

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](#)

[News sources screen](#)

[List of news screen](#)

[Article details screen](#)

[Widgets screen](#)

[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: Implement network part](#)

[Task 4: Implement ContentProvider](#)

[Task 5: Implement Build variants and product flavors](#)

[Task 5: Implement part with Google play services](#)

GitHub Username: OlegLeskinImprove

Find out first!

Description

Find out first! Stay up to date with the latest world news from the business, politics, fashion, games and much more. You no longer need to install a bunch of applications from different news releases, all the news in one application. Share hot news with friends in social networks.

Intended User

The app is for people who value their time and want to stay up to date with the latest developments.

Features

- Sorting of articles
- Sorting of news sources
- Sharing news

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



Start screen with news sources. User can choose only one news sources.

Screen 2



Screen with a list of articles. Articles can be sorted by TOP, LATEST, and POPULAR categories.

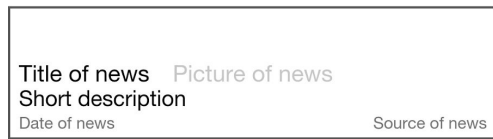
Screen 3



Text of news

Screen with article details. A user can share an article via social networks.

Screen 4



Screen with widgets (4x1 and 2x1). The widget can show the latest news from last user news source.

Key Considerations

How will your app handle data persistence?

I will create own Content Provider. I will get content from following API (<https://newsapi.org/>).

Describe any corner cases in the UX.

Application hasn't any corner cases.

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

Application will be using following libraries:

- **Glide** - to handle the loading and caching of images;
- **Retrofit** - for network operations;
- **Palette** - for generation color on Article details screen;
- **Butterknife** - for inflating views;
- **net.simonvt.schematic** - for generation ContentProvider.

Describe how you will implement Google Play Services.

I will use Google ads and Google analytic services. Google ads will be used only for the free version of the app. With Google analytics, I can receive information about popular sources and categories of news, and then push them to the fore.

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

Create project and configure:

- Create project in Android studio;
- Configure all libraries.

Task 2: Implement UI for Each Activity and Fragment

Build all UI for screens:

- Build UI for NewsSourcesActivity;
- Build UI for NewsListActivity;
- Build UI for ArticleActivity;

Task 3: Implement network part

- Create POJO models;
- Create API interface (using Retrofit);
- Build requests for API;
- Implement Sync Adapter for updating and caching data at regular intervals;
- Implement Loaders for NewsSourcesActivity, NewsListActivity and ArticleActivity.

Task 4: Implement ContentProvider

Build ContentProvider for the application using **net.simonvt.schematic** library.

Task 5: Implement Build variants and product flavors

- Implement **release** and **debug** build variants. Attach keystore for release variant;
- Implement **free** and **pay** product flavors.

Task 6: Implement part with Google play services

- Create project in Google analytic service;
- Implement sending of events in the app;
- Implement Google ads banner on News sources screen (only for free flavor).

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**"