# App Report

I have been tasked with creating a history app to inspire young minds and educte the future generation of software developers. I have created an Android App that compares user’s ages to famous figures from History, and gives them a short description about the famous figure. This app will be called “The History App”.

## Objective

The objective of creating this app was to test my knowledge of Kotlin and Android Studio. To create a begginer friendly app that makes uses of decision making such as if statements and when statements.

## Functionality

The core functionality of this app is, it takes in a user’s age as input, checks that this is a valid .eg 39 or 95. The app will then compare the age to the ones that we have within our app display an appropriate message such as “Your are 39 years old, which is the same age as Martin Luther King Jr. Martin Luther King Jr. was an American Baptist minister and civil rights activist, known for his role in the advancement of civil rights using nonviolent civil disobedience.”. Or a alternative message such as “There is no historical figure known to be around 42 years old” if there is no known record of a historical figure of that age within our app.

## Target Audience

The target audience for this app is aspiring software engineers. This is to show them what is possible with just a few weeks of coding practice. This application will allow aspiring software developers to test their skills and knowledge when it comes to recreating the project or brining their own ideas to life.

## Unique Selling Point

The unique selling point of this app it that it has wide verity of famous figures from all over the world and from different eras in history such as Leonardo da Vinci, Princess Diana and Alexander the Great. Within our app there are more than 35 famous figures, with an accurate description of who they are any why they are famous.

## Platfrom Comaptibility

This app will run natively on android operating systems. This app will be able to run android 9 and older. Or any operating system with an appropriate emmulator such as bluestack on windows.

# Generating Famous Figures

ChatGPT was used for creating a list of all famous figures. Chat GPT was provided with the following prompts:

1. Create a table of the most famous dead people. The table should have a column for their age of death, name and surname, and what the person was famous for in one sentence.
2. Create a new list. This list must include famous figures from the age of 20 to 100 years old.
3. The list must have famous figures all the the ages.
4. The list should have at least 40 famous figures

ChatGPT provided me with the following information:

1. Joan of Arc (Age at Death: 19) - French heroine and saint who played a key role in the Hundred Years' War.
2. Alexander the Great (Age at Death: 32) - Ancient Macedonian ruler who created one of the largest empires in history.
3. Anne Frank (Age at Death: 15) - Jewish diarist during the Holocaust, known for "The Diary of a Young Girl."
4. Wolfgang Amadeus Mozart (Age at Death: 35) - Influential composer of the Classical era.
5. Vincent van Gogh (Age at Death: 37) - Dutch post-impressionist painter whose work is known for its emotional intensity and bold colors.
6. Cleopatra (Age at Death: 39) - Last active ruler of the Ptolemaic Kingdom of Egypt, known for her intelligence and strategic prowess.
7. Malala Yousafzai (Age at Death: Alive) - Pakistani activist for female education, youngest Nobel Prize laureate.
8. Alexander Fleming (Age at Death: 73) - Scottish biologist and pharmacologist, discovered penicillin.
9. Albert Einstein (Age at Death: 76) - Nobel Prize-winning physicist famous for developing the theory of relativity.
10. Nelson Mandela (Age at Death: 95) - Anti-apartheid revolutionary and former President of South Africa, known for his efforts to end racial segregation.
11. Queen Victoria (Age at Death: 81) - Long-reigning monarch of the United Kingdom during the Victorian era.

(Open AI 2024)

This information was later refined to make sure that all the ages were within our range.

# Design considerations

For this app I have gone for a minimal User Interface (UI). This design choice was chosen as it allows for easier legiblity of content. This design made it possible to create consistency with my fonts, spacing, colour of buttons, and font weights such as bolding texts, and consistent font size such as 20px.

## Consistency

We must consider uniform design decisions within our app, such as, fonts, colours and button styles says Babich, N. (2016).

## Legible Content

Smartphones have relatively small screens, which means that one of the challenges of mobile design is to fit a lot of information on a small UI says Babich, N. (2016).

The following image shows what the final app layout will look like.



## Background

For this project I chose an image that a wide variety of contrast.

# GitHub Utilization

GitHub was an integral part of the development of this app. GitHub allowed me to document my code using a README file. This has helped me in explaining the purpose of my code. Allowing others to understand the project with ease.

# GitHub Actions

We did not use GitHub actions for this project, due to time constraints.

# Logic Design

This app’s logic is structured in such a way that it checks if the user has entered a number in the appropriate field. This is done by checking that there is no null value in the age variable and display an appropriate message if there is. For example

if(etAge == null) { t

vResult.text = “Please enter a valid age e.g 39 or 95”

}

The next step in this app logic is to check if the age is within our range. This can be done by using the else statement directly after the above messioned if statement, and display an appropriate message if the age is not within our range, such as, 15 or 110. For example:

else {

if( age.toString().toInt() < 20 || etAge.toString().toInt() > 100 ) {

tvResult.text = “Please enter an age that is within the range of 20 to 100 years old”

}



A screenshot of a computer screen

Description automatically generatedThe following message will also be displayed if the user’s input is not within the range.

Once the user’s input has been checked against the other conditions, and manages to read this step. We must check if there is a match for the user’s input. This can be done by nesting a when statement inside of an else statement directly after the above if statement. For example:

Else {

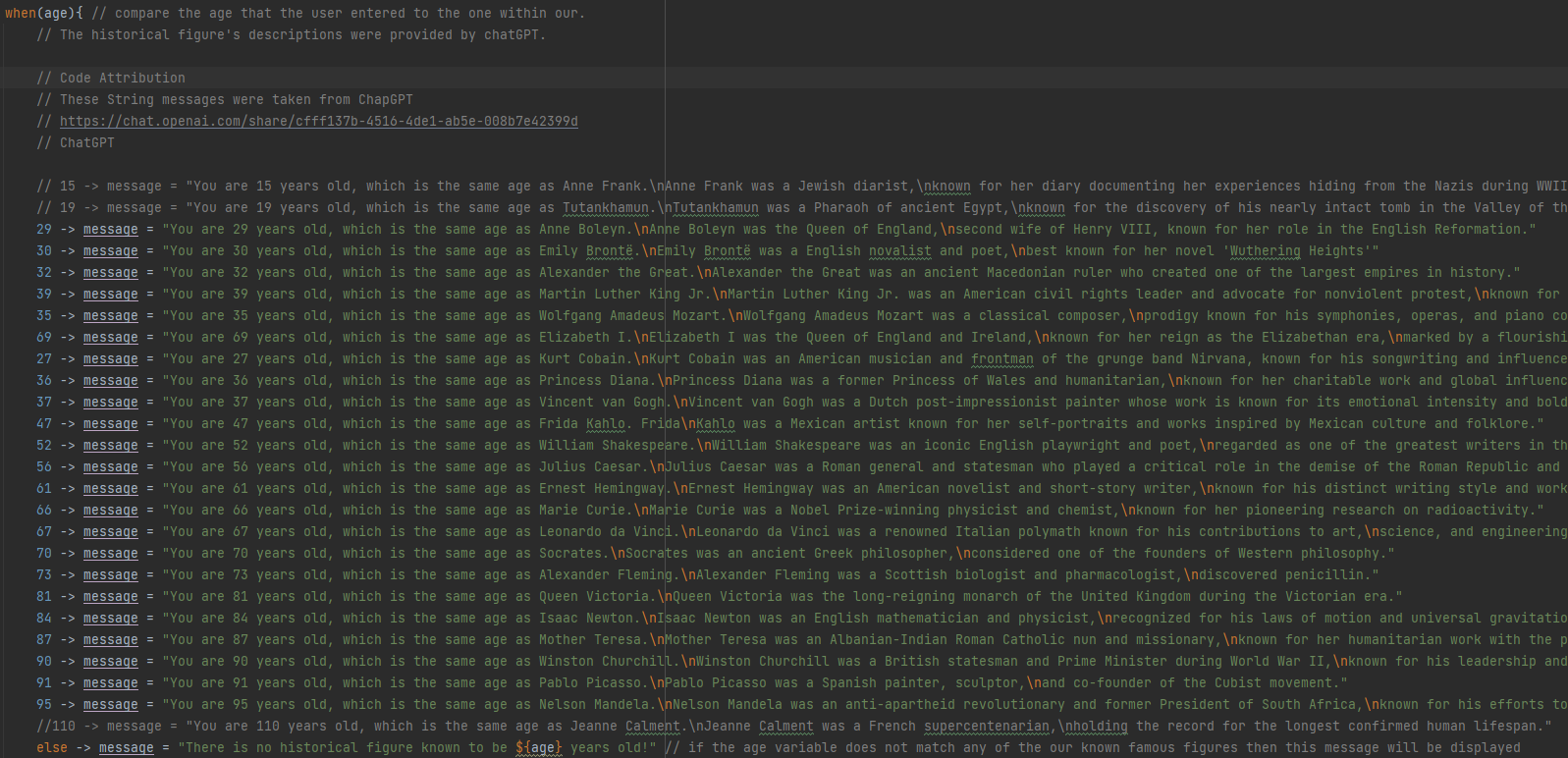
when(etAge.toString().toInt()){

39 -> message = “Hi you are 39 years old, which is the same age as Amelia Earhart. Amelia Earhart was an American aviation pioneer and the first female aviator to fly solo across the Atlantic Ocean."

else -> message = “There is no historical figure known to be this age within our app.”

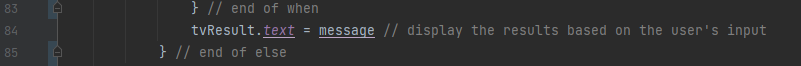
}

}



The last step we will need to do is to display an appropriate message passed on the user’s input.

tvResult.text = message



For functionality we will also create a clear button that will clear the age input field and and result field. This will be done as folows:

btnClear.setOnClickListener {

val clear = “” // this will

etAge.clear() // use the build in method to clear this field

tvResult.text = clear // set the text attribute to an empty string

}

A screenshot of a computer program

Description automatically generated

# Implementation Detials

There was no use of third-party libraries or APIs for this application.

There were a few technical challenges faced with this application, such as, the background constraints. During this process, I struggled to get the position right for all my other components such as tvResult component.

# Testing and Quality Assurance

There was no testing or qaulity assurance for this application. Due to time constraints.

# Conclusion

This was a good indication of my programming abilities and highlighed all the areas that I need to work on to become a better developer. Such as, Logic design, naming conventions and commenting.

I am looking forward to utilizing more tools in future such as GitHub Actions. So that I can understand how my application will work within the real world.

I think one of my biggest setbacks during this project was my time management. In future I will be creating list of all the todo items for any project. This will hopefully allow me to have structure within my work, and allow me to stay on track with my tasks.

# YouTube Link

<https://youtu.be/0a7-F620sSY?si=6OrHoK5Rg1xHy0dE>

# Reference list

Open AI. (2024). ChatGPT 3.5

Unsplash. “Photo by Adrianna Geo on Unsplash.” Unsplash.com, 29 Oct. 2019, unsplash.com/photos/a-painting-on-the-ceiling-of-a-building-1rBg5YSi00c. Accessed 5 Apr. 2024.

Babich, N. (2016). Mobile UX Design: Key Principles. [online] Medium. Available at: https://uxplanet.org/mobile-ux-design-key-principles-dee1a632f9e6 [Accessed 5 Apr. 2024].

www.uxdesigninstitute.com. (2023). Your ultimate guide to mobile app design. [online] Available at: https://www.uxdesigninstitute.com/blog/ultimate-guide-to-mobile-app-design/ [Accessed 5 Apr. 2024].