GymNote Technical Report

Dermot Grace – 20081469

# Functionality

## Overview

The idea for GymNote was born from an issue I face while recording the details of workouts I do at the gym. My programming is supplied through an app but this app does not have a note taking feature or ability to log something as you go. You need to log and submit your workouts in one go. I had previously been using the notes app on my iphone but that’s not ideal as those notes are mixed in with other general notes.

A solution was to create the GymNote app which would house my notes before I submit them online.

All user logins and user data will be stored in a Firebase Realtime Database.

## What it does

Each user has a login on the app and once logged in they will be able to create a new note, save and update it and view their notes. Each note will be created as ‘incomplete’ and will remain in that state until the user submits it to their own chosen final location. This could be in a notebook, to a personal trainer or another app/website.

For me, my notes will be incomplete until I log them onto a website or another app.

## A sample use case for a new user

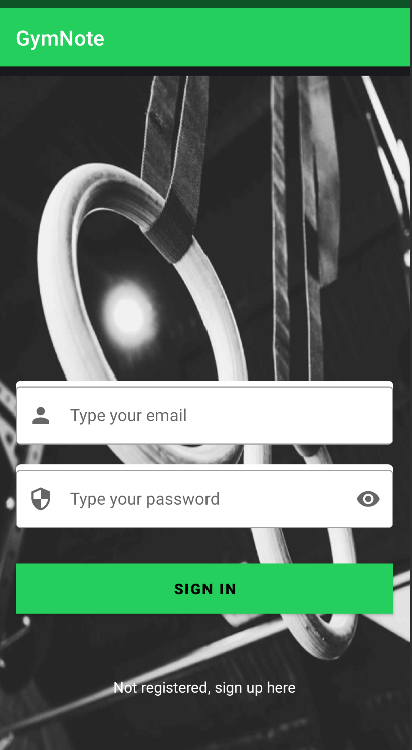
#### Launch screen

A splash screen will be displayed for 3 seconds before the login screen is displayed:

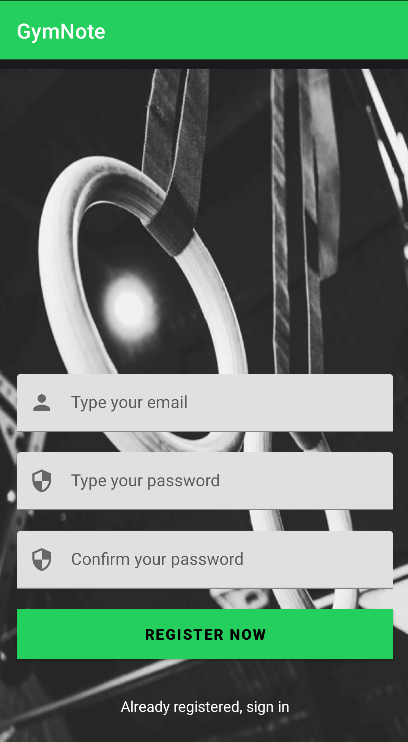


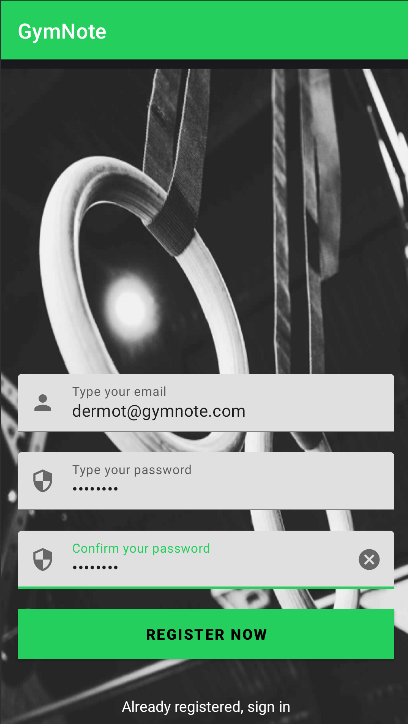
#### Register

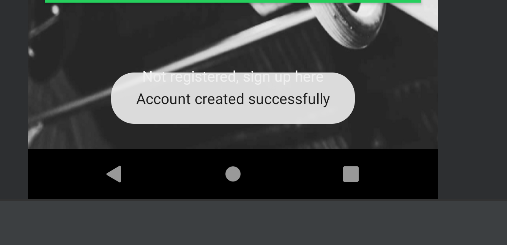
User will click the ‘Not registered, sign up here’ link at bottom of login page.



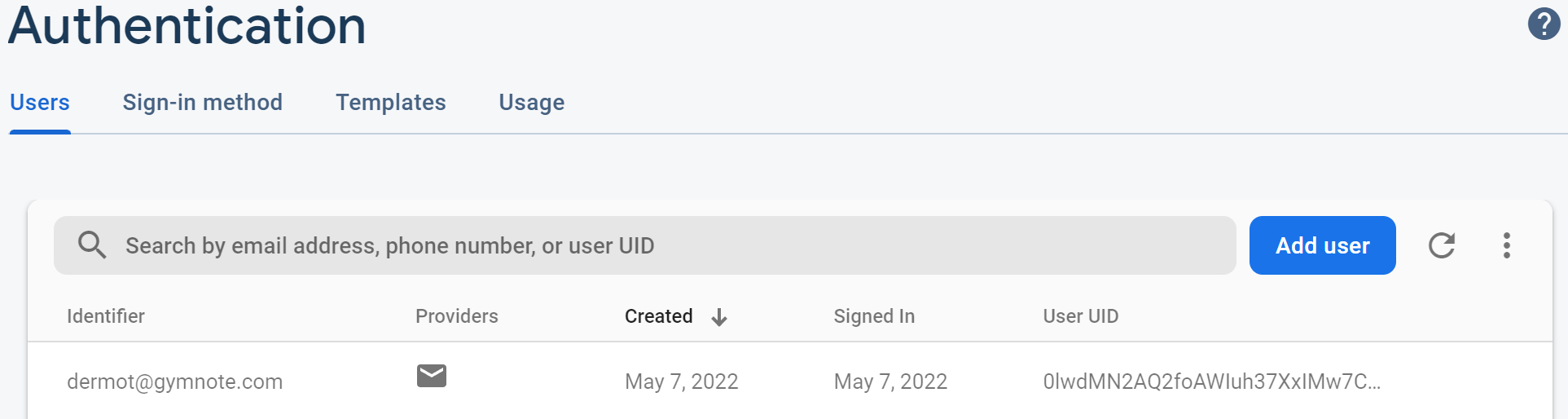
User will input their email address and chosen password and click ‘Register Now’. A success/error message will be displayed on account creation or if there’s an error.







The user will be created in Firebase Authentication. The user uid will be used by the app for filtering of notes when the user logs in.



#### Notes list

No notes will be displayed once a new user logs in.



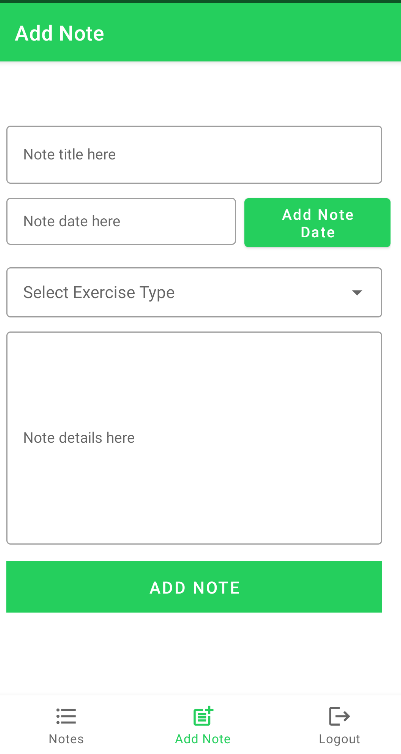
#### Navigation

Three navigation options will be displayed on the bottom of the app with the following options:

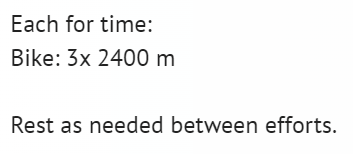
* Notes
* Add Note
* Logout

#### Add a note

User will click on ‘Add Note’ and the following screen will be presented:

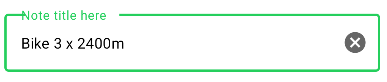


Example workout to log:

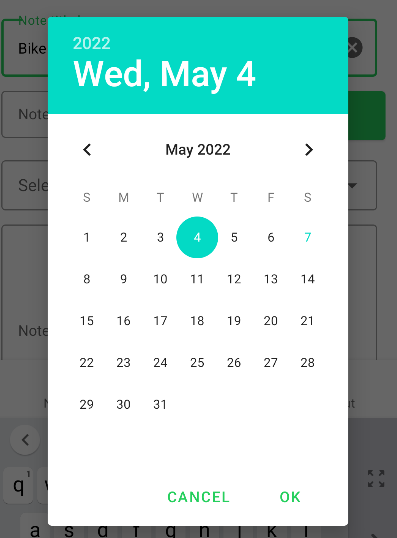


The following options will be presented:

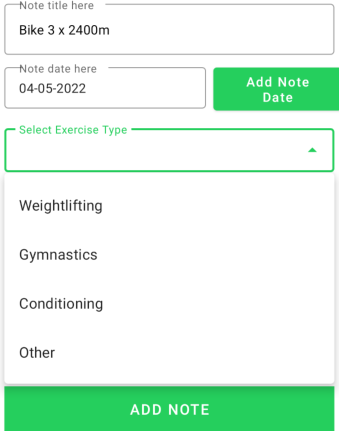
* Note Title
  + Free text entry for the note title. The title will be displayed in the notes list



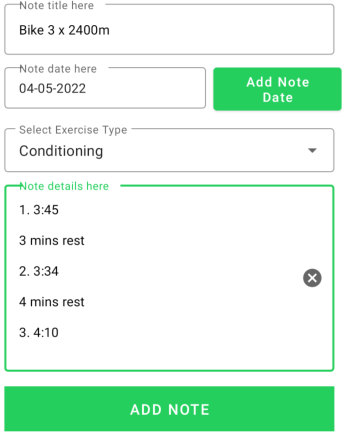
* Note Date
  + Date entry for the note



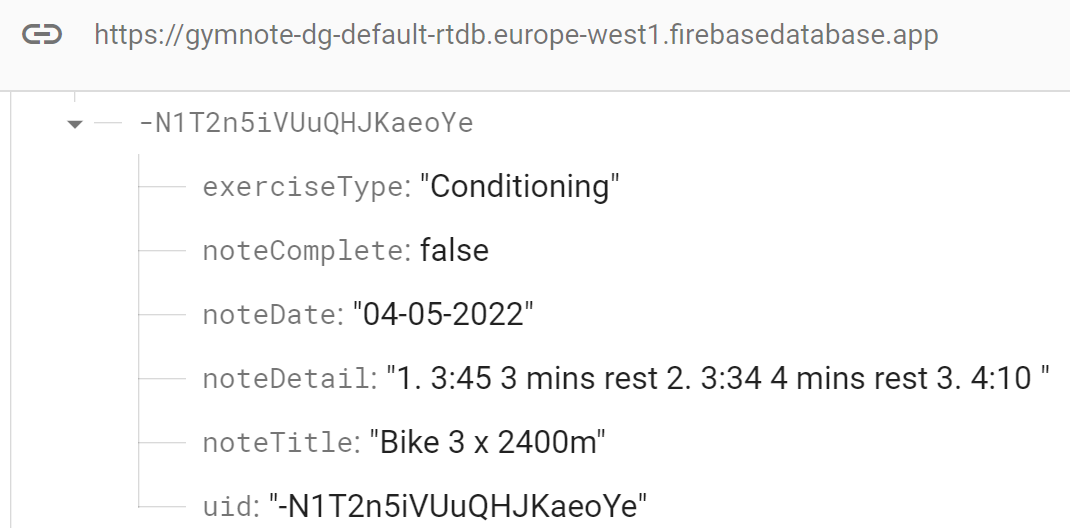
* Exercise Type
  + Select list for the exercise type. Options available are:
    - Weightlifting
    - Gymnastics
    - Conditioning
    - Other



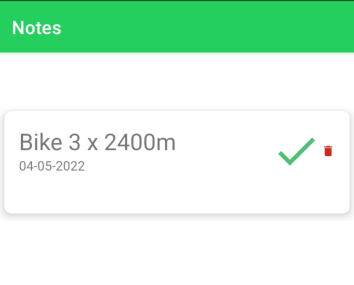
* Note Details
  + Free text entry that has multiple lines for the note details



User clicks ‘Add Note’ which will save the note with a status of incomplete to the database.



This note will also be displayed in the notes list with a check mark(to complete) and a bin(for delete)

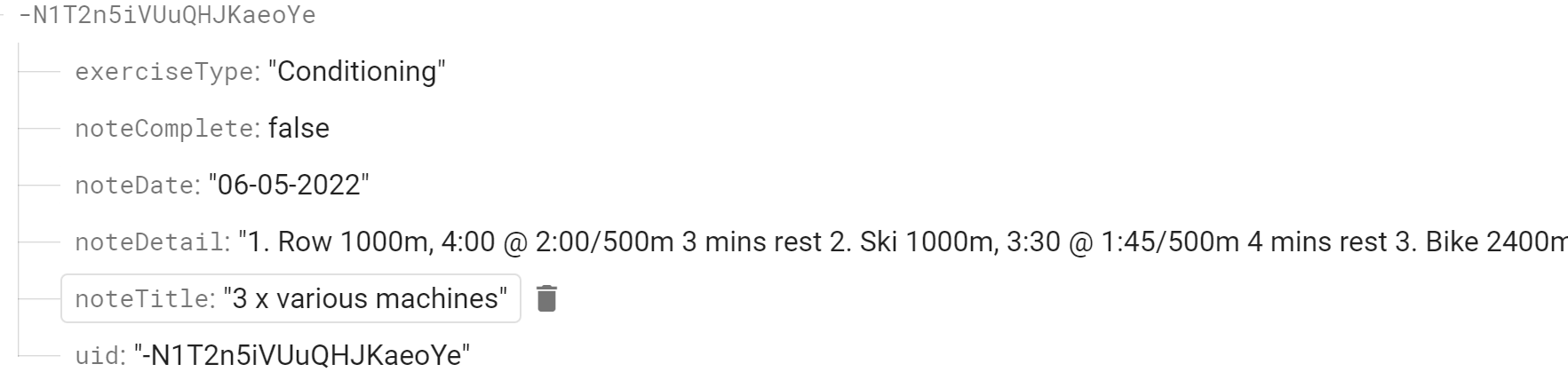


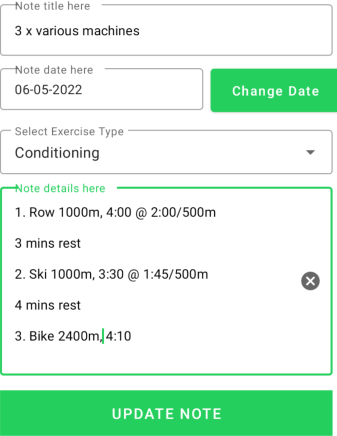
The user has three options at this stage:

* Edit the note by clicking on the card in the view and update it. Text on buttons will be updated to signify that it is an update.
* Mark the note as complete(to signify that they have added it to their program)
* Delete note

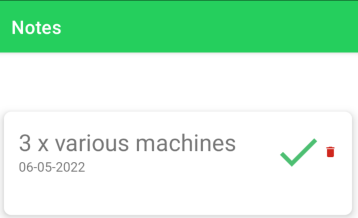
#### Update/Edit note

Clicking a note in the view will open it for update. Updating the note added above to these details will update the note in the database to the details input. Note that the unique id of the note is the same as the above(original) note.



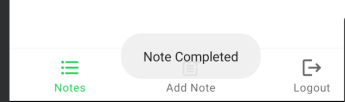


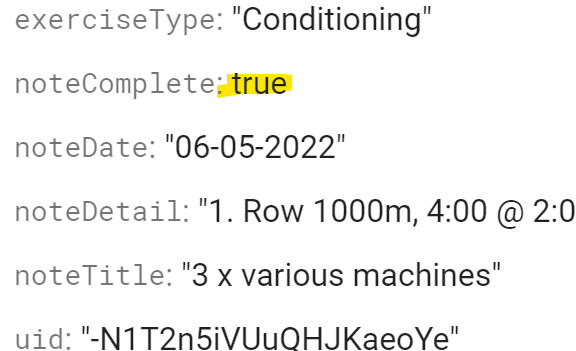
The note will also be updated in the notes list:



#### Complete a note

Clicking on the check mark on the note will ‘complete’ it and it will not be shown any more. A toast will be shown to signify what has been done. This change will be reflected in the database.



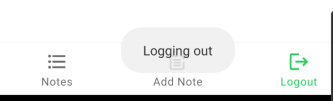


#### Delete a note

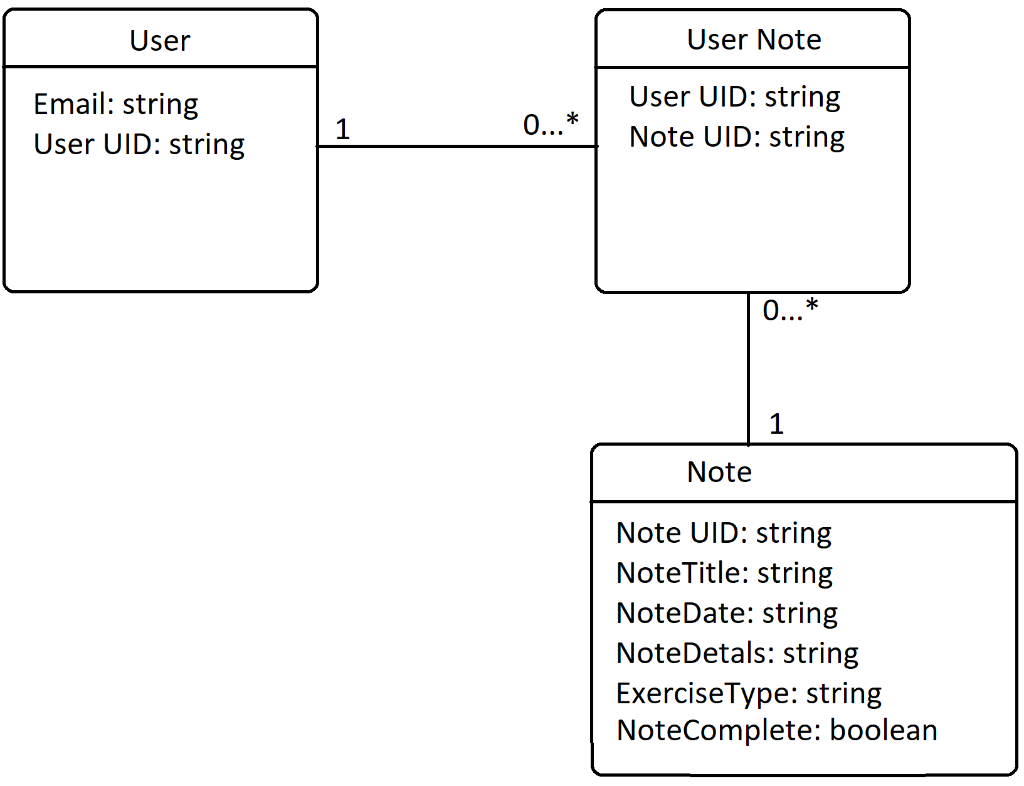
Clicking the bin icon will delete the note from the database and it also will not be shown any more.

#### Logout

Clicking on the logout navigation icon will log the user out and return them to the login/register screen.



## Class Diagrams



# UX/DX Approach

The approach taken was to minimise the amount of clicks a user would need to add a note and keep all options available to the user via the navigation menu within a couple of clicks.

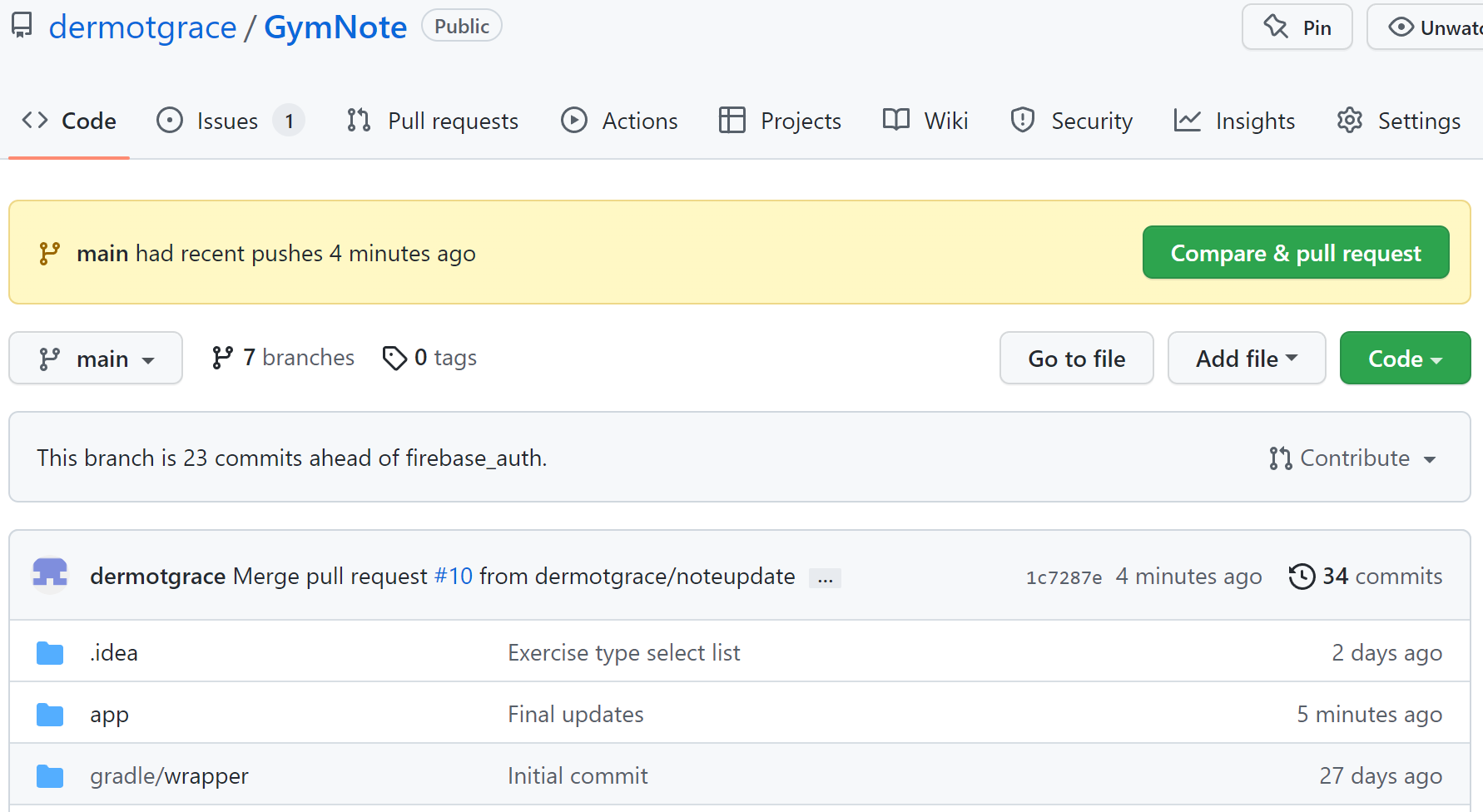
At most a user will take two clicks to get to the add note screen, and be within two clicks for completion or deletion of a note.

Visually, the app has a clean and minimal look on each screen. The only screens with background images are the login/register screens with the notes list and add note screen having a plain white background.

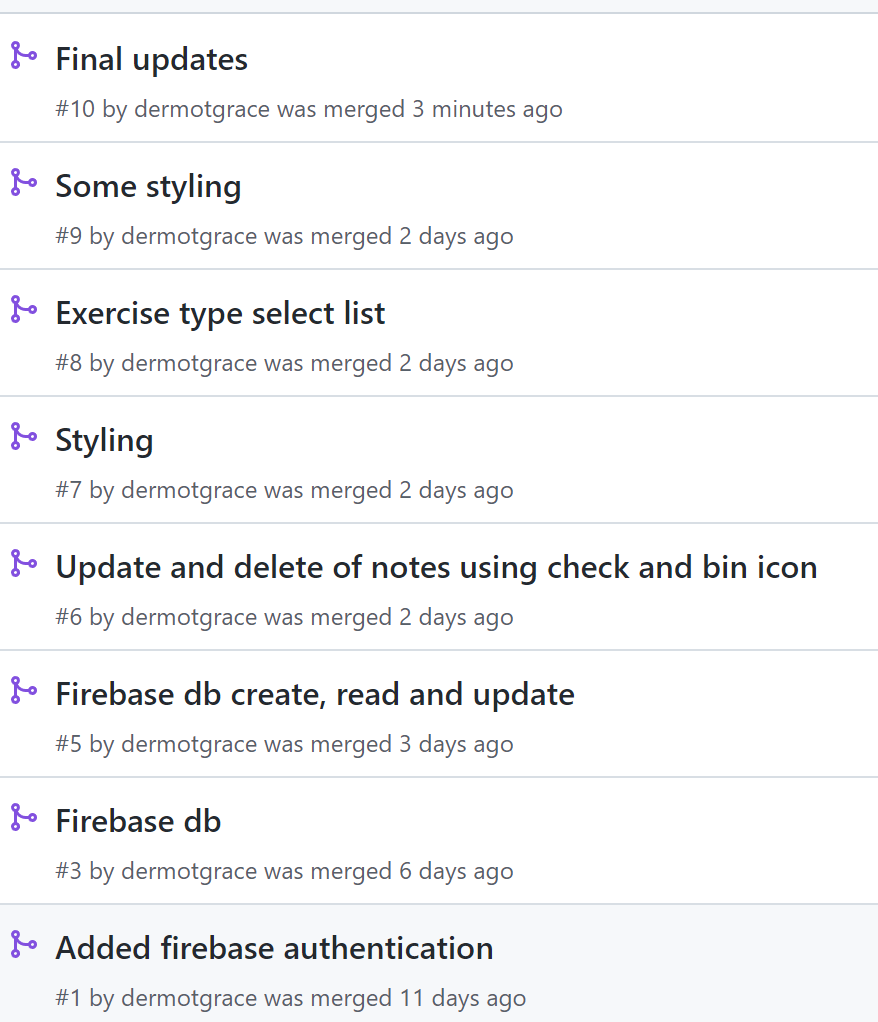
All navigation is performed via the bottom navigation menu and three options were included only.

# Git Approach

A master branch was used and pull requests into this branch were performed for each feature that was added. This approach allowed the reversal of changes if a feature set did not perform correctly.

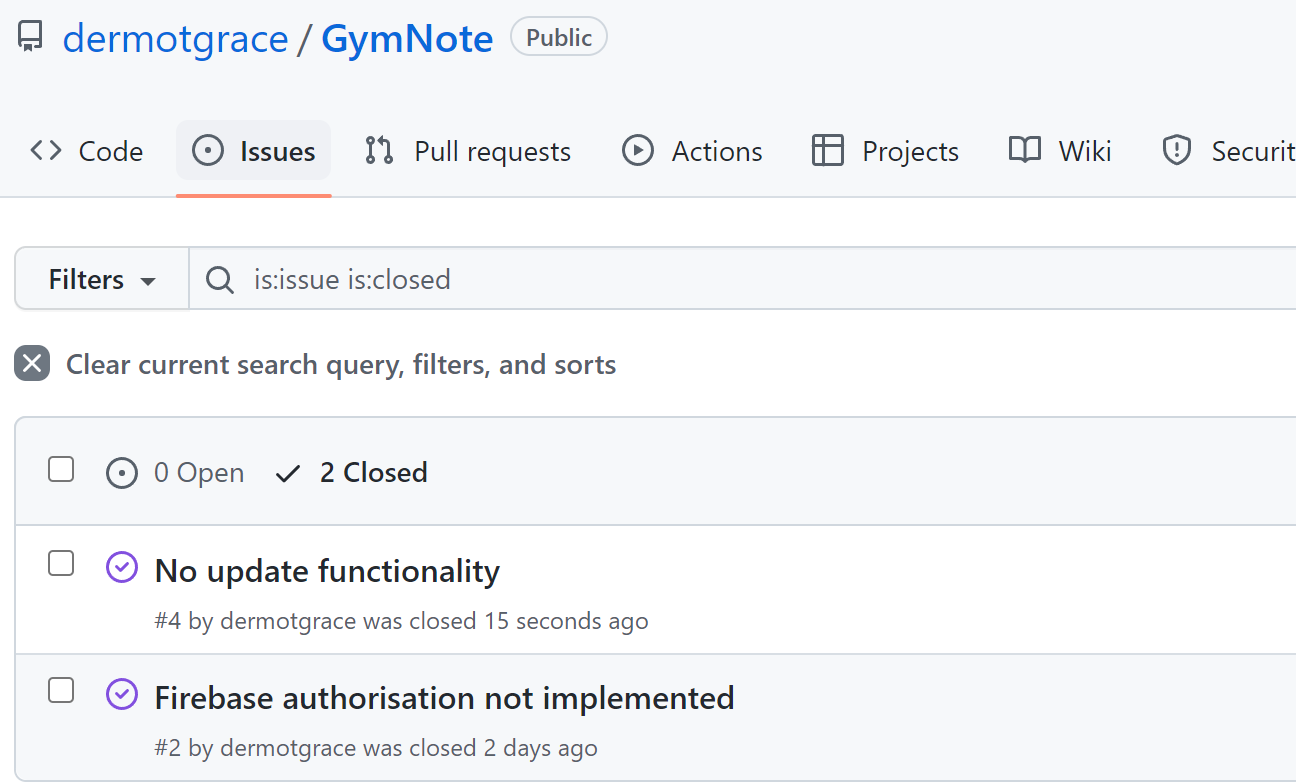


#### Pull requests



#### Issues

Issues were created on GitHub to simulate bugs/features/DevOps tasks that would be encountered in the day to day development of an application:



# Personal Statement

## How it went

I work as a full time developer using Javascript and other other object orientated languages(C#/C++) so programming in Kotlin was not too much of a jump. I had also previously programmed using this language in another module so had some experience with the language and IntelliJ.

I work fulltime and was given two hours each week to watch a lecture or do lab work. I also spent time during the evenings each week to do lab work or the assignment. This worked out well until halfway through the module when the workload started getting more substantial and time started getting tight. I’m a parent of two small kids and their sleep schedule was not the most accommodating to my study/lab/assignment needs.

Deciding when to start the assignment was also something I struggled with during the module. I knew I needed to make an app with authorisation, an online database and navigation menu but had to wait until that material was covered in lectures before I could start it. I ended up starting the app a few times but then stopping with that version and starting again when the appropriate material was covered in lectures or labs. In the end though I feel I started the assignment at the correct time.

There were some health issues in my family also during the last couple of weeks before the assignment submission which drastically reduced the time I could spend on the assignment. I had to spend some days at Beaumont hospital with my wife during that time as well as working when I could so time was made very tight. I did manage to get a couple of days off work though to finish what I could of the app. After discussion with my lecturer, I decided to submit the assignment on time instead of getting an extension as mentally there had been enough happening the last few weeks and I wanted to get it submitted and off my plate.

## Possible enhancements

There are upgrades to the app that I would have liked to implement given the time(or a version 2)

* Styling – There is basic styling on the app and I would have liked to make it more visually appealing
* Filtering – I would like to implement a date filter, complete/incomplete filter etc on the notes list
* Recyclerview card – The card used for the recycler view could contain different information but title and date were decided on

## Known bugs

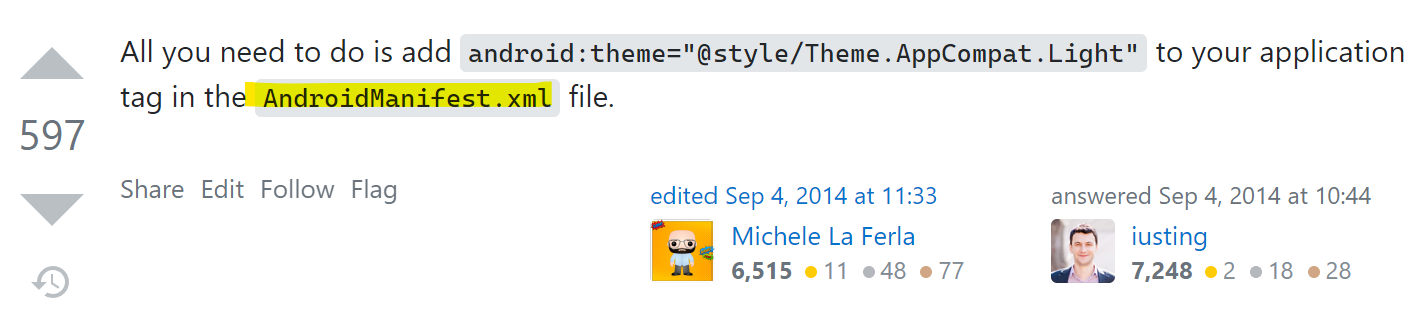
There are a couple of bugs in the app and are logged on github as issues:

* Editing a note will not return you to the notes list and the navigation bar does not work after the update
* Text is centred in the note details text input on the add note screen

# References

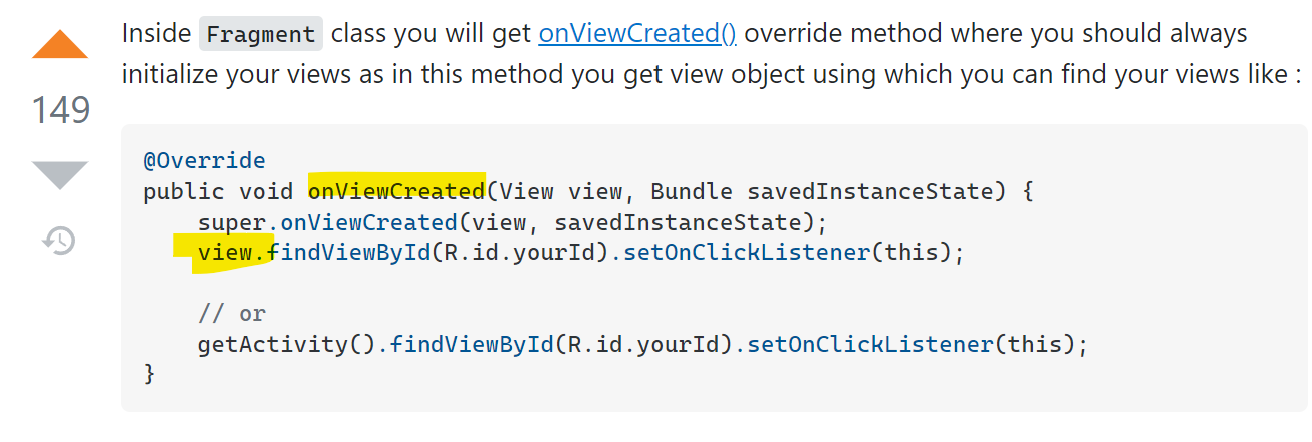
During the development process I leaned heavily on the lectures, labs, android developer guid and youtube tutorials to implement features. Stack overflow was also used to find the root cause of particular errors that occurred during development.

Some references below are for features that I experimented with or were needed at one point but not needed as the app progressed.

1. Splash screen video tutorial: <https://www.youtube.com/watch?v=Loo4i5IrZ4Y>
2. Error when clicking add: “The style on this component requires your app theme to be Theme.AppCompat”. [Stack overflow link with hint](https://stackoverflow.com/questions/21814825/you-need-to-use-a-theme-appcompat-theme-or-descendant-with-this-activity)

I inspected my AndroidManifest and saw I had no theme set for the activity I was launching. Setting this fixed the issue.

1. Date picker on Add Note fragment: [Android developer guide](https://developer.android.com/guide/topics/ui/controls/pickers%23kotlin)
2. Using findViewById in fragment: [stackoverflow](https://stackoverflow.com/questions/6495898/findviewbyid-in-fragment%20)



1. Passing data between fragments using communicator interface: [Youtube link](https://www.youtube.com/watch?v=7sN53PQ4jr8)
2. Manually navigating between fragments on button clicks etc: [Andriod developer page](https://developer.android.com/guide/navigation/navigation-navigate#id)
3. Error connecting to database: Database lives in a different region. Please change your database URL to xxxxxx. Database url is specified correctly in google config file but application was not picking it up. [Stackoverflow answer](https://stackoverflow.com/questions/65558463/firebase-database-points-to-wrong-database-url)
4. Exception thrown when inflating the SignupActivity: java.lang.IllegalArgumentException: The style on this component requires your app theme to be Theme.AppCompat (or a descendant) <https://stackoverflow.com/questions/21814825/you-need-to-use-a-theme-appcompat-theme-or-descendant-with-this-activity>
5. Checkboxes on recycler view card <https://developer.android.com/guide/topics/ui/controls/checkbox>
6. Inserting check and bin icons in recycler view <https://developer.android.com/guide/topics/graphics/drawables>
7. Toggle switch on notes list to filter on complete/all

<https://developer.android.com/guide/topics/ui/controls/togglebutton>