College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
IN721: Design and Development of Applications for Mobile Devices
Level 7, Credits 15

Practical 03: User Interfaces & Material Design

Assessment Table

Assessment Activity	Weighting	Learning Outcomes	Assessment Grading Scheme	Completion Requirements
Practicals	25%	1, 3, 4	CRA	Cumulative
Language Translator	20%	1, 3, 4	CRA	Cumulative
Wishlist	25%	1, 3, 4	CRA	Cumulative
Exams 1-5	30%	2, 3, 4	CRA	Cumulative

Conditions of Assessment

This assessment will need to be completed by Friday, 12 June 2020.

Pass Criteria

This assessment is criterion-referenced with a cumulative pass mark of 50%.

Submission Details

You must submit your program files via **GitHub Classroom**. Here is the link to the repository you will be using for your submission – https://classroom.github.com/a/ifyWTPlw. For ease of marking, please submit the marking sheet with your name & student id number via **Microsoft Teams** under the **Assignments** tab.

Authenticity

All parts of your submitted assessment must be completely your work and any references must be cited appropriately.

Policy on Submissions, Extensions, Resubmissions & Resits

The school's process concerning **Submissions, Extensions, Resubmissions and Resits** complies with Otago Polytechnic policies. Students can view policies on the Otago Polytechnic website located at https://www.op.ac.nz/about-us/governance-and-management/policies.

Extensions

Please familiarise yourself with the assessment due dates. If you need an extension, please contact your 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

Students may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are completed within a short time frame (usually no more than 5 working days) and usually must be completed within the timing of the course to which the assessment relates. Resubmissions will be available to students who have made a genuine attempt at the first assessment opportunity. The maximum grade awarded for resubmission will be C-.

Learning Outcomes

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

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

Assessment Overview

In this practical, you will complete a series of tasks covering today's lecture. This practical is worth 1% of the final mark for the Design and Development of Applications for Mobile Devices.

The purpose of today's task is to take an existing application & add additional functionality using the lecture slides & online resources. This will be a frequent occurrence in this course & its important that you achieve good **Android** research skills by the end of this course. When researching, you must consider whether or not the code you are looking at is idiomatic & is it relevant to modern **Android** development practices.

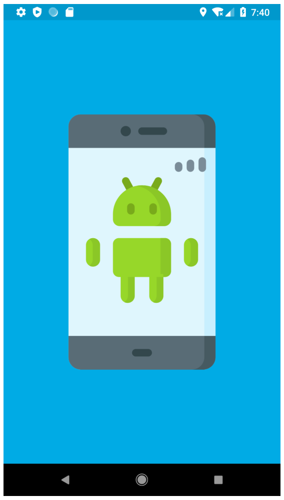
Going forward, the practicals will build on top of each other. In the next few weeks, you will develop a large application that will be the building blocks for **Assessment 01: Language Translator**.

Task 1

- I have provided a **.zip** containing an application you will use in this practical. Open the application up in Android Studio & familarise yourself with the code I have written. Run the application to view the user interface & functionality. In the bottom right hand corner of the screen, you will see three floating action buttons with a **Yelp**, **YouTube** & **Google** vector image. Create three inner classes which implement **View.OnClickListener**. For each inner class, create an **implicit** intent that navigates to their respective website. For example, when you click the **Google** floating action button, you should navigate to **https://www.google.com**.

Task 2

- Using the lecture slide 10, 11 & 12, create a splash screen for the application. My **activity_splash.xml** has a single **ImageView** which displays the vector image **ic_android.xml**. Alternatively, you can choose your own vector image. Resource - https://www.flaticon.com/. Please reference the owner of the vector image at the top of the activity. You don't want to be liable for copyright...

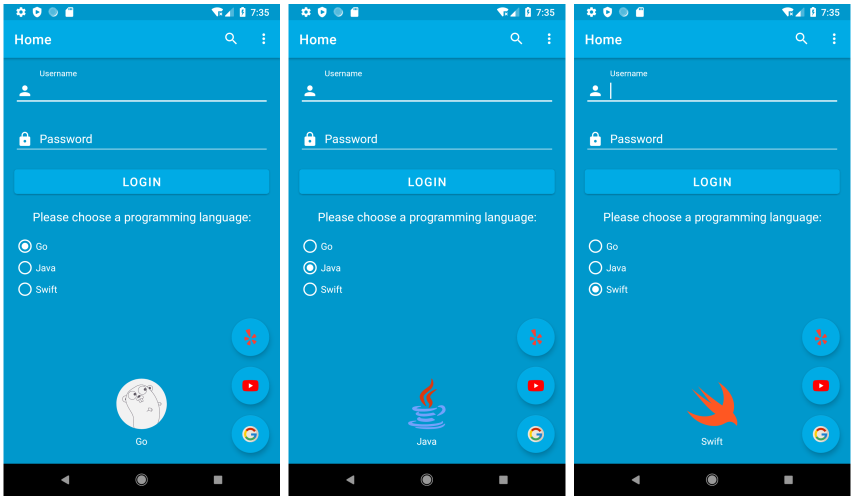


Task 3 - Research

- As mentioned in the assessment overview, it is important that you achieve good **Android** research skills by the end of this course. In this task, you will research how to create a custom toast message programmatically. There are plenty of resources online including code snippets (some may be in **Java**; a simple copy/paste/convert will do the trick) that will help you complete this task. The layout file should have an **ImageView** displaying the programming language vector image & a **TextView** displaying the **RadioButton** text value. If you do get stuck/confused, please ask for help whether that it is your classmate or the lecturer.

Task 4 - Research

- Make sure you have completed task 3 before starting this task. In **MainActivity.kt**, I have provided you with a new event listener called **RadioGroup.OnCheckedChangeListener**. I have bounded this event to a **RadioGroup** widget to an inner class called **RadioButtonOnCheckedChangeListener**. When you check a RadioButton, it should show the custom toast message. I have provided the appropriate vector images for this task. Please check the drawable resource directory.



Home
Username
Password
LOGIN
Please choose a programming language:
● Go
○ Java
○ Swift
Go

Home
Username
Password
LOGIN
Please choose a programming language:
○ Go
● Java
○ Swift
Java

Home
Username
Password
LOGIN
Please choose a programming language:
○ Go
○ Java
● Swift
Swift

Submission

- Create a new branch named 03-checkpoint within your practicals GitHub repository
- Create a new pull request and assign Grayson-Orr to review your submission
- Deadline: Friday, 12 June at 5pm

Note: Please don't merge your own pull request.