GitHub Username: Vishvnath96

Insta News

Description

A News Feed app which keeps a user up to date with updated news from the global related to a particular topic, location and person. We have used, [Guardian API](#). This is a well-maintained API which returns information in a JSON format.

Intended User

This app is for all the users they are intended to read news for all topics or selected topics.

Features

- Save news for offline reading
- Ability to choose from different type of sources of news.
- Save news for offline reading.

- Share news with friends.
- World news
- Technology news
- Science news
- Sports news
- Environment news
- Color theme
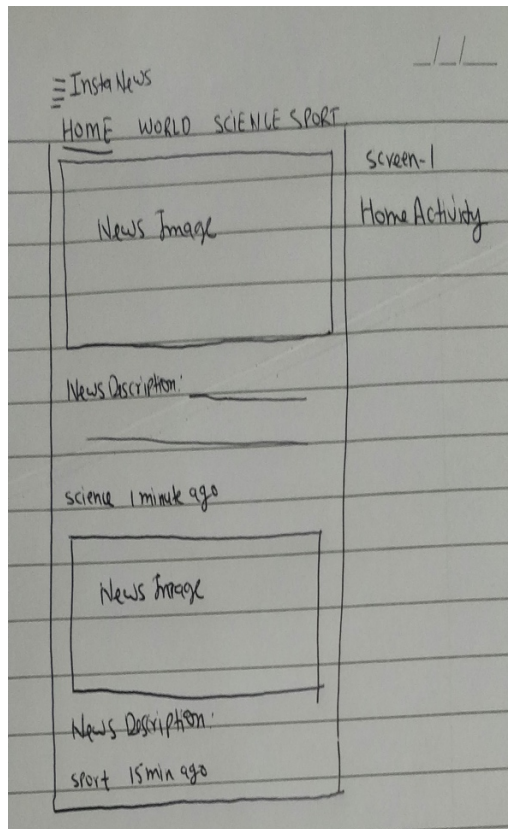- Custom Date to select news

Tech features:

- Guardian API
- JSON Parsing
- Glide/Picasso
- CardView
- RecyclerView
- SharedPreferences
- Navigation Drawer
- Fragments
- ViewPager plus TabLayout
- Loaders
- Intent

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow).
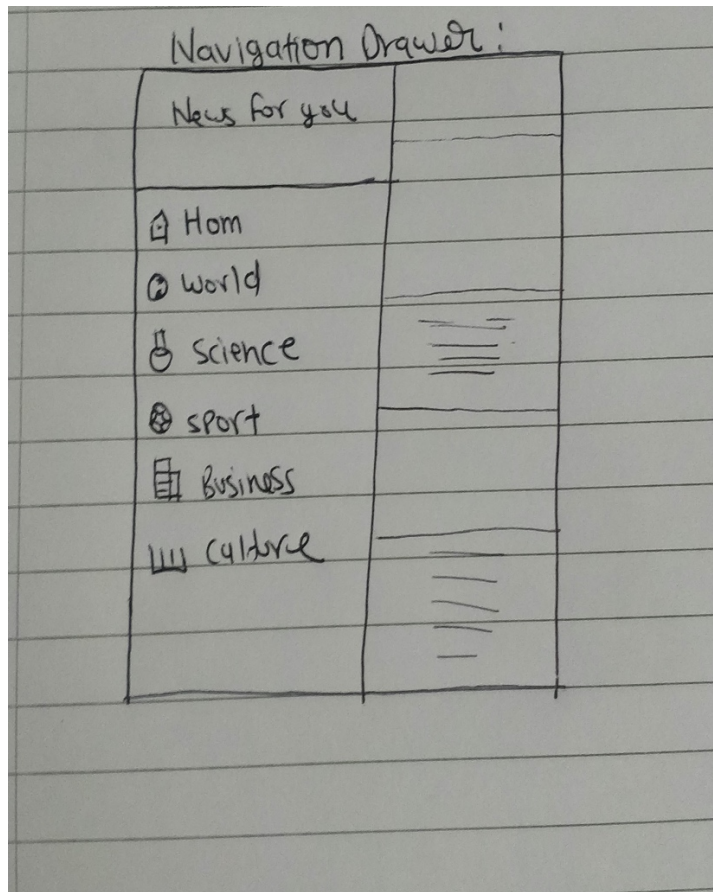
Screen 1: Home Activity

Home screen contain following:
1. Main Content (View Pager ,Recycler View): displays the set of news with title, image, news category and published time .
2. Overflow menu: contain options to sort news by top, popular or latest news.
3. Navigation Drawer (see Screen 2)

Screen 2: Navigation Drawer

Navigation Drawer contain following:

To select different types of news following possible options:

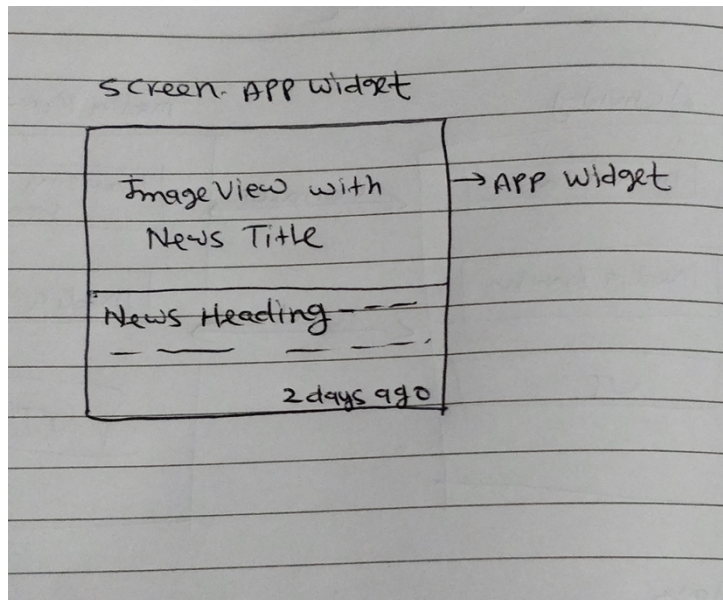Business, entertainment, gaming, general, music, politics, science-and-nature, sport, technology.

Screen 3: Detail Activity
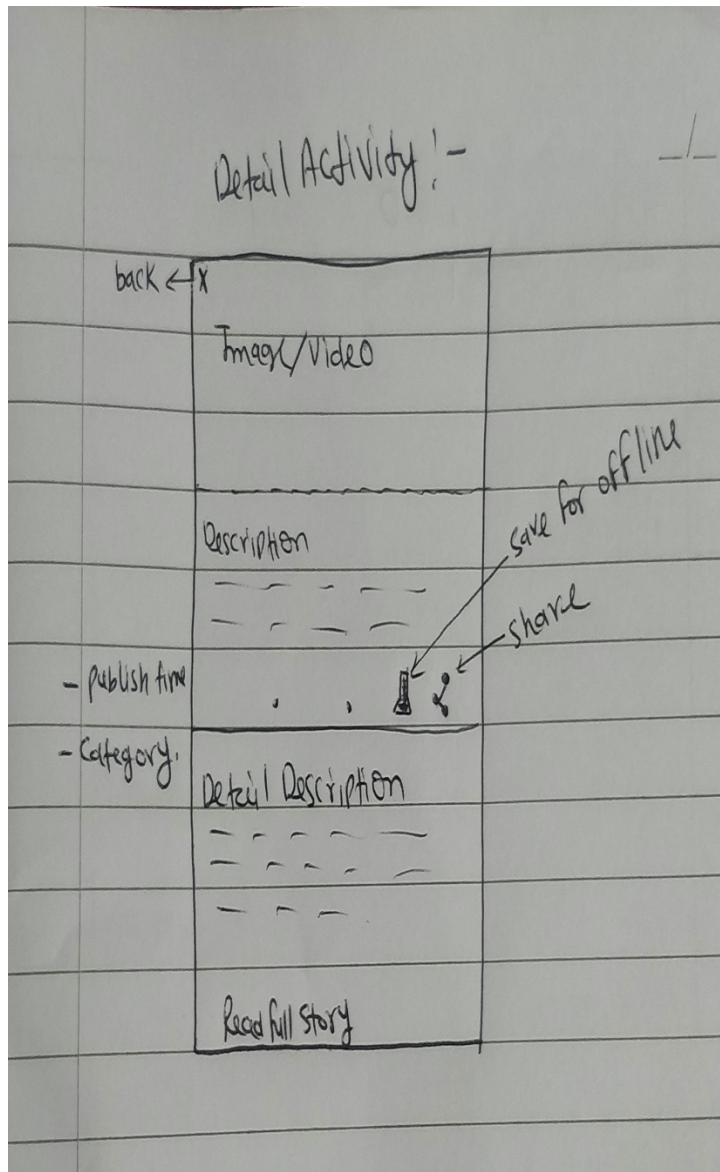
Detail screen contain following:
Detail Activity contains all information about news with read more, share button to share and favorite button to save news to read or listen offline. Back button to navigate to home page.

Screen 4: App Widget

App widget screen contain following:
News image heading and update time.

Key Considerations

How will your app handle data persistence?

Data persistence will be handled using Room database and Shared Preferences. In Room database data stored and retrieved using Content Provider.

Describe any edge or corner cases in the UX.

On device rotate not required loading of data again. Try to implement swipe to refresh feature.

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

-Glide/Piccasso to handle the loading and caching of images. -DataBinding and view binding.
-Retrofit to load news data from Guardian api.

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

I will use following google play services:

1. Google Admobs for showing ads.
2. Firebase for storage, analytics and crash reporting.

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

- Configure libraries and all dependency required for set up.
- Create a debug and build variant.

Task 2: Database

- Create database
- Content provider
- Data model for news and category

We will use Room Database with LiveData and View Model for data persistence and retrieval and Latest Android Architecture components.

Task 3: Network Data and Loaders

- Network task for fetching data
- Parsing data
- Loader for loading saved data.

Task 4: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for Detail Activity
- Share function for sharing news.

Task 5: Network Tasks and Loaders

- Network task for fetching data
- Parsing data
- Loader for loading saved data.
- it will regularly pulls or sends data from API, app updates data in its cache at regular intervals using a SyncAdapter or JobDispatcher.

Task 6: Implement UI for Each Activity and Fragment

- Build UI for Main Activity
- Build UI for Detail Activity
- Share function for sharing news.
- Save news for offline reading

Task 7: Implement Google Play Services

- ● Google Admobs
- ● Google Firebase

Task 8: Handle Error Cases

- Error cases handling
- Error message and loader
-

Task 9: Clean up the Project

- Move all strings in strings.xml
- Delete build files

Task 10: Language support

- We will use java language to develop whole app

Task 11: Libraries and Studio version

Tool/library name	Version
Android studio	3.6
Gradle	6.1.1
okhttp	3.11.0
om.google.firebase:firebase-analytics	17.2.1
androidx.test.espresso:espresso-core	3.1.0
androidx.recyclerview:recyclerview	1.0.0
Junit	4.12
androidx.lifecycle:lifecycle-viewmodel-ktx	2.1.0
androidx.lifecycle:lifecycle-extensions	2.1.0
androidx.room:room-runtime	2.0.0
com.google.code.gson:gson	2.8.0