Potential Sample Reflections

Work Reflection – Kotlin Bus Tracker

Date: April 4, 2024

Project Overview

In my mobile app development course, I developed a bus tracking application for android platforms using Halifax Transit's real-time API similar to other transit APIs such as the City of Toronto's myttc.ca. This project helped build my familiarity with reading and utilizing API systems in mobile development. The primary goal was to create a user-friendly application with smooth navigation and an intuitive interface while adhering to modern coding practices. The app was built using Kotlin, Android Studio, and MapBox to provide an interactive map for tracking buses in real time using popular development languages, tools, and software.

Key Decisions & Actions Taken

Our class was directed to utilize **Android Studio** as the primary development environment. However, I found that Android Studio had some frustrating issues that produced a lot of complications. As I did not have access to an android device I was limited to only Android Studio's built in device emulators which were often slow and crashed often which made testing my app very difficult. After completing this project for my mobile development course, I have began looking into other development tools and asking peers in related fields their preferences and I found Xcode could be a better alternative for this project's future development.

This was my first time using MapBox in any of my projects at NSCC and it was very simple to integrate into my app as MapBox is built with Kotlin. MapBox provided a very simple and clean interface and was a great help in my app's development.

It is common for the user's location to be required in most map-based applications and mine is no exception. Android Studio's phone emulators base locations are in Googleplex, California USA. I needed to make a request from the user to allow access to their current location data so that app could provide accurate and relevant information. In order to achieve this I had to research how Kotlin can gain these permissions.

When it came to what information our app would provide to users our professor allowed us some freedom on what information we wished to share. As our project was limited in time I chose three primary screens in this phase of development. I chose a Map, Alert, and Route screen as I believed these would be the most relevant information desired by the potential users.

Skills Applied & Developed

This project was developed using Kotlin, requiring a deep understanding of Scaffold, ViewModels, asynchronous data loading, StateFlows, and API integration. Scaffolding and ViewModels helped structure my UI efficiently while ensuring that my app wouldn't need to make multiple calls to the transit API when the user switched screens. By implementing asynchronous data loading I was able to prevent UI blocking allowing for a smooth user experience. Additionally, working with StateFlows and API integration enabled real-time bus tracking, dynamically updating the UI based on live data.

Project Outcomes

By the end of the allocated time, I successfully delivered a functional prototype featuring a seamless navigation system across three primary screens using Jetpack Compose's Scaffold. Additionally, I developed an interactive map screen with Mapbox, displaying real-time bus locations, route numbers, and occupancy status/capacity. status/capacity.

Areas for Improvement

In future updates, I plan to expand the displayed bus details by incorporating more data from the transit API as the current version only showcases a limited portion of the available information. Additionally, I aim to improve the UI design for the information screens. Currently, the Alert and Route screens display plain black text on a white background with minimal styling. By implementing a consistent design layout, the app will offer a more user-friendly and visually appealing experience. As part of the course requirements, I implemented test API calls to demonstrate proper API integration to my professor. Moving forward, I will remove these test calls and refine the codebase to enhance performance and sustainability. Finally, I plan to introduce a "Favorite Route" feature, allowing users to save frequently searched routes for quick access. This will significantly improve usability and provide a more personalized experience for potential commuters.

Conclusion

This project was valuable experience in developing a real-time mobile application with API integration. It strengthened my skills in **Kotlin, UI design, and state management**, while also highlighting areas for future enhancement. Moving forward, I aim to refine the app's functionality and user experience to make it more useful for daily commuters.