

[Description](#)
[Intended User](#)
[Features](#)
[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#) ☐

GitHub Username: zhejun90

API News Stand

Description

Find news from an assorted collection of various categories and a wide variety of sources.
Powered by News API

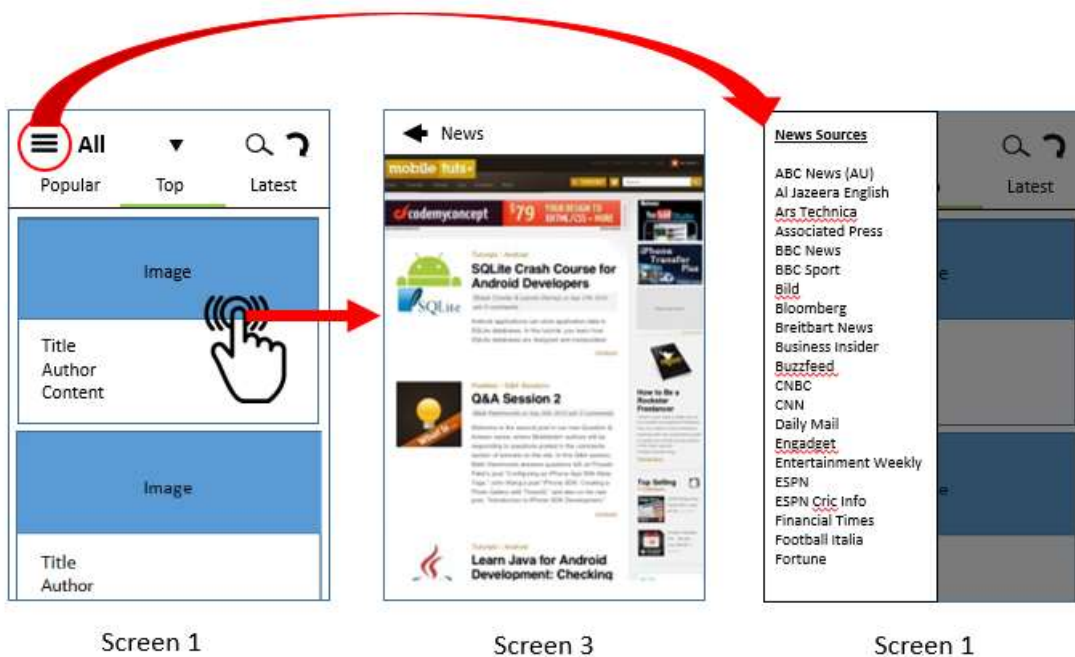
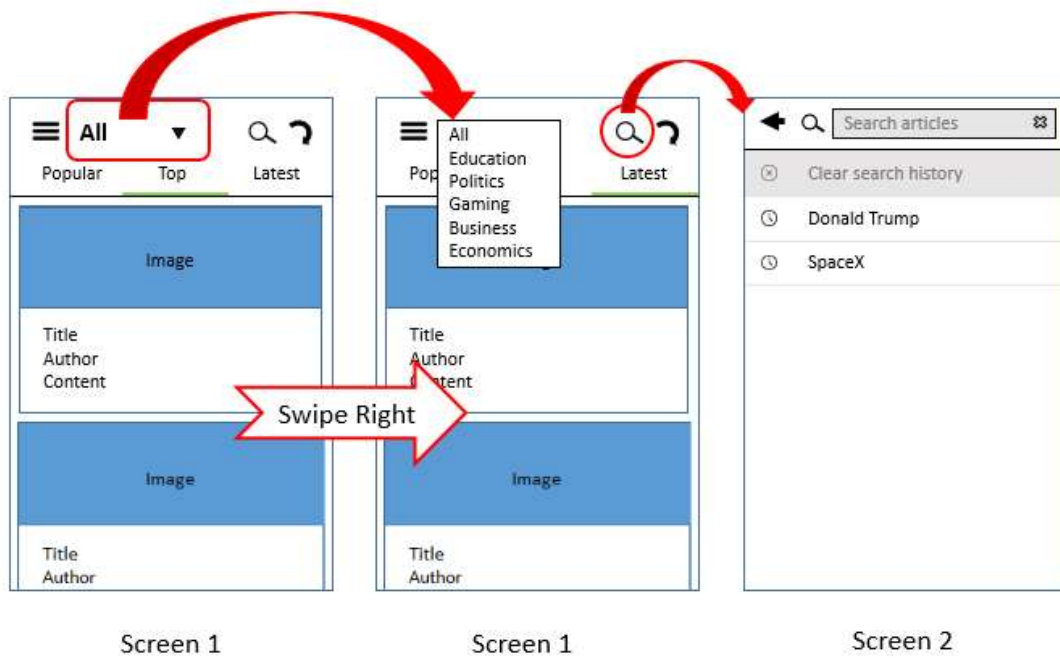
Intended User

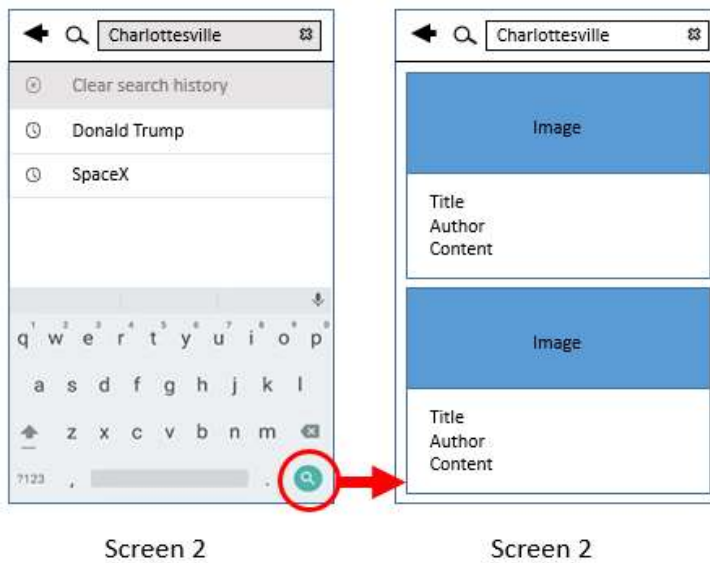
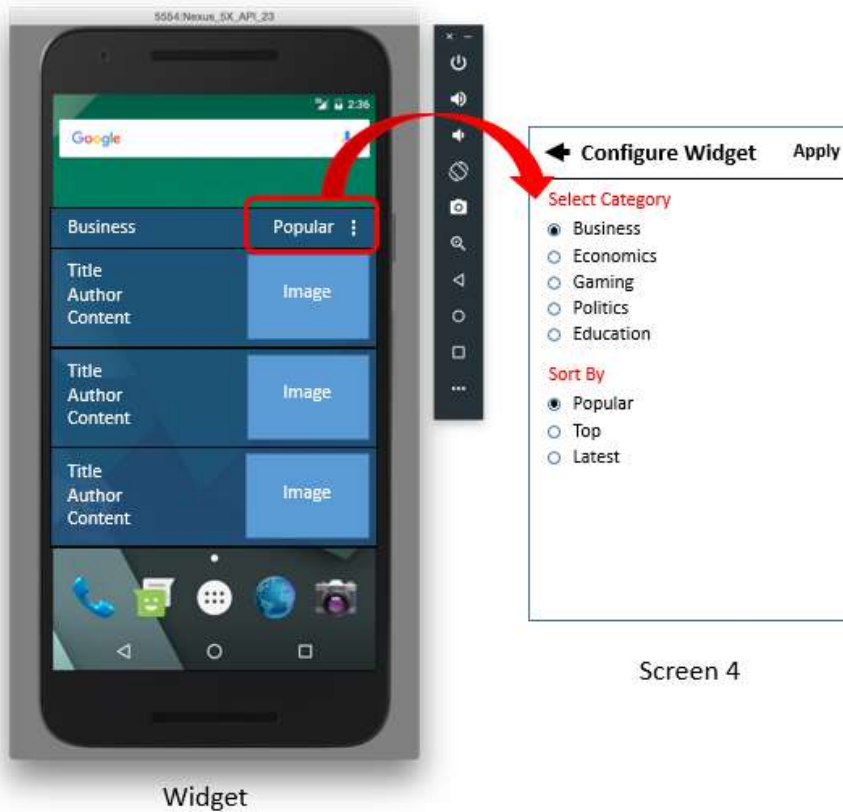
News readers

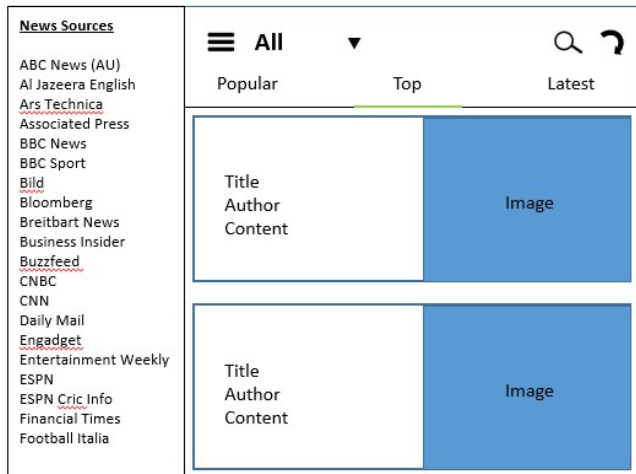
Features

- Load articles list
- List articles based on user preferences

User Interface Mocks







Tablet Screen

Key Considerations

How will your app handle data persistence?

Obtain data from GET request with IntentService, then store them in local database.
 SyncAdapter for updating data once in a while, activated through widget.
 ContentProvider to provide methods for database read/write.
 Cursor loader loads data in MainActivity and Widget.

Describe any edge or corner cases in the UX.

No internet activated.

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

Glide – to handle loading/caching
 Butterknife – dependency injection

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

AdMobs – ads
 Analytics – analyse searches that users conduct

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 Google play services, build.gradle, libraries etc

Task 2: Fetch data

Get data from News API, validate incoming data
Create SQLite database to hold request data
Implement ContentProvider methods to access database

Task 3: Implement UI for Each Activity and Fragment

MainActivity (Screen 1 & Tablet Screen)

- List of all articles using a RecyclerView
- Implement ViewPagerAdapter for tab control for different sort (popular, top, latest)
- In lieu of tablet screen, implement a fragment for holding the views
- Include a navigation drawer in layout
- For navigation drawer, contents to be wrapped in fragment (also due to tablet screen)
- Miscellaneous (AppBar, search button, refresh, list selection)

SearchActivity (Screen 2)

- Show history of searches(if any)
- Show search results and hide search history
- RecyclerView for search listing

WebViewActivity (Screen 3)

- URL from selected article in MainActivity

ConfigurationActivity (Screen 4)

- Configure widget
- Only available when widget is added
- Back button goes to Screen 1/Tablet Screen

Task 4: Testing

UI tests

Task 5: Widget

Implement widget as shown in mockup. SyncAdapter for widget to be implemented so as to allow for the most updated news. Sync timer should be every few hours for low battery usage.

Task 6: Admobs and Analytics

Integrate test ads into app, as well as analytics in searches.

Task 7: Signing

Configure signing.