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

GitHub Username: https://github.com/drmaluta

News Now

Description

The app provides articles from various news sources. Users will be able to choose if they want to see articles from all sources or one particular source of their choice. Java language will be used for development of the app.

Intended User

Everyone who wants to get news.

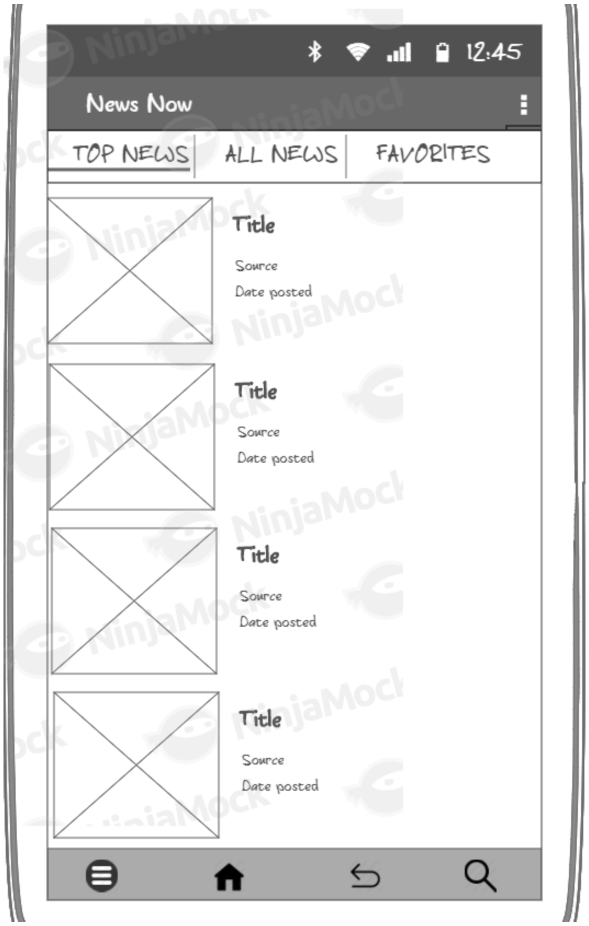
Features

Here are some of the features of the app:

- Loads articles from the Internet
- Shows Top news, All News and Favorite articles
- Lets use choose source of the news or see all news
- Provides opportunity for user to mark an article as “Favorite”
- Provides opportunity for user to share an article
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts
- App includes support for accessibility. That includes content descriptions.

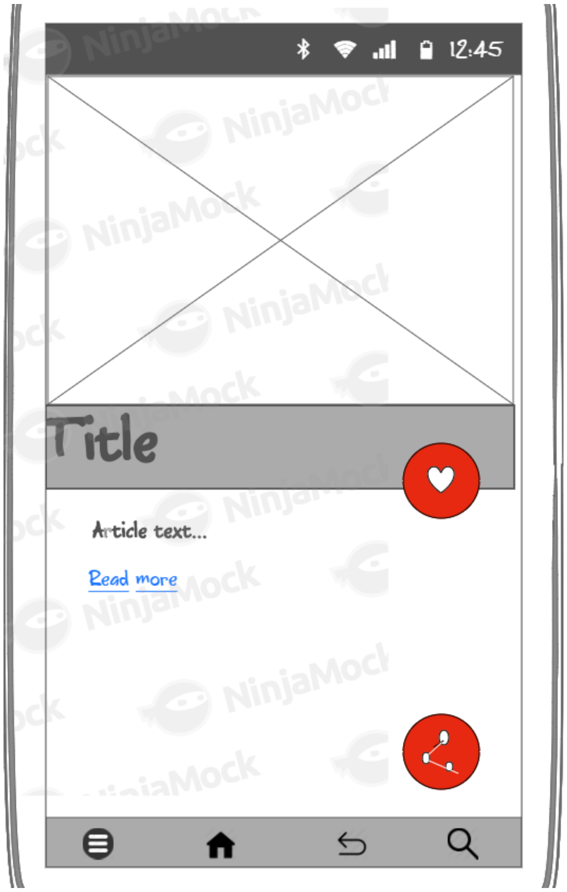
User Interface Mocks

Screen 1 Main Page



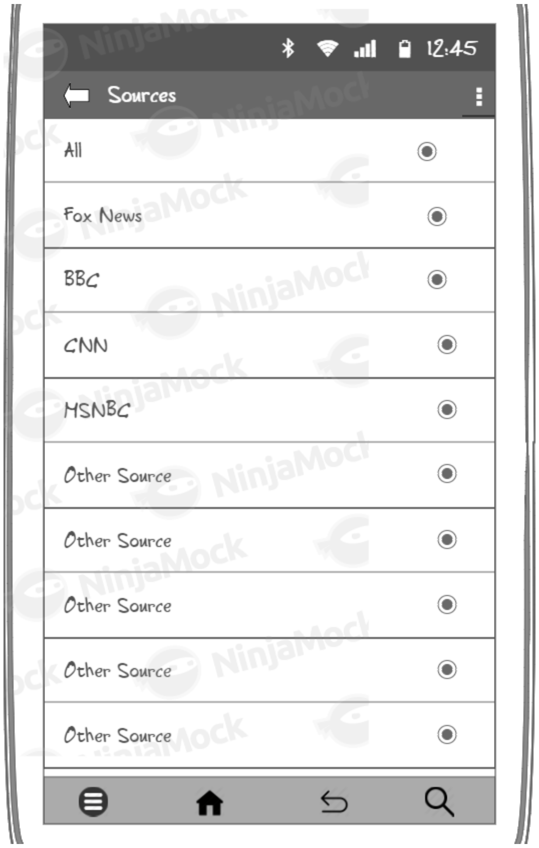
Articles will be presented on main page in a RecyclerView. They will be grouped in three categories (tabs): TOP NEWS, ALL NEWS and FAVORITES.
The difference between phone and tablet view will be number of columns: 1 for the phone and 2 for tablet. Clicking on the article navigates to Article details screen Clicking on the Settings and clicking on Sources navigates to Sources screen.
User can refresh list by using swipe-to-refresh action.
Also, the list includes native Ads (free version only).

Screen 2 Article Detail



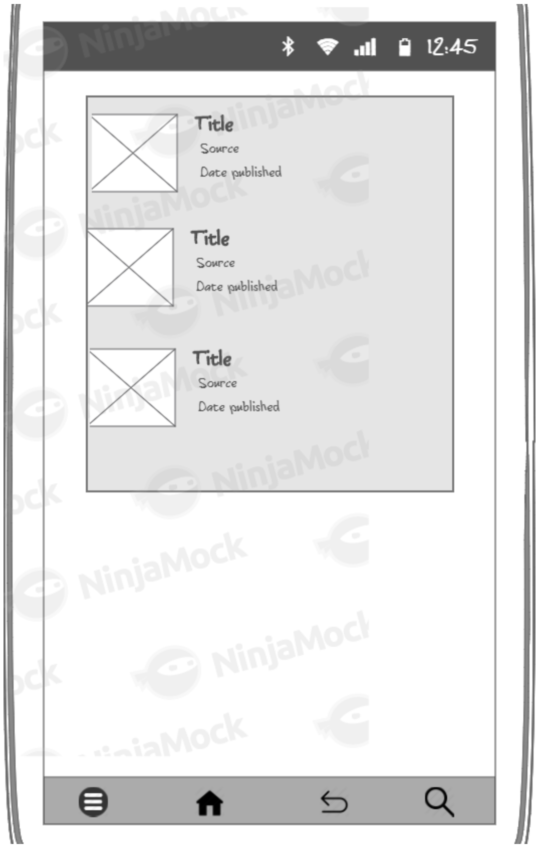
This screen displays the article content, its image, title, author, source name and published date. User can add article to favorite or to share it. The link “Read more” navigates user to the article website. Sliding left navigated to the next article, sliding right navigates to previous article.

Screen 3 Sources Screen



This screen will display a list of News Sources available, so the user will be able to choose if he/she wants to get news from all sources or one particular source.

Screen 4 Favorite articles widget



Favorite articles widget will display 3 (for bigger devises 6) articles as on Main screen. Clicking on article navigates to Article details screen. Clicking on the widget itself navigates to Main screen.

Key Considerations

How will your app handle data persistence?

The app will load articles from News API (https://newsapi.org/) using Retrofit2 library. The local database stores data only for Favorite articles. I will be using Room, LiveData and ViewModel for all database interactions and IntentService to update views in the Favorite articles widget.

Describe any edge or corner cases in the UX.

When the app first starts the Favorite articles screen will be empty, so the app will start with Top News. The Favorite articles screen and Favorite articles widget will have a placeholder for the case when there are no Favorite articles.

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

- Android Design Support Libraries: 27.1.1 – Material based design and other fantastic features
- Gradle: 3.1.4 -- for code compilation..
- Retrofit2: 2.4.0 – Used to download data from the data source asynchronously, simplifies http client and fetching JSON from URL
- Room: 1.1.1 -- for database implementation
- Gson: 2.4.0 -- for mapping Lists of Objects
- Picasso: 2.5.2 -- to handle the loading and caching of images.
- Butterknife: 8.8.1 – Optimize code with DataBinging and save lines code.
- Timber Log.: 4.7.1 -- Optimize code for log files

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

- Google Mobile Ads to display test ads
- Firebase Performance Monitoring to monitor performance 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

:

- Create project
- Configure libraries
- Create paid and free flavors .

Task 2: Implement UI for Each Activity, Fragment and Widget

:

- Build UI for MainActivity
- Build UI for ArticleDetailsActivity
- Build UI for ArticleDetailsFragment
- Build IU for SourcesActivity
- Build UI for SourcesFragment

- Build UI for FavoriteArticlesWidget

Task 3: Implement Favorite Articles Database

Implement Favorite Articles Database using Room, LiveData and ViewModel.

- Create AppDatabase
- Create AppExecutors
- Create ArticleDAO

Task 4: Implement MainActivity

Implement functionality for displaying loaded articles in RecyclerView in MainActivity:

- Create ArticlesAdapter

Task 5: Implement ArticleDetailsActivity

Implement functionality for displaying an article and actions:

- Implement “Add to Favorite” action
- Implement “Share” action
- Implement “Read more” action

Task 6: Implement SourcesActivity

Implement functionality for displaying article sources and letting user choose his/her favorite source:

- Add sources array.
- Display sources list with radio buttons.
- Implement SharedPreferences to store the ID of chosen source.
- Implement Retrofit2 calls for each source.

Task 7: Include Google AdMob

Google AdMob native ads should be included in MainActivity and only in free version.

Task 8: Firebase Performance Monitoring

Firebase Performance Monitoring will be used to monitor performance of the app and efficiency of the network calls.

Task 9: Implement FavoriteArticlesWidget

FavoriteArticlesWidget will display the last 3 (on tablet 6) favorite articles added by the user.

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**"