GitHub Username: geaden

Hacker News Reader

Description

The app allows to read top stories from https://news.ycombinator.com/. You can easily read story content from the app, receive updates, read comments, bookmark the stories you like mostly.

Intended User

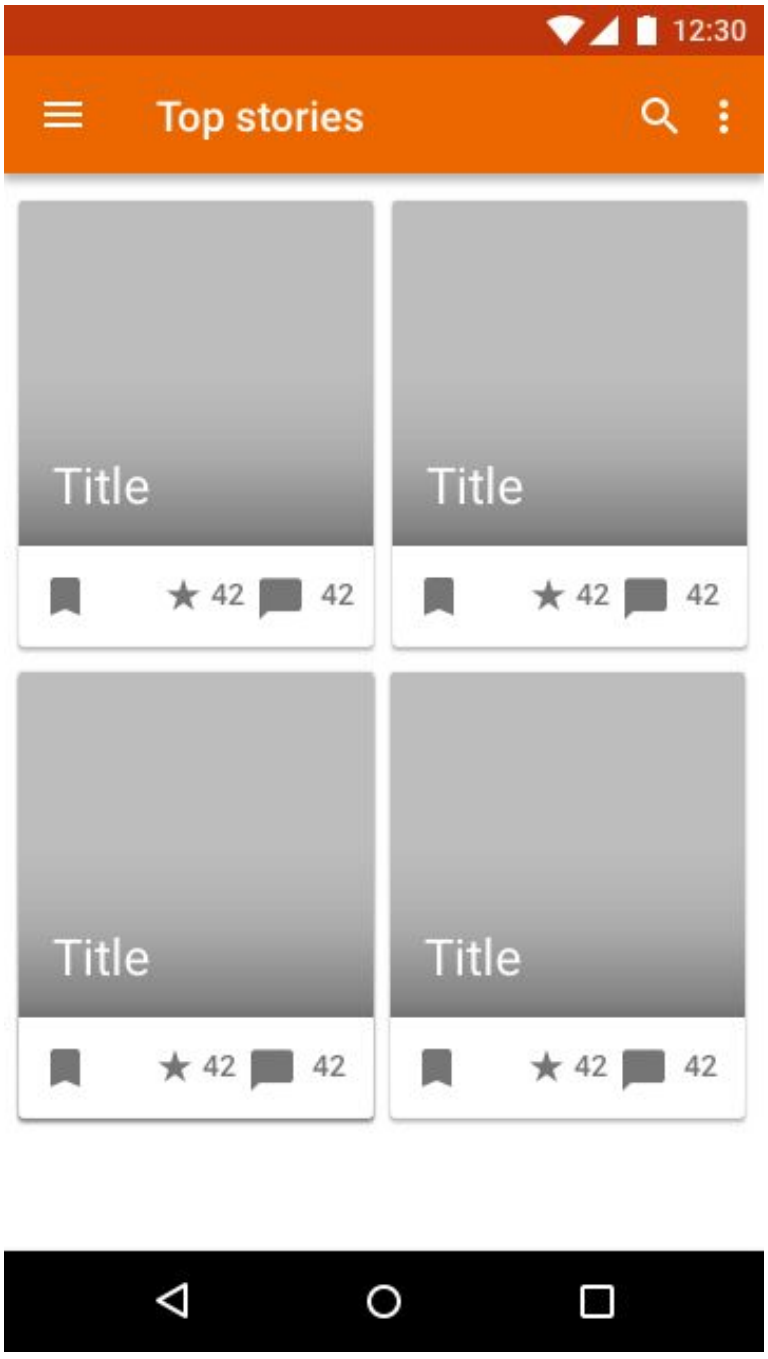
The app is intended to keep any interested person informed about the news around Silicon Valley, new technologies and startup companies.

Features

- Read top stories from Hacker News.
- Save stories for future reading
- Share story

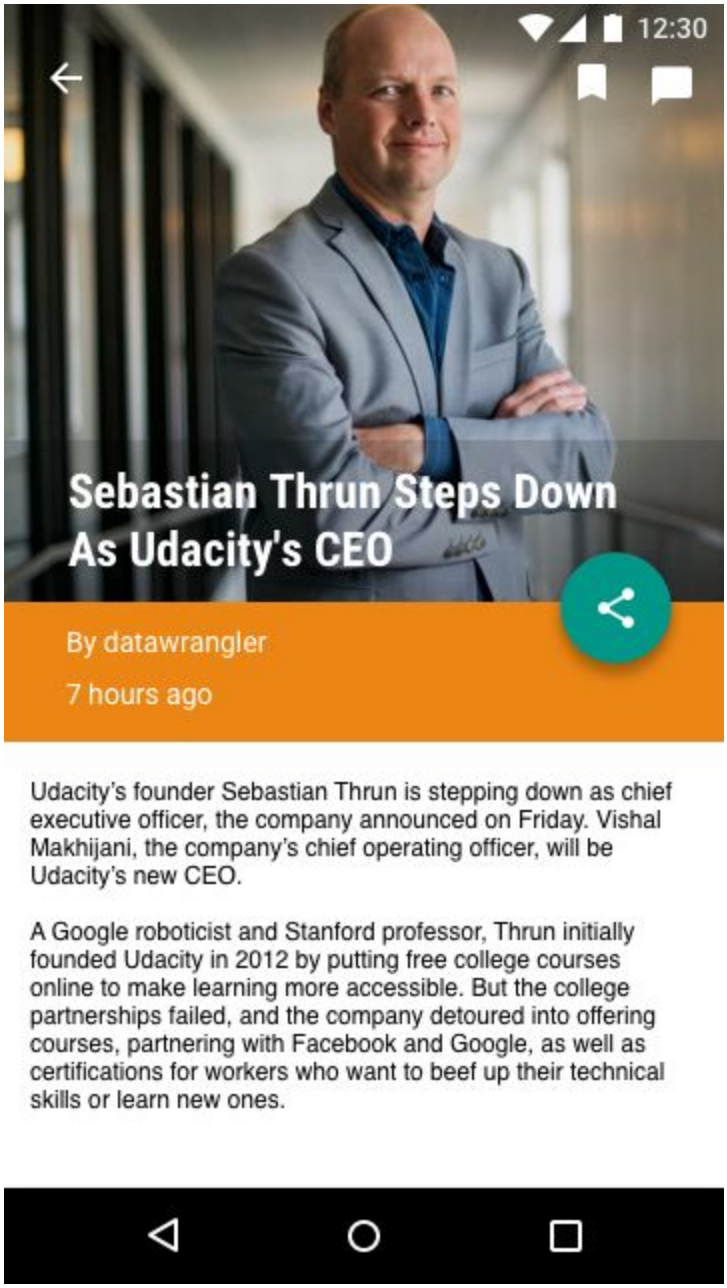
User Interface Mocks

Screen 1



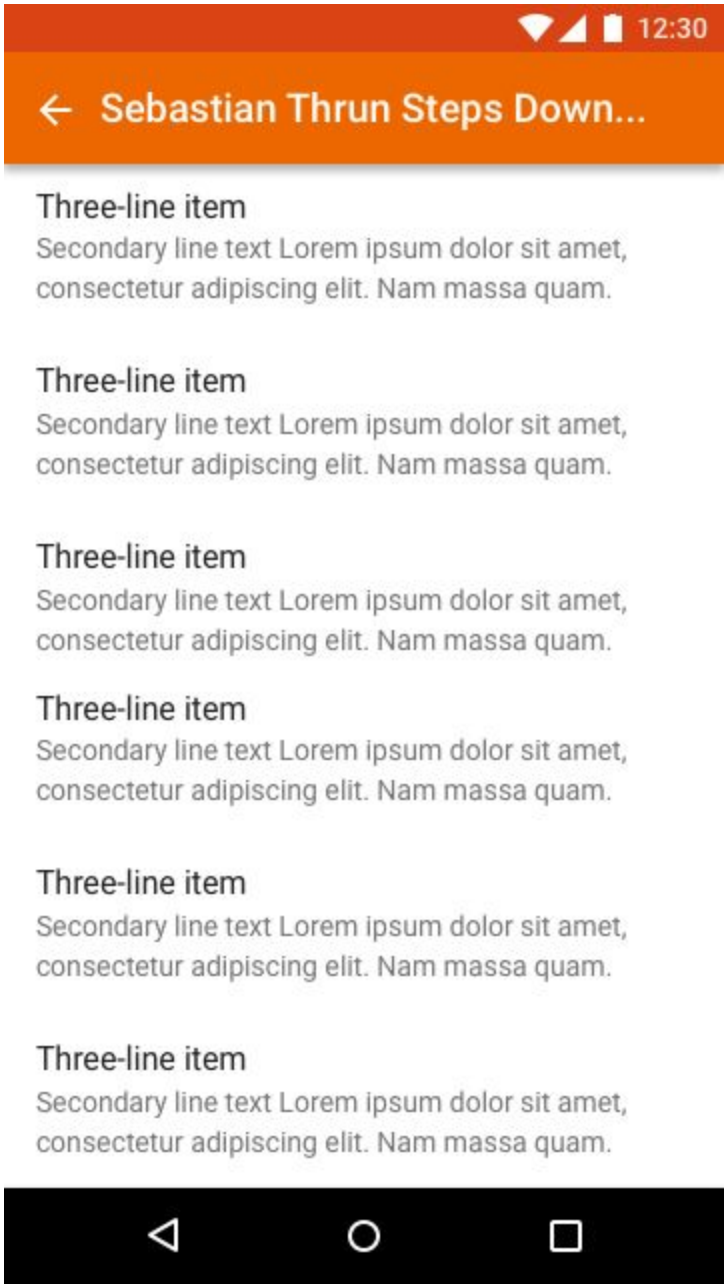
Main activity. Represents grid of articles retrieved from backend.

Screen 2



Story detail activity

Screen 3



Comments activity for single story

Key Considerations

How will your app handle data persistence?

The app will have custom content provider to store data. The content provider will be generated by DBFlow library.

Describe any corner cases in the UX.

To access comments of the story the user has to click icon in toolbar and new activity will be opened with list of comments

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

In Android App

- Glide for image loading and processing.
- Butterknife to easily locate views in the view.
- Raizlabs DBFlow for data manipulation and Content Provider generation.
- Dagger2 for dependency injection.
- Android Support Library.

For backend

- Firebase - to interact with hacker new api (https://github.com/HackerNews/API)
- Boilerpipe (https://github.com/kohlschutter/boilerpipe) - to extract main content from article if possible.

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

1. Setup backend.

Backend will be hosted in Google App Engine and will use Firebase library to access Hacker New API and Goose library to extract main content (image and story content). Some stories will be prepopulated and will be updated (by removing old items) from Hacker News on daily basis.

Backend will have several endpoints:

GET /v1/topstories

List of top stories. Mirror of https://hacker-news.firebaseio.com/v0/topstories.json?print=pretty

GET /v1/topstories/{id}

Get story content. Mirror of https://hacker-news.firebaseio.com/v0/item/{id}.json?print=pretty

POST /v1/topstories/{id}/bookmark

Bookmarks the story

GET /v1/topstories/{id}/comments

Gets list of comments related to a story.

GET /v1/bookmarks

Gets list of bookmarked stories for a user.

Backend will have cron task to periodically update stories from hacker news.

2. Configure libraries versions that will be used in project.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Activity
- Build UI for Details Activity
- Build UI for Comments Activity

Task 3: Implement Database configuration

Implement common entities used in the app.

- Create story entity

- Create comment entity
- Create entity to store bookmarked entities.

Task 4: Populate data from backend

Configure backend to retrieve top stories from it.

Task 5: Configure periodic sync with the backend

Setup sync adapter of job scheduler to periodically sync stories with the backend.

Task 6: Handle different cases

Handle different cases, such as no network available and user clicked bookmark the story, story data is not properly extracted, etc..

Task 7: Implement Google+ Sign-in

In order to store information about bookmarked stories by the user, the user will have to sign-in with Google+ account.