



College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
IN721: Mobile Application Development
Level 7, Credits 15
Project

Assessment Overview

In this assessment, you will develop & publish a travelling application using **Kotlin** in **Android Studio** & **Google Play Store**. **Android** features such as **ViewModel**, **LiveData**, **Room Database** & **Google Maps** were formally covered in the teaching sessions. The main purpose of this assessment is not just to build a mobile application, rather to demonstrate your ability to effectively learn intermediate/advanced Android features independently. In addition, marks will be allocated for application robustness, code elegance, documentation & **Git/GitHub** usage.

The travelling application will help you sound like a local abroad & help you adapt to a new culture. You will begin by selecting a continent & country tool. For example, if you were travelling to Spain, you will have text translation & text to speech support, a selection of key Spanish phrases & an interactive quiz to test your knowledge of Spanish culture. A user of your travelling application must be able to select from at least two country tools excluding Antarctica.

Learning Outcomes

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

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

Assessment Table

| Assessment Activity | Weighting | Learning Outcomes | Assessment Grading Scheme | Completion Requirements |
|---------------------|-----------|-------------------|---------------------------|-------------------------|
| Practical | 20% | 2, 3 | CRA | Cumulative |
| Project | 80% | 1, 2, 3 | CRA | Cumulative |

Conditions of Assessment

You will complete this assessment during your learner managed time, however, there will be availability during the teaching sessions to discuss the requirements & your progress of this assessment. This assessment will need to be completed by Tuesday, 2 March 2021 at 5:00 PM.

Pass Criteria

This assessment is criterion-referenced with a cumulative pass mark of 50% over all assessments in IN721: Mobile Application Development.

Authenticity

All parts of your submitted assessment must be completely your work & any references must be cited appropriately including, externally-sourced graphic elements. Provide your references in a README.md file. All media must be royalty free (or legally purchased) for educational use. Failure to do this will result in a mark of zero for this assessment.

Policy on Submissions, Extensions, Resubmissions & Resits

The school's process concerning submissions, extensions, resubmissions & 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>.

Submissions

You must submit all program file(s) via **GitHub Classroom**. Here is the link to the repository you will use for your submission – https://classroom.github.com/a/FWk_XkTA. The latest program files in the **main** branch will be used to run your application. Late submissions will incur a 10% penalty per day, rolling over at 5:00 PM. 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 & 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 & achieved a D grade (40-49%). The maximum grade awarded for resubmission will be C-.

Resits

Resits & reassessments are not applicable in IN721: Mobile Application Development.

Exemplars

A range of application exemplars are available [here](#).

Post-Assessment Evaluation

As part of this assessment, you will be required to organise a meeting with the course lecturer to discuss your feedback. The meeting will be conducted after you receive your assessment result. This post-assessment evaluation will not be graded.

Instructions

For this assessment, you will need to submit an application & documentation that meet the following requirements:

Functionality & Robustness - Learning Outcomes 1, 3, 4 (40%)

- Application must open without file structure modification in **Android Studio**.
- Application must run without code modification on multiple mobile devices. The mobile devices used to test your application will be:
 - Pixel 2 - 5.0"
 - Pixel XL - 5.5"
 - Pixel 3a XL - 6.0"
- Application must run at minimum API 28: Android 9.0 (Pie)
- Text translation support for at least two countries per continent. If a country is multilingual (use of more than one language), choose only one language.
 - Use an asynchronous task & Yandex Translate API to translate text from one language to another. To use the Yandex Translate API, you will need an API key. A key is available in the Microsoft Teams course channel, under the Files tab.
 - Display a progress dialog while text is being translated.
 - Handle incorrectly formatted input fields. For example, a blank or empty edit text widget.
 - **Resource:** <https://tech.yandex.com/translate>
- Text-to-speech support for at least two countries per continent.
 - If a country is not supported, handle correctly with either a snackbar widget or a toast message.
 - **Resource:** <https://developer.android.com/reference/kotlin/android/speech/tts/TextToSpeech>
- Selection of at least four key phrases per country per continent. For example, if you travel to Australia, a key phrase you could hear is "No worries, mate, she'll be right".
- An interactive quiz for at least one country per continent.
 - Quiz topics may include animals, culture, food & drink, geography & sport.
 - Each quiz must have at least five questions.
 - Questions can be multi-choice &/or true or false.
 - Each question must have an image. Multi-choice questions must have four answers.
 - Display appropriate feedback for correct & incorrect answers by either using a text view widget or a toast message. If a question is answered incorrectly, display the correct answer.
 - At the end of each quiz, store the score in a SQLite database table.
 - For each quiz, display the top three scores in either a recycler view or a text view widget.
- Localization support for at least two countries per continent. Localize one activity or fragment only.
 - **Resource:** <https://developer.android.com/guide/topics/resources/localization>
- Application can be exited via a dialog fragment. The dialog fragment must prompt the user when the user double taps the mobile device's back button.
- For each country, display a Google marker on a Google map. The marker's information window must display the name of the country & capital city. The marker's coordinates will be the latitude & longitude of the capital city.

- **Resource:** <https://developers.google.com/maps/documentation/android-sdk/start>
- Switch widget which toggles between light & dark mode.
 - The state (true or false) of the switch widget must be stored in shared preferences.
 - The mode will be based off the state of the switch widget. For example, true equals dark mode & false equals light mode.
 - **Resource:** <https://developer.android.com/guide/topics/ui/look-and-feel/darktheme>
- Splash screen with an image view widget & transition animation.
 - The transition animation must be a custom animation XML file.
 - **Resource:** <https://developer.android.com/guide/topics/resources/animation-resource>
- Adaptive launcher icon which displays a variety of shapes across different mobile devices.
 - **Resources:**
 - * https://developer.android.com/guide/practices/ui_guidelines/icon_design_adaptive
 - * <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>
- Bottom navigation view widget which navigates the user to the appropriate activities. For example, a menu icon for translation support, text-to-speech support, key phrases, interactive quiz, etc.
 - **Resource:** <https://developer.android.com/reference/com/google/android/material/bottomnavigation/BottomNavigationView>
- Display the application's privacy policy in a web view widget.
 - **Resource:** <https://play.google.com/about/developer-content-policy>
- Visually attractive user-interface with a coherent graphical theme & style using Material Design.
- Application is published to Google Play Store.
 - To published to Google Play Store, you will need a Google Play Console account. The account's credentials are available in the Microsoft Teams course channel, under the Files tab. The account will be available to all learners in the course. Do not disable any applications published on the account.
 - When you create the application, name the package appropriately. For example, **op.<username>.travelling**. **Note:** replace **username** with your Otago Polytechnic Ltd username. If you are working in a group, choose one username.
 - **Resources:**
 - * <https://support.google.com/googleplay/android-developer/answer/113469?hl=en>
 - * <https://developer.android.com/studio/publish/app-signing>
- Ability to download the application from Google Play Store on to multiple mobile devices. The mobile devices used to download your application will be:
 - Pixel 2 - 5.0"
 - Pixel XL - 5.5"
 - Pixel 3a XL - 6.0"

Code Elegance - Learning Outcomes 1, 4 (45%)

- Kotlin & XML files contain no magic numbers/strings. Store the values in the appropriate XML files. For example, numbers should be stored in an integer or dimension file & strings should be stored in a string file.
- Use of intermediate variables. No method calls as arguments.
- Idiomatic use of control flow, data structures & other in-built functions.
- Code adheres to various Object-Oriented design principles. For example, DRY & SOLID.
- Efficient algorithmic approach.

Documentation & Git Usage - Learning Outcomes 3, 4 (15%)

- Provide the following in the repository README file:
 - Privacy policy which discloses user information collected by the application. For example, does this application require storage access or internet?
 - Sketched wireframes of the application. The wireframes must be sketched/designed using online software. This must be completed before you start coding.
 - * **Resource:** <https://mockups.com>
 - Step-by-step user guide detailing each activity. The user guide must contain a screenshot of each activity in the application.
 - Commented code is documented using KDoc & generated to **Markdown** using Dokka.
 - * **Resources:**
 - <https://kotlinlang.org/docs/reference/kotlin-doc.html>
 - <https://github.com/Kotlin/dokka>
- At least 10 feature branches excluding the **master** branch.
 - Your branches must be prefix with **feature**, for example, **feature-`<name of functional requirement>`**.
 - For each branch, merge your own pull request to the **master** branch.
- Commit messages must reflect the context of each functional requirement change. Do not rewrite your Git history. It is important that course lecturer can see how you worked on the assessment over time.
 - **Resource:**
 - * <https://www.freecodecamp.org/news/writing-good-commit-messages-a-practical-guide>