

ITIS/ITCS 4180/5180 Mobile Application Development  
In Class Assignment 8

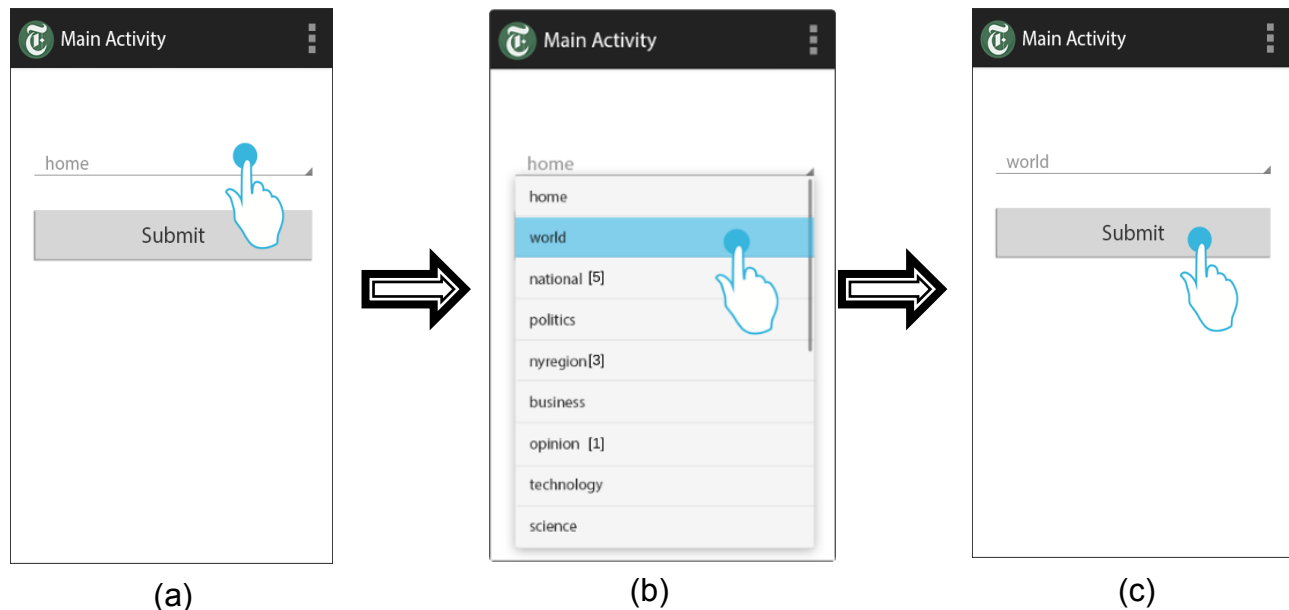
**Basic Instructions:**

---

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 Moodle for each group.
  - b. The file name is very important and should follow the following format:  
**Group#\_InClass08.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 8 (100 Points)

This assignment is a continuation of last in class Assignment. In this assignment you will focus on using SQLite operations to store bookmarked news articles. Using the NY Times Top Stories API you will retrieve the Top News in JSON format and display them in your own app. The app is composed of three activities, namely **MainActivity**, **TopStoriesActivity** and **DetailsActivity**.



**Figure 1. The Main Activity Wireframe**

### **Part 1 (10 Points): MainActivity**

The Main activity displays a Spinner that enables the user to select a section. The list of sections is provided at [http://developer.nytimes.com/docs/top\\_stories\\_api/](http://developer.nytimes.com/docs/top_stories_api/). Figure 1 shows the Main Activity. The requirements include:

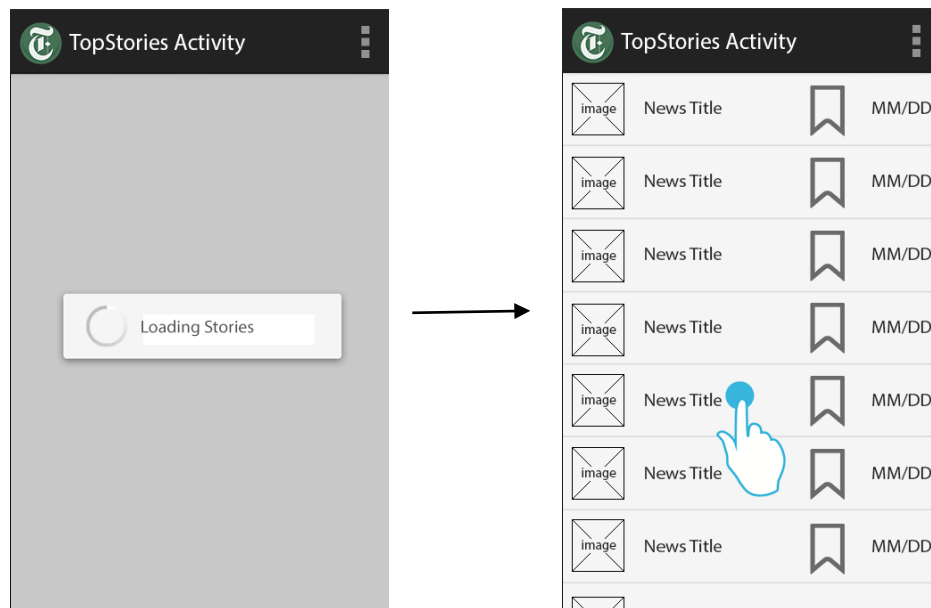
1. Spinner should have a Dropdown with a list of sections. The sections are:  
**home, world, national, politics, nyregion, business, opinion, technology, science, health, sports, arts, fashion, dining, travel, magazine, realestate**
2. A count of number of articles that are book marked and stored in the database of each category should be included with the name of the category as shown in Figure 1b. If a category doesn't have bookmarks then the count should not be shown.
3. Upon clicking the submit button start the TopStories Activity. The intent should include the section name using extras. You will not be given any points if you use a static variable to share data between activities.

### **Part 2 (80 Points): TopStories Activity**

The TopStories activity is responsible for retrieving the Top Stories results based on the section term sent from the Main Activity. All the story items are under Results array.

Each array item includes the story title, story abstract, story byline and many other fields. In addition, each story item includes a thumbnail.

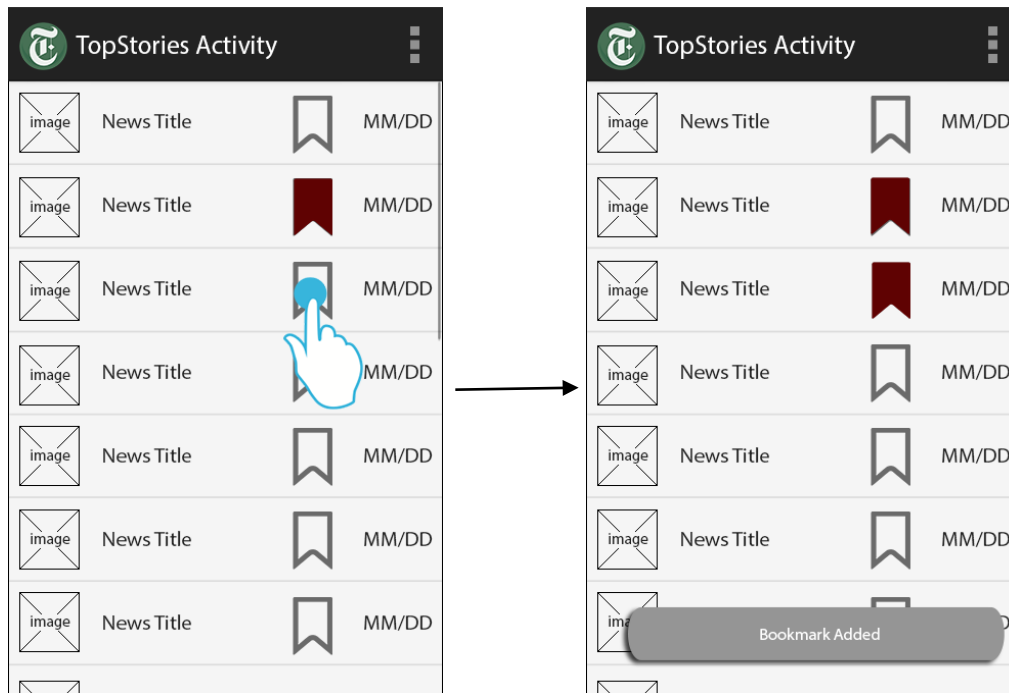
The NY Times Top Stories API should be used to retrieve the results matching the section, referring to the documentation provided at the following location: [http://developer.nytimes.com/docs/top\\_stories\\_api/](http://developer.nytimes.com/docs/top_stories_api/)



**Figure 2. The TopStories Activity**

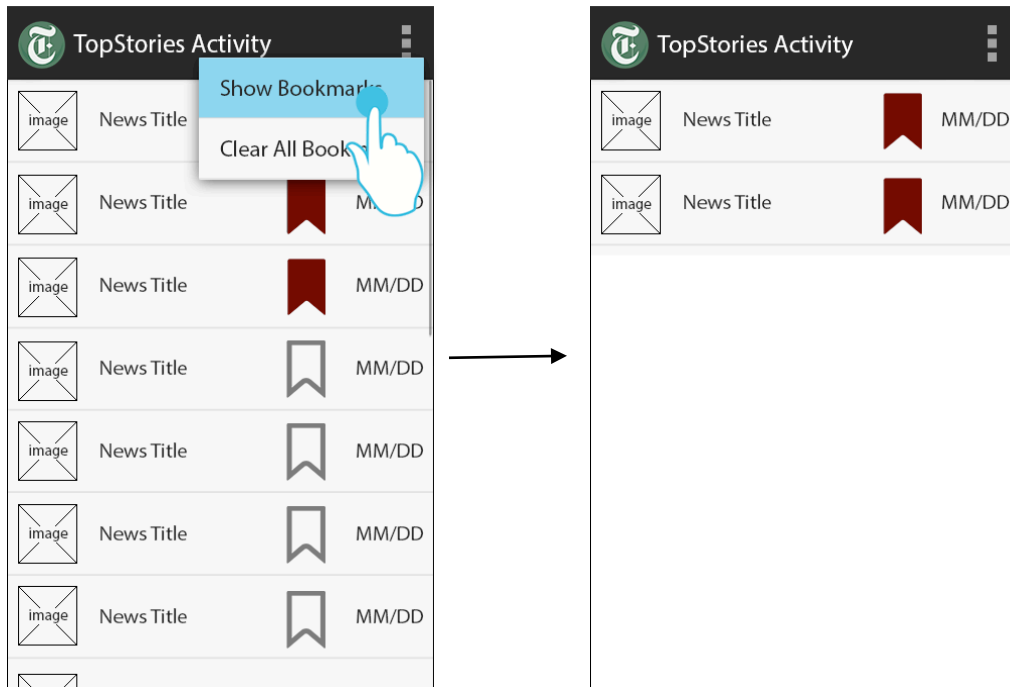
The TopStories Activity wireframe is shown in Figure 2. The requirements include:

1. 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 JSON is being downloaded and parsed, display a ProgressDialog as indicated in Figure 2a. The JSON parser should return a list of story objects.
2. Create a Stories Database which contains a table with six columns: **Story title**, **Story byline**, **Story abstract**, **Story created\_date**, **Story thumb\_image\_url** and **Story normal\_image\_url**.
3. The ProgressDialog should be dismissed after the parsing is completed, and the story items should be displayed in a list. **This list should be created using a customized ListView**. Each item in the ListView should contain a TextView showing the *Story title*, ImageView showing the *Story thumbnail* and *Story created\_date* in "MM/DD" format as shown in Figure 2b.
4. Already stored story objects in the stories database table should show a filled bookmark icon. If the story is not stored in the database then show an empty icon.

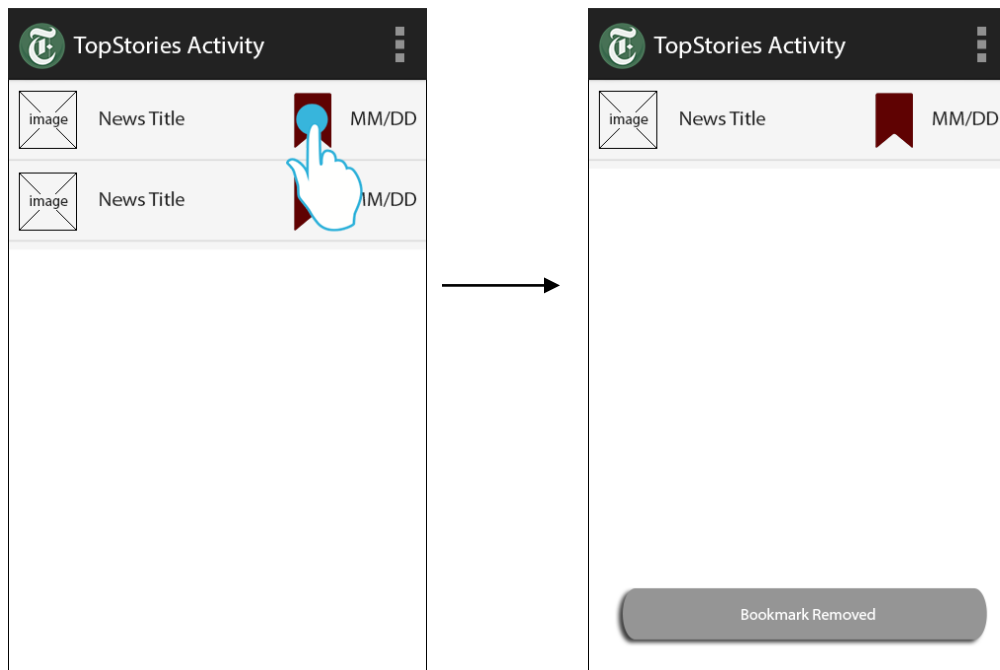


**Figure 3. Showing bookmarks**

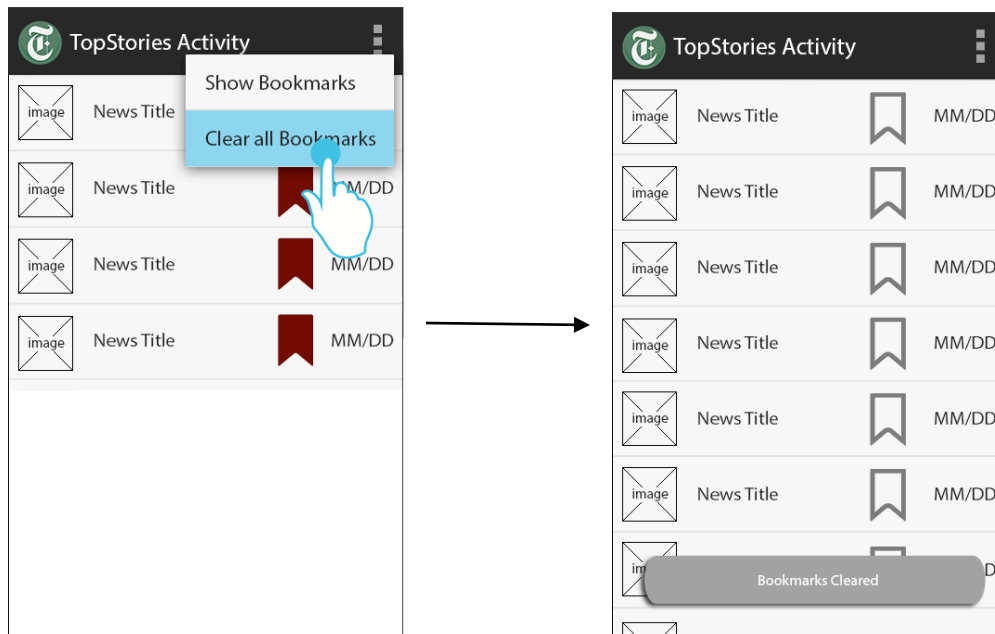
5. Clicking a list item should start the Details Activity. The intent should include the required story information to be sent to the Details activity using extras. You will not be given any points if you use a static variable to share data between activities.
6. A LongClick on the list item should have the following behavior:
  - a) If the Story item is not in the Stories SQLite table, then add the Story item to the Stories table and turn the Bookmark flag ON by showing the Red Color bookmark icon.
  - b) If the Story item is already in the Stories SQLite table, then delete the Story item from the Stories table and turn the Bookmark flag OFF by showing the grey bookmark icon.
  - c) Display a toast message indicating the status of the performed operation, successful addition or deletion from the database or error.
7. The ActionBar menu should display two options, "Show Bookmarks" and "Clear All Bookmarks," as shown in Figure 4.
  - a) When the "Show Bookmarks" is selected the list should only show the Stories that are in the Stories table with Red book mark icon, as shown in Figure 4a.
  - b) User should be able to clear a specific bookmark from this list as well, as shown in the Figure 5.
  - c) When the "Clear All Bookmarks" is selected the list should delete all the Stories in the database and all the list items should show grey bookmark icon, as shown in the Figure 6.



**Figure 4. Showing bookmarks**



**Figure 5. Clearing a bookmark**

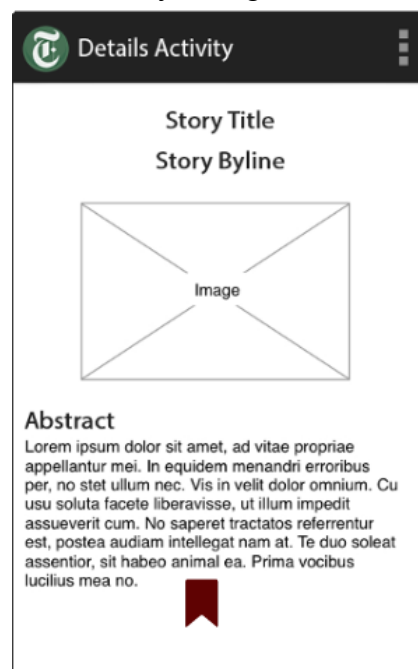


**Figure 6. Clearing all bookmarks**

### **Part 3 (10 Points): DetailsActivity**

This activity should display the story details, which includes the Story Title, Story byline, Story abstract and story normal image, which are received from the TopStories Activity.

1. Do not use the main threads to download the selected photo. You may use Picasso or an AsyncTask. If there is no photo image, use a placeholder image.
2. The “Bookmark” icon should be updated to reflect if the story is bookmarked or not.
3. Pressing back should end this activity and go back to the TopStories Activity.



**Figure 7. The Detail Activity Wireframe**