

# Assignment 01: Building an Expenses Tracker

Barry Denby

September 20, 2023

## 1 Assignment Information

Course:	MSCC/MSCBD
Stage / Year:	1
Module:	Mobile Development
Semester:	1
Assignment:	1 of 3
Date of Issue:	2023-09-25
Assignment Deadline:	2023-11-05 @ 23:55 (End of week 7)
Assignment Submission:	Upload to Moodle
Assignment Weighting:	10% of Module

## 2 Introduction

**NOTE: read the whole assignment brief first before implementing it contains very important information**

In this assignment you will be tasked with building an Android application using only the core composables provided by the SDK. The application will be an expenses tracker that a user will log their expenses on a day to day basis so they can see where they are spending their money.

A reasonable application of this description should permit a user to enter an income and a number of expenses, and figure out how much money they have left or owe at the end of the month. One assumption made is that the user will earn a salary once a month

The score for your assignment will depend on:

1. How fully features, complete, and robust your code is. Along with how well your UI is thought out (80%)
2. How well documented your code is

**NOTE: This is an individual assignment. It is not a group assignment. You can discuss ideas/algorithms but you cannot share code/documentation.**

### 3 Submission and Penalties

You are required to submit two separate components to the Moodle

- An archive containing your complete Kotlin Android Studio Project. XML based layouts are not permitted. The accepted archive formats are: zip, rar, 7z, tar.gz, tar.bz2, tar.xz. The use of any other archive format will incur a 10% penalty before grading. Your archive should contain your .git directory which contains your git repository **If you do not provide a git repository your code will not be marked.**
- A PDF containing documentation of your code. **If you do not provide documentation your code will not be marked.** Copying and pasting code (directly or by screenshot) into a PDF does not count as documentation.

There are also a few penalties you should be aware of

- Code that fails to compile will incur a 30% penalty before grading. At this stage you have zero excuse to produce non compiling code. I should be able to open your project and be able to compile and run without having to fix syntax errors.
- The use of libraries outside the SDK will incur a 20% penalty before grading. You have all you need in the standard SDK. I shouldn't have to figure out how to install and use an external library to get your app to work
- **An omission of a git repository attached to your email address that is registered for GCD will result in your application and documentation not being graded.**
- The standard late penalties will also apply

You are also required to submit as part of your archive a working Git repository.

- When I unpack your archive there should be a .git directory as part of it.
- This should be a fully working **local** git archive. It should not require access to a remote repository
- You are not permitted to upload your work to Github, Gitlab, or any other publicly visible git repository (assignment will be marked as a zero if it is)
- There must be a minimum of seven commits in the git repository.

**Very Important: Take note of the groups listed below. These are meant to be completed in order. Groups must be completed in full before the next group will be evaluated. Completed will mean that all tasks in the groups are visible and testable. If a single one is not visible and testable further groups will not be considered. e.g. if there are four tasks in Group 1 and task 3 is skipped or not visible or testable then Groups 2, 3 and 4 will be ignored. Documentation**

**will be treated separately irrespective of how many Groups you have completed.**

You should also be aware that I will remove marks for the presence of bugs anywhere in the code and this will incur a deduction of between 1% and 10% depending on the severity. If you have enough of these bugs it is entirely possible that you may not score very many marks overall. I want robust bug free code that also validates all user input to make sure it is sensible in nature. Please be aware of the major bugs section. If any of these bugs are present in your application you will lose 10% for each one up to a maximum of 30%

## 4 Plagiarism

Be aware that we take plagiarism very seriously here. Plagiarism is where you take someone else's work and submit it as if it was your own work. There are many different ways plagiarism can happen. I will list a few here (this is not exhaustive):

- Finding something similar online (full implementation or tutorial) that does the same job and submit that.
- Finding something similar online (full implementation or tutorial) and transcribing (i.e. copying it out by hand)
- Working together on an individual assignment and sharing code together such that all implementation look the same.
- Getting a copy of someone else's code and submitting/transcribing that
- Paying someone to do your assignment. **NOTE: if you are caught participating in either side of such a transaction upto 5 years after you graduate you can be stripped of your degree.**
- Logging into someone elses Moodle account, downloading their assignment and uploading it to your own Moodle account.

I've had to deal with many cases of plagiarism over the last ten years so I can spot it and diagnose it easily, so don't do it. To prevent plagiarism include but not limited to the following:

- Do all your code by yourself
- Don't share your code with anyone, particularly if anyone looks for a copy of your code for reference.
- Don't post your code publicly online. Remember the use of GitHub, Gitlab, BitBucket etc is prohibited.
- If you need to find information online only query about very specific problems you have don't look for a full assignment or howto.
- Change the default password on your Moodle account. The default password can be determined if someone is connected to you through social media or they get one or two details from you.
- If you need to refer to anything online your only permitted source to reference is StackOverflow.
- Please note that AI tools such as OpenAI or ChatGPT will count as plagiarism. There use is strictly prohibited

Be aware that if you submit your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.

Also be aware that if you are caught for plagiarism you will not get another opportunity or a second chance to resubmit the assignment.

If you see the words **“pending review”** in your assignment feedback it is 99% likely that you will be called to a plagiarism meeting.

## 5 Coding Tasks (80%)

- Group 1 tasks (20%)
  1. Create two Kotlin files in your project. MainActivity.kt should contain the activity that your application starts with. SharedComposables.kt should contain all of your functions that are tagged with the “@Composable” annotation. If you need additional Kotlin files you have free choice of name for them, except these two.
  2. Create a single page layout that maps the expenses for a month using a LazyList
  3. Have a display for the income for the month and allow the user to edit the income for the month
  4. Provide a mechanism that triggers an activity or dialog to add a new expense
- Group 2 tasks (40%)
  5. Enable the user to input or change the income for a month
  6. Provide an additional display that shows the surplus/deficit for that month
  7. When an expense is added update the surplus/deficit for the month to take account of the expense
  8. Add another activity where the user can have separate versions of the current interface. One for each month
- Group 3 tasks (60%)
  9. Provide a mechanism to create a new sheet and set the month and year of that sheet
  10. Provide a clickable LazyList that will display the appropriate expenses sheet when it has been clicked
  11. Create an SQLite Database that stores all of your information
  12. Provide the ability to edit or delete individual expenses
- Group 4 tasks (80%)
  13. Whenever a new sheet is created, automatically default the month and year for that sheets to the current date
  14. Whenever a new expense is added provide a date for that expense and default it to the current date
  15. Add in the ability to remove or edit monthly expense sheets.
  16. UI design: well thought out UIs that are easy and intuitive to use.
- Major bugs (presence of any one of these will be a 10% reduction in mark up to a maximum of 30%)
  - Editing or deleting an expense alters the wrong expense
  - Editing or deleting an expense sheet alters the wrong expense sheet
  - Changes made to an expense sheet are not preserved between different runs of the application

## **6 Documentation Brackets (20%)**

NOTE: Documentation should be around 1,500 words in length total

1. (20%): Document every method in your code from a high level perspective.  
i.e. give an overview of what the method does. Do not copy and paste code you will be penalised for this.