**Signal Clone App - Android Project**

**Project Overview**

This project is a clone of the Signal application, focusing on creating core user interaction flows like setting up a profile, entering a phone number, and creating a PIN. The app is developed using Android Studio, with layouts built using XML for static screens and Kotlin for handling activity logic. The primary goal is to simulate the key initial onboarding steps seen in the Signal app, using intuitive and modern UI/UX practices.

**Key Features:**

* **Profile Setup Screen**: Users can enter their profile information, such as first name, last name, and profile picture.
* **Phone Number Screen**: Users enter their phone number to sign up and select a country code using a drop-down spinner.
* **Create PIN Screen**: Users are prompted to create a secure PIN with options for an alphanumeric password.

**Setup Instructions**

**Prerequisites:**

* **Android Studio**: Ensure Android Studio is installed on your machine.
* **Git**: Make sure Git is installed and configured properly.
* **Android SDK**: Ensure the required SDK versions are installed via the SDK Manager in Android Studio.

**Steps:**

1. **Clone the repository to your local machine**:

bash

git clone **https://github.com/Malaika-Farid/MySpotify.git**

1. **Open the project in Android Studio**:
   * Navigate to File > Open in Android Studio, then select the cloned project folder.
2. **Sync the project with Gradle**:
   * After opening the project, Android Studio will prompt you to sync the Gradle files. Click on Sync Now to ensure all dependencies are correctly downloaded.
3. **Run the app on an emulator or a physical device**:
   * **Emulator**:
     + If you don’t have an Android Virtual Device (AVD) set up, go to Tools > AVD Manager to create one.
     + After setting up the AVD, click Run > Run 'app' or press Shift + F10 to run the app on the emulator.
   * **Physical Device**:
     + Connect your Android device to your machine via USB.
     + Ensure USB Debugging is enabled on your device (Settings > Developer Options > USB Debugging).
     + Select your connected device from the Run/Debug configurations dropdown.
     + Click Run > Run 'app' or press Shift + F10 to install and run the app on your physical device.
4. **Install missing dependencies (if needed)**:
   * If you encounter any errors regarding missing SDK versions or libraries, open the SDK Manager in Android Studio (Tools > SDK Manager) and install the required SDKs and dependencies.

**Screens Designed and Their Purpose**

**1. Profile Setup Screen**

* **Layout file**: activity\_profile\_setup.xml
* **Purpose**: This screen allows the user to set up their profile by providing their name and profile picture. There’s a form to enter a first name (required) and last name (optional), as well as an option to add a profile picture.

**2. Phone Number Screen**

* **Layout file**: phone\_number\_layout.xml
* **Purpose**: This screen asks the user to enter their phone number and choose a country code from a drop-down list. The layout includes an editable field for the phone number and a spinner for selecting the country code.

**3. Create PIN Screen**

* **Layout file**: activity\_create\_pin.xml
* **Purpose**: This screen prompts the user to create a secure PIN that will help encrypt and secure their data. There is a button to toggle between creating a numeric or alphanumeric PIN. This is essential for ensuring the security of user accounts.

**Technical Challenges Faced**

**1. Spinner for Country Code:**

* Implementing the spinner for country codes required ensuring the design and functionality worked seamlessly across different device configurations. The challenge was aligning it visually with the phone number input field to ensure a clean look.

**2. PIN Entry Input:**

* Implementing a masked input for the PIN, ensuring that the dots are displayed for user input while securely masking the digits, required handling focus and keyboard interaction properly.

**3. Custom XML Styling:**

* Creating custom backgrounds, rounded corners, and shadows for buttons and input fields was another challenge. Achieving a balance between a modern aesthetic and maintaining compatibility with multiple screen sizes involved considerable fine-tuning.

**Future Plans**

1. **Backend Integration**:
   * Add Firebase or another backend service to handle user authentication, real-time messaging, and secure storage of user profile data.
2. **Advanced Error Handling**:
   * Implement detailed error messages and validation feedback for the phone number and PIN entry screens.
3. **User Authentication**:
   * Implement OTP (One-Time Password) authentication for phone number verification, ensuring secure user registration.
4. **Dark Mode Support**:
   * Add support for system-wide dark mode, ensuring the app dynamically changes based on user settings.
5. **Security Enhancements**:
   * Encrypt all sensitive information like the user's PIN and other credentials stored on the device to enhance security.