I was tasked by Sarah, my close family friend with creating an app that compares users ages with famous figures from history. Sarah envisioned that this app is going to help her students learn, while having fun in the process which to me is a great way of creating a lasting impact on upcoming generations. Hence, I took on the task and created my app called “The History App”.

My app boasts 2 comprehensive features. Firstly, allowing users of the app to input their age and see which famous figure in history they are the same age as, and each famous figure presented to students when they match with somebody has a certain fact about them that is taught to the student. So not only does app provide a curios experience making students wonder who they match with in history, but it also gives them the opportunity to learn something new. My app has15 famous figures from history which also allows students to play a guessing game to see how many ages they can match with and how many facts they can learn each time they guess a new figure from history.

The UI (User Interface) of my app is very simple with a gradient-coloured background for the simple reason that there aren’t many apps that use that colour and the ones that do are very eye-catching for example the Instagram logo. My app provides a space for users to enter their age, or any age and it prompts them to enter their age. It has 2 buttons; one to clear any inputs that have already been entered and another one to generate or find what historical figure the user’s age matches with. Under these two buttons is a white box that shows the results of the matching age. The box tells the user who they match with, when and at what age they died, and what they were famous or widely known for. The header or top section of my app has many clocks symbolising connotations of time and history, with a large header in bold white, saying “The History App”.

The easy part of building the app for the purpose of matching peoples ages with those of famous historical figures was building the UI of my app. The hard part was coding the Kotlin of my app. It was a little bit simpler because I have had lots of practice in the past from my lecturer on firstly, understanding the problem and then laying out steps to figuring out what needs to be done to produce a specific outcome. While coding I started with declaring my variables and all the elements of the app that were to be utilized by the user such as the: space to enter an age, the button to clear any inputs, the button to display what figure my user matches with, and the area or text that is used to display the output of the match that has been made. Within my code, I have included comments such as the following:

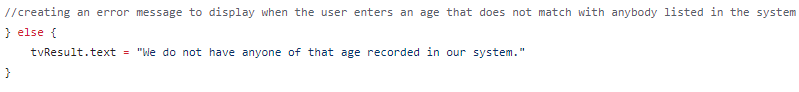






that detail each step of what I am doing or coding the app to do when certain things happen. And another part of my app that I had to include was how it was going to handle errors.

My app has a range of what ages can and cannot be entered. I used the appropriate code to tell my system that users cannot enter an age under 20 or an age above 100. If a user did enter any of those ages, I coded my app to tell the user that they have entered an age outside of the coded range and to prompt them to enter an age within the range of 20 to 100. My app displays separate messages for when a user enters an age below 20 and an age above 100. Also in my app, I have inbuilt or coded 15 specific people and their ages so my app is not like Google where it will search for people that are the same age as the user but if there is a match, my app displays the necessary data, and if there isn’t anybody listed in my code that matches with a specific users age then my system displays the following message:



Something I found interesting about building this app is the many ways in which you can code. I mostly used ‘if else’ statements in my code but while I was coding, I realised that even while statements could be used and when statements as well. Since I decided to go the simple route and not confuse myself, I used ‘if’ statements, and there were a lot of if statements. I had to use if statements to create the age range and I used if statements to list all the famous figures from history for my app. I used a ton of ‘if else’ statements, but I like the style of coding I used because it is easy and simple for me to understand which will also showcase to the marker, my level of understanding when it comes to coding.

When all my code had been written and my UI had been finished. I ran my app and tested to see if it worked the way I wanted it to, and it did, so then it was time to push my code into my repository in GitHub. I decided to search for a YouTube video explaining how to push code or a Kotlin project from Android Studio into my repository in GitHub (I have linked the video at the end of this report). After I successfully pushed my code into GitHub, came the part I actually struggled with the most. In the assignment outline of what we were supposed to do during the assignment, I was prompted to use GitHub Actions to test my app and boy did I struggle. I used a link to a GitHub blog (that I will also link at the end of this report) to set up a workflow in my repository to use GitHub actions to test my app; keep in mind that my code is fine, and it runs perfectly in Blue Stacks. I followed the exact steps listed in the blog, I copied a piece of code from the blog and did code attribution in the workflow to ensure that I do not get a 0 for not referencing properly, and then I committed the changes and waited for the workflow to finish running a test. I click on ‘Actions’ to see the results of the test and I see a red ‘X’ for failure, and I tried and tried and tried but the workflow just would not budge. Until I realised that I created the workflow in the ‘main’ file of my repository instead of the ‘master’ file where I pushed all my code and where all my files were. So, I changed the default file from the main file to the master file and then I followed the same process to create a workflow to run a GitHub action and guess what; IT STILL DID NOT WORK. That is when I decided to give up on trying to run a GitHub action test on my app and just resolved that ‘if it works perfectly in Blue Stacks then I am good’. So that is the story of how I created my “History App”.

My personal favourite historical figure listed in my app is Michael Jackson because I grew up listening to his songs and I taught myself how to dance by watching and copying all his videos. (I can do the best moonwalk!)

# Links:

<https://youtube.com/watch?v=d0uith-LE3o>

<https://github.com/marketplace/actions/automated-build-android-app-with-github-action>