Lappeenrannan teknillinen yliopisto

School of Business and Management

Software Development Skills

Ville Virtanen, 000616559

LEARNING DIARY, MOBILE DEVELOPMENT MODULE

**LEARNING DIARY**

18.9.2021  
  
I started the course by reading through the general course information and goals for the course. I also checked the mandatory assignments and the instructions how to complete the course. The main focus of the course is to provide tools for creating unique projects (to give and advantage in the job market) and to find my passion as a software developer. As a software development major (in South-Eastern Finland University of Applied Sciences – Xamk) I think I have already found my passion as a software developer, but I chose this course to specifically get myself familiar with mobile app development, which I think will be a major focus in future computing. Also, there isn’t any course in my syllabus to cover this specific area of software development.

I already had a sufficient development environment installed (including Git, VS Code and IntelliJ Idea - which has the same functionality as Android Studio, which it is based upon). I have not used Git very much in the past, so I took upon myself to get more familiar with it. I also used Github before, so I made a Bitbucket account for this course and installed Sourcetree, which seems like a good free Git GUI app (too bad it isn’t yet optimized for the M1 Macs that I’m using, but regardless of that it seemed to work fine). I started by watching the Git introduction video but stopped it after a while since I found that the instructor didn’t provide sufficient background information about repositories and their function (and local vs. remote repos etc.). I therefore decided to read a couple of Git introduction articles first after which I resumed to watch the video.

Also, the video focuses very much on the commands although I think using a GUI client will be easier and more user-friendly way to manage the project. I would have liked if the video instructor would have used some visual graphs and explained the idea of working with git more than just telling what commands to feed and not actually telling what they mean and do. The instructor also suggests that with GUI clients you will not learn using Git and the inner workings of it, but I would say that by copy-pasting commands to a terminal you will not learn any more than by using a GUI tool.

This was how far I got the first day. I had some trouble with Git and understanding the concepts of it (the video just assumes I already know what a repository is and what is a staging area and what the commands fed to terminal actually do and so forth). Inspired by this difficulty I enrolled in a Git course to University of Tampere.

19.9.2021

The next day - after getting the hang on basics of Git - I made a repository in Bitbucket and cloned it to my local computer. After that I used IntelliJ Idea to make the first Android project in the repository. At this point I had doubts on whether I first should have made the project and after that I should have used git init to initialize it and git push to push it to the remote repository (Bitbucket). That is hoe the introduction video makes out the process to be like, but I think it works better if the repository is made and synced before adding files to the repository.

I started the project by watching the first Android instruction video. I tried to follow it along and make the same project as I watched the video. I did have to pause the video very often to be able to do the same stuff as the phase of the video was quite fast. Luckily, I have some experience in Java programming language (and had Java and Java SDK installed) so it wasn’t completely new stuff for me.

I decided to try and manage this course with IntelliJ Idea instead of Android Studio, since it should have exactly the same functions and layout etc. I guess we will see how that goes after I get to do more developing with it.

I selected to make a phone/tablet app and chose the SDK version to be version 26 (to support Android versions 8 and up). I did also choose to use the Kotlin language for this app, since that is the recommended language by Google and the video is several years old and not up to date with that. We will again see, if this causes any problems, but I doubt it since the syntax and workings of Java and Kotlin should be very similar to each other. I also had some troubles following along with the video instructions, since the Android Studio layout seems a little different from the video version. So not all functionality was the same or in the same places.

As a result of the discrepancies between the video and my IntelliJ Idea installation I decided to try out Android Studio after all. Since my project was in its very early stages I decided to make a new project from Android Studio to have a clean slate with it. This time I also chose earlier API 21 (to support Android 5.0 and up). This was the default, so I guess it is also recommended if the very newest features are not needed.

So doing a clean install I got the app build eventually working (after having to download and install Android SDK 30 too (Android Studio installed SDK 31 by default at installation). That way I was able to get the app build to actually run through without errors. The next problem though was with Android Emulator, which didn’t run the app after the build was successful. I tried to install a nev AVD (Android Virtual Device) with Android API 30 (Android 11) installed and see if that was the problem, since the build was done with the same API version too. That seemed to work since on the new virtual device I did get the app to run.

As I was watching the first video, I was trying to use the Java syntax in my app (which I decided to write in Kotlin). Not a single line of code worked as is so I think it will be a very difficult to follow through the three instruction videos. Nevertheless, I think it is better to use new and Google backed up technologies than learning a new skill already with outdated instructions. I will continue to try and watch the videos, but I will reference the developer.android.com instructions as to how to get started with Kotlin and Android development.

I finished the first video and I think I understood all the concepts that it showed. The problem that arose was that the Java code shown in the video didn’t work in my setup (as I was using Kotlin). So, the references to the Views needed to be changed and the whole code section would need to be written in Kotlin. I did learn more about Views and ViewGroups from the Android Developer documents (I would have thought that such an introductory video would have used some time to introduce those basic components)

20.9.2021

I went through the Android App development basic lessons on developer.android.com. Those helped me further understand the layout of Android Studio and have an idea of what different panels are for. I also learned about resources that are imported into the project (like pictures for example) and some error messages that were present after the first instructional video too. Like hardcoded text error/warning message that says you shouldn’t hardcode text into the app but use a string resource to store that string. That way it can be reused and translated, and it can be better used in different orientations and so on.

The official instructions also did a better job at introducing the attributes panel, so that I actually now know what that is and how to use it. I also learned about the component tree and order in that panel, which works like layers in so that bottom most are the top layers and the topmost are at the bottom. That seemed a little counterintuitive as most image manipulation softwares (like Photoshop) use a different order in layers.

I also learned a lot about the Kotlin language from the Android documents. It basically covered the fundamentals of variables, classes, functions and datatypes. Also, some methods were introduced (like random() and repeat() methods). Next, I was going to learn more about the Activities and interactivity in Android Studio and how to reference them in Kotlin. This was about where the first introduction video ended up without explaining anything about activities really (I really didn’t like the introduction videos as you may have noticed from this diary). I hope the official documents will be clearer and more explanatory on the use of these as well. I managed to get the input from the EditText View and change it to string and after that to on integer after reading through a couple of online guides. It was quite hard, because the type wasn’t a string to begin with so it couldn’t be parsed to integer directly. Also the syntax to get the text input has changed recently from .GetText() to just .text which is simpler, but most of the online instructions are older.

21.9.2021

The second video was about changing activities (or jumping to a different app) from one activity to another. Activity is explained as “a rectangular are that displays something” in the video.

I also learned how to pass information from one activity to another one. This is done with the .putExtra method of the intent. That information is passed in as key-value pair (the video didn’t specify if multiple pairs can be passed in as arguments though, but I gathered from the internet that this needs to be done in another putExtra() line. The information is then available through intent.getStringExtra(key) in the another activity. Or .getIntExtra if the passed information is and integer and so forth.

The second video also showed how to launch an activity that is outside of our own app. An example would be launching Chrome browser from your app and going to [www.google.com](http://www.google.com). I got both Intents working as showed in the video (but of course in Kotlin), but I’m not really sure what the code does exactly. As I think that is the main thing to learn I will continue further with the official documentation to see if there will be more about Intents and Activities than in the video. I’m sure it will come eventually, but if it is not explained pretty early, I will skip that and use the working code in the course project even though I have no idea why/how it actually works. Seems that moving from activity to another comes in Unit 3: Navigation of the official Android Basics in Kotlin course. I have so far done the Unit 1 only, but the Unit 2 looks to be shorter than the first one. I intend to do these 5 units that are available during this course and I think that should cover all the basics that are required to complete this course and build the project app.

24.9.2021

The next lesson on the Android Developer course was about XML layouts. I have some background from web development, so the basic idea was familiar (as it is pretty much the same as HTML). New stuff was about the namespaces that are used in the XML files. The xmlns:android for example denotes that the attribute is defined by the Android system. Namespaces are a little unfamiliar for me, but I think I understood enough to get an idea how they are used.

I also learned how to use binding instead of the findViewById methos to get the views from the layout/xml of the app. This seemed like a simple way to use views, but it took some prework to get it working, so for small apps I’m not sure if it is better to just use the “old” method. There was a small section about method chaining in the lesson, which inspired me to read more about it, since that has been at this point one thing that I haven’t really understood. I mean I know to use multiple methods in a row if the code is given to me (copy/paste) but I haven’t really figured out what it does and how it works. That was my next foray into stackoverflow and some blog texts.

25.9.2021

Today I started learning about theming android apps. I learned how to use the material.io site to choose complementing colors and how to change the color values in the app (primary and secondary and their variant colors too). Also, I tested the use of dark/night themes which seemed to work fine too. Most important thing to note here was that the choice of colors should be pleasing, and they should provide enough contrast so that text remains legible in every situation.

Also related to theming was a lesson about changing the app icon and choosing one that provides a coherent user experience on different devices (as manufacturers can have different icon shapes for their launchers). Android provides adaptive icons that can have different shapes and your icon should look good on all of them.

I also checked out the material components lesson, which gave the basics on ready made components for Android Apps. They should be the preferred way to make Android Apps, since that provides same look and feel for the whole operating system and all apps. That way users will learn to use your app more quickly and it provides them a cohesive user experience even when using different apps.

After learning about theming an app the next lesson was about lists. This was interesting since they are the basic data storing types in many programming languages and as so an integral part of learning a new technology or software development area (like Android development). This was already familiar stuff for me, but the notations and syntax changes from one language to another so it was a good thing to learn this stuff in Kotlin too.

27.9.2021

The third video of the course material considered mainly ListView elements (and it assumed knowledge of the previously taught functionality – such as moving between activities). First thing I learned from the video was that you can use string-arrays in the strings.xml file to provide the app with arrays of strings that can be used in ListView (they probably can be used in other elements too).

The second new thing was using adapters and the LayoutInflater to fill views with information. This was again explained not very thoroughly, but fortunately there were a little more information about adapters in the Android Developer course that I also have been going through the last days. Again, I found that reading the available material (in Android Developer site and especially the course material there) was a better way to learn how this stuff actually works than watching a video (which to my mind was too fast to follow and understand and it didn’t explain the logic behind using some of the available methods in android).

I think I have also settled on the project app I want to make for the course project task. It will be a helper/learning app for kindergartens to help dressing up children in the right order and with the selected clothing. My girlfriend works as a kindergarten teacher and that kind of app has been something she has been wanting to have there (insread of using printed out clothing cards). The app should work so that it presents different clothes and tells the user to choose the ones he/she wants to put on. After that the user presses a button, which takes him to another screen that shows the selected clothes in a predefined order (to show in which order they should be put on). After putting an item on that clothing item is pressed and the item is marked as put on (with color or some other method). When all clothing items have been put on, the user presses the DONE-button which again takes the user to a new screen that shows a thumbs up picture and an opportunity to start the process over.

I also just double-checked the course completion instructions and noticed that the material from the exercise projects should have been uploaded to the Git repository that is returned. Since I have not been doing the same apps as in the video instructions (but mostly the Android Developer Course projects/apps) I do not have those files to submit to the repo. I will however submit the other apps (that I think are way more polished and complex anyway) instead. I had been using one repo per app as is the usual way of using git repos, but I will upload all of them to the one single repo (under the Coursework folder). I hope that those sufficiently show my learning together with this diary. It is also not advised to use git repositories inside another git repositories, so I’m not entirely sure how this will affect the repos working. I did try to fix this by deleting the .git directory in each of the repos and then using git rm --cached path/to/folder to delete the repos and start using just the top level repo that contains all of the course material. I’m not sure if this worked as planned, but I will try to test this later before submitting the whole work.

28.9.2021

Found a decent way to autofit cards in gridlayout from stackoverflow <https://stackoverflow.com/questions/33575731/gridlayoutmanager-how-to-auto-fit-columns> As this app will be used on tablet and phone it would be nice to have the layout autofit the right amount of columns. This also helps with landscape use as it calculates the width and autofits a different number of columns when in landscape and when in portrait mode.

I got the basic functionality working of the project app. I had a problem with the ClothingAdapter RecyclerView recycling the checked cards as base to other items that weren’t checked. This was fixed by giving every item a new property of checked (that was false at the beginning) and drawing the card checked status based on that in the RecyclerView. Then I just added to the setOnClickListener a line that changes the items checked status: clothItem.checked = !clothItem.checked.

Next problem I had was the second activity (DressingActivity) crashing when moved to it from the start screen. I couldn’t find a reason for that, so I had to use logcat to debug the crash. I found that the recyclerView binding wasn’t initialized (after using the: private lateinit var recyclerView: RecyclerView). I had it initialized in the onCreate function as val recyclerView = binding.recyclerView which I gather was making a new variable rather than using the old one and it was fixed by removing the val keyword.

After this all I pretty much have left to do is implement and onClick on the DressingActivity to check that the item has been put on and implement a “Valmis” button to show next screen with a thumbs up symbol. I would have liked to make a flip animation to the cards when they have been put on, but I’m not sure if that is a little too much work and fixing the alredy written code. We will see, what I can find online to have feedback to the card press on that screen too (I wouldn’t like to use the same as previous screen).

30.9.2021

I decided to implement a drag and drop interface to the project app. That was completely new thing to do and it took quite a bit of googling. I eventually found a decently simple instruction that I could implement in my code. The code is from here: <https://yfujiki.medium.com/drag-and-reorder-recyclerview-items-in-a-user-friendly-manner-1282335141e9> and it took just a little bit of maneuvering to get working on my app. I struggled a little bit with the list order not sticking, but I eventually figured that it rearranged itself every time the layout was changed (or the state of the app reloaded). That was fixed by moving the reorder function from the DressingAdapter to the ClothingAdapter (now the list orders itself every time a new item is added to it). In the DressingActivity screen it is now possible to rearrange items and that new order doesn’t get overridden if layout is changed.