

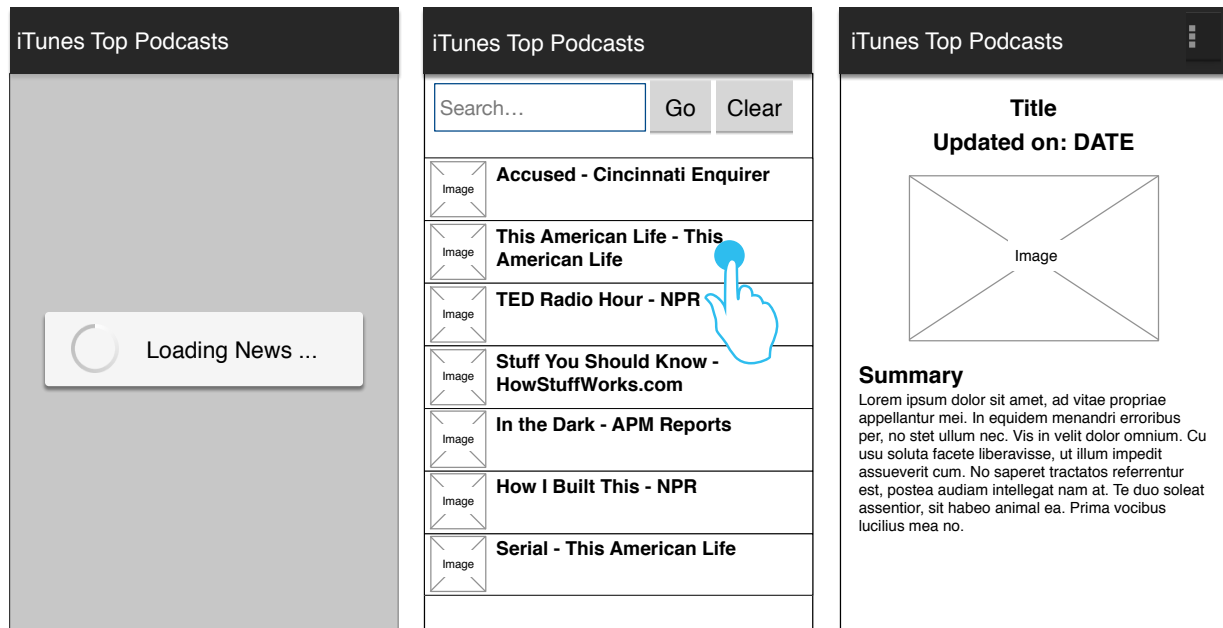
ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 7

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a) Assignment #.
 - b) Full name of the student.
2. This is an individual assignment. You will be responsible for doing this assignment by yourself.
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. Submission details:
 - a) Export your Android project and create a zip file which includes all the project folder and any required libraries.
 - b) The file name is very important and should follow the following format: **InClass07_YourLastName.zip**. You should submit the assignment through Canvas: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 07 (100 Points)

In this assignment you will build an app that shows you the top iTunes podcasts. In this assignment you will continue to practice parsing RSS (XML or JSON) feeds, using Threads/AsyncTasks and dynamic UI management.



(a) Loading Progress Dialog

(b) Clickable Item List

(c) Item Details



(d) Searching for an item

(e) Found the Item and showed in a colored box

(f) Clear the color and show the list as (b)

Figure 1, App Wireframe

Part A: Main Activity (70 Points)

The launching activity is the Main Activity. Figures 1(a) and 1(b), show the UI for the Main Activity. The activity displays a list of the items provided by iTunes Top Podcasts. The requirements are as follows:

1. Use a thread pool (or AsyncTask) to communicate with the iTunes API and to parse the result. Do not use the main threads to download the results. You are free to use XML or JSON based on the following apis:
 - a) The XML API is: <https://itunes.apple.com/us/rss/toppodcasts/limit=30/xml>
 - b) The JSON API is: <https://itunes.apple.com/us/rss/toppodcasts/limit=30/json>
 - c) The limit should be always set to 30.
2. Parse the XML or JSON and return the result to the Item List.
3. The list should be created using **ListView**, each row should be formatted as shown in Figure 1(b).
 - a) Create a custom adapter, that should be implemented in a separate file and should not be included in the activity as an inner class.
 - b) Your code should makes use of the row item pool provided by the adapter's getView method.
 - c) The story title is the new item "title" element and the image is the smallest square image in the "im:image".
 - d) The image should be loaded using a thread or AsyncTask. Feel free to use the Android Picasso Library, <http://square.github.io/picasso/>
4. For each item retrieve the title, summary, release date, the smallest image url as thumbnail, and largest image url.
5. The Items list should be sorted in an Ascending Order of the release date value. You can use the **SimpleDateFormat** in Android to format the retrieved dates. <http://developer.android.com/reference/java/text/SimpleDateFormat.html>.
6. Clicking a row item should start the Item Details Activity, which provides more details about the current news item.
7. When the user types a keyword and clicks the Go button, the app should find the items that match the keyword, should move the matches rows to the top of the list and should highlight the matched rows by changing their background to green as shown in Figure 1(d) and 1(e).
8. Clicking on "Clear" button should clear the search box, clear any perviously colored rows to a clear background, and resort the list to display items as indicated in (5), as shown in Figure 1(b).

Part B: Item Details Activity (30 Points)

Detailed Activity shows details about a selected item. The user can view more details about the news such as update date, summary etc.

The requirements are as follows:

1. The activity's wireframe is shown in Figure 1(c). The item details should include the title, last update date, large image thumbnail, and Summary. Clicking the back key should return to the Main Activity.
2. The update date should be formatted as follows: MM/DD/YYYY HH:MM AM/ PM
3. The image should be retrieved using a background thread (Thread or AsyncTask).
4. You can use Picasso to retrieve the image. Check <http://square.github.io/picasso/>