

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. 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#_InClass07.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.**

In Class Assignment 7 (100 Points)

This assignment is a continuation of the last InClass assignment. You will make HTTP requests and parse JSON data. Using the iTunes API you will retrieve list of applications in JSON format and display them on your own app using customized ListView. Additionally, you will learn to add, retrieve and delete items from an SQLite database. This application is composed of 2 activities, namely **MainActivity**, and **PreviewActivity**.

API Description

- The API used in this assignment is the iTunes JSON RSS API. It should retrieve 25 top grossing applications provided by the iTunes store. Use the following api:

<http://itunes.apple.com/us/rss/topgrossingapplications/limit=25/json>

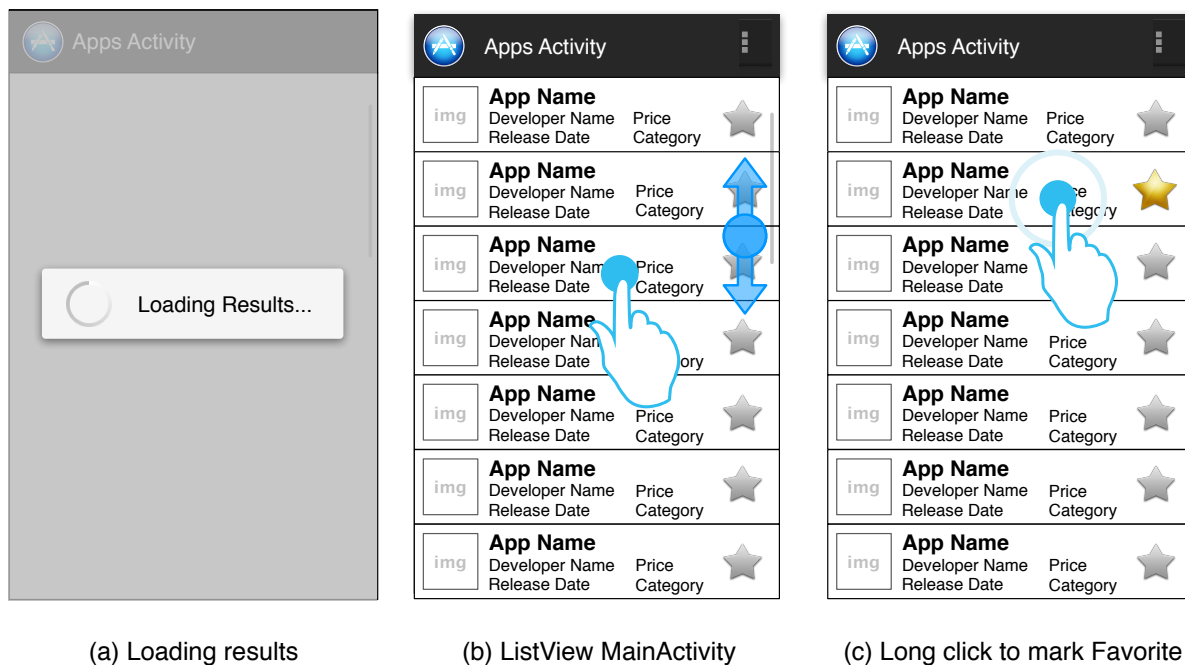


Figure 1, Application User Interface

Part 2 (90 Points): Main Activity

- The progress dialog shown in Figure 1(a) should be dismissed after the parsing is completed, and the app items should be displayed in a list. This list should be created using a customized ListView. Each item in the ListView should contain a number of TextViews showing the related app information, and two ImageViews showing the photo thumbnail and Favorites status icon, all shown in Figure 1(b).
- Create a AppList Database which contains a table with six columns: appName, developerName, releaseDate, price, category, and imageURL.
- The app item row should be clickable to provide the following functionality:
 - Clicking an app item should start the PreviewActivity which provides more details about the current app item.

- 2.2. A LongClick on an app item
 - (a) If the app item is not in the AppList SQLite table, then add the app item to the AppList table and turn the Favorites status on by showing the yellow star icon.
 - (b) If the app item is already in the AppList SQLite table, then delete the app item from the AppList table and turn the Favorites status off by showing the grey star icon.
 - (c) Display a toast message indicating the status of the performed operation, successful addition or deletion from the database or error.
3. The ActionBar menu should display two options, "Show Favorites" and "Show All", as shown in Figure 2(a). When the "Show Favorites" is selected the list should only show the apps that are in the favorites table. When the "Show All" is selected the list should display all the apps retrieved from the web service.

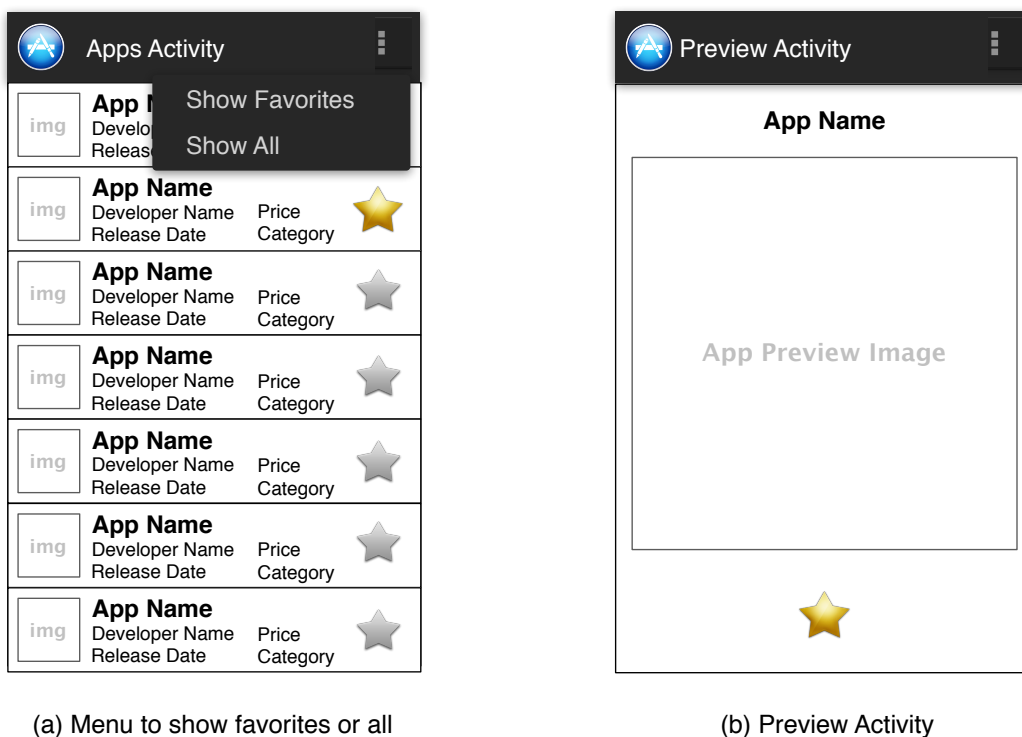


Figure 2, list and preview activity

Part 3 (10 Points): PreviewActivity

This activity should display the application title, and the application preview image. The PreviewActivity should receive the app information from the MainActivity. Figure 2(b) shows the PreviewActivity. The implementation requirements, as before, include:

1. Use a thread pool (or AsyncTask) to retrieve the application preview image. Do not use the main threads to download the selected photo.
2. Note, that the favorite image should be updated to indicate if the previewed app is in the favorites or no.
3. Pressing back should end this activity and go back to the MainActivity.