
GitHub Username: BLTuckerDev

Hacker News Android Reader

Description

Keep yourself up to date on the latest news and trends in software development with this native client for Hacker News.

Intended User

Programmers with android phones who want to read hacker news in an app.

Features

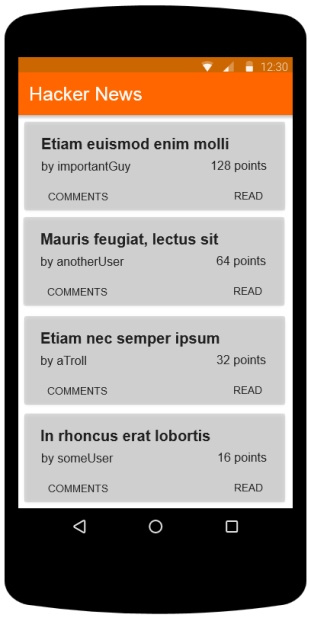
List the main features of your app. For example:

- Share Stories
- Read Comments
- Set "Read Later/At Home" reminders

User Interface Mocks

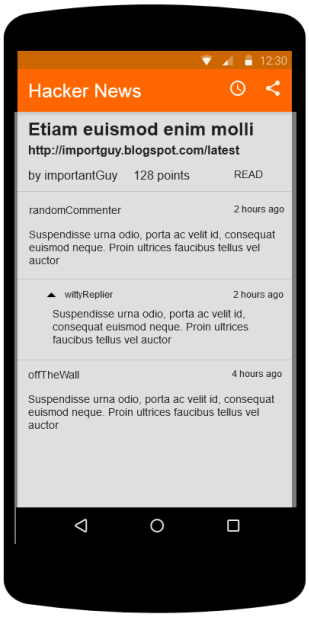
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



This will be the home screen for the activity. The user will be able to quickly see the top stories trending on hacker news from here. I will use cardview from the design library to help give each story its own space. The screen will be in a fragment using a recyclerview and in tablet mode it will appear on the left hand side of the screen much like we did in the spotify streamer app.

Screen 2



This will be a “details” screen for the app. On a table UI it will be what appears in the second pane on the screen. The user will be able to read the comments about the story here as well as click read to be taken to the actual article in a webview. I will also allow them to share the story or set a reminder to read the story when they get home. I am including the functionality to let them choose to be reminded to read when they get home so that I can take advantage of the google play services location api and help fulfill the requirements for the rubric.

Key Considerations

How will your app handle data persistence?

I will sync story meta data and comments to the device with a content provider and a sync adapter. I’ll use a loader to get the data to UI. It should be very similar to how we handled data in the sunshine app.

Describe any corner cases in the UX.

I don’t foresee any corner cases in my app’s UX.

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

I plan on using schematic to make the implementation of the content provider easier. I first learned about it during one of the udacity webcasts and would like to give it a try.

I also plan on using retrofit + okhttp to take care of communication with the api.
I will be using the android support libraries + the design library to make the app have a consistent and modern look and feel while supporting more than just the latest devices.
Timber for logging.
If the need arises I may also take advantage of rxJava and EventBus.

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

- Initialize git repo
- Create project via Android Studio project templates
- Add dependencies

Task 2: Implement UI for Each Activity and Fragment

- Create a story “headline” layout for the recyclerview on screen 1
- Create a fragment to contain the top stories recyclerview
- Wire the recyclerview up to an adapter that can receive paged data loaded as the user scrolls
- Create a second fragment to contain a header with a recyclerview for comments
- Put both fragments inside of a controlling activity

Task 3: Application Logic

- Sprinkle google analytics around the app
- Make use of the location api by getting a user’s home location and setting up callbacks to display a notifcation about reading material when they get home
- Setup a retrofit service that can pull data from the hacker news api
- Setup a content provider to store data that we pull from the api
- Setup a notification manager to handle our notification logic
- Work with loaders to display the information in our fragments

---

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**"