

CS31620 Mobile Development with Android Assignment 2023-24: Workout planner

Chris Loftus

Release date: Tuesday, 17th October 2023

Weighting in module: 80%

Submission date Part A: 1pm Friday, 3rd November 2023 (GMT)

Type of hand in: Blackboard/Turnitin

Weighting: 20%

Submission date Part B: 1pm Tuesday, 9th January 2024 (GMT)

Type of hand in: Blackboard/Turnitin

Weighting: 80%

Problem

I need a simple exercise workout planning app. This app must allow a user to see their typical workout week, which could be every day of the week or a subset of the days. The app must allow the user to add, edit or remove weekdays, to a maximum of 7 workout sessions (1 per day only) and a minimum of 1 workout session (on 1 day).

Each workout session must have some way of showing what the focus of the workout is, e.g., “legs day” or “chest and biceps” and the approximate length of the session, e.g., 1 hour.

Each workout session must provide further detail of the exercises to be undertaken during the session. It would be good to see a summary of these listed, with an image¹, the title of the exercise, the number of sets, the number of repetitions per set and the weight in kilograms. It must be possible to add and remove exercises for a specific session.

There must be a separate way to create, edit and remove the possible exercises and view the list of all exercises. These could be shared across workout sessions. Each exercise has a title, image¹, number of sets, number of repetitions, weight in kilograms. There must be a drop-set feature that when activated, will allow for 3 sets of an exercise that are performed in quick succession but where the weight is reduced for each set and where there is a 3-minute break between repetitions of the drop sets. You can assume a 10 second pause between each set.

There is **no** requirement for the app to record a user’s progress, although this might be an extension in the future.

¹ These can be hardcoded rather than require the use of a camera. Using the camera would be considered flair.

There is **no** requirement for the app to support a calendar schedule of workout sessions. The app only needs to deal with the typical week.

There is **no** requirement to have session-specific adjustments to an exercise, e.g., drop-set one day and non-drop-set of the same exercise another day. If the user wants this then they will have to create separate exercises.

Part A: UI Design of the app

For many kinds of app, the market is very crowded, and your app will stand out if it is easier or better to use than the other apps that do the same thing. This means that app user interface design is very important. You need to think through what the best design for your app might be, and prototype it before you build it. You need to hand in evidence that you have done this kind of prototyping.

For this app, when in the gym the user needs to access the exercises for that day quickly, and to see at a glance what is required. Adding, editing and removing session exercises, exercises and workout sessions can be slower operations (more taps and searching, but within reason). Having some nice graphics will make the app more attractive, but the key thing is speed of access.

You may use any tool that you wish to build the prototype. Please don't hand draw. Example approaches include: a set of linked screens in a presentation tool such as PowerPoint, or a set of animated screens in a prototyping tool such as Figma, Adobe XD or FluidUI. Other tools are possible. If appropriate include a URL to the running tool. I have support for viewing Adobe XD prototypes and PowerPoint.

This covers the learning outcome:

“2. Design, implement and evaluate the human user interface of an Android app against manufacturers' guidelines and current best practice.” [1].

Part A submission

By the Part A deadline, you should submit:

- Via **Turnitin**, a PDF report of your UI design that includes screenshots from your prototype, the URL link to the online prototype (if any) and justification for the design, e.g., how does it meet Material Design recommendations, and how your screens are linked.
- Via **Blackboard**, any associated prototype file, e.g., PowerPoint, Adobe XD.

Part B: Construction of your app

Make sure you have a software design using appropriate UML diagrams. Clearly, you will iterate between implementation and design as you proceed. Module workshops slides and materials cover all the aspects of app construction that you will need for the implementation. In particular, you are required to use:

- Jetpack Compose;
- Kotlin;

- Jetpack Architecture components for a local SQLite database. For this app store data locally on the device, i.e., don't use a remote server such as Firebase.

You should aim to build a working app to match your UI and software designs. **That said, you may alter the UI prototype design to match your actual implementation. Discuss such changes in your report.**

This covers the learning outcomes:

“2. Design, implement and evaluate the human user interface of an Android app against manufacturers’ guidelines and current best practice.”

3. Select and apply relevant design patterns in Android app implementation.

4. Build and test mobile applications for Android, using appropriate language features.

5. Build Android apps that function as part of a larger distributed architecture.” [1].

Testing of the app

You should aim for automated testing of your app, e.g., using Compose Testing (Workshop 21). You will need to decide what is to be tested, as well as how it is tested. However, it does not seem to be working on some PCs. Try the example from Workshop 21. If it works for on your PC then use automated testing.

If you cannot achieve automated testing, then make sure that you have a manual test table instead. This choice will not be penalised.

Provide **screenshots** of your app running and of any automated test results.

This covers the learning outcome:

“4. Build and test mobile applications for Android, using appropriate language features.” [1].

Discussing a possible REST API for remote persistence

Assume that you have been asked to create a separate web server that will provide persistence instead of storing the data locally. Provide a 500-word discussion of what a suitable RESTful API for this purpose would look like. Define the API in the following form:

HTTP method URL

A list of significant HTTP headers sent and received.

A maximum of two possible returned status codes.

A statement of the content sent and/or received.

An example would be:

GET /myserver.com/hr/staff/1

Sending headers: Accept: application/json; q=1.0, application/xml; q=0.8, text/*; q=0.1

Received headers: Content-Type: application/json

Content received: the JSON or XML or plaintext representation of the resource.

Status codes: 200 OK, 404 Not Found

If exists, receives JSON file representing the member of staff with ID 1.

Make sure you provide some justification for the API chosen.

This covers the learning outcomes:

“1. Compare, critically evaluate and discuss different options for the implementation of Android apps.

5. Build Android apps that function as part of a larger distributed architecture.” [1].

Part B Submission

By the deadline, you should submit:

- Via **Blackboard**, a zipped copy of your app (hopefully completely functional, but you should hand it in even if it is not functional as not all the marks are for functionality). This must be your Android Studio project and contain all your source code. Try to make sure it builds so that it can be run via Android Studio.
- Via **Turnitin**, a PDF report on the project containing the following sections:
 - a. **Software design section**: A discussion of the software design. This should include appropriate UML diagrams and justification for the software structure. See earlier description.
 - b. **A testing section**: A plan for testing the app, and details on how testing went. See earlier description.
 - c. **Discussion of alternative approaches for data persistence**.
 - d. **Reflection section**: A reflection on how the assignment went - what went well, what might have gone better and why, what you have learned that will help you to do better in future projects, and a summary of how well you think you did the project and what mark you think it deserves. Did you change the UI design during implementation, and if so how and why? **Have a subsection for each**.

Submission details

In submitting your work, you are confirming that you comply with the Department and University statement and policy on Unacceptable Academic Practice. When you submit your report, you will be confirming that you understand the policies and confirm that your work meets those policies.

If you cannot access Turnitin or Blackboard, then you need to email your report and/or code to Chris Loftus (cwl@aber.ac.uk) by the time of the advertised submission. Your email should also explain the problem that has prevented you from uploading to Blackboard. If you send the email after the deadline for submission, it will be treated as a late entry and will be awarded a zero. Please see the Student Handbook [2] for details on how this would be processed.

Breakdown of marks for the project

(next page)

Marks for the project will be assessed on the following basis:

Criterion	Value
Part A	
Quality of the documented UI design (prototype and its description in the report)	20%
Part B	
Quality of the documented software design	5%
Quality of the code approach taken including comments, whether object oriented, best Android practices etc	5%
Fulfilling the functional and non-functional requirements (does it run correctly, is it robust?)	30%
Discussion of alternative implementation choices	10%
Quality of the user interface in the final app	5%
Quality of testing (plan and execution)	5%
Quality and coverage of the report	10%
Flair: going above and beyond expectations, e.g. adding further features or further tests or the quality of the tests. Ask me if you are not sure.	10%

A detailed mark breakdown is provided separately.

References

- [1] Aberystwyth University (2023) "Module Information, CS31620" (Online) <https://www.aber.ac.uk/en/modules/deptcurrent/CS31620/AB1/> (Accessed 3rd October 2023)
- [2] Computer Science Department (2023) "Student Handbook" (Online) <https://impacs-inter.dcs.aber.ac.uk/en/cs-undergraduate/official-information/student-handbooks> (Accessed 3rd October 2023)