

Learning summary report

Ta Quang Tung - 104222196

Overview

This report provides a personal reflection on what I have learned in the unit COS30017 – Software Development for Mobile Devices. It demonstrates that I have achieved all the unit learning outcomes (ULOs) at a **High Distinction** level.

In this unit, I developed a **level 3** custom application to manage smart home devices. The app utilizes Jetpack Compose for the UI and the Room and Preferences DataStore APIs for the data layer. Working with Jetpack Compose allowed my UI to be accessible out of the box, and I have also customized the semantics of my app to better work with TalkBack, an Android accessibility service.

Evidence (in Portfolio Pieces)

I have completed the following assignments (the screenshot of these tasks is on the last two pages):

- All Core outcomes
- All Extension tasks
- A project at Level 1
- An app at Level 2 (including level 1)
- An extended project at Level 3 (including levels 1 and 2)

Learning Summary

- 1. Explain the key differences between the development of systems to run on mobile devices and typical personal computing or internet-based environments, and apply this knowledge in the design of mobile device software.***

A task that demonstrates this learning outcome well is the Mobile vs. Web discussion. In this task, I discussed how Spotify mobile does not let users reorder songs in a playlist, unlike its web version. This is because on mobile, the required drag-and-drop interaction is very awkward and the screen size is limited. These limitations do not exist on PC or the web, so the feature can be implemented easily.

Another task that demonstrates this outcome is Core 1 where I had to design an app whose state must survive when the user rotates the screen. Unlike PC or web-based applications, Android applications do not automatically survive such configuration changes. To achieve this, I used ViewModels to persist state. Using this technique also allowed me to follow Android's separation of concerns principle.

2. *Design effective applications for a mobile device by taking into consideration the underlying hardware-imposed restrictions such as screen size, memory size, and processor capability.*

Core 3 best demonstrates this ULO. In this task, I had to work with RecyclerView to build a performant list with many items. RecyclerView allowed me to reuse Views for new data items, which was more resource-efficient than destroying and creating new Views. This efficiency is important because mobile devices have limited memory and processing power.

Another task that demonstrates this ULO is the Extension – Performance task. In this task, I had to experiment with three different strategies for loading a large number of images and measure their performances with the Profiler. I found that I should not pre-load all images at the start of the application as this will waste memory. I also discussed how we should defer long-running tasks to a separate thread to unblock the UI thread and prevent App Not Responding errors.

3. *Build, test, and debug graphical applications for mobile devices by using the standard libraries that are bundled as part of the developers' toolkit for the mobile device.*

The task that best demonstrates this outcome is my custom app. During the app's development, I had to use many of Android Studio's features such as Logcat (to print and see debug messages), App Inspection (to view the Room database), Device Explorer (to view the filesystem), Layout Inspector (to see how the UI hierarchy as the app is running), and the emulator (to run the app). I also adopted the latest approach to building Android UI with Jetpack Compose and theming with Material UI.

Core 2 also briefly deals with this learning outcome. One of its tasks required me to write simple UI tests using Espresso to make sure my activities were working and exchanging data as expected.

Challenges in Mobile Development

***What are the key areas that need attention during (Android) mobile application development?
You can use a visual model here if it helps.***

The key areas that stick out for me the most are:

- **Limited screen sizes:** Mobile devices don't have as much space as PCs or laptops, which requires developers to lay out their content appropriately. Texts should be placed near the top of the screen (where the user's eyes are), while buttons and input elements should be close to the bottom (where the user's hands are).
- **The possibility that apps will be interrupted:** This can happen when the user receives a phone call, which brings the current activity to the background. Developers need to be aware of the lifecycle states that the app goes through and save state appropriately.
- **The importance of not blocking the UI thread:** Blocking the UI thread for at least 5 seconds results in an ANR error. Long-running processes (such as data fetching or IO operations) should be run in a separate thread.

Elaborate on aspects that you found challenging or different (to expectations) and why.

- Designing the layout for mobile applications was quite challenging for me as I am more used to designing for larger screens. However, this unit introduced me to UI design patterns, which made the process a lot easier because I could quickly apply them without having to do everything from scratch.
- State management is another area I found challenging. Since Android destroys and recreates an activity when the device configuration changes, I had to ensure the UI state was saved and restored correctly. Initially, I used the `onSaveInstanceState` API for this purpose, which was quite cumbersome and limited. However, I soon discovered ViewModels, which took care of this process automatically.

Assumptions and Expectations

I find that mobile development is quite similar to web development. With the Views approach, the UI is described in an XML file while the logic for the UI is controlled by the Kotlin code. This is similar to HTML and JavaScript. With the Jetpack Compose approach, the UI and its logic are specified in Kotlin similar to React.

Activity lifecycles are an area that I didn't expect from Android development. When I started this unit, I thought that an Android application only went through the stages of most typical

applications: start, run, and end. I never expected it to consist of different activities, each of which can go through 6 states: create, start, resume, pause, stop, and destroy. In hindsight, I suppose this division of state makes sense as so many unexpected things can happen when an Android app is running, and handling the resources for that app when it is partially/fully obscured or in the background is important for the device's performance.

Explorations

In this unit, I learned Jetpack Compose to build my custom app. Jetpack Compose is currently Android's recommended way to build UI and comes with Material UI by default. This toolkit allowed me to build a modern and accessible user interface. I also explored the Room database and Preferences DataStore to store data for my custom app. My app followed the two-layer architecture recommended by Android, which consists of a UI layer and a data layer. This architecture made my app more scalable and maintainable.

Two areas that I would like to learn further are user authentication and remote databases. During the mini-conference, I saw that one of my peers had implemented user authentication in their custom app, which made their app usable for multiple users. I would also like to integrate a remote database into my app. Currently, it uses a local database that cannot be shared across devices. Perhaps I will look into Firebase, which is a Google cloud service that offers both of these features. Firebase also has a well-maintained SDK for Android development, which will make the process easier.

Final Words

What is the most useful thing you will take away from this unit?

The most useful thing I have taken away from this unit is the skill to build apps with a layered architecture. During the development of my custom app, I tried my best to follow Android's architectural best practices, one of which involves separating the app into a UI layer and a data layer. This separation of concerns has made my code much more readable and scalable. I believe the same principle can be carried over to other types of development such as web development.


Grades for Quang Tung Ta

 [Print Grades \(javascript:window.print\(\)\)](#)

















Course





2024-HX01-COS30017-Sc 

Arrange By

Assignment Group 

Apply

Name	Due	Submitted	Status	Score
Discussion: Comparing Mobile vs Web Core tasks	Jan 21 by 11:59pm	Jan 21 at 2:46pm		2 / 3 
Quiz: Lifecycle Core tasks	Feb 4 by 11:59pm	Feb 4 at 10:38pm		5 / 5
Assignment: Core 1 Core tasks	Feb 25 by 11:59pm	Feb 7 at 6:12pm		1 / 1   1 
Discussion: User Stories for Real World Apps Core tasks	Feb 25 by 11:59pm	Feb 23 at 10:49pm		2 / 3 
Take 2: Comparing Mobile vs Web Core tasks	Mar 8 by 11:59pm		missing	- / 3 
Take 2: User Stories for Real World Apps Core tasks	Mar 18 by 11:59pm		missing	- / 3 
Assignment: Core 2 Core tasks	Mar 24 by 11:59pm	Mar 25 at 5:34pm	late	1 / 1   1 
Assignment: Core 3 Core tasks	Mar 24 by 11:59pm	Mar 26 at 11:30am	late	1 / 1   1 
Assignment: Extension on mobile UX patterns Extension tasks	Mar 24 by 11:59pm	Mar 24 at 9:10pm		1 / 1   1 

Name	Due	Submitted	Status	Score	
Assignment: Extension on Performance Extension tasks	Mar 31 by 11:59pm	Mar 26 at 9:54pm		1 / 1	
Mini-conference video Custom project	Mar 31 by 11:59pm	Apr 3 at 2:53pm	late	1 / 1	
Quiz: persistent data and concurrency. Core tasks	Mar 31 by 11:59pm	Mar 31 at 8:44pm		5.5 / 6	
Weekly progress Custom project	Mar 31 by 11:59pm	Apr 3 at 4:31pm	late	5 / 5	  4 
Interview Custom project	Apr 9 by 9am	Apr 8 at 10:44pm		∅ / 1	
Learning Summary Report Core tasks	Apr 10 by 5pm			∅ / 1	
Final report Custom project	Apr 10 by 7pm			∅ / 30	
Portfolio Marking Portfolio (All items listed below makes up your portfolio)				∅ / 100	
Portfolio (All items listed below makes up your portfolio)				∅	
Core tasks				∅	
Extension tasks				∅	
Custom project				∅	

∅ Your instructor has not posted the grade. While your instructor has not posted the grade, grade and comment information is unavailable. 