Lappeenrannan teknillinen yliopisto
School of Engineering Sciences

Sofware Development Skills

Eric Liski, 000452179

LEARNING DIARY, MOBILE MODULE

LEARNING DIARY

6.7.2024

Today I will be starting my third anytime-course about Mobile Software Development. I have already completed the two previous self-study anytime-courses about Full-stack and Front-end development, and I have really enjoyed the style and the freedom of these anytime-courses which allow me to study at any time when I feel like I can do it to my best ability. I also feel like I have learned a lot from these anytime-courses about software development and different technologies used in it. Today I'm excited to get to know deeper about the mobile development aspect of software development. I have previously completed the Java course called "Olio-ohjelmointi" back in first year of my studies but since I have forgotten a lot of it, and I haven't even used Java much since either. This course will be great way to remember and relearn Java as well as Android Studio. From reading the general information section of the Moodle page it seems that this course works exactly like the other two previous courses I have completed which is no surprise. I would say I will use Visual Studio Code as my code editor for this course but for the most part I think the development will happen in Android Studio in this course. But without wasting more time, I will proceed to Android Mobile Development section in Moodle and start watching the recommended video series about Android Studio. I will now be watching the first part of the Module task list "Introduction". First small issue I ran into during the first part of the series is that I couldn't find the Preferences menu in Android Studio to enable automatic imports and optimize imports on the fly. After little Googling, I found official Android Development documentation (Configure Android Studio | Android Developers) and was able find out that it can be enabled in the File -> Settings -> General -> Auto Import tab. Another problem I soon ran into was that my Android Emulator wouldn't launch and had bad performance. I read some Stackoverflow threads about errors I got (Android Emulator closed because of an internal error: GPU Found - Stack Overflow) and other issues I had with the emulator (The system ui isn't responding in android emulator (Flutter) - Stack Overflow) and eventually were finally able to make it work for now by following some of the comments on these threads. I also were able to optimize the emulator a bit more after watching this video about the issue Optimizing Android Studio Emulator Speed - 5 Proven Strategies 2024 🔥 -

YouTube in which the thing that helped me the most was going into File -> Settings -> Tools -> Emulator and turning off the “launch in a tool window”. Turning this setting off made the emulator open in its separate window instead of opening in the Android Studio tool window which I believe helped the performance of the emulator a bit.

7.7.2024

Today I continued the first part of the series and finished it. Thankfully this time around I didn’t run into really any issues. The first part of the series helped me remember how Android Studio worked such as simple things like creating new Text Views, Buttons, etc. As well as utilizing them in the Main Activity Java code like referring to them and getting the elements and interacting with them through the code. Something completely new I learned is debugging in Android Studio. I don’t remember ever utilizing the given debugging tools in the Android Studio. These tools seem quite handy and useful later down the road. I learned some basic debugging in Android Studio like running the program step by step to see where and at what point the errors/issues occurred. After the first episode I feel more confident in my coding abilities in Java as well as my Android Studio usage.

8.7.2024

Today I needed to upload my progress on the series to GitHub. I didn’t know how to effectively upload Android Studio Projects to GitHub, so I decided to find and watch a video about this topic. I found a great video about uploading projects to GitHub from Android Studio by Practical Coding on YouTube (How to upload an Android Studio project to GitHub | 2023 - YouTube). I followed the tutorial in which I also learned more about using GitHub such as creating GitHub access tokens which you can use to login. I will be going now onto the second part of the series and hopefully not run into any major issues. I learned about four core elements of Android Development which are Activity, Intent, IntentService, and BoadcastReceivers. I also learned about how to effectively use constraints on views as well as how to align them together properly. Later on, I learned about how to create new

activities inside of a project. I then ran into an issue that I had quite a problem solving. The issue was that when I was trying to use function ‘getApplicationContext()’ and ‘startActivity()’ it would hit me with an error about ‘Cannot resolve method ‘getApplicationContext()’ and ‘Cannot resolve method startActivity()’. I tried to start a completely new project and the issues still occurred. I spent some time trying to find a fix for the problem but ended up giving up and decided to continue the next day.

9.7.2024

Today I will continue my journey on fixing this issue. I ended up spending most of the day googling and reading Stackoverflow threads about similar issues and tried multiple different suggested fixes from which none worked for me. Some of the Stackoverflow threads: android - Cannot resolve method getApplicationContext() - Stack Overflow and android - Cannot resolve method startActivity() - Stack Overflow. I also looked up multiple YouTube videos about intent and moving between activities, but nothing helped me out in the end. Then I decided to watch the second part of the series completely again and created a new project in which I decided to tick a box that wasn’t even in the tutorial that said ‘Use legacy android.support libraries’ and followed the tutorial again completely. This time around however it was finally fixed, and I was able to use the methods that previously wouldn’t work for me. I believe what fixed it was the tick in the box about legacy android.support libraries. Well thankfully now I can finally continue after spending yesterday evening and most of this day finding a fix. I’m now done with the second part of the series. I learned about how intent works, and you can transfer information from an activity to another activity. I also learned how to launch an activity outside of the main application. Overall, I had no problems with the rest of the video, and it quite smooth sailing since I fixed the first issue in the beginning of the video.

10.7.2024

Today I pushed the second application “Quicklauncher”, which I finished yesterday. Next up I’m planning on watching the third part of the series which hopefully goes smoothly unlike the last two parts of the series. This time I will turn on the legacy android.support libraries to hopefully avoid any problems in advance. I’m only 20 minutes in and I have already learned so much new. For example, I have learned about values in Android Studio and how to utilize them in code for an easier way to change values everywhere. Values in Android Studio are incredibly useful for managing resources like strings, colours, and dimensions. By defining these in XML files under the res/values directory, I can easily reference and modify them across the entire application. This not only makes the code cleaner but also significantly reduces the risk of errors. For instance, instead of hardcoding a colour value multiple times throughout the project, I can define it once in colors.xml and refer to it by its resource ID. This way, if I need to change the colour, I only need to update it in one place. I've also learned how to create and utilize layouts effectively. Understanding the layout XML files and the different types of layouts available in Android Studio, such as LinearLayout, RelativeLayout, and ConstraintLayout, has been invaluable. By leveraging these layouts, I can design user interfaces that are both functional and aesthetically pleasing. One completely minor issue I ran into is when I was creating a Java class it wouldn’t let me select the ‘superclass’ like displayed in the video but I easily fixed that by manually importing it later by using ‘import android.widget.BaseAdapter;’ and changing the public class to extend the BaseAdapter by typing ‘public class ItemAdapter extends BaseAdapter’. Then I could implement methods like in the video and continued forward as expected.

11.7.2024

Today I will be continuing the third part of the series where I left off previously. I instantly ran into an issue. When I implemented the new layout, I ran into an issue when trying to run the application. I get an error message: ‘ERROR: C:\Users\Eric\AndroidStudioProjects\ListApp\app\src\main\res\layout\my_listview_detail.xml:27: AAPT: error: '50' is incompatible with attribute layout_marginTop (attr) dimension’. I’m now trying to solve this issue. This was quite an easy issue to fix I just simply forgot to add the units ‘dp’ when I was creating the new layout for my list view then

after trying to run the application again I got an error: ‘FAILURE TO REDEFINE Unable to perform redefinition of 'Lcom/example/listapp/MainActivity;': Field changed from 'items' (sig: [Ljava/lang/String;) to 'descriptions' (sig: [Ljava/lang/String;)!’ but then after I restarted the application and reinstalled it now everything works as expected. I’m now done with the third and the last video of this course. Towards the end of the video, I learned about Image views and how to display images in the application and how they can be utilized. I also learned about Bitmap factory and for example how it can used to scale images according to the device screen size, so the image looks clean and doesn’t break the application.

12.7.2024

Today I pushed the ListApp to GitHub. I will also start my project for this course. For project I just decided to do an application that contains multiple views. One of the views is a to-do view, where like the name suggests you create to-dos and mark them off when you do them. Another view is an Animal list where there are different animals, and you can click to view them. Then I also have the food list view which was created in the part 3 of the coursework.