

ITIS/ITCS 5180 Mobile Application Development  
In Class Assignment 5

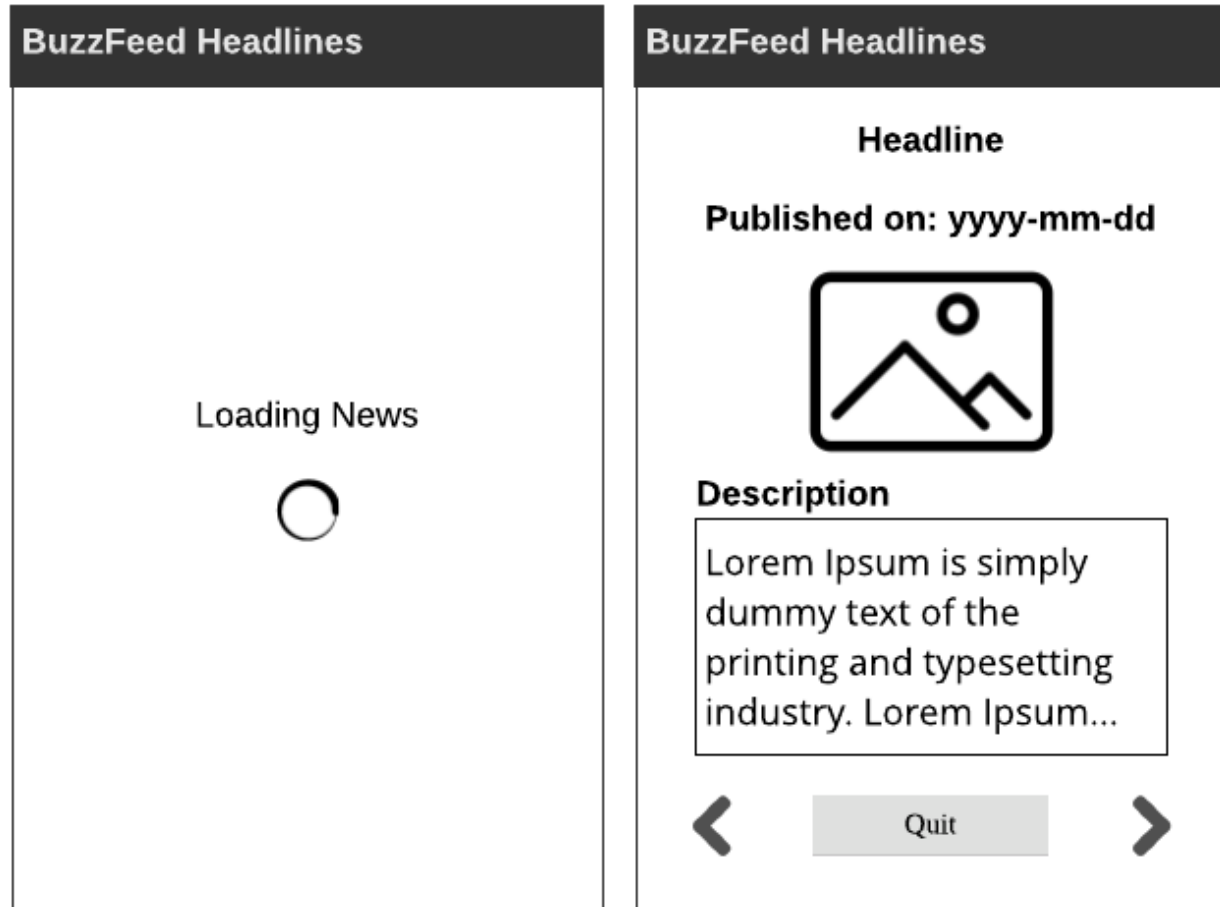
---

**Basic Instructions:**

1. In every file submitted you **MUST** place the following comments:
  - a) Assignment #.
  - b) Full name the student.
2. 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.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Submission details: It is very important to follow the following format: **800#\_InClass05.zip**. You should submit the assignment through Canvas: Submit the zip file.
5. **Failure to follow the above instructions will result in point deductions.**

### In Class Assignment 05 (100 Points)

In this assignment you will develop the “Buzzfeed Top Headlines” app. The App displays news feeds from News API’s Buzzfeed source. Here you will learn how to parse JSON news feeds using Threads/AsyncTasks and dynamic UI management.



(a) Loading ProgressBar

(b) News Headlines

**Figure 1, App Wireframe**

### Main Activity (100 Points)

Please follow the instructions to develop the application:

1. Please visit News Api webpage, register, and get an API Key. (<https://newsapi.org/register>)
2. Login and visit Buzzfeed API documentation page. (<https://newsapi.org/s/buzzfeed-api>)
3. Please go through the example given there.
  1. The base API link will be: `https://newsapi.org/v2/top-headlines?sources=buzzfeed&apiKey=<YOUR_KEY>`
  2. There will be 10 news headlines.
4. You need to parse the **articles** array.

1. In each item of the array, you will see several attributes, such as author, title, description, url, urlToImage, publishedAt, and content.
2. You need to parse the **title**, **publishedAt**, **urlToImage**, and **description**.
5. Create an ArrayList of News objects.
  1. Each News object should contain **title**, **publishedAt**, **urlToImage**, and **description**.
6. Parse all the data when the app starts.
  1. Display a ProgressBar while you parse the data and save it in the ArrayList.  
HINT: You may use Alert Dialog along with ProgressBar. Or, you can play with visibility for ProgressBar and TextView.
  2. Keep all ten news items in ArrayList.
7. After the items are parsed and saved, display the items as it is shown in Figure 1(b).
  1. TextView 1: put the title.
  2. TextView 2: put the publishedAt, e.g, 2018-10-05.
  3. ImageView: load urlToImage using Picasso.
  4. Large TextView: put the description. **HINT:** you can use a ScrollView having a fixed height, and put the TextView inside it with "wrap\_content" for both height and width attributes.
  5. ImageButtons: two ImageButtons should be there the browse through the headlines.
    1. Previous button should be used to go to the previous news.
    2. Next button should be used to browse the next news.
    3. If it's the first item in the list, and you press the Previous Button, make a toast to let you know the you are on the first item.
    4. Please follow the similar rules for the Next button.
8. There should be a Button to Quit the app.

NOTE: In order to easily view the JSON feed, you can copy and paste the URL/JSON to <https://codebeautify.org/jsonviewer> and click on "Beautify".

Picasso: <http://square.github.io/picasso/>