Capstone_Stage1

App Demo: https://pr.to/VMHL34/

GitHub Username: ijzepeda

EZRSS

Description

Simple RSS Reader, Add your favorite RSS link, and keep updated.

Intended User

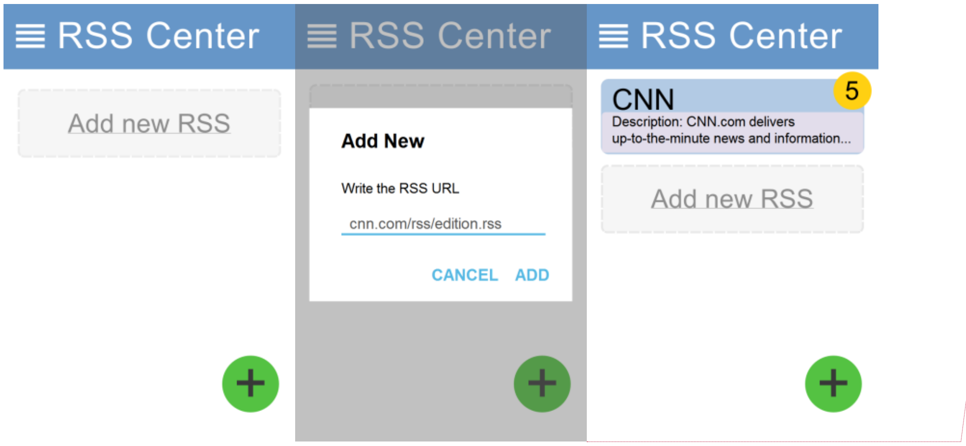
Productive users. Anyone that want to keep updated on their favorite sites, without entering every one of them

Features

- Saves Favorite RSS
- Notifications for new articles/posts.
- Material Design (Recycler View, Palette, Floating Action Button)

User Interface Mocks

Screen 1: MainActivity Site List



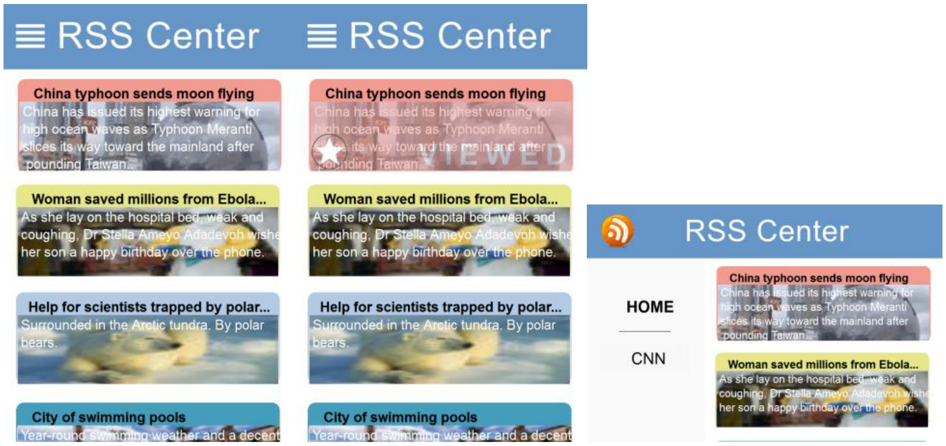
Write the RSS URL

cnn.com/rss/edition.rss

Description: CNN.com delivers up-to-the-minute news and information...

Mainscreen will display a recyclerview listing all saved sites, and A Floating Action Button to Add a new URL.

Screen 2: Site Articles list



China typhoon sends moon flying

China has issued its highest warning for high ocean waves as Typhoon Meranti slices its way toward the mainland after pounding Taiwan...

Woman saved millions from Ebola...

As she lay on the hospital bed, weak and coughing, Dr Stella Ameyo Adadevoh wishe her son a happy birthday over the phone.

Help for scientists trapped by polar...

Surrounded in the Arctic tundra. By polar bears.

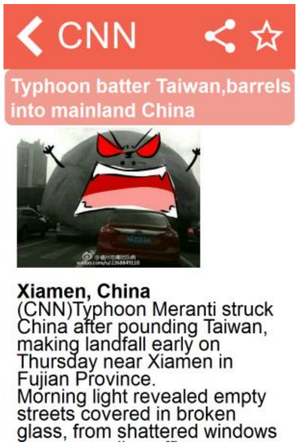
City of swimming pools

Year-round swimming weather and a decent

After clicking a Site you will see all latest posts, displaying title, description and the image. [for Landscape or tablets a master detail layout will be active. Listing the sites and the posts, or posts and details].

Commented [IZ1]:

Screen 3: Post Details



Typhoon batter Taiwan,barrels into mainland China

Xiamen, China
(CNN)Typhoon Meranti struck China after pounding Taiwan, making landfall early on Thursday near Xiamen in Fujian Province.
Morning light revealed empty streets covered in broken glass, from shattered windows

Article details will show the full description of the post, and a direct link to open on browser.

Key Considerations

How will your app handle data persistence?

SQLite, Content Provider.

Describe any corner cases in the UX.

The user will be navigating using direct taps on elements such as cards or actionbar.
Because articles are made from online XML the user will always start/resume the app at the mainactivity.

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

Picaso, OKHTTP.

Describe how you will implement Google Play Services.

AdMob to have different flavors, with and without adds.
Analytics. To keep track of its usage.
GCM: show notifications with new RSS.

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

- Configure libraries
- Create Activities for MainActivity, SiteActivity, DetailsActivity
- Create recyclerviews , adapters and cards
- Utils Class to reuse methods like Network connections. Placing Image with Picasso. Or Palette colors
- Edit the manifest to add required permissions

.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Site Posts
- Build UI for Article Details
- Build UI for MasterDetail for larger resolutions

Task 3: Create Server

Create a Server class to keep updated the RSS posts

- Server will be fetching new RSS, every hour, or selected time in Settings.
- AsyncTask will fetch RSS Everytime the user opens the App
- Make a toast to notify user that it is updated

Task 4: Notifications

- When the server fetch new rss, trigger a PendingIntent to display Notification on the statusbar
- Clicking on Notifications, will pop the Site Activity, if it has been only fetched updates for one site, or Main Activity if there are more post from different sites.

Task 4: Gradle

- Add dependencies for Services, Picasso, OkHttp, and any other needed.
- Create different flavors to add AdMob to a free version.
- Add API KEYS
- Select Appropriate SDK, paths and configuration.

App Demo: https://pr.to/VMHL34/