Simone Bergamini 160300967 Mobile App Development Essay CSC3122

At the beginning of this project I had never written an android app nor a piece of code with a user interface excluding the simple java browser that I wrote in Stage 1 of the course. This said it was quite a challenge to get the application done and many things had to be learnt as much of the code for applications is unlike the console type code I was used to writing. Also, my choice of writing the application in Kotlin gave me the opportunity to also learn a new and useful language which I now think is superior to Java for the development of mobile applications.

As my application would need to initially display a list of categories, with more being created by the user my first task was to achieve a way to display this ever-growing list. Initially it displayed only an array of strings and eventually I changed it to an array of the category class. Doing this I learnt how to use the Recycle Viewer component.

For the Recycle Viewer and for the rest of the user interface I had to learn the xml language which is used to organize components and the general layout of the application. The recycle viewer had me create a template which every item in the category list would use to display itself. I also added a delete button so that the user could delete a category he no longer needed.

The next step was to set on click listeners to both the delete button and the whole item in the Recycle Viewer item. This was done by creating a controller class that would use an interface to accomplish this.

Eventually I needed to redirect to another activity when the user clicked on any item on the recycle viewer list or execute a method if the delete button was clicked. I had to learn how to pass data to the next activity and eventually save the changed made in the next activity back to the original activity and list. Initially I had attempted to make the category class parcelable to pass the instance to the next activity, but this gave me the problem that whatever change was made in the next activity would not be saved if the user went back to the main activity or closed the app, it was only a temporary change. This led me to experiment with global variables and I created a class exclusively to store the list of categories. When going to the category activity from the main one, the position of the specific category is passed but all the changes are done to the class with the global list.

To create new categories and add new timers I had to learn how to create dialogues that would check the user input to check for invalid inputs. This had me experimenting with positive buttons and negative buttons to represent different actions and input checking. I learnt that I could make red error texts with exclamation marks on the field that had erroneous inputs.

The actual timer activity was also very challenging to code. For this activity I first had to learn how to use the countdown timer class to create a simple timer that started straight away. The biggest challenge was implementing the pause and stop buttons and their functionality. For this to work I created an enum class for the timer status to know the current state in which the timer is at any point. I also had to create a new class to store values for the timer. Like this I was able to save the current time remaining to the external class if the timer were to be paused and create a new timer with those values if the start button was played again. The timer also always needed to initial value saved in case the stop button was clicked then those initial values would have to be retrieved.

For the timer finished activity I had to learn and experiment with how sounds work in android and how to play sounds until a button is clicked. Initially I had accomplished this by playing a custom file with the method, but I discovered I could play the set ringtone on the android settings by using the ringtone alarm method.

Finally, my last challenge was to store the data created when the app ran into the memory disk of the phone so that the information could be retained when the app was closed and reopened. In order to this I had to learn how to use the shared preferences and how to save a list of a custom class to them. For this I used the gson class and the toJson() method.

I think the way I approached this task was very efficient as I slowly built up the code by achieving very simple tasks to work my way upward to the final product rather than starting to write extremely complicated code. This allowed me to research very simple tasks and how to do them online and still be able to execute them in the context of my application and to add all the features that I wanted to each part. One thing I would have done differently is that once I learnt a technology or how to implement a certain feature, I would implement every single feature that require those skills. Rather than implementing just one and moving to a different kind of feature and returning to the previously learnt one afterwards. This would have made my coding more efficient as the feature I was implementing would have been fresh in my mind rather than having to come back to it weeks later to add a similar feature but not remembering exactly how to do it.

To conclude I must state that this project and module in general has been the one I have enjoyed the most from the entire the course. I enjoyed the project to the extent that I will be continuing to work on my application and other application I have in mind throughout the summer as app development has become my favorite type of coding.