

ITIS/ITCS 4180/5180 Mobile Application Development  
Homework 07

---

**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#\_HW07.zip
  - c. You should submit the assignment through Canvas: Submit the zip file.
7. **The minimum SDK version should be set to 14 and the maximum SDK version should be set to 23.**
8. **Failure to follow the above instructions will result in point deductions.**

## Homework 07 (100 Points)

In this assignment you will build a podcast app. Your app should display and play episodes from “**TED Radio Hour**” channel. You can get access to this channel RSS feed from “NPR” website on this link:

<http://www.npr.org/rss/podcast.php?id=510298>

Your app should enable displaying the podcast items in two modes, in a linear list and in grid view. In this assignment, you will learn how to use RecyclerView, Action Bar and Dynamic Layout Management. In addition to Http connection and XML parsing.

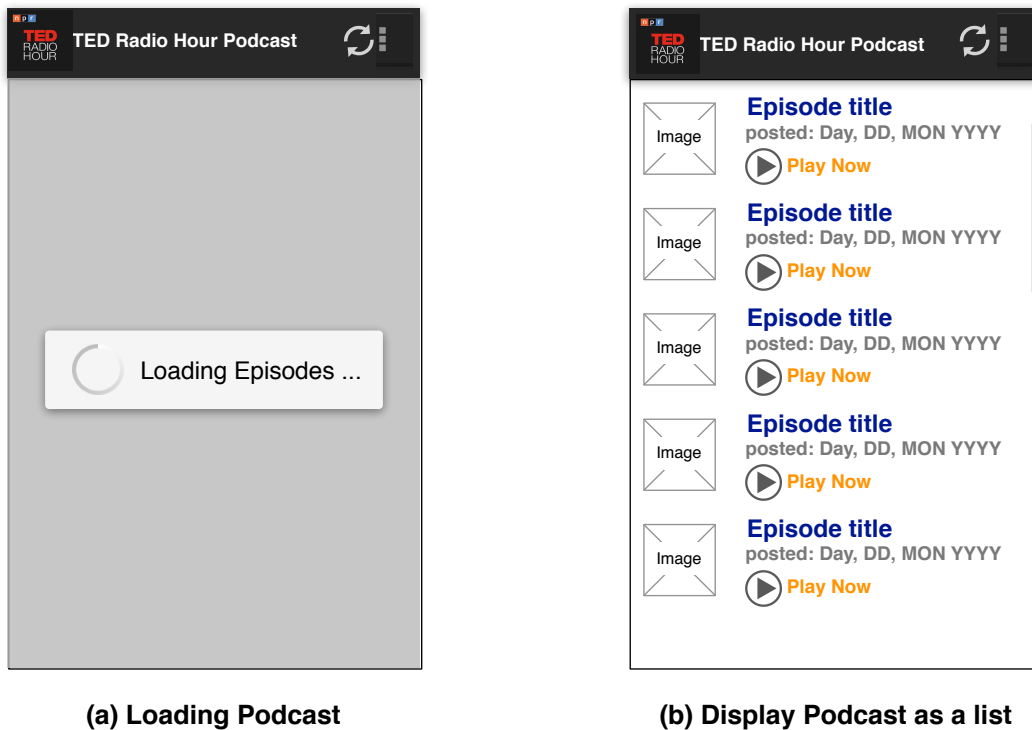


Figure 1: Podcast as a List

### Part A (85 Points): MainActivity

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. The **MainActivity** is responsible for the loading the podcast rss feed from provided URL. The implementation requirements include:

1. Use a thread pool (or AsyncTask) to communicate with the podcast and to parse the generated result. Do not use the main threads to download the rss feed results. A progress dialog should be displayed during the retrieval and parsing of the XML data.
2. Sample XML item representing an episode looks like Figure 2. The fields needed for this assignment are the Title, Description, Publication date, Image URL, Duration, and MP3 file URL.
3. You are required to use **RecyclerView**, to get the list to display as shown in Figure1(b). This widget is a container for displaying large data sets that can be

scrolled very efficiently by maintaining a limited number of views. To know more about this widget see:

- <https://developer.android.com/training/material/lists-cards.html>
- <https://developer.android.com/reference/android/support/v7/widget/RecyclerView.html>

```

<item>
  <title>Transformation</title>
  <description>
    <![CDATA[
      Are we simply the sum of our experiences? Or can we
      choose our own path? In this hour, TED speakers share
      stories of undergoing remarkable transformations despite
      extraordinary challenges. (Original broadcast date:
      September 19, 2014)
    ]]>
  </description>
  <pubDate>Fri, 24 Jul 2015 00:23:00 -0400</pubDate>
  <copyright>...</copyright>
  <guid>9c22f2a2-39c2-4a4f-9be3-47a4d7186b5e</guid>
  <itunes:author>NPR</itunes:author>
  <itunes:summary>...</itunes:summary>
  <itunes:subtitle>...</itunes:subtitle>
  <itunes:image
    href="http://media.npr.org/assets/img/2015/07/22/trh_transfo
    a9c6cd48c914e94e14f82054ebec842e994e8e46.jpg?s=1400"/>
  <itunes:duration>3227</itunes:duration>
  <itunes:explicit>no</itunes:explicit>
  <enclosure url="http://play.podtrac.com/npr-
  510298/npr.mc.tritondigital.com/NPR_510298/media/anon.npr-
  mp3/npr/ted/2015/07/20150724_ted_transformation.mp3?
  orgId=1&d=3227&p=510298&story=425242267&t=podcast&e=42524226
  length="0" type="audio/mpeg"/>
</item>

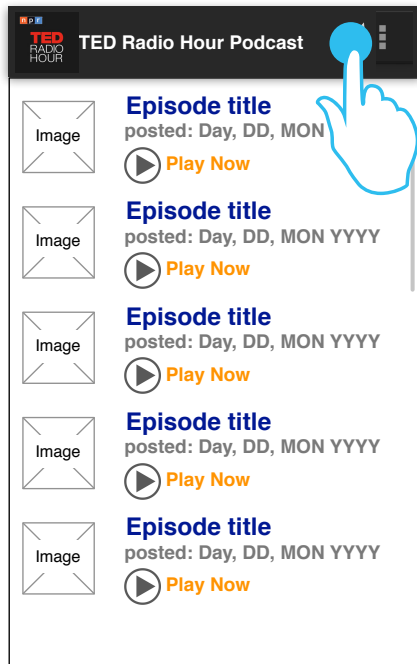
```

Figure 2: Sample XML episode

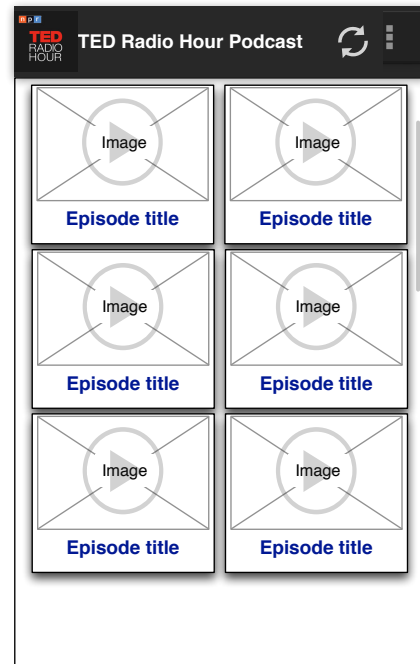
- Each list item should display retrieved episode titles, release date, thumbnail image and play button for the media URL. (see Figure3(a)).
- List items should be ordered in descending order by release date.
- The activity should have an **Action Bar**, which is a window feature that identifies the user location, and provides user actions and navigation modes. To learn more about Action bars and how to implement them : <http://developer.android.com/guide/topics/ui/actionbar.html>
- Action bar should contain one action item ( switch view) which is dictated by the icon on the top right corner. Tapping this icon should switch between the vertical linear list to a grid list displaying the same information but displayed differently, see Figure 3. **You are required to use only RecyclerView(s) for both display modes. Given that it supports both options.**
- Each item in the grid should display retrieved episode image, a semi-transparent play icon over the image, and episode title (see Figure3(b)). Items in grid should also be ordered by release date in descending order.
- Play button (in both views) should online stream the mp3 file retrieved, i.e, mp3 file should not be saved/ downloaded. Once the play icon is tapped, player controls UI elements should appear (play/pause button with a progress bar), they should

disappear again once the media play is finished. See Figure 4.

10. Tapping on the list or grid item should start the **PlayActivity** and to pass it the required information to display detailed information about the episode.

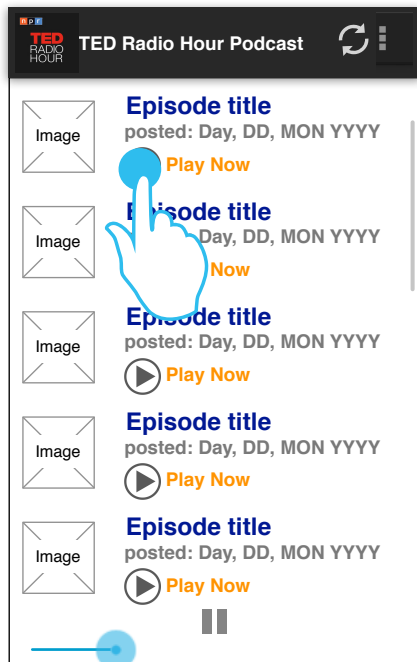


(a) Vertical List Format

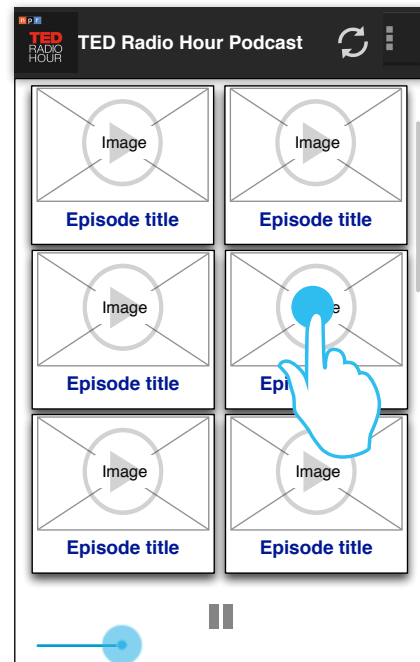


(a) Grid Format

Figure 3: Switch view modes



(a) Vertical Recycler View

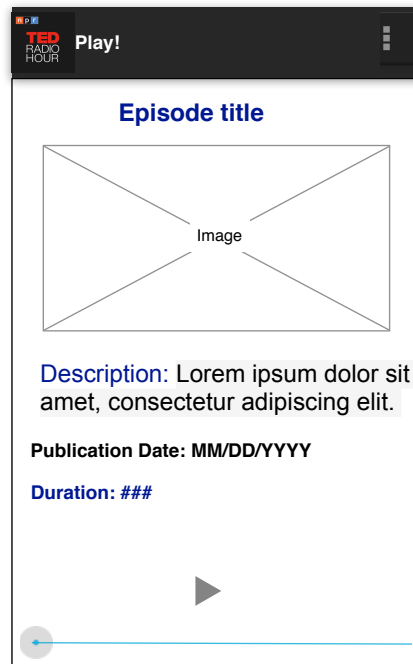


(b) Grid View

Figure 4: Play Button

### **Part B (15 Points): PlayActivity**

This activity should display the episode title, image, description, duration, and publication date. It should also enable the user to stream the media file retrieved for the selected episode. *You should not download the file, rather your app should stream online media file.* Tapping on the device's back button should finish this activity (See Figure 5).



**Figure 5: Play Activity**