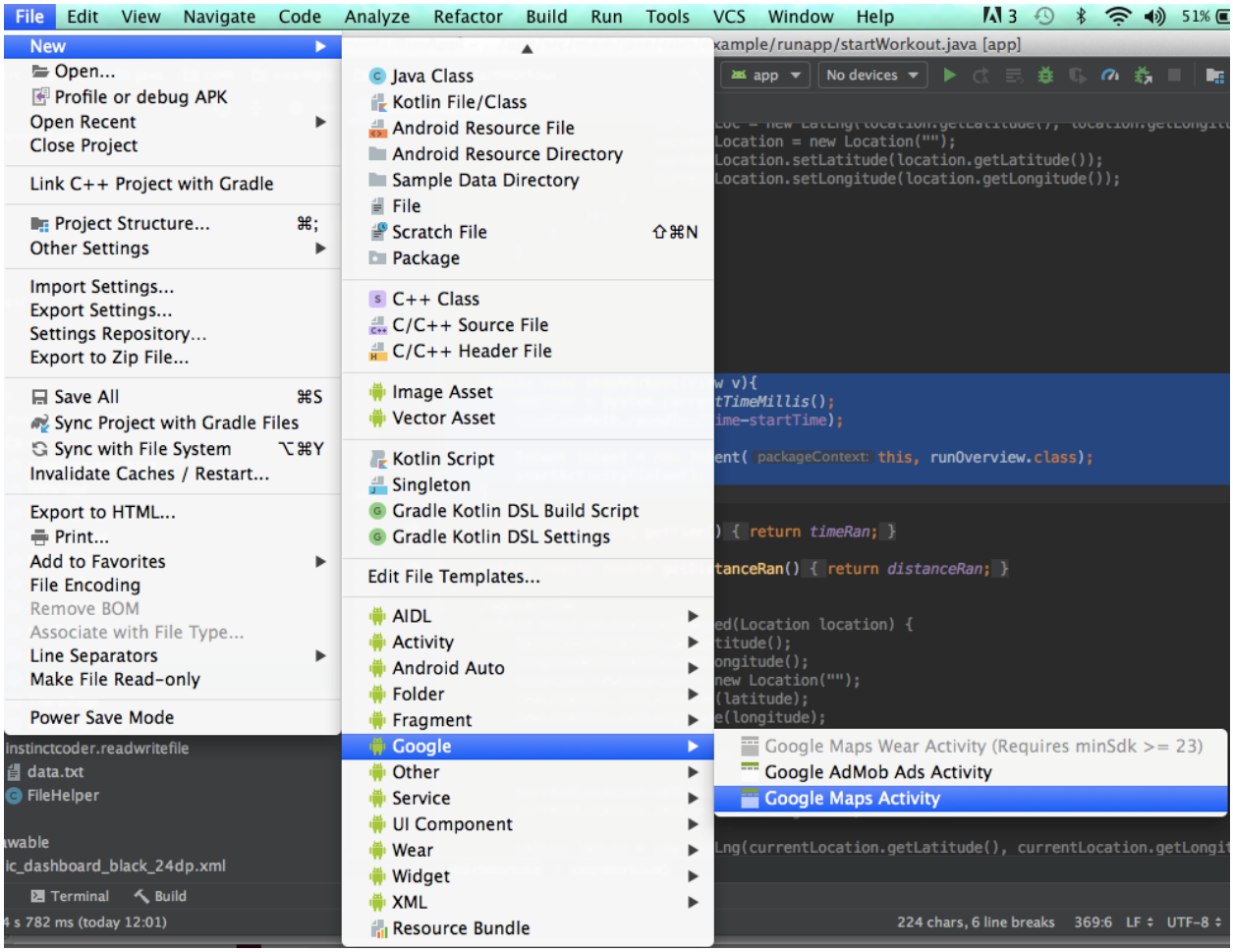
Stage 4: Working with Maps

In this stage, I work on the map, which appears when the "START WORKOUT" button in the **"training space" page** is pressed. I'm going to attempt to make it look like in the professional running apps including highlighting of the route ran, showing current location of the user, and calculating distance, time and pace of the user.

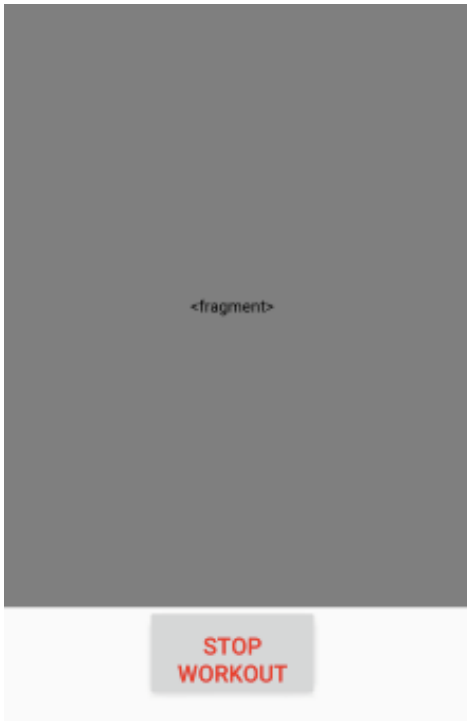
To start with the easy one, calculating time is very easy to do. Right at the start, when the maps are first opened, the program starts the stopwatch and when the user pressed "STOP WORKOUT", the stopwatch stops.

```
public long startTime = System.currentTimeMillis();
    .
    .
    .
public void stopWorkout(View v){
    endTime = System.currentTimeMillis();
    timeRan=Math.round(endTime-startTime);

    Intent intent = new Intent(this, runOverview.class);
    startActivity(intent);
}
```

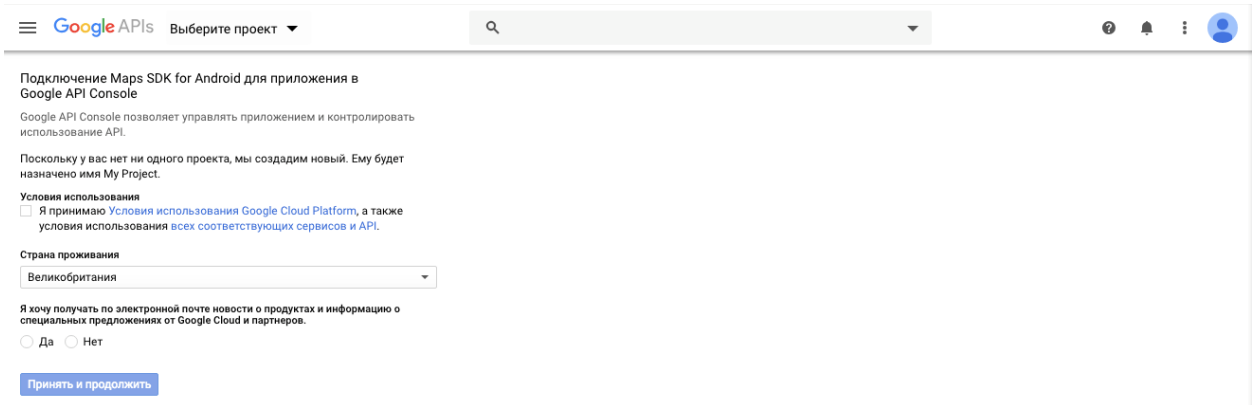
Next stage is to set up the map. Android Studio actually has a "Map Activity":



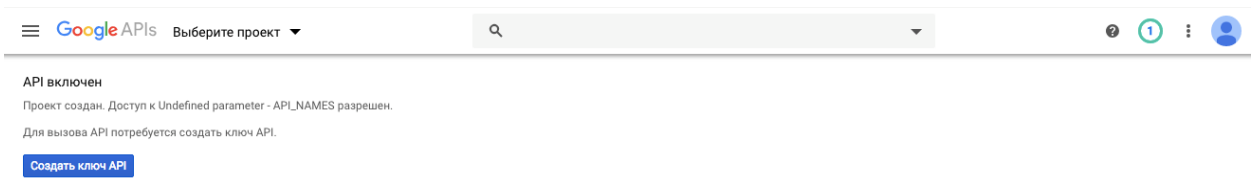
Firstly I went to the XML and created the “STOP WORKOUT” button:



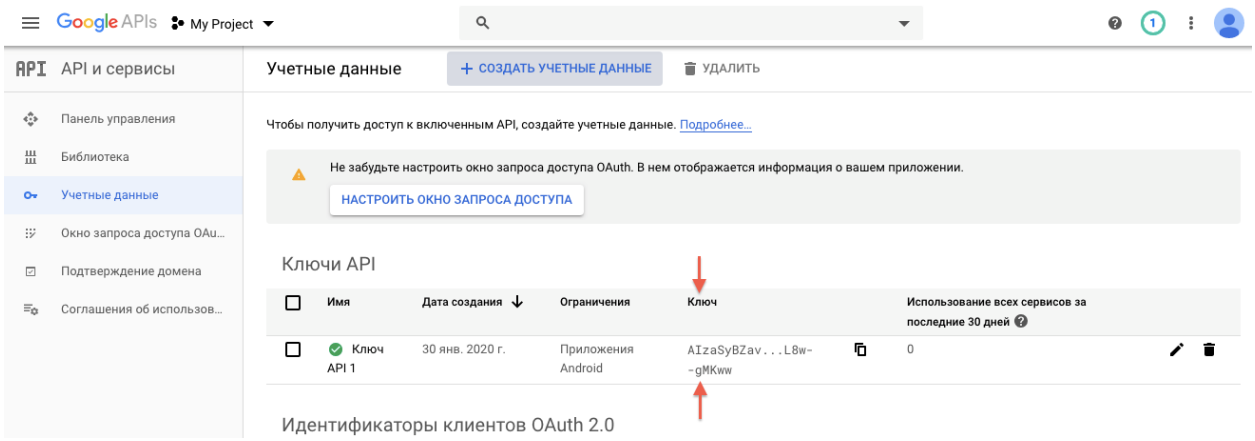
There was one more thing I had to do to get the maps howing and working: I had to get the Google Maps API key. I had to go to a website mentioned in in the XML file. It took me to this page:



Don’t mind all the Russian. This first page asked that I have read and understood terms and conditions for using Google Cloud Platform and also asked me to input which country I live in. After that was done, it took me to this page:



Here I just had to press “create API key”. And then the next page:



The two red arrows is where the key is. I just had to copy it and paste it in the “google_maps_api.xml” like such:

```
<string name="google_maps_key" templateMergeStrategy="preserve"
translatable="false">AIzaSyBZaveQCoTPcTY8lXOiolM60L8w--gMKww</string>
```

Now over at the Java file:

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start_workout);
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        mContext = this;
        locationManager = (LocationManager)getSystemService(LOCATION_SERVICE);
        mContext.getSystemService(Context.LOCATION_SERVICE);


    }

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to be used.
 * This is where we can add markers or lines, add listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the device, the user will be prompted to install
 * it inside the SupportMapFragment. This method will only be triggered once the user has
 * installed Google Play services and returned to the app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    // Add a marker in Sydney and move the camera
```

```
    LatLng sydney = new LatLng(-34, 151);
    mMap.addMarker(new MarkerOptions().position(sydney).title("Marker in Sydney"));
    mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney));
}
```

But actually an important part of this is what comes earlier than this:

```
public class startWorkout extends FragmentActivity {···}
```

This is important for reasons that will become clear soon. I first started looking into how to get the user's location continuously. After some research, I bumped into this aspect called LocationListener (https://developer.android.com/reference/android/location/LocationListener). This feature is useful because I code a function called onLocationChanged(), which does what the title suggests, every time there is a location change, it can do whatever I code into it. I tried to code it out manually:

```
private final LocationListener mLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(final Location location) {

    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {

    }

    @Override
    public void onProviderEnabled(String provider) {

    }

    @Override
    public void onProviderDisabled(String provider) {

    }
};
```

However the top line that I was talking about earlier got underlined and here is the error message it was outputting:

```
Class 'startWorkout' must either be declared abstract or implement abstract method 'onLocationChanged(Location)' in 'LocationListener'
```

StackOverflow (https://stackoverflow.com/questions/36829998/class-must-either-be-declared-abstract-or-implement-abstract-method) suggested to put "implements OnMapReadyCallback" in after "public class startWorkout extends FragmentActivity". And this did get rid of the message.

The next step is to see what to put into the onLocationChanged to meet my criteria.

I've started by trying to get the program to highlight the route as the user is running as well as calculate the distance as they are running. I've learnt about the aspect of "markers" which I used in the following way:

```
public static Location currentLocation;
    .
    .
    .
@Override
public void onLocationChanged(android.location.Location location) {
    latitude=location.getLatitude();
    longitude=location.getLongitude();
    Location newLocation = new Location("");
    newLocation.setLatitude(latitude);
    newLocation.setLongitude(longitude);

    distanceRan = distanceRan + currentLocation.distanceTo(newLocation);

    currentLocation.setLatitude(latitude);
    currentLocation.setLongitude(longitude);



}
```

The idea behind the code is: as there has been a location change, the program gets latitude and longitude of the new location. Then distance from the previous location (*currentLocation*) and new location (*newLocation*) is added to the total distance. Then we set the new location as current location so the distance can be calculated again once there has been a location change. Now what I need to do is define *currentLocation* right at the start for this code to work. This is how I did it:

```
Location location = new Location("");
LatLng currentLocation = new LatLng(location.getLatitude(), location.getLongitude());
```

I've put this in the onCreate() function and ran the code. The app was not even opening on my phone and all it said was:

```
03/22 17:36:00: Launching 'app' on HUAWEI HUAWEI NXT-AL10.
$ adb shell am start -n "com.example.mapapp/com.example.mapapp.MapsActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
Waiting for process to come online...
```

And then it said connection timed out and nothing turned on.

I went back a step and just ran the code that was originally given to me (the very top one, where it says “Marker in Sydney”) along with my onLocationChanged() code. Here is the outcome:

That is a positive result. So what is going wrong when I add that extra code and how do I know my onLocationChange() function is working? Those are the main question I shall be focusing for a while.

So first of all, focusing on whether onLocationChanged() works. In the process of last running the program, I spotted the following error (highlighted):

```
W/System.err: "gps" location provider requires ACCESS_FINE_LOCATION permission.
D/HwRTBlurUtils: check blur style for HwPhoneWindow, themeResId : 0x7f100006, context : com.example.runapp.startWorkout@f373e10, Nhwext : 0, get Blur : disable with , null
D/ActivityThread: add activity client record, r= ActivityRecord{d2fba13 token=android.os.BinderProxy@7e484c2 {com.example.runapp/com.example.runapp.startWorkout}} token= an
D/OpenGLRenderer:   HWUI Binary is  enabled
I/Choreographer: Skipped 33 frames!  The application may be doing too much work on its main thread.
I/PressGestureDetector: onAttached begin
I/PressGestureDetector: onAttached end
I/PressGestureDetector: enabledInPad = false,isPcCastMode = false
```

So I understand there is some difficulty in the user permission again (similar to the read and write commands earlier). This makes sense as I ask the user for their latitude and longitude every time they are changing location. So once again I went to the AndroidManifest.xml file and inserted the following line:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

I ran the program again. Similar errors kept coming up. I had to do more research as to why the program is not getting user’s permission. I’ve found that just writing that line was enough for older versions of Android Studio, but with the newer version, you had to write specific code, which asks the user for their permission (https://stackoverflow.com/questions/57098852/how-to-ask-for-location-permission-in-android-studio). Here is the code I inserted:

```
if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {

    // Permission is not granted
    // Should we show an explanation?
    if (ActivityCompat.shouldShowRequestPermissionRationale(this,
            Manifest.permission.ACCESS_FINE_LOCATION)) {
        // Show an explanation to the user *asynchronously* -- don't block
        // this thread waiting for the user's response! After the user
        // sees the explanation, try again to request the permission.
    } else {
        // No explanation needed; request the permission
        ActivityCompat.requestPermissions(this,
                new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                1);

        // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
        // app-defined int constant. The callback method gets the
        // result of the request.
    }
} else {
    // Permission has already been granted
}




try {
    LocationManager mLocationManager;
    mLocationManager = (LocationManager) getSystemService(LOCATION_SERVICE);

    mLocationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1,
            1, mLocationListener);
}
catch(SecurityException err)
{
    System.err.println(err.getMessage());
}
```

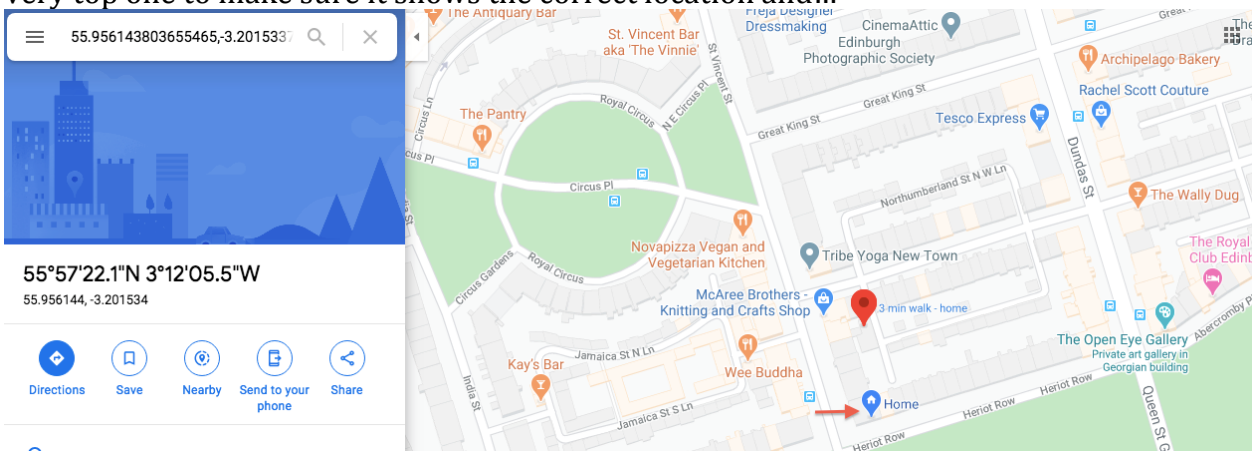
I had to insert it inside the onCreate() function. To make the code a bit easier to run, I've changed the code inside onLocationChanged() to the following:

```
System.err.println(location.getLatitude()+","+location.getLongitude());
```

So every time there is a change in location, the program should output the new latitude and longitude. I've run the code and it finally worked. Here was the output:

```
I/DynamiteLoaderV2: [71] Googlecertificates
W/zygote64: Skipping duplicate class check due to unrecognized classloader
I/HwApiCacheMangerEx: apicache pi PackageInfo{5821408 com.example.mapapp}
I/zygote64: Do full code cache collection, code=123KB, data=101KB
I/zygote64: After code cache collection, code=120KB, data=71KB
W/System.err: 55.956143803655465,-3.2015337590176838
W/System.err: 55.956150909,-3.201543574
W/System.err: 55.955933986,-3.201499868
W/System.err: 55.955747754,-3.201455172
```

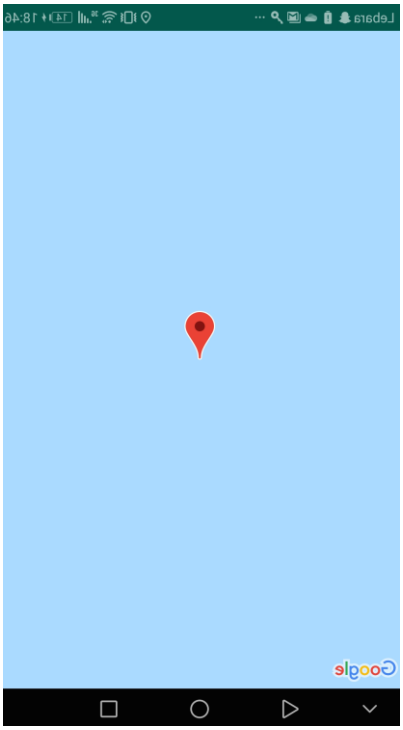
Everything that starts with “W/System.err:” is the code that output my location. I tried the very top one to make sure it shows the correct location and…



The red marker is where the program said I was and the blue marker that said home (red arrow) is where I actually was. This reassured me that I was moving in the right direction. The was also a pop up when I started running the app, which asked me whether I allow the app to get my location, but unfortunately I forgot to screenshot it and when ran again it didn’t ask me again. Next step is to make the program actually show my location on my phone just like Google maps shows me the marker above. So here is the code I added in the onMapReady() function instead of the marker in Sydney code:

```
@Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        Location location = new Location("");
        LatLng currentLocation = new LatLng(location.getLatitude(), location.getLongitude());
        mMap.addMarker(new MarkerOptions().position(currentLocation));
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLocation, 50));
    }
```

Here is the result:



There are two possible explanations to this: 1. It's showing I'm in the middle of an ocean, or 2. It's zoomed in too much and here is why I think so: the second parameter of CameraUpdateFactory.newLatLngZoom() (the 50) is how much to zoom in by. I inserted that number myself so it's probably to big. So I looked up what the perfect zoom is and it said 16.0f (f signalising a float) ← https://stackoverflow.com/questions/29868121/how-do-i-zoom-in-automatically-to-the-current-location-in-google-maps-api-for-an. I've run the program again. Unfortunately it showed me the same thing. I started zooming out and it was showing that I am indeed in the middle of an ocean:

I fond that to be fairly confusing as onLocationChnaged() gave the right coordinates. So decided to write the same code there and see if the result differs.

And indeed it does:

You can also see that it kept on adding markers as I was walking around the room. This is going to be useful when I try to highlight the route the user is running as they are running it and also to calculate distance.

So next I tried to make the program closer to meeting the criteria. Here is what I wrote in onLocationChanged():

```
latitude = location.getLatitude();
longitude = location.getLongitude();
LatLng currentLocation = new LatLng(latitude, longitude);
coordinates.add(currentLocation);
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLocation, 16.0f));
redrawLine();
```

where latitude and longitude are global double variables, and coordinates is an ArrayList<LatLong> (also a global variable). And the redrawLine() function is as follows:

```
public void redrawLine(){
    mMap.clear();

    PolylineOptions polylineOptions = new
PolylineOptions().width(5).color(Color.BLUE).geodesic(true);
    for (int i = 0; i<coordinates.size(); i++){
        LatLng point = coordinates.get(i);
        polylineOptions.add(point);
    }
    addMarker(); //adds current location
    line = mMap.addPolyline(polylineOptions);
}
```

and addMarker() function is:

```
public void addMarker(){
    mMap.addMarker(new MarkerOptions().position(new LatLng(latitude, longitude)));
}
```

The redrawLine() function does the following: firstly clears the map from all markers and lines, then it plots all coordinated that were gained from onLocationChanged() onto the map and draw a “ployline” between them.

I’ve tried running this code and every step I took (ie changed location) the map was reopening and nothing was appearing on it, until finally it crashed and the error message was as follows:

```
03/22 21:24:48: Launching 'app' on HUAWEI HUAWEI NXT-AL10.
$ adb shell am start -n "com.example.mapapp/com.example.mapapp.MapsActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
Waiting for process to come online...
Connected to process 28211 on device 'huawei-huawei_nxt_al10-CJL5T16109017783'.
Capturing and displaying logcat messages from application. This behavior can be disabled in the "Logcat output" section of the "Debugger" settings page.
I/HwApiCacheMangerEx: apicache pi PackageInfo{5821408 com.example.mapapp}
    apicache pi PackageInfo{1a07f57 com.example.mapapp}
Process 28211 terminated.
D/AndroidRuntime: Shutting down VM
E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.example.mapapp, PID: 28211
```

My original thoughts is that change of location happens too often for the program to run the redrawLine() function. Also because the map kept reopening every change in location could be caused by mMap.clear() in the redrawLine() function.

I looked at other methods of running some code in the background and came across “Runnable” (https://developer.android.com/training/multiple-threads/define-runnable). Here is how I implemented it:

```
public Handler handler;
    .
    .
    .
public Runnable runLocation = new Runnable(){
        @Override
        public void run(){
            LatLng latLng = new LatLng(latitude, longitude);
            coordinates.add(latLng);
            redrawLine();
            handler = new Handler();
            MapsActivity.this.handler.postDelayed(MapsActivity.this.runLocation, 2000);
        }
    };
```

I've run this code and the results were disappointing. The following page showed:

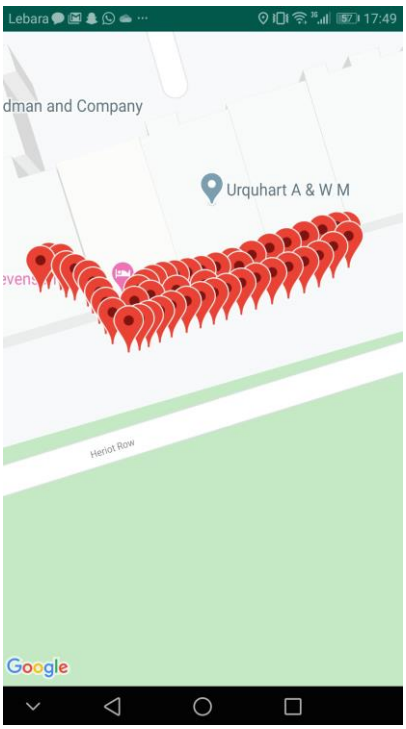
and nothing was happening.

But then I realised that because I commented out all the code in onLocationChange(), latitude and longitude were not getting defined anywhere, so I have uncommented the code out and ran it again. The results were more positive as now it was showing my location and the camera was moving as I was moving but it wasn't drawing a line.

I went back a step once again. I returned my old code in onLocationChange() and commented out all the code in Runnable. The onLocationChange() code was like this:

```
@Override
public void onLocationChanged(final Location location) {
    latitude = location.getLatitude();
    longitude = location.getLongitude();
    LatLng currentLocation = new LatLng(latitude, longitude);
    coordinates.add(currentLocation);
    mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLocation, 16.0f));
    mMap.addMarker(new MarkerOptions().position(currentLocation));

    //redrawLine();
}
```

A significant part to point out is that I commented redrawLine() out, as I was testing to see whether my code will crash again without that line in. And surprisingly it did – whenever a change in location was detected, the program crashed. So that told me that the problem is not with redrawLine(), it's with something else. So I commented "coordinates.add(currentLocation);" out and ran the program again. And it did work:



And looking back at the terminal when “coordinates.add(currentLocation)” was in Runnable, I saw this error:

```
E/AndroidRuntime: FATAL EXCEPTION: main
    Process: com.example.mapapp, PID: 6789
    java.lang.NullPointerException: Attempt to invoke virtual method 'boolean java.util.ArrayList.add(java.lang.Object)' on a null object reference
        at com.example.mapapp.MapsActivity$1.onLocationChanged(MapsActivity.java:94)
        at android.location.LocationManager$ListenerTransport._handleMessage(LocationManager.java:356)
        at android.location.LocationManager$ListenerTransport.-wrap0(Unknown Source:0)
        at android.location.LocationManager$ListenerTransport$1.handleMessage(LocationManager.java:236)
        at android.os.Handler.dispatchMessage(Handler.java:108)
        at android.os.Looper.loop(Looper.java:166)
        at android.app.ActivityThread.main(ActivityThread.java:7529) <1 internal call>
        at com.android.internal.os.Zygote$MethodAndArgsCaller.run(Zygote.java:245)
        at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:921)
```

I looked it up (https://stackoverflow.com/questions/28409089/nullpointerexception-when-adding-an-object-to-arraylist-in-android) and found that it could be because I didn’t define coordinates properly at the start and that was correct. I changed:

```
private ArrayList<LatLng> coordinates;
```

to

```
private ArrayList<LatLng> coordinates = new ArrayList<>();
```

And ran the program again, this time with “coordinates.add(currentLocation)” not commented out, but “redrawLine()” still commented out. The program doesn’t crash. So then I ran the program again with both not commented out (but I did comment out “mMap.addMarker(…)” in the onLocationChange()), and here were the results:

Finally it worked. The red marker signified my location and as I was moving, it was moving with me and the blue line was showing up to signify my previous path.

Now the final part is to calculate the distance, time and average pace of the run. Here is the code I wrote to get the distance ran and the time ran:

```
private static double distanceRan;
private static Location currentLocation;
private static LatLng currentLoc;
public long startTime = System.currentTimeMillis();
public long endTime;
private static long timeRan;
    .
    .
    .
private final LocationListener mLocationListener = new LocationListener() {
    @Override
    public void onLocationChanged(final Location location) {
        latitude = location.getLatitude();
        longitude = location.getLongitude();
        Location newLocation = new Location("");
        newLocation.setLatitude(latitude);
        newLocation.setLongitude(longitude);


        distanceRan = distanceRan + currentLocation.distanceTo(newLocation);
        currentLocation.setLongitude(longitude);
        currentLocation.setLatitude(latitude);

        currentLoc = new LatLng(latitude, longitude);
```

```
        coordinates.add(currentLoc);
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLoc, 16.0f));
        redrawLine();

    }
    .
    .
    .
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    currentLocation = new Location("");
    currentLocation.setLongitude(currentLocation.getLongitude());
    currentLocation.setLatitude(currentLocation.getLatitude());


}
```

For time: the program gets the current time when the map is opened and when the "STOP WORKOUT" button is pressed, the program once again gets the current time. The difference in time is how long the program has ran for. The "STOP WORKOUT" button code is as follows:

```
public void stopWorkout(View v){
    endTime = System.currentTimeMillis();
    timeRan=Math.round(endTime-startTime);

    Intent intent = new Intent(this, runStats.class);
    startActivity(intent);
}
```

For distance: In the onMapReady() – when the map is first opened – the program gets the latitude and longitude of the current location and sets them as latitude and longitude of Location variable that I called "currentLocation". Then when the change in location has been detected, the new coordinates are also recorded under new Location variable that I called "newLocation". Then using "currentLocation.distanceTo(newLocation)", the program calculates the distance between the two points. That distance is added to the distance total "distanceRan". After that, I set the coordinates of "newLocation" to "currentLocation", so when there is location change again, the distance can be calculated again.

But in order to check it works, I needed to create a new activity. This activity is going to output all the stats about the run. Here is how it looks:

And in order to be able to access “distanceRan” and “timeRan” from the map activity to the “runStats” activity, I added this code to the map activity:

```
public static double getDistanceRan(){
    return distanceRan;
}

public static double getTimeRan(){
    return timeRan;
}
```

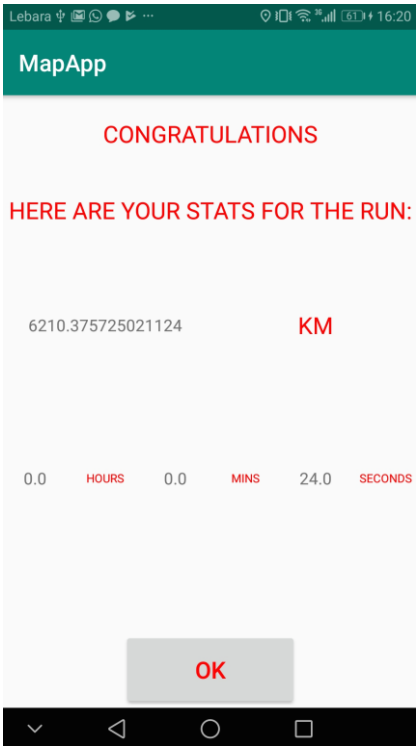
And in the new activity, the code looks like this:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_run_stats);


    TextView distanceRan = (TextView) findViewById(R.id.distance_ran);
    String distance = Double.toString(MapsActivity.getDistanceRan()/1000);
    distanceRan.setText(distance);

    double time = MapsActivity.getTimeRan();

    if (time==0.0){
        TextView timeRanHours = findViewById(R.id.time_ran_hours);
        timeRanHours.setText("0");
        TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
```

```
        timeRanMinutes.setText("0");
        TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
        timeRanSeconds.setText("0");
    }else{
        TextView timeRanHours = findViewById(R.id.time_ran_hours);
        String timeHours =
Double.toString(Math.round(MapsActivity.getTimeRan()/(60*60*1000)));
        timeRanHours.setText(timeHours);
        time -= Double.valueOf(timeHours)*60*60*1000;

        TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
        String timeMinutes = Double.toString(Math.round(time/(60*1000)));
        timeRanMinutes.setText(timeMinutes);
        time -= Double.valueOf(timeMinutes)*60*1000;

        TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
        String timeSeconds = Double.toString(Math.round(time/1000));
        timeRanSeconds.setText(timeSeconds);
    }


}
```

This seems like a lot of very confusing code but it's just some mathematical calculations to transform "timeRan" from milliseconds to hours, minutes and seconds, as that is more suitable to human understanding.

So I ran all this code and here is "runStats" activity:



I only ran it for 24 seconds as can be seen, and the only distance I covered was from one end of my room to the other. So I was very confused as to why it says I ran over 6,000 km. At first I thought I got the messurements wrong and didn't convert it properly and it was 6,000 mm or something, but then I remembered that when I ran the code a while back and I put the code, which gets current location, inside onMapReady() and told the program to show me that place on the map, it was in the middle of an ocean. So to test where the program thinks I am right when the map first opens, I added the following line of code to onMapRead():

```
System.err.println(currentLocation.getLatitude() + "," + currentLocation.getLongitude());
```

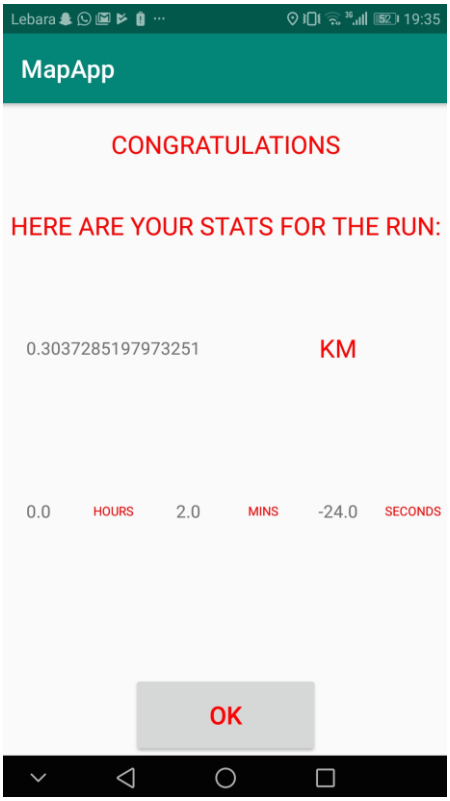
This line is going to print out the coordinates of "currentLocation". Here was the result (the highlighted bit):

So that's why the app thinks I'm travelling over 6,000 km in 24 seconds. The first value that "currentLocation" takes is (0,0) and then it jumps to my location. To solve this I had to be creative. I know that all coordinates, including my first "real" starting point, are recorded into ArrayList "coordinates". Therefore if after all the coordinates have been recorded and the distance ran (including the extra many kilometres from (0,0) to my real starting point) has been recorded, I can separately calculate the distance from (0,0) to the first stored element in "coordinates" and take that distance away from "distanceRan". And here is the code for that (this is inside stopWorkout(), which is called when the user presses "STOP WORKOUT" button):

```
double startingLat = coordinates.get(0).latitude;
double startingLng = coordinates.get(0).longitude;
Location startingPoint = new Location("");
startingPoint.setLatitude(startingLat);
startingPoint.setLongitude(startingLng);
Location start = new Location("");
start.setLatitude(0);
start.setLongitude(0);
double distanceToTakeAway = start.distanceTo(startingPoint);
distanceRan = distanceRan - distanceToTakeAway;
```

(Top two lines were aided by https://stackoverflow.com/questions/15264884/getting-latitude-and-longitude-from-latlng-object-in-android).

I've ran the code again:



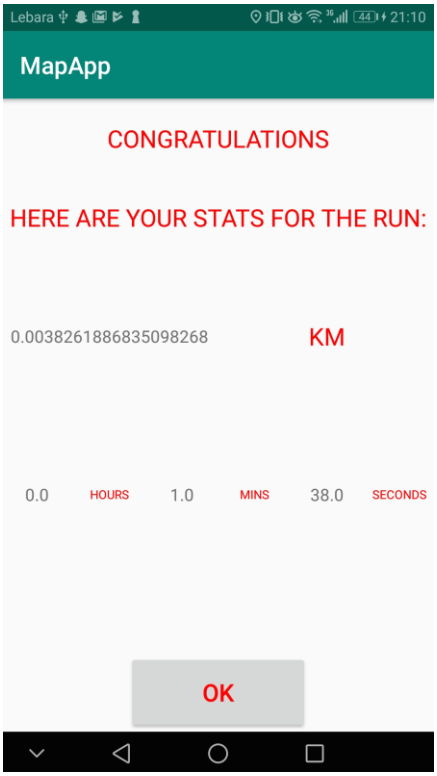
As can be seen, the problem with the distance was solved but the time measurements seem off. I was checking how accurate the time measurements were so timed the run on my stopwatch as well. At the end of the run the time on the stopwatch read 1minute 36seconds, which is the same as 2minutes minus 24seconds. So the time the app is showing is correct, but the way it is showing it wrong.

It was fairly difficult to understand what was going wrong with the time, so I decided to take the example I had above (1minute 36seconds) and run in by hand through my program. I've realised when doing that that there's a problem when rounding up. The new code looks like this:

```
if (time==0.0){
    TextView timeRanHours = findViewById(R.id.time_ran_hours);
    timeRanHours.setText("0");
    TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
    timeRanMinutes.setText("0");
    TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
    timeRanSeconds.setText("0");
}else{
    TextView timeRanHours = findViewById(R.id.time_ran_hours);
    if
(Math.round(MapsActivity.getTimeRan()/(60*60*1000))>MapsActivity.getTimeRan()/(60*60*1
000)){
        String timeHours =
Double.toString(Math.round(MapsActivity.getTimeRan()/(60*60*1000))-1);
        timeRanHours.setText(timeHours);
        time -= Double.valueOf(timeHours)*60*60*1000;
    }else if
```

```
(Math.round(MapsActivity.getTimeRan()/(60*60*1000))<MapsActivity.getTimeRan()/(60*60*1
000)){
        String timeHours =
Double.toString(Math.round(MapsActivity.getTimeRan()/(60*60*1000)));
        timeRanHours.setText(timeHours);
        time -= Double.valueOf(timeHours)*60*60*1000;
    }


    TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
    if (Math.round(time/(60*1000))>(time/(60*1000))){
        String timeMinutes = Double.toString(Math.round(time/(60*1000))-1);
        timeRanMinutes.setText(timeMinutes);
        time -= Double.valueOf(timeMinutes)*60*1000;
    }else if (Math.round(time/(60*1000))<(time/(60*1000))){
        String timeMinutes = Double.toString(Math.round(time/(60*1000)));
        timeRanMinutes.setText(timeMinutes);
        time -= Double.valueOf(timeMinutes)*60*1000;
    }


    TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
    String timeSeconds = Double.toString(Math.round(time/1000));
    timeRanSeconds.setText(timeSeconds);
}
```

To test it I ran it again. I tried running it for 1minute and 36seconds again. Result:



I tried to get it to 1minute 36seconds, but it overran by 2seconds, but the takeaway is that the time now works.

Stage 4: Review

What has been done:
The main achievement of this stage was to create a map activity, which would show the user's location as they are running and highlight the route that they are running, as they are running it. The activity would also calculate distance and time and using those two, it would calculate average pace. All three would be displayed on the page, which opens when the user clicks "STOP WORKOUT".

How it has been tested:
Each bit of code responsible for different sections: getting the map to show the current location of the user, get the map to highlight the route of the user as they are moving, calculate the distance of the journey in kilometres and calculate the time of the journey in hours, minutes and seconds – was tested straight after each section was coded.

How it meets the success criteria and user expectation:
This is the part of the app, which is responsible for tracking the user's workout and displaying it on the screen as the user is running. The stats surrounding the workout: distance, time and pace are all displayed after the workout.

Success criteria met/being met:
- Have a clear transition between pages. The user knows how to go both forward and back a page.
- Use GPS to track routes and work out distances ran during the workout.
- Use time measurements to work out and display paces and times.

Summary of the whole project as a prototype at this stage:
This is the final stage of the process complete. Unfortunately, the project doesn't have full functionality due to problems in **Stage 2: Saving of data into a text document**, however at least the other parts are working.

Stage 5: Final Testing

Each page in Stage 1 has been tested as it was programmed. Some of the tests have been talked about in "***Stage 1: The Questionnaire***". These, however, were mainly tests to make sure the page looks like it's supposed to on a mobile device. The following tests were done to test the functionality:

1. **First Page:** firstly tests were made to make sure the buttons were functional. The things I was making sure were working were that when a button is pressed, it goes of darker colour to its neighbouring buttons, and if another button was pressed afterwards, that button turned dark and the previously pressed button goes lighter. The functionality of the buttons behind the user interface was also tested, to make sure the correct Boolean variable became true when a button was pressed (e.g. if "5K" button was pressed, then (Boolean) fiveK = true and all other = false). So I've changed the code in "goToSecondPage()" function talked about earlier to the following:

```
public void goToSecondPage(View v) { //when submit button is pressed
    System.out.println(fiveK);
    System.out.println(tenK);
    System.out.println(eight);
    System.out.println(fifteen);
    System.out.println(Half);
    System.out.println(no);
}
```

   When the "5K" button was pressed, the order was as follows: true, false, false, false, false, false. Then I pressed the "800m" button and the order was: false, false, true, false, false, false. So the buttons were working correctly.

   I've talked about the tests for the type-in boxes briefly above. As talked about in the design section, here is the table with inputs, whether it should be accepted or rejected and result filled in:

Input	**Accept/Reject**	**Result**
0**hours** 15**minutes** 50**seconds** (5k entry)	Accept	Accepted
0**hours** 1**minute** 59.89**seconds** (800m entry)	Accept	Rejected
Hello **hours**	Reject	Rejected – there was no option of entering letters in
3.14 **hours** 15**minutes** 16**seconds**	Reject	Rejected – didn't allow me to enter a decimal
Paste in *If-* by Rudyard Kipling into **hours/minutes/seconds**	Reject	Rejected – didn't allow me to paste words in
Paste in the first 100 digits of pi into **hours/minutes/seconds**	Reject	Rejected

As can be seen one of the tests that was meant to be accepted, got rejected. I then changed that the "seconds" box can take in decimals so that solved that problem.

The main tests were trying to paste in any non-quantitative data into the box, which wasn't letting me. I have also tried to use the decimal point, as the keyboard has one. Trying to type in "3.14" wasn't letting me. Then I tried to paste in the first 100 digits of pi and it pasted it in without the decimal point (31415…). And other tests involved typing in unrealistic/wrong values (example 1: press "800m" button and for the PB boxes type in: 3h 14min 36sec. The value is unrealistic because to complete 2 laps of around a track in that time would include a couple of naps. Example 2: typing in a value bigger than 60 into any minutes and seconds boxes). The program handled the tests well and outputted a "Toast" (pop-up message) on the screen telling the user where the issue lies. Obviously the testing stage also involved inputting perfectly reasonable values and making sure it works.

2. **Second Page:** this isn't a massively exciting page and not a lot of tests were held for it. Since the page only has buttons as user interference, the same tests that were held for the buttons in the first page were held for these buttons. The result was also positive.

3. **Third Page:** this page is also very similar to the **First Page**. The buttons are exactly the same so I didn't test them as all the tests have already been done.

   The date type-in boxes undergo somewhat similar tests that the time type-in boxes do in the **First Page**. The boxes once again only allow numbers to be entered so entering a non-numerical value was impossible. Same with trying to type in a decimal. Checking that the values entered are in the right range is the easy bit. The tricky bit is that the user might enter a date that has already been. There isn't a command, which compares the dates so I had to write the code for that myself. Here is the code that I wrote:

```
public void noRaces(View v){
    TextView day = findViewById(R.id.page_3_question_1_day_enter);
    TextView month = findViewById(R.id.page_3_question_1_month_enter);
    TextView year = findViewById(R.id.page_3_question_1_year_enter);

    Date today = Calendar.getInstance().getTime(); //gets the current date


    SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
    String formattedDate = curFormater.format(today);

    //Checking the date entered is within a "valid" range
    if (Integer.parseInt(day.getText().toString())>31 || Integer.parseInt(day.getText().toString())<28 ||
            Integer.parseInt(month.getText().toString())>12 || Integer.parseInt(month.getText().toString())<1){
        Toast.makeText(this, "The date entered is invalid", Toast.LENGTH_LONG);
```

```
    }
    else{
        //checking the date entered hasn't already passed:
        if (Integer.parseInt(formattedDate.substring(5, 9))==Integer.parseInt(year.getText().toString())){
            if (Integer.parseInt(formattedDate.substring(3, 4))<Integer.parseInt(month.getText().toString())){
                Toast.makeText(this,"The date entered has already passed", Toast.LENGTH_LONG).show();
            }
            else if (Integer.parseInt(formattedDate.substring(3, 4))>Integer.parseInt(month.getText().toString())){
                Intent intent = new Intent(this, fourth_page.class);
                startActivity(intent);
            }
            else{
                if (Integer.parseInt(formattedDate.substring(0, 1))>Integer.parseInt(day.getText().toString())){
                    Intent intent = new Intent(this, fourth_page.class);
                    startActivity(intent);
                }
                else{
                    Toast.makeText(this,"The date entered has already passed", Toast.LENGTH_LONG).show();
                }
            }
        }
        else if (Integer.parseInt(formattedDate.substring(5, 9))>Integer.parseInt(year.getText().toString())){
            Intent intent = new Intent(this, fourth_page.class);
            startActivity(intent);
        }
        else{
            Toast.makeText(this,"The date entered has already passed", Toast.LENGTH_LONG).show();
        }
    }



}
```

I used https://stackoverflow.com/questions/8654990/how-can-i-get-current-date-in-android to learn about how to get the current date in Android Studio.

As can be seen, the code consists of a lot of iteration loops. The first if statement checks that the date entered is within a valid range. If it is then the next if (){}...else if(){}...else{} loop check whether the year entered is the same as the year right now, higher, or lower. If higher, immediately accepts the entry. If lower, immediately denies the entry, making a pop-up screen asking the user to review the entry. If it is the same, the program does the same as it did for the year to the month entry. If it is the same, the program does the same for the day entry. The current day's date is saved as a String in the "dd/mm/yyyy" format. So in order to compare the numbers, I had to use Integer.parseInt(String) and substring.

I checked the code with the inputs I made in the design section:

Input	**Accept/Reject**	**Result**
dd: 17; **mm:** 12; **yyyy:** 2021	Accept	Accepted
The day's date	Reject	Rejected
dd: 17; **mm:** 12; **yyyy:** 1969	Reject	Rejected
dd: 33	Reject	Rejected
Paste in *If-* by Rudyard Kipling into **dd/mm/yyyy**	Reject	Rejected – unable to paste in words

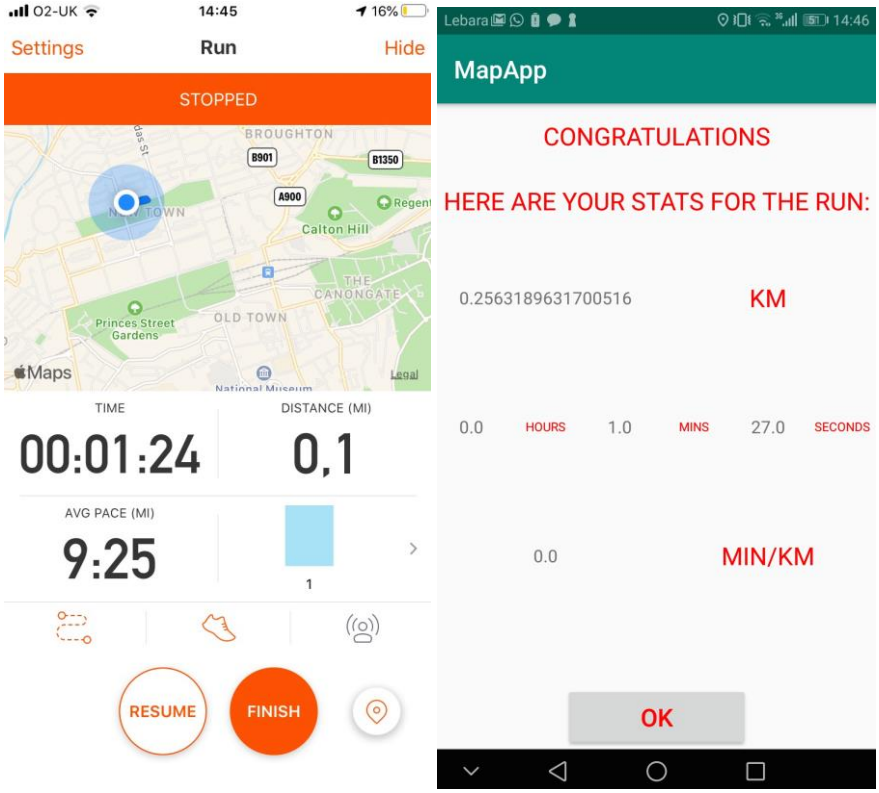
Paste in the first 100 digits of pi into **dd/mm/yyyy**	Reject	Rejected

The results completely match the predicted outputs, so it was a success.

4. **Fourth Page:** in this page, the user selects where they usually live and train. I would say the hardest part about this was the actual programming and as long as it is functional, the testing couldn't really bring any negative result. If the user clicks on a country, the city list must change to just being the cities in that country.

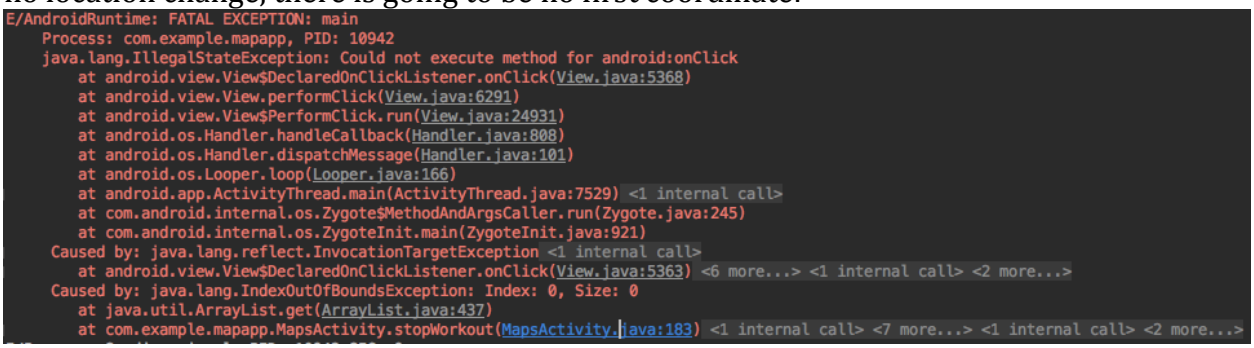
Stage 2 is the stage responsible for saving data to and reading data from files on the user's device. This stage cannot really be tested for robustness, only for functionality: it either works or it doesn't. If it's functional, it is already robust.

Stage 4 was the stage that created the actual map, which the user can use to track the workout as they are doing it. Many tests have occurred during the development stage to check the functionality. So there is only one more test that can be made on this stage: the accuracy test. I took two devices – one had Strava turned on, the other one had my app turned on. The two were turned on at the same time as I started my mini workout. Both apps were also stopped at the same time. Since Strava is a world wide used app, I could confidently say that having the same distance and time measurements as them would be considered accurate. So here were the results:



Strava (on the left) vs my app (on the right)

There is a minor problem however when using the map: if you click “START WORKOUT”, but then don’t change location and click “STOP WORKOUT”, it crashes. That’s because part of the code is getting the distance from (0,0) to first recorded coordinate, which if there is no location change, there is going to be no first coordinate:



Checklist

Using the checklist I created back in the design section, I have gone through the program and put a Y in the checklist if the part of program works or N – if it doesn’t.

Action to Test	Does it work? Y/N
Check the functionality of each type in box – make sure it only takes in correct and reasonable values.	Y
Pressing “SUBMIT” button (or other buttons with similar functionality) checks the validity of the information entered by the user and takes the user to the next page.	Y
Saving the information that the user entered into a text file, including dates, distances and locations.	Y
Being able to access the text file, which stores user’s answer to the questions later in the program.	Y
Ability to go both forward and back a page.	Y
Have side menu navigation at the main page with many other pages linked to it.	N
Pressing any option from the side menu navigation takes the user to different pages, specifically assigned to the user’s choice.	N
Pressing “START WORKOUT” button takes the user to the GPS window.	Y
Use GPS to track routes and work out distances ran during the workout.	Y
Use time measurements to work out and display paces and times.	Y
Pressing “STOP WOTKOUT” button takes the user to a page, which outputs the statistics about the user’s run.	Y
Saving those statistics to the history archive text document, along with the date.	Y
Checking those statistics against the achievements and replace the achievements if the user has bettered their previous personal best.	Y

This is going to be my final draft of the program, therefore I’m ready to get it tested by the stakeholders.

Stage 6: Stakeholder testing

Since downloading Android Studio and getting it to run on a device is probably a hustle for people who haven't previously worked with Android Studio, I decided to give them the program on my device. I'm in contact with three out of four stakeholders on daily basis, so that was simple. For Cameron, since he is away in university in England, I sent him a video of the program running and asked him the same questions as I asked all my stakeholders:

1. What did you think about the user interface? Did you think it was user-friendly and pleasant to the eye?
2. Did you think the questions about you were clearly constructed? Did you have any issues trying to answer those questions?
3. What did you think about the GPS tracking system, while you were completing the workout?
4. Did you find it useful to have a page after the workout, which summarises the whole workout for you?
5. What did you think of the app as a whole? How does it compare to other apps that you use daily? Anything you would have wanted me to do differently?

Responses:

Cameron:

"

1. It was really nice. I was expecting way worse to be honest. It wasn't bad but it wasn't as appealing to the eye as some top trending apps out there. (Sorry)
2. I thought the questions were good – definitely something you would expect from the "get in shape" sort of apps. Does the app use the information about the races I have in the future somewhere? Like will it warn me that I have a race in a week so I should back off training or something?
3. I thought that was mighty impressive. I like how the red pointer constantly points to my location and the route gets highlighted.
4. Yea I think that's definitely a necessity in those kind of apps.
5. That was a great coding job. I definitely never could have done that. As I said before, it's not the same as some leading apps like strava in terms of looks but functionality is getting there. The only thing I'd add is: seeing the distances and time and pace while I'm running. Other than that, great job!

"

I spoke to everyone else in person, so I conducted a speaking interview afterwards. Here is roughly what they said:

Thomas:

1. I like the fact that it uses red a lot, makes it seem more interactive. Overall I'd say it was a very well designed gui.
2. Yes I liked that part.
3. I was of kind expecting it to look like that. It is fairly similar to the one in Strava, but it's missing some functionality, like a recenter button and having the app read out when you have passed one kilometre.
4. Yes that's definitely interesting and important to see. I would also however output those of the screen as I am running.
5. I thought the app was great – some bits I think you could improve on but overall very good job.

Alice:

1. Yes, I thought it was very easy to use and it was very clear what each button was doing, so no complaints there.
2. No, that part I found easy to understand.
3. I liked it a lot. The only thing I would say is when I was running and tried to zoom in on other parts of the map, after a couple of seconds it would jump back to my current location. It's not a very important thing, just thought I'd let you know.
4. Yes, I like to see how much I've ran (distance) because I need to run enough to increase my fitness levels. The only thing I'd add would be showing the distance before I press stop workout. Because I want to see how much more I have.
5. I think it was done very well and I'm very impressed because I can say from previous experience that it is hard to work with Android Studio.

Elsa:

1. I think it is very appealing to the eye. I like it.
2. The questions were clear. I couldn't really relate because I don't focus on any distance and I don't have any PBs nor races coming up so I was mainly just trying to crash your program. And I was very impressed, it's pretty resilient – I couldn't crash it.
3. It was great. I liked how it draws the blue line, as you are moving.
4. Yes, very. I only care about the distance to be completely honest, but I imagine more serious runners like yourself care about speed and all that. I would however want to see the distance as I am running without having to stop the workout. But I really liked it.
5. Really impressive. I really enjoyed testing it out.

Review of the stakeholder testing:

It seems like generally everyone enjoyed testing the app. Everyone seemed to have positive comments about the user interface and interaction with the app through means of buttons and type-in boxes. There were obviously a few comments that it's not quite at the level of professional apps, but I think that will go under my things to improve list. The GPS tracking activity seemed to also be liked amongst the testers, but the big thing that every single stakeholder picked up on is the fact that it doesn't show distances or time while they are doing the workout. But the run overview page was described as important and interesting. This gave me a good understanding of what could be revised in the future versions of the app.

Evaluation

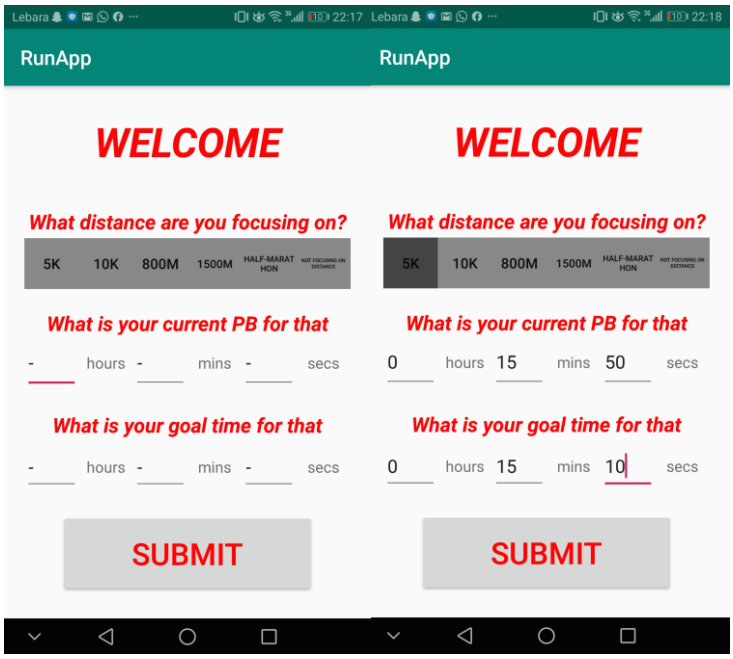
Criteria Met

Success Criteria:

Criteria	Criteria Met? YES/NO
Reasonably good and user-friendly graphical user interface.	YES
Easy and clear to answer questions to find out information about the user.	YES
Have a page, which asks the user about their preferred distance to train and their personal bests for that distance.	YES
Saving the user's preferred distance to train and their personal bests to a text file.	YES
Being able to access the text file, which stores user's answer to the questions later in the program.	YES
Have a clear transition between pages. The user knows how to go both forward and back a page.	YES
Have a page, which asks the user whether they have a race coming up.	YES
Have a page, which asks the user about when their future races are.	YES
Saving the dates for user's future races.	YES
Have a page, which asks the user where they usually live, and train.	YES
Saving the destination of where the user usually lives and trains in a text document.	YES
Have side menu navigation at the main page with many other pages linked to it.	**NO**
Have a page, which outputs the user's workout history.	YES
Have a page, which outputs user's achievements: longest distance/longest time/fastest average speed ran.	YES
Have a page, which allows manual entry of workouts.	YES
Use GPS to track routes and work out distances ran during the workout.	YES
Use time measurements to work out and display paces and times.	YES
Clearly show good routes to the user.	YES
Suggest good workouts to the user to get to their goal faster	YES

Evidence:

Upon opening the app, the following page is seen:



This page is linked to the success criteria in the following ways:

1. Reasonably good and user-friendly interface. **Evidence**: the app has been given to the stakeholders and four out of four said that they found the UI very easy to use and quite appealing to the eye.

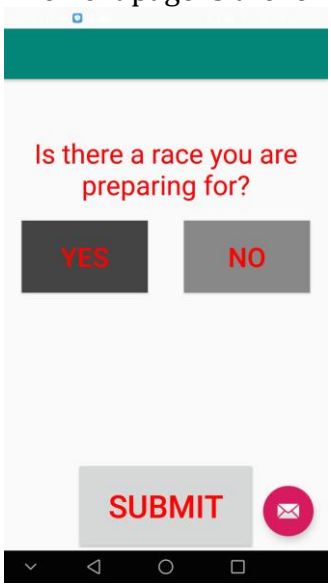
2. Easy and clear to answer questions to find out information about the user. **Evidence**: the app has been given to the stakeholders and all of them commented that questions were an integral part of the app and a very useful way of getting information about the user.

3. Have a page, which asks the user about their preferred distance to train and their personal bests for that distance. **Evidence:** there are three questions on the page: "what distance are you focusing on?", "What is you current PB for that", and "What is your goal time for that".

4. Saving the user's preferred distance to train and their personal bests to a text file. **Evidence:** the app makes a pop-up in the event that the information is saved saying "Saved to File". As well this information is going to be used later to show the user suggested workouts for their preferred distance.

5. Have a clear transition between pages. The user knows how to go both forward and back a page. **Evidence:** the app has been given to the stakeholders and they have replied that they didn't struggle to navigate through the app.

The next page is the following:



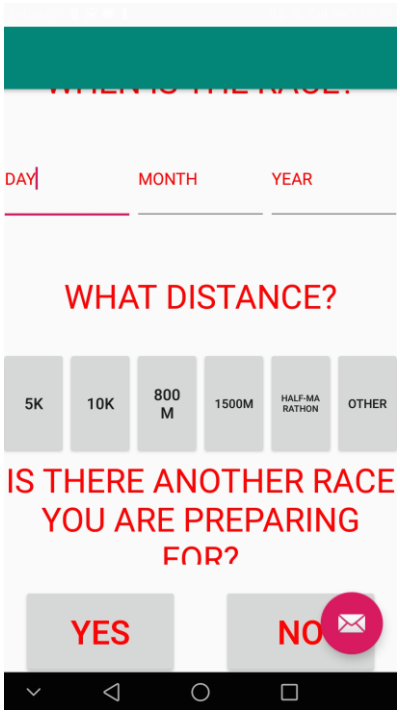
This page is linked to the success criteria in the following ways:

1. Have a page, which asks the user whether they have a race coming up. **Evidence:** the page contains a single question with a yes or no answer saying "Is there a race you are preparing for?". Upon clicking yes, the user is taken to a page where they enter the details about their future race. Pressing no, takes the user to a different page with the next question.

2. Easy and clear to answer questions to find out information about the user. **Evidence**: the app has been given to the stakeholders and all of them commented that questions were an integral part of the app and a very useful way of getting information about the user.

3. Have a clear transition between pages. The user knows how to go both forward and back a page. **Evidence:** the app has been given to the stakeholders and they have replied that they didn't struggle to navigate through the app.

Upon clicking “YES”, the user is taken to the following page:



This page is linked to the success criteria in the following ways :

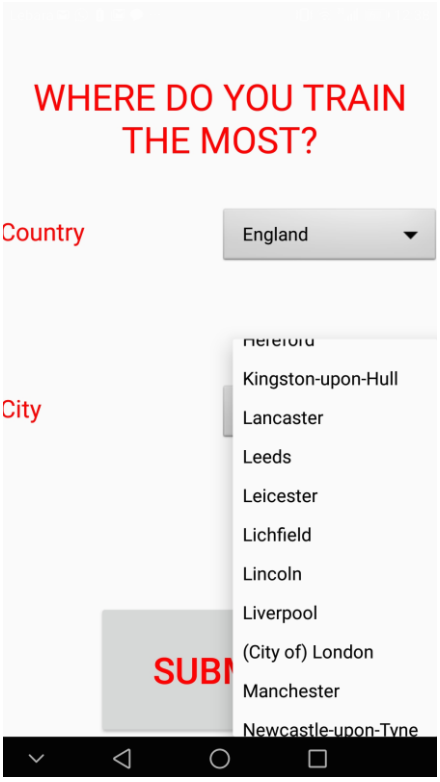
1. Have a page, which asks the user about when their future races are. **Evidence:** there are three questions on the page: “When is your race?”, “What Distance?”, and “Is there another race you are preparing for?”.

2. Saving the dates for user’s future races. **Evidence:** the app makes a pop-up in the event that the information is saved saying “Saved to File”.

3. Easy and clear to answer questions to find out information about the user. **Evidence**: the app has been given to the stakeholders and all of them commented that questions were an integral part of the app and a very useful way of getting information about the user.

4. Have a clear transition between pages. The user knows how to go both forward and back a page. **Evidence:** the app has been given to the stakeholders and they have replied that they didn’t struggle to navigate through the app.

If the user clicked “NO” however (or if the user click “NO” in the page above), the user is taken to this page:



This page is linked to the success criteria in the following ways :

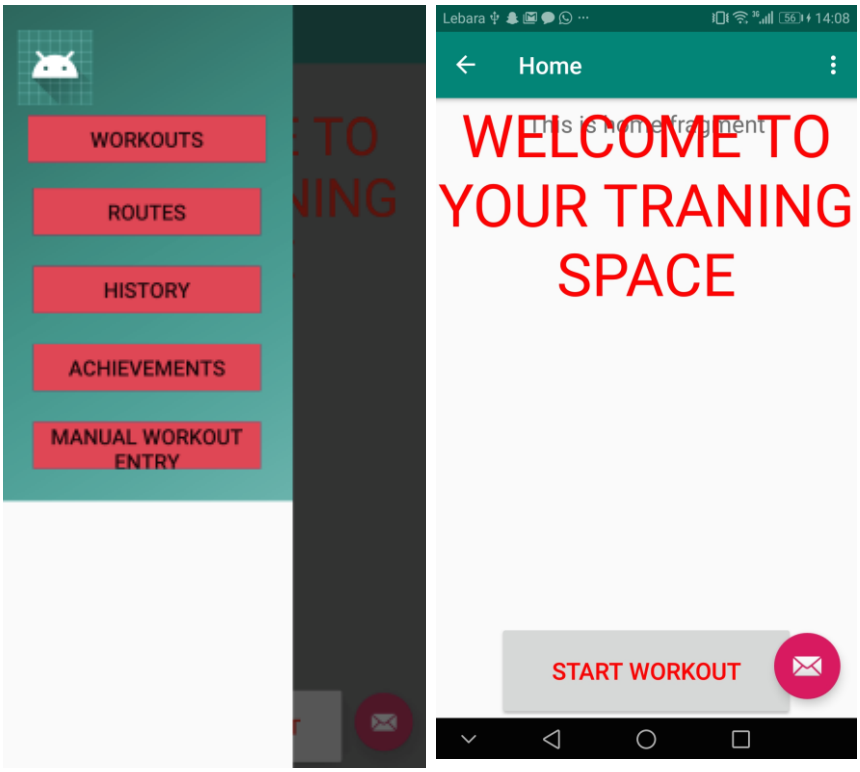
1. Have a page, which asks the user where they usually live, and train. **Evidence:** there is one question on this page “Where do you train the most?”, to which the answer consists of two parts: Country and City. When the user chooses a country from the drop down menu (only includes countries in the UK), the city drop down menu options become only the cities from that country.

2. Saving the destination of where the user usually lives and trains in a text document. **Evidence:** the app makes a pop-up in the event that the information is saved saying “Saved to File”. As well this information is going to be used later to show the user good routes in their area.

3. Easy and clear to answer questions to find out information about the user. **Evidence**: the app has been given to the stakeholders and all of them commented that questions were an integral part of the app and a very useful way of getting information about the user.

4. Have a clear transition between pages. The user knows how to go both forward and back a page. **Evidence:** the app has been given to the stakeholders and they have replied that they didn't struggle to navigate through the app.

After clicking the "SUBMIT" button in the previous page, the user is taken to the "training space page":



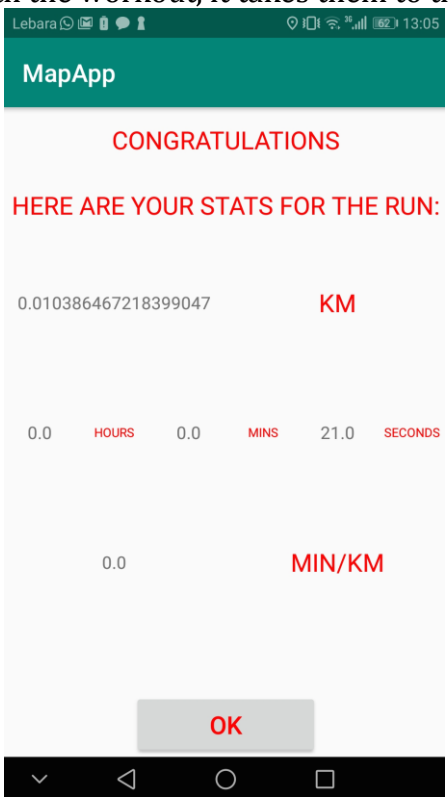
How it was supposed to look like (on the left) and how it actually looks (on the right)

Unfortunately the side menu navigation problem to this day remains unsolved.

When the user clicks "START WORKOUT", the following shows up:

And as the user start moving the following shows up:

And after the user is done with the workout, it takes them to the following page:



This page is linked to the success criteria in the following ways:

1. Use GPS to track routes and work out distances ran during the workout. **Evidence:** As can be seen from the second image, the route the user is running gets highlighted with a blue line, and as can be seen in the third image, the distance ran is displayed.

2. Use time measurements to work out and display paces and times. **Evidence:** As can be seen from the third image, time (in hours, minutes and seconds) as well as pace (in minutes per kilometre) are displayed after the workout.

3. Reasonably good and user-friendly interface. **Evidence**: the app has been given to the stakeholders and four out of four said that they found the UI very easy to use and quite appealing to the eye.

4. Have a clear transition between pages. The user knows how to go both forward and back a page. **Evidence:** the app has been given to the stakeholders and they have replied that they didn't struggle to navigate through the app.

Limitations

The biggest limitation is that the side menu navigation is not functional. That limits the aid to the user of navigating through the app and better understanding the structure of the app. More importantly, it limits some functionality of the app like history, achievements, etc. This is definitely something I can fix given more time.

Another limitation that every stakeholder has pointed out is the fact that the app doesn't show distances, times, nor paces when the user is doing their workout. I understand how that can be an important factor in everyone's workout. Also well suggested by Thomas, maybe adding a voice over to the app, which notifies the user when they have ran a certain distance (every 1k for example), or notify the user if they are slowing down. This helps the user have the phone locked and put away in the pocket and still know how far they are into their run.

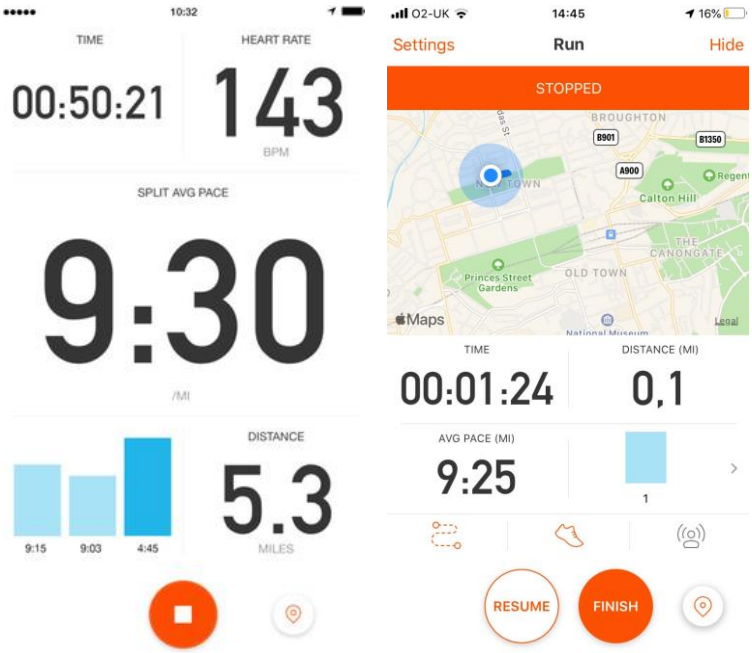
While on the topic of improving the map activity, one other limitation is the following scenario: if the user clicks "START WORKOUT", but then doesn't change location and clicks "STOP WORKOUT", the app crashes. That's because part of the code is getting the distance from (0,0) to first recorded coordinate, which, if there is no location change, there is going to be no first coordinate.

The final limitation to this app in my opinion is the looks. One can definitely tell a difference between a professional app and my app, because my app looks way worse. I talk about how I think this could be solved along with how every other limitation could be solved in the next section.

How to solve these limitations/further development:

In my opinion it's very hard to involve both a map and output distance, time and pace of the run due to the complexity in structure. I haven't looked into it but from what I have learnt from working with android studio, it is impossible to add a <TextView> on a <fragment> like a map fragment, so I don't know whether that's possible. Looking at the approach Strava took, when the user clicks the button to start a workout, it takes them to a page where it only outputs distance, time and pace (on the left) and then when the user clicks stop, the app takes them to a new page (on the right), where it displays the map with the user's route highlighted and takes up half of the page and the other half are the distances, times and paces from the run. The user then has an option to press resume if they want to resume their run or finish if they want to stop.

I think this would be the best solution if I were to try and make it better.



The other limitation is if the user clicks stop soon after clicking start and not changing their location in between. This can be solved very easily with the following code:

```
if (coordinates.size==0){
     distanceRan = 0;
}else{
  .
  .
  .
}
```

This signifies that “coordinates” is empty and the program shouldn’t get its first element and just set the distance total to zero.

To tackle my last limitation and see how I can make the app look more professional, I decided to look at top leading apps. Here is what I noticed: for starters they use a much broader variety of colour than just red (#f00) and light green (provided by Android Studio), which were the only colours I used, so that’s definitely something to improve on. The other thing is the shapes and sizes of texts, buttons, boxes, etc. What I mean by that is if to look closer to even the screenshot above, you can see that Strava uses different text sizes for different writings in such a way that it almost seems unsymmetrical, which is definitely better than the same monochromatic tone that I set in my app with my writing. Also different boxes have different sizes, but the biggest thing is the usage of round buttons. I think that makes the app look way less formal than rectangular buttons. However those top trending apps hire professional psychiatrists who specifically design how the apps should look in order to draw the user back to the app. I can’t quite get it to that level but I think there is a lot I can do to make it look better.

Maintenance

The only difficulty about spreading the app – and I ran into this when I was trying to get the stakeholders' approval for the development – is that people who are not technologically inclined do not have Android Studio just downloaded on their laptop, therefore they can't just run the folder of code that I will send them. I think the easiest way to get the app to other people by means of downloading an app. This link: https://help.dropsource.com/docs/documentation/after-dropsource/publishing-your-app/submitting-an-android-app-to-google-play/ contains an article of how to upload your Android Studio project to play store. This is definitely something I would want to try to do after perfecting my app. This is also by far the easiest way for any person with a smartphone to get hold of my app.

That sentence actually brings up another limitation I've talked about right at the beginning: people who do not own an Android operated phone would not be able to view my app. This is also solvable as talked about in https://www.devteam.space/blog/how-to-convert-an-android-app-to-ios/. The process is apparently very simple if done by a tool called "Mechdome". This is once again something that I haven't looked into a lot but will be interesting to go into further detail in the future.

The code that I wrote is split into two languages: Java and XML. Each activity has one of each. XML is responsible for the design and the looks, while Java is responsible for the content. I have talked about and explained a lot of the important code in the development section but the following section will contain all the code that is needed to run this app.

Final Code

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:id="@+id/overall_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="10dp"
        android:layout_marginBottom="20dp"
        android:orientation="vertical"
        app:layout_constraintTop_toTopOf="parent"
        tools:layout_editor_absoluteX="1dp">
        <LinearLayout
            android:id="@+id/welcome_message_layout"
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:layout_margin="10dp"
            android:orientation="vertical"
            app:layout_constraintTop_toTopOf="parent"
            tools:layout_editor_absoluteX="1dp">
            <TextView
                android:id="@+id/welcome"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginStart="8dp"
                android:layout_marginTop="8dp"
                android:layout_marginEnd="8dp"
                android:textAlignment="center"
                android:textSize="40sp"
                android:text="WELCOME"
                android:textStyle="bold|italic"
                android:textColor="#f00"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent" />
        </LinearLayout>

        <LinearLayout
            android:id="@+id/question1"
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:layout_margin="10dp"
            android:orientation="vertical"
            app:layout_constraintTop_toTopOf="parent"
            tools:layout_editor_absoluteX="1dp">
            <TextView
                android:id="@+id/question1_text"
                android:layout_width="match_parent"
                android:layout_height="30dp"
                android:textAlignment="center"
                android:text="What distance are you focusing on?"
                android:textStyle="bold|italic"
                android:textColor="#f00"
                android:textSize="20dp" />

            <LinearLayout
                android:id="@+id/question1_buttons_layout"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:gravity="center_horizontal|center_vertical"
```

```
            android:orientation="horizontal">

            <Button
                android:id="@+id/question1_button1"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="5k"
                android:onClick="button1Pressed"/>

            <Button
                android:id="@+id/question1_button2"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="10k"
                android:onClick="button2Pressed"/>

            <Button
                android:id="@+id/question1_button3"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="800m"
                android:onClick="button3Pressed"/>

            <Button
                android:id="@+id/question1_button4"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="1500m"
                android:textSize="11dp"
                android:onClick="button4Pressed"/>

            <Button
                android:id="@+id/question1_button5"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="Half-Marathon"
                android:textSize="8dp"
                android:onClick="button5Pressed"/>

            <Button
                android:id="@+id/question1_button6"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="Not focusing on distance"
                android:textSize="5dp"
                android:onClick="button6Pressed"/>

        </LinearLayout>

    </LinearLayout>

    <!--Question 2 stuff starts here-->
    <LinearLayout
        android:id="@+id/question2"
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:layout_margin="10dp"
        android:layout_marginTop="20dp"
        android:orientation="vertical">
        <TextView
            android:id="@+id/question2_text"
            android:layout_width="match_parent"
            android:layout_height="50dp"
            android:textAlignment="center"
            android:text="What is your current PB for that distance?"
```

```
            android:textStyle="bold|italic"
            android:textColor="#f00"
            android:textSize="20sp"
            android:layout_weight="1"
            />
        <LinearLayout
            android:id="@+id/question2_buttons_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <EditText
                android:id="@+id/question2_hours_type_in"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="-"
                android:inputType="number"
                android:layout_weight="1"
                />
            <TextView
                android:id="@+id/question2_hours_text"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="hours"
                android:gravity="center"
                android:textSize="15sp"
                android:layout_weight="1"
                />
            <EditText
                android:id="@+id/question2_minutes_type_in"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="-"
                android:inputType="number"
                android:layout_weight="1"
                />
            <TextView
                android:id="@+id/question2_minutes_text"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="mins"
                android:gravity="center"
                android:textSize="15sp"
                android:layout_weight="1"
                />
            <EditText
                android:id="@+id/question2_seconds_type_in"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="-"
                android:inputType="number"
                android:layout_weight="1"
                />
            <TextView
                android:id="@+id/question2_seconds_text"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="secs"
                android:gravity="center"
                android:textSize="15sp"
                android:layout_weight="1"
                />
        </LinearLayout>

    </LinearLayout>

    <!--Question 3 stuff starts here-->
    <LinearLayout
        android:id="@+id/question3"
        android:layout_width="match_parent"
        android:layout_height="80dp"
```

```
            android:layout_margin="10dp"
            android:layout_marginTop="20dp"
            android:orientation="vertical">
            <TextView
                android:id="@+id/question3_text"
                android:layout_width="match_parent"
                android:layout_height="30dp"
                android:textAlignment="center"
                android:text="What is your goal time for that distance?"
                android:textStyle="bold|italic"
                android:textColor="#f00"
                android:textSize="20sp"
                />
            <LinearLayout
                android:id="@+id/question3_buttons_layout"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:orientation="horizontal">
                <EditText
                    android:id="@+id/question3_hours_type_in"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="-"
                    android:inputType="number"
                    android:layout_weight="1"
                    />
                <TextView
                    android:id="@+id/question3_hours_text"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="hours"
                    android:gravity="center"
                    android:textSize="15sp"
                    android:layout_weight="1"
                    />
                <EditText
                    android:id="@+id/question3_minutes_type_in"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="-"
                    android:inputType="number"
                    android:layout_weight="1"
                    />
                <TextView
                    android:id="@+id/question3_minutes_text"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="mins"
                    android:gravity="center"
                    android:textSize="15sp"
                    android:layout_weight="1"
                    />
                <EditText
                    android:id="@+id/question3_seconds_type_in"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="-"
                    android:inputType="number"
                    android:layout_weight="1"
                    />
                <TextView
                    android:id="@+id/question3_seconds_text"
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:text="secs"
                    android:gravity="center"
                    android:textSize="15sp"
                    android:layout_weight="1"
                    />

            </LinearLayout>
```

```
        </LinearLayout>

        <!--Submit button-->
        <LinearLayout
            android:id="@+id/submit_button_layout"
            android:layout_width="match_parent"
            android:layout_height="300dp"
            android:layout_margin="10dp"
            android:gravity="top|center_horizontal"
            android:orientation="vertical">

            <Button
                android:id="@+id/submit_button"
                android:layout_width="250dp"
                android:layout_height="80dp"
                android:gravity="center_horizontal|center_vertical"

                android:onClick="goToSecondPage"
                android:text="SUBMIT"
                android:textColor="#f00"
                android:textSize="30sp" />
            <!--android:layout_marginTop="220dp"
                android:layout_marginLeft="50dp"-->

        </LinearLayout>



    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java:

```
package com.example.runapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.instinctcoder.readwritefile.FileHelper;

import java.io.IOException;

public class MainActivity extends AppCompatActivity {

    private int STORAGE_PERMISSION_CODE = 1;

    public static boolean fiveK;
    public static boolean tenK;
    public static boolean eight;
    public static boolean fifteen;
    public static boolean half;
    public static boolean no;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);


        EditText question2_time_hours = (EditText)
findViewById(R.id.question2_hours_type_in);
        EditText question2_time_minutes = (EditText)
findViewById(R.id.question2_minutes_type_in);
        EditText question2_time_seconds = (EditText)
findViewById(R.id.question2_seconds_type_in);

        Button button = (Button) findViewById(R.id.question1_button1);
        Button button2 = (Button) findViewById(R.id.question1_button2);
        Button button3 = (Button) findViewById(R.id.question1_button3);
        Button button4 = (Button) findViewById(R.id.question1_button4);
        Button button5 = (Button) findViewById(R.id.question1_button5);
        Button button6 = (Button) findViewById(R.id.question1_button6);


        button.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void goToSecondPage(View v) { //when submit button is pressed
        EditText question2_time_hours = (EditText)
findViewById(R.id.question2_hours_type_in);
        EditText question2_time_minutes = (EditText)
findViewById(R.id.question2_minutes_type_in);
        EditText question2_time_seconds = (EditText)
findViewById(R.id.question2_seconds_type_in);

        if (ContextCompat.checkSelfPermission(MainActivity.this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            if (!fiveK && !tenK && !eight && !fifteen && !half && !no) {
                Toast.makeText(this, "CHOOSE A DISTANCE", Toast.LENGTH_SHORT).show();
            } else {
                if (fiveK) {

                    if (Integer.parseInt(question2_time_hours.getText().toString()) >
1 ||

Integer.parseInt(question2_time_minutes.getText().toString()) > 60 ||

Integer.parseInt(question2_time_seconds.getText().toString()) > 60) {
                        Toast.makeText(this, "THE DETAILS ENTERED ARE INCORRECT",
Toast.LENGTH_LONG).show();
                    } else {
                        if (FileHelper.writeInternalStorage("data.txt", "5K") &&
FileHelper.writeInternalStorage("data.txt", question2_time_hours.getText().toString())
&& FileHelper.writeInternalStorage("data.txt",
question2_time_minutes.getText().toString()) &&
FileHelper.writeInternalStorage("data.txt",
question2_time_seconds.getText().toString())) {
                            Intent intent = new Intent(this, secondPage.class);
                            startActivity(intent);
                            Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                        } else {
                            Toast.makeText(MainActivity.this, "ERROR!!",
Toast.LENGTH_SHORT).show();
                        }
                    }
                } else if (tenK) {
                    if (FileHelper.writeInternalStorage("data.txt", "10K") &&
FileHelper.writeInternalStorage("data.txt", question2_time_hours.getText().toString())
```

```
&& FileHelper.writeInternalStorage("data.txt",
question2_time_minutes.getText().toString()) &&
FileHelper.writeInternalStorage("data.txt",
question2_time_seconds.getText().toString())) {
                        Intent intent = new Intent(this, secondPage.class);
                        startActivity(intent);
                        Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                    } else {

Toast.makeText(MainActivity.this,"ERROR!!",Toast.LENGTH_SHORT).show();
                    }
                } else if (eight) {
                    if (FileHelper.writeInternalStorage("data.txt", "800") &&
FileHelper.writeInternalStorage("data.txt", question2_time_hours.getText().toString())
&& FileHelper.writeInternalStorage("data.txt",
question2_time_minutes.getText().toString()) &&
FileHelper.writeInternalStorage("data.txt",
question2_time_seconds.getText().toString())) {
                        Intent intent = new Intent(this, secondPage.class);
                        startActivity(intent);
                        Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                    } else {

Toast.makeText(MainActivity.this,"ERROR!!",Toast.LENGTH_SHORT).show();
                    }
                } else if (fifteen) {
                    if (FileHelper.writeInternalStorage("data.txt", "1500") &&
FileHelper.writeInternalStorage("data.txt", question2_time_hours.getText().toString())
&& FileHelper.writeInternalStorage("data.txt",
question2_time_minutes.getText().toString()) &&
FileHelper.writeInternalStorage("data.txt",
question2_time_seconds.getText().toString())) {
                        Intent intent = new Intent(this, secondPage.class);
                        startActivity(intent);
                        Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                    } else {

Toast.makeText(MainActivity.this,"ERROR!!",Toast.LENGTH_SHORT).show();
                    }
                } else if (half) {
                    if (FileHelper.writeInternalStorage("data.txt", "HALF") &&
FileHelper.writeInternalStorage("data.txt", question2_time_hours.getText().toString())
&& FileHelper.writeInternalStorage("data.txt",
question2_time_minutes.getText().toString()) &&
FileHelper.writeInternalStorage("data.txt",
question2_time_seconds.getText().toString())) {
                        Intent intent = new Intent(this, secondPage.class);
                        startActivity(intent);
                        Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                    } else {

Toast.makeText(MainActivity.this,"ERROR!!",Toast.LENGTH_SHORT).show();
                    }
                } else if (no) {
                    if (FileHelper.writeInternalStorage("data.txt", "NONE")) {
                        Intent intent = new Intent(this, secondPage.class);
                        startActivity(intent);
                        Toast.makeText(MainActivity.this, "Saved to file",
Toast.LENGTH_SHORT).show();
                    } else {

Toast.makeText(MainActivity.this,"ERROR!!",Toast.LENGTH_SHORT).show();
                    }
                }
            }
        } else {
            requestStoragePermission();
```

```
        }


    }

    private void requestStoragePermission() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
            //if user denied the permission before but tries to access it again
            new AlertDialog.Builder(this)
                    .setTitle("Permission needed")
                    .setMessage("This permission is needed to save data into a text
file")
                    .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                        }
                    })
                    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    })
                    .create().show();
        } else {
            //requesting permission
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Toast that permission is granted
            } else {
                //Toast that permission wasn't granted
            }
        }
    }

    public void button1Pressed(View v) {
        fiveK = true;
        tenK = false;
        eight = false;
        fifteen = false;
        half = false;
        no = false;

        Button button = (Button) findViewById(R.id.question1_button1);
        Button button2 = (Button) findViewById(R.id.question1_button2);
        Button button3 = (Button) findViewById(R.id.question1_button3);
        Button button4 = (Button) findViewById(R.id.question1_button4);
        Button button5 = (Button) findViewById(R.id.question1_button5);
        Button button6 = (Button) findViewById(R.id.question1_button6);


        button.setBackgroundColor(Color.DKGRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }
```

```
public void button2Pressed(View v2) {
    tenK = true;
    fiveK = false;
    eight = false;
    fifteen = false;
    half = false;
    no = false;

    Button button = (Button) findViewById(R.id.question1_button1);
    Button button2 = (Button) findViewById(R.id.question1_button2);
    Button button3 = (Button) findViewById(R.id.question1_button3);
    Button button4 = (Button) findViewById(R.id.question1_button4);
    Button button5 = (Button) findViewById(R.id.question1_button5);
    Button button6 = (Button) findViewById(R.id.question1_button6);


    button.setBackgroundColor(Color.GRAY);
    button2.setBackgroundColor(Color.DKGRAY);
    button3.setBackgroundColor(Color.GRAY);
    button4.setBackgroundColor(Color.GRAY);
    button5.setBackgroundColor(Color.GRAY);
    button6.setBackgroundColor(Color.GRAY);
}

public void button3Pressed(View v3) {
    eight = true;
    fiveK = false;
    tenK = false;
    fifteen = false;
    half = false;
    no = false;

    Button button = (Button) findViewById(R.id.question1_button1);
    Button button2 = (Button) findViewById(R.id.question1_button2);
    Button button3 = (Button) findViewById(R.id.question1_button3);
    Button button4 = (Button) findViewById(R.id.question1_button4);
    Button button5 = (Button) findViewById(R.id.question1_button5);
    Button button6 = (Button) findViewById(R.id.question1_button6);


    button.setBackgroundColor(Color.GRAY);
    button2.setBackgroundColor(Color.GRAY);
    button3.setBackgroundColor(Color.DKGRAY);
    button4.setBackgroundColor(Color.GRAY);
    button5.setBackgroundColor(Color.GRAY);
    button6.setBackgroundColor(Color.GRAY);
}

public void button4Pressed(View v4) {
    fifteen = true;
    fiveK = false;
    tenK = false;
    eight = false;
    half = false;
    no = false;

    Button button = (Button) findViewById(R.id.question1_button1);
    Button button2 = (Button) findViewById(R.id.question1_button2);
    Button button3 = (Button) findViewById(R.id.question1_button3);
    Button button4 = (Button) findViewById(R.id.question1_button4);
    Button button5 = (Button) findViewById(R.id.question1_button5);
    Button button6 = (Button) findViewById(R.id.question1_button6);


    button.setBackgroundColor(Color.GRAY);
    button2.setBackgroundColor(Color.GRAY);
    button3.setBackgroundColor(Color.GRAY);
    button4.setBackgroundColor(Color.DKGRAY);
    button5.setBackgroundColor(Color.GRAY);
    button6.setBackgroundColor(Color.GRAY);
```

```
    }

    public void button5Pressed(View v5) {
        half = true;
        fiveK = false;
        tenK = false;
        eight = false;
        fifteen = false;
        no = false;

        Button button = (Button) findViewById(R.id.question1_button1);
        Button button2 = (Button) findViewById(R.id.question1_button2);
        Button button3 = (Button) findViewById(R.id.question1_button3);
        Button button4 = (Button) findViewById(R.id.question1_button4);
        Button button5 = (Button) findViewById(R.id.question1_button5);
        Button button6 = (Button) findViewById(R.id.question1_button6);


        button.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.DKGRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button6Pressed(View v6) {
        no = true;
        fiveK = false;
        tenK = false;
        eight = false;
        fifteen = false;
        half = false;

        Button button = (Button) findViewById(R.id.question1_button1);
        Button button2 = (Button) findViewById(R.id.question1_button2);
        Button button3 = (Button) findViewById(R.id.question1_button3);
        Button button4 = (Button) findViewById(R.id.question1_button4);
        Button button5 = (Button) findViewById(R.id.question1_button5);
        Button button6 = (Button) findViewById(R.id.question1_button6);


        button.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.DKGRAY);
    }


}
```

activity_second_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".secondPage">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
```

```
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_second_page" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:srcCompat="@android:drawable/ic_dialog_email" />

    <LinearLayout
        android:id="@+id/second_page_overall_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="Is there a race you are preparing for?"
            android:textSize="30sp"
            android:textColor="#f00"
            android:layout_weight="1"
            android:gravity="bottom|center_horizontal"
            />
        <LinearLayout
            android:id="@+id/second_page_button_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <Button
                android:id="@+id/second_page_yes_button"
                android:layout_width="180dp"
                android:layout_height="80dp"
                android:layout_margin="20dp"
                android:text="YES"
                android:textSize="30sp"
                android:textColor="#f00"
                android:textAlignment="center"
                android:layout_weight="1"
                android:onClick="moreRaces"
                />
            <Button
                android:id="@+id/second_page_no_button"
                android:layout_width="180dp"
                android:layout_height="80dp"
                android:layout_margin="20dp"
                android:text="NO"
                android:textAlignment="center"
                android:textColor="#f00"
                android:textSize="30sp"
                android:layout_weight="1"
                android:onClick="noMoreRaces"
                />

        </LinearLayout>

        <LinearLayout
            android:id="@+id/second_page_submit_button_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:gravity="bottom|center_horizontal"
            android:layout_weight="1">
```

```
            <Button
                android:id="@+id/second_page_submit_button"
                android:layout_width="200dp"
                android:layout_height="100dp"
                android:text="SUBMIT"
                android:textColor="#f00"
                android:textSize="35sp"
                android:onClick="goToNextPage"
                />

        </LinearLayout>

    </LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

secondPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.view.View;
import android.widget.Button;

public class secondPage extends AppCompatActivity {

    public boolean race;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second_page);


        Button buttonYes = (Button) findViewById(R.id.second_page_yes_button);
        Button buttonNo = (Button) findViewById(R.id.second_page_no_button);

        buttonYes.setBackgroundColor(Color.GRAY);
        buttonNo.setBackgroundColor(Color.GRAY);
    }

    public void moreRaces(View v){
        Button buttonYes = (Button) findViewById(R.id.second_page_yes_button);
        Button buttonNo = (Button) findViewById(R.id.second_page_no_button);

        buttonYes.setBackgroundColor(Color.DKGRAY);
        buttonNo.setBackgroundColor(Color.GRAY);
        race = true;
    }

    public void noMoreRaces(View v){
        Button buttonYes = (Button) findViewById(R.id.second_page_yes_button);
        Button buttonNo = (Button) findViewById(R.id.second_page_no_button);

        buttonYes.setBackgroundColor(Color.GRAY);
        buttonNo.setBackgroundColor(Color.DKGRAY);
        race = false;
    }

    public void goToNextPage(View v){
        if (race){
            goToThirdPage();
```

```
        }
        else if (!race){
            goToFourthPage();
        }
    }

    private void goToThirdPage(){
        Intent intent = new Intent(this, ThirdPage.class);
        startActivity(intent);

    }

    private void goToFourthPage(){

    }

}
```

activity_third_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ThirdPage">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_third_page" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:srcCompat="@android:drawable/ic_dialog_email" />

    <LinearLayout
        android:id="@+id/page_3_overall_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:id="@+id/page_3_question_1_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">
            <TextView
                android:id="@+id/page_3_question_1"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="WHEN IS THE RACE?"
                android:textSize="30sp"
```

```
            android:textColor="#f00"
            android:gravity="center"
            />
        <LinearLayout
            android:id="@+id/page_3_question_1_type_in_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <EditText
                android:id="@+id/page_3_question_1_day_enter"
                android:layout_width="100dp"
                android:layout_height="80dp"
                android:text="DAY"
                android:inputType="number"
                android:textSize="15sp"
                android:layout_weight="1"
                android:textColor="#f00"
                />
            <EditText
                android:id="@+id/page_3_question_1_month_enter"
                android:layout_width="100dp"
                android:layout_height="80dp"
                android:text="MONTH"
                android:inputType="number"
                android:textSize="15sp"
                android:layout_weight="1"
                android:textColor="#f00"
                />
            <EditText
                android:id="@+id/page_3_question_1_year_enter"
                android:layout_width="100dp"
                android:layout_height="80dp"
                android:text="YEAR"
                android:inputType="number"
                android:textSize="15sp"
                android:layout_weight="1"
                android:textColor="#f00"
                />
        </LinearLayout>

    </LinearLayout>

    <LinearLayout
        android:id="@+id/page_3_question_2_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="WHAT DISTANCE?"
            android:gravity="center"
            android:textColor="#f00"
            android:textSize="30sp"
            android:layout_weight="1"
            />
        <LinearLayout
            android:id="@+id/page_3_question_2_type_in_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1"
            android:layout_gravity="center">
            <Button
                android:id="@+id/page_3_question_2_button1"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="5k"
```

```
                android:onClick="button1Pressed"/>

            <Button
                android:id="@+id/page_3_question_2_button2"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="10k"
                android:onClick="button2Pressed"/>

            <Button
                android:id="@+id/page_3_question_2_button3"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="800m"
                android:onClick="button3Pressed"/>

            <Button
                android:id="@+id/page_3_question_2_button4"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="1500m"
                android:textSize="11dp"
                android:onClick="button4Pressed"/>

            <Button
                android:id="@+id/page_3_question_2_button5"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="Half-Marathon"
                android:textSize="8dp"
                android:onClick="button5Pressed"/>

            <Button
                android:id="@+id/page_3_question_2_button6"
                android:layout_width="50dp"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="OTHER"
                android:textSize="11dp"
                android:onClick="button6Pressed"/>

        </LinearLayout>


    </LinearLayout>

    <LinearLayout
        android:id="@+id/page_3_question_3_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="IS THERE ANOTHER RACE YOU ARE PREPARING FOR?"
            android:textColor="#f00"
            android:textSize="30sp"
            android:gravity="center"
            />
        <LinearLayout
            android:id="@+id/page_3_question_3_button_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
```

```
                <Button
                    android:id="@+id/page_3_question_3_yes_button"
                    android:layout_width="180dp"
                    android:layout_height="80dp"
                    android:layout_margin="20dp"
                    android:text="YES"
                    android:textSize="30sp"
                    android:textColor="#f00"
                    android:textAlignment="center"
                    android:layout_weight="1"
                    android:onClick="moreRaces"
                    />
                <Button
                    android:id="@+id/page_3_question_3_no_button"
                    android:layout_width="180dp"
                    android:layout_height="80dp"
                    android:layout_margin="20dp"
                    android:text="NO"
                    android:textAlignment="center"
                    android:textColor="#f00"
                    android:textSize="30sp"
                    android:layout_weight="1"
                    android:onClick="noRaces"
                    />
            </LinearLayout>


        </LinearLayout>



    </LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

ThirdPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class ThirdPage extends AppCompatActivity {



    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_third_page);


    }

    public void button1Pressed(View v){
```

```
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.DKGRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button2Pressed(View v2){
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.DKGRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button3Pressed(View v3){
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.DKGRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button4Pressed(View v4){
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.DKGRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button5Pressed(View v5){
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
```

```
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.DKGRAY);
        button6.setBackgroundColor(Color.GRAY);
    }

    public void button6Pressed(View v6){
        Button button1 = (Button) findViewById(R.id.page_3_question_2_button1);
        Button button2 = (Button) findViewById(R.id.page_3_question_2_button2);
        Button button3 = (Button) findViewById(R.id.page_3_question_2_button3);
        Button button4 = (Button) findViewById(R.id.page_3_question_2_button4);
        Button button5 = (Button) findViewById(R.id.page_3_question_2_button5);
        Button button6 = (Button) findViewById(R.id.page_3_question_2_button6);


        button1.setBackgroundColor(Color.GRAY);
        button2.setBackgroundColor(Color.GRAY);
        button3.setBackgroundColor(Color.GRAY);
        button4.setBackgroundColor(Color.GRAY);
        button5.setBackgroundColor(Color.GRAY);
        button6.setBackgroundColor(Color.DKGRAY);
    }

    public void noRaces(View v){
        Intent intent1 = new Intent(this, fourth_page.class);
        startActivity(intent1);

        TextView day = findViewById(R.id.page_3_question_1_day_enter);
        TextView month = findViewById(R.id.page_3_question_1_month_enter);
        TextView year = findViewById(R.id.page_3_question_1_year_enter);

        Date today = Calendar.getInstance().getTime(); //gets the current date


        SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
        String formattedDate = curFormater.format(today);

        //Checking the date entered is within a "valid" range
        if (Integer.parseInt(day.getText().toString())>31 ||
Integer.parseInt(day.getText().toString())<28 ||
                Integer.parseInt(month.getText().toString())>12 ||
Integer.parseInt(month.getText().toString())<1){
            Toast.makeText(this, "The date entered is invalid",
Toast.LENGTH_LONG).show();
        }
        else{
            //checking the date entered hasn't already passed:
            if (Integer.parseInt(formattedDate.substring(5,
9))==Integer.parseInt(year.getText().toString())){
                if (Integer.parseInt(formattedDate.substring(3,
4))<Integer.parseInt(month.getText().toString())){
                    Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
                }
                else if (Integer.parseInt(formattedDate.substring(3,
4))>Integer.parseInt(month.getText().toString())){
                    Intent intent = new Intent(this, fourth_page.class);
                    startActivity(intent);
                }
                else{
                    if (Integer.parseInt(formattedDate.substring(0,
1))>Integer.parseInt(day.getText().toString())){
                        Intent intent = new Intent(this, fourth_page.class);
                        startActivity(intent);
```

```
                    }
                    else{
                        Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
                    }
                }
            }
            else if (Integer.parseInt(formattedDate.substring(5,
9))>Integer.parseInt(year.getText().toString())){
                Intent intent = new Intent(this, fourth_page.class);
                startActivity(intent);
            }
            else{
                Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
            }
        }



    }

    public void moreRaces(View v){

        TextView day = findViewById(R.id.page_3_question_1_day_enter);
        TextView month = findViewById(R.id.page_3_question_1_month_enter);
        TextView year = findViewById(R.id.page_3_question_1_year_enter);

        Date today = Calendar.getInstance().getTime(); //gets the current date


        SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
        String formattedDate = curFormater.format(today);

        //Checking the date entered is within a "valid" range
        if (Integer.parseInt(day.getText().toString())>31 ||
Integer.parseInt(day.getText().toString())<28 ||
                Integer.parseInt(month.getText().toString())>12 ||
Integer.parseInt(month.getText().toString())<1){
            Toast.makeText(this, "The date entered is invalid",
Toast.LENGTH_LONG).show();
        }
        else{
            //checking the date entered hasn't already passed:
            if (Integer.parseInt(formattedDate.substring(5,
9))==Integer.parseInt(year.getText().toString())){
                if (Integer.parseInt(formattedDate.substring(3,
4))<Integer.parseInt(month.getText().toString())){
                    Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
                }
                else if (Integer.parseInt(formattedDate.substring(3,
4))>Integer.parseInt(month.getText().toString())){
                    Intent intent = new Intent(this, fourth_page.class);
                    startActivity(intent);
                }
                else{
                    if (Integer.parseInt(formattedDate.substring(0,
1))>Integer.parseInt(day.getText().toString())){
                        Intent intent = new Intent(this, fourth_page.class);
                        startActivity(intent);
                    }
                    else{
                        Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
                    }
                }
            }
            else if (Integer.parseInt(formattedDate.substring(5,
9))>Integer.parseInt(year.getText().toString())){
                Intent intent = new Intent(this, fourth_page.class);
```

```
                startActivity(intent);
            }
            else{
                Toast.makeText(this,"The date entered has already passed",
Toast.LENGTH_LONG).show();
            }
        }
    }

}
```

activity_fourth_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".fourth_page">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="WHERE DO YOU TRAIN THE MOST?"
                android:textSize="30sp"
                android:textColor="#f00"
                android:gravity="center"
                />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_weight="1"
                android:text="Country"
                android:textColor="#f00"
                android:textSize="20sp"
                />
            <Spinner
                android:id="@+id/page_4_country_drop_down"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_weight="1"
                android:background="@android:drawable/btn_dropdown"
                android:spinnerMode="dropdown"
                />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="horizontal">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:text="City"
```

```
                android:textSize="20sp"
                android:textColor="#f00"
                android:layout_weight="1"
                />
            <Spinner
                android:id="@+id/page_4_city_drop_down"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_weight="1"
                android:background="@android:drawable/btn_dropdown"
                android:spinnerMode="dropdown"
                />
        </LinearLayout>

        <LinearLayout
            android:id="@+id/page_4_submit_button_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical"
            android:gravity="center_horizontal|bottom">
            <Button
                android:id="@+id/page_4_submit_button"
                android:layout_width="200dp"
                android:layout_height="110dp"
                android:textSize="30sp"
                android:textColor="#f00"
                android:text="SUBMIT"
                android:onClick="goToFifthPage"
                />
        </LinearLayout>

    </LinearLayout>



</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

fourth_page.java:

```
package com.example.runapp;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;


import com.instinctcoder.readwritefile.FileHelper;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Spinner;

public class fourth_page extends AppCompatActivity {

    private int STORAGE_PERMISSION_CODE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fourth_page);
```

```
        //https://stackoverflow.com/questions/13377361/how-to-create-a-drop-down-list

        Spinner country = findViewById(R.id.page_4_country_drop_down);
        String[] countries = new String[]{"England", "Scotland", "Wales", "N
Ireland"};
        ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, countries);
        country.setAdapter(adapter);


        Spinner city = findViewById(R.id.page_4_city_drop_down);
        if (country.getSelectedItem().toString().equals("England")){
            String[] cities = new String[]{"Bath", "Birmingham", "Bradford", "Brighton
& Hove", "Bristol", "Cambridge",
            "Canterbury", "Carlisle", " Chelmsford", "Chester", "Chichester",
"Coventry", "Derby", "Durham", "Ely", "Exeter",
            "Gloucester", "Hereford", "Kingston-upon-Hull", "Lancaster", "Leeds",
"Leicester", "Lichfield", "Lincoln", "Liverpool",
            "(City of) London", "Manchester", "Newcastle-upon-Tyne", "Norwich",
"Nottingham", "Oxford", "Peterborough", "Plymouth",
            "Portsmouth", "Preston", "Ripon", "Salford", "Salisbury", "Sheffield",
"Southampton", "St Albans", "Stoke-on-Trent", "Sunderland",
            "Truro", "Wakefield", "Wells", "(City of) Westminster", "Winchester",
"Wolverhampton", "Worcester", "York"};
            ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
            city.setAdapter(adapter2);
        }else if (country.getSelectedItem().toString().equals("Scotland")){
            String[] cities = new String[]{"Glasgow", "Edinburgh", "Aberdeen",
"Dundee", "Paisley", "East Kilbride", "Livingston",
            "Hamilton", "Dunfermline", "Cumbernauld", "Kirkcaldy", "Perth",
"Inverness", "Ayr", "Kilmarnock", "Coatbridge", "Greenock",
            "Glenrothes", "Stirling", "Airdrie", "Falkirk", "Irvine", "Dumfries",
"Motherwell", "Rutherglen", "Wishaw", "Cambuslang",
            "Bearsden", "Newton Mearns", "Clydebank", "Elgin", "Arbroath",
"Bishopbriggs", "Bathgate", "Renfrew", "Kirkintilloch", "Musselburgh",
            "Dumbarton", "Bellshill", "Peterhead", "Barrhead", "St Andrews",
"Bonnyrigg", "Blantyre", "Grangemouth", "Kilwinning",
            "Penicuik", "Johnstone", "Viewpark", "Erskine", "Broxburn"};
            ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
            city.setAdapter(adapter2);
        }else if (country.getSelectedItem().toString().equals("Wales")){
            String[] cities = new String[]{"Bangor", "Cardiff", "Newport", "St
Davids", "St Asaph", "Swansea"};
            ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
            city.setAdapter(adapter2);
        }else{
            String[] cities = new String[]{"Armagh", "Belfast", "Lisburn"};
            ArrayAdapter<String> adapter2 = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_dropdown_item, cities);
            city.setAdapter(adapter2);
        }


    }
    public void goToFifthPage(View v){


        if (ContextCompat.checkSelfPermission(fourth_page.this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED){
            Spinner city = findViewById(R.id.page_4_city_drop_down);
            FileHelper.writeInternalStorage("residence.txt",
city.getSelectedItem().toString());
            Intent intent = new Intent(this, fifthPage.class);
            startActivity(intent);
        }else {
            requestStoragePermission();
        }
```

```
    }

    private void requestStoragePermission() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
            //if user denied the permission before but tries to access it again
            new AlertDialog.Builder(this)
                    .setTitle("Permission needed")
                    .setMessage("This permission is needed to save data into a text
file")
                    .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            ActivityCompat.requestPermissions(fourth_page.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                        }
                    })
                    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    })
                    .create().show();
        } else {
            //requesting permission
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Toast that permission is granted
            } else {
                //Toast that permission wasn't granted
            }
        }
    }

}
```

activity_fifth_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">

    <include
        layout="@layout/app_bar_fifth_page"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
```

```
        app:headerLayout="@layout/nav_header_fifth_page" />
        <!-- app:menu="@menu/activity_fifth_page_drawer" -->

</androidx.drawerlayout.widget.DrawerLayout>
```

app_bar_fifth_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".fifthPage">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </com.google.android.material.appbar.AppBarLayout>

    <include layout="@layout/content_fifth_page" />

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/fab"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="@dimen/fab_margin"
        app:srcCompat="@android:drawable/ic_dialog_email" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:gravity="center"
                android:text="WELCOME TO YOUR TRANING SPACE"
                android:textSize="50sp"
                android:textColor="#f00"
                />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="1"
            android:gravity="bottom|center_horizontal"
            >
            <Button
                android:layout_width="250dp"
                android:layout_height="80dp"
                android:layout_gravity="center_horizontal|bottom"
                android:gravity="center"
                android:text="START WORKOUT"
```

```
                android:textColor="#f00"
                android:textSize="20sp"
                android:onClick="StartWorkout"
                />

        </LinearLayout>

    </LinearLayout>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

nav_header_fifth_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/side_nav_bar"
    android:gravity="bottom"
    android:orientation="vertical"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:theme="@style/ThemeOverlay.AppCompat.Dark">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/nav_header_desc"
        android:paddingTop="@dimen/nav_header_vertical_spacing"
        app:srcCompat="@mipmap/ic_launcher_round" />

        <Button
        android:id="@+id/nav_drawer_option_1"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="WORKOUTS"
        android:background="#f21f4d"
        android:textSize="20sp"
        android:layout_margin="10dp"
        android:onClick="workoutsPage"
        />

    <Button
        android:id="@+id/nav_drawer_option_2"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="ROUTES"
        android:background="#f21f4d"
        android:textSize="20sp"
        android:layout_margin="15dp"
        android:onClick="routesPage"
        />
    <Button
        android:id="@+id/nav_drawer_option_3"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="HISTORY"
        android:background="#f21f4d"
        android:textSize="20sp"
        android:layout_margin="15dp"
        android:onClick="historyPage"
        />
    <Button
        android:id="@+id/nav_drawer_option_4"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="ACHIEVEMENTS"
```

```
        android:background="#f21f4d"
        android:textSize="20sp"
        android:layout_margin="15dp"
        android:onClick="achievementsPage"
        />
    <Button
        android:id="@+id/nav_drawer_option_5"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:text="MANUAL WORKOUT ENTRY"
        android:background="#f21f4d"
        android:textSize="20sp"
        android:layout_margin="15dp"
        android:onClick="manualEntryPage"
        />
<!--        />-->

</LinearLayout>
```

fifthPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.os.Bundle;

import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;

import android.view.View;

import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.google.android.material.navigation.NavigationView;

import androidx.drawerlayout.widget.DrawerLayout;

import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;

import android.view.Menu;

public class fifthPage extends AppCompatActivity {

    private AppBarConfiguration mAppBarConfiguration;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_fifth_page);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
        DrawerLayout drawer = findViewById(R.id.drawer_layout);
        NavigationView navigationView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.


        NavController navController = Navigation.findNavController(this,
```

```
R.id.nav_host_fragment);
//        NavigationUI.setupActionBarWithNavController(this, navController,
mAppBarConfiguration);
        NavigationUI.setupWithNavController(navigationView, navController);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.fifth_page, menu);
        return true;
    }

    @Override
    public boolean onSupportNavigateUp() {
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        //return NavigationUI.navigateUp(navController, mAppBarConfiguration)
        return super.onSupportNavigateUp();
    }

    public void workoutsPage(View v){
        Intent intent = new Intent(this, workoutsPage.class);
        startActivity(intent);
    }

    public void routesPage(View v){

    }

    public void historyPage(View v){

    }

    public void achievementsPage(View v){
        Intent intent = new Intent(this, achievementsPage.class);
        startActivity(intent);
    }

    public void manualEntryPage(View v){

    }

    public void StartWorkout(View v){
        Intent intent = new Intent(this, startWorkout.class);
        startActivity(intent);
    }
}
```

activity_workouts_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu_workouts_page" />
```

```
    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <LinearLayout
        android:id="@+id/workouts_page_overall_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center">
        <TextView
            android:id="@+id/workouts_text"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_horizontal|top"
            android:text="HERE ARE SOME WORKOUTS WE RECOMMEND YOU:"
            android:textColor="#f00"
            android:textSize="30sp"/>
        <TextView
            android:id="@+id/workouts_page_workouts"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            />
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

bottom_nav_menu_workouts_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="HOME"
        android:onClick="goHome"/>

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"
        android:onClick="goToSettings"/>


</menu>
```

workoutsPage.java:

```
package com.example.runapp;

import android.Manifest;
import android.app.backup.FileBackupHelper;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.instinctcoder.readwritefile.FileHelper;
```

```
import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;


public class workoutsPage extends AppCompatActivity {

    private int STORAGE_PERMISSION_CODE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_workouts_page);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
                R.id.navigation_home, R.id.navigation_dashboard,
R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);

        TextView workout = findViewById(R.id.workouts_page_workouts);

        if (ContextCompat.checkSelfPermission(workoutsPage.this,
                Manifest.permission.READ_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
            if (FileHelper.readFirstLine("data.txt").equals("5K")){
                workout.setText(FileHelper.ReadFile("fiveK_training_plan.txt"));
            }
            else if (FileHelper.readFirstLine("data.txt").equals("10K")){
                workout.setText(FileHelper.ReadFile("fiveK_training_plan.txt"));
            }
            else if (FileHelper.readFirstLine("data.txt").equals("800")){
                workout.setText(FileHelper.ReadFile("800_training_plan.txt"));
            }
            else if (FileHelper.readFirstLine("data.txt").equals("1500")){
                workout.setText(FileHelper.ReadFile("800_training_plan.txt"));
            }
            else if (FileHelper.readFirstLine("data.txt").equals("HALF")){
                workout.setText(FileHelper.ReadFile("half_training_plan.txt"));
            }
            else if (FileHelper.readFirstLine("data.txt").equals("NONE")){
                workout.setText("THERE ARE NO WORKOUT SUGGESTIONS FOR YOU");
                workout.setTextSize(30);
            }
        }else {
            requestStoragePermission();
        }

    }

    private void requestStoragePermission() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.READ_EXTERNAL_STORAGE)) {
            //if user denied the permission before but tries to access it again
            new AlertDialog.Builder(this)
                    .setTitle("Permission needed")
                    .setMessage("This permission is needed to save data into a text
file")
```

```
                    .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            ActivityCompat.requestPermissions(workoutsPage.this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                        }
                    })
                    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    })
                    .create().show();
        } else {
            //requesting permission
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Toast that permission is granted
            } else {
                //Toast that permission wasn't granted
            }
        }
    }

    public void goHome(View v){
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }

}
```

bottom_nav_menu_routes_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="HOME"
        android:onClick="goHome"/>

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"
        android:onClick="goToSettings"/>



</menu>
```

activity_routes_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu_routes_page" />

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/routes_page_first_route"
            />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/routes_page_first_route_description"
            />
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/routes_page_second_route"
            />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:id="@+id/routes_page_second_route_description"
            />

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

routesPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.content.res.Resources;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.instinctcoder.readwritefile.FileHelper;
```

```
import java.io.InputStream;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

public class routesPage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_routes_page);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
                R.id.navigation_home, R.id.navigation_dashboard,
R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);


        ImageView route1 = findViewById(R.id.routes_page_first_route);
        ImageView route2 = findViewById(R.id.routes_page_second_route);

        TextView route1description =
findViewById(R.id.routes_page_first_route_description);
        TextView route2description =
findViewById(R.id.routes_page_second_route_description);


        if (FileHelper.ReadFile("residence.txt").equals("Edinburgh")){
            route1.setImageResource(R.drawable.route1);
            try{
                Resources res = getResources();
                InputStream in_s = res.openRawResource(R.raw.route1description);
                byte[] b = new byte[in_s.available()];
                in_s.read(b);
                route1description.setText(new String(b));
            }catch (Exception e){
                route1description.setText("Error: Can't show description");
            }

            route1.setImageResource(R.drawable.route2);
            try{
                Resources res = getResources();
                InputStream in_s = res.openRawResource(R.raw.route2description);
                byte[] b = new byte[in_s.available()];
                in_s.read(b);
                route2description.setText(new String(b));
            }catch (Exception e){
                route1description.setText("Error: Can't show description");
            }
        }
        else{

        }
    }

    public void goHome(View v){
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }

}
```

bottom_nav_menu_history_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">


    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="HOME"
        android:onClick="goHome"/>

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"
        android:onClick="goToSettings"/>



</menu>
```

activity_history_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu_history_page" />

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:id="@+id/history_text"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_horizontal|top"
            android:text="HERE ARE ALL THE WORKOUTS YOU HAVE EVER DONE:"
            android:textColor="#f00"
            android:textSize="30sp"/>
        <TextView
            android:id="@+id/history_page_history"
            android:layout_width="match_parent"
```

```
            android:layout_height="match_parent"
            />

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

historyPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.instinctcoder.readwritefile.FileHelper;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

public class historyPage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_history_page);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
                R.id.navigation_home, R.id.navigation_dashboard,
R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);

        TextView history = findViewById(R.id.history_page_history);

        history.setText(FileHelper.ReadFile("history.txt"));
    }

    public void goHome(View v){
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }

}
```

bottom_nav_menu_achievements_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="HOME"
        android:onClick="goHome"/>

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"
```

```
        android:onClick="goToSettings"/>



</menu>
```

activity_achievements_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu_acievements_page" />

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/welcome"
                android:text="HERE ARE YOUR ACHIEVEMENTS"
                android:textAlignment="center"
                android:textColor="#f00"
                android:textSize="25sp"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="FURTHEST DISTANCE:"
                android:textAlignment="center"
```

```
                android:textColor="#f00"
                android:textSize="20sp"
                android:textStyle="italic"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/furthest_distance"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="LONGEST TIME:"
                android:textSize="20sp"
                android:textColor="#f00"
                android:textAlignment="center"
                android:textStyle="italic"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/longest_time"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="FASTEST AVERAGE PACE:"
                android:textAlignment="center"
                android:textColor="#f00"
                android:textSize="20sp"
                android:textStyle="italic"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/fastest_average"
                />
        </LinearLayout>

    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

achievementsPage.java:

```
package com.example.runapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.instinctcoder.readwritefile.FileHelper;

import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
```

```
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

public class achievementsPage extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_achievements_page);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
                R.id.navigation_home, R.id.navigation_dashboard,
R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);

        TextView furthest_distance = findViewById(R.id.furthest_distance);
        TextView longest_time = findViewById(R.id.longest_time);
        TextView fastest_average = findViewById(R.id.fastest_average);

        furthest_distance.setText(FileHelper.ReadFile("distance_achievements.txt"));
        longest_time.setText(FileHelper.ReadFile("time_achievements.txt"));
        fastest_average.setText(FileHelper.ReadFile("pace_achievements.txt"));
    }

    public void goHome(View v){
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }

}
```

bottom_nav_menu_manual_entry_page.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/navigation_home"
        android:icon="@drawable/ic_home_black_24dp"
        android:title="@string/title_home"
        android:onClick="goHome"/>

    <item
        android:id="@+id/navigatio_settings"
        android:icon="@drawable/ic_menu_manage"
        android:title="SETTINGS"
        android:onClick="goToSettings"/>


</menu>
```

activity_manual_entry.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingTop="?attr/actionBarSize">

    <com.google.android.material.bottomnavigation.BottomNavigationView
```

```
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="?android:attr/windowBackground"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu_manual_entry_page" />

    <fragment
        android:id="@+id/nav_host_fragment"
        android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toTopOf="@id/nav_view"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/mobile_navigation" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="MANUAL ENTRY PAGE"
                android:textAlignment="center"
                android:textSize="25sp"
                android:textColor="#f00"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:textColor="#f00"
                android:text="Description: If you do not have measurements of either
time or distance, put a dash (-) in the box"
                android:textSize="15sp"
                android:textAlignment="center"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="DISTANCE:"
                android:textSize="20sp"
                android:textColor="#f00"
                android:textStyle="italic"
                android:layout_weight="1"
                android:textAlignment="center"
                />
            <EditText
                android:id="@+id/distance_entry"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
```

```
                android:inputType="numberDecimal"
                android:layout_weight="1"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="TIME:"
                android:textSize="20sp"
                android:textColor="#f00"
                android:textStyle="italic"
                android:layout_weight="1"
                android:textAlignment="center"
                />
            <EditText
                android:id="@+id/time_entry"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:inputType="numberDecimal"
                android:layout_weight="1"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="1">
            <Button
                android:layout_width="150dp"
                android:layout_height="100dp"
                android:text="SAVE"
                android:textAlignment="center"
                android:layout_gravity="center"
                android:textColor="#f00"
                android:textSize="20sp"
                android:onClick="save"
                />
        </LinearLayout>
    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

manualEntry.java:

```
package com.example.runapp;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import com.google.android.material.bottomnavigation.BottomNavigationView;
import com.instinctcoder.readwritefile.FileHelper;

import java.io.File;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

import androidx.annotation.NonNull;
```

```
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

public class manualEntry extends AppCompatActivity {

    private int STORAGE_PERMISSION_CODE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_manual_entry);
        BottomNavigationView navView = findViewById(R.id.nav_view);
        // Passing each menu ID as a set of Ids because each
        // menu should be considered as top level destinations.
        AppBarConfiguration appBarConfiguration = new AppBarConfiguration.Builder(
                R.id.navigation_home, R.id.navigation_dashboard,
R.id.navigation_notifications)
                .build();
        NavController navController = Navigation.findNavController(this,
R.id.nav_host_fragment);
        NavigationUI.setupActionBarWithNavController(this, navController,
appBarConfiguration);
        NavigationUI.setupWithNavController(navView, navController);


    }

    public void save(View v){ //when "SAVE" is clicked

        if (ContextCompat.checkSelfPermission(manualEntry.this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) { //checks whether has permission to write into
memory
            EditText distance_entry = findViewById(R.id.distance_entry);
            EditText time_entry = findViewById(R.id.time_entry);

            Date today = Calendar.getInstance().getTime(); //gets the current date
            SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
            String formattedDate = curFormater.format(today);


            if (!distance_entry.getText().toString().equals("-") ||
!distance_entry.getText().toString().equals(" ")){
                if (!time_entry.getText().toString().equals("-") ||
!time_entry.getText().toString().equals(" ")){
                    double distance =
Double.parseDouble(distance_entry.getText().toString());
                    double time = Double.parseDouble(time_entry.getText().toString());
                    double average_pace = time/distance; //PS I'm not dumb - this
gives minutes per k
                    //which is what runners tend to operate at

                    String details = formattedDate + ": " +
distance_entry.getText().toString() + "km; " + time_entry.getText().toString() + "min;
"
                            + Double.toString(average_pace) + "min/k;";
                    FileHelper.writeInternalStorage("history.txt", details);
                    Intent intent = new Intent(this, fifthPage.class);
                    startActivity(intent);


                    Double previous_distance_record =
Double.parseDouble(FileHelper.ReadFile("distance_achievements.txt").toString());
                    Double previous_time_record =
Double.parseDouble(FileHelper.ReadFile("time_achievements.txt"));
```

```
                    Double previous_pace_record =
Double.parseDouble(FileHelper.ReadFile("pace_achievements.txt"));

                    if (distance>previous_distance_record){
                        String newRecord = distance_entry.getText() + "km (" +
formattedDate + ")";

FileHelper.writeToAchievementsFile("distance_achievements.txt", newRecord);
                    }
                    if (time>previous_time_record){
                        String newRecord = time_entry.getText() + "min (" +
formattedDate + ")";
                        FileHelper.writeToAchievementsFile("time_achievements.txt",
newRecord);
                    }
                    if (average_pace<previous_pace_record){
                        String newRecord = Double.toString(average_pace) + "min/k (" +
formattedDate + ")";
                        FileHelper.writeToAchievementsFile("pace_achievements.txt",
newRecord);
                    }


                }
                else{
                    String details = formattedDate + ": " +
distance_entry.getText().toString();
                    FileHelper.writeInternalStorage("history.txt", details);
                    Intent intent = new Intent(this, fifthPage.class);
                    startActivity(intent);

                    double distance =
Double.parseDouble(distance_entry.getText().toString());
                    Double previous_distance_record =
Double.parseDouble(FileHelper.ReadFile("distance_achievements.txt").toString());
                    if (distance>previous_distance_record){
                        String newRecord = distance_entry.getText() + "km (" +
formattedDate + ")";

FileHelper.writeToAchievementsFile("distance_achievements.txt", newRecord);
                    }
                }
            }
            else{
                if (!time_entry.getText().toString().equals("-") ||
!time_entry.getText().toString().equals(" ")){
                    String details = formattedDate + ": " +
time_entry.getText().toString();
                    FileHelper.writeInternalStorage("history.txt", details);
                    Intent intent = new Intent(this, fifthPage.class);
                    startActivity(intent);


                    double time = Double.parseDouble(time_entry.getText().toString());
                    Double previous_time_record =
Double.parseDouble(FileHelper.ReadFile("time_achievements.txt"));
                    if (time>previous_time_record){
                        String newRecord = time_entry.getText() + "min (" +
formattedDate + ")";
                        FileHelper.writeToAchievementsFile("time_achievements.txt",
newRecord);
                    }

                }
                else{
                    Toast.makeText(this, "Nothing Entered",
Toast.LENGTH_SHORT).show();
                }
            }
        }else{
            requestStoragePermission();
```

```
        }



    }

    private void requestStoragePermission() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
            //if user denied the permission before but tries to access it again
            new AlertDialog.Builder(this)
                    .setTitle("Permission needed")
                    .setMessage("This permission is needed to save data into a text
file")
                    .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            ActivityCompat.requestPermissions(manualEntry.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                        }
                    })
                    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    })
                    .create().show();
        } else {
            //requesting permission
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Toast that permission is granted
            } else {
                //Toast that permission wasn't granted
            }
        }
    }

    public void goHome(View v){
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }

}
```

FileHelper.java:

```
package com.instinctcoder.readwritefile;

import android.content.Context;
import android.os.Environment;
import android.provider.ContactsContract;
import android.util.Log;

import com.example.runapp.R;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
```

```
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;

import static java.lang.System.err;

public class FileHelper {
    final static String TAG = FileHelper.class.getName();

    public static String ReadFile(String file) {
        String line = "";

        String directory = System.getProperty("user.home");
        String absolutePath = directory + File.separator + file;

        try {
            BufferedReader bufferedReader = new BufferedReader(new
FileReader(absolutePath));
            line = bufferedReader.readLine();
            while (line != null) {
                line = bufferedReader.readLine();

            }
        } catch (IOException e) {
            System.err.println();
        }
        return line;

    }

    public static boolean saveToFile(String file, String data) {
        boolean done = false;

        String directory = System.getProperty("user.home");
        String absolutePath = directory + File.separator + file;


        File f = new File(absolutePath);

        if (f.exists()) {

            try {
                FileWriter fileWriter = new FileWriter(absolutePath);
                fileWriter.write(data);
                fileWriter.close();
                done = true;
            } catch (IOException err) {
                System.err.println(err.getMessage());
            }
            return done;
        } else {
            try {
                boolean success = f.createNewFile(); // if file already exists will do
nothing
                FileWriter fileWriter = new FileWriter(absolutePath);
                fileWriter.write(data);
                fileWriter.close();
                done = true;
            } catch (IOException err) {
                System.err.println(err.getMessage());
            }
            return done;
        }



    }
```

```
public void createNewFile(String name) {

    try {
        File myObj = new File(name + ".txt");
        if (myObj.createNewFile()) {
            System.out.println("File created: " + myObj.getName());
        } else {
            System.out.println("File already exists.");
        }
    } catch (IOException e) {
        System.out.println("An error occurred.");
        e.printStackTrace();
    }

}


public static String readFirstLine(String file) {
    String text = null;
    try {
        File File = new File(file);
        BufferedReader br = new BufferedReader(new FileReader(File));
        text = br.readLine();
        br.close();
    } catch (FileNotFoundException ex) {
        Log.d(TAG, ex.getMessage());
    } catch (IOException ex) {
        Log.d(TAG, ex.getMessage());
    }
    return text;
}

public static boolean writeInternalStorage(String fileN, String textToWrite) {
    boolean done;
    File fileName = new File(fileN);
    File root = android.os.Environment.getExternalStorageDirectory();
    File dir = new File (root.getAbsolutePath() + "/download/");
    dir.mkdirs();
    File file = new File(dir.getAbsolutePath()+"/"+fileName);
    //File file2 = new File(R.raw.route1description);
    try {
        FileOutputStream f = new FileOutputStream(file);
        PrintWriter pw = new PrintWriter(f, true);
        pw.println(textToWrite);
        pw.flush();
        pw.close();
        f.close();
        done=true;
    } catch (FileNotFoundException e) {
        done=false;
        e.printStackTrace();
        Log.i(TAG, "******* File not found ");
    } catch (IOException e) {
        done = false;
        e.printStackTrace();
    }
    return done;
}

public static boolean writeToAchievementsFile(String fileN, String textToWrite){

    boolean done;
    File fileName = new File(fileN);
    File root = android.os.Environment.getExternalStorageDirectory();
    File dir = new File (root.getAbsolutePath() + "/download/");
    dir.mkdirs();
    File file = new File(dir.getAbsolutePath()+"/"+fileName);
    try {
        FileOutputStream f = new FileOutputStream(file);
        PrintWriter pw = new PrintWriter(f, false); //deletes the contents of the
```

```
file
            pw.println(textToWrite);                                    //before writing
to it
            pw.flush();
            pw.close();
            f.close();
            done=true;
        } catch (FileNotFoundException e) {
            done=false;
            e.printStackTrace();
            Log.i(TAG, "******* File not found ");
        } catch (IOException e) {
            done = false;
            e.printStackTrace();
        }
        return done;
    }
}
```

activity_start_workout.xml:

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">


    <fragment
        android:id="@+id/map"
        android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="527dp"
        tools:context=".startWorkout" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:layout_weight="1"
        android:layout_gravity="center_horizontal"
        android:gravity="center_horizontal">

        <Button
            android:layout_width="150dp"
            android:layout_height="80dp"
            android:layout_gravity="center_horizontal|top"
            android:gravity="bottom|center_horizontal"
            android:text="STOP WORKOUT"
            android:textColor="#f00"
            android:textSize="20sp"
            android:onClick="stopWorkout"/>
    </LinearLayout>
</LinearLayout>
```

startWorkout.java:

```
package com.example.runapp;

import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;

import android.Manifest;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
```

```
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Toast;

import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.Polyline;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.android.gms.tasks.Task;

import java.io.IOException;
import java.security.Provider;
import java.util.ArrayList;

public class startWorkout extends FragmentActivity implements OnMapReadyCallback{

    //https://stackoverflow.com/questions/42218419/how-do-i-implement-the-
locationlistener
    private GoogleMap mMap;
    private ArrayList<LatLng> coordinates = new ArrayList<>();
    double latitude;
    double longitude;
    Polyline line;
    public Handler handler;
    private static double distanceRan;
    private static Location currentLocation;
    private static LatLng currentLoc;
    public long startTime = System.currentTimeMillis();
    public long endTime;
    private static long timeRan;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_start_workout);
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);


        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                != PackageManager.PERMISSION_GRANTED) {

            // Permission is not granted
            // Should we show an explanation?
            if (ActivityCompat.shouldShowRequestPermissionRationale(this,
                    Manifest.permission.ACCESS_FINE_LOCATION)) {
                // Show an explanation to the user *asynchronously* -- don't block
                // this thread waiting for the user's response! After the user
                // sees the explanation, try again to request the permission.
            } else {
```

```
                // No explanation needed; request the permission
                ActivityCompat.requestPermissions(this,
                        new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
                        1);

                // MY_PERMISSIONS_REQUEST_READ_CONTACTS is an
                // app-defined int constant. The callback method gets the
                // result of the request.
            }
        } else {
            // Permission has already been granted
        }




        try {
            LocationManager mLocationManager;
            mLocationManager = (LocationManager) getSystemService(LOCATION_SERVICE);

            mLocationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1,
                    1, mLocationListener);
        }
        catch(SecurityException err)
        {
            System.err.println(err.getMessage());
        }



    }
    private final LocationListener mLocationListener = new LocationListener() {
        @Override
        public void onLocationChanged(final Location location) {
            latitude = location.getLatitude();
            longitude = location.getLongitude();
            Location newLocation = new Location("");
            newLocation.setLatitude(latitude);
            newLocation.setLongitude(longitude);


            distanceRan = distanceRan + currentLocation.distanceTo(newLocation);
            currentLocation.setLongitude(longitude);
            currentLocation.setLatitude(latitude);

            currentLoc = new LatLng(latitude, longitude);
            coordinates.add(currentLoc);
            mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLoc, 16.0f));
            //mMap.addMarker(new MarkerOptions().position(currentLocation));

            redrawLine();
        }

        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {

        }

        @Override
        public void onProviderEnabled(String provider) {

        }

        @Override
        public void onProviderDisabled(String provider) {

        }
    };
```

```
    public void redrawLine(){
        mMap.clear();

        PolylineOptions polylineOptions = new
PolylineOptions().width(5).color(Color.BLUE).geodesic(true);
        for (int i = 0; i<coordinates.size(); i++){
            LatLng point = coordinates.get(i);
            polylineOptions.add(point);
        }
        addMarker(); //adds current location
        line = mMap.addPolyline(polylineOptions);
    }

    public Runnable runLocation = new Runnable(){
      @Override
      public void run(){

          handler = new Handler();
          startWorkout.this.handler.postDelayed(startWorkout.this.runLocation, 2000);
      }
    };

    public void addMarker(){
        mMap.addMarker(new MarkerOptions().position(new LatLng(latitude, longitude)));
    }



    public void stopWorkout(View v){
        endTime = System.currentTimeMillis();
        timeRan=Math.round(endTime-startTime);

        double startingLat = coordinates.get(0).latitude;
        double startingLng = coordinates.get(0).longitude;
        Location startingPoint = new Location("");
        startingPoint.setLatitude(startingLat);
        startingPoint.setLongitude(startingLng);
        Location start = new Location("");
        start.setLatitude(0);
        start.setLongitude(0);
        double distanceToTakeAway = start.distanceTo(startingPoint);
        distanceRan = distanceRan - distanceToTakeAway;


        Intent intent = new Intent(this, runOverview.class);
        startActivity(intent);
    }

    public static long getTimeRan(){
        return timeRan;
    }

    public static double getDistanceRan(){
        return distanceRan;
    }


    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;

        currentLocation = new Location("");
        currentLocation.setLongitude(currentLocation.getLongitude());
        currentLocation.setLatitude(currentLocation.getLatitude());

        System.err.println(currentLocation.getLatitude() + "," +
currentLocation.getLongitude());
    }
```

```
}
```

activity_run_overview.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".runOverview">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:gravity="center">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">
            <TextView
                android:id="@+id/congrats"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="CONGRATULATIONS"
                android:layout_gravity="center"
                android:gravity="center"
                android:textSize="20sp"
                android:textColor="#f00"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="HERE ARE YOUR STATS FOR THE RUN:"
                android:textColor="#f00"
                android:textSize="20sp"
                android:gravity="center"
            />
        </LinearLayout>
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:id="@+id/distance_ran"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="KM"
                android:textSize="20sp"
                android:textColor="#f00"
                android:gravity="center"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
```

```
            android:orientation="horizontal"
            android:layout_weight="1">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/time_ran_hours"
                android:layout_weight="1"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="HOURS"
                android:layout_weight="1"
                android:textColor="#f00"
                android:textSize="10sp"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/time_ran_minutes"
                android:layout_weight="1"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="MINS"
                android:layout_weight="1"
                android:textColor="#f00"
                android:textSize="10sp"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/time_ran_seconds"
                android:layout_weight="1"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:text="SECONDS"
                android:layout_weight="1"
                android:textColor="#f00"
                android:textSize="10sp"
                android:gravity="center"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="horizontal">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:id="@+id/pace"
                android:gravity="center"
                />
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:text="MIN/KM"
                android:textSize="20sp"
                android:textColor="#f00"
```

```
                android:gravity="center"
                />
        </LinearLayout>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:orientation="vertical">
            <TextView
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:id="@+id/achievement"
                android:layout_weight="1"
                />
            <Button
                android:layout_width="150dp"
                android:layout_height="match_parent"
                android:text="OK"
                android:layout_weight="1"
                android:gravity="center"
                android:layout_gravity="center_horizontal|bottom"
                android:textSize="20sp"
                android:textColor="#f00"
                android:onClick="backHome"
                />
        </LinearLayout>


    </LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

runOverview.java:

```
package com.example.runapp;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.Manifest;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.LocationListener;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.instinctcoder.readwritefile.FileHelper;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;

public class runOverview extends AppCompatActivity {

    private int STORAGE_PERMISSION_CODE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_run_overview);

        TextView distanceRan = (TextView) findViewById(R.id.distance_ran);
        String distance = Double.toString(startWorkout.getDistanceRan() / 1000);
```

```
        distanceRan.setText(distance);

        double time = startWorkout.getTimeRan();
        double timeForPace = startWorkout.getTimeRan() / (60 * 1000);
        double distanceForPace = startWorkout.getDistanceRan();

        if (time == 0.0) {
            TextView timeRanHours = findViewById(R.id.time_ran_hours);
            timeRanHours.setText("0");
            TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
            timeRanMinutes.setText("0");
            TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
            timeRanSeconds.setText("0");
        } else {
            TextView timeRanHours = findViewById(R.id.time_ran_hours);
            if (Math.round(startWorkout.getTimeRan() / (60 * 60 * 1000)) >
startWorkout.getTimeRan() / (60 * 60 * 1000)) {
                String timeHours =
Double.toString(Math.round(startWorkout.getTimeRan() / (60 * 60 * 1000)) - 1);
                timeRanHours.setText(timeHours);
                time -= Double.valueOf(timeHours) * 60 * 60 * 1000;
            } else if (Math.round(startWorkout.getTimeRan() / (60 * 60 * 1000)) <
startWorkout.getTimeRan() / (60 * 60 * 1000)) {
                String timeHours =
Double.toString(Math.round(startWorkout.getTimeRan() / (60 * 60 * 1000)));
                timeRanHours.setText(timeHours);
                time -= Double.valueOf(timeHours) * 60 * 60 * 1000;
            }


            TextView timeRanMinutes = findViewById(R.id.time_ran_minutes);
            if (Math.round(time / (60 * 1000)) > (time / (60 * 1000))) {
                String timeMinutes = Double.toString(Math.round(time / (60 * 1000)) -
1);
                timeRanMinutes.setText(timeMinutes);
                time -= Double.valueOf(timeMinutes) * 60 * 1000;
            } else if (Math.round(time / (60 * 1000)) < (time / (60 * 1000))) {
                String timeMinutes = Double.toString(Math.round(time / (60 * 1000)));
                timeRanMinutes.setText(timeMinutes);
                time -= Double.valueOf(timeMinutes) * 60 * 1000;
            }


            TextView timeRanSeconds = findViewById(R.id.time_ran_seconds);
            String timeSeconds = Double.toString(Math.round(time / 1000));
            timeRanSeconds.setText(timeSeconds);
        }

        TextView pace = findViewById(R.id.pace);
        double avgPace = timeForPace / distanceForPace;
        pace.setText(Double.toString(Math.round(avgPace)));


        if (ContextCompat.checkSelfPermission(runOverview.this,
                Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) { //checks whether has permission to write into
memory

            Date today = Calendar.getInstance().getTime(); //gets the current date
            SimpleDateFormat curFormater = new SimpleDateFormat("dd/MM/yyyy");
            String formattedDate = curFormater.format(today);


            //Saving the run stats into the history page
            String details = formattedDate + ": " + distanceForPace + "km; " +
timeForPace + "min; "
                    + avgPace + "min/k;";
            FileHelper.writeInternalStorage("history.txt", details);
            Intent intent = new Intent(this, fifthPage.class);
            startActivity(intent);
```

```
            double previous_distance_record =
Double.parseDouble(FileHelper.ReadFile("distance_achievements.txt").toString());
            double previous_time_record =
Double.parseDouble(FileHelper.ReadFile("time_achievements.txt"));
            double previous_pace_record =
Double.parseDouble(FileHelper.ReadFile("pace_achievements.txt"));

            if (distanceForPace > previous_distance_record) {
                String newRecord = distanceForPace + "km (" + formattedDate + ")";
                FileHelper.writeToAchievementsFile("distance_achievements.txt",
newRecord);
            }
            if (timeForPace > previous_time_record) {
                String newRecord = timeForPace + "min (" + formattedDate + ")";
                FileHelper.writeToAchievementsFile("time_achievements.txt",
newRecord);
            }
            if (avgPace < previous_pace_record) {
                String newRecord = avgPace + "min/k (" + formattedDate + ")";
                FileHelper.writeToAchievementsFile("pace_achievements.txt",
newRecord);
            } else {
                if (distanceForPace > previous_distance_record) {
                    String newRecord = distanceForPace + "km (" + formattedDate + ")";
                    FileHelper.writeToAchievementsFile("distance_achievements.txt",
newRecord);
                }
            }


            if (timeForPace > previous_time_record) {
                String newRecord = timeForPace + "min (" + formattedDate + ")";
                FileHelper.writeToAchievementsFile("time_achievements.txt",
newRecord);
            } else {
                Toast.makeText(this, "Nothing Entered", Toast.LENGTH_SHORT).show();
            }

        } else {
            requestStoragePermission();
        }
    }

    private void requestStoragePermission() {
        if (ActivityCompat.shouldShowRequestPermissionRationale(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)) {
            //if user denied the permission before but tries to access it again
            new AlertDialog.Builder(this)
                    .setTitle("Permission needed")
                    .setMessage("This permission is needed to save data into a text
file")
                    .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            ActivityCompat.requestPermissions(runOverview.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
                        }
                    })
                    .setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            dialog.dismiss();
                        }
                    })
                    .create().show();
        } else {
            //requesting permission
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE}, STORAGE_PERMISSION_CODE);
```

```
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {
        if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                //Toast that permission is granted
            } else {
                //Toast that permission wasn't granted
            }
        }
    }

    public void backHome(View v) {
        Intent intent = new Intent(this, fifthPage.class);
        startActivity(intent);
    }
}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.runapp">

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".manualEntry"
            android:label="@string/title_activity_manual_entry"></activity>
        <activity
            android:name=".routesPage"
            android:label="@string/title_activity_routes_page" />
        <activity
            android:name=".historyPage"
            android:label="@string/title_activity_history_page" />
        <activity
            android:name=".achievementsPage"
            android:label="@string/title_activity_achievements_page" />
        <activity android:name=".runOverview" />
        <!--
             The API key for Google Maps-based APIs is defined as a string resource.
             (See the file "res/values/google_maps_api.xml").
             Note that the API key is linked to the encryption key used to sign the
APK.
             You need a different API key for each encryption key, including the
release key that is used to
             sign the APK for publishing.
             You can define the keys for the debug and release targets in src/debug/
and src/release/.
        -->
        <meta-data
            android:name="com.google.android.geo.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
```

```
            android:name=".startWorkout"
            android:label="@string/title_activity_start_workout" />
        <activity
            android:name=".workoutsPage"
            android:label="@string/title_activity_workouts_page" />
        <activity
            android:name=".fifthPage"
            android:label="@string/title_activity_fifth_page"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".fourth_page"
            android:label="@string/title_activity_fourth_page"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".ThirdPage"
            android:label="@string/title_activity_third_page"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".secondPage"
            android:label="@string/title_activity_second_page"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```