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: andaaraus007

49ers.com RSS Reader

Description

This app is an RSS reader specifically for the 49ers.com website. When there are articles that you want to save to read later, there is no easy way to save them. I would like to create an app that will allow the user to save the articles they are interested in. This project will be solely written in the Java Programming Language. The app will utilize stable release versions of all libraries, Gradle and Android Studio.

Android Studio	4.0.1
Gradle	6.1.1
appcompat	1.2.0
constraintlayout	2.0.1

Material	1.2.1
com.squareup.retrofit2	2.9.0
room	1.1.1
recyclerview	1.1.0

App will integrate Admob for monetization and Analytics for crash reporting.

Intended User

The intended users are the 49ers fanbase.

Features

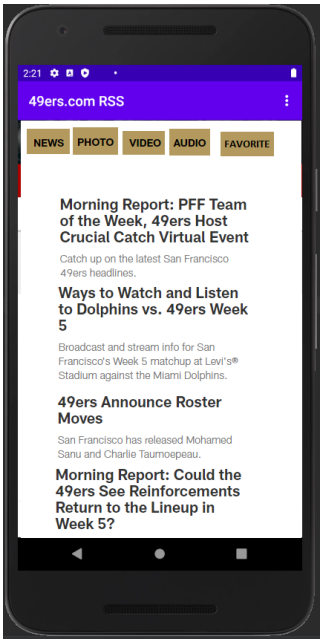
List the main features of your app. For example:

- View items from the following RSS feeds from 49ers.com
    - News
    - Photo
    - Video
    - Audio
- Save items into a group called Favorites

User Interface Mocks

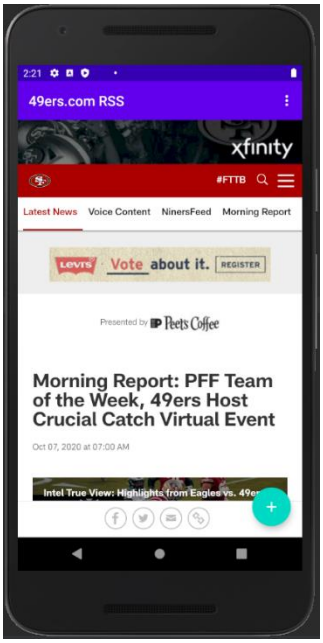
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



This is the main screen for the app. On start up, the News feed will be displayed. User will be able to swipe left/right to navigate between the different feeds. The last tab is the favorites which contains all the Favorited articles. When the user clicks on an item, the detailed view is shown.

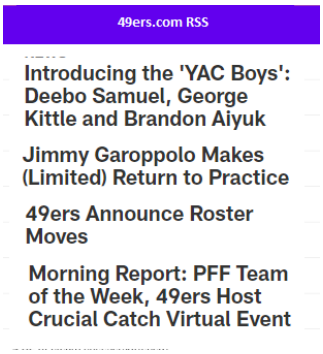
Screen 2



In this view, the use can read the article or save it for later viewing by clicking the FAB. The '+' will turn to an 'x' if the article was previously saved (favorited). Clicking on the 'x' will remove the article from the favorites list

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

Screen 3



This is the companion widget. The widget will display the headline of all the saved articles. When the user clicks on a headline, The app will open to the detailed view for that article. IntentService will be used to fetch data and display it.

Key Considerations

How will your app handle data persistence?

All favorited items will be stored using ROOM.

Describe any edge or corner cases in the UX.

This is a fairly straight forward app. There should not be any edge or corner cases in the UX

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

Retrofit will be used to handle the http requests and converter-simplexml to parse the response. ROOM will be used for persistent storage.

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

I will be using the following end points for the RSS feeds

- https://www.49ers.com/rss/news
- https://www.49ers.com/rss/galleries
- https://www.49ers.com/rss/videos
- https://www.49ers.com/rss/audio

.

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 the project using an Empty Activity
- Add Internet permissions in the Manifest
- In the app build.gradle file add the following libraries
    - Appcompat
    - Recyclerview
    - Cardview
    - Retrofit2
    - Room
    - Design
    - Constraint-layout

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
    - Build Fragment for content on main page
        - Navigation tabs on top
        - List of articles in body
- Build UI for Detail Activity
    - Build Fragment for content on detail page
        - Use webview to display article
        - Add FAB to bottom right
            - Action is to add or remove item in database
- All strings will be kept in strings.xml and enables RTL layout switching on all layouts.

Task 3: Implement network call

Implement network call to endpoint for news feed

- Use retrofit to make network call

- Store items in XML data into an item model

Task 4: Recyclerview Adapter

Create adapter for recycleview to display data on main activity

Task 5: Navigation Tab

Handle click event from topic tabs as well as swipe left/right to load list of items.

Task 6: Implement Detail Activity

Handle click event from item to launch Detail activity.

Task 7: Create database to store favorited items

Handle click event from topic tabs as well as swipe left/right to load list of items.

- Implement Room to store items
- If no new item is stored in database, there shall be no unnecessary calls made to the database.

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**"