ITIS/ITCS 4180/5180 Mobile Application Development
Homework 4
Date Posted: 09/28/2015 at 22:00
Due Date: 10/05/2015 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 loose 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 moodle 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 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 “iTunes Feed RSS App”. The App displays media feeds from iTunes Store related to latest content release details like AudioBooks, Movies , iOS apps, etc. In this assignment, you will learn how to parse JSON media feeds, using threads , and dynamic UI management.

Part A: Main Activity (10 Points)

The launching activity is the Main Activity, which is shown in Figure 1. The activity displays a list of the Media Types provided by iTunes Store. The URLs for the different JSON streams is shown in Figure 2. The requirements are as follows:

1. The list should be created using combination of ScrollView and LinearLayout, each row could be a TextView. You should not use a ListView in this assignment. The goal is to expose you to the managing a list with out using the ListView component.
2. Clicking on any category should start the “MediaListActivity”, passing it the url of the selected category.

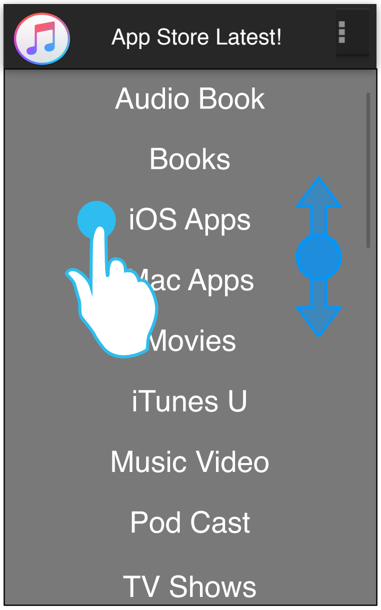


Figure 1, Main Activity Wireframe.

Type	URL
Audio Books	https://itunes.apple.com/us/rss/topaudiobooks/limit=25/json
Books	https://itunes.apple.com/us/rss/topfreeebooks/limit=25/json
iOS Apps	https://itunes.apple.com/us/rss/newapplications/limit=25/json
iTunes U	https://itunes.apple.com/us/rss/topitunesucollections/limit=25/json
Mac Apps	https://itunes.apple.com/us/rss/topfreemacapps/limit=25/json
Movies	https://itunes.apple.com/us/rss/topmovies/limit=25/json
Podcasts	https://itunes.apple.com/us/rss/toppodcasts/limit=25/json
TV Shows	https://itunes.apple.com/us/rss/toptvepisodes/limit=25/json

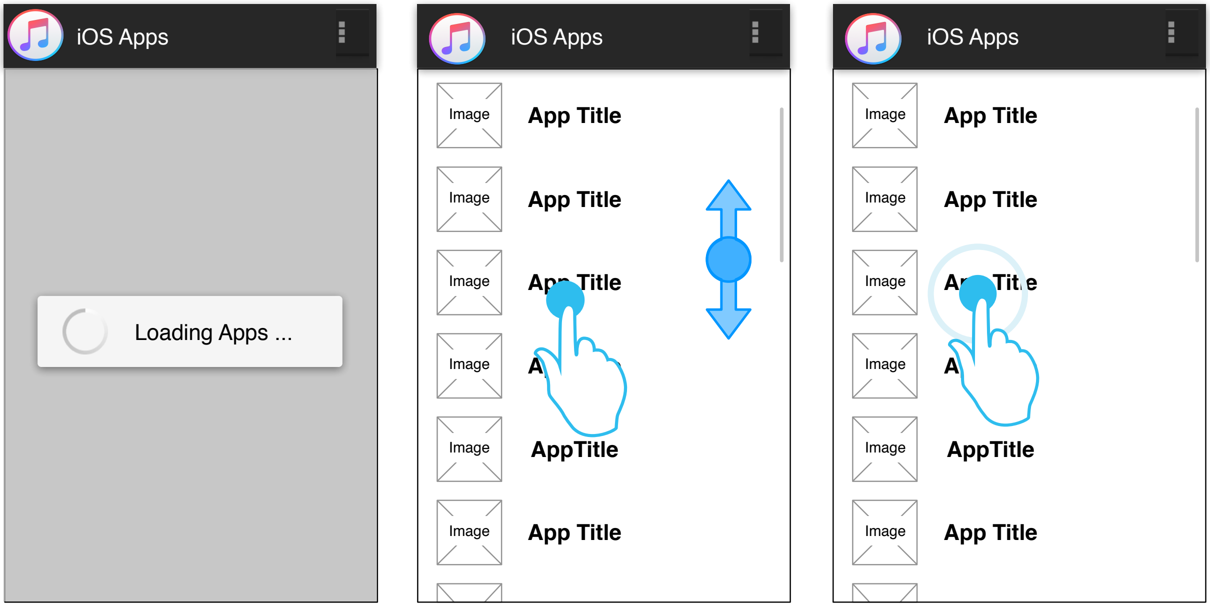
Figure 2, Request URLs

For each item we are interested in the price, currency, small and large images, artist (label), title (label), link (url) to media, Category (label), and release date (label).

Part B: Loading and Parsing RSS Feed (MediaListActivity) (60 Points)

RSS Feed Activity should display brief media feed of the selected category. The url to the rss feed for the selected will be provided by the Main Activity (in the extras). The requirements are as follows:

1. The MediaListActivity should be used for all Media Types, Activity Title should be adjusted to reflect the selected Media Type.
2. Each media item includes the media title, artist (composer), price, currency, release date and many other fields. In addition, each media item includes a thumbnail, under im:image, providing both a small and large thumbnail urls (according to height property). **Note : Every Media Type will have slightly different JSON response structure.**
3. The Media List Activity should use the Android Shared Preferences to store the previously retrieved media items for the different categories. Check this tutorial: http://developer.android.com/guide/topics/data/data-storage.html#pref
   (a) If the requested media category has not been retrieved previously or the version stored for this media category is older than 2 minutes, then the media category data should be downloaded using the provided url. Hint: In the Shared Preferences you can store the whole JSON retrieved previously or you can store a reduced version based on only the retrieved media items or you can store a serialized version of the media items retrieved.
   (b) If the requested media category has been retrieved in the past 2 minutes, then the media category data should be retrieved from the version stored in the Shared Preferences. In this case, no network access is required to retrieve the JSON feed for the selected category.
4. Use thread or AsyncTask to establish a connection to iTunes server using the url passed by the previous activity. All the JSON parsing and network connections should be performed by the child thread.
5. Display a Progress Dialog during the download and/or parsing of the JSON and dismiss it once these operations are completed as indicated in Figure 3(a).
6. Parse the JSON stream and return the result to the Media List. The media items should be displayed in a list, this list should be created using combination of ScrollView and LinearLayout, each row could be a LinearLayout containing an ImageView and a TextView. You should not use a ListView in this assignment. The goal is to expose you to the managing a list with out using the ListView component. See Figure 3(b).
7. In the media items list, for each media item display it's small thumbnail image and the media title. See Figure 3(b). A separate thread or AsyncTask should be used to retrieve the media thumbnail. You can use Picasso Library to retrieve and display the thumbnails. Check http://square.github.io/picasso/
8. The media item row should be clickable to provide the following functionality:
   (a) Clicking a media item, should start the DetailedMediaActivity which provides more details about the current media item.
   (b) A LongClick on a media item, should delete the media item from the list and update the UI accordingly. You should not redraw the entire list, instead only remove the selected media item from the UI. In addition, the removed row should also be removed from the stored cache in the Shared Preferences.



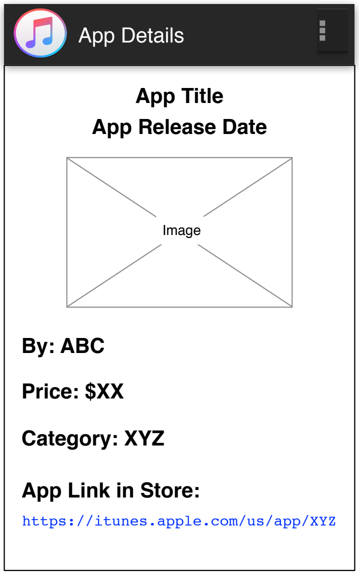
(a) Progress Dialog (b) Clickable and Scrollable (c) Long Click

Figure 3, Media List Activity Wireframe when iOS Apps is selected.

Part C: DetailedMediaActivity (30 Points)

Detailed Media Activity shows details about a selected media item. The user can view more details about the media and then return back to the list. The requirements are as follows:

1. The activity's wireframe is shown in Figure 4. The media details will depend on the media type selected. This figure shows the detailed needed for iOS Apps. Details about different categories provided in Table 1. Pressing the back key should return to the Media List Activity.
2. The image should be retrieved using a background thread (Thread or AsyncTask).
3. As mentioned Earlier, Different Categories have different Details. The following table shows what is need to be parsed for different Media Types.
4. Layout of the Activity should be managed at run time as the UI elements will be formed according to media type selected.
5. All Media items have link to preview the media details in iTunes website. Tapping the link on DetailedMediaActivity should start **PreviewActivity** which should display the media link using a WebView component, the WebView should support URL redirects



App Details

App Title

App Release Date

By: ABC

Price: $XX

Category: XYZ

App Link in Store:

https://itunes.apple.com/us/app/XYZ

Figure 4, Detailed Activity

and should enable the user to view and interact with the website. The DetailedMediaActivity should send the URL to the **PreviewActivity**.

Media Type	MediaListActivity	DetailedMediaActivity
Audio Books	Thumbnail image, Title	Title, Large Image, Artist, Duration, Artist, Category, release Date and link to preview in App store
Books	Thumbnail image, Title	Title, Large Image, Artist, Summary, Category, Price, release date and link to preview in App store
iOS Apps	Thumbnail image, Title	Title, Large Image, Artist, Price, Category, release date and link in to preview in App store
iTunes U	Thumbnail image, Title	Title, Large Image, Artist, Summary, Category, Price and link to preview in App store
Mac Apps	Thumbnail image, Title	Title, Large Image, Artist, Summary, Category, Price, release date and link to preview in App store
Movies	Thumbnail image, Title	Title, Large Image, Artist, Price, Category and link in to preview in App store
Podcast	Thumbnail image, Title	Title, Large Image, Artist, Price, Summary and link in to preview in App store
TV Shows	Thumbnail image, Title	Title, Large Image, Artist, Price, Summary, Category, release Date and link in to preview in App store

Table 1: Details for different Media Types