How to Use this Template
1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions
1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: steveburns

Simplify for Reddit

Description

Simplify for Reddit simplifies the time and effort it takes a Reddit user to keep up with many subreddit subscriptions without missing the posts they are most interested in.

Intended User

Simplify for Reddit is for Reddit users who subscribe to a large number of subreddits and are finding it tough keeping up with all of their subreddit subscriptions and don't want to be constantly tied to it.
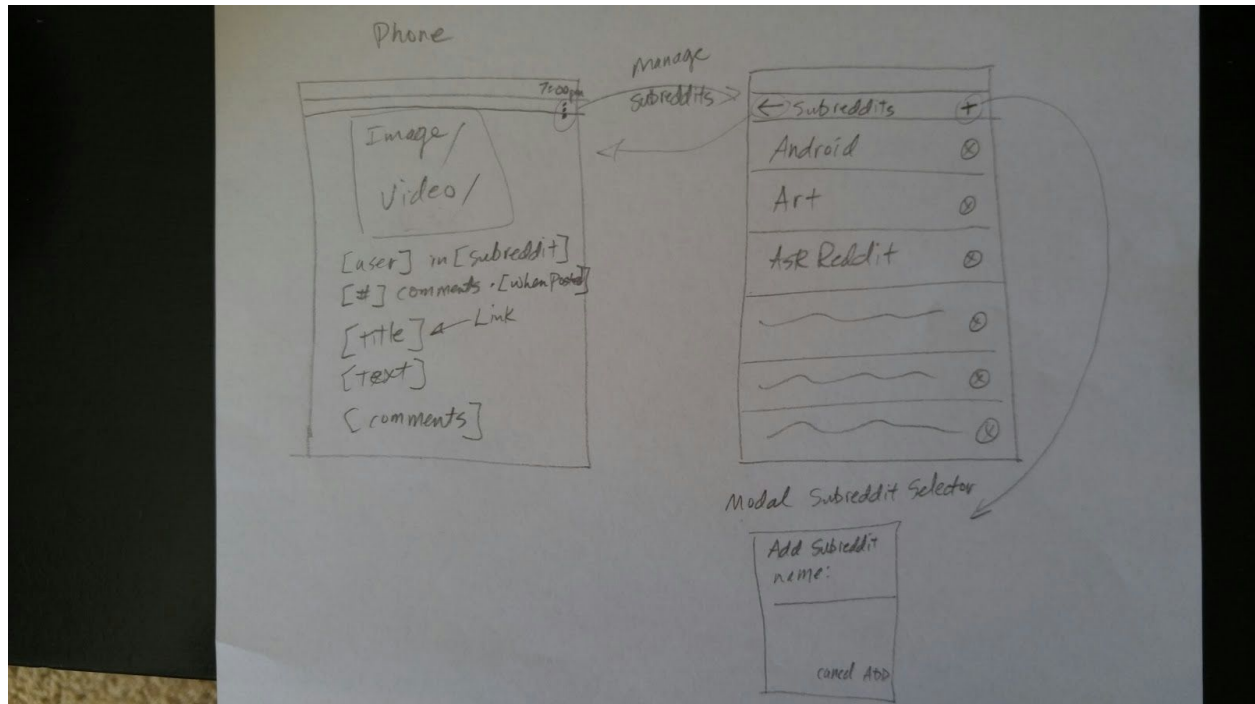
Features

List the main features of the app:
- The app can access Reddit APIs without requiring the user to have an account so I do NOT plan to prompt for or login with user credentials.
- The app will come pre-installed with a default list of "safe for work" subreddits that have wide user appeal, therefore, as soon as the app starts up it will go right into displaying posts.
- The app will allow the user to modify the list of subreddits for which posts will be displayed.
- The app will randomly present the user with a post from one of the subreddits they have selected displaying image/video, comments, title, text, etc.
- Users can interact with the post by viewing an image, a video, or displaying as web content.
- When the user dismisses the randomly selected post by swiping, they will receive another.
- The app will use other apps, components, and services to display web content, photos, videos, etc as needed.
- The app will show a small ad banner in the free version.
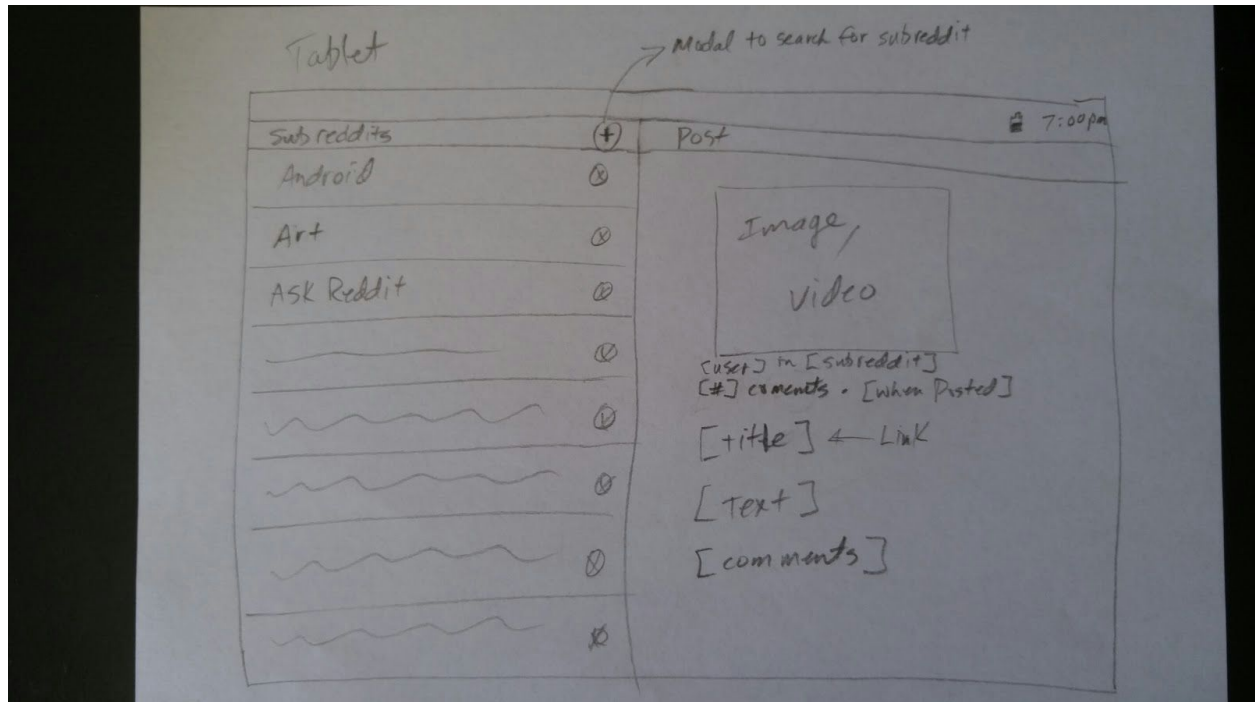
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1

Hopefully what I'm proposing here makes sense. Since the app will come pre-installed with a list of default subreddits, as soon as the user runs the app, they will see the detail you see on the left, which shows lots of information about the current randomly selected post. If the Image is a photo then clicking on it displays it full screen, but if it's a video then it is played. Clicking on anything else shows the post in the user's favorite web content viewing app. Clicking on an icon in the tab bar will display the list of subreddits they are currently subscribed to as shown in the view on the right. Clicking the "+" button displays a model that lets them search for a subreddit that they want to add to their list. Clicking the "x" button on a subreddit removes it from the list. One thing I forgot to show is the ad banner at the bottom of the detail view for the "free" app flavor.

Screen 2

On tablets, users will see the same info, but with the list of subreddits on the left where they can easily edit them by clicking the "x" to remove or the "+" to use the "add subreddit" modal. Since this isn't a real master/detail design, I've wondered if it might make more sense to place the subreddit list on the right as well as make its width take up less than half the screen width. One thing I forgot to show is the ad banner at the bottom of the detail view for the "free" app flavor.

Key Considerations

How will your app handle data persistence?

The app comes pre-installed with a set of subreddits that the user can modify so it only needs to be concerned with changes to that list as well as getting the data for the next randomly selected post. I definitely plan to use a Content Provider for this app for all access to the data. I don't have all those details worked out yet, but my initial thinking is that the app will need to pre-fetch the next couple of randomly selected posts so the app is very responsive when the user swipes to see the next post.

Describe any corner cases in the UX.

Some corner cases I've thought of are:
1. If the user changes the list of subreddits then what happens to the pre-fetched random post selections?
2. How will the app guarantee that it doesn't display the same post twice?
3. How does the user return to the exact location in a running video when the back button is pressed or the device is rotated?
4. The first time the user runs the app they may not know that they need to swipe to dismiss the current post to see the next post. I need to implement the standard way of showing the user how to do this (some bubble text or something).

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

Picasso to handle the loading and caching of images.
Plan to use a third party library to interface with Reddit from this list of choices:
- https://github.com/jReddit/jReddit
- https://github.com/thatJavaNerd/JRAW

Use Retrofit or Volley to access Reddit endpoints unless the third party Reddit library has one.
Reddit Application Only OAuth (see https://github.com/reddit/reddit/wiki/OAuth2#application-only-oauth)
AdMob to display ads in the free version.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Steps to setup and/or configure this project:
- Setup Reddit application only OAuth
- Develop database schema
- Create content provider
- Research third party Reddit libraries, make a selection, configure in app

Task 2: Implement UI for Each Activity and Fragment

Tasks:
- Build UI for Main Activity (post detail activity)
  - Create layout
  - Interface with Content Provider
- Build UI for subreddit list
  - Create layout
  - Interface with Content Provider
- Build UI for "add subreddit" modal
  - Create layout
  - Interface with Content Provider

Task 3: Free App Flavor

Setup free app flavor:
- Create free flavor of app
- Configure AdMob
- Implement AdMob in app

Task 4: Tests

Create tests:
- I know I want to have some unit tests in this project and so I plan to have at least a few for the content provider

Task 5: Release

Release steps:
- Create key
- Create signed APK
- Put on Google Play

Submission Instructions
1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"