

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Navigation Drawer Implementation](#)

[Task 4: Simple Settings Page](#)

[Task 5: Implementing search articles function](#)

[Task 6: Updating news articles](#)

[Task 7: Create a home screen app widget](#)

[Task 8: Implement Google Play Services](#)

[Task 9: Handle Error cases and test](#)

[Task 10: Make the App material and refactor](#)

GitHub Username: [SanketBhat](#)

News 4 You

Description

Searching the web for the latest news? Do you want to get the latest news instantly? Here is an app for you. "News 4 You" gives the latest news from many popular sources and allows reading top news by few clicks. News updates are shown in the home screen saves a lot of time.

Explore news articles by different categories or search the article you want to read. News you read is stored for offline.

Intended User

This app is for anyone who wants to stay up to date with the latest news. This is for one who wants to read all categories news article within a single app, wherever they go whenever they want.

Features

- Shows latest headlines in the MainActivity as well as home screen widget
- Updates news frequently and notifies the user
- Shows the news articles by different categories such as sports, science etc.
- Allows the user to search news article by the keyword
- Stores loaded articles for offline reading

User Interface Mocks

Screen 1



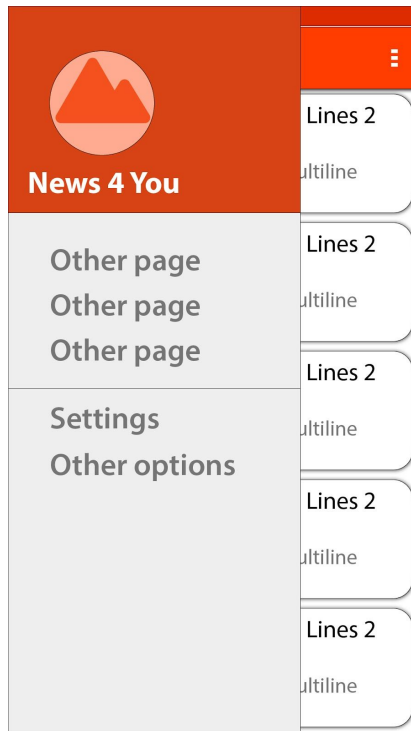
The home screen of the app contains top headlines taken from all sources, displayed in the order of published date descending. The user can click on of the news article to launch detail screen or navigate to the web page of the corresponding website.

Screen 2



When the news article item is clicked the respective detail activity should launch containing more details about news such as author(if exist), source, published time, content and a link to the complete article. The screen has a FAB used to share the current article link.

Screen 3



App implements navigation drawer for easy navigation between pages.

Screen 4



The app allows the user to explore news article by its category ex: science, politics, sports etc.

Screen 5



App provides home screen widget to show instant news updates.

Key Considerations

How will your app handle data persistence?

App uses room database and view model implementation for data persistence. Recent headlines are stored in the database for offline reading. And they are deleted once they become too old.

Describe any edge or corner cases in the UX.

- The user navigates top headlines, news category activity pages using navigation drawer.
- Clicking news articles in the news list will launch detail activity or webpage that has more details about the news.
- The user can also click the home screen widget to open top headlines activity.

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

- Using Picasso library for image loading and caching
- Using retrofit for retrieving JSON data from the "NewsAPI.org"
- Using Room database for data persistence

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

- The app uses google analytics for error reporting
- Uses google ads for showing ads(Test ad)

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
- Create POJO's according to the JSON data fetched, and map JSON to Java object.
- Create room database and attach it with ViewModel implementation

Task 2: Implement UI for Each Activity and Fragment

Subtasks

- Build UI for MainActivity
- Build the RecyclerView item layout to show a single news item
- Connect RecyclerView with the activity ViewModel such that list shows articles stored in the database
- Build News categories activity with the ViewPager and tab layout implementation
- Build the header layout for navigation drawer, and create option items for navigation drawer

Task 3: Navigation Drawer implementation

Subtasks:

- Implement navigation drawer to show all necessary pages to navigate in the app
- Implement navigation item click event and navigate clicked pages properly

Task 4: Simple Settings Page

Subtasks:

- Create a simple settings page and add an entry point to the navigation drawer
- Configure shared preference files
- Retrieve user selected option in the activities and fragments to make it work according to user preference

Task 5: Implementing search articles function

Subtasks:

- Allow the user to search articles based on a search string
- Implement functionality to search gives results from API

Task 6: Updating news articles

Subtasks:

- Using job scheduler schedule job to fetch new news articles from the server
- Again schedule a job to regularly maintain the database ex: delete the articles that are too old

Task 7: Create a home screen app widget

Subtasks:

- Create an app widget and make it show new headlines from the database
- Update home screen widget as soon new articles loaded into the device

Task 8: Implement Google Play Services

Subtasks:

- Implement Google Analytics and report the errors
- Show ads in activities

Task 9: Handle Error cases and test

Subtasks:

- Handle error cases and show the proper message to the user
- Test the app against various situations

Task 10: Make the App material and refactor

Subtasks:

- Make all layouts, margins, paddings, colors are according to the material design guidelines.
 - Do refactoring, for example: Extract hardcoded strings, dimensions etc
-