
GitHub Username: hrajaraman89

BBC News Reader

Description

Get the top stories from BBC right into your app.

Intended User

Anyone who likes to use BBC as a news source.

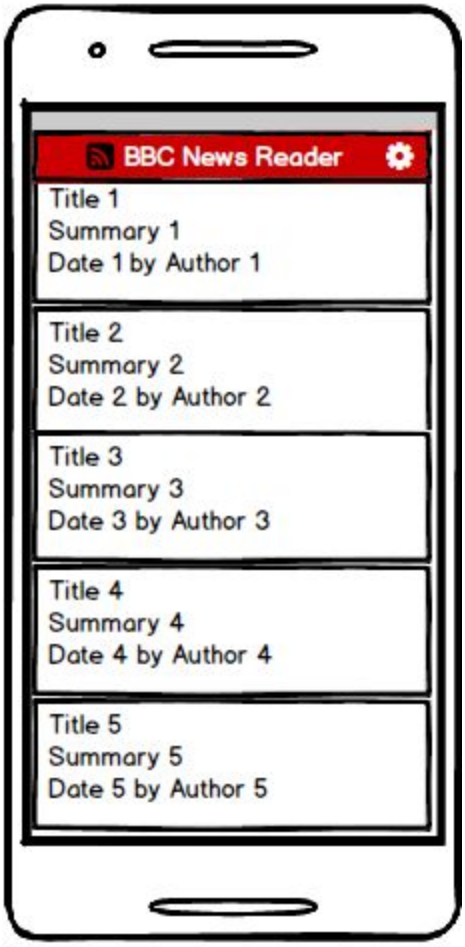
Features

Main features:

- Gets the latest news
- Save news for offline mode
- Alerts on latest news on publish

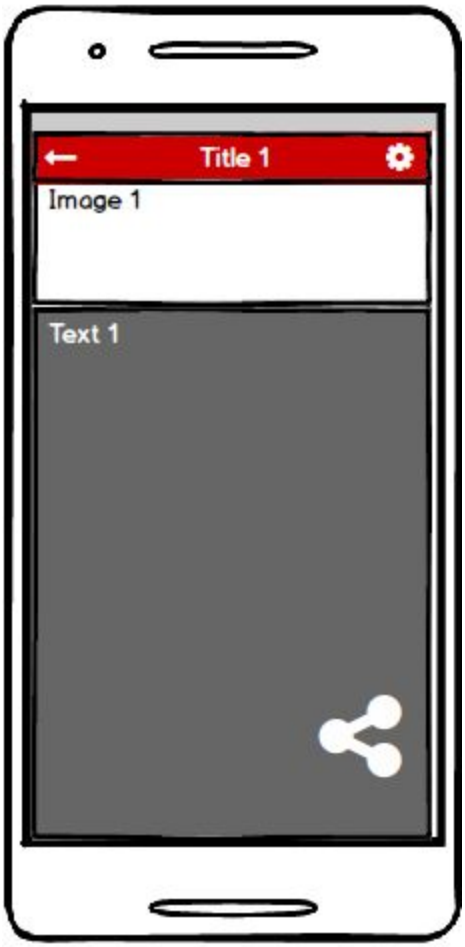
User Interface Mocks

Main Screen



This is the main screen where the users can see the list of articles based on the sources that they have selected. By default, “Top News” will be enabled so that this screen is always populated. User’s can also swipe the card away to dismiss a story. Otherwise, they can click on settings or the story itself

Screen 2

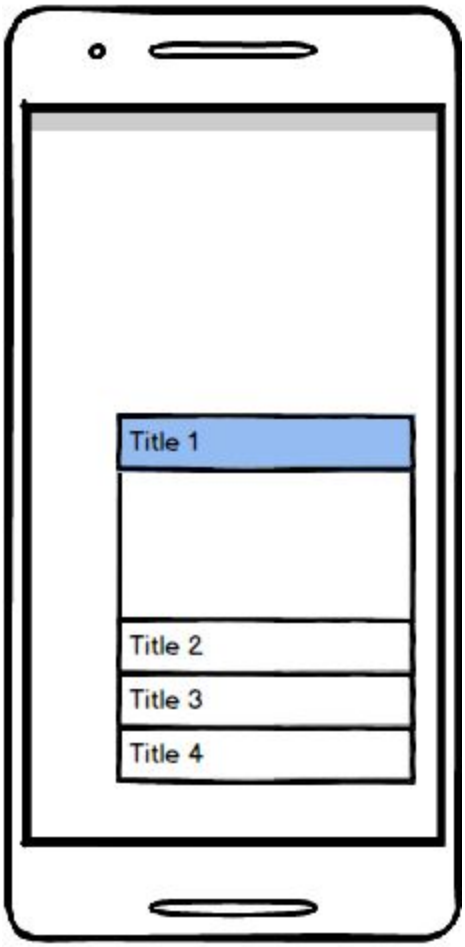


When the user clicks on a card, it will show an immersive view of the article, including the image and the actual content. If there are embedded media within the article, that may also be rendered. User can back out using the arrow. A FAB to share will be prominently displayed for engagement.

Screen 3

The settings screen. Users can select news of their liking, as well as the amount of notifications they would like.

Screen 4 - Widget



This is the widget screen that can be placed as part of the app. It allows users to glance at stories' titles with the top story getting a summary.

Key Considerations

How will your app handle data persistence?

It will use a SQL Db to store article content. A script to 'purge' articles that are older than a few days can be run on launch to keep the data clean.

There will also be a table to map categories to RSS urls, but the user's settings will be stored in UserPreferences.

Describe any corner cases in the UX.

In the immersive experience, we need to set the text color so that it's contrasting with background image. For example black text on an image of the night sky won't make for a good experience.

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

Picasso, or Volley for network images.
ButterKnife for view binding

Describe how you will implement Google Play Services.

It is feasible to have a backend that polls the RSS feeds and pushes down the 'delta' using GCM. Firebase can also be used to back up user settings.

Another option is to have the client poll for the content at a frequency using the job schedule feature from Google Play Services.

Next Steps: Required Tasks

Task 1: Project Setup

Setup project

- Git
- Android Studio
- Decide on libraries
- Gradle file update
- Get a blank activity up and running

Task 2: Setup Data Access Layer

Create a 'storable' object that the app can interact with

- FeedName -> FeedUrl
- ArticleContent for the actual articles

Task 3: Setup Db

All things storage

- Finalize Sql schema
- Add content providers
- Write logic to read/write into Db from the storable object in Task 2

Task 4: Settings

Build the user preferences layout and have it stored in SharedPrefs

Task 5: ListActivity

UI + Business Logic

- Build the layout
- Build the loaders, job scheduler, poller to fetch data
- Update the view, notifications based on the data

Task 5: DetailActivity

UI + Business Logic

- Build the layout
- Get the current article
- Render the information by feeding to layout