

How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

[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: vinitagrawal

Bullets

Description

With our busy schedule and growing content over the internet, it gets difficult to read through those articles in our very small time frame. Enter Bullets. An application that summarizes an article into few bullets with only the important and relevant information.

Bullets uses Aylien News API to summarize the news articles into small consumable chunks.

Intended User

Anyone who wants to read news.

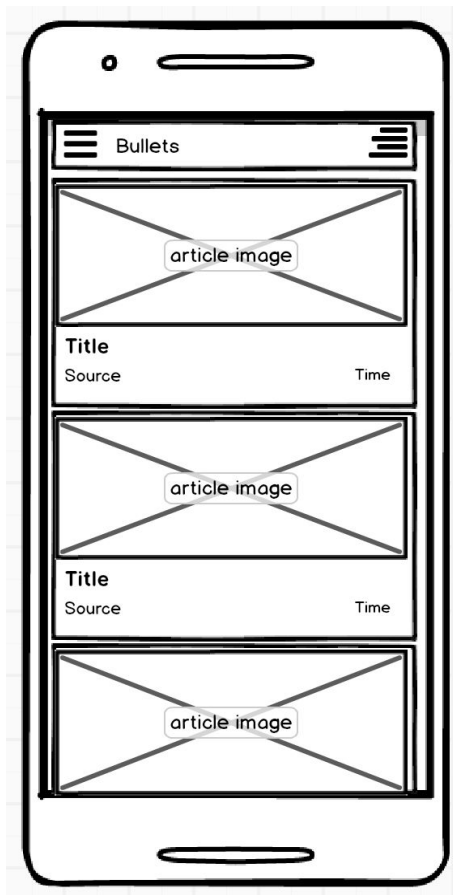
Features

- News summarized into bullet points
- Presents important and relevant information
- Sort articles based on recency or trending

User Interface Mocks

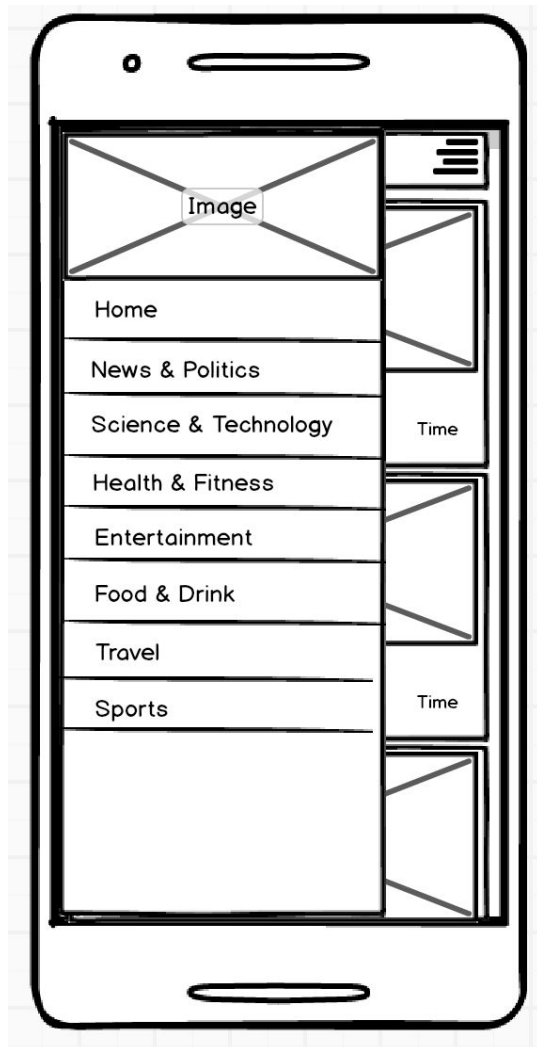
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



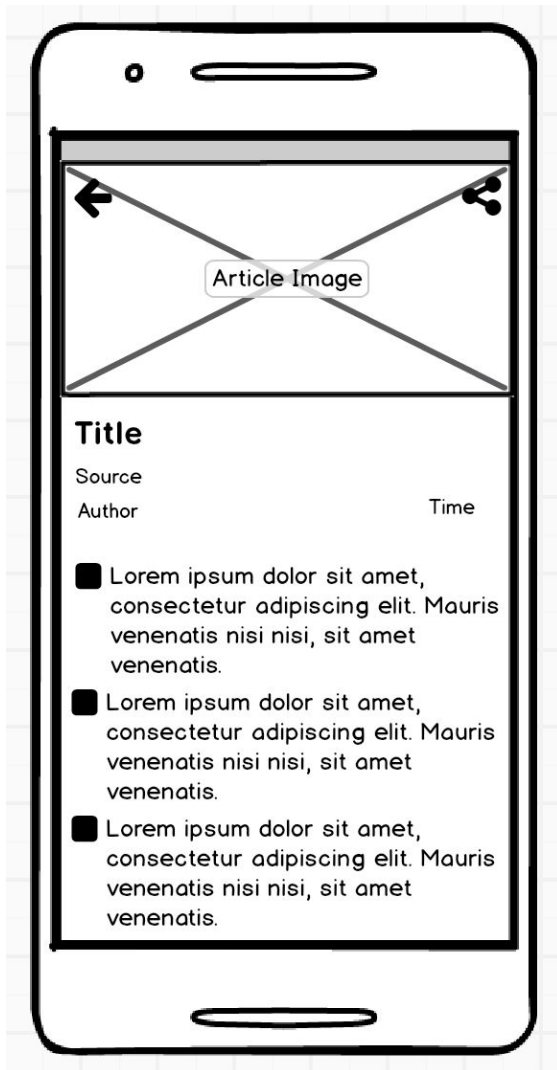
Home Page of the application which shows trending news. User can change sort option using the sort menu button. When user clicks on hamburger icon, navigation drawer opens up.

Screen 2



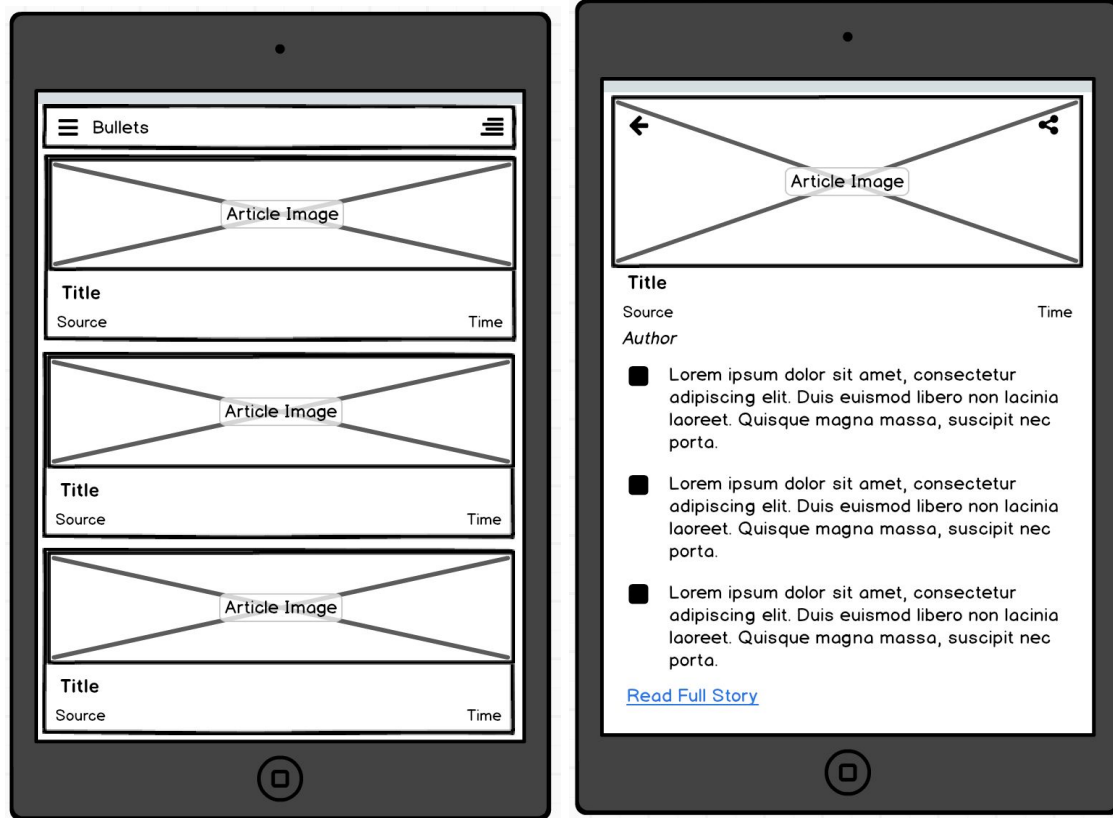
User can switch between categories using the navigation drawer. The home option will show news from all the categories combined.

Screen 3:



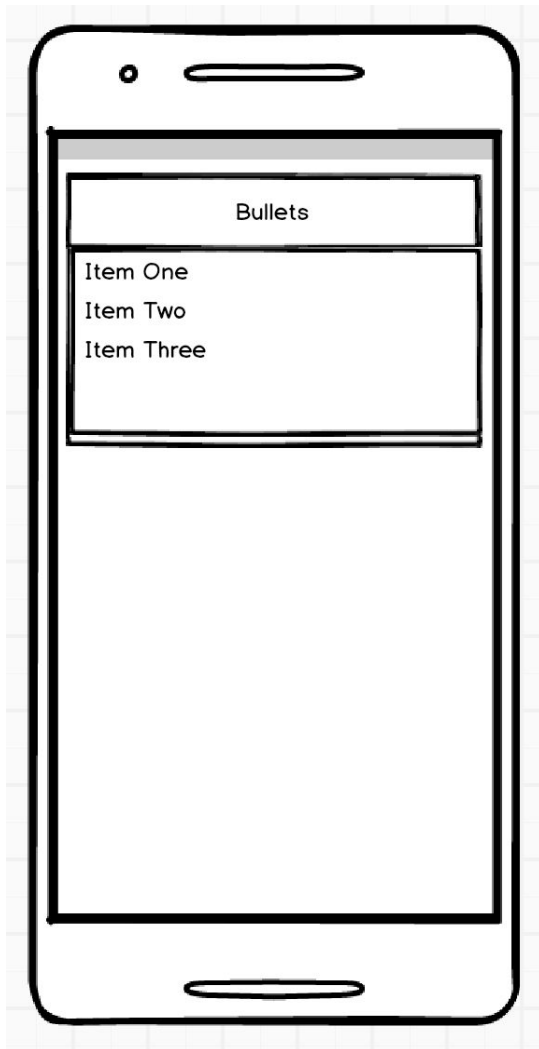
When user taps on an article a detail view like above opens up. This page will contain the summarized version of the article in bullets. There will be a link at the bottom of the page which will take the user to the original article. User can share the article using the share menu button or go back to the previous screen using back menu button or hardware back button.

Screen 4: Tablet



Above is a tablet view of the application which shows trending news. User can change sort option using the sort menu button. When user clicks on hamburger icon, navigation drawer opens up. Clicking on an article opens the detail view screen.

Widget



Widget to display articles in a list view

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

App will handle data persistence using content provider. (I am also searching for a suitable ORM for this purpose)

Describe any corner cases in the UX.

User can switch between categories using navigation drawer.

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

Retrofit - to fetch news from aylien news api

Picasso - to handle the loading and caching of images.

Describe how you will implement Google Play Services.

Admob - to display ads in the application

Analytics - to analyse how many times user interacts with the application

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Setup the project following the below tasks

- Add Retrofit, Picasso and depending libraries
- Add Google Play Plugin for admob and analytics

Task 2: Create Home Screen

- Build UI for HomeActivity with a Navigation Drawer
- Supply Navigation Drawer with different categories
- Create UI for the Home Screen to display articles in a recycler view.
- Add a menu button that allows user to change the sort order
 - Sort order can be by recency or trending

Task 3: Get Articles from Aylien News API

- Implement Retrofit to fetch articles from Aylien News API based on sort order.
- Handle different cases
 - No Internet Connection
 - No Data returned
 - Error message returned

Task 4: Data Persistence

- Implement a content provider.
- Fetch articles from API on demand
- Save the articles in SQLite database.
- Implement loaders to move data to views.

Task 5: Show Articles

- Create UI for RecyclerView to display articles in a list format.
- Each Article must be inside a CardView.
- Each Card will show Article
 - Image
 - Title
 - Source
 - Published Time
- Populate RecyclerView to show articles.

Task 6: Switch between categories

- Allow user to switch between categories using navigation drawer.
- When new category is selected, call the news api to fetch articles for the selected category.
- Refresh the home screen to display articles for that category.

Task 7: View Article

- Allow user to tap on an article and transition to a details screen.
- Create UI for details screen.
- Show additional information on the screen :
 - Media Image
 - Source
 - Author
 - Published time
 - Important and relevant information in bullets.
- Add a share button and implement sharing functionality, use intent extras to share the content.
- Pressing the back button will bring the user back to the home screen.

Task 8: Read full story

- Add a link on details screen to redirect the user to read the full story.
- Full story opens up in a browser.
- Pressing the back button brings the user back to the details screen.

Task 9: Add Analytics and Admob

- Display ads in the application using Admob services from google.
- Integrate Analytics within the application.

Task 10: Create a Widget

- Create a widget to display trending news.

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

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"