

Lappeenrannan teknillinen yliopisto  
School of Business and Management

Software Development Skills

**Joona Niemenmaa, 001823383**

**LEARNING DIARY, MOBILE MODULE**

# LEARNING DIARY

24.11.2024

I started out by reading through the course page and getting an understanding of what the course is about.

I'm already familiar with git usage and Android development with Android studio from other university courses and my own experience.

Thus I will be using Android studio as my IDE and GitHub to host my git repository. As for my language I will be using Java.

I've already decided on the project I want to make which will be a simple mail user agent. Essentially a mail user agent is an application that can receive, read, compose and send electronic mail over the internet.

I'll later on detail a proper design document to detail what the project should be capable of to limit the scope to a reasonable level.

For now I'll focus on completing all the exercise tasks first.

The first thing to learn was how to set up an Android studio project. The setup process was basically the same as it was in the provided video, although the empty activity template used in the video is nowadays only implemented in Kotlin, so I had to use instead the empty views activity template instead as it provided a Java option as well.

I followed along and learned what an Android studio project is composed of and how the general workflow happens. A lot of this was already familiar to me from an OOP course I had completed in the previous semester. I learned about how the user interface is built and how you can reference UI components in your code to make things happen. I followed along to the video and built the example app.

Lastly something new which I haven't used before: A debugger. You can add breakpoints to your code in the editor, and afterwards you can begin running your app in debug mode. Once the app reaches your specified breakpoint in code it stops and allows you to move forward step by step and the IDE also gives you insight into the state of your program. This allows you to find where your program is going wrong more easily and thus fix the bug.

25.11.2024

Started work on the second video today.

A recap of what I learned today:

Android apps made using Android studio are composed of Activities. An activity has an user interface and some logic attached to it and essentially it's the unit of what the user will be interacting with.

The layout for an Activity is defined in a .xml file and the logic is defined in a class that inherits the Activity class. Code for an activity is called using different methods such as onCreate() or onPause() or onDestroy() which are called when the app enters different states.

An Intent is basically a request for the Android system to do something. The most common use case is to change the current activity to another one. Also when using an intent to move from one activity to another you can pass in data to it with putExtra().

An IntentService is a class whose responsibility is to handle intent requests. It can process them on a separate thread to not freeze the user interface.

BroadcastReceiver receives a broadcast that an intent process is complete.

29.11.2024

Started work on the third video.

It becomes apparent to me that these videos are a tad outdated at this point.

The listview component showcased on the video is a legacy component that's now been replaced by the recyclerview.

So instead of learning how to use the listview I'm just going to learn the recyclerview component instead.

Spent the day reading documentation from [here](#) on how to implement a recyclerview. Apparently it's more performant than a listview so it's probably better I learnt it instead.

Learned how to create a layout for a recyclerview viewholder and an adapter for how to apply it to an existing recyclerview.

I still watched the video as it did have some good insights such as how you can store variables in .xml files in the res folder or how to work with images.

Decided to spend some more time reading the documentation on Android app fundamentals found [here](#).

30.11.2024

Today I started work on the actual project for the course. I decided to break development into two phases.

First I will implement a console application just using Java that's capable of sending / reading emails. The idea is that I can make the networking code work first before implementing it in an Android environment.

The second phase will be building out the actual application in Android Studio. If I implement the first phase correctly most of the internal logic should already be in place, so it will be a case of using Android's tools to make the UI and display information in a pleasing way.

To start things off I need to have an email server my application can talk to. After doing a bit of poking around the internet I landed on a combination of dovecot and postfix. Both of them are available on Linux which is my working desktop environment. I'll need to make a guide on how to get an email server up and running on a local machine when I return this project.

The server will be running locally since I don't want security to be a large concern. Once I start writing the actual code I'll make proper specifications for it but it's good to mention here that it won't be very good information security. The goal is to be able to read and implement RFC specifications for both SMTP and IMAP.

After a lengthy session of tinkering and reading documentation I managed to install both dovecot and postfix and have them running locally on my machine.

15.12.2024

Hello again,

A progress report on what I've been working on.

I started properly working on my project. My current idea is to build an email client capable of acting as an SMTP and IMAP client to locally running mail server. Having a local mail server means I don't have to worry about DNS or security concerns, which considering my current abilities is a reasonable constraint.

I made a design document which essentially is just a hodgepot of ideas on a document that roughly identify what the project should look like. It includes a list of tasks I need to get done, overview of the project, some specifications and notes on SMTP.

I've already mostly read through the [SMTP specifications](#) and have made a working implementation of the SMTP client to be used in the Android app.

Next up I should read about MIME which defines how email messages should be constructed and then after writing some unit tests I'll have the SMTP side fully functional.

Then I'll just have to implement the IMAP client and then I can start work on the user interface for the application in Android studio.

24.09.2025

Decided to add in a final entry here, looks like I kinda forgot to keep making these. Hopefully that isn't too big of a problem.

As I am writing I am taking the course a second time hence the much later date. I forgot to finish and submit the project in time (mostly due to a very busy third period) which is why I had to take the course again.

Now I'm just cleaning up the github repository and preparing the project for submission.

One thing of note is that I'm pretty sure the project contains a few race conditions for the Receiver class. I wrote the code before taking an OS class so I had no idea that could be a problem, but I'm still returning the project as I don't want to spend too much time figuring out code I wrote last year when I have other courses now to worry about.