

How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
 2. Name your document file: “**Capstone_Stage1**”
 3. Replace the text in green
-

□

[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: [silkster87](#)

News Wiz

Description

This app shows the top headline news from a variety of sources and categories that the user can sort and filter. They can also search for articles related to a specific subject. A widget is available to minimally show the top headlines. It will be a free app with no ads.

Intended User

This app is intended for everyone who wants to catch up with local and worldwide news at a glance. It will not be an app to replace any official news company app but will fetch data from [newsapi.org](#)

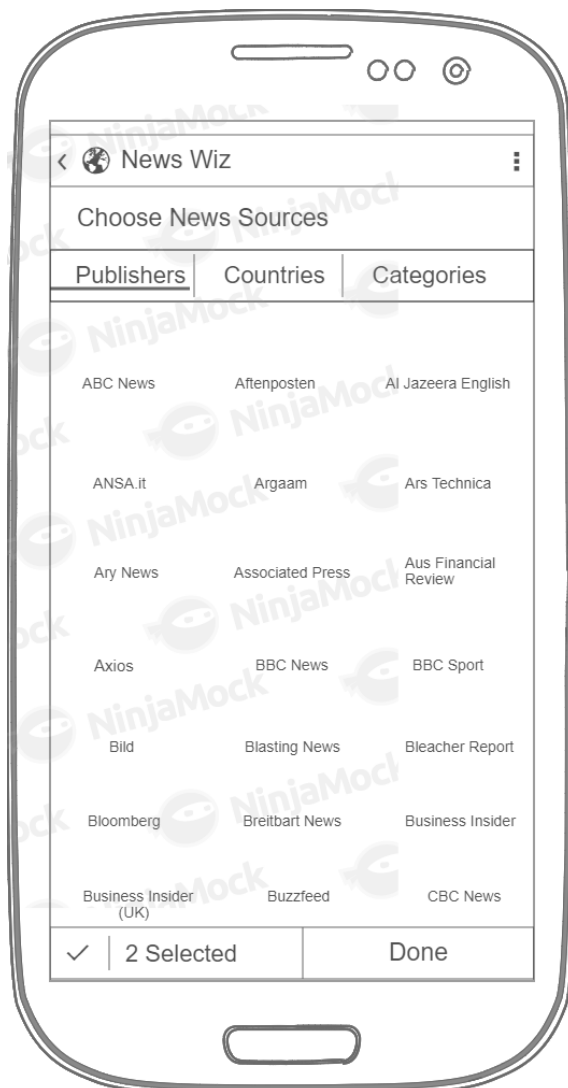
Features

- The main feature of the app is displaying top headline news from variety of sources and categories chosen by the user
- User can bookmark news article

- Get the app to display headline articles in widgets

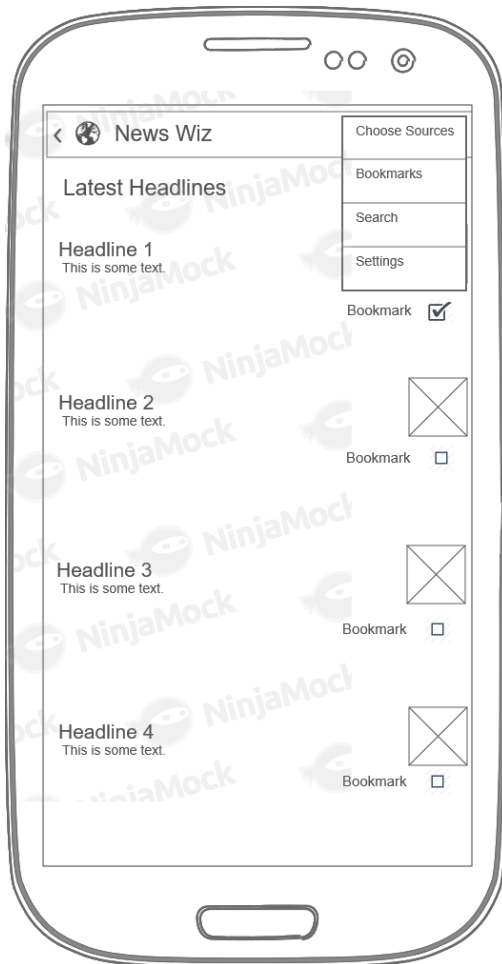
User Interface Mocks

Screen 1



The user chooses a variety of news sources filtered from publishers, countries and categories. Once user is satisfied with selection they can hit “Done” to then display the news from their selection. If the user does not select any then a dialog box will display “Please select at least 1 source”.

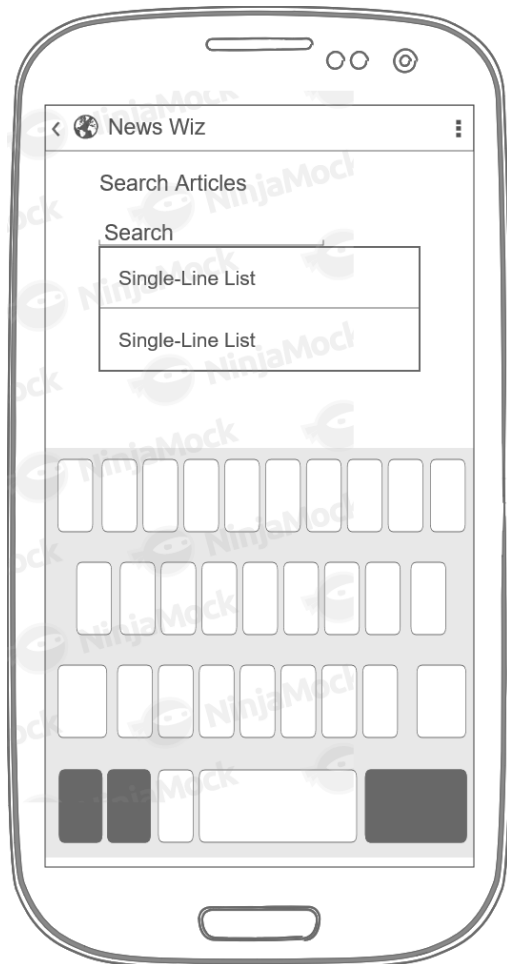
Screen 2



The main app will display the latest headlines from the selection of categories that the user has chosen. When the user clicks onto the headline link it will direct them to the page via the web link.

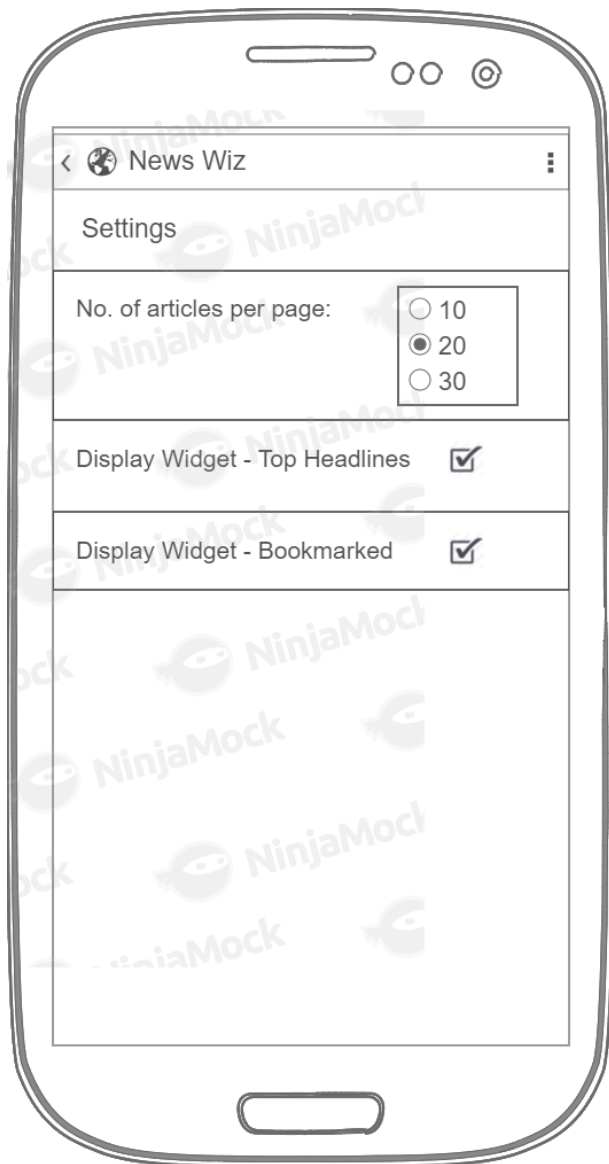
The user can also go into the settings and navigate back to choosing sources, bookmarks or change the settings of the app.

Screen 3



The user can search for articles by clicking onto the menu options and clicking on "Search". Here they can type relevant keywords to bring up related topics relevant to the keyword. As they type a key feature will be a drop down list of suggestions that the user can choose to search.

Screen 3



Key Considerations

How will your app handle data persistence?

The app will use a content provider to handle bookmarked articles to persist data. The Android Room Persistence Library will assist in storing data for offline use.

Describe any edge or corner cases in the UX.

If the user hits the back button when showing list of articles it will navigate to the main screen of choosing the sources of news.

If the user hits the back button in the “Search” activity it will navigate back to the list of articles.

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

Use of Picasso to help load and cache images. Butterknife for View elements, RxJava for responsive searching.

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

The app will implement Firebase authentication, allowing a personalised experience for the user. I will also be implementing Firebase Crashlytics for the stability of the app.

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 Firebase setup – import the Google Services Json file.
Add Butterknife, RxJava, OkHttp, Picasso, Room Persistence, RecyclerView and Android JetPack dependencies in Gradle Build file.

Task 2: Implement UI for Each Activity and Fragment

- Build the UI for sign-in. User will sign in and then be directed to “Choose Sources” activity.
- Build UI for the “Choose Sources” Activity – use fragments for the “Publishers”, “Countries” and “Categories” views to keep it within the “Choose Sources” activity.
- Follow the layout in the screen 1 mock. Ensure a grid layout of 3 columns in recycler view of the sources.

Task 3: Build UI for Latest Headlines

- Create the layout for the articles – use recycle view.
- From the sources chosen a request is made using the API news key through OkHttp to get a response.

- Each article will have a brief description, a link to the article and a picture if available. Use Picasso to display images.

Task 4: Build UI for Search Articles

- The main feature is that the text field will implement RxJava so that a list of suggestions will be displayed in the drop down list as user types.
- A recycle view layout will be displayed of the results

Task 5: Build UI for Settings

- The items in the settings will be the user preferences and they will be saved for the user.
- Create simple layout to display the user preferences.
- The default should be displaying 10 articles per page and not to show widgets.

Task 6: Create Widgets

- Make simple layout for the widgets, both top headline widget and bookmarked widget. It should at the minimum have a description of the articles.
- Will be displayed if user has chosen it to be displayed.

Task 7: Handle Error Cases

- There are specific error codes in the api requests – identify them and display the relevant dialogue to notify the user.
- Display dialogue box to show the user if they are offline so no connection.

Add as many tasks as you need to complete your app.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"