



College of Engineering, Construction and Living Sciences  
Bachelor of Information Technology  
ID721001: Mobile Application Development  
Level 7, Credits 15  
**Project: Client Application**

## Assessment Overview

In this **individual** assessment, you will develop a mobile application using **React Native** and **Expo**, and publish it to **Google Play Store** or **Apple App Store**. In addition, marks will be allocated for code elegance, documentation and **Git** usage.

## Learning Outcomes

At the successful completion of this course, learners will be able to:

1. Implement and publish complete, non-trivial, industry-standard mobile applications following sound architectural and code-quality standards.
2. Identify relevant use cases for a mobile computing scenario and incorporate them into an effective user experience design.
3. Follow industry standard software engineering practice in the design of mobile applications.

## Assessments

Assessment	Weight	Due Date	Learning Outcomes
Project: Client Application	60%	10-11-2023 (Friday at 4.59 PM)	1, 2, 3
Practical: Skills-Based	20%	22-09-2023 (Friday at 4.59 PM)	1, 2, 3
Presentation: Client Application	20%	10-11-2023 (Friday at 4.59 PM)	2, 3

## Conditions of Assessment

You will complete majority of this assessment during your learner-managed time. However, there will be time during class to discuss the requirements and your progress on this assessment. This assessment will need to be completed by **Friday, 10 November 2023 at 4.59 PM**.

## Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID721001: Mobile Application Development**.

## Authenticity

All parts of your submitted assessment **must** be completely your work. Do your best to complete this assessment without using an **AI generative tool**. You need to demonstrate to the course lecturer that you can meet the learning outcome(s) for this assessment.

However, if you get stuck, you can use an **AI generative tool** to help you get unstuck, permitting you to acknowledge that you have used it. In the assessment's repository **README.md** file, please include what prompt(s) you provided to the **AI generative tool** and how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** and **GitHub**.

Failure to do this may result in a mark of **zero** for this assessment.

## Policy on Submissions, Extensions, Resubmissions and Resits

The school's process concerning submissions, extensions, resubmissions and resits complies with **Otago Polytechnic** policies. Learners can view policies on the **Otago Polytechnic** website located at <https://www.op.ac.nz/about-us/governance-and-management/policies>.

## Submission

You **must** submit all project files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – <https://classroom.github.com/a/yitUo0I6>. Create a **.gitignore** and add the ignored files in this resource - <https://raw.githubusercontent.com/github/gitignore/main/Node.gitignore>. The latest project files in the **master** or **main** branch will be used to mark against the **Functionality** criterion. Please test before you submit. Partial marks **will not** be given for incomplete functionality. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

## Extensions

Familiarise yourself with the assessment due date. If you need an extension, contact the course lecturer before the due date. If you require more than a week's extension, a medical certificate or support letter from your manager may be needed.

## Resubmissions

Learners may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are to be completed within a negotiable short time frame and usually **must** be completed within the timing of the course to which the assessment relates. Resubmissions will be available to learners who have made a genuine attempt at the first assessment opportunity and achieved a **D grade (40-49%)**. The maximum grade awarded for resubmission will be **C-**.

## Resits

Resits and reassessments **are not** applicable in **ID721001: Mobile Application Development**.

## Instructions

You will need to submit a mobile application and documentation that meet the following requirements:

### Functionality - Learning Outcomes 1, 2, 3 (50%)

- The mobile application needs to run without code or file structure modification in **Visual Studio Code**.
- Adhere to the functionality requirements outlined by the client.
- Usable on a variety of mobile devices, i.e., devices with different screen sizes.
- Free of bugs that significantly effect the usability.
- The mobile application is published to **Google Play Store** or **Apple App Store**.
  - To published to **Google Play Store** or **Apple App Store**, you will need an account. The account's credentials will be privately given to you on **Microsoft Teams**. **Do not** disable any applications published on this account.
- Ability to download the mobile application from **Google Play Store** or **Apple App Store** on to a variety of mobile devices.

### Code Elegance - Learning Outcomes 1, 3 (40%)

- A **Node.js .gitignore** file is used.
- If applicable, a **.env** and **.env.example** file is used.
- Appropriate naming of files, variables, functions and components.
- Idiomatic use of control flow, data structures and in-built functions.
- Efficient algorithmic approach.
- Sufficient modularity.
- Each **component** file **must** have a **JSDoc** header comment located immediately before the **import** statements.
- In-line comments where required. It should be for code that needs further explanation.
- Code is formatted.
- No dead or unused code.

### Documentation and Git/GitHub Usage - Learning Outcomes 2, 3 (10%)

- **GitHub** project board to help you organise and prioritise your work.
- Provide the following in your repository **README.md** file:
  - Link to the mobile game on **Google Play Store** or **Apple App Store**.
  - At least five initial functionality requirements.
  - Wireframes of the mobile application's screens. The wireframes can be either hand-drawn or created using a digital tool.
  - How do you setup the environment, i.e., after the repository is cloned?
  - If applicable, known bugs.
- Use of **Markdown**, i.e., headings, bold text, code blocks, etc.

- Correct spelling and grammar.
- Your **Git commit messages** should:
  - Reflect the context of each functional requirement change.
  - Be formatted using an appropriate naming convention style.

### Additional Information

- **Do not** rewrite your **Git** history. It is important that the course lecturer can see how you worked on your assessment over time.
- You need to provide the five initial functionality requirements and wireframes to the course lecturer before you begin development.