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: konstantinosKech

News App

App is written solely in the Java Programming Language.

App utilizes stable release versions of all libraries, Gradle, and Android Studio.

Android Studio	3.1.3
JRE	1.8.0_253
Gradle	4.4
Firebase-Messaging	com.google.firebase:firebase-messaging:17.3.0

Analytics	firebase-core:16.0.1
Schematic	net.simonvt.schematic:schematic:0.6.3
Android Target SDK version	26
RecyclerView	com.android.support:recyclerview-v7:26.1.0
Preference	com.android.support:preference-v7:26.1.0
AppCompat	com.android.support:appcompat-v7:26.1.0
Firebase	firebase-core:16.0.1

App includes support for accessibility. That includes content descriptions, navigation using a D-pad, and, if applicable, non-audio versions of audio cues.

App validates all input from servers and users. If data does not exist or is in the wrong format, the app logs this fact and does not crash.

App uses async task for image loading and for loading data from content provider.

App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

App keeps all dimensions in a dimens.xml file.

Each service imported in the build.gradle is used in the app.

App creates only one analytics instance.

App theme extends AppCompat.

App uses an app bar and associated toolbars.

App uses standard and simple transitions between activities.

Description

The app allows you to select the source from which you want to receive news. The user subscribes to the channels he wishes and the news from this source appear in his home screen.

Intended User

Every one who reads the news.

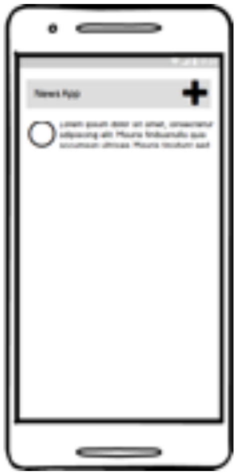
Features

- User sees news in the main screen.
- Allows the user to subscribe in multi news channels.
- Lets user unsubscribe from news channels.
- User can see news in home screen by using the widget.

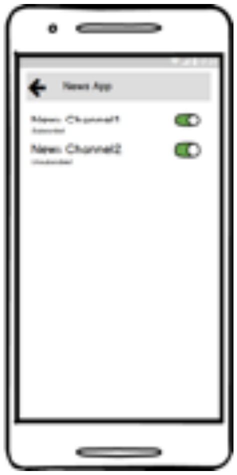
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

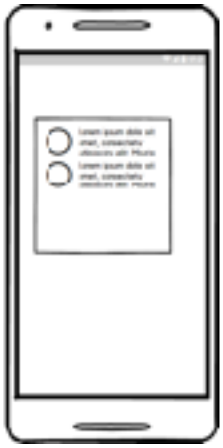
Screen 1



Screen 2



Screen 3



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 use Content provider to keep news from selected sources.
- App will use Firebase Cloud Messaging in order to provide news from different sources.

Describe any edge or corner cases in the UX.

The app will return to the home screen hitting the back button when the user will select at least one source.

Big articles will have a read more button that will transfer user from the app to the article page.

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

- ButterKnife for binding fields.
- Firebase for the Cloud Messaging.
- Google Analytics to get insights.
- Schematic for the content provider.

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

App uses Firebase Cloud Messaging and Analytics.

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 Libraries.
- Configure Firebase.
- Configure Gradle.

Task 2: Implement UI for Main Activity

- Implement Recycler View for Main Activity.
- Implement Main Activity UI.
-

Task 3: Implement UI for Main Activity

- Implement Source Selection Activity UI.
- Implement shared-preferences to keep choices.
- Implement Source Selection Activity Fragment.

Task 4: Content Provider

- Implement Content Provider.

Task 5: FCM

- Implement Firebase Cloud Messaging.

Task 6: Google Analytics

- Implement Google Analytics.

Task 7: Home Screen Wigdet

- Implement home screen widget.

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**"