International Institute of Information Technology
Bangalore

Software Defined Network and Network Function Virtualization
NC 864

Project Report

Submitted by: *Yash Dharmesh Mogal* (IMT2020537)

Contents

1 Problem Statement:

Create a login screen for an Android. This login screen allows user to register with his email id, and allows him to store user metadata (you can assume any structure of metadata.). Also, the app login screen supports cross authentication using OAUTH with Google and Facebook accounts apart from registering. After Login, just display that the "`<Your Name>` has logged in using `<email | gmail |github>`.

2 Project Description:

The goal of this project is to implement a simple and secure method of user authentication in the Android application. To make this process more user-friendly and smooth, additional login options, such as through a Google/Github account, will be used. So this Android app-based login screen will consist of two elements - Sign-up page, Login page.

2.1 Features:

- On the login page, the user will be asked for its `<email id | username>` and password. If the entered values match against corresponding values in the database then it allows user to login or else asks the user to signup on the sign-in page.
- The OAUTH protocol will be used to implement this functionality.
- If a new user wants to use the application, they will be using signin page to register themselves into the database by providing required metadata.
- Users are asked to submit Name, Email, Date of Birth, Mobile Number, Gender, Password as metadata required for registration purpose. This is subject to change if additional data required to facilitate login via OAuth.

3 Methodology used to implement:

3.1 OAuth Protocol:

OAuth (Open Authorization) is a framework that enables third-party applications to access user resources without requiring the user's credentials. OAuth is intended for use in several contexts, such as login, social network integration, and API access. OAuth can be used in the context of an Android app to build a login screen that allows users to sign in using their Google or Github or LinkdIn credentials. This can be done by utilizing the OAuth authorization code flow. The OAuth authorization code flow works as follows:

- The user clicks on the "Sign in with Google" or "Sign in with Github" button on the login screen.
- The app redirects the user to the OAuth provider's authorization page.
- The user authenticates with the OAuth provider and grants the app the necessary permissions.
- The OAuth provider redirects the user back to the app with an authorization code.
- The app uses the authorization code to obtain an access token from the OAuth provider.
- The app uses the access token to access the user's resources.

3.2 Configuration of Application in Firebase:

- Authentication and database Firestore are the two modules that our Firebase project uses. For OAuth-based login, the authentication module adds an extra layer of security and requires us to add the Security Hash Algorithm (SHA-1 and SHA-256) keys.
- Currently, our app is set up for Android-based devices, but iOS support may be added in the future. Tools like the Flutterfire CLI are used to accomplish this configuration.
- Moreover, it is necessary to create a new GitHub OAuth project on GitHub in order to use GitHub-based authentication. This project supplies the client ID and server secret.

3.3 Design Flow:

- The user will input their username and password on the login screen. The user will be logged in if they have an account in the database. They can also use their Google or Github accounts to log in.
- The login page will prompt the user to register or create a new user if the user is not already registered in the database and will display an error. There will be a sign-up page where the user may enter their details and generate a new password for their account.
- A user will not be able to login using their Google or Github account if not registered already. Therefore, it will ask the user to grant permission to share metadata from their Google or Github account to the application register the user into their database.
- The user will be brought to the sign-up page if they have not yet registered for the application, even if they choose to sign in using their Google or Github account. On this screen, they must enter their information and create a password for their account. If checking in using their Gmail or Github account fails, this password will allow them to login directly using their credentials.
- When the user clicks the sign-up button, their information is saved in the database. The user should be able to log in using their Google or Github accounts, as well as the password and email address they provided.

3.4 Tech Stack:

The Flutter and dart tech stack will be utilized to design the application UI and implement the backend functionality for a Flutter app. The database utilized in the project to store user meta-data will be Firebase.

3.5 Setup and Installation of Implementation Environment:

The following steps are required for installing and setting up the required technology stack for the SDN NFV OAuth Project built with Flutter.

1. **Install Flutter:** Follow the official documentation for installation instructions: https://docs.flutter.dev/get-started/install.
2. **Install and Set up Android Studio:**
   (a) Download and install Android Studio from: https://developer.android.com/studio/install.
   (b) For Linux/Ubuntu users, Android Studio is readily available in the Ubuntu Software Center.

(c) Set up Android SDK Command-line Tools:

i. Open Android Studio.

ii. Click "More Actions" on the boot window and select "SDK Manager".

iii. In the "Android SDK" section, navigate to "SDK Tools".

iv. Check the box for "Android SDK Command-line Tools (latest)" and click "OK".

(d) Accept Android licenses: Run `'flutter doctor --android-licenses'` in your terminal.

3. **Clone the Project:** Download the project code from the GitHub repository: https://github.com/Y09mogal/SDN_NFV_OAuth_Project.

4. **Run the Application:**

(a) Start an Android emulator or connect your Android device via USB and enable USB debugging:

i. Emulator setup: https://developer.android.com/studio/run/managing-avds.

ii. Device setup: https://developer.android.com/studio/run/device.

(b) Open a terminal in the project directory.

(c) Run `'flutter run'` in your terminal.

(d) Select the desired device to run the application on.

4 Future Works:

Some of the potential features that could be implemented on top this project as future aspects are as follows:

- Inclusion of additional authentication providers like Outlook or Facebook for a broader range of login options.
- Addition of a "Remember Me" feature to enhance user convenience and streamline the login process.
- Implementation of two-factor authentication to validate the authenticity of accounts using Email/Password-based authentication.

5 Github Repo:

Use this link to my github repository which contains all the code required to build this project.

6 Reference:

1. https://www.youtube.com/watch?v=4fucdtPwTWI&ab_channel=MitchKoko
2. For Flutter setup https://docs.flutter.dev/get-started/install
3. For Andriod Studio setup https://developer.android.com/studio/install
4. https://xiaomiui.net/how-to-enable-developer-options-on-xiaomi-devices-2504/
5. https://auth0.com/intro-to-iam/what-is-oauth-2