

# Mobile App Development - Deliverable 2

Due date: Sunday 12 May, 11:55pm

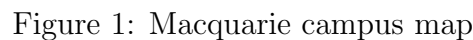
This is the second part of the major work assignment, which is worth 25% of the overall marks for the unit. You are required to build a Minimum Viable Product (MVP) demonstration of the mobile application you have designed. Your MVP should demonstrate the core functionality that your app will provide and be usable to carry out some transaction or activity by the user. Think of it as the demo that you will take to investors to get them to fund the full version of your application.

## 1 Requirements

The core requirements on your application are as follows:

1. Your app must include all or part of this Macquarie campus map image as an element of one page - can be decorative or functional but must appear somewhere.
2. Use Themes and Color Schemes to design your app (including a dark mode support)
3. Some form of persistent storage within your application (user preferences, news stories, coffee order history)
4. Optional bonus: Some interaction with a remote server (Firebase, Google APIs, other web APIs or databases).
5. You will also be graded for good programming practices such as naming conventions, project structure, documentation, etc.

Note that if your app design implements something like placing coffee orders or other actions that would normally require a back-end server, you can **simulate these in your MVP as local transactions instead**. So, coffee orders would be created but not actually sent to a back-end service. If you decide to implement some remote service, you could store the coffee order in the remote database as a proxy for a real coffee shop. Actually ordering coffee is out of scope for this assignment!



- A brief summary of your application, target audience, main functionality
- Comments on how the design of your application differs from your Deliverable 1 design (if it does)
- Notes on the devices you have used to develop and test your application
- any extra information the markers would need to evaluate your work.
- You should try to test your app on at least the Android simulator and the Chrome browser devices. Note any device incompatibilities in the README.md file.

Your project should include tests. Ideally, every screen or major widget should have associated widget tests; every non-widget class (eg. the application state class(es)) should have associated unit tests. At least some of the widget tests should include tests of interactions with the widget. Tests should be run-able via flutter test in the normal way and your submitted code should pass your tests.

You may also include some integration tests of the whole application functionality running on-device, but these are not required.

### 3 Submission

Your submission will be through Github Classroom. Click on the link to accept the assignment, this will create a repository for you. The repository will be empty initially, you should create your Flutter project and push to this repository. Make regular commits and push your updates so that you have a record of the work you have been doing.

Github link: <https://github.com/MQCOMP3130-2024/comp3130-major-work-tinger529>

### 4 Marking

This part of the assignment is worth 25% of the overall marks for the unit. These marks will be assigned based on the following marking guide.

- **Completeness (7 marks):** you present a complete MVP demonstrating the capabilities of your app. A pass grade will be given for a basic set of features that is workable, higher grades for deeper functionality. If you do not meet all three core requirements you will not score well on this criterion. Including interaction with remote servers or implementing more advanced features may attract a bonus if well integrated into your app
- **Design (3 marks):** your app is usable, accessible and well designed.
- **Code Quality (3 marks):** your code is well structured, clear, well documented and uses Flutter capabilities well.
- **Tests (7 marks):** you have included appropriate tests. A pass grade will be given if there is at least one widget test per screen and at least two unit tests. Including integration tests may attract bonus marks if they add value to your test suite
- **Git history and usage (2 marks)** you have a good git commit history, you may have used feature branching, there is evidence that you have self-managed your project well.
- **Documentation (3 marks):** your README.md file and other documentation within your code is complete and helpful to future developers of this codebase.

Note that bonus marks mentioned above might bring your total for that criterion above the stated 7 marks, but that your total for the assignment will be capped at 25 marks.