

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[RSS Feed Search](#)

[All Feeds Screen](#)

[The Article View screen](#)

[The Widget](#)

[Key Considerations](#)

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

[How will your app handle authentication?](#)

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

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

[Next Steps: Required Tasks](#)

[Task 1: Setup Google Plus OAuth](#)

[Task 2: Mirror the server database within the app and setup Content Provider](#)

[Task 3: Set up searching with Feedly RSS Search](#)

[Task 4: Refresh feeds](#)

[Task 5: Load articles](#)

[Task 6: Ordering feeds, Deleting feeds and logging out](#)

[Task 7: Allow re-populating an existing user's feeds from remote server](#)

[Task 8 : Create widget](#)

[Task 9 : \(Optional\) Optimize for tablet](#)

# Sprocket News

## Description

If a psychologist asked me what word I most associate with widget, I would say "Sprocket!". That's what my RSS newsreader application will be about. Bringing a user's favorite articles from their favorite news sites straight to their home screens.

I had something like this before, but my favorite Android app was sold and retired! I want to bring it back and make it better than ever.

Please read the following excerpt from [a user on Reddit.com](#) as they explain the state of things:

“For a while now I've used Pulse as my news reader for work. It's not the best RSS reader in the world, but it does one thing I find essential: displaying sources and articles in rows so I can quickly scan the headlines by feed. None of this newspaper/magazine layout nonsense. LinkedIn, being the destroyers of everything good, decided to abandon the version of Pulse with iPad support in favor of a useless iPhone app.”

Although this Reddit user is lamenting the demise of the IOS app, I too lament the demise of the Android one. Years ago I fell in love with the “Pulse” newsreader application. It was one of the early Android applications that never felt old, and was the best at what it did. Aside from the great grid layout, the next thing I loved about it was the program flow.

It was a little unconventional, because pressing the app's back button after reading the story would bring you back from your story into the application, even if you launched the application from a widget. I would like to offer this back button functionality as an opt-in configuration option. I will also offer a button in the title bar which will bring the user back to their “all feeds” view. This “all feeds” main screen layout, as well as the widget layouts, were my favorite for RSS readers, and no other RSS reader has worked as well.

## Intended User

In a nutshell, People looking for a grid-based RSS reader with grid-based widget options as well. Initially, I will be testing and setting up the feeds around my favorite tech feeds.

## Features

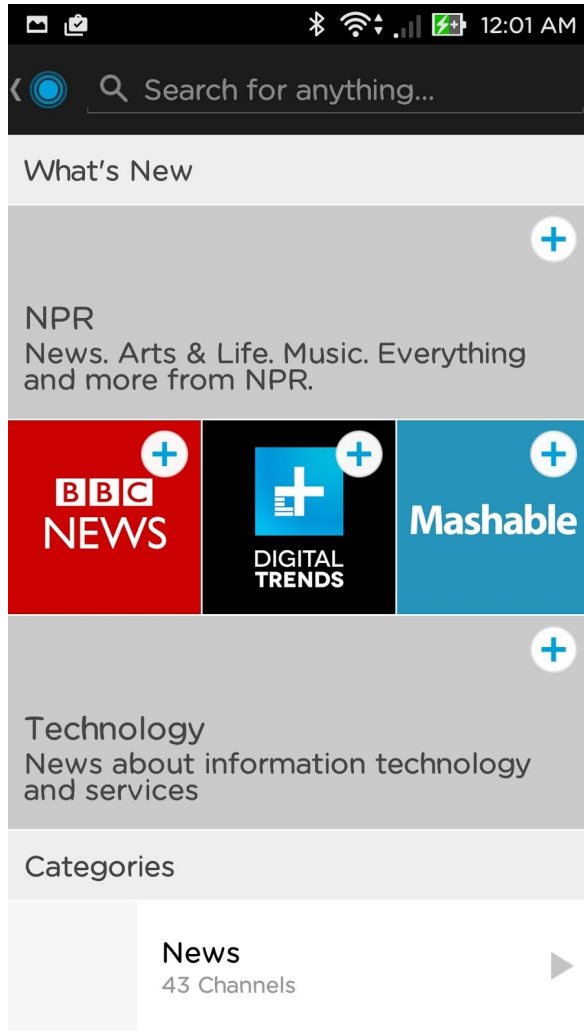
List the main features of your app. For example:

- Uses grid-style views for displaying RSS feeds
- RSS feed searching and configuring using Feedly
- Records the articles a user has already viewed on device and on server
- Offers a filter for feeds to display read/unread/all articles
- Opt-in back button configuration to bring you back to ALL FEEDS screen
- Uses logos for feeds, thumbed images for rss articles
- Offers a configurable widget to pick between RSS feeds being followed
- (future/optional) tablet optimized layout
- (future/optional) article tag analysis - find a user's favorite subjects, highlight unread stories on these subjects
- (future/optional) mixed/aggregated feed option
- (future/optional) article viewing history and favorited articles

## User Interface Mocks

I will use screenshots from the old, discontinued version of pulse for this section.

### RSS Feed Search



This is the retired pulse app's feed search screen. My app will have a search area in the title bar, but I don't plan to have categories or a what's new section. I'll display search results ideally with an icon and a title and description of the feed, in a vertical list.

## All Feeds Screen



The all my feeds screen shows all of the feeds which you're following. They will have been found from the feed search.

## The Article View screen

I don't have a mock for this but it will be a simple WebView with options to:

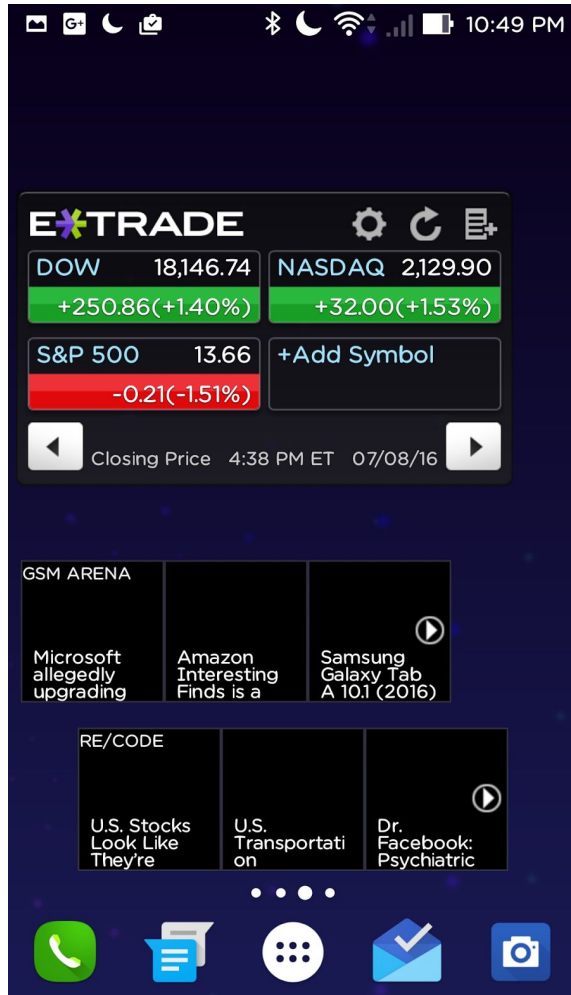
- favorite the article
- share the article
- go back to the all feeds screen

The article view can be launched from the all feeds screen, or from the widget.

By default the back button will bring you to the previous page in the article. If you launched the article from the widget, pressing back will exit. However, there will be a configuration option that can be set to have the back button bring you back to all feeds after launching from the widget.

This behavior matches what Pulse did. I enjoyed this because many times I would press back after reading a story, but I enjoyed being presented with other articles to read.

## The Widget



Please see above the Widgets with the text "GSM Arena" and "RE/CODE". The widget will allow you to follow unread stories from any of your followed feeds on your desktop. Before the application was deprecated, there were preview images for the stories in each box.

## Key Considerations

### How will your app handle data persistence?

- Use a PHP API through free host X10Hosting - I have lots of experience with PHP/MySQL so this is the most comfortable way for me to create an API for my app.

## How will your app handle authentication?

- Google play services auth.

## Describe any corner cases in the UX.

- Configurable back button: option to match old Pulse application/widget behavior.
- Saving state: App should remember position on page for article, and all feeds.

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

- Glide - to handle the loading and caching of article preview thumbs.
- Data Binding **without** butterknife - follow new Google Developer guidelines
- RSS Search API - Feedly - Only for search endpoint
- GSON - with X10Hosting server
- OkHTTP - with X10Hosting server
- Not a library, but, use RecyclerView for horizontal scrolling
- Google Play Services Auth - setup auth with Google for account management

## 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: Setup Google Plus OAuth

<https://developers.google.com/identity/sign-in/android/start-integrating>

Only offer Google Plus Authentication at the beginning:

- Follow the guide at the link
- Use the PHP server already setup at X10Hosting

### Task 2: Mirror the server database within the app and setup Content Provider

- Transfer MySQL schema to SQLite schema
- Setup Content provider
- Setup retrieval for common data requests

### **Task 3: Set up searching with Feedly RSS Search**

- Create a material design actionbar that turns into a searchbar
- Use OkHTTP to send a web request to Feedly search
- Parse the search results successfully

### **Task 4: Display search results**

- Display search results along with preview image
- Allow the saving of feeds to the phone and to the server

### **Task 4: Refresh feeds**

- Refresh feeds and save the articles locally
- Display Title, Author, Preview Image using horizontally scrolling recyclerview

### **Task 5: Load articles**

- Use WebView to load articles. Save article load and click time to local database and remote database.
- Save “leaving the WebView” event time to local and remote database
- Set up share intent for articles in webview actionbar
- Handle orientation changes for this fragment

### **Task 6: Ordering feeds, Deleting feeds and logging out**

- Allow user to edit their “all feeds” screen, allow re-ordering feeds
- Allow the user to delete feeds locally, mark as deleted remotely
- Allow user to logout and login again
- Handle orientation changes for this fragment

### **Task 7: Allow re-populating an existing user’s feeds from remote server**

- Server recognizes existing user
- App downloads saved feeds

### **Task 8 : Create widget**

- Create a per-feed widget as illustrated
- Allow customization of refresh interval and colors when the widget is created, but choose sensible defaults
- Create “scrolling” buttons to go back to older articles
- (optionally) create filter button to hide/show already read articles
- Back button from article returns you to all feeds display

## Task 9 : (Optional) Optimize for tablet

- Optimize widget resizing for larger display
- Optimize app with master/detail view
- Orientation changes

---

### 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**"