

Reddit Android App for YourParkingSpace

by Rafal Ozog

A demo app presenting architecture based on MVVM, Repository and Interactor patterns - including Dependency Injection implementation.

Features

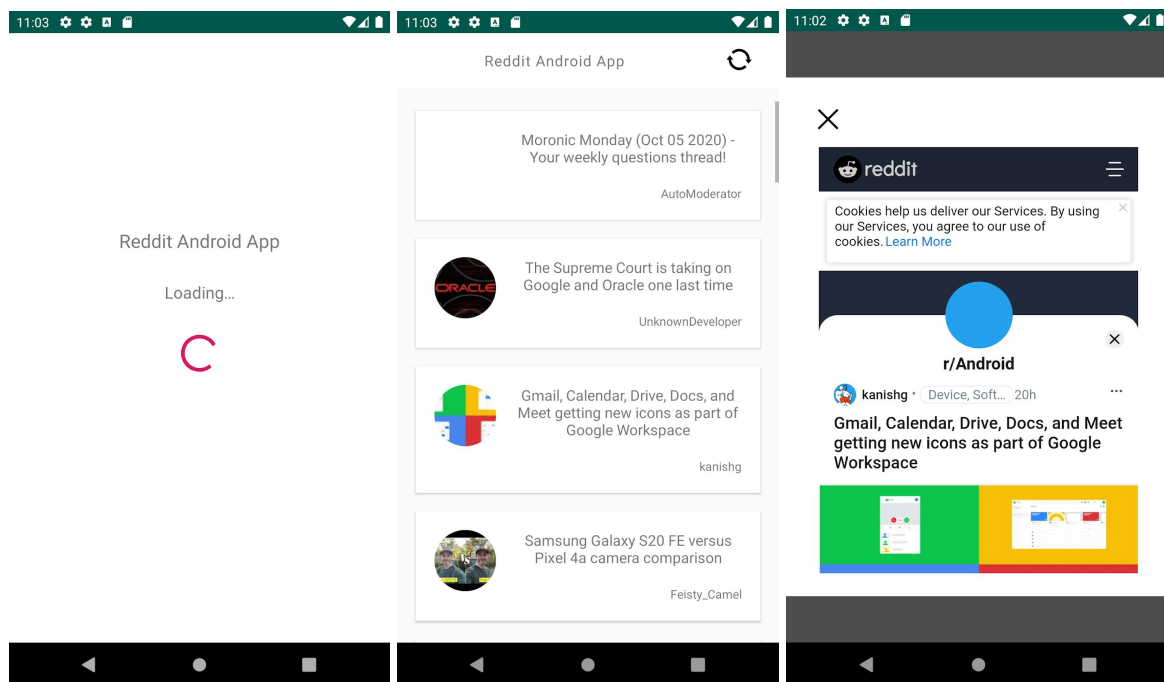
1. Main feed view.
2. Detailed view displaying the full article (using WebView) - when the user clicks on the list item.
3. Loading/ Error/ Success states handling.
4. Offline caching (Room library).
5. Pagination.
6. Refresh (button).
7. Testing (Unit and UI tests).

Architecture

1. Architecture of the app has been based on the MVVM design pattern.
2. Each view consists of Activity/ Fragment (View layer) and its appropriate ViewModel.
3. Model layer (data) has been organized using the Repository design pattern. It's the only gate for data access from viewmodels' perspective.
4. Repository uses Interactors to communicate with API (Network Interactor) and with internal Room database (DataBase Interactor).
5. Communication with the Back-End has been constructed using Retrofit library (connection with the external API, and proper data flow).

6. Internal database (caching) has been based on the Room library.
7. To make the app maintainable and testable I have decided to use Dagger2 library, providing one-way direction of injecting dependencies. In this situation we can be sure that Activity contains ViewModel, but ViewModel doesn't know about Activity. Similarly ViewModel contains Repository, Repository contains Interactors. All of these dependencies are injected dynamically.
8. Code of the app has been organized into 3 main directories - Injection (dependency injection files), Data (model layer files), and Features (having each view and its related files stored as separate sub-direction).
9. Testing part of the app consists of Unit and UI Testing. Unit Tests have been written for each architectural layer (viewmodels, repository, interactors) using JUnit and Mockito libraries. UI Tests are created using the Espresso framework.

Screenshots



Please do not hesitate to ask in case of any further questions. I will be happy to clarify each uncertainty.

I hope you like my app.