

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Main Screen - Subreddit/FrontPage Viewer](#)

[Post Detail \(Comments\) Screen](#)

[Key Considerations](#)

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

[Describe any edge or 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 or other external services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Reevaluate MVP and Implement Any Extras](#)

[Task 4: Ensure App is Material](#)

[Task 5: Polish](#)

GitHub Username: [JacquesSmuts](#)

ThresheR

Description

ThresheR is a simple Reddit client app. Like all mobile Reddit clients, the app will allow you to browse the frontpage and various subreddits on Reddit, as well as to log in and submit and read comments. Unlike most other Reddit apps, Thresher will automatically hide content that the user has already seen, based upon the user settings. The idea being to make sure that the user does not see anything he has already seen before.

If the user browses Reddit for 15 minutes, closes the app, and opens it again 30 minutes later, his front page will look completely different, even if the actual Reddit front page does not.

Intended User

ThresheR is an app intended for any regular Reddit user.

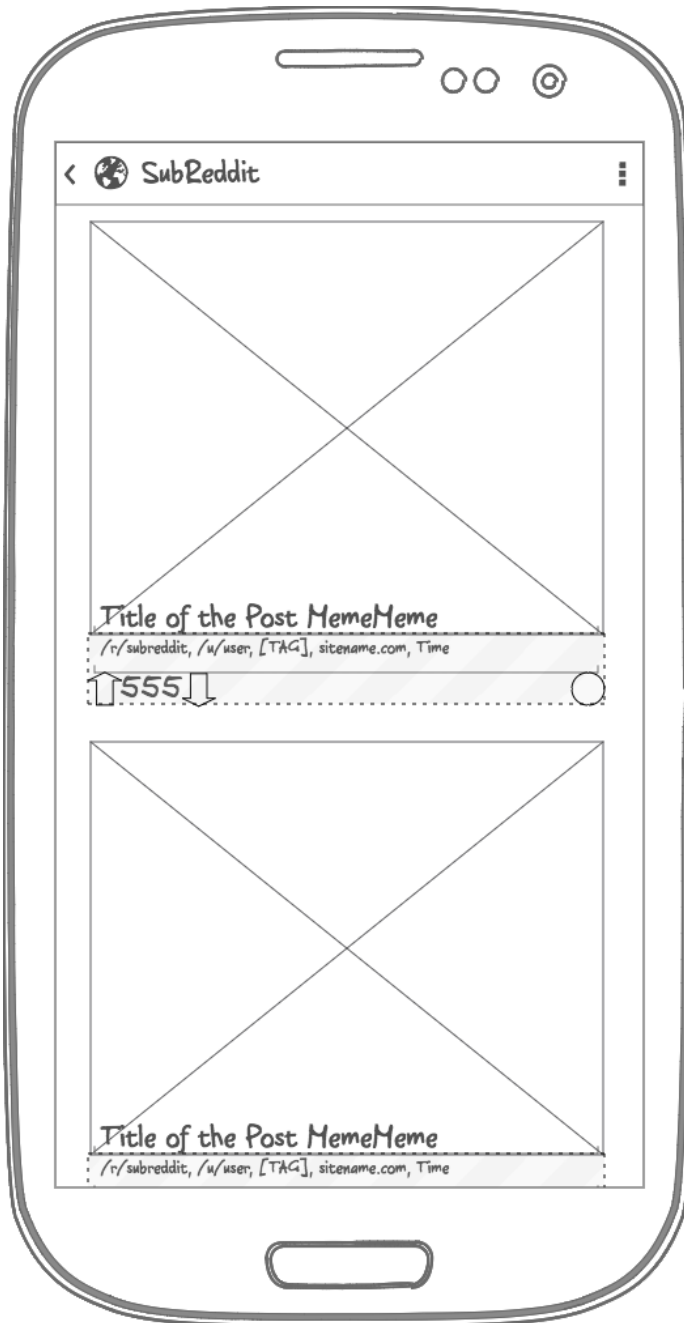
Features

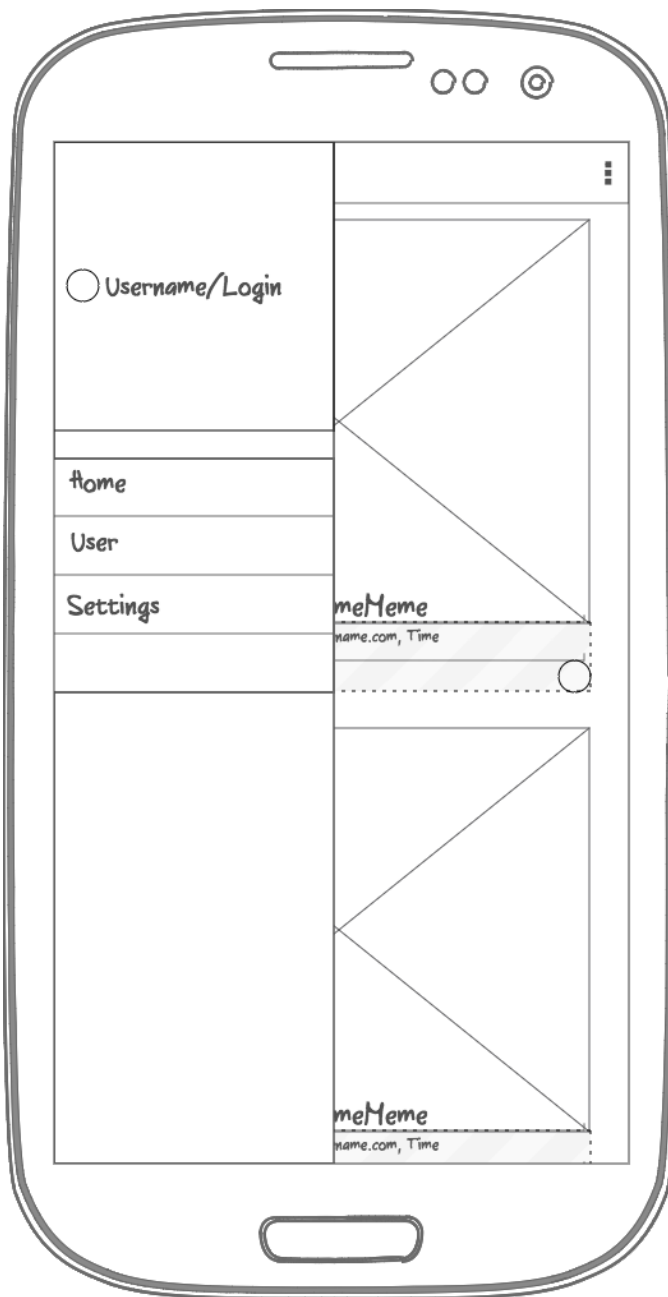
- Browse Reddit
- Make Posts
- Read and Create Comments
- Activate ThresheR mode to automatically hide content you've seen

User Interface Mocks

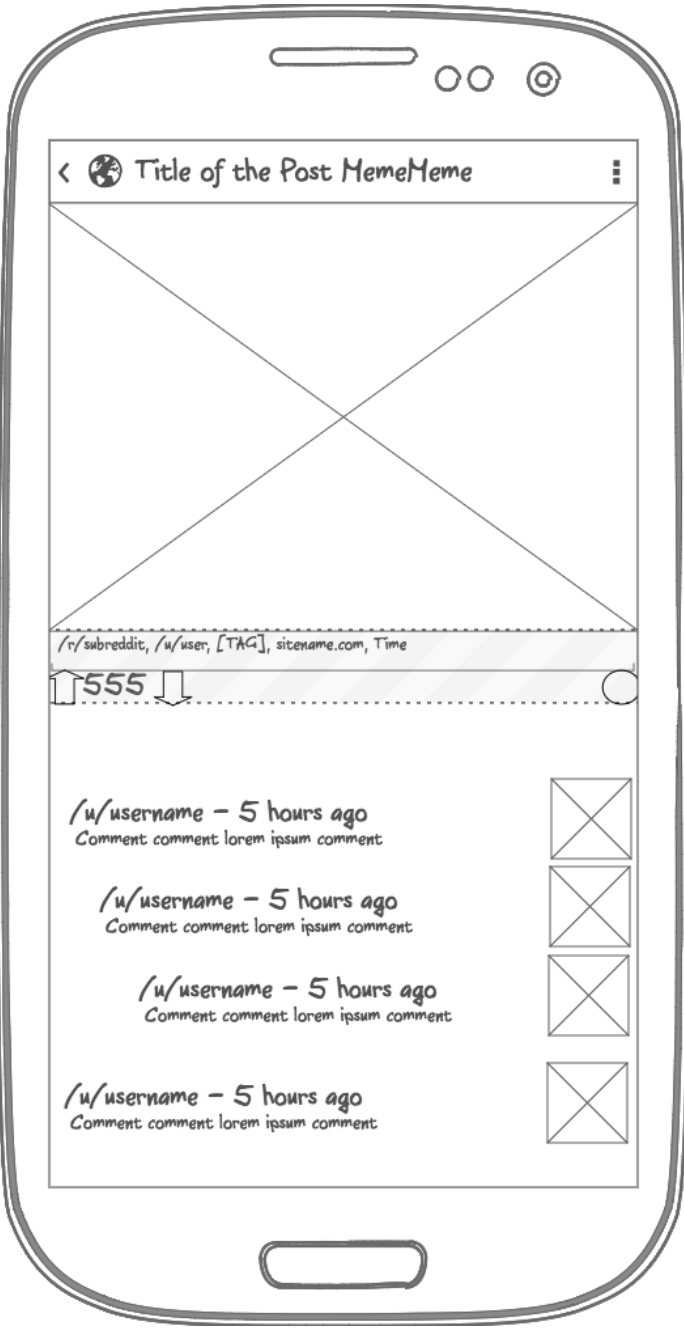
NinjaMock was used to create these Mockups

Main Screen - Subreddit/FrontPage Viewer

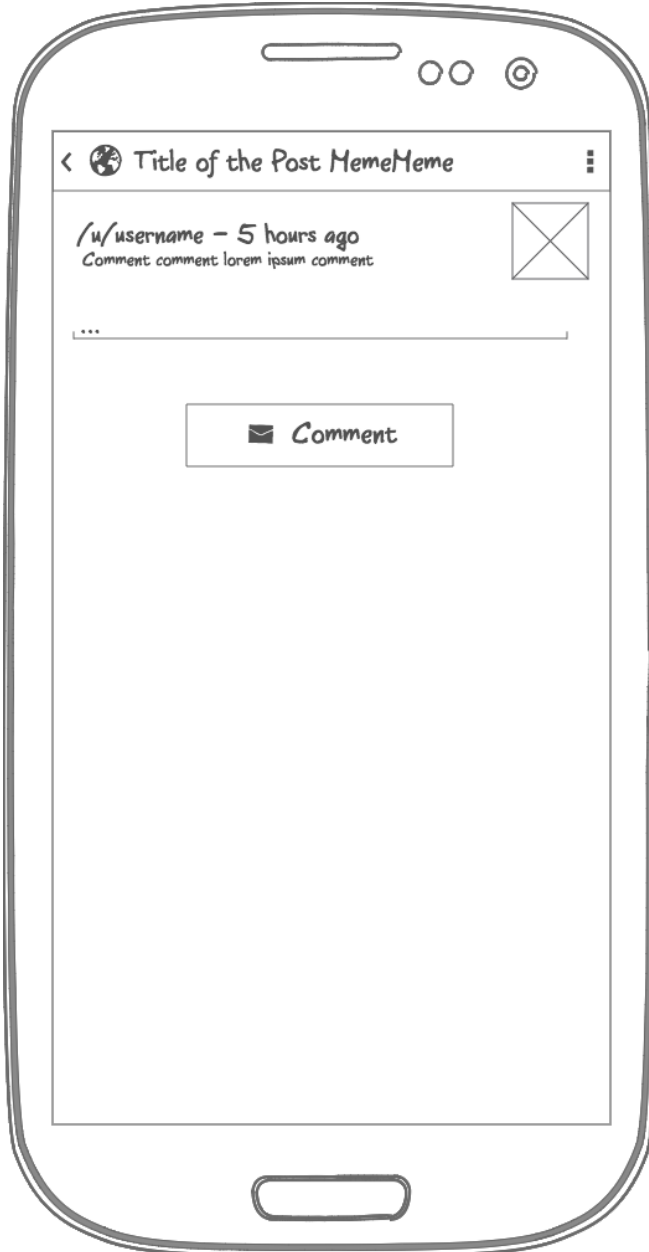




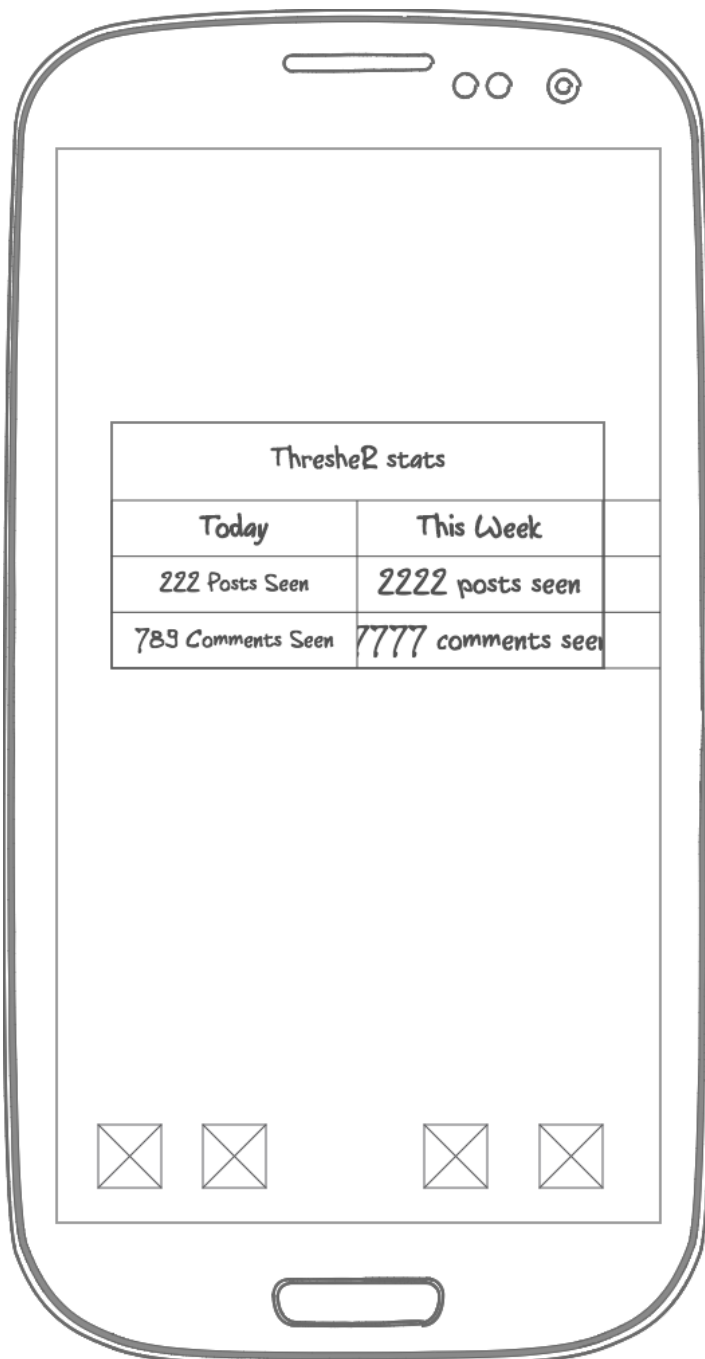
Post Detail (Comments) Screen



Create Comment Screen



Widget



The widget will provide simple statistics for the user, depending on user settings, showing how many posts and comments the user has seen and skipped, for example.

Key Considerations

How will your app handle data persistence?

ThresheR will use a ContentProvider and SQLite to store data locally. Old data will be scrubbed. Proper data management will be very important to the core functionality of that

app. All data will be obtained using AsyncTask, since new data will only be downloaded when the user opens the app. This data will be accessed from all activities using a Loader.

Describe any edge or corner cases in the UX.

The app needs to keep track of whether or not the user is logged in, and what the user settings are for whether or not to show the post that has been loaded.

The app also needs to be able to handle bad connections or failed API calls. A simple Snackbar message will be used to indicate to the user if any API call has failed, with the option to retry where applicable. If the user has data on his phone he can browse what he has, but otherwise an aesthetically pleasing no-content screen must be shown to the user. This screen will show when the user opens the app for the first time and no data is loaded yet.

Furthermore, Reddit has a large variety of valid posts. Html links, videos, images, gifs, comment links and self posts. The app must be able to handle all of these.

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

Butterknife, for simple Dependency Injection

Picasso, for showing Images

Various Android Support Libraries for the basic AppCompatActivity components

Espresso, for testing

OkHttp, for network calls

GSON, for JSON-to-POJO conversions

Material-Dialogs, for simple Material-Design popups, where necessary.

Crashlytics, to log crashes during alpha testing

LeakCanary, for Leaks

Timber, for easier logging.

IcePick, for easier savedInstanceState management

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

The app will use Google Location Services to determine the country of the user to provide localized Reddit content to the user.

The app will use Google Analytics to track user app usage.

Next Steps: Required Tasks

Task 1: Project Setup

The Project Setup will take place with the following steps, more or less in order:

- Create Initial Project
- Connect to GitHub
- Set up libraries
- Set up Models of Reddit POJOs
- Set up Content Provider and Database Schema
- Receive data through Reddit API

Task 2: Implement UI for Each Activity and Fragment

For this part, the app must be a MVP which can view subreddits, posts and comments, as well as login and make comments.

- Create a FeedViewerActivity
 - Set up RecyclerView in FeedViewerActivity, with ViewModels to display Content
 - Create Spinner to switch between subreddits/frontpage
 - Create Spinner to switch between What's Hot/New/Top/Rising
- Display a post detail Activity for comments
 - Set up a nested RecyclerView to display comments
- Allow user to register/login
 - Create an appropriate sidebar in the Main Activity to house User Details, Settings, Messages
- Create SettingsActivity
 - Set up settings for filtering and any other settings
- Create CommentActivity
 - And send api call if user leaves a comment

Task 3: Implement Widget

- Create an AppWidgetProvider and Widget Layout
- Create IntentService for Widget
 - User can click the Widget to cycle through different stats and timeframes
 - Widget will display to latest stats every hour, or when the user opens the app.

Task 4: Reevaluate MVP and Implement Any Extras

This part requires re-assessing which libraries are required, and make sure anything missing is spotted.

- Implement Google Location
- Implement Google Analytics
- Implement Espresso and build missing Tests
- Implement any additional required libraries and classes
- Make sure Tablet and landscape and RTL layouts are working.
- Make sure orientation changes are handled

I am currently researching RxJava, and I may consider adding it at this point.

Task 5: Ensure App is Material

This step is to make sure the app conforms to all of the Material Design guidelines.

- All layouts must have correct Elevation specified if it doesn't have a default
- All transitions must be as smooth and Material as possible
- Add animations where it makes sense

Task 6: Polish

This part is to ensure overall polish. Any glitches, bugs, potential crashes and suboptimal performance must be ironed out here.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

