candidate: 1043
PGR208 - Android exam 2022

# Android exam report

In the application presented for this exam, we were tasked with creating a dietary tracking app. The app is supposed to create a fit for most people solution to tracking the calorie intake of normal everyday people as well as giving them a big selection of different recipes and ingredients to look into.
Based on the pitch that was given in the exam handout there are quite a few features here. Firstly the use of an API that has a ton of recipes and ingredients to choose from. So that needs to be implemented in a way that makes sense to the user what kind of recipes they want.
Secondly a tracking system for calories. When the user logs onto the app for the first time, they should first make their way over to the settings menu to set their daily intake. We will use this number to calculate from the recipes how much they plan on eating. In this settings menu as well there should be settings for meal preferences, dietary options, and amounts.
Thirdly the solution is to have a search history where all of the results that have been shown to the user is gathered up and shown. The user can then from there see the name of the recipe and search it up once again to see it again. The user interface should be simple and not too in your face and be thought of more as a tool than a fun toy.

The solutions as of the end of this exam are like so. There are 3 activities that are the backbone of this application.

## MainActivity

This is the first activity the user sees and is where they will likely spend most of their time. The application runs the API based on the button the user presses. Two buttons are shown either based on preference or on the time of day. Based on preference uses the shared preference values and base the API query on that. Sending a specialized link to get specialized recommendations.
The recyclerview which dominates most of the screen is filled with different recipes and up to 20 of them. The user can click on either the text or the picture and be sent to a share via a link that takes them to the recipe for that meal. Should they wish to eat it and have that a hit on their calorie intake? Then they can only hit select and a toast will show up notifying the user that it has been subtracted from the daily calorie intake. If a recipe should exceed the daily calorie intake then there will be a subtle but noticeable change to the text color on the calorie number under the title. This is to signify that if you choose to have this one you will be

exceeding your calorie intake for the day and it will go into the negative numbers. This calorie intake is not reset either unless the user does so.
The items shown in the recyclerview deserve a greater mention. The items that populate the view all have certain values in them, which have first been set in an adapter. Users will not be able to see the url for sharing as this happens automatically when a picture is clicked, but the object has it stored for any other use it can have. Images are created via bitmaps with a function from the main activity that takes the image url and converts it into a bitmap that the image view has been set to.

## SearchHistoryActivity

This activity is the simplest one of the three. In this one we have a simple text view which is updated from a file on device that reads to it. This file is populated every time a api request is sent and the file is updated each time. If this would scale on a larger platform is unlikely. But for this solution it works for now. More on that later.

## SettingsActivity

This activity has a lot of things going on. Thanks to the use of shared preferences all of the settings are stored using this technique. And are all set when the activity is loaded. These settings are different values that can be changed by the user and will be saved and sent off with a toast when they hit the save button. To make sure the user is not confused if the settings are changed or not, the main activity is ended so if the user is to get back to the main activity they have to restart the app or save the settings. This is to combat and save conflicts and problems that could arise from this.

On this project, I was the only group member, due to my intense schedule and wanting to challenge myself. I did everything from planning to designing to implementing everything. So what would I have done alternatively?
I understand now that maybe having a group for a project of this capacity would have helped. This project was way bigger than I could have ever thought, but I pulled through. With the capacity, I as an individual have there are some shortcomings to this solution. First of all the storage solutions. For the settings menu, the solution of using shared preference was a good choice. It would not scale up to more users but changing over to a bigger solution when user handling is something that is in the picture it would make more sense then.

However, the storage of recipes is not what I would have wanted it to be. Despite my countless hours of trying to create an SQLite database using Room, I could not get it to work. I settled for a file-writing type of storage to make it work but it is by no means a perfect solution or even one I would expect to have at this point. But it works.
I am a little confused about the favorite system since this feature is something that doesn't really show up outside of it is an object that I store to not show anywhere. A better solution would have maybe been to have a favorite list as well as a history list, but with the resources I have, it would not have been possible within my abilities.

In conclusion, I would not say this is a magnum opus of android development but still a very good try from a single person. Stories of individual developers making amazing applications are something I can see that is not something that just happens. But it happens because of hard work and a focus on doing the best one can. I enjoyed quite a few parts of this android development assignment and hope that I may cross paths with this platform again. Maybe I'll make something even better that time.