

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

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

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: [robertkiszeli](#)

News Selector

Description

News Selector is a global news application. Using the api from newsapi.org.

- With the application you can get the top headlines from 53 different countries in 7 different categories.
- Search for articles in 13 language.
- Save article link for later read.
- With the refreshing widget you can get the selected country top headlines without opening the application.

If you want to get the latest news from a country this is your application.

Intended User

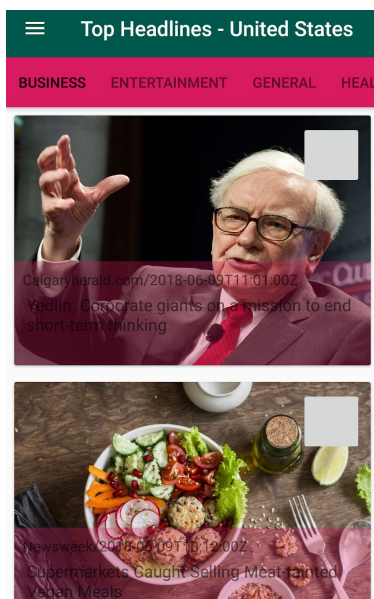
Every user who likes to get the latest news.

The supported countries are listed here <https://newsapi.org/sources>.

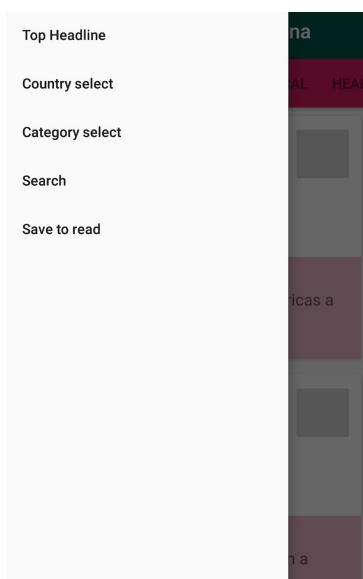
Features

- Top headlines from a country
 - 53 optional country
 - 7 different category
- Search in articles
 - In 13 different language
- Save for later read
- Widget to show selected country top general headlines

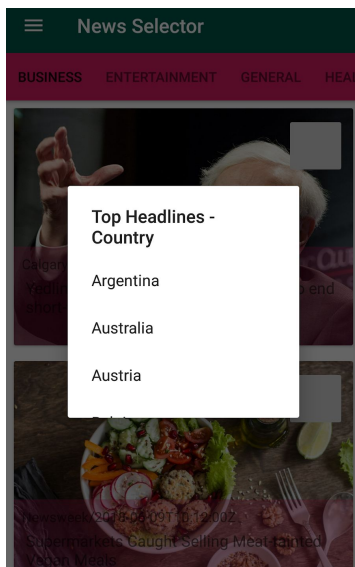
User Interface Mocks



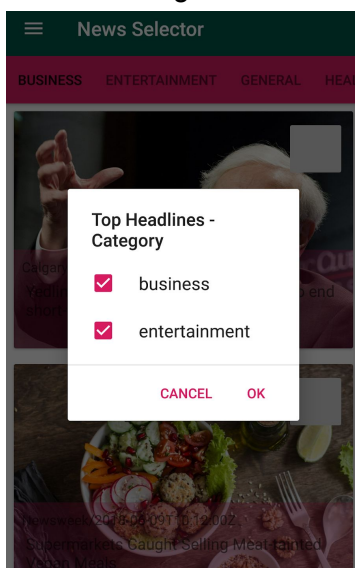
The main screen shows the top headlines from a country.



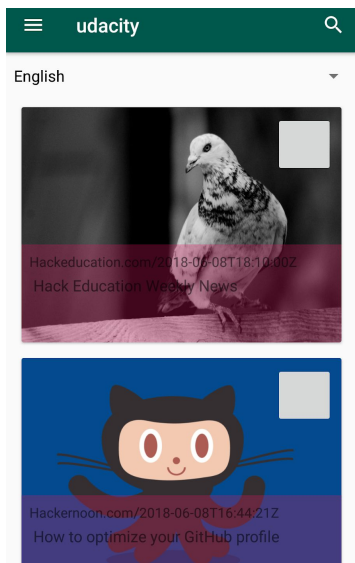
A navigation drawer will provide the different actions.



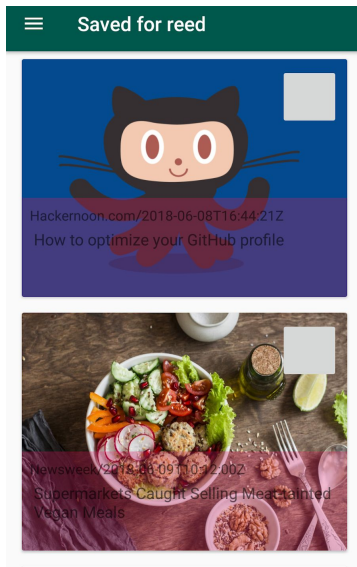
An alert dialog to select one country.



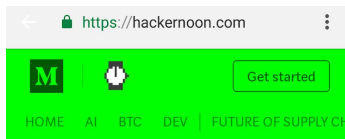
An alert dialog to pick category or categories.



In a search section we can search for articles in 13 language.



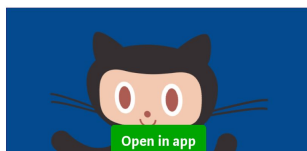
Every article card has a button to save it to a later time to read.



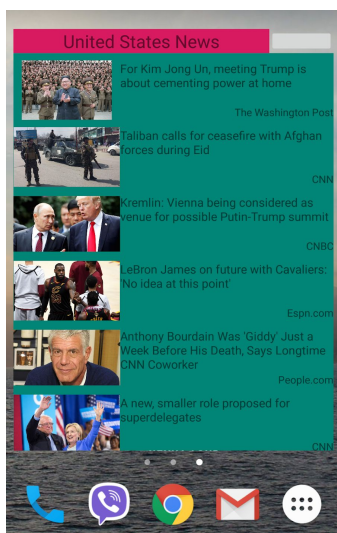
Pavel Malos
Growth Hacker @ejobs.ro | Tech.
Startups, Anthropology.
Jun 8 · 5 min read

How to optimize your GitHub profile

From Profile to Portfolio



Each article will open with Chrome Custom Tabs.



The widget will show the general top headlines from the selected country. Refreshing in every 30 minute, or with a button.

Key Considerations

How will your app handle data persistence?

The top headlines, search field and widget will get data from the network.

The saved for later read view will get the data from a database using a content provider.

Describe any edge or corner cases in the UX.

Every view is a different fragment the user can switch between the views using the navigation drawer.

The top headlines fragment has a ViewPager that is containing the categories. To switch between the categories the app uses a tablayout.

Without internet connection the sites are not loading and the application only shows articles in the saved for later read view. Everywhere else will be a text or image to show the problem.

The back button will navigate back to the app from the Chrome Custom Tabs view.

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

The programming will be in solely in Java using Android Studio. With gradle version 3.1.2.

1. "Com.android.support:appcompat-v7:27.1.1"
2. "com.android.support:support-v4:27.1.1"
3. "Com.android.support:design:27.1.1" - using for navigation drawer
4. "com.android.support:recyclerview-v7:27.1.1" - create article lists
5. "com.android.support:cardview-v7:27.1.1" - to show article data in list
6. 'com.android.support.constraint:constraint-layout:1.1.0' - for fragments xml views
7. "com.android.support:customtabs:27.1.1" - to use custom tabs
8. 'com.github.bumptech.glide:glide:4.6.1' - to load images
9. 'com.jakewharton:butterknife:8.8.1' - binding views
10. 'Com.jakewharton:butterknife-compiler:8.8.1'
11. "Com.google.android.gms:play-services-ads:11.8.0" - to show ads
12. 'Com.google.android.gms:play-services-analytics:11.8.0' - to use analytics

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

I will use play-services-ads to use banner and analytics to check user usage.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new blank project and all implementation to the gradle
- Set min/target sdk
- Set compile sdk

Task 2: Implement UI for Each Activity and Fragment

- Create the fragments for each view
 - Top headlines
 - Search
 - Save for read
- Connect the fragments to the activity

Task 3: Navigation drawer

- Add navigation drawer to the main activity
- Set the buttons to switch fragments

Task 4: Get data from network

- Create the AsyncTaskLoader and JsonParse for the
 - Top headlines
 - Article search
- Add data to the views RecyclerView.

Task 5: Save data

- Create Content Provider to save article data for later read
- Set article card button to save article
- Set save for later read view to refresh when article is deleted

Task 6: Set widget

- Create widget layouts
- Create a service to fetch and load data from network

Task 7: Ads

- Add a Banner ad

Task 8: Add the analytic

- Use google analytic

Task 5: Iterate project

Now I have a Minimum Viable Project, iterate till it doesn't meet the specifications.

- Set default images for every image view
- Save strings in the string.xml
- Set colors in colors.xml
- Configure to Material Design
- Set RTL to switch write direction for languages that are writing in right to left.
- Implement D-pad usage.
- Set keystore and password for signing.