Flutter Fitness – Testing the Solution

Table of Contents

Comparison with original design specifications

Provides tracking for running activities

The app succeeds in providing tracking for running activities and meets the original specification. It has taken this specification and extended it to also include tracking for walking-based activities. The app uses a phone's GPS sensor to help track the location of the user and then calculate metrics like distance and altitude gained, helping the user to track their fitness activity. Furthermore, the app interfaces with the pedometer to help track the number of steps that are taken and then calculate statistics like total steps and cadence. Overall, the app is able to provide tracking for running (and general fitness) activities, meeting this design specification.

Notifications which provide motivation to run

My app partially fulfills this design specification. It provides daily notifications reminding the user to run. The notification text is selected from a number of possible notifications by using random number generation. The frequency of the notifications is limited by Android battery management, so they are inconsistent on some devices, in both their time and whether they appear when the app is compiled with release mode. Hence, it partially fulfills this specification.

Map of run

The app is able to produce a map of the run, meeting this design specification. While originally it was planned to use a Stack and CustomPaint widget, I discovered that the Flutter library for Google Maps already supports adding lines natively. Hence the technical aspects of this specification were slightly modified but the product still meets this design specification overall.

Provides cloud account integration

This design specification is met by the app. Users can easily sign up and login to the app and all their data is automatically synced. They can sync run data to and from the main server. Universal unique usernames also allow for individual user lookup. The project includes code for a central server.

Achievements

I was not able to implement an achievements system due to time and technical constraints, meaning that this design specification was not fulfilled in the final project. Implementing achievements would have increased the complexity of the UI significantly since many UI components would need to dynamically adapt based on whether an achievement was earned. Achievements would also have complicated the server code, since they must constantly be synced to the server and would also need to be synced to other devices. This would require me to implement a system which dynamically notifies a user's other devices when an achievement has been earned, while the app is almost certainly not running on their device. Unfortunately, my time and technical expertise constraints prevented me from being able meet this design specification since it would require significant code refactoring on both the client and server side, in addition to a significant amount of code running on Google Firebase (currently

the only main reliable way to notify Flutter apps of external messages, another piece of information I was not aware of when starting my project).

Ability to find nearby users

My app mostly meets this design specification. By going to the “nearby” tab a user can see nearby users. I have extended this feature from the one originally proposed by allowing all users in the world to be seen, not only those nearby. Users can see people nearby them and click on their icon to see more information about them, enhancing the socially driven nature of the app, as originally intended. This specification has been slightly modified in that while a user can see the locations of other users near them, they cannot tell whether they are in a run or not. Furthermore, the app handles the privacy issue of users not wanting to share their location with others by offering a switch to stop appearing in the nearby map within the settings menu. The original specifications stated that users would be able to designate whether their friends could see their data or not, but I felt that implementing such a feature would slow down server-side code since there would have to be constant checks to confirm whether users are friends or not. Therefore, the privacy features were retained as a switch to turn on and off location and the social features were retained by allowing everyone to see a users location should their settings allow it.

Leaderboards

Leaderboards are available in the app and are only slightly different from the original design specification. There are 3 different types: Distance, Steps and Time, within each type a user can see a world, friends and country leaderboard. Leaderboards are able to handle a variety of cases such as users on equal values and showing a user their rank even when they are not in the top 10. They refresh fast and help to add to the social aspect of the app as outlined in the design specifications. The design specifications state that the leaderboards should let users “track their progress over time”, while this has been partially fulfilled by the implementation of a weekly leaderboard, providing monthly and yearly leaderboards was omitted due to limited server compute capacity. In an attempt to maintain the user experience when using the leaderboard page, these were not provided. Besides that, the leaderboards function as described in the design specification and give the user a useful insight of where they stand within multiple categories of people. Being able to compare to friends, country and world was not explicitly stated in the original specifications and is an addition.

Personal Bests

These required a similar implementation to the achievements feature and hence were not able to be implemented, deviating from the design specifications. They would need a very dynamic UI and constant client-server syncing to ensure that some devices are not left with old Personal Bests. Another complication that arose when implementing this was that Android developers cannot assume app data will still be stored on a user’s phone when they next start the app, hence Personal Best data would have to be primarily stored on the server and storing it on a user’s devices would have been a risk.

Historical run data should be available to users

I believe that this design specification has been met. Users can see their historical runs from the history tab and can tap on any specific run to get more detailed information about it.

Users should be able to friend others

The main parts of this design specification have been met, but some of the subtleties have been changed. Users are able to friend others, but friending others does not make an impact on how much information other users are able to find out about a user. However, this is not a big issue since the amount of information visible to all users is not extensive. User friend lists are also independent of others, meaning that if user A friends user B, then user B is not automatically a friend of user A. Meaning that users are able to manage their friend lists independently of others and users may not feel sad if somebody else rejects their friend request. This is a slight change to the design specification in which the wording implied that friending a user would make them both friends of each other.

Users should be able to pause and continue runs

I was not able to implement this feature, due to the complexity starting an android geolocator stream, pausing one midway through a run and then starting it again, while still obtaining accurate results.

Provides accurate statistics

The app provides statistics about distance travelled, time taken and steps. Fulfilling part of the specification. The statistics are relatively accurate but I found that 5% was too tight a margin for low-end devices, so it was modified to 10%. I was able to achieve the modified specification of 10%, more information is provided within the section titled “Testing the accuracy of the data provided”.

Gives a level of privacy to users

This design specification has been primarily met. Users are able to stop sending their location to the server so they do not appear on the nearby map. They can also request their run data to be deleted from the server. Lastly, when another user signs in on the same device after logging another user out, the logged out user’s data is not visible. User data is not sold to third parties. The design specifications have been changed in that they originally granted more information to friends of users, however changes to the nature of data available to all users, rendered these safeguards redundant.

Most tasks should be able to be accomplished with one tap from the main menu

The app fulfills this design specification. Most screens can be accessed directly from the navbar, which is the primary form of navigation in the app. Starting a run can be done directly from the home screen.

Client – Server, Just-in-time architecture

The app primarily meets this design specification, leaderboard data, nearby users and history data is synced when the user requests. Data is sent using REST APIs and most information is stored centrally on the server. The client does cache more run history than was planned in the design specifications, however is able to fetch these from the server should they be deleted. This represents a minute change to the design specification and I feel that my app meets the specification overall.

Uses the Flutter framework

My app uses the Flutter framework, satisfying this design specification.

Speed

The app performs with very little to no log on the vast majority of devices. The server also responds relatively fast to queries. Hence, I feel it meets this design specification.

App should be inclusive and accessible to all

The app is actually more performant than originally expected and runs on a wide range of devices including low end ones (see system testing). It implements components from the latest Material Design 3 (including but not limited to: snackbars, buttons, navigation bar, cards, sheets, dividers, Material lists, tabs, switches, dropdown menus, dark mode, text fields and dialogues), meaning that users are more easily able to transfer their existing knowledge of how to use other apps into using this one. Since the app was rated as easy to use by all my testers and performed well on their devices (including the lower end ones) I feel my app has met this design specification. However, it should be noted that the app requires certain sensors such as a pedometer, so that it can collect data accurately. A user cannot use the app properly if these sensors are not present or not working, but only a very small number of phones in the past 8 years have been shipped without having these basic sensors. So, I maintain that my app meets this design specification.

Unit / Module testing

The unit testing was carried out by testing the results of the APIs using a program called Insomnia, in development the debug feature was used where individual requests could be crafted and their historical results could be viewed. In final testing the test menu was used and an automated test suite was created for more streamlined testing. A configuration file with both the debug queries and test suite has been included with the submission, it can be found at “server/insomnia.json”. Further testing of UI elements in the app was done using the app compiled in “debug” mode on my physical Samsung Galaxy A23 (Android 13, Wireless debugging mode, Wifi connected – 5GHz network, LTE connected – full bars (-85dBm), 62GB storage free, non-rooted), the app was originally not given any privileges allowing it to override normal battery optimization behaviour. However, issues with background notifications led me to disabling all battery optimisations for the app. It was compiled using the latest stable version of Flutter available at the time (Flutter 3.11.0-0.1.pre linux).

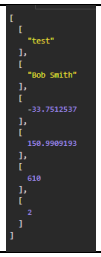
The server code was run on an Oracle Cloud VM.Standard.E2.1.Micro VM running Ubuntu 22.04.2 LTS with 1 GB ram and 1 CPU core of a AMD Epyc 7551 clocked at 2.66 GHz. The relevant software versions included Python 3.10.6 with package versions: Flask==2.3.1, Flask-Cors==3.0.10, geopy==2.3.0.

RED indicates tests with unexpected results

API tests

Test Description	Data Sent	Data Received	Comments
Signup a user	["unittest", "111", "Ben", "Chen"]	true	While the test returns true, its success can only really be verified

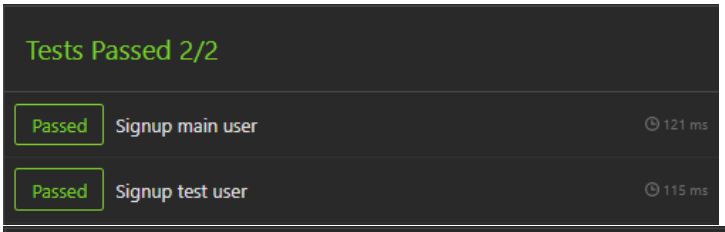
			by performing the following two tests
Login a user	["unittest", "111"]	[true, "Ben", "Chen", true, "Australia"]	As expected, this confirms the signup worked
Try signing up again but with the same username	["unittest", "111", "Ben", "Chen"]	false	As expected, the server does not allow another signup with the same username
Get user data	["unittest", "og"]	["Syed Ahmad", false, 0.00, 0, 0, "og", "Australia"]	As expected, note the og user was created prior to this query
Send rundata		true	As expected, again further tests are used to confirm successful receipt and processing
Get rundata	["unittest", "1684488081470.json"]	 (the output continued but was cropped for readability purposes)	As expected, confirms successful storage of runs
Friend user	["unittest", "og", false]	True	As expected
Get leaderboard	["unittest"]	{ "countryDistance": [[1, "unittest", 0.03], [2,	As expected, note og in the friends section as

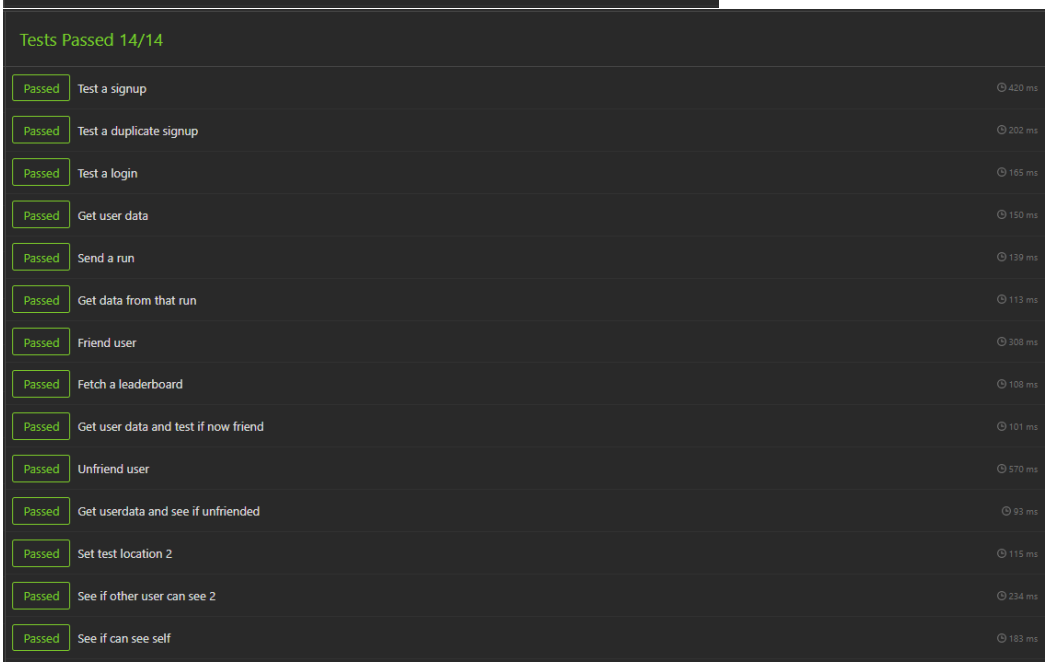
		"og", 0], [2, "test", 0]], "countrySteps": [[1, "unittest", 48], [2, "og", 0], [2, "test", 0]], "countryTime": [[1, "unittest", 1], [2, "og", 0], [2, "test", 0]], "friendsDistance": [[1, "unittest", 0.03], [2, "og", 0]], "friendsSteps": [[1, "unittest", 48], [2, "og", 0]], "friendsTime": [[1, "unittest", 1], [2, "og", 0]], "worldDistance": [[1, "unittest", 0.03], [2, "og", 0], [2, "test", 0]], "worldSteps": [[1, "unittest", 48], [2, "og", 0], [2, "test", 0]], "worldTime": [[1, "unittest", 1], [2, "og", 0], [2, "test", 0]] };	well since they were friended, this confirms that the leaderboard creator respects friend settings
Get user data – make sure the user has been friended	["unittest", "og"]	["Syed Ahmad", true, 0, 0, 0, "og", "Australia"]	As expected, the user is friended
Unfriend user	["unittest", "og", false]	true	As expected
Get user data – make sure the user has been unfriended again	["unittest", "og"]	["Syed Ahmad", true, 0, 0, 0, "og", "Australia"]	As expected, the user is once again unfriended
Send location for nearby users tracking	["test", -33.7512537, 150.9909193]	true	As expected, again a user with username test was created prior to running the test
Get other user locations	["unittest", -33.7457554, 150.9910882]		As expected, the last value is variable as it represents seconds since last updated, the distance of 610 is correct
Get other user locations but the only other user is the requesting user	["test", -33.7457554, 150.9910882]	[[], [], [], [], [],	As expected, the server does not return the requesting user's location

		[]]	
Delete run data for a user	[“og”]	True	As expected
Delete run data for a user who has no run data	[“og”]	True	As expected, what is more important here is that the server did not crash due to trying to delete a directory that does not exist
Server retains data on restart	Shutdown server Retry duplicate signup Retry login Get user history	Server has error Login is successful and gives correct information History is retained	Not expected, only the login, first name and last name lists were being saved, the country and friends were not
Server retains data on restart	Shutdown server Retry duplicate signup Retry login Get user history	Duplicate signup is rejected Login is successful and gives correct information History is retained	As expected, since the login gives correct information that means that user data has been retained

ADD TESTING INSMONIA MS RESULTS HERE

Insomnia automated test results





App Unit/Module testing

Test Description	Input	Output	Comments
Decimal input on step/time input box	0.2 entered into text field via keyboard	The program accepted 0.2	This is not expected functionality, allowing the user to set a goal of fractional steps, to rectify this issue, I implemented more aggressive Regex input formatters which made sure to not allow decimal values
Decimal input on step/time input box (Take 2)	0.2 entered into text field via keyboard	Only accepted the 2 input	As expected, the 0. Input was invalidated and not accepted since you cannot have fractional steps / minutes (fractional minutes have been

			purposely not supported)
Distance box only accepts valid inputs	0.24241242154325, 0.55.33 .0 .0.1 8/1 0.45 All the values above were inputted	All 6 inputs were accepted	This is not expected functionality, to fix this much stronger regex filters were required. There are a number of things the input should filter, not allow values like .0 which mean nothing, also should not allow characters other than numbers and decimal points, only allow one decimal point and to restrict the number of precision digits after the decimal point to 2 – a reasonable amount. I was able to create a regex filter to satisfy this purpose.
Distance box only accepts valid inputs	0.24241242154325, 0.55.33 .0 .0.1 8/1 0.45 All the values above were inputted	The invalid parts of the first 5 were not accepted and only the last one was accepted completely	As expected, filters out inputs that are not accepted values
First / Last name changing box does not accept empty input	No input just click send	App displays warning message and does not execute call	As expected
App chooses unit correctly during run	Goal of singular unit eg. 1 minute, 1 step	1 minute, 1 step	As expected, other values display “minutes” and “steps” as required
Signup page does not allow empty inputs	Submit with no inputs Submit with only username Submit with username password and confirm password	Username already taken error Accepts Accepts	Results given in order of testing, empty inputs were not denied
Signup page does not allow empty inputs	Submit with no inputs Submit with only username	Gives correct error for all	As expected

	Submit with username password and confirm password		
Signup page does not allow spacebars	Abc Just a spacebar	Spacebar was not accepted	As expected
Step tracking	Start a run and take a step	Step counter increases	As expected
Location tracking	Start a run	Location is printed correctly	As expected
Leaderboard refresh	Pull down on leaderbords screen	Leaderboard is refreshed	As expected
Run file is saved	Full run data	Run file in JSON	As expected, checked using Android Studio Device file manager
Leaderboards screen shows	Tap on friends button	Screen shows	As expected
Nearby screen shows	Tap on nearby button	Screen shows	As expected
History screen shows	Tap on history button	Screen shows	As expected
Settings screen shows	Tap on settings button	Screen shows	As expected

Program Testing

Note: unless specified most tests were conducted using the account with username og and fill server with data means running the “testleaderboard.py” python file

Test Description	**Input / Methodology**	**Output / Effect**	**Comments**
Check countdown before run	Hit start run	Full countdown animation with audio	As expected
Program notifies on goal achieved	Multiple tests: Goal of 1 minute Goal of 0.2km Goal of 2 minutes Goal of 3 steps	The app notified for all of them, automatically raising the audio,	As expected. Since whether the goal is achieved is only checked periodically, the response time was slightly slow, the correct text for singular units was used on the during run screen and with the speech to text
App does not allow navigation to pages when not logged in	Open instance of app where not logged in Try navigating to other pages	App gives error	Not expected
App does not allow navigation to pages when not logged in	Open instance of app where not logged in Try navigating to other pages	App open login screen	As expected

Leaderboard page refresh	Load leaderboard Change data for another user using another phone (eg. Make Britt go on another run) Refresh leaderboard page	Updated leaderboard	As expected
Leaderboard but two users have same data	Leaderboard but two users values double up	Leaderboard but the users with same value have their value set as rank despite being in the correct position	I was using the wrong index when adding a rank on to the user, this is why the order was correct but the rank number was not
Leaderboard but two users have same data	Leaderboard but two users values double up	Correct leaderboard	As expected
Leaderboard after data for a user is deleted	Delete run data for user og Check leaderboard	Empty leaderboard	Error, when deleting the run data the folder for a user was also deleted since the code iterated through the folder names it could not find any folder with the requested username and gave an error. To fix this I refactored my code to iterate through all users and then handle the case where no folder exists separately
Leaderboard after data for a user is deleted	Delete run data for user og Check leaderboard	Correct leaderboard	As expected
Leaderboards reset each week	Fill server with data Wait for Sunday 0000 GMT Check leaderboard	Leaderboard where all users do not have any data	As expected
Leaderboard handles case when requesting user is not in the top 10	Fill server with data such that the og user is outside the top 10 Refresh leaderboard page	Correct leaderboard with rank 10 then og rank	As expected

Friend Leaderboard	Fill server with data Make og friend Britt and Ben Refresh leaderboard page	Friend leaderboard contains Britt, Ben and og in correct order and no other leaderboards are changed	As expected, also proves the friending system is working
Country Leaderboard	Fill server with data Change country of Britt to Canada Change country of og to Canada Refresh leaderboard page	Country leaderboard contains Britt and og in correct order and no other leaderboards are changed	As expected
Leaderboard handles case where two users have the same value and hence the same rank	Fill server with data such that two users should be given the same rank	Leaderboard in correct order but the rank number is the same as the value	Not expected, incorrect list indices were used after ranking
Leaderboard handles case where two users have the same value and hence the same rank	Fill server with data such that two users should be given the same rank	Correct leaderboard with the two users on the same rank and only max of 11 users shown	As expected
Leaderboard ranks based on seconds not minutes but only displays minutes	Fill server with data Check leaderboard	Users which should be separated based on seconds all have equal rank	Not expected, the code was not taking into account ranking by seconds
Leaderboard ranks based on seconds not minutes but only displays minutes	Fill server with data Check leaderboard	Users are ranked correctly separated by seconds where possible	As expected
Tabs on leaderboards page	Tap on different tabs on leaderboards page	The tab is changed and the correct leaderboard is shown	As expected, but a new bug where if you tapped on a text field before changing tabs, the text field would stay selected on the previous tab
Get user data	Go to leaderboards page Enter Britt into search for a user textfield	First a quick loading animation then Britts information with correct friend status	As expected
Get user data for a non-existent user	Go to leaderboards page Enter the name of an unknown user into the search for a textfield (I used aslkf)	Infinite loading animation	Not expected, the server was returning false but the code was not handling false

Get user data for a non-existent user	Go to leaderboards page Enter the name of an unknown user into the search for a textfield (I used aslkf)	Loading animation stops and a snackbar informing the user is shown	As expected
Nearby map appears	Go to nearby page	Nearby map appears with correct user location	As expected
Nearby map appears when immediately go to nearby page	Turn on app and immediately go to nearby page	Error appears	Not expected, the program had not fetched a location already and hence the map could not be
Nearby map appears when immediately go to nearby page	Turn on app and immediately go to nearby page	Works, map appears with correct location	As expected, sometimes take a little longer to load while location is fetched
Nearby Map users appear and Nearby user card	Logon with one user on one phone (eg. Og on phone 1) Logon with another usr on another phone (eg. Test on phone 2) Look at nearby map Tap on nearby user	The other user displays on the map, clicking on their icon displays their information correctly and the time since last updated refreshes periodically and provides correct data, distance measurements are correct	As expected
Nearby card disappears when the user location is older than the threshold and they disappear from map	Same setup as previous, open up card of nearby user Wait until they disappear from map	Card disappears as well	As expected
History page	Tap on history page	Historical runs come up with correct icons and info	As expected
History page when no history	Tap on history page (on an account that has no previous runs)	Message acknowledging that the user has nor previous runs	As expected, the main thing we were looking for is that there are no errors
Historical run page	Tap on historical run	Historical run page appears, data is correct, map appears and graph appears with correct data	As expected
Name changes	Tap on settings	Name changes	As expected

	Press first name or last name Enter a new name Press submit Check username update with api or different phone with another profile signed in		
Name box handles empty input	Tap on settings Press first name or last name Enter a no input Press submit Check username update with api or different phone with another profile signed in	App accepts empty name	Not expected, input validation needs to be updated
Name box handles empty input	Tap on settings Press first name or last name Enter a no input Press submit Check username update with api or different phone with another profile signed in	App does not accept empty name, name is unchanged	As expected
Location updates setting works when turning off	Tap on settings Change location updates to off	Location updates stop sending but old location still appears on nearby page	Not expected, I did not delete the storage of the last known location from the server
Location updates setting works when turning off	Tap on settings Change location updates to off	Location updates stop sending and previous location disappears	As expected
Location updates setting works when turning on, from off position	Tap on settings Change location updates to on	Location updates start being updated immediately	As expected
Country change works	Tap on settings Tap on the change country Check the country of the user either via API or by getting user data from another device	Country is changed	As expected

Dark mode works	Press dark mode button	Dark mode activates on all pages and widgets	As expected
Dark mode persists across app restarts	Turn on dark mode, turn off app, turn on app	Dark mode status persists across restarts	As expected
Logout user button works	Sign in to an account, go to settings, press Logout	User is redirected to home screen and is logged out of app	As expected
Notification appears	Start app Wait for notification to appear	Notification appears (but late) and subsequent notifications are different due to RNG	I am unable to control the precise time of notifications due to the nature of Android background tasks
All notifications have an icon (including the notification for background location tracking)	Start app Start run Finish run Close app and wait for notification	All notifications use correct icon which complies with Android 13 guidelines	As expected
Run gets cancelled when the user has 0 distance	Start run Cancel during countdown Alternative test: Start run Wait for countdown Cancel shorty after starting but before any distance would have been recorded	Run is cancelled and not saved	As expected
Notifications provide different messages	Wait for notification Wait for another notification	Different notification messages	As expected, this test confirms the random number generation to select notification messages is working. It is possible based on change to get the same notification twice or thrice in a row, but unlikely. This was primarily tested with the notification interval set to 15min rather than 24 hours
Notification icon appears in notification	Wait for notification	All notifications use the separate notification icon	As expected

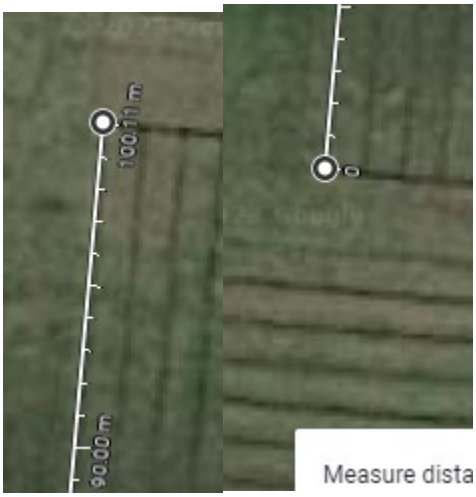
Logging in with different account doesn't expose previous user's data	Sign in with an account that has run data (eg. og) Sync with server by going to history page Log out of account Sign in with another account Navigate to history page	The data is correct for the second user and the data of the first user is not displayed	As expected
App can fetch required run data	Clean install app Login to an account with more than 1 previous run Go to history page	Al runs are available and have synced	As expected
App can use cache data when cant access run history	Login with an account that has run history Sync with server by going to history tab Go on aeroplane mode Go back to history tab	Data is cached and available on the history tab	As expected
App Icon appears (on home screen)	Compile app with icon submitted	Icon comes up but is very very small	Not expected, the icon had too much padding around it
App Icon appears (on home screen)	Compile app with icon submitted	Icon comes in expected form	As expected

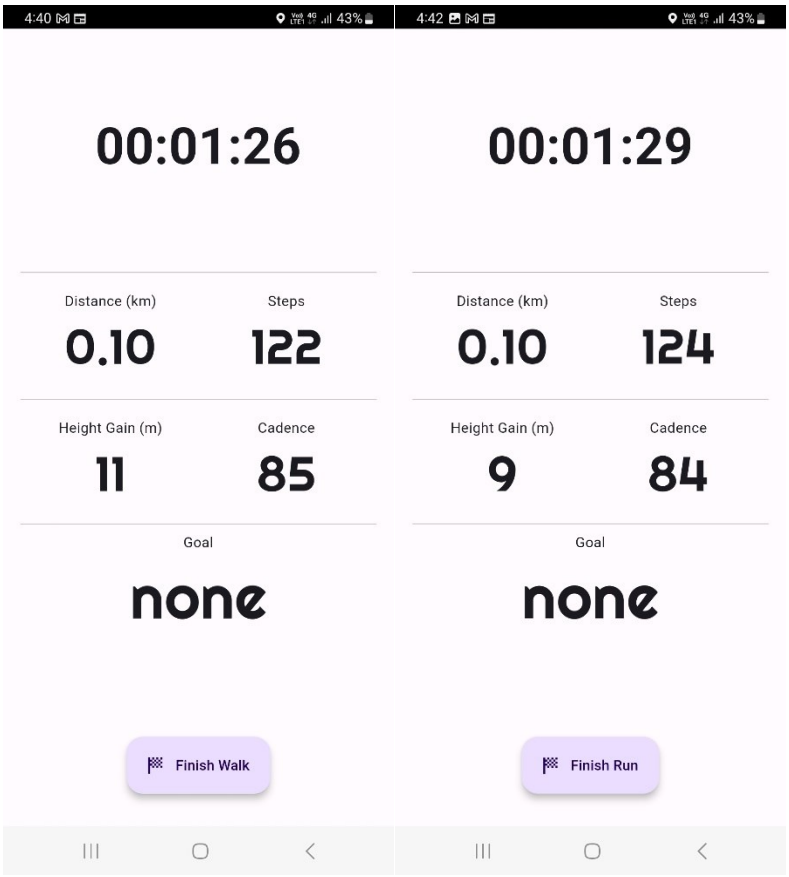
Additional Testing Documentation

Testing the accuracy of the data provided

To test the accuracy of the data provided, I went to a local running track with a 100m sprint area marked on the field. To verify that the length of the strip was indeed 100m, I used the Google Maps measure distance tool, to measure the distance from the start to the end of the track. I measured the distance along one of the lines dividing the lanes so that my measurement of the track was not influenced due to measuring a diagonal, rather than the straight distance. Every care was taken to align the start and endpoints of the measurement to the start and the end of the track. The results of the measurement (a length of 100.11m) and zoomed in screenshots of the start and endpoints of the measurement have been included below. It can be concluded that the track is either exactly 100m or very close to 100m in length.


Limousines
feur Driven |...
78Z
100.11 m
80.00 m
60.00 m
40.00 m
20.00 m
0
Measure distance
Click on the map to add to your path
Total distance: 100.11 m (328.46 ft)


100.11 m
90.00 m
0
Measure dista



I have included two screenshots from three trials (I forgot to take one of the screenshots and then its history data got overwritten due to a bug in my code at the time) of the 100m track. In the first 2 instances the counter hit 0.10 around 4 metres from the finish line, while in the third it hit 0.1 around 2 metres from the finish line. From this we can conclude that the app would measure around 103m on average for the 100m track. This is a good result on a mid-range device.

However the step tracking was slightly less accurate. It measured by me counting my steps while conducting the three rounds of the distance experiment and comapring this value to the one given by the app. The reported step counts were: 122, 134, 124 and my actual step counts were: 131, 139 and 132. These do not fall in a 5% range but fall in a 10% one. Since we are measuresing in percentages, we can assume that the results will stay roughly within a 10% range of the actual value even for larger values and are limited by the capabilities of the sensor onboard the device and not necessarily the app code.

Testing the Leaderboard and Friends features

To test these features, test data was required. To add test data to the server, I created a python file which first creates an account for a given user and then pushes some run data to their account. The run data was very limited and only a snapshot of a full run data file, only including the necessary elements required for testing. Hence, if a user tries to login with the credentials used by the generated users, they may encounter unexpected behaviour. The names were generated using https://randomwordgenerator.com/name.php. The code also requires the user to manually update the start variable which represents the epoch time when the run was supposedly done, usually this requires

setting the epoch time to any time within the past week (as measured in GMT time with a week starting on Sunday).

The same code can be rerun multiple times because the server will simply return a false if a user has already signed up and the python code will continue to push the run data without throwing an error. If a run has already been pushed the server overwrites it, so that means that the python code running multiple times will not mutate the data unless intended. The code is available at “server/testleaderboard.py”

```python
import requests

# The start value should be adjsuted to be part of the current week
start = 1685403852075
server = "http://sdd.syedahmad.tech:8080"

fnames = ["Nettie", "Britt", "Clare", "Lynn", "Jerry", "Carey", "Julie", "Eloy", "Savannah", "Josef", "Ben"]
lnames = ["Savage", "Burgess", "Graves", "Singh", "Lin", "Edwards", "Winters", "Saunders", "Erickson", "Arroyo", "Dickens"]

unames = ["Nettie", "Britt", "Clare", "Lynn", "Jerry", "Carey", "Julie", "Eloy", "Savannah", "Josef", "Ben"]
times = ["00:01:07", "00:01:08", "00:01:09", "00:01:10", "00:01:11", "00:01:12", "00:01:13", "00:01:14", "00:01:15", "00:01:06", "00:02:06"]
distances = ["0.04", "0.05", "0.06", "0.07", "0.08", "0.09", "0.10", "0.11", "3.03", "0.12", "0.12"]
steps = ['30000', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11']

# NOTE while these accounts do have valid logins, logging into them may cause unintended behaviour due to the stripped down nature of the request being sent here

for i in range(len(fnames)):
    data = {"username": unames[i], "start": start, "timeString": times[i], "totalDistance": distances[i], "totalSteps": steps[i], "exercise": "Run"}
    signup = f"{server}/signup"
    requests.post(signup, json=[unames[i], "000", fnames[i], lnames[i]])
    rundata = f"{server}/rundata"
    requests.post(rundata, json=data)
```

The data filling code

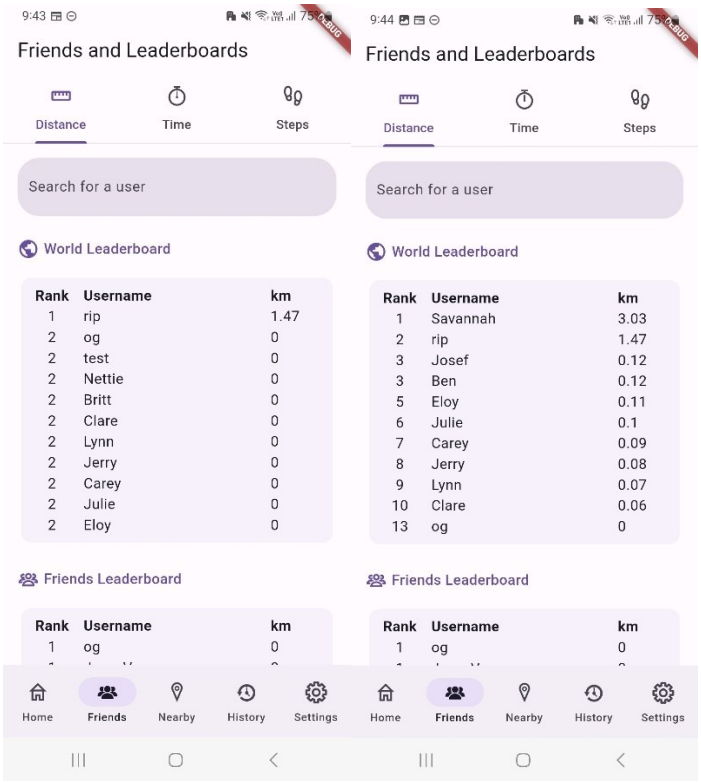


Image of before running the data filling code vs after running the data filling code (Note the filling code had been run in a previous week, so all the accounts already existed in this screenshot)

The code allowed me to test a variety of features mainly relating to the leaderboard system. These included: users with the same rank, displaying the current user at the bottom of the leaderboard (with their current rank) if they are not part of it and the leaderboard dynamically resizing to a smaller size if less than 10 users exist. During these tests I found some issues such as one where when users had an equal rank, their order would be correct but their rank number would appear as their distance/time/steps value. This was due to me using a wrong index in one place. There was also another issue where users with the same number of minutes but different seconds would be given the same rank since the code was only checking minutes, since that was the only value displayed by the app UI. By creating fake users, I was also able to verify that the friends system worked as expected and that country/friends leaderboards were updated correctly.

System Testing

I tested the system on a variety of devices operated by a number of different people. These people provided live test data for my project from their varying systems and reported back their feedback on the speed of the app, interface design and any errors experienced. To provide an additional test of how intuitive the app UI was, the testers were not given any specific instructions on how to operate the app.

Age of Tester	**Device specifications**	**Comments / Evaluation**
15 - 20	Device: Samsung J7 OS version: Android 7.1.1 CPU: Exynos 7580 Octa (Octa-core 1.5 GHz Cortex-A53) RAM: 1.5 GB Other relevant information: Screen size: 5.5", 720 x 1280	Tester (experienced runner): App ran better on my phone than Strava, I also liked the UI more than Strava. I found it easy to sign up and the app fits on my screen nicely. One of the first things I did after installing the app was to turn on dark mode and I found it to look very clean. Overall, I really like the app any may stop using Strava. There was a warning before installing the app which made me fear that it may be malware. The tester also mentioned that they enjoy the sight of themselves being above me on the leaderboards. Me: This result is very impressive, I previously doubted that the app would install on such an old phone but that was not the case. On top of that the user found the app to perform very well, stating that Strava was laggy on their phone and this app was not. Proving the viability of my app running on lower-end devices (The CPU on this device is very similar to the one

		used on the Raspberry Pi 3B+ but has a few more cores). I also was not sure how the app would perform on different screen sizes, despite creating code that should have adapted to smaller phones. It is also good to see that the user found the UI to be fast and easy to use, so much so that they found this app better than existing solutions on the market. The user also provided a test for leaderboard functions and dark mode on an older device.
15 - 20	Device: Samsung S20 FE OS version: Android 13 CPU: Exynos 990 - Version 1 (2x2.73 GHz Mongoose M5 & 2x2.50 GHz Cortex-A76 & 4x2.0 GHz Cortex-A55) RAM: 6 GB Other relevant information: Screen size: 6.5", 1080 x 2400	Tester: I found the app very smooth and easy to use. I like the leaderboard feature and the nearby feature which helps you find nearby users of the app. I used the app in split screen with it occupying the top half of the phone screen. When installing the app, there was some kind of security warning, but I clicked install anyway. Me: The user presented and interesting and unsupported use case for the app, using it in split screen. Despite being an unsupported scenario, the app reportedly performed admirably and displayed no issues. The user also found the app to run with smooth animations and performance overall. This kind of performance on a mid-flagship phone from a couple of years ago, reinforces the ability of my app to perform with little noticeable lag on the majority of devices.
35 - 40	Device: Samsung A53 5G OS version: Android 13 CPU: Exynos 1280 (2x2.4 GHz Cortex-A78 & 6x2.0 GHz Cortex-A55) RAM: 6 GB	Tester: the app worked very well on my device. I found the account setup easy to use and there were no issues with the app. Me: Again, this result is consistent with having an easy to use interface and performant app. I saw the tester using the app and the

		animations were very smooth on their phone and seemed be adapting well to the higher refresh rate display.
15 - 20	Device: Asus Zenfone 9 OS version: Android 13 CPU: Snapdragon 8+ gen 1 (1x3.19 GHz Cortex-X2 & 3x2.75 GHz Cortex-A710 & 4x1.80 GHz Cortex-A510) RAM: 8 GB Other relevant information: Phone is rooted and runs Google Pixel ROM	Tester: The app was easy to use. I did not expect the nearby users feature to work but it worked perfectly fine. I further tested this app by going on an elevator to test the altitude feature and it worked as intended, recording what seemed like correct measurements. The leaderboard seemed slow to update. There was one bug though, when doing one run after another the steps value did not reset. Me: The user has tested the app in a variety of circumstances, including a specialised altitude test. It was interesting that they reported the leaderboards being slow to update, checking the server logs I could not find any real explanation beside the possibility that poor network conditions may have prevented data from syncing with the server and hence the leaderboards may not have been updated completely. The bug the user mentioned was one that I had found previously but forgot about, it was promptly fixed by making the steps variable reset on a new run start.
40 - 45	Samsung A23 (my device but with another user testing to help provide live test data, the app was reset prior to the test) OS version: Android 13 CPU: Snapdragon 680 4G (4x2.4 GHz Kryo 265 Gold & 4x1.9 GHz Kryo 265 Silver) RAM: 4 GB	Tester: I found the app easy to use. It was not laggy at all and the maps added at the end of the run were a nice touch. Me: This result once again proves that the app is easy to use for a wide variety of people and can run mid-range devices.
30 - 35	Device: Samsung S22 5G OS version: Android 13	Tester (software developer): UI is intuitive and overall the app looks very polished, One minor bug: when you start walk session, it shows stop

	CPU: Snapdragon 8 Gen 1 (1x3.00 GHz Cortex-X2 & 3x2.50 GHz Cortex-A710 & 4x1.80 GHz Cortex-A510) RAM: 8 GB	run to finish the session. I think it should instead show the type session, the Google Map didn't appear at the end of the run because I didnt have internet on the phone at that time – it appears now in the history Me: Good to see that a software developer approves of my UI and the project worked correctly on his device, including the no-internet sync feature which he must have automatically used, when he had no internet