Tebogo Nchabeleng

IMAD

Assignment 1

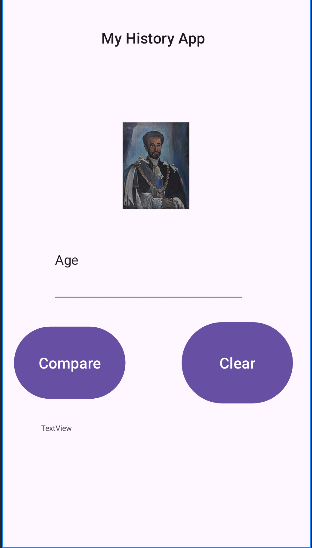
ST10435751

**APP report**

**Purpose of the App**

The purpose of our history app is to offer users a unique perspective on the passage of time by providing insights into individuals who died at specific ages throughout history. By highlighting notable figures who passed away at a particular age, our app aims to personalize historical events and milestones, making them more relatable and engaging for users. Through curated profiles, biographical information, and contextual details, users can explore the lives, achievements, and legacies of individuals from various epochs and cultures. Whether users are curious about the accomplishments of historical figures or seeking inspiration from their life stories, our app offers a captivating journey through time, fostering a deeper appreciation for the human experience across different eras. Ultimately, we aim to encourage reflection, curiosity, and a greater understanding of the interconnectedness of history and the individuals who have shaped it. (OpenAI, 2024)

Design considaration



Intuitive Design

The Design is easy to follow as there is a clear button which clears the age that th user has inserted in the app.The second button is comapre which is button that the user will use to compare the ages .At the very top the heading is my history app ehich shows the user what the app is about .Then finally there is a text view which shows the user the information about the person who passed on at a specific age.

Content Prioritisation

Neccesary interface elements are included in the app for the benefit of the user .The app has a great coding that runs behind to show information to the user as he /she types in the age of the person.This a fun way to interact with the user and the user with the app.

Legible Text Content

The Font is large enough for the user to read .The text font is in letters that a child can read we specifically avoided italics as some kids may have a strain with reading the words on the app. The text size is large enough for the user to read .The text size was designed in a way that it will be easy for a child to understand what he /she wants to archieve on the app.

Make Interface Elements Clearly visible

The user can clearly see the components on the screen as they are not over crowded and the space is perfectly set for the user to not fell the app is clusted

Hand Position control

The hand position control interface enhances the user experience by allowing users to interact with the app using gestures or hand movements. For example, users can activate the hand position control feature by raising their hand in front of the device's camera or using touch gestures on the screen to simulate hand movements.

Images

One reason we chose Haile Selassie's image for our history app is his significant and multifaceted impact on history. As the Emperor of Ethiopia from 1930 to 1974, Haile Selassie played a pivotal role in shaping not only the history of Ethiopia but also exerted influence on the broader geopolitical landscape of Africa and the world. His leadership during Italy's invasion of Ethiopia in the 1930s and his efforts to modernize his country have left a lasting legacy. Additionally, Haile Selassie's role as a prominent figure in the Rastafari movement, where he is revered as the Messiah by followers, adds another layer of cultural and historical significance. By featuring his image, we aim to highlight his complex and impactful contributions to history, inviting users to explore his life and legacy within the broader context of world events.

Here are some key aspects of Haile Selassie's life and legacy:

* **Leadership during Italian Invasion**: Haile Selassie's leadership during Italy's invasion of Ethiopia in 1935-1936, known as the Second Italo-Ethiopian War, brought him international recognition. Despite Ethiopia's eventual defeat, Haile Selassie's appeal to the League of Nations for assistance highlighted the brutality of Italian colonialism and earned him admiration as a symbol of African resistance.
* **Restoration of Sovereignty**: After Ethiopia was liberated from Italian occupation in 1941, Haile Selassie returned to power and embarked on a series of reforms aimed at modernizing the country. He implemented measures such as the abolition of slavery, the introduction of a constitution, and the establishment of modern educational and healthcare systems.
* **Pan-Africanism**: Haile Selassie was a key figure in the Pan-African movement, advocating for unity and solidarity among African nations. He played a significant role in the establishment of the Organization of African Unity (OAU) in 1963, which later became the African Union (AU), aiming to promote cooperation and development across the continent.
* **Rastafari Movement**: Haile Selassie's coronation in 1930 was a pivotal event for the Rastafari movement, a religious and cultural movement that emerged in Jamaica in the early 20th century. Many Rastafarians believe Haile Selassie to be the messiah, interpreting his titles such as "Conquering Lion of the Tribe of Judah" as evidence of his divine status.
* **Deposition and Legacy**: Haile Selassie's reign came to an end in 1974 when he was overthrown in a coup d'état led by the Derg, a Marxist-Leninist military junta. He spent his final years under house arrest until his death in 1975. Despite his deposition, Haile Selassie remains a revered figure in Ethiopian history and continues to be celebrated by Rastafarians worldwide.
* **Visual Engagement**: Pictures capture users' attention and create an immediate visual connection, drawing them into the app's content. By placing pictures at the top, we engage users from the moment they open the app, enticing them to explore further.
* **Contextualization**: Pictures provide context for the app's content, offering visual cues that help users understand the historical period, event, or figure being discussed. Whether it's a portrait of a historical figure, a scene from a significant event, or an artifact from a particular era, pictures help users visualize and contextualize the information they're encountering.
* **Enhanced Storytelling**: Pictures enhance the storytelling aspect of the app by complementing textual information with visual imagery. They can evoke emotions, convey meaning, and bring historical narratives to life in a way that text alone cannot. By incorporating pictures at the top of the app, we create a more immersive and compelling storytelling experience for users.
* **Navigation Aid**: Pictures can also serve as navigational aids, guiding users to different sections or topics within the app. For example, a series of rotating images at the top of the app could represent different historical periods or themes, allowing users to click or swipe through them to explore further.
* **Aesthetic Appeal**: Placing pictures at the top of the app enhances its aesthetic appeal and visual appeal. Well-chosen images can create a sense of elegance, professionalism, and quality, making the app more inviting and enjoyable to use.

Design principles

* **Simplicity**: One of the most important design principles is simplicity. A history app should present information in a clear and straightforward manner, avoiding clutter and unnecessary complexity. This means using clean layouts, intuitive navigation, and concise content to ensure that users can easily find and understand the information they're looking for.
* **Consistency**: Consistency is essential for creating a cohesive user experience. In a history app, consistency can be achieved through uniformity in design elements such as color schemes, typography, and iconography. Consistent navigation patterns and layout structures also help users navigate the app more easily and feel more comfortable exploring its content.
* **Visual Hierarchy**: Visual hierarchy guides users' attention and helps them prioritize information. In a history app, important historical events, figures, or periods should be prominently featured using visual cues such as larger text, bold colors, or prominent images. This helps users quickly identify key content and navigate through the app more efficiently.
* **Accessibility**: Accessibility is a fundamental design principle that ensures all users, regardless of their abilities or disabilities, can access and use the app effectively. This includes considerations such as providing alternative text for images, ensuring color contrast for readability, and implementing navigation features that are usable with assistive technologies.
* **Engagement**: To keep users engaged and encourage continued exploration, a history app should incorporate elements of interactivity and personalization. This could include features such as quizzes, timelines, interactive maps, or customizable user profiles. By making the app more interactive and tailored to individual users' interests, you can enhance engagement and encourage repeat usage. (OpenAI, 2024)

3. 1. Create a New GitHub Repository:

Go to the GitHub website (<https://github.com/>) and sign in to your account.

Click on the "+" icon in the top right corner and select "New repository".

Give your repository a name (your student number and name – in one word), add a description (use IMAD5112 Assignment 1) and choose public.

Click on the "Create repository" button.

2. Initialise the Repository with a README File:

After creating the repository, you'll see an option to "Initialize this repository with a README". Check this option to create a README file.

Click on the "Create repository" button to finalize the creation of the repository.

3. Commit and Push Your Project Files to the GitHub Repository:

In Android Studio, go to VCS (Version Control System) -> Import into Version Control -> Share Project on GitHub.

Log in to your GitHub account if prompted, and select the repository you created earlier.

Click on the "Share" button to push your project files to the GitHub repository.

4. Regularly Commit and Push Your Code as You Make Progress:

After the initial push, continue making changes to your project in Android Studio.

Whenever you make significant progress or changes, commit your changes locally in Android Studio using VCS -> Commit Changes.

Once committed, push your changes to the GitHub repository using VCS -> Git -> Push.

**Testing and Automated Testing**:

1. Conduct Manual Testing:

Manually test your app to ensure it functions seamlessly and offers an enjoyable educational experience for learners.

To test various features and user interactions do the following:

1. Create a New Test Class:

In your Android project, navigate to the tests directory (or create it if it doesn't exist).

Create a new Kotlin file for your test class. Name it appropriately to indicate what component or functionality you are testing.

2. Write Test Methods:

Inside the test class, write test methods like the example below.

3. Use assertions to verify the expected behaviour of your code.

4. Run the Tests:

Run the tests using the testing framework's tools provided by Android Studio or through the command line.

Sample of a test class using JUnit:

import org.junit.Assert.\*

import org.junit.Test

class MyUnitTest {

@Test

fun testWhenStatement() {

// Test case for a when statement

val result = when (25) {

21 -> "Some Body, famous for …, died at this age" // use your own wording (copy from your code) – only need 3 ages from your code for the test here.

34 -> "Another Some Body, famous for ……., died at this age"

63 -> "Someone Else, famous for …..., died at this age"

else -> "Nobody famous known to me died at this age"

}

assertEquals("Nobody famous known to me died at this age", result)

}

}

2. GitHub Actions for Automated Testing:

Set up GitHub Actions to automatically run tests and build your code whenever changes are pushed to the repository.

Create a GitHub Actions workflow (.github/workflows/tests.yml) to run tests automatically on every push:

Create a .github/workflows directory in your project repository.

Inside this directory, create YAML files defining your GitHub Actions workflows for testing and building.

**Sample GitHub Actions Workflow for Testing (tests.yml):**

name: Run Tests

on: [push]

jobs:

test:

runs-on: ubuntu-latest

steps:

- name: Set up JDK

uses: actions/setup-java@v2

with:

distribution: 'adopt'

java-version: '11'

- name: Check out code

uses: actions/checkout@v2

- name: Build and test

run: ./gradlew test

Set Up Automated Build:

Create a GitHub Actions workflow (.github/workflows/build.yml) to build the APK automatically on every push.

**Sample GitHub Actions Workflow for Building (build.yml):**

name: Build APK

on: [push]

jobs:

build:

runs-on: ubuntu-latest

steps:

- name: Set up JDK

uses: actions/setup-java@v2

with:

distribution: 'adopt'

java-version: '11'

- name: Check out code

uses: actions/checkout@v2

- name: Build APK

run: ./gradlew assembleDebug

Test your workflows by pushing changes to your repository and observing the actions running in the "Actions" tab on GitHub.

References

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

(OpenAi,2024)

References

OpenAI. (2024). ChatGPT 4.0.

OpenAI. (2024). ChatGPT 4.0

OpenAI. (2024). ChatGPT 4.0.