

## ITIS/ITCS 4180/5180 Mobile Application Development

### Homework 4

Date Posted: 02/16/2015 at 21:00

Due Date: 02/26/2015 at 23:55

---

#### 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 as follows:
  - a. From eclipse, choose "*Export...*" from the File menu.
  - b. From the Export window, choose *General* then *File System*. Click *Next*.
  - c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
  - d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
  - e. When exporting make sure you select *Create directory structure for files*.
  - f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
6. Submission details:
  - a. All the group members should submit the same zip file.
  - b. The file name is very important and should follow the following format:  
**Group#\_HW04.zip**
  - c. You should submit the assignment through Moodle: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

## Homework 4 (100 Points)

In this assignment you will develop a simple NPR client application (see <http://www.npr.org/api/index>). The application should be able to load information NPR programs, topics and stories. Users of the application will be able to add/remove stories to their local favorites. The favorites should be stored using Shared Preferences. **Do not use SQLite in this assignment;** the assignment is designed to give you more experience with Shared Preferences. For this homework you will be using a JSON parser as data loaded from the NPR API should be in the JSON format.

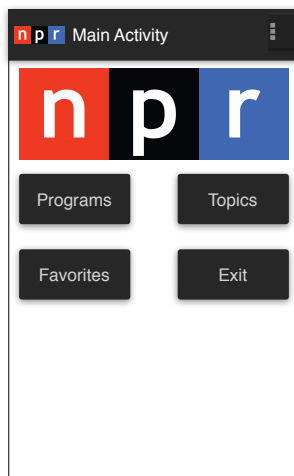
### Notes:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 19**. The app should display correctly on Nexus 5 sized device. **Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.**
2. All image downloading should be performed using Threads (or AsyncTask) and your code should not block the main thread.
3. For further information about the NPR api check the documentation available at the <http://www.npr.org/api/index>. You will need to apply for an api key.

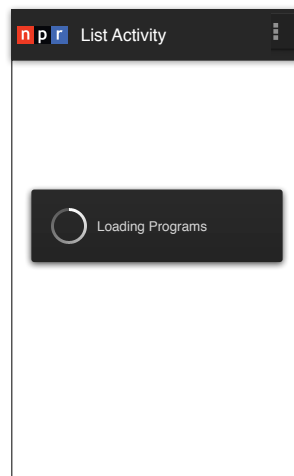
### Part A: Main Activity

The Main Activity wireframe is shown in Figure 1(a). The Main Activity provides the user with 4 buttons, which include, “Programs”, “Topics”, “Favorites” and “Exit”.

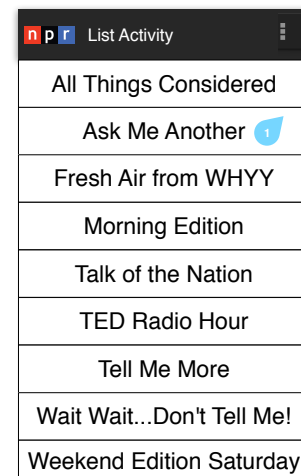
- 1) Tapping on the Exit button should exit the entire application.
- 2) Tapping on the “Programs” should start the List Activity and display the list of programs. Tapping on the “Topics” should start the List Activity and display the list of topics.
- 3) Tapping on the “Favorites” should start the Stories Activity and display a list of favorites stored in device’s the Shared Preferences.



(a) Main Activity Starting



(b) List Activity Loading



(c) Populated List View

Figure 1, The Main Activity and List Activity Wireframes

## **Part B: List Activity (20 Points)**

The List activity is responsible for the loading the list of programs or topics based on the selected option in the Main activity.

- 1) If “Topics” was selected then the topics list should be loaded by parsing the list returned from the NPR API: <http://api.npr.org/list?id=3002&output=JSON>
- 2) If “Programs” was selected then the program list should be loaded by parsing the list returned from the NPR API: <http://api.npr.org/list?id=3004&output=JSON>
- 3) The content of the JSON API should be retrieved by establishing a HTTP connection to the service. All the parsing and HTTP connections should be performed by a worker thread or an AsyncTask and should not be performed by the main thread. While, the list is being loaded you should indicate that using the progress alert as indicated in Figure 1(b).
- 4) The loaded list should be displayed in a ListView, as shown in Figure 1(c).
- 5) Clicking a list item should display the NPR stories for this program or this topic by starting the Stories Activity.

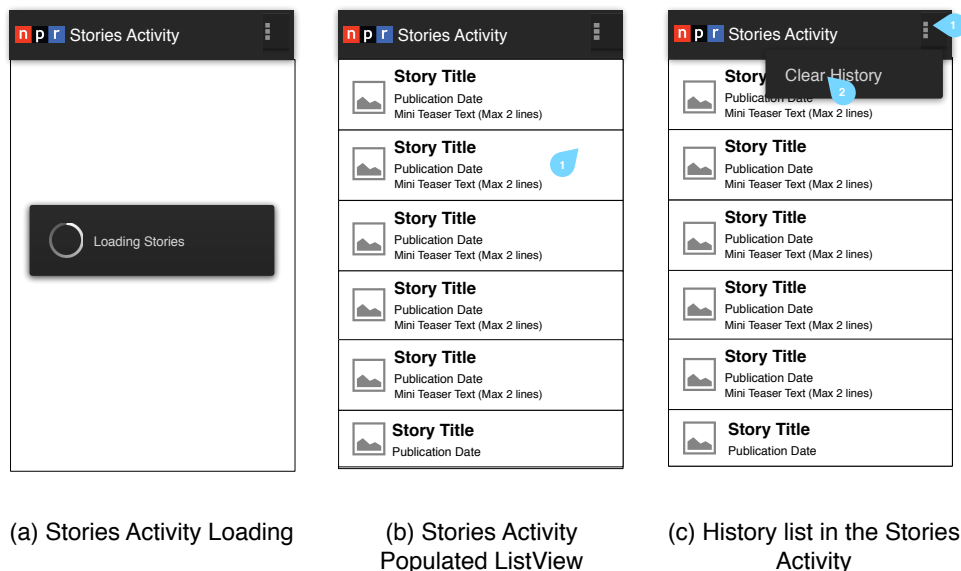


Figure 2, The Stories Activity Wireframe

## **Part C: Stories Activity (40 Points)**

The Stories activity is responsible for the loading the list of stories based on the topic or program selected in the List activity or if the favorites button is selected in the main activity.

- 1) For the topic or programs:
  - a) The List activity should send the Stories activity the id of the topic or program for which the stories list should be displayed.
  - b) Use the query api (<http://api.npr.org/query>) provided by the NPR API to retrieve the list of stories for the topic or program. Configure the API to request at least 25 stories and the format is JSON. Use the query generator (<http://www.npr.org/api/queryGenerator.php>) to get better idea of how to use the API.

- c) The content of the JSON API should be retrieved by establishing a HTTP connection to the service. All the parsing and HTTP connections should be performed by a worker thread or an AsyncTask and should not be performed by the main thread. While, the list is being loaded you should indicate that using the progress alert as indicated in Figure 2(a).
  - d) The loaded list should be displayed in a ListView, as shown in Figure 2(b). Note that for each story the list item should display the title, publication date, thumbnail, and the mini teaser limited to two lines. For image loading/caching you must use the Picasso library.
    - a) Main Information: <http://square.github.io/picasso/>
    - b) If the image is present in the cache it should be retrieved from the cache and displayed. If the image is not in the cache then it should be downloaded, displayed and loaded in the cache.
  - e) Clicking a list item should display the story details by starting the Story Activity.
- 2) For the favorites:
- a) Stories Activity could also be activated by the user selecting “Favorites” in the Main Activity. In this case, the Stories Activity should retrieve the list of favorite stories from the Shared Preferences.
  - b) The loaded list should be displayed in a ListView, as shown in Figure 2(b). Note that for each story the list item should display the title, publication date, and the mini teaser limited to two lines.
  - c) Clicking a list item should display the story details by starting the Story Activity.

#### **Part D: Story Activity (40 Points)**

The Story activity will display more details related to the story selected in the Stories activity. The Story activity wireframe is shown in Figure 3.

- 1) The story title, reporter’s name, date aired, length of the broadcast audio (mins and secs) and the complete teaser text.
- 2) The activity should also indicate if the loaded story is either in the favorite list or not. If the story is not in the favorites this should be indicated by using the icon in Figure 3(b), and if it is in the favorites it should be indicated using the icon in Figure 3(c). Clicking not in favorites icon should add the current story to the favorites and should change the icon to the in favorites icon. Similarly clicking the in favorites icon should remove the current story from the favorites and should change the icon to the not in favorites icon. You can get these icons from the android icon pack available at [http://developer.android.com/downloads/design/Android\\_Design\\_Icons\\_20120814.zip](http://developer.android.com/downloads/design/Android_Design_Icons_20120814.zip)
- 3) Upon clicking the back icon (left arrow icon), the Story activity should finish and show the previous activity.
- 4) Upon clicking on the globe icon, an intent should be used to open the story’s website in devices web browser.
- 5) Clicking on the speaker icon should show the Media Controller at the bottom of the activity as indicated in Figure 4. Note that the Media Controller should enable the user to play, pause, rewind and fast forward the audio associated with the displayed story. The audio url can be retrieved from the NPR story api. The JSON story object contains an audio object that contains information about the audio stream for the given story, make sure to select the url for the MP3 stream. The MP3 stream url does

not directly point to the stream instead the MP3 stream provided has to be called in order to get the actual stream url. The android media player does not support re-direct, so it has to be provided with the actual stream url. In that case, you should first get the MP3 stream url from the API, then you should make a GET request to this url. The response text will be the streaming url which should be provided to the media player. The support files provide example code for managing Audio in Android.

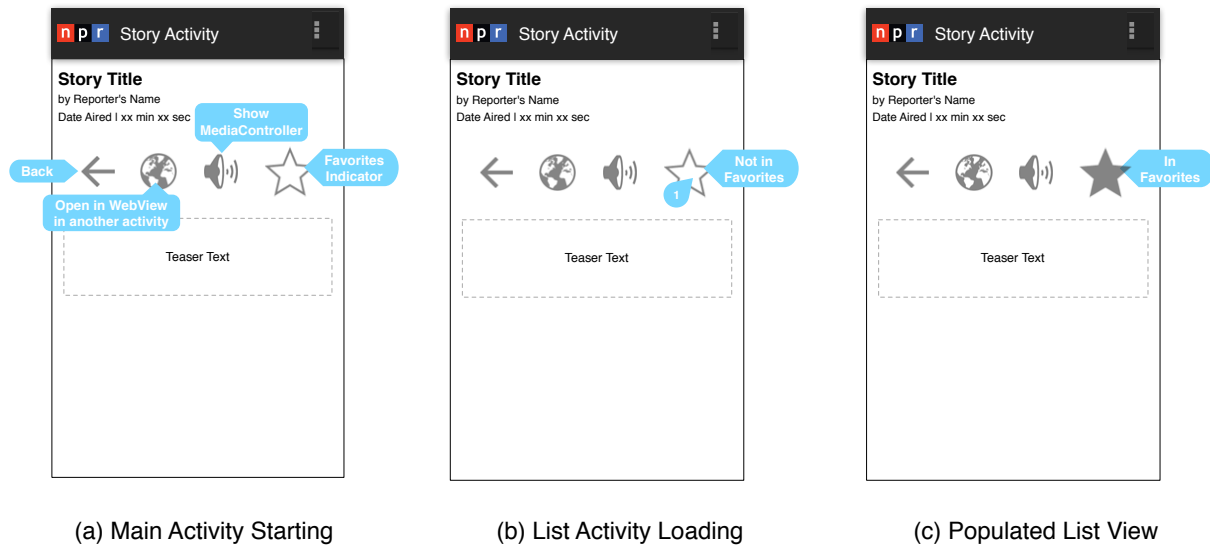


Figure 3, The Story Activity Wireframe

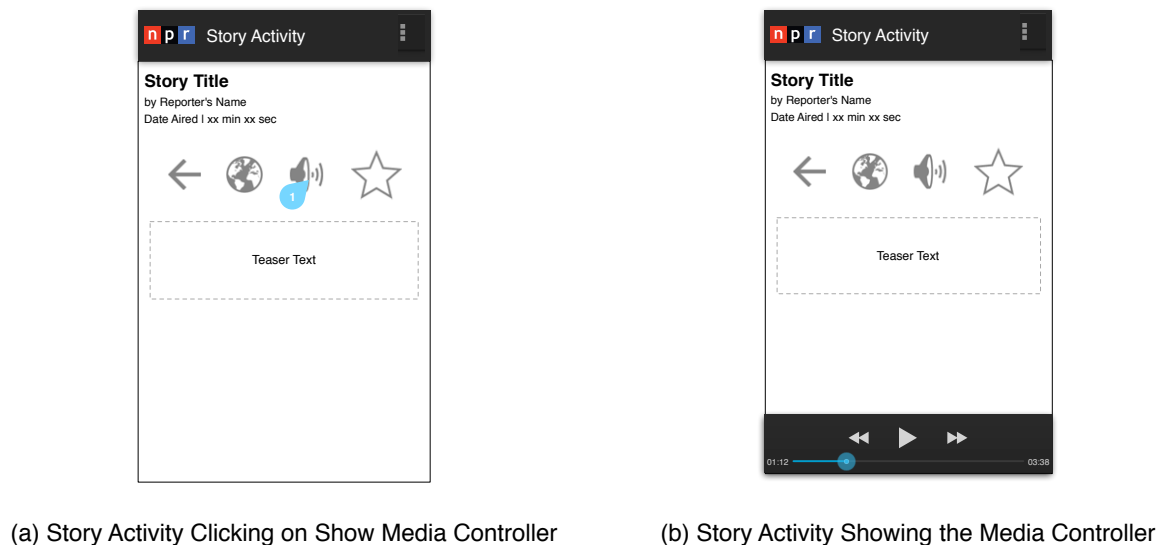


Figure 4, The Story Activity Wireframe