

ITIS/ITCS 5180 Mobile Application Development
Homework 4

Basic Instructions:

1. In every file submitted you **MUST** place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
 - a. Only a single group member is required to submit on Canvas for each group.
 - b. The file name is very important and should follow the following format:
Group#_HW04.zip
 - c. You should submit the assignment through Canvas: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**
8. **The required Android Virtual Device (AVD) should have minimum SDK version set to 20 and target SDK at 25.**

Homework 04 (100 Points)

In this assignment we are going to develop a news application using the XML RSS feeds provided by CNN. The urls for the different news categories are shown in the below table.

Category	XML RSS feed
Top Stories	http://rss.cnn.com/rss/cnn_topstories.rss
World	http://rss.cnn.com/rss/cnn_world.rss
U.S.	http://rss.cnn.com/rss/cnn_us.rss
Business	http://rss.cnn.com/rss/money_latest.rss
Politics	http://rss.cnn.com/rss/cnn_allpolitics.rss
Technology	http://rss.cnn.com/rss/cnn_tech.rss
Health	http://rss.cnn.com/rss/cnn_health.rss
Entertainment	http://rss.cnn.com/rss/cnn_showbiz.rss
Travel	http://rss.cnn.com/rss/cnn_travel.rss
Living	http://rss.cnn.com/rss/cnn_living.rss
Most Recent	http://rss.cnn.com/rss/cnn_latest.rss

Table 1. XML RSS feeds

Connecting with the API

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, and strings.xml to create the user interface. Perform the following tasks:

1. This app should use the XML RSS feeds listed in Table 1 based on the category selected by the user.
2. Clicking on the “Go” Button should display a list of categories as shown in Figure 1. You can either Alert Dialog or Spinner to implement it.
3. Clicking on a category should do the following:
 - a) The TextView will hold the search category clicked by the user.
 - b) Use AsyncTask/Thread to connect to the corresponding XML RSS feed URL to retrieve and parse the XML returned for the selected category. The AsyncTask/Thread class should be in a separate file/class not inner to the main thread. i.e. You should manage passing parameters to the class and then pass back the result image downloaded to the UI.
4. When the api data is retrieved:

- a) Display the first news item from the list of retrieved news items as shown in Figure 1.
 - b) Use Picasso to retrieve and display the first image associated with the displayed news item and display it in the ImageView. The API will return several images for each news item, display only the first image for the displayed news item.
5. The Next and Previous news item icons should be disabled when the application is launched, and enabled after the first news item is displayed. The buttons will also remain disabled in the case there is only 1 or 0 news items corresponding to the provided category. Use icons provided in Support Files for setting the image icons (**next.png, prev.png**)
 6. **Do not** store the photos, simply download and display the retrieved photos. Also do not attempt to download all the URLs receive, and your app should only download and display a single news photo at any given time.
 7. Upon clicking the “Next” icon, you should display the next news item from the retrieved news items and download the news photo accordingly. If the currently displayed news item happens to be the last news item, you should display the news item at index 0 (first news item) and download the news photo accordingly.
 8. Clicking the “Previous” icon, you should display the previous new item from the retrieved news items and download the news photo accordingly. If the currently displayed news item happens to be the first news item, you should display the last news item and download the news photo accordingly.
 9. Notice the “1 out of 20” text at the bottom of the screen, this should be updated according to the currently displayed news item.
 10. If the api returns an empty list of news items then clear the currently displayed news item and display an error Toast message “No News Found”.
 11. Your application should connect with the api and download the requested photo only if there is an established internet connection. If there is no internet connection you should display a Toast message indicting that there is no internet connection and do not attempt to send the HTTP request.
 12. Upon clicking the image or the news title, open the news URL for the displayed news item in the emulator’s browser.

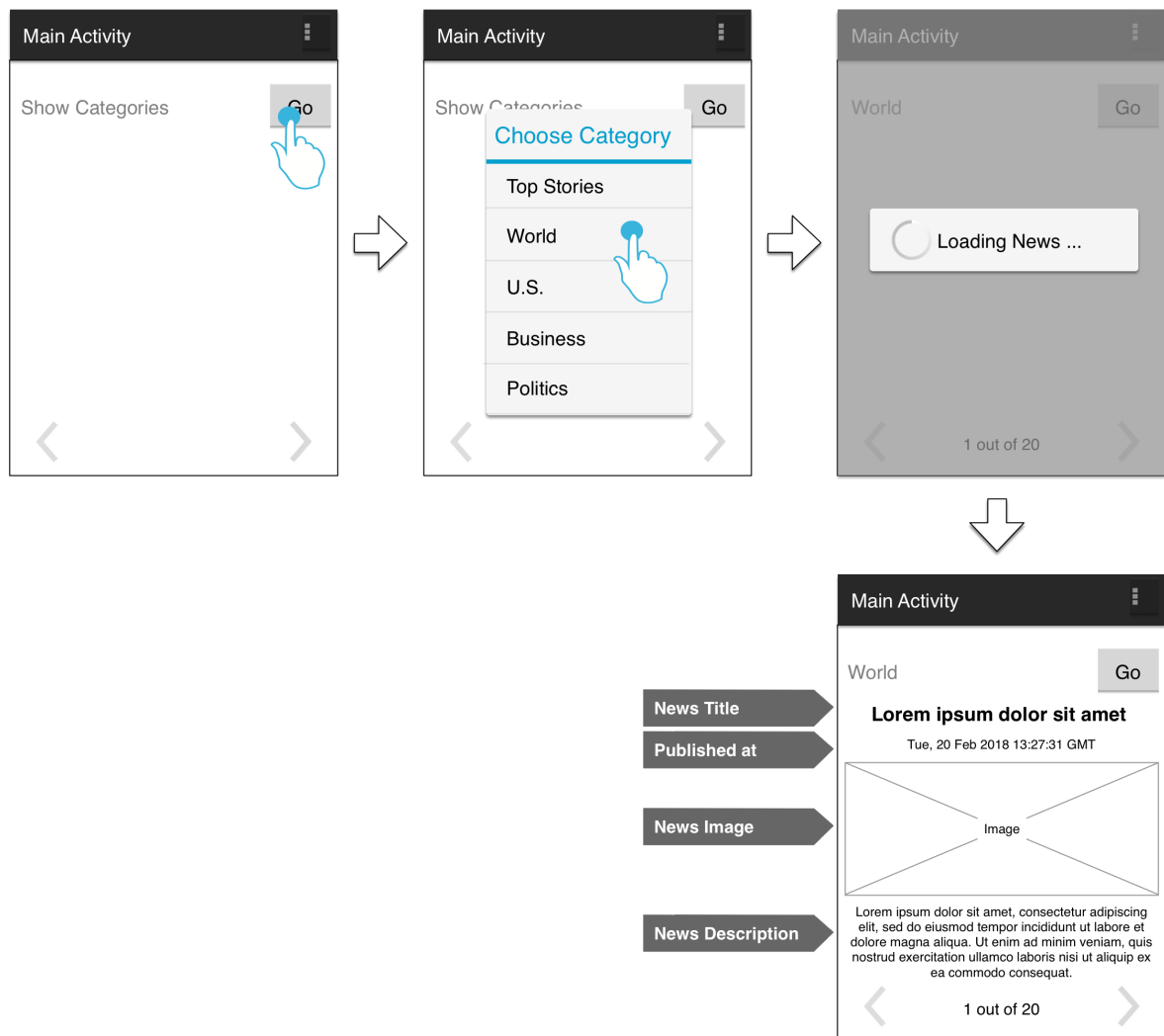


Figure 1, Application Wireframe