Overview
This group project is designed for you to demonstrate the skills that you have learned in this course. The final project that you submit in the last week of classes will be a completed mobile application. Non-functional requirements, especially those associated with production-readiness, will be considered extremely important when marking this project. You are expected to work in a group of three to five students when completing this project. Students are not permitted to work alone on the project, as this eliminates one of the learning objectives of this assessment. **Peer feedback forms will be required for all three phases of the project to ensure group equity of work.**

Note: *Any projects from individual students will not be accepted, except if special permission has been given by the instructor in advance.*

Detailed Instructions
Choosing a Topic
The project topic is, for the most part, up to you. Therefore, ensure that you choose a project topic that lets you demonstrate the skills learned in this course. Consideration will be given to projects whose functionality is rather different from sample applications and those developed in assignments in this course. When evaluating your project, I will consider this as requiring extra work. More work done often equates to a higher grade.
It is acceptable if you want to do a project related to industry. If someone you know wants a web application developed, and it lets you demonstrate the skills you've learned in this course, then you can use it for your project (even if you plan to sell that web application when you are finished). Please keep in mind that nothing your prospective buyer says or does will affect the due date or expectations that I have for this project. No matter what happens, this project is due when it is due, my expectations will be based on the content of this course, and I will expect a certain degree of professionalism and production-readiness. Anything outside of the scope of this course will likely not earn you much, in terms of marks. Proceed with caution.

Basic Functional Requirements
It is your job to incorporate as many course concepts into your project as possible. At a minimum, your project must include the following:

- **Dialogs and pickers**
- **Multiple screens and navigation**
- **Snack bars**
- **Notifications**
- **Local storage (SQLite) / Cloud storage (Firestore or other)**
- **HTTP requests**

The actual size of the project (in terms of the number of screens, number of use cases, or amount of code) will differ from group to group. Ultimately, the factor being considered is how much work appears to have gone into the project. **Larger groups will be expected to do proportionally more work.**

If you incorporate concepts outside of this course (e.g. game engines, 3D graphics, sound) you will get credit, but in the subjective part of the evaluation only. **Thus, ensure that you meet the minimum requirements, outlined above, first.**

Also, ensure you do not design your project such that it needs specialized hardware (e.g. robotics) or special access privileges of any kind (e.g. It has a sign in page and I can't create my own account). If your project fails to run correctly when I attempt to run it directly from my Android phone in my office you will lose marks. It should be immediately obvious to a user of your app how it works without needing to read any instructions.

Optional Functional Requirements

Projects will receive more marks for implementing some number of the following:

- Data tables
- Charts
- Maps
- Geolocation
- Geocoding
- Internationalization
- Camera

No individual one of these is specifically required for the project, and these should only be included if they serve a significant purpose for your application's typical use cases.

Game Development Alternative

Students who want to create a mobile game have the option to do so, but this topic won't be covered until near the end of the course. Therefore, it is expected that this option will require significant self-learning to get a head start before the main lectures/examples covering game development. The functional requirements, listed above, will be relaxed quite a bit for groups developing a game, but the expectations are just as high. If you wish to pursue this option, please contact the instructor so that we can work out a set of expectations for the major project.

Evaluation

When evaluating this project, the instructor and the TAs will attempt to give a metric to the amount of work involved, considering several important factors (design, cleanliness of code, code comments, variable/function naming, security checks, error checking, usability/user-friendliness/aesthetic, accessibility, and performance). This metric will be affected by the size of your group (i.e. what will be evaluated is the average work done per group member).

The marking will occur in three phases. The idea behind these phases is that your project should improve over time, so that the final product is comprehensive, professional, and production-ready. It is hoped that your project will make a great portfolio item for when you apply for jobs. The phased marking should also prevent groups from waiting until just a few days before the due date before starting their project.

Proposal

The first phase of marking will take place sometime **during week 6**. This phase will evaluate a written proposal about what you plan to do for your project, including but not limited to the functional requirements and overall project design with UML diagrams (or equivalent). This first marking phase has the purpose of ensuring that your project is reasonable in scope and will incorporate all the necessary components from the course. This evaluation will be worth 10% of your final grade. Unless there is a significant issue of group equity, which should be noted in your README.md and peer feedback forms, all group members will receive the same grade for this evaluation.

Written Report (10 marks total)

Max Score	Requirement
1.00	overview of project
1.00	list of group members and their responsibilites
2.00	list of features (including functional requirements) and how/where they will be used
2.00	code design (e.g. UML)
2.00	mockup of user interface
1.00	writing quality
1.00	scope of project is reasonable for the number of group members

Formative Assessment

The second phase of marking will take place sometime **during week 10**, meaning that you will need to commit the code for your project by the end of week 9. This will evaluate whether or not you have included the topics from the first weeks of the course, which will be mostly objective, but will also include a small subjective mark describing the work done, the user interface design, and the code/design quality. A rubric is listed below, so you can verify in advance which topics will be included. The purpose of the pre-evaluation is to give you an idea of the quality of your project before the end of the term, so that you can make any adjustments necessary to get the grade you want. This evaluation will be worth 15% of your final grade. Unless there is a significant issue of group equity, which should be noted in a document submitted to me, and peer feedback forms, all group members will receive the same grade for this evaluation.

Functional requirements (12 marks total)

Max Score	Requirement
2.00	Multiple screens/navigation
2.00	Dialogs and pickers

Max Score	Requirement
1.00	Notifications
1.00	Snackbars
2.00	Local storage
2.00	Cloud storage
2.00	HTTP Requests

Non-functional requirements (3 marks total)

Max Score	Requirement
1.00	Amount of work done
1.00	User interface design/usability
1.00	Code and design quality

Summative Assessment

The final phase of marking will be carried out at the end of the course. This evaluation will be carried out more strictly, with higher expectations. In addition to your fully developed application, each member will need to submit a ~5 minute demonstration video presentation of the app individually, which includes a listing of the project components you personally worked on. This evaluation will be worth 30% of your final grade, divided into 25% (mobile application, shared by the group) and 5% (individual demo). Students who do not submit any individual demo will receive a 0 for all parts of the summative assessment.

Functional requirements (5 marks total)

Max Score	Requirement
0.50	Multiple screens/navigation
0.50	Dialogs and pickers
0.50	Notifications
0.50	Snackbars
2.00	Local storage / cloud storage
1.00	HTTP Requests

Non-functional requirements (20 marks total)

Max Score	Requirement
3.00	code and design quality

Max Score	Requirement
5.00	user interface design and usability
12.00	amount of work done

Individual Demo (5 marks total)

Max Score	Requirement
2.50	demonstration of application
2.50	explanation and presentation of individual contribution to the project

How to Submit

You will submit your final code on Canvas.

Note: *Any instances of plagiarism will result in the student(s) receiving a mark of zero for the project, and further disciplinary action will be taken. Plagiarism includes, but is not limited to:*

- Copying of (any amount of) work from the Internet, without proper citation
- Submitting a body of work, cited or not, that is primarily not your own work
- Copying of (any amount of) work from another student, past or present, without proper citation
- Allowing your own work to be copied by a fellow student

Getting Help

If you run into difficulty, you may wish to check out some of the following resources:

- https://api.flutter.dev/ - The standard documentation for Flutter, all classes and methods.
- https://dart.dev/tutorials - A series of tutorials for the Dart programming language, focusing entirely on the features of Dart.
- https://flutter.dev/docs/reference/tutorials - A series of tutorials for the Flutter platform, focusing mainly on the widgets and API.
- https://flutter.dev/docs/codelabs - A series of deep-dive, more comprehensive, tutorials to learn more about the Flutter platform.
- https://flutter.dev/docs/cookbook - A set of recipes for commonly used features of Flutter.
- https://github.com/flutter/samples/ - A repository containing some completed Flutter applications.
- http://stackoverflow.com/ - A forum for asking questions about programming. I bet you know this one already!

Of course, you can always ask us for help! However, learning how to find the answers out for yourself is not only more satisfying, but results in greater learning as well.