GitHub Username: abhrajitmukherjee

Hacker News Cards

Description

Hacker news (https://news.ycombinator.com/news) is a very popular news curation portal for tech enthusiasts. Their interface is very minimalist but functional. This app will get data from hacker news api and display in intuitive card form. It will have multiple layouts for different device types. One main view and detail view will be present in the app.

Intended User

This app is intended towards tech enthusiasts who follow Hacker News.

Features

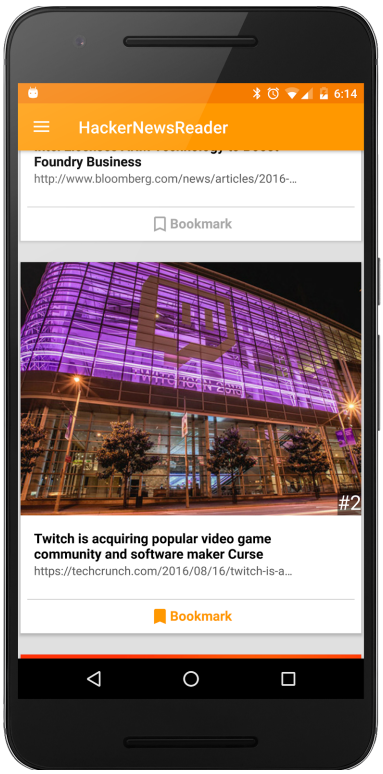
List the main features of your app. For example:

- Displays the hacker news feed in Google plus style cards.
- Has a detail view to show the curated content.
- Dynamic card re-ordering based on ranking change at the api end.
- Periodic background updates of the data so that user always gets the latest data.

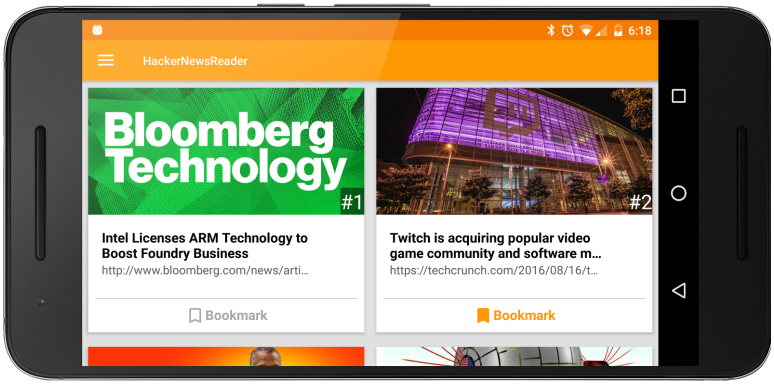
User Interface Mocks

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.

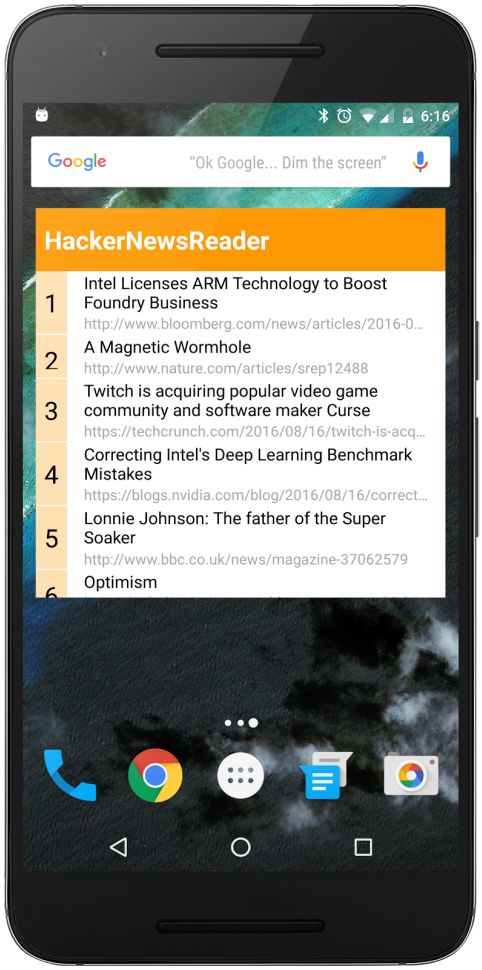
Main Screen



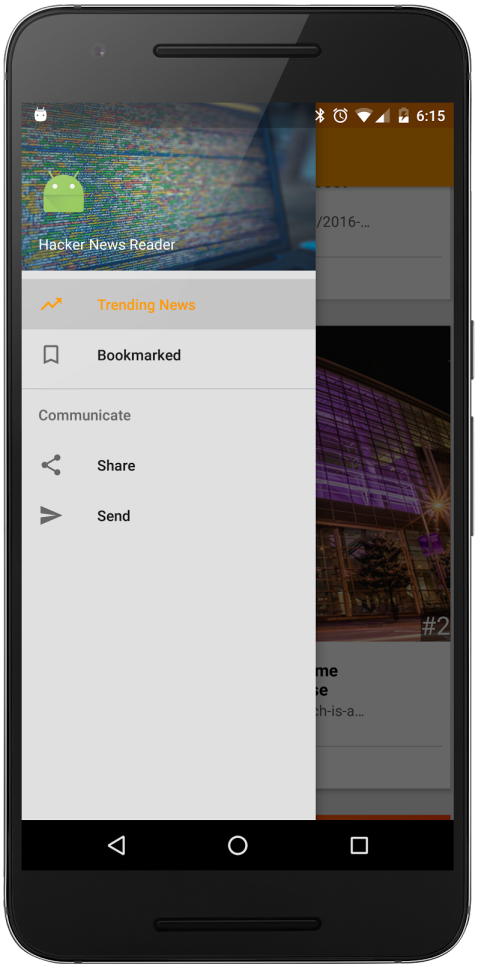
Portrait (Phone)



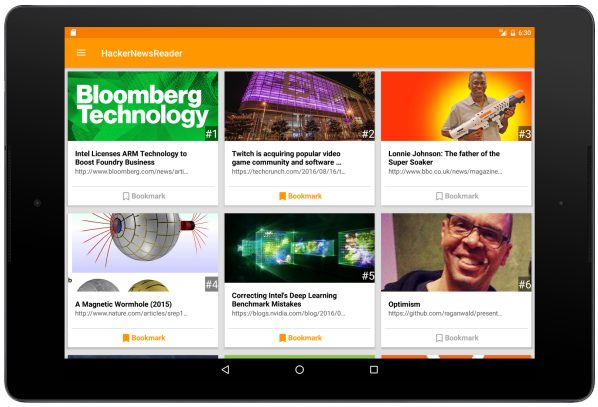
Landscape (Phone)



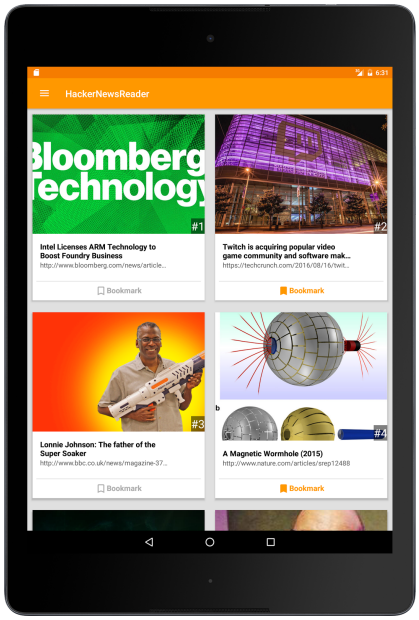
Widget



Navigation Drawer



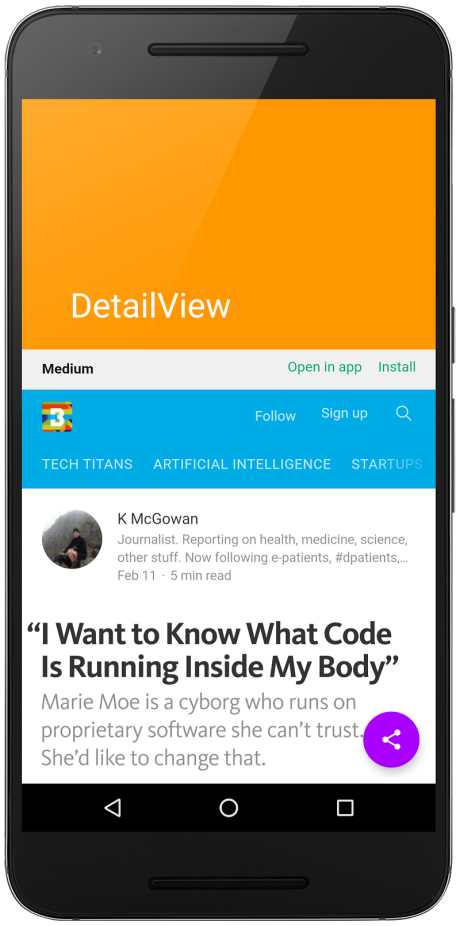
Tablet Landscape



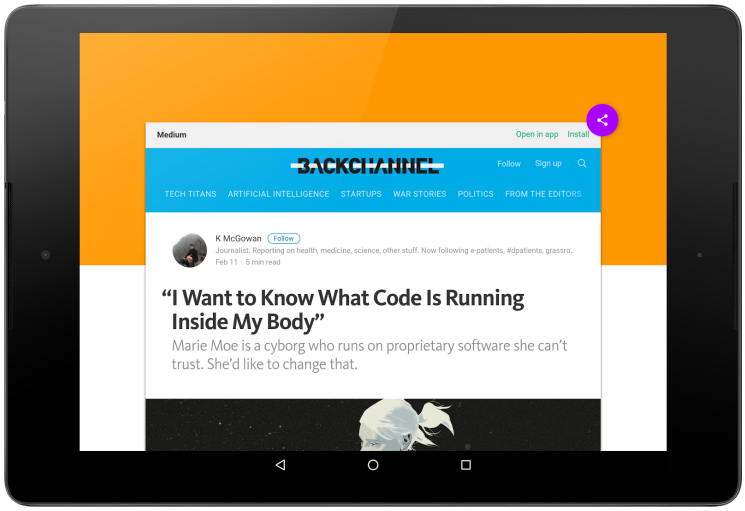
Tablet Portrait

The main phone screen will display the news feed in cards format, it will have an extracted image with title and target link. The rank will be present as an overlay on the image. The layouts and number of cards in the grid will vary based on the device type and orientation. All cards will be clickable to take it to the detail view where the main story will be shown. Where no image extraction will be found, it will be replace with one palette color with an overlay of the first letter of the title. There will be a nav bar to switch from trending news to bookmarked news. Also a widget will provide a quick list of trending news on the home screen.

Story View



Phone



Tablet

The detail story view will show the link contents from the main page that was clicked. This layout will vary based on device type. It will have a fab button will enable the user to share the link through other apps.

Key Considerations

How will your app handle data persistence?

All data will be stored in SQLite database and fetched and moved through views using Content provider, cursor adapter and loader combination.

Describe any corner cases in the UX.

None

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

- Hacker News API- https://github.com/HackerNews/API
- Jsoup- To parse the weblinks and enable extraction of main image from the link.
- Picasso- For image processing in Image views.
- Okhttp- For web network calls.
- Simonvt.Schematic- For content provider setup.

Describe how you will implement Google Play Services.

- GCM (Google Task Service)- For periodic task to load data and keep the app news feed updated in backend.
- Google Analytics- To measure user activity to named screens

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

To setup project following steps to be followed:-

- Start with an empty activity project.
- Add all module dependencies to the build.gradle files.
- Prepare dimens.xml, strings.xml for various sizes and orientation (for ex. Land, w820 etc)

Task 2: Implement UI for Each Activity and Fragment

- Build the main UI layout with dummy image and headlines. Replicate the layout for various device kinds by filling in dimensions and other details in dimens.xml file for that layout.
- Create two detail layout. One for phone and the other for w820. The layout should look like the mock provided above.
- Link UI using onclick listeners.

Task 3: Build image extraction routine

Write an image extraction class and dependent procedure. This will parse a given link using jsoup and extract the main image from the link and return it back. This has to take care of keeping top performance and not parse the page endlessly. Also this has to select and pick an image which is of high resolution and avoid picking ads or other filler images. If no image is found of relevant specifications it will return null.

Task 4: Content Provider, Cursor Adapter, Loader

- Build the content provider along with support db and contract files.
- Bind the recycler view adapter with the cursor adapter.
- Initiate loader to move data betwewn views.

Task 5: Build intent service and GCM service

- Write an intent service class which will call the GCM service on load and during manual refresh.
- GCM Task Service
    - The gcm service task to be written to read from hacker news api and load the relevant column data into the database.

    - It should only get the top 500 news check if its there in DB, then just update rank else make an entry to the DB. This will make each refresh faster with fewer network calls.
    - The records not found in top 500 to be deleted from DB.

Task 6: Integrate Analytics Service

- Initiate the analytics service and bind it with all named view
- Run the app and check if stats are captured.

Task 7: Build and Connect Widget

- Design layout for widget.
- Connect using pending intent to main app.

Task 8: Accessibility and Localization Readiness

- Check accessibility readiness of the project by checking content description, rtl support
- Check for localization readiness checking no harcoded strings present and non-translatable string explicitly marked in strings.xml.

Task 9: Refractor and Final Build

- Refractor the project for removing unnecessary log statements, checking indentation, accessibility, localization.
- Generate signed apk.