

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Key Considerations](#)

[Next Steps: Required Tasks](#)

GitHub Username: [soenkek](#)

RedditViewer

Description

RedditViewer is the app for Reddit fans.

It helps to keep track of all the subreddits you're interested in. You can subscribe to your favourite topics and RedditViewer shows you recent posts.

Intended User

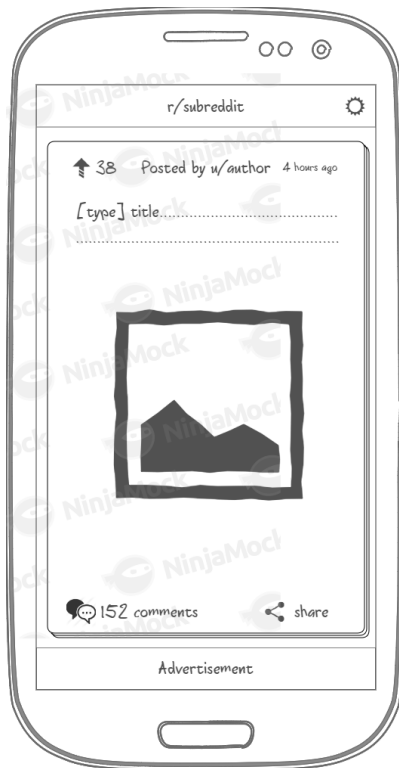
Reddit users who subscribed to many subreddits and users that want to be updated with recent posts on the go.

Features

- Follow subreddits
- Display posts in app and widget

User Interface Mocks

Main Screen



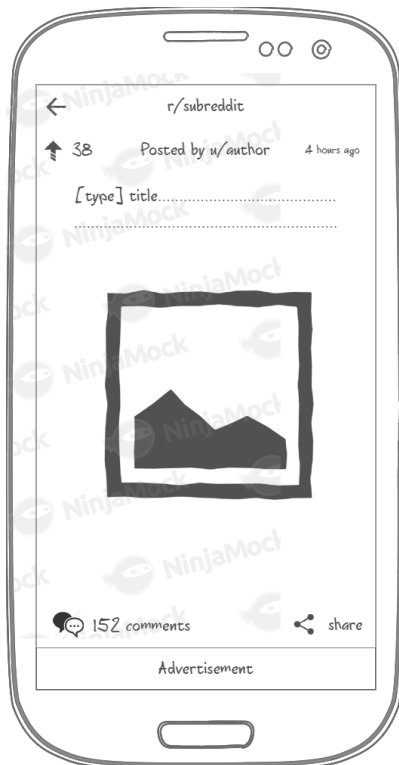
The main screen is the entry point of the app. Posts from subscribed subreddits are shown here. The user can swipe the card from the screen to see a new post. After a click on the gear wheel icon in the top right corner the subscriptions screen is shown.

Subscriptions



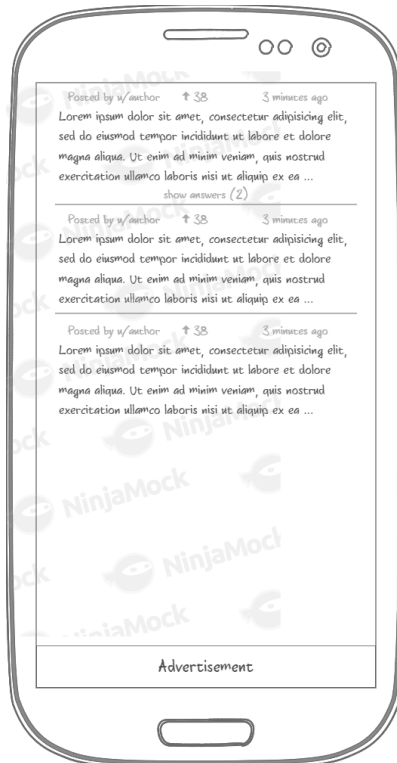
On the subscriptions screen the user can search for subreddits to subscribe to and unsubscribe from subreddits.

Post Detail



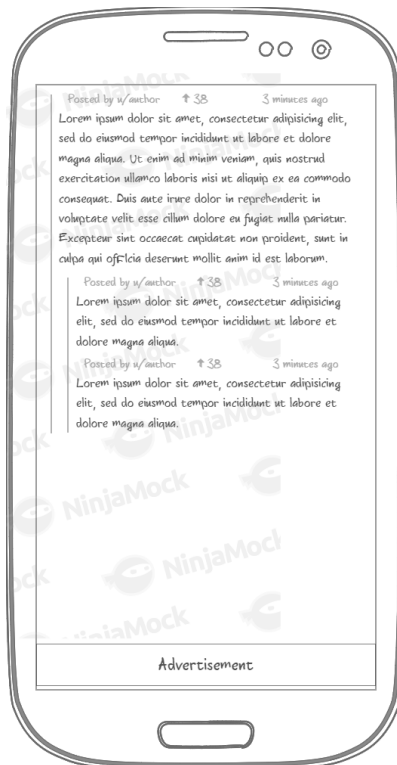
When the card with a post on the main screen is clicked the card expands to the full screen and the post detail screen is shown. It displays the full post and is scrollable.

Comments Preview



After a click on the comments icon or label of a post the comments preview screen is shown. It displays a scrollable preview of all comments. To show text that is cut off and answers the user can tap the comment or “show answers”-label.

Comment Detail



The comment detail screen displays the full comment and answers to it.

Key Considerations

Data persistence

The names of the subscribed subreddits and a few posts for offline usage will be stored in two separate tables in a SQLite database and accessed with the help of a Content Provider.

Libraries

Picasso will be used to load and cache images.

Google Play Services and other external services

The app will use AdMob to display ads to the user and Drive to save the names of the subscribed subreddits so that the subscriptions can be restored.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries and dependencies
- Get API key
- Create custom objects to hold data

Task 2: User Interface

- Build UIs for all Activities
- Build UI for Widget

Task 3: API

- Implement API requests
- Convert JSON result to custom objects

Task 4: Database

- Set up database and Content Provider
- Set up helper methods

Task 5: Display Data

- Load data from db/network
- Populate views
- Create widget

Task 6: Advertisement

- Display test ad

Task 7: BackUp Subscriptions

- Connect to drive
- Save to/load from drive

Task 8: Design & Animation

- Implement Material Design
- Add transitions

Task 9: APK Signing

- Sign apk