GitHub User Profile Android App - Documentation

Overview:

The **GitHub User Profile App** is an Android application built with **Kotlin**, **Jetpack Compose**, and **MVVM architecture**. The app communicates with the **GitHub public API** to fetch and display a user's profile information, including the list of repositories, and shows detailed information about each repository.

This documentation covers the following topics:

1. Architecture Overview

This app follows the **Model-View-ViewModel (MVVM)** architecture pattern:

- **Model**: Responsible for the business logic and fetching data from the GitHub API.
- **ViewModel**: Exposes the state to the UI and handles user interactions. It uses Kotlin Coroutines to fetch data asynchronously.
- **View (Jetpack Compose)**: Displays the UI based on the state from the ViewModel.

2. Functional Requirements

2.1 Core Features

1. **User Profile Input**: The user can input a GitHub username, and the app will display the user's name and avatar.
2. **Repository List**: The app shows a scrollable list of all the repositories owned by the specified user, including their names and descriptions.
3. **Repository Details**: Clicking on a repository in the list will navigate to a detail screen, showing more information like stars and forks.
4. **Total Forks Badge**: On the repository detail screen, the app displays the total number of forks across all repositories. If the total number of forks exceeds 5000, a star badge (with red/gold color) is displayed.

2.2 Additional Requirements

- **Senior Developer Feature**: The app displays the total number of forks for all repositories. If forks exceed 5000, a special colored text badge is displayed.

3. Design

3.1 User Interface

The app uses **Jetpack Compose** for the UI:

- **User Avatar and Name**: Displays the user's avatar and name at the top of the screen.
- **Repository List**: A scrollable list showing the name and description of each public repository.
- **Repository Detail Screen**: Shows detailed information about the repository (e.g., stars, forks).
- **Responsive Layout**: The UI automatically adjusts to different screen sizes.

3.2 Navigation

- **Navigation** between the repository list and the detail screen is handled using the NavController provided by Jetpack Compose.

4. API Usage

The app communicates with the **GitHub API** to retrieve the necessary data.

4.1 API Endpoints

1. **User Information**:
    - **Endpoint**: https://api.github.com/users/{userId}
    - **Response Fields**:
        - name: String (the user's display name)
        - avatar_url: String (URL to the user's avatar image)
2. **User Repositories**:
    - **Endpoint**: https://api.github.com/users/{userId}/repos
    - **Response Fields**:
        - name: String (repository name)
        - description: String (repository description)
        - stargazers_count: Integer (number of stars)
        - forks: Integer (number of forks)

4.2 Data Models

The app uses two data models for API responses:

- UserResponse: Contains the user's name and avatar URL.
- RepoResponse: Contains the repository's name, description, stargazers count, and forks count.

5. Implementation Details

5.1 Dependencies

The project uses the following dependencies:

- **Jetpack Compose**: For UI.
- **Retrofit**: For making API requests.
- **Gson Converter**: For parsing the JSON responses.
- **Kotlin Coroutines**: For handling asynchronous tasks.
- **Coil**: For loading images (e.g., user avatars).

5.2 MVVM Structure

5.2.1 ViewModel: UserViewModel

- Manages the app's state using MutableStateFlow.
- Fetches data from the **GitHub API** using the GithubRepository.
- Exposes the following states:
    - Loading: Indicates that data is being fetched.
    - Success: Contains the user's profile and repositories data.
    - Error: Displays an error message if the request fails.

5.2.2 Repository: GithubRepository

- Interacts with the **Retrofit** client to make API calls to fetch user and repository data from GitHub.

5.2.3 Retrofit Client

- **RetrofitClient** sets up the GitHub API client using the base URL https://api.github.com/.
- Provides methods to get user details and repositories.

5.3 UI Implementation

The UI is written entirely in Jetpack Compose and consists of:

1. **MainActivity**: The entry point, which displays the user screen.
2. **InputScreen**: Input github Id and and click on search button to retrieve the data.
3. **UserScreen**: A composable function that displays the user's profile and repositories.
4. **UserDetails**: A composable function that displays the detailed information about a user's repositories.

5. **LoadingScreen**: A simple loading spinner to indicate network activity.

Code Structure

- **MainActivity**: Initializes the UserViewModel and renders the UserScreen.
- **UserScreen**: Fetches and observes UserViewModel state to render the user's profile and repository list.
- **UserDetails**: Displays a detailed view of a selected repository.

6. Testing

6.1 Unit Testing

The app uses **JUnit** to test the business logic in the ViewModel.

6.2 UserViewModelTest

- **Test Case 1**: validate user data is fetched successfully.
- **Test Case 2**: Validate network error is handled when exception occurs.

6.3 RepositoryViewModelTest

- **Test Case 1**: Validate repositories are fetched successfully.
- **Test Case 2**: Validate repository API error handled when exception occurs.

7. Running the Project

7.1 Prerequisites

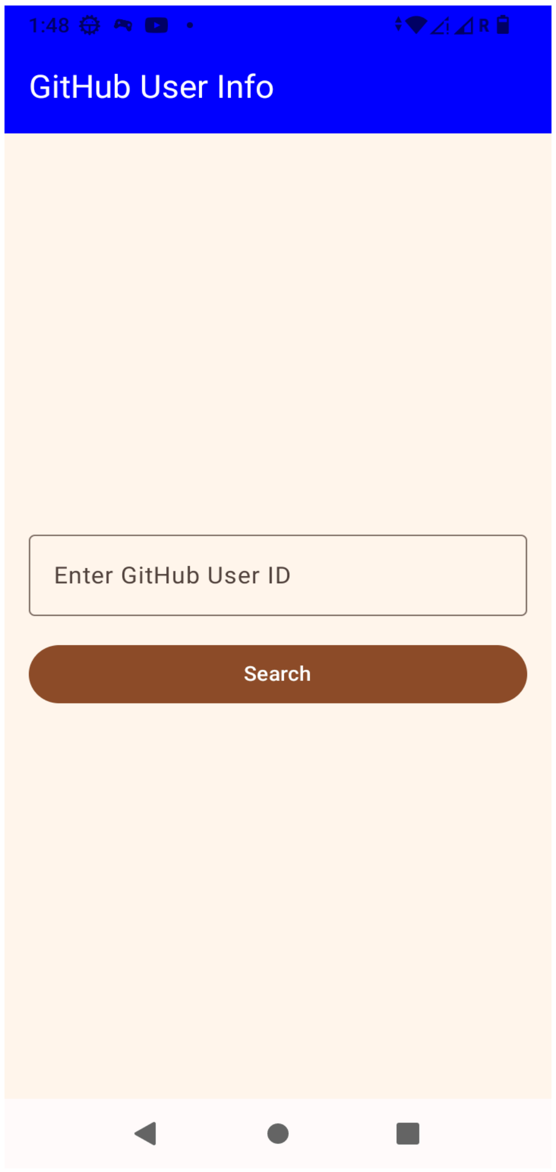
Ensure the following are installed:

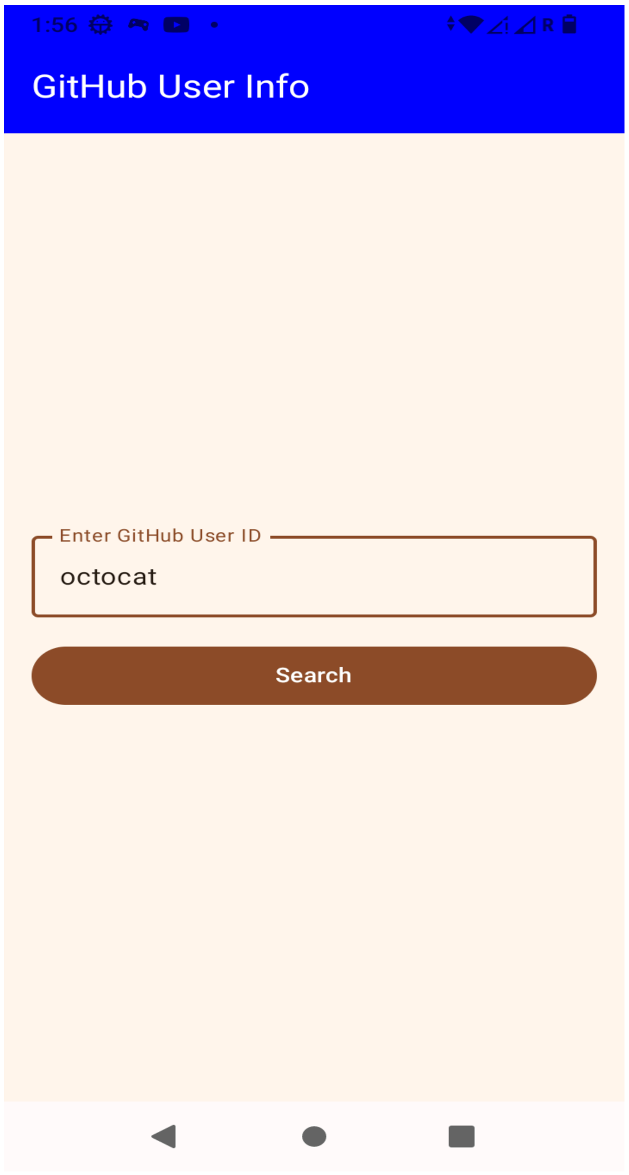
- **Android Studio** (latest version)
- **Android SDK** (API level 21 or above)

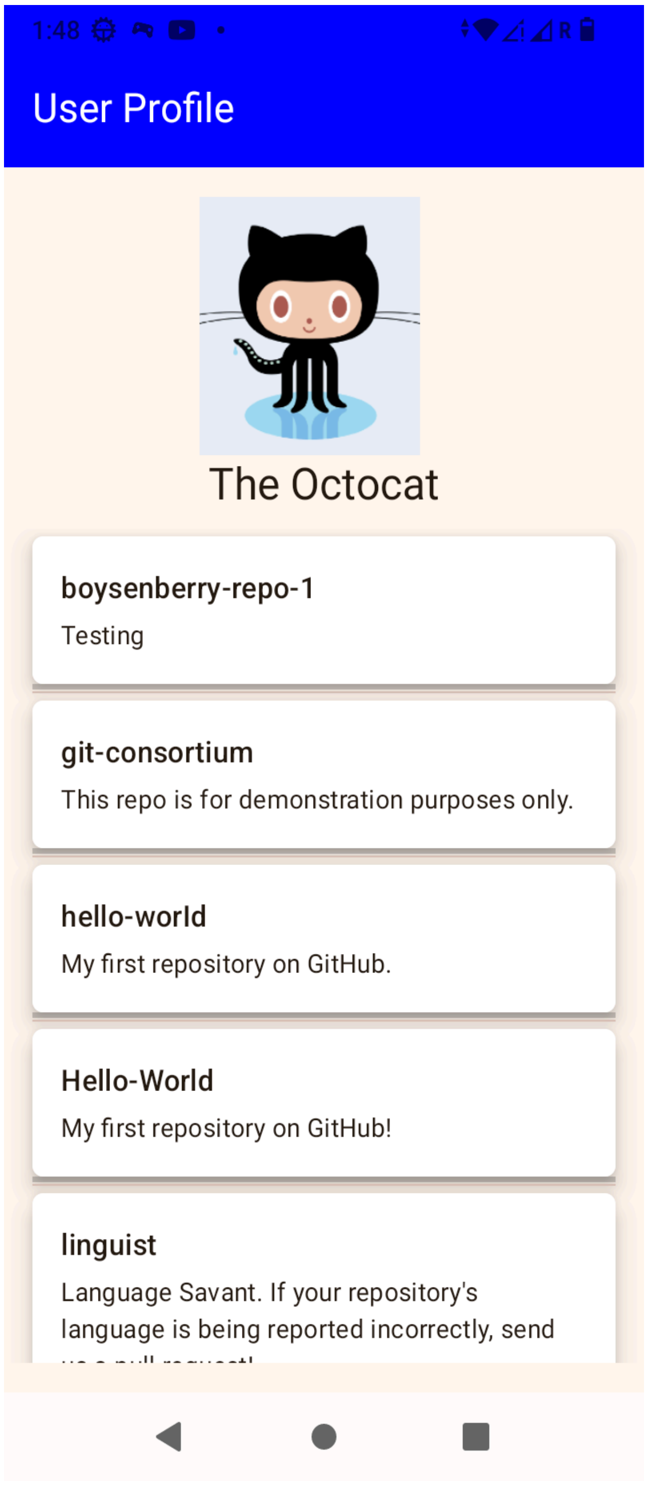
7.2 Steps to Run

1. **Clone or download the project**.
2. **Open the project** in Android Studio.
3. **Sync Gradle** to ensure all dependencies are installed.
4. **Run the project** on an Android emulator or physical device.
5. **Enter a GitHub username** (e.g., octocat) and view the user’s profile and repositories.

8. Application Screenshot:







User Profile

The Octocat

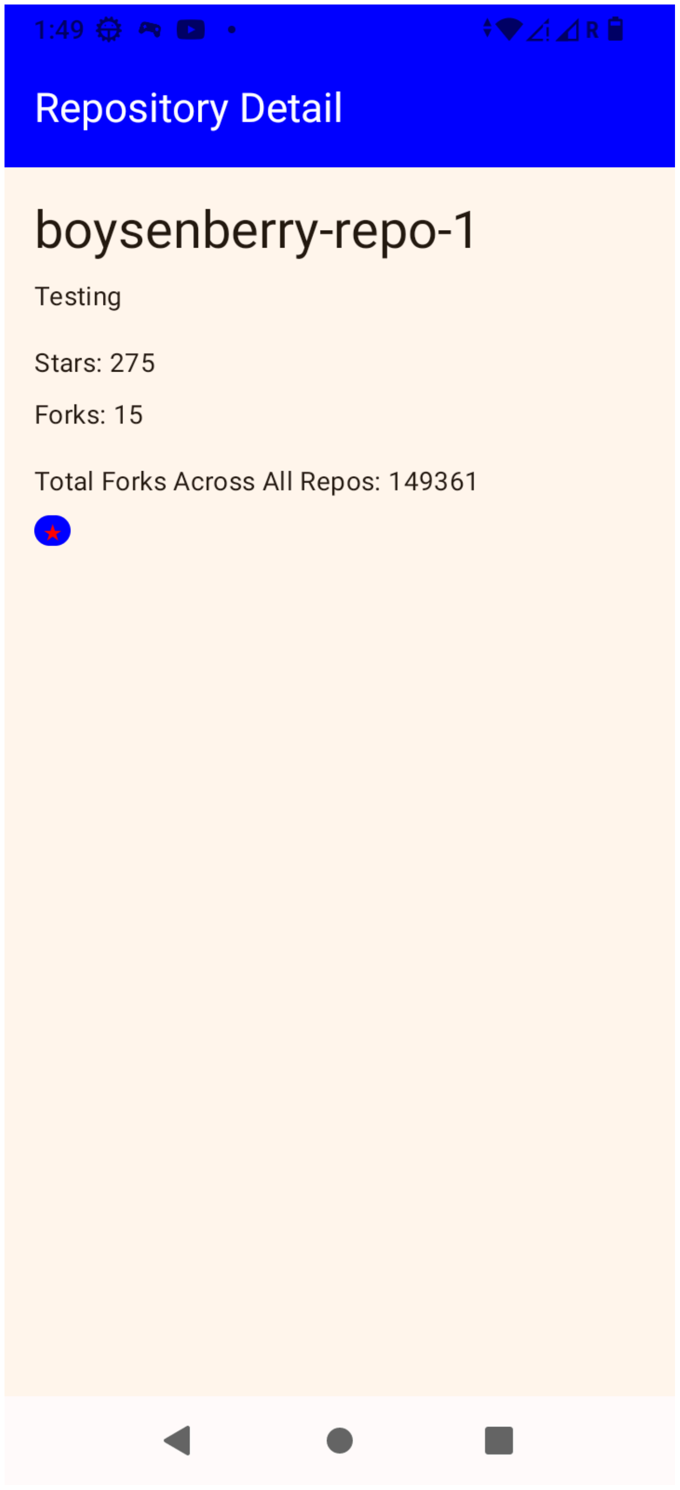
boysenberry-repo-1
Testing

git-consortium
This repo is for demonstration purposes only.

hello-world
My first repository on GitHub.

Hello-World
My first repository on GitHub!

linguist
Language Savant. If your repository's language is being reported incorrectly, send



Repository Detail

boysenberry-repo-1

Testing

Stars: 275

Forks: 15

Total Forks Across All Repos: 149361