**Report: Tus Parking Mobile Application**

**Introduction**

The purpose of this report is to outline the development and features of the "Tus Parking" mobile application, designed to enhance the parking experience for students at Tus. The app aims to provide a convenient solution for students to find, book, and pay for parking spots on campus

**Application Overview**

**1. Name: Tus Parking**

The chosen name reflects the app's primary function, focusing on providing a streamlined parking experience for Tus students.

**2. Features**

**2.1 User Authentication**

The application begins with user authentication, ensuring secure access to the features. Students can log in using their credentials, creating a personalized experience.

**2.2 Location Services**

Upon logging in, the app directs users to Google Maps, leveraging location services for easy navigation and parking spot selection.

**2.3 Search and Selection**

Users can search for parking locations within the Tus campus. The app displays available parking spots, allowing students to choose the most convenient location based on their needs.

**2.4 Booking and Payment**

Once a parking spot is selected, the app facilitates booking and payment processes. Students can make secure payments through the app to reserve their chosen spot.

**Development Challenges**

Throughout the development process, several challenges were encountered, with a notable issue arising during the transition from design to implementation.

**Figma to Android Studio Integration**

The design of the app was initially created using Figma, a versatile design tool. However, exporting the design from Figma to Android Studio posed challenges. The integration process required careful consideration of design elements, resulting in some discrepancies.

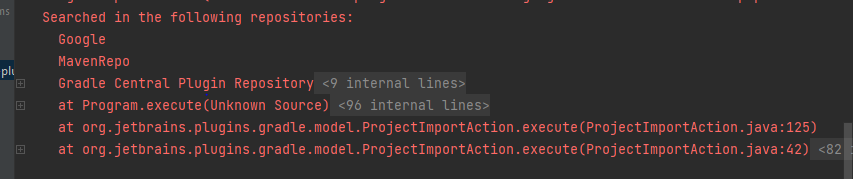
To address this issue, a thorough review of the design components was conducted, and adjustments were made during the implementation phase. Despite efforts to resolve the Figma-related challenges, the issues persisted, and the integration between Figma and Android Studio remained problematic. Faced with time constraints and a need for progress, I opted to take a pragmatic approach. Consequently, I resorted to hard coding several pages within the project. In order to streamline the development process and focus on critical functionalities, I decided to scale down the project scope. I prioritized the implementation of key screens such as login, sign up, and the landing screen where users can search for locations. Even though Figma wasn't playing nice, this change in strategy helped me finish the key parts of the project without getting stuck in the Figma mess.

**FireBase**

We were supposed to use firebase for the project as our main source of info storage . I used it for the login and signup screens and also for the landing page where the map would be displayed showing the parking spots that are empty and occupied. I encountered a few problems during this process. For some reason every single time I would connect a view model and connect to the screen my app would crash, the app would stop running. I have recently identified the reason for the particular problem. Wrong page linking could cause this problem or missing important information. Its hard to spot this problem as no error its displayed in the console of android studio.

**API For Google maps**

I ran into another problem while trying to add Google Maps to Android Studio. Turns out, getting an API key required creating a billing account and putting in credit card details, which wasn't ideal. Luckily, our class tutor came to the rescue and shared her account with me and some classmates. This way, we could implement the feature without any problems. But not everything went as planned I still got into an issue my gradle project didn’t wanted to sync with the new id’s / dependencies I added that would allow me to connect to the map.



I have identified a resolution to the issue I encountered on Stack Overflow. It appears that another user had faced a similar problem, and the solution was related to a missing dependency in my project configuration.

buildscript **{** dependencies **{** *classpath*("com.google.android.libraries.mapsplatform.secrets-gradle-plugin:secrets-gradle-plugin:2.0.1")  
 **}**

Another issue concerning maps is the compatibility with various phone versions in the emulator. Unfortunately, not all emulator instances support map functionality due to the requirement of an updated Google Play Store. The update process for the Google Play Store can be time-consuming. To address this, I experimented with different phone emulations until I identified one that successfully displays the maps

**Payments**

An attempt for payments has been made. I used a view model for it to store the user input into the firestore such as the card number, expiry date and cvv. Unfortunately it didn’t go as planned. The user information wasn’t added to the firestore the biggest reason it could be because the collections weren’t made right. I tried to find videos how to create a payment collections but couldn’t find any specific so I tried my best to create it myself.

**Conclusion**

In conclusion, while the Tus Parking app holds great promise for simplifying parking for Tus students, it has yet to reach its full potential in terms of user adoption. Despite encountering and overcoming development challenges, such as integrating designs from Figma, Firebase issues, and Google Maps complications, the app is still in the early stages of gaining widespread usage.

The transition from Figma to Android Studio required strategic decision-making, and although we successfully implemented key functionalities, the app has not yet achieved the desired level of user engagement. Challenges with Firebase and Google Maps API, despite resolutions, have contributed to a slower-than-expected user adoption rate.