
GitHub Username: himakiran

geekNews

Description

A self contained news app catered to the geek community. Select the news sources of your choice and the app delivers the latest and popular news from those sources .

You can read all the articles/posts in the app itself. The app also allows you to browse and see videos from youtube that are related to the news article.

Intended User

Technology buffs and geeks who need to stay on top of the cutting edge and the latest developments in tech.

Features

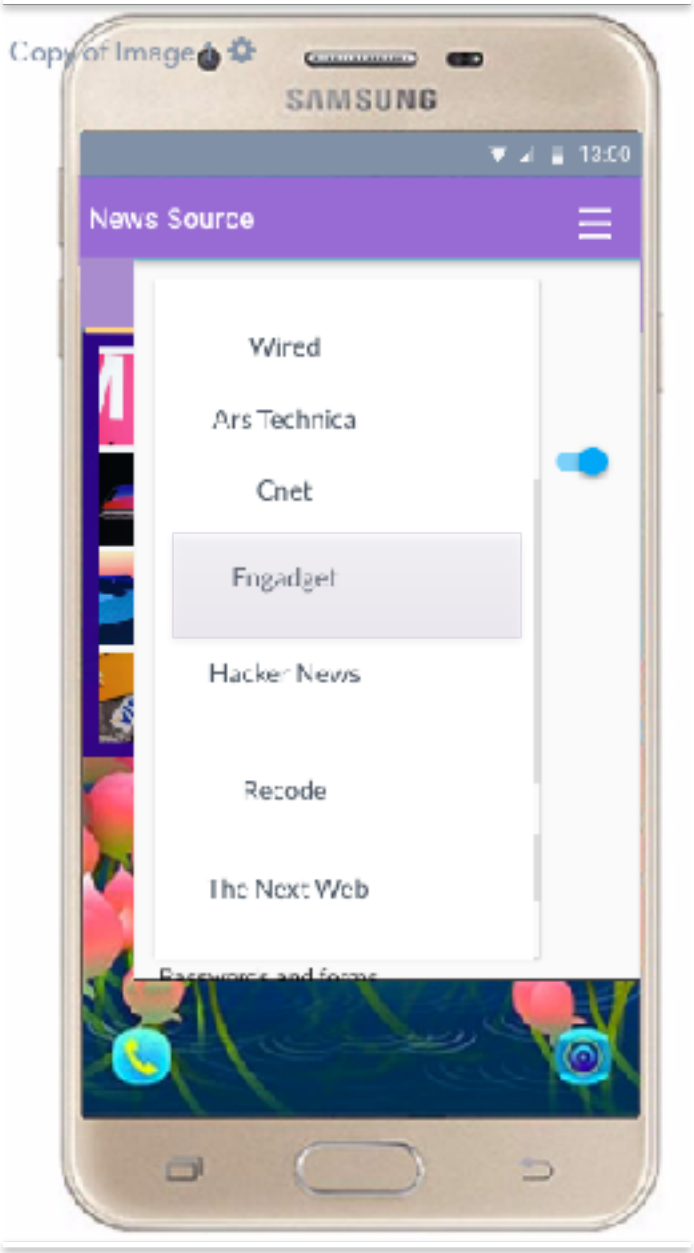
- Allows you to select the source of tech news.
- Shows the entire article in the app itself.
- Comprehensive list of news sources which include Ars Technica, Wired, Cnet etc.
- Shows videos related to the news article.

User Interface Mocks

Screen 1

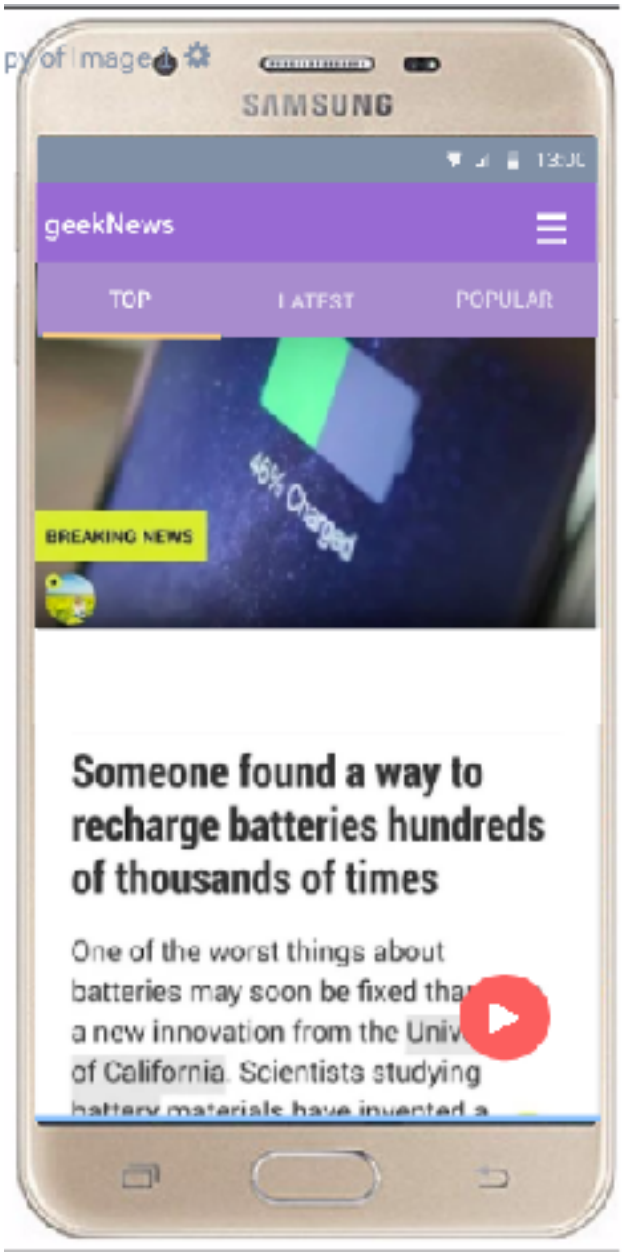
This is the first screen when the app is launched. It contains the top news articles from a singe default source. The user can select latest and popular to see the respective feed.

Screen 2



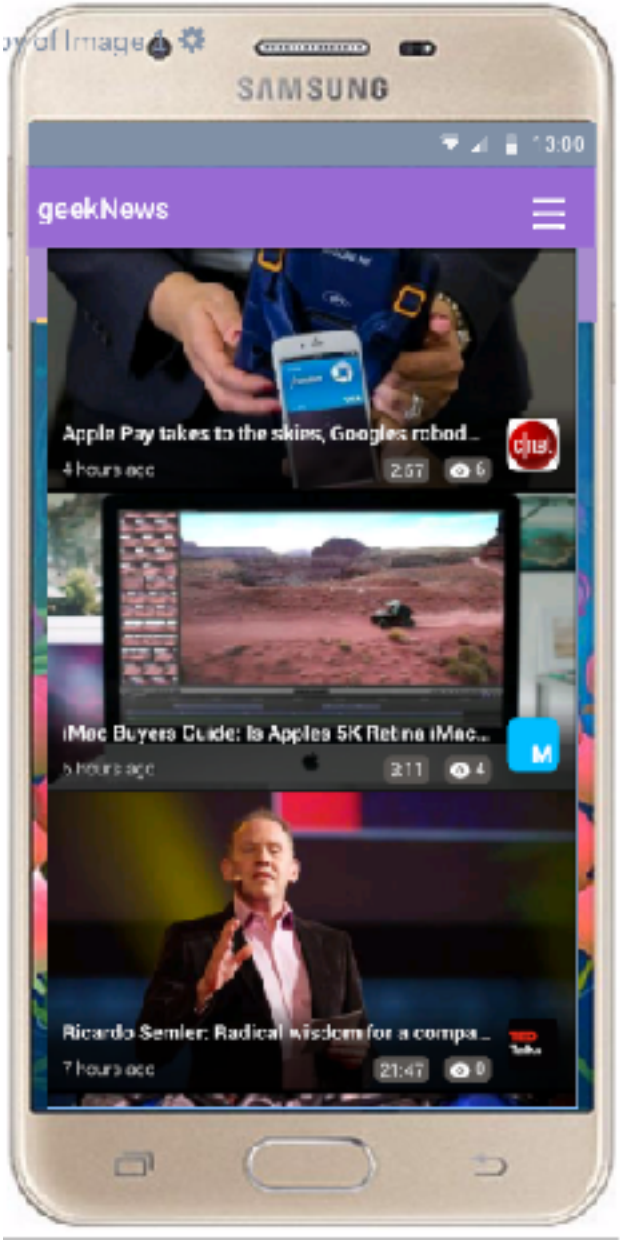
The user can select a source from the list by clicking on the menu button and selecting the desired source.

Screen 3



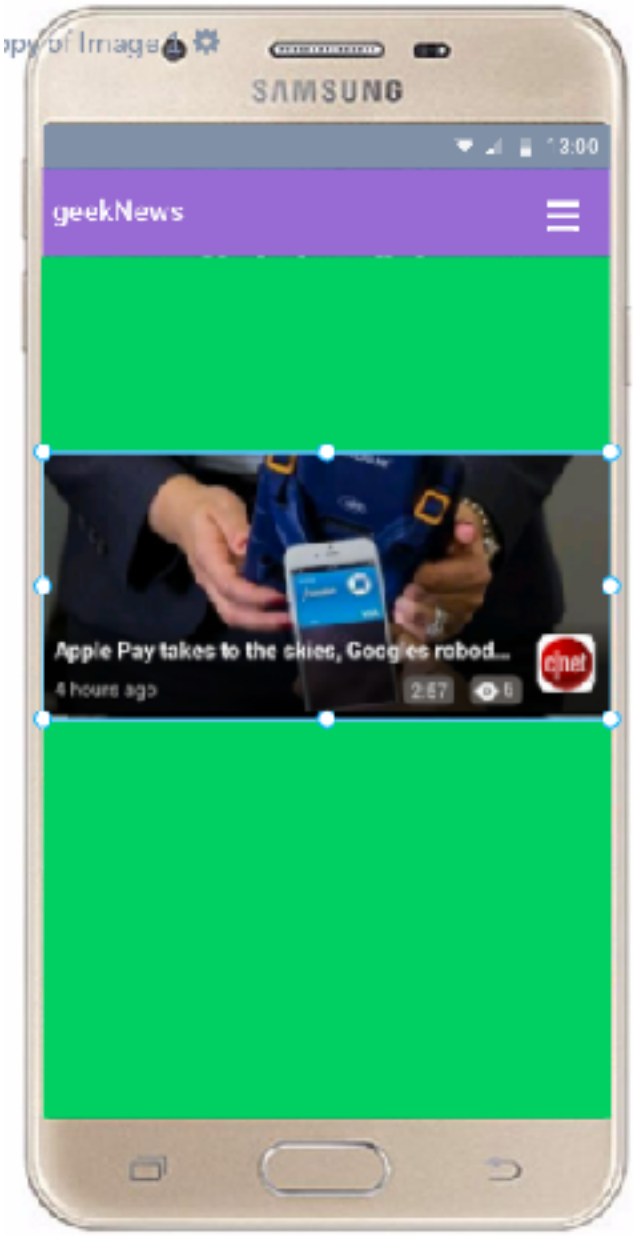
Upon clicking on any article on the previous screen the user reaches this screen where the entire article is displayed along with an image and a title.

Screen 4



Upon clicking on the FAB on the previous screen the user reaches this screen where videos on youtube related to the previous article are displayed. If no related videos are found a toast is displayed and the user is returned back to the previous screen.

Screen 5



Clicking on any video on the previous screen plays the video.

Screen 6

The above screen shows the geekNews Widget

Key Considerations

How will your app handle data persistence?

The data shall be stored in a content provider and a sync adapter shall be used to keep the data up to date.

Describe any edge or corner cases in the UX.

In all the screens shown above the back button shall the user back to the preceding screen he was in.

One special case would be when user changes the source from any screen. In that case the user will be transported to screen 1 which shall show the refreshed feed as per his selection.

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

The Picasso library shall be used for displaying images.
Volley shall be used for networking.
Gson shall be used for JSON string to object conversion.
Exoplayer shall be used to play videos.
Espresso shall be using for testing.

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

The main content for the app shall be fetched from newsapi.org.
The following APIs from google shall be used.

- YouTube Data API for fetching videos.
- The Google Play Developer API for publishing apps and Subscriptions.
- AdMob for displaying ads in the free version of the app.

(The free version of the app shall limit the number of sources.)

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

1. Setup an empty activity with fragments initially.
2. Setup the project for Single Activity Multiple fragments.
3. Each screen will have its won Fragment.java file
4. Libraries shall be added as when the respective feature is being implemented.
5. The minimum SDK version and target SDK version shall be decided by looking up the statistics from the google play store.
6. Appcompat activity shall be extended by Mainactivity.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity - will be a blank container UI
- Build following UI
    - Home screen with Recycler view to display content embedded in ConstraintLayout with App Bar and tab swipes and menu.
    - DetailArticle View screen shall use a ObservableScrollView inside a frame layout.
    - Implement a Recycler view to show list of related videos.
    - Implement a fragment layout to display full size video.

Task 3: Implement AysncTask and JSON Parser.

- Both of the above shall get JSON content from the newsapi.org site and the parser shall parse and break up the content into fields like title, thumbnail , shortdescription.
- The url field shall be used to fetch the entire article.

Task 4: Implement the ContentProvider ,SyncAdapter and Loader.

- The SQLite database will be designed and this shall be used to fashion the ContentProvider.
- The SyncAdapter shall be coded.
- Implement Loader to transfer the data from the content provider to the UI screens.

Task 5: Implement the Home Screen.

- The MainFragment.java file shall be coded and using the home layout file will be used to display the home screen.
- The menu inside the app bar with source selection shall be coded.
- Top , Latest and Popular shall be coded as per Material Design specs and implemented.

Task 6 : Implement the Detail Screen.

- The onClick method on each of the items in the recycler view of the previous screen shall be coded to launch the detailFragment which shall display the entire article along with an image in a scrollview.

Task 7: Implement the listVideos Screen.

- The FAB button on the previous screen shall be coded to launch the listVideos fragment which shall display all the related videos.
- This shall be done by sending the title as a keyword to the Youtube API and the list of videos fetched and displayed.
- Exoplayer shall be used to display the video in fullscreen upon clicking of each screen.

Task 8: Widget

Implement widget provider and widget remote views factory.

.

Task 9: Testing

Android Studio test recorder shall be used to create tests using the espresso library.