

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)□

**GitHub Username:** yadakshay

# AppForReddit

## Description

App displays Articles from different subreddits subscribed by the user. User can choose which of the subreddits are displayed on the main screen. And interact (read or dismiss to fetch a new one) with the articles from each subreddit.

## Intended User

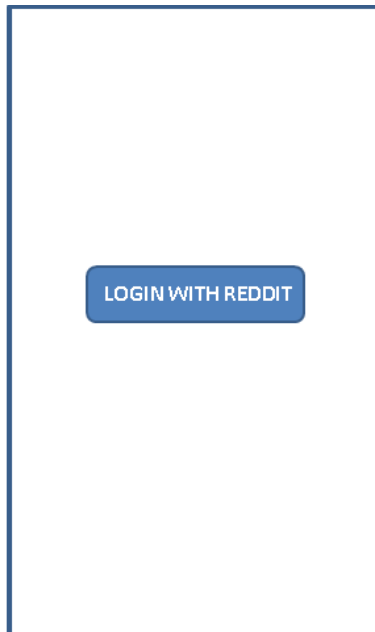
Reddit users who need feed from each of their subscribed subreddits equally.

## Features

- Gets access to retrieve data from users reddit account (using oAuth).
- Stores users subreddit preferences in offline database
- Able to show media content

# User Interface Mocks

## Screen 1



The first screen on launch will contain only a single button to login to reddit.

## Screen 2



After login the main screen will be as shown above. It will have all the topics the user has chosen to get articles from in for of a list of card views. The card will have a 'x' button to dismiss the article in the subreddit after the x is hit the article will be replaced by another article. When user hits the FAB button he will be navigated to screen 3.

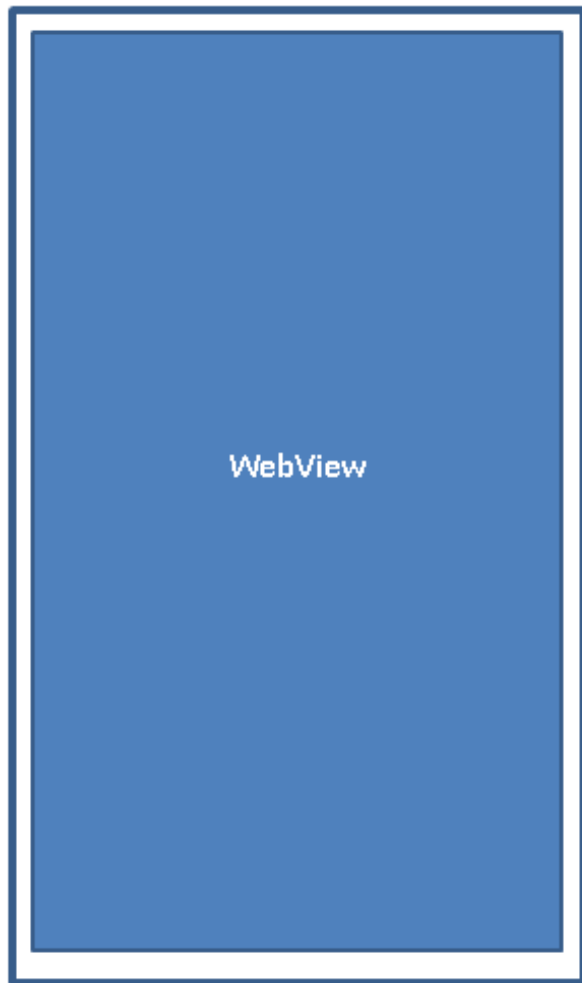
### Screen 3



The screen will be shown when the user hits the FAB button on the screen 2. It will display all the subreddits the user has subscribed to. Each subreddit topic will have a button in front of it using which the user can toggle the subreddit to get displayed on the main screen(screen 2). The toggle button will change the database entry of the subreddit in the local database

### Screen 4

The 4th screen will be a web view to display the media content of the subreddit of the user clicks on the card view in screen 2



## Key Considerations

**How will your app handle data persistence?**

I will build a local sqlite data base with a content provider. The Db will have a single table with two fields the subredditId and its corresponding state (Show or hide).  
Use Shared preference to Save the login session state.

**Describe any edge or corner cases in the UX.**

If the user is not subscribed to any subreddit or if he turns all the toggle buttons to hide the subreddits.

**Describe any libraries you'll be using and share your reasoning for including them.**

Picasso: easier to load images in ImageViews  
okHttp: to make network calls easier.

**Describe how you will implement Google Play Services or other external services.**

Might use JobScedular to refresh the access token which expires in an hour  
Also to update the subreddits database if the user has subscribed to any new subreddit

## **Next Steps: Required Tasks**

### **Task 1: Project Setup**

- Implement oAuth and check if you can get an Access Token and make a sample Api call to any endpoint.
- Configure libraries

### **Task 2: Implement UI for Each Activity and Fragment**

- Build UI for Login Page. And implement getting access to token
- Build UI for main page.
- For choose sub reddit page.
- For web view

### **Task 3: Create database and implement content provider**

- Create a database with a single table with two fields Subreddit Id and display subreddit. Set default value of the display subreddit field to show.
- Also customise the subreddit list launched on FAB button to show the list from the database. And implement the toggle switch to change the field in the data base.

### **Task 4: Configure the Main Page**

- Using the subreddit list in the database make Async calls to retrieve data from Api endpoints
- Implement recycler view to display the articles as cards.
- Implement the onclick for the cross button to replace the article by a new one.

### **Task 5: Handle special cases for web view**

- The articles displayed contain various type data or media. Handle cases accordingly in web View

### **Task 6: Implement services**

- Implement Job Schedulers or dispatcher to refresh access tokens and database sync. Handle login session management and error cases.