COS30017

Software Development for Mobile Devices

Word Count: 1183

Huu Nhan Le
104171133

Acknowledgement

No Generative AI tools were used for this task

Introduction

In this reflection, I will try to summarize what I have learned in this course and justify how I should receive the Credit grade for my efforts. More specifically, I will go over how I deal with the learning materials and assignments in this unit, what I have done to achieve all 3 ULOs, the challenging aspects as well as what I want to learn beyond this course, and finally the key takeaways from the unit.

Reflection

How did you approach this unit?

Firstly, I tried to pay attention during the lectures from the tutor. However, at home I frequently needed to look at the lectures again to further grasp the complicated concepts in mobile development. After that, I usually did the labs in the Canvas as they are the building blocks for the first 2 assignments in this course. In the assignments, similar to most courses at Swinburne, I often self-learn some outside aspects of Android development, like changing language inside app and RatingBar Espresso test for assignment 2.

How have you met each of the three ULOs? Include relevant examples of key work

1. **ULO1 (Explain the key differences between development of systems to run on mobile devices and on typical personal computing or internet-based environments, and apply this knowledge in the design of mobile device software.)**

A mobile device is often more limited in CPU power, RAM memory, battery life, and screen size compared to traditional PC and laptop, requiring the developers to follow the guidelines and convention to make the user experience the best. For this ULO, the best example is the implementation of SQLite in assignment 3 with the Room library. Because the sync process between the device and remote service can be very long and frustrating for the user, it is recommended that you should use a light service such as SQLite for local database. In my password manager app for assignment 3, I implemented all 4 CRUD operations to manage the list of passwords stored locally with view model, repository, adapter classes, indicated by the architecture of the app in figure 1.

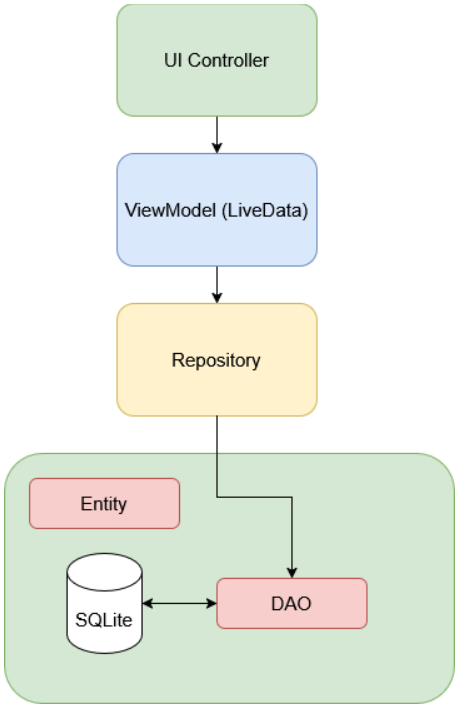

Figure 1: Architecture Design of Assignment 3 App

2. **ULO2 (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)**

This ULO is best illustrated through my UI designing process in the assignments. As the Android device's screen size is much smaller than a laptop, you would want to use all parts of the screen and communicate as much to the users as possible. However, it is also important to not go overboard as the app then would be unfriendly for users. In assignment 2, when I was required to design an app to rent musical instrument, in MainActivity, I tried to make the instrument's image as large as possible while also displaying other details like name, price, weight, type, as shown in the figure . The navigation buttons is placed at the bottom for convenience with the user's fingers.

Figure 2: Portrait Layout of MainActivity in Assignment 2

3. **ULO3 (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**)

I have used Espresso tests to test the UI/UX for all 3 assignments in this unit even though it is only required in assignment 2. For instance, in assignment 1, I designed 2 Espresso test with one for the scenario the user click the Climb button once and one for the scenario the user click the Climb button 9 times. Or in assignment 2, when I am forced to design a more complex test for an app with multiple activities, the BookingActivityTest1 stimulates the following list of actions: unselect the Saxophone chip, click the Prev button twice then the Borrow button, click the Language Toggle button, enter the email and drag the sliders.

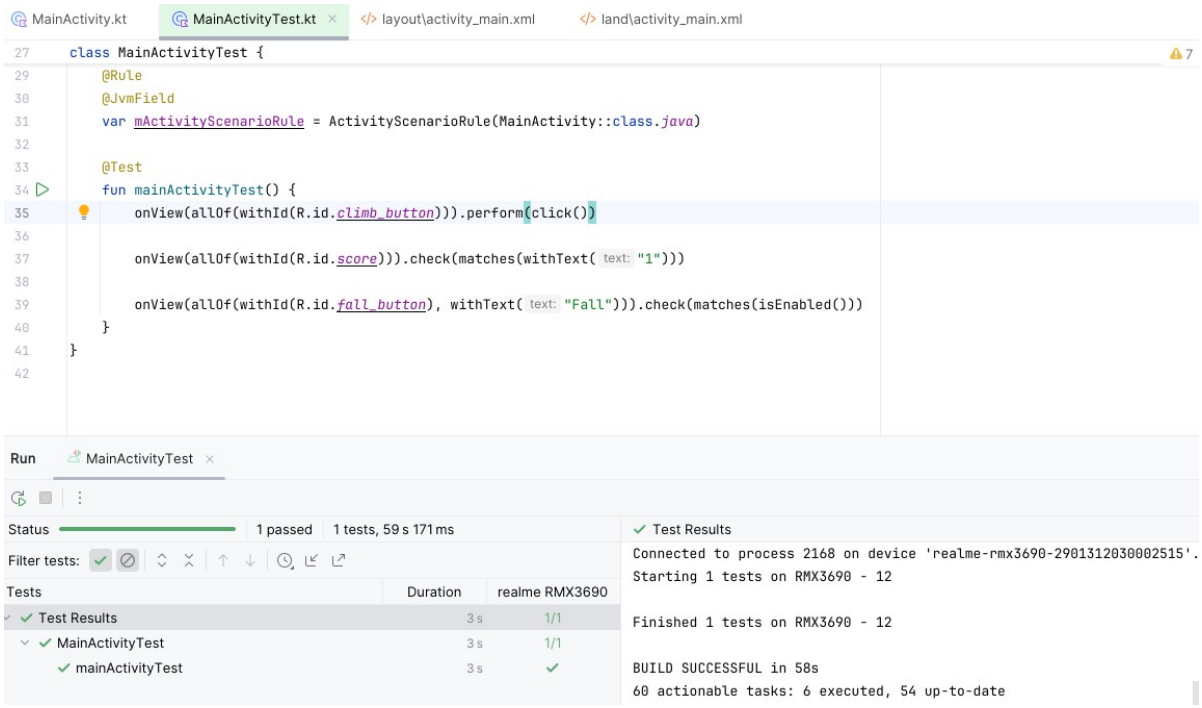

Figure 3: MainActivityTest Espresso Test in Assignment 1

MainActivity.kt | MainActivityTest.kt | layout\activity_main.xml | land\activity_main.xml

```
class MainActivityTest {
    @Rule
    @JvmField
    var mActivityScenarioRule = ActivityScenarioRule(MainActivity::class.java)

    @Test
    fun mainActivityTest() {
        onView(allOf(withId(R.id.climb_button))).perform(click())

        onView(allOf(withId(R.id.score))).check(matches(withText( text: "1")))

        onView(allOf(withId(R.id.fall_button), withText( text: "Fall"))).check(matches(isEnabled()))
    }
}
```

Run | MainActivityTest

Status: 1 passed | 1 tests, 59 s 171 ms

Filter tests:

Tests	Duration	realme RMX3690
Test Results	3 s	1/1
MainActivityTest	3 s	1/1
mainActivityTest	3 s	✓

Test Results

```
Connected to process 2168 on device 'realme-rmx3690-2901312030002515'.
Starting 1 tests on RMX3690 - 12

Finished 1 tests on RMX3690 - 12

BUILD SUCCESSFUL in 58s
60 actionable tasks: 6 executed, 54 up-to-date
```

```
fun bookingActivityTest1() {
    //Part 1, first unselect the Saxophone chip, click the Prev button twice, then the Borrow butt
     Booking Activity
    onView(allOf(withId(R.id.chipSaxophone))).perform(click())

    repeat(2) {
        onView(allOf(withId(R.id.btnPrev))).perform(click())
    }

    onView(allOf(withId(R.id.btnBorrow))).perform(click())

    onView(allOf(withId(R.id.instrumentName))).check(matches(withText("Yamaha FG800 Acoustic")))

    onView(allOf(withId(R.id.labelRentalNumber))).check(matches(withText("Number: 1")))

    onView(allOf(withId(R.id.labelRentalTime))).check(matches(withText("Rental Time (Month): 1")))

    onView(allOf(withId(R.id.rentalPrice))).check(matches(withText("Total Price $220.75")))

    onView(allOf(withId(R.id.creditAccount))).check(matches(withText("4000.0")))

    onView(allOf(withId(R.id.appName))).check(matches(withText("Instrument Rental")))

    onView(allOf(withId(R.id.btnConfirm))).check(matches(withText("Confirm")))

    onView(allOf(withId(R.id.btnCancel))).check(matches(withText("Cancel")))
    //End of part 1
```

Figure 4: Part of the BookingActivityTest1 Espresso Test in Assignment 2

What did you find challenging or different (to expectations) about mobile development and why? Use examples from this and other units.

1. **Usage of Kotlin**

Even though Kotlin is somewhat based on Java, it is still a new language for me to learn. One of the primary differences is the val and the var keyword. I needed to pay attention when I should use the var keyword when the variable can be changed during the operation of the app. Like for the Instrument data class in assignment 2, I only used the var keyword for the stockNumber properties as it would obviously change when the user rent something.

```
data class Instrument(
    val name: String,
    val type: InstrumentType,
    val price: Float,
    var stockNumber: Int, //var to enable it being reduced when borrowed
    val weight: Float, // Weight in kg
    var rating: Float
) : Parcelable {
    constructor(parcel: Parcel) : this(
        parcel.readString() ?: "",  // For name string, if null return empty string
        InstrumentType.valueOf(parcel.readString() ?: InstrumentType.GUITAR.name),
        parcel.readFloat(),  //?: 1.0f,  // For price float, if null return 1.0f
        parcel.readInt(),  //?: 0,  // Read stockNumber integer, if null return 0
        parcel.readFloat(), // Read weight
        parcel.readFloat(),  //?: 1.0f    // Read rating float, if null returns 1.0f
    )

    override fun writeToParcel(parcel: Parcel, flags: Int) {
        parcel.writeString(name)
        parcel.writeString(type.name)
        parcel.writeFloat(price)
        parcel.writeInt(stockNumber)
        parcel.writeFloat(weight)
        parcel.writeFloat(rating)
    }

    override fun describeContents(): Int = 0
```

Figure 5: Instrument data class in Assignment 2

2. **Android Emulator**

Ironically, I can't use the emulator to test the app during this course at all as it just consumes too much memory. Luckily, I had an Android device to run the app but it still kinda slow and obviously can't be tested on other resolution or memory size. The development, debugging, and testing processes all far longer than any other courses at Swinburne that I have learnt so far, so I had to dedicate more time to do the assignments.

What have you explored (or now want to explore) beyond the unit (in the context of the ULOs)?

1. **Jetpack Compose**

This is probably my number one priority if I pursue mobile app development in the future. In this course, I just learned to use XML to design the layout although this is an outdated technology at this moment. Nearly all the companies have transitioned to Jetpack Compose as it more up-to-dated and support more features, so a new developer should learn about it, matching with the third ULO to use standard libraries to build, test and debug graphical applications

2. **Custom functions in Espresso test**

Throughout all 3 assignments, I had to use Espresso test extensively to test the UI/UX of the apps. However, the materials in the course just gave some basic instructions to use Espresso test, while there are any elements not supported in Espresso like RatingBar and Sliders. Therefore, in the spirit of ULO3, I researched on the internet a bit to design some custom functions especially for the purpose of testing some elements in assignment 2 and 3.

What is your key takeaway from this unit?

- Use Android Studio effectively to develop mobile applications
- Learning Kotlin-specific features to use
- Use Espresso test to test UI/UX
- Implement ViewModel and data binding
- Use intents for multi-activity apps
- Use RecyclerView
- Implement Room library to use SQLite database in the app

Conclusion

Overall, this unit has been a valuable learning experience that has significantly strengthened my understanding of mobile development. Through engaging with the lectures, labs, and assignments, I have built practical skills in Kotlin, Android Studio, and testing tools like Espresso. Despite facing challenges with the emulator and adapting to new concepts, I remained committed to learning and improving. I believe I have demonstrated a clear grasp of the unit learning outcomes and applied them meaningfully in my work. This reflection highlights not just what I have achieved, but also my eagerness to explore further, especially in areas like Jetpack Compose and advanced testing techniques.