CS31620 Assignment 2023-24 - Part B

Workout Timetable Planner App

Student ID – ktd1

Student Ref - 210064199

Table of Contents:

Design Adjustments & Justification

Several adjustments were made to the original design to improve user experience, accessibility and to fulfil requirements which I had either misunderstood or not noticed and majority are based on feedback received from part A. This was done before implementation so that I would have a clear picture of how the app would look with the several screens associated.

Home

Firstly, the home screen of the app has received a few design changes. Notably the new integrated top app bar to house the settings button and the title text + Icon, the colouring of objects to fit with M3 colour standards and the removal of the “Welcome back User” text which is now a regular “hello there!” text. This has been changed to be more open ended, this could be potentially replaced with encouraging phrases, praise, more advanced greetings or removed entirely as it was only added to make the app more friendly towards the user.

Another important change was the table format itself. Before the table text was quite small and could be hard to read, however I worked on improving this by only placing the title and time to a separate row like in the original “Tomorrow’s Table” design which I used throughout every table within the app. On the “Tomorrow's Table” design, I had put an expand button on the bottom right of the screen, I thought this button was way too small to be user friendly, so I decided to change the size of the button to be on a new row and in the middle with the width being the length of the table.

Settings

The next Screen is the settings which received minor changes. The first is the removal of the username setter as well as the sound effects slider, this is because they were redundant features which had no use in the requirements. The settings menu is also no longer on the bottom app bar, following M3 design it is on the top app bar. The settings menu has one navigation option which is to go back.

Bottom app bar

The next part is the bottom app bar, it is coloured using the regular colour of a navigation bar the on-surface colour. It has 4 destinations, left is home, 2nd is timetable view, 3rd is timetable edit and last is an exercise viewer.

Timetable

The timetable view screen is a new simple screen that only shows the timetable without any interactable buttons like edit buttons, this is to allow the user to view their workout plan without accidentally pressing an edit button interrupting the experience.

Timetable Edit

The next screen is the timetable edit screen which I have kept mostly the same with some minor changes to reduce total taps. Instead of having an edit button revealing more buttons, this screen is dedicated to editing the timetable and has an edit button in the top right corner of each workout day table as well as a new delete button on the bottom of each table. The edit button leads to the edit day menu while the delete will open a dialog to confirm deletion.

Exercises

The final navigation bar screen is the exercises screen which is also a new addition due to misunderstanding the requirements for the application. This screen contains a table like the original timetable screen table. An edit button in the top right which when pressed will expose an edit button for every exercise within the list. When pressing an edit button for a specific exercise it will open an edit exercise menu where you can change the information or delete it. The exercise menu has a Floating Action Button (FAB) which will be used to add the exercise to the list.

Add Day

The next screen is the add day screen which has received a small change with the day selection being replaced for a dropdown list because this would be more suitable than the original design which uses a restructured Nav Bar for the select day.

Edit Day

This is very similar to the add day however it includes an additional dropdown list of the exercises with a table below showcasing the exercise with a remove button at the end of each exercise.

Edit Exercise

This screen was kept the same except when the drop-down button is pressed it shows an additional two number inputs for the drop-down set weights.

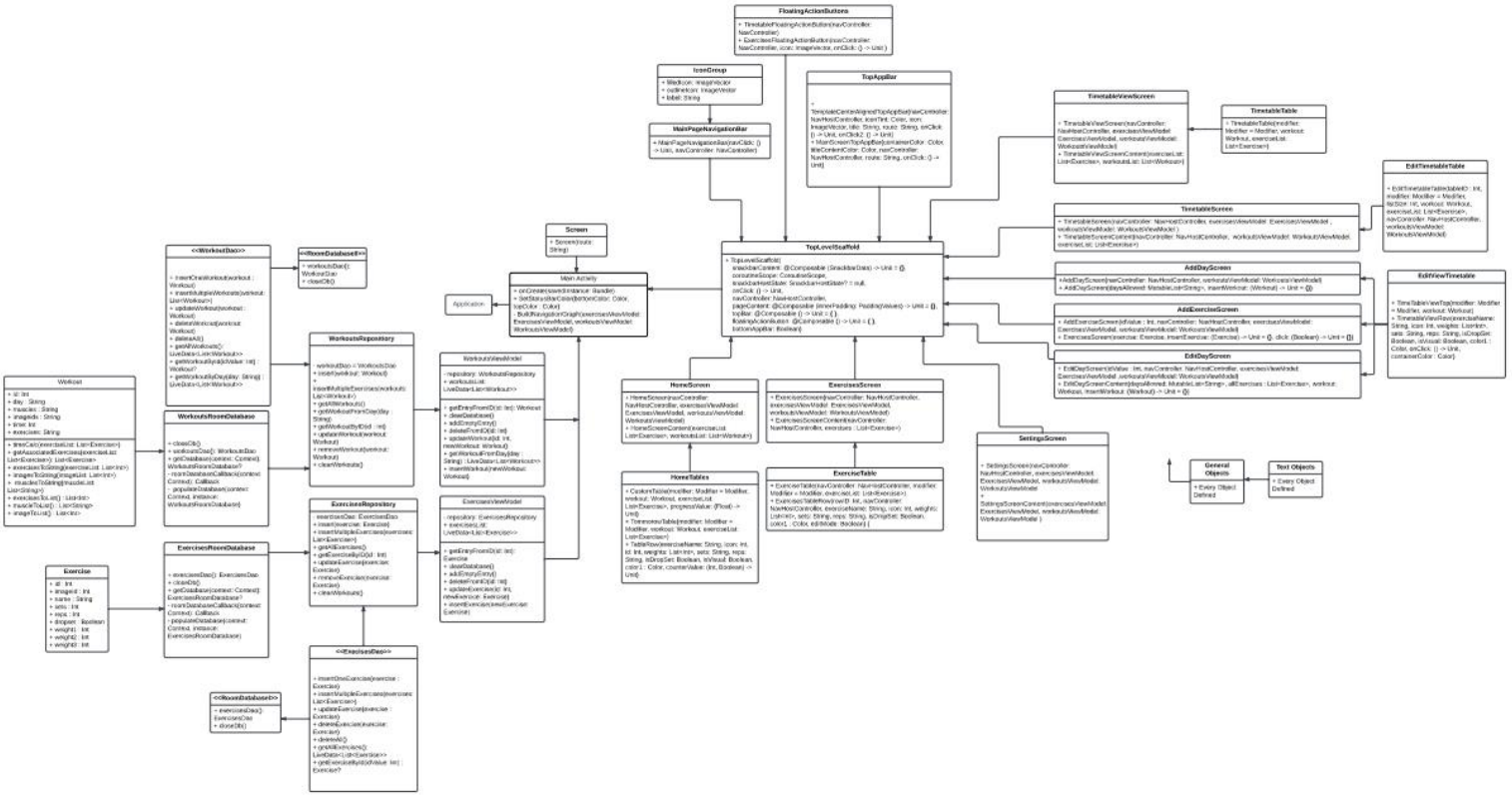
The updated design is showcased in this FIGMA link:

https://www.figma.com/file/8f16gLxzDv2wOIQCLwIvgH/Untitled-(Copy)?type=design&node-id=0%3A1&mode=design&t=oF4FOcBPJyE1Nbmh-1

This is a purely visual prototype and not interactive unlike the original for use as a reference.

Construction of my application:

UML Diagram: (Link in the references)



The colour scheme I used for my application stayed the same as the one in part A of my report.

I used several components to make my application, functional, comfortable to use. The application handles screen rotation very well, not losing any data when rotating the phone with a layout that adapts to varying screen sizes. I have several checks throughout the code ensuring any invalid user input is handled correctly or negated.

I disabled double taps because I didn't need any user functionality that involved tapping multiple things on the screen or zooming in. Double taps caused several issues in code which would be caused by pressing 2 buttons at the same time.

I made use of object-oriented code for parts of the program. I stored my data in classes, made use of several methods to get the functionality that I desired. I used 2 Room databases to gain functional persistent data storage, one database for exercises and one for workouts.

Shared Preferences

I spent several hours trying to implement dark mode using shared preferences. Shared preferences are a way of storing small pieces of information so that the user can access them when closing and reopening the app. I successfully managed to get these shared preferences to function correctly by saving the Boolean state of my dark mode switch every time the switch state changes and loading it when initialising the composable. This allowed me to store and load the shared preference however I was unable to get this value to load in any other scope of my code like the Theme.kt file. I was hoping to get the application to load either dark or light theme based on the shared preference I was storing however it didn't work and I wasn't sure what to do about it.

Room database

The room database is a database abstraction library making the use of a local database in an application much easier to accomplish. I had 2 databases, one for exercises and one for workouts. I had initially planned to link these databases together through parent and child columns; however, I was having too many issues to not use excessive time debugging I instead settled for an easier to manage 2 databases and linked them externally using id values assigned.

Despite my classes holding lists of values, all values that the database stored were basic variables, I used Strings, Booleans and Integers exclusively which I converted to and from the database. I simply created methods within the class to convert lists of a variable to a string separated by commas which I ensured could not be interfered with. Ideally, I would have linked the databases together and used type conversions to achieve simpler code, but I couldn't get the database to function correctly when converting to and from a list, so I instead settled on converting to and from a String externally within the application.

Implementation Order

- I initially started with the home screen, which I implemented the first initial prototype in the main activity. I made the application look as close to the amended design as possible.
- Next to add multiple screen functionality, I added the navigation bar with the nav graph and controller.
- Organised composable objects into separate files for reuse throughout screens.
- I created the class objects for exercise and workouts for use within the home table.
- I next added the Timetable view screen and the view Exercises screen which I created separate table composable for as they all had differences in terms of content within the table.
- I added the add exercise screen (later changed to edit exercise)
- Added the FAB to exercise screen.
- Added the add day screen.
- Added the edit day screen.
- Added the room databases and everything related,
- Allowed the edit screens to interact with the database.
- Added drop sets to the tables.

- Added the class methods for workout. Like the time calc & conversion methods.
- Added the settings menu (Tried to implement the dark mode button)
- Bug fixes and colour changes + final UI changes.

Other important method:

TimeCalc: This calculated the approximate time for the workout, it took in the list of exercises as a parameter and found the exercises corresponding to the workout based on the ID values. It then took the sets on the exercise and whether it was a drop set or not and when it was a drop set multiply the sets by 120 + 30 + 180 which is 2 minutes for the set, 30 seconds in between the drop sets and 3 minutes rest period. If it's a regular exercise I just multiply the sets by 180 which is 3 minutes. I then divide the seconds value by 60 to get minutes and finally by 5 to get 5-minute intervals, I round the value up to the nearest whole number and convert it to int so that I can just multiply the time by 5 to get the approximate time for the workout in minutes.

Testing

To conduct testing of my application I used manual testing because I couldn't get compose unit testing to work on my laptop. I created 31 tests which I recorded and uploaded for visual proof of the application in working in runtime. I have also supplied screenshots as per the requirements.

I passed 30 of the 31 tests I created, the one fail is due to me not fully implementing the dark mode switch within settings. Although I managed to save preference so the application can recall the dark mode switch state when reopening the app, I didn't know where and how to change the theme of the app based on the save preference.

https://www.youtube.com/watch?v=WeAJqJ2xFLU

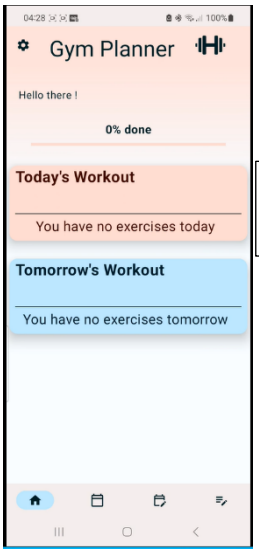
I am quite pleased with the test results and with how polished my application is.

Test ID	Test Description	Input Data	Expected Result	Actual Result	P/F
1	Starting the app starts on the home screen	Opening the app	Shows the home screen	Shows the home screen	P
2	When starting the app for the first time it only has one empty workout for Monday on the timetable screen	Opening the app, navigating to timetable	Shows one workout	Shows one workout	P
3	Navigation between all screens	Navigating to every screen from every screen	It navigates to the correct destination	It navigates to the correct destination	P
4	Exercises Screen FAB adds one exercise to the list of exercises	Go to exercises menu, press the FAB	Adds one exercise to the list	Adds one exercise to the list	P
5	Clicking the edit button on two exercises	Adding 2 exercises in the exercises menu and pressing edit on both	Opens the edit menu for only one exercise	Opens the edit menu for only one exercise	P
6	Adding the same exercise to different workouts	Add 2 days on edit timetable screen, add the same exercise to both workouts	It displays the workouts correctly without issues	It displays the workouts correctly without issues	P
7	Deleting an exercise removes it from the list of exercises	Adding 2 exercises in the exercises menu and pressing edit on both	Opens the edit menu for only one exercises and d	Opens the edit for only one menu,	P
8	Resetting the workouts in settings	Navigating to the settings, press the reset button	Resets the workouts list in timetable and adds Monday	Resets the workouts list in timetable and adds Monday	P

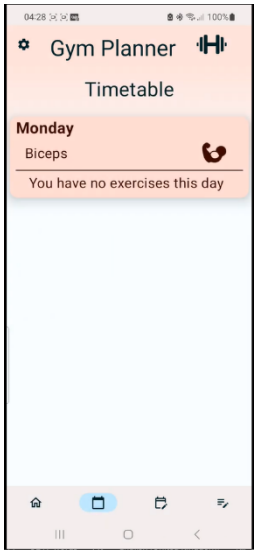
9	Resetting the exercises in settings	Navigating to the settings, press the reset button	Resets the list of exercises in the exercise's menu	Resets the list of exercises in the exercise's menu	P
10	Dark mode switch in settings	In settings, press the dark mode switch	Changes the device theme	Does not change the device theme	F
11	Add a day in edit timetable	In edit timetable screen add a day	Adds a day	Adds a day	P
12	Check adding a day doesn't accept	Press on FAB in edit timetable, press save without a day selected	Doesn't save and gives a warning	Doesn't save and gives a warning	P
13	Check home menu for workout display if you add one for the same day as today	Add a workout day for today in edit timetable.	Displays a blank workout on the home screen	Displays a blank workout on the home screen	P
14	Check if exercises display on the home screen for today	Edit the today workout by adding an exercise	Displays the exercises on the home screen with a checkbox for each exercise	Displays the exercises on the home screen with a checkbox for each exercise	P
15	Test the checkbox on the home screen	Tap the checkbox on the home screen	Check box ticks, table row greys out, progress bar updates	Check box ticks, table row greys out, progress bar updates	P
16	Test if the checkbox state is remembered with the progress bar	Test 15 -> Then navigate to other screens then back	Still displays the same selected checkboxes	Still displays the same selected checkboxes	P
17	Tomorrow table displays a workout when there is a workout day for tomorrow	Edit timetable add a workout for the day tomorrow, go to home screen	Displays an empty workout for the tomorrow table	Displays an empty workout for the tomorrow table	P
18	Tomorrow table gains a dropdown button when there is an exercise tomorrow	Edit timetable, edit tomorrow exercise, add an exercise, save and go to home screen	Home screen displays a button on the bottom of the tomorrow table	Home screen displays a button on the bottom of the tomorrow table	P
19	If the dropdown button is pressed on the tomorrow table, it shows the exercises on from tomorrows workout	Add exercises to the workout for tomorrow, go to home and press the dropdown	It displays the exercises upon pressing the dropdown button	It displays the exercises upon pressing the dropdown button	P
20	Tomorrow dropdown menu collapses when pressing the button when expanded	Open and close the tomorrow timetable	The dropdown menu closes	The dropdown menu closes	P
21	If deleting a workout gives a dialog that must be confirmed	Add a workout on edit workout, then	A dialog shows up, when	A dialog shows up, when	P

		press the delete button	accepted deletes the workout	accepted deletes the workout	
22	If deleting an exercise shows a dialog that must be confirmed	Add an exercise, edit it, press delete button	A dialog box appears asking the user to confirm deleting	A dialog box appears asking the user to confirm deleting	P
23	If you add an exercise to a workout then change the original exercise, if it changes in the workout	Add an exercise to a workout, change the exercise, check the timetable	The exercise changes in the timetable to the updated version	The exercise changes in the timetable to the updated version	P
24	Making the exercise a drop-set makes it display as a drop-set in each table	Add an exercise, make it a drop-set, add to workouts in timetable	The drop-set exercise is clearly visible as a drop-set exercise	The drop-set exercise is clearly visible as a drop-set exercise	P
25	Adding an exercise with more than 0 sets to a workout will change the approximate time	Make an exercise with 10 sets. Add it to a workout, check timetable	The timetable displays the time as 30 minutes for the workout	The timetable displays the time as 30 minutes for the workout	P
26	Backing out of a change exercise without up saving, doesn't update the exercise	Add an exercise, open the edit menu, change the name to test and exit	The timetable still displays the exercise as exercise	The timetable still displays the exercise as exercise	P
27	If you can add multiple muscle groups to a workout.	Go to edit timetable menu, add a day, select biceps and quadriceps from the muscle's menu	The workout for the day selected shows biceps and quadriceps with the icons	The workout for the day selected shows biceps and quadriceps with the icons	P
28	That you can only input numbers into the number text fields on the exercise menu	Go to exercises, add a new exercise, edit it, try to input characters.	The number input doesn't allow anything apart from numbers	The number input doesn't allow anything apart from numbers	P
29	Changing the sets in an exercise within a workout updates the approximate time	Test 25-> Change the sets to 2 sets on the exercise	The time shown in the timetable is now 10 minutes	The time shown in the timetable is now 10 minutes	P
30	You can only change the day within a workout to a different available day.	Add 3 new workouts Monday, Tuesday, Wednesday, press edit on one day,	The days available within the dropdown are the rest of the week + the	The days available within the dropdown are the rest of the week + the	P

		check what days are available	day of the one editing	day of the one editing	
31	The workouts appear in weekday order (Mon-Sun)	Add a few workouts with different days, check the order of workouts in timetable	The workouts appear to be in order of weekday starting on Monday	The workouts appear to be in order of weekday starting on Monday	P
32	The workouts are kept after closing the app	Add empty workouts, restart the app	The additional workouts still appear	The additional workouts still appear	P

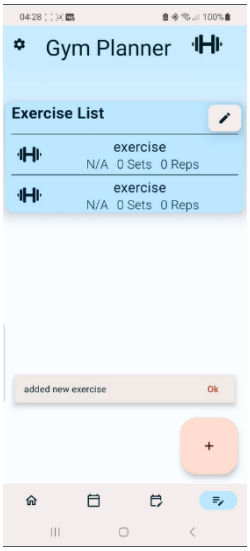


Test1



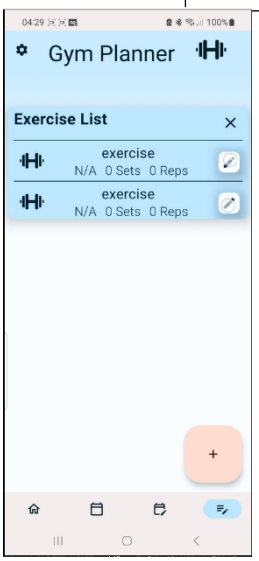
Test2

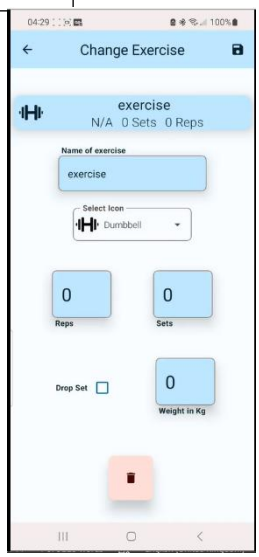
Test3: Video



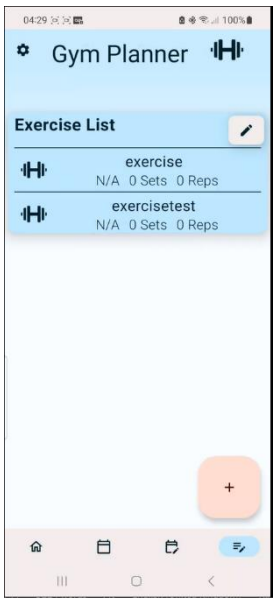
Test4

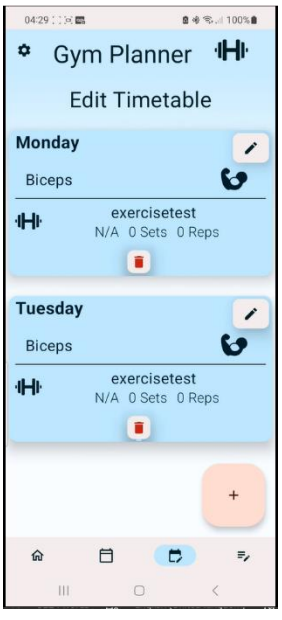
Test 5



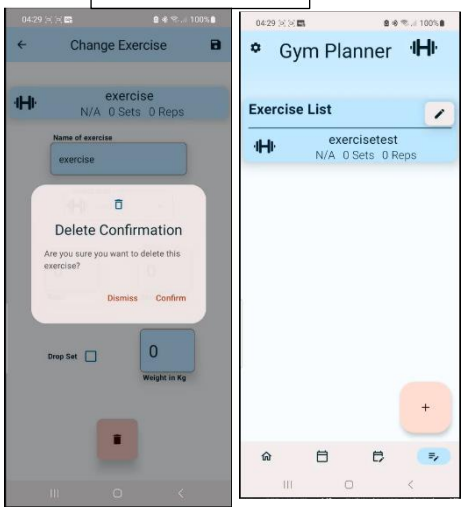


Test 6

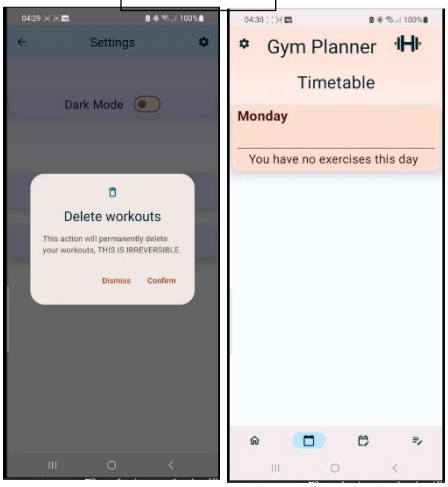




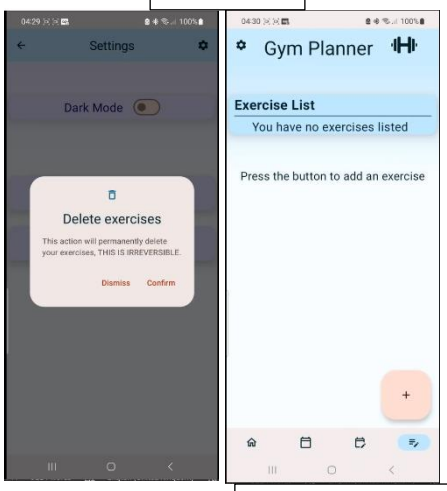
Test 7 & 20



Test 8

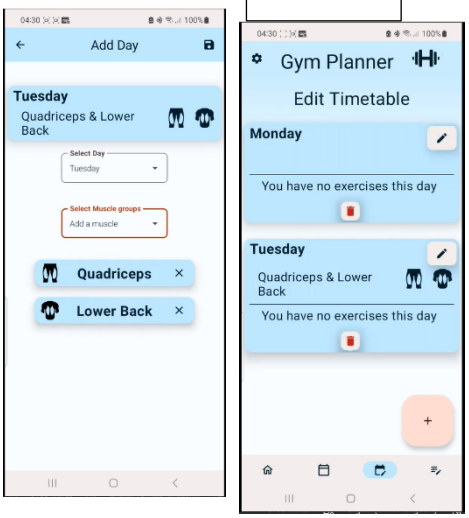


Test 9

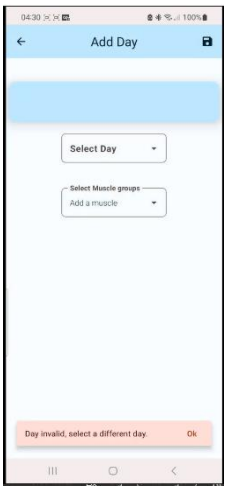


Test 10: Video

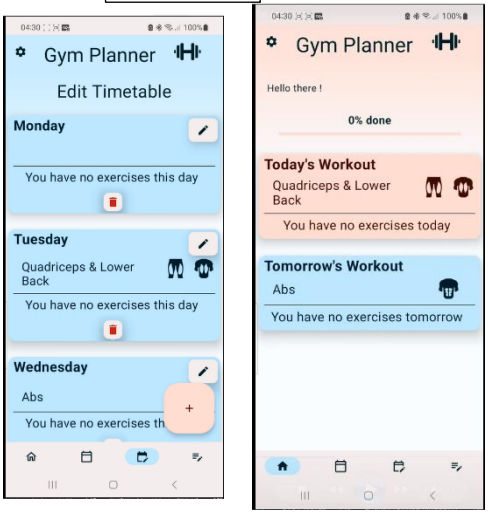
Test 11



Test 12

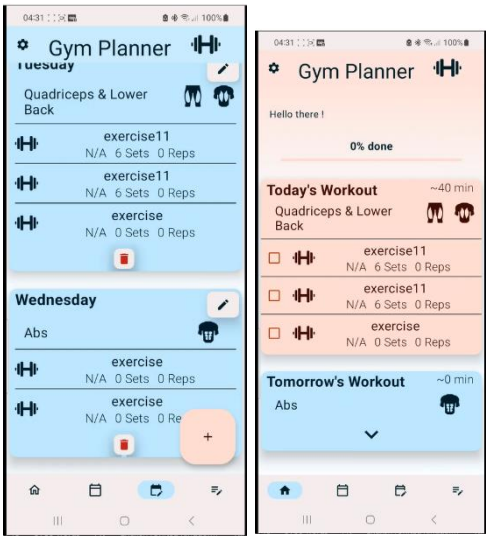


Test 13

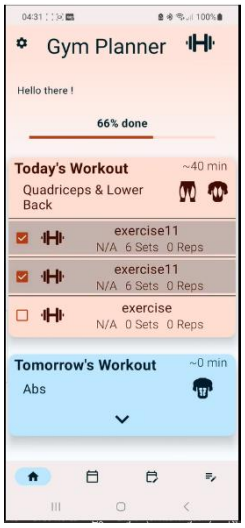


Test 16: Video

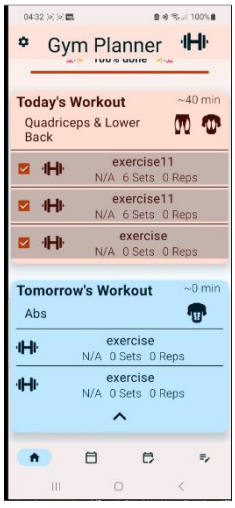
Test 14 & 17 & 18



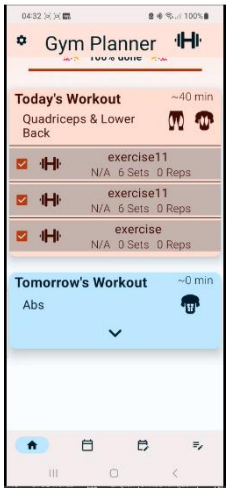
Test 15



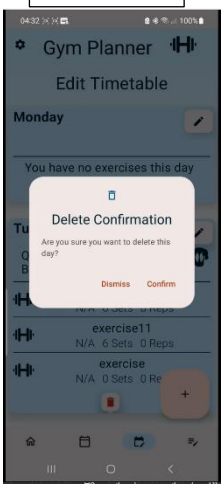
Test 19



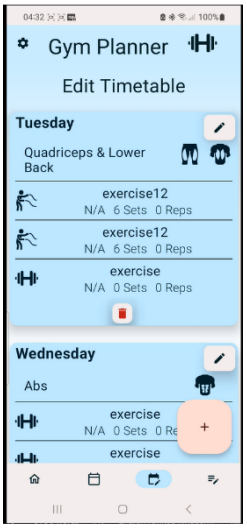
Test 20



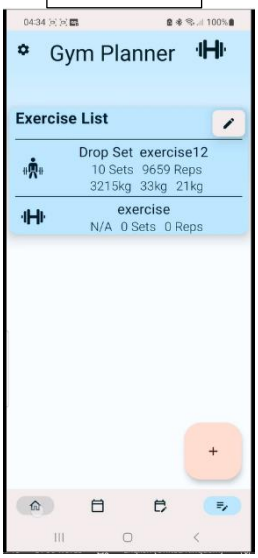
Test 21



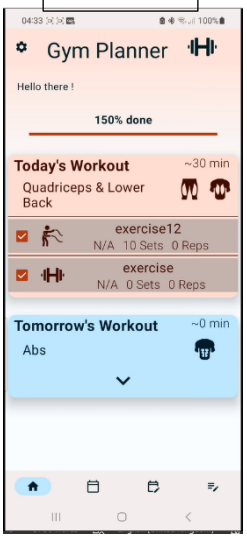
Test 23



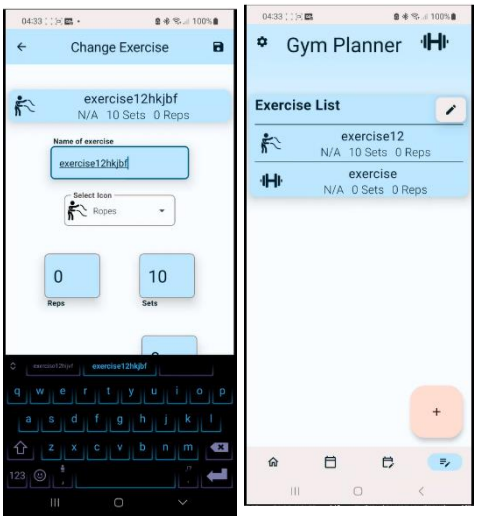
Test 24



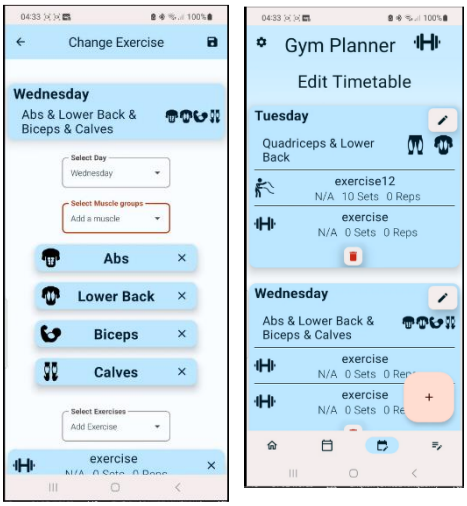
Test 25



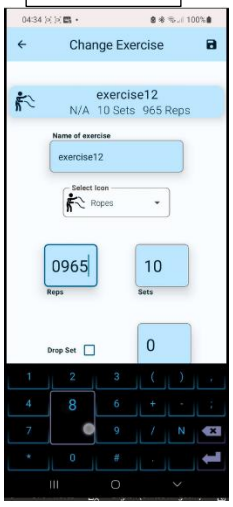
Test 26



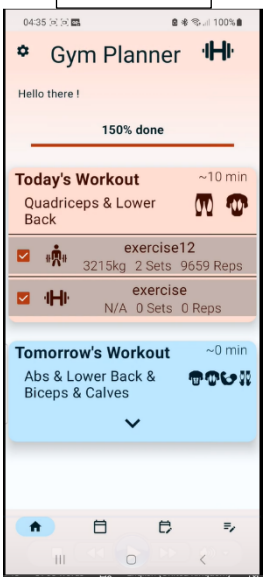
Test 27



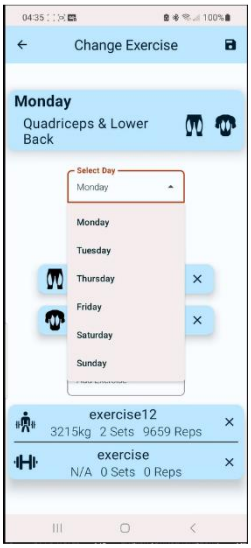
Test 28



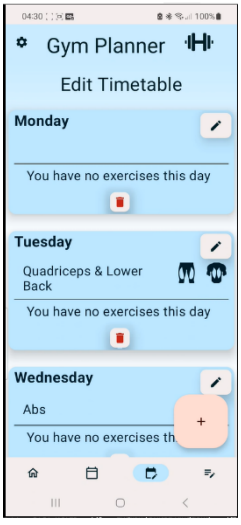
Test 29



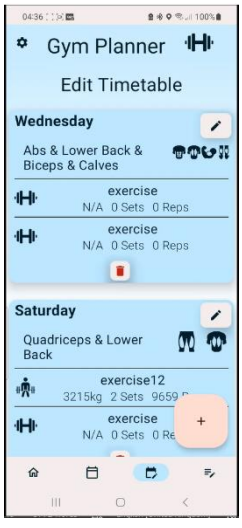
Test 30



Test 31



Test 32



Discussing a possible REST API for remote persistence

For my separate web server, I chose the RESTful API Firebase Realtime Database, this is because the data I'm storing is relatively small in content.

Example:

GET https://gymApp.firebaseio.com/users/userID/workouts.json

Sending headers: Authorization If-Match, If-None-Match

Received headers: Content-Type : application/json Location Cache-Control

Status codes: 200 OK & 404 Not Found

Content received: The data would be sent and received in json format.

The userID would be replaced with the users actual ID, and it would return the workouts json for the corresponding user.

I chose firebase for several reasons, the first being that it is a real time database. This means changes can occur synchronously between multiple connections. Allowing for live updates to the database.

Firebase is very well documented meaning it would allow for easy implementation into the app. It has plenty of in depth guides and loads of information available to the user.

A different web server RESTful API could potentially be MongoDB as it provides high level NoSQL Querying capabilities making it better than firebase in some respects. One of the main reasons I choose firebase over MongoDB is that it has offline support. This is very important for the gym app, often a user at the gym may not have internet access and using MongoDB this app would be rendered ineffective, however firebase will allow for offline use of the database and simply update interactions when coming online again.

Firebase also has the advantage of no server management tasks. This is important, mostly for smaller teams where time spent managing a database server could be seen as downtime.

A different API that has is functionally better in some respects is Apache Cassandra. This API excels in handling large amounts of data allowing for read and write queries to be much more effective in this database. However, firebase is much more robust, its simpler to use and has global servers, which allows for data to be transferred across the world much faster than Cassandra.

Firebase is the right choice for an app starting off, with the ability to migrate the database to a more suitable candidate later along the line when data scaling starts to become an issue for the firebase. It is perfect for the initial database due to the abstraction it provides, making it the simplest choice to develop with for a smaller team.

Evaluation and Reflection

UI Changes

Before implementation I created a revised UI based on feedback received in part A. I tried to fulfil user requirements while keeping the application intuitive and easy to use. However, some small changes I made in the final implementation that differ from the design are the addition of:

previews of the content in the edit day/exercise. This was to show the user what they're adding visually before they press save.

I also changed text for certain buttons for icons in the top app bar, this was to require less reading and more visual cues to identify what a button does immediately.

I also changed the contents of the settings to include database reset buttons which give the user an easy way to clear everything without having to manually press delete on every individual item while keeping it in a place that isn't accessed commonly, within the settings.

I changed some of the original colours to match the M3 colour scheme.

I added icons to the top app bar buttons instead of it being text buttons. It gives the user more information faster than reading text.

I added snackbars to my program which notify the user when they add an exercise to the list, and one which warns the user to select a valid day in the add day screen.

I also added dialog boxes to confirm whether the person wants to delete a workout or exercise.

Evaluation

Overall, I am very happy with my final application, I find it quite enjoyable to use and it fulfils every requirement stated in the assignment while staying nice in appearance. I think the code is well documented, with comments describing the code clearly, the code is robust allowing for new screens to be added with ease. The files are also well organized in the code with meaningful names that can be located easily, and I have added several additional features which improve user experience throughout the app with the main one being the home screen entirely.

Reflection

The areas to improve in my opinion are a slightly improved database structure, removing code redundancy in areas (for the tables and drop down composables) as the code was similar, fixing the preferences datastore problem for the dark mode switch in settings and fixing minor bugs on certain menus like screen rotation on add day or edit exercise as the program recomposes and since I'm storing a class variable crucial to updating the exercise or workout, I cannot use remember Saveable to keep its state on recomposition, I believe I could have added several variables to hold the state in recomposition but I didn't like that

solution. If I had planned slightly better, I believe I could have made a complete app with little to no bugs with all the features I originally wanted.

Oher than that I believe that considering the quality of the application and its potential for expansion I think this project deserves at least 80%.

Tools Used + References:

Icons by Icon8 – icons8.com

Android Studio

Figma Design

UML Diagram made in LucidChart:

https://lucid.app/lucidchart/a631145d-6db4-4034-96d5-c5ef1eb0619d/edit?viewport_loc=-2105%2C-693%2C6182%2C2747%2C0_0&invitationId=inv_b9746e38-e88e-448f-9a03-fb77a5742dbc