

Lappeenranta Teknillinen Yliopisto
School of Business and Management

Software Development Skills

Victor Kathan Sarker, 000373591

LEARNING DIARY, MOBILE MODULE

LEARNING DIARY

27.05.2021

Finally, I have enrolled myself in Moodle page of the 'Software Development Skills: Mobile' with the course key received earlier. I checked the course information immediately to get an understanding of how the course will run. After reading the general information, the first thing to do was to setup the environment for the course. I downloaded Android Studio version 4.2.1 and Git and installed, and I had the Visual Studio Code editor pre-installed as it is my favorite text editor for programming due to its flexible features and ease of coding, i.e., auto-completion. I have some basic idea about the concept, but have not used version control software such as Git or BitBucket regularly since I have been keeping locally managed personal changelog for individual programming projects in other languages. So, I was interested to see it in more details and wanted to refresh my previous knowledge. I started watching the video tutorial and I set up the Git environment following their instructions on git-scm.com. Then I spent some time to get myself familiar with the basic git commands. As I was developing on a Windows® PC, there was a GUI version available for the Git. However, for the setup and configuration, I used the Git bash command line interface. The provided video was really precise and everything went smooth with setting up Git on my computer and performing the initial configuration. In addition, I created a quick reference file of useful Git commands for future. Time to start working with Android Studio now!

I started watching the introductory video in part one on Android Studio. According to the prerequisites stated in the video, I installed Java Runtime Environment (JRE) and Java Development Kit (JDK). I familiarized myself with the layout of Android Studio, particularly different menu items and development environment sections. Turning on auto-importing of libraries was a good tip from the video as I have faced this issue in some other programming languages where it usually took some time to find what library was needed to be imported in order to use a particular function. I followed the first Android application development step by step and soon got an app running in the emulator. The application is a simple adder where it takes two numbers and when the user presses the button 'Add', it shows the result. However, the initial *gradle* build process inside the Android Studio took some time, but the subsequent compile process and start-up of the application was

comparably fast. The application runs well as expected and I just made my first application commit to GitHub! Then I set a breakpoint in the editor and used debug menu for checking out what is happening under the hood when the button is clicked. The new version of Android Studio has a different layout and button arrangement than the one shown in the lecture video, but I figured it out quickly. Particularly, the keyboard shortcuts such as *Shift+F9*, *F7*, *F8* key combinations during debugging were very useful.

29.05.2021

Today I watched the video from part two. The video started with the concept on core elements of Android application development. It discussed about *Activity*, *Intent*, *IntentService* and *BroadcastReceivers*. Essentially, an *Activity* is any rectangular area which display something on the screen of the phone. A typical Android application usually has multiple activity. In addition, the activity layout defines how different elements such as buttons and text are arranged and viewed. An *Intent* is an action the device should try to perform which is requested either by a user or another internal or external process or application on the phone. Linking to that, an *IntentService* is a service which can receive a request for an *Intent* and can handle the associated tasks defined in the request. Similarly, the *BroadcastReceivers* receives an *Intent* often implying that some work or task had completed. I think the last one is useful in updating some status or notifying about something which might be useful to the user or the application, for example, a notification when the internet connection is lost or is back again.

The exercise in this video was to create an application which demonstrates the implementation of *Intent* and *Activity*. Specifically, I learned about how to work with the methods *onCreate()* and *findViewById()*. In addition, I got to write code involving methods *getIntent()*, *putExtra()*, *getExtras().getString()* and *startActivity()*. I followed along the video and created an application with two buttons, one when clicked takes to a second screen (i.e., a second Activity), and another which attempts to open a preset web address (in this case, <http://www.google.com>) outside the application. However, unlike shown in the video the *getPackageManager()* function did not work. I started debugging using the console by utilizing *Log.d(<key>, <value>)* method and logging value of the return from the function. I found that the function returns *null* value if everything is alright, i.e., there is an application

to open the requested URL. So, I changed my code a bit and then it worked as like as the outcome in the lecture video. This was a fruitful debugging experience indeed! However, one particular mater I have noticed while working on Android Studio is that sometimes it can show properly and correctly written code in red and complain about having an error, probably due to background linking process or similar. This can also occur after changing and deleting some major parts of the code and probably Android Studio has not resolved the layout files with the activity class files. In such cases, it is good to just clean the project, save, close the project and reopen to get rid of those apparent errors. Also, if the Android device emulator gets stuck while debugging, it is useful to perform a cold boot of the emulator from Android Virtual Device (AVD) manager.

02.06.2021

I just watched the lecture video of part three and followed along and created an application mostly focusing on *ListView* for displaying content as a normal list and a customized one. In addition, the video also expanded the example of using lists for making use of *ImageView* module to display images. It was interesting to be able to view the image particularly when individual items on the list was pressed. This cleared the concept of using *Intent* and switching to a different activity screen, which was introduced in the video of the second part. Also, use of adapters to populate the *ListView* was quite interesting. Specially, the customized *ListView* using own layout was something good to learn because in real life applications, data is most often shown in an application specific way rather than a simple list of items. One very simple but troubling particular issue took me some time to figure out why the project build was failing without showing any error in the code editor is that I had put a single apostrophe (') in one of the items inside *string-array* element in *strings.xml* resource file. I found that although this did not show any error, I needed to use an escape character to get rid of the build failure.

03.06.2021

With some more knowledge after watching all the videos, I started to think about the project. Since the project requirements are quite minimal and open ended, I decided to work on an application for children. I named it as 'LearnerKid'. The application description is provided in the *ReadMe.md* file in *GitHub* so I am just including it here (in blue) for convenience.

Upon launching the application, the application shows the menu in which 'Alphabet', 'Five Fruits' and 'Five Vegetables' are shown. A user can select one of the options and the application shows the further details of that option. For example, the alphabet option shows the English alphabet in full, the vowels and the consonants for easy learning. The 'Mathematics' option opens up a calculator-like page in which a user can enter two numbers to perform four different operations (addition, subtraction, multiplication, division) just by pressing the buttons. A 'divide by zero' check is also included to prevent the application from unintentional crashing. The 'Five Fruits' option displays a list of five common fruits in a customized list view. In addition, it also shows a small description and an image of individual fruits. Similarly, for the 'Five Vegetables' option provides a list of five commonly used vegetables along with its short description and a preview image. And lastly, the 'Take a Test' option opens up a test for the user to verify their knowledge. The test consists of 10 questions which are shown one by one. After answering all of the questions, the score along with number of correct answers are shown. To pass the test, a minimum of 60% score is required.

09.06.2021

The goal of the project was to fulfil all the requirements and to utilize the knowledge gained to integrate new modules and different method of using the same modules shown in the lecture videos. For example, I have incorporated customized *ListView* in the main activity, buttons, full screen view, linear view, intents and passing of extra information from one activity to another. In addition, for showing the fruits and vegetables options in my project, I have used the same layout and my custom submenu adapter. Doing so saved me writing another list adapter and reuse the same layout, however, with different content, e.g., once with fruits and another time with vegetables. I have also added a 'divide by zero' preventive division calculation just in case if the second number input (the divisor) from the user is zero. Finally, I had to think a bit how to include the test questions and answers along with the test report in the application. I wrote a class for the question parameters, namely '*TestQuestion*'. Presenting a single question was easy, however, showing all 10 questions using the same activity one by one was little tricky. But I thought of a way so that I can use the same layout and just change the content until there are no questions left to answer. At the end, I included a test report activity so as to inform the user about his or her test score. The user has to obtain at least 60 percent score so as to pass. The application shows the

scores and related congratulatory message in deep violet color when a user earns a passing score. In contrast, if the user fails, it shows the score in red. I started with a simple activity for showing the test report. But as there is the navigation button at the bottom, it was not a good User Interface design. Additionally, the user could go back and change the answers. For this reason, I have modified the code and switched to a full screen activity while hiding the navigation controls so that after the test is complete, it is not possible to go back to the test questions page and alter the score. Instead I modified the full screen activity template from Android Studio's built-in templates for always hiding the navigation controls and going back to the main activity page of the application after the test report page is viewed. This way, the user can revise the materials and take the test as many times as he/she wants, but cannot change the answers while taking an individual test. Finally, I cleaned up the project code a bit and added comments in most places and it is now ready to submit along with a video I have recorded from the Android Virtual Device Emulator.

Overall, the course was a good learning experience and a solid starting point from which I can continue to explore application development in Android Studio. I am happy that now I can have applications on my phone developed by me! A big thanks goes to the organizers, teachers and staff of the course.