

ITIS/ITCS 4180 Mobile Application Development Midterm Exam

Basic Instructions:

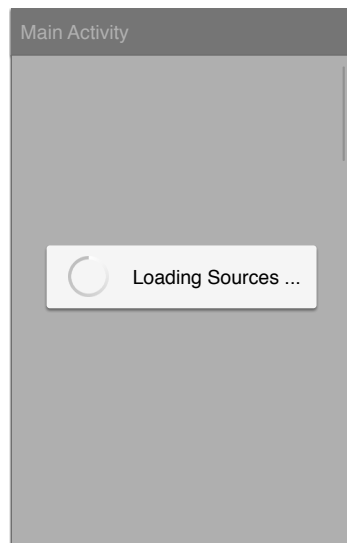
1. This is the Midterm Exam, which will count for 20% of the total course grade.
2. This Midterm is an individual effort. Each student is responsible for her/his own Midterm and its submission.
3. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
4. During the exam, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other online persons.
5. Answer all the exam parts, all the parts are required.
6. Please download the support files provided with the Midterm and use them when implementing your project.
7. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
8. Export your Android project and create a zip file which includes all the project folder and any required libraries. The file name is very important and should follow the following format: **800#_Midterm.zip**. Submit the exported file using the provided canvas submission link.
9. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
10. **The required Android Virtual Device (AVD) should have minimum SDK version set to 20 and target SDK at 25.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Midterm Exam (100 Points)

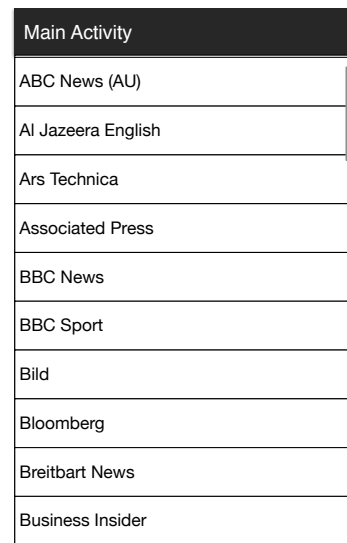
In this midterm exam assignment you will develop a news app. The app will enable the user to select the news source and will display the news for the selected news source. The api used for this assignment is the News API <https://newsapi.org/> .

API Setup:

1. Go to <https://newsapi.org/register> and to setup a new account and get an API key.
2. You are provided with a Postman file that includes the sources and articles apis to be used in this app.
3. In the PostMan file, Make sure to update the articles api **to use your API key**.



(a) Main Activity loading the news sources.



(b) Main Activity ScrollView listing news sources.

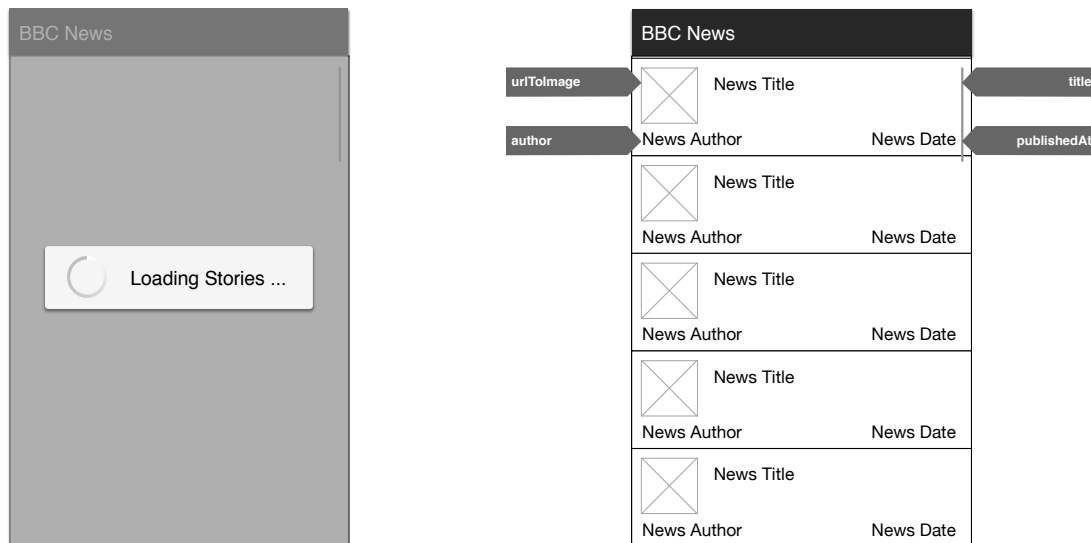
Figure 1, Main Activity Wireframe

Part 1: Main Activity (35 points)

The **Main** activity is responsible for the loading and displaying the different sources available at NewsAPI. The requirements are as follows (Figure 1):

1. Upon loading the Main Activity, the app should connect to the "Sources" NewsAPI api to retrieve the JSON list of the sources provided by the NewsAPI.
 - a) Create a "Source" class containing variables: id and name, which should be populated from the retrieved JSON.
 - b) The app should use a child thread (or AsyncTask) to perform data retrieval and data parsing. The parsed JSON should return a list of Source objects. The list of news sources should be then returned to the Main Thread to be displayed.

- c) Display a Progress Dialog during the download and parsing of the JSON and dismiss it once these operations are completed as indicated in Figure 1(a).
2. The Source items should be displayed in a list, this list should be created using a combination of ScrollView and Linear Layout or RelativeLayout, where each row item includes a TextView. See Figure 1(b). You are also free to use any other approach that provides scrollable interfaces such as ListView or RecyclerView.
3. Upon clicking a list item it should open the News Activity, you should also pass the news source *Source* object to the News Activity.



(a) Loading Stories in News Activity

(b) ScrollView listing the news items for BBC News

Figure 2, News Activity Wireframe

Part 2: News Activity (65 points)

This activity displays the top news items for the selected news source. The requirements are as follows:

1. Update the activity title to display the name of the name of the news source.
2. Upon loading the News Activity, the app should connect to the “Articles” NewsAPI api to retrieve the JSON list of the articles for the selected source.
 - a) Create an “Article” class containing variables: author, title, url, urlToImage and publishedAt, which should be populated from the retrieved JSON.
 - b) The app should use a child thread (or AsyncTask) to perform data retrieval and data parsing. The parsed JSON should return a list of Article objects. The list of news articles should be then returned to the Main Thread to be displayed.
 - c) Display a Progress Dialog during the download and parsing of the JSON and dismiss it once these operations are completed as indicated in Figure 2(a)

3. The Article items should be displayed in a list, this list should be created using a combination of ScrollView and Linear Layout or RelativeLayout, see Figure 2(b). You are also free to use any other approach that provides scrollable interfaces such as ListView or RecyclerView.
 - a) The image for each news item should be downloaded using a child thread (or AsyncTask).
4. Upon clicking a news item it should open the url for the selected news article in the browser.

Grading Key

Topic	Grade %
Main Activity: Parsing the sources api in a background thread.	12
Main Activity: Displaying and dismissing the progress dialog correctly.	5
Main Activity: Displaying the sources list.	13
Main Activity: Responding to item clicked.	5
News Activity: Displaying the title correctly.	5
News Activity: Parsing the articles api in a background thread.	20
News Activity: Displaying and dismissing the progress dialog correctly.	5
News Activity: Displaying the articles list.	15
News Activity: Displaying the images for articles list.	10
News Activity: Responding to item clicked.	10
Total	100