ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 5

**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#_InClass05.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 05 (100 Points)

In this assignment we are going to make a Photo Gallery application, which will use a URL that retrieves a text file containing a dictionary of keywords and image URLs related to the associated keyword. The application consists of a single activity that enables the user to download and view online photos. The photos related to the keyword for this app should be retrieved from the below URL:

http://dev.theappsdr.com/apis/spring_2016/inclass5/URLs.txt

The retrieved text contains several lines of pair (keyword, URL). Here, "keyword" is the search keyword. For example, "UNCC" have five URLs associated with it. Other keywords may have more or less. Each keyword and Photo URL is formatted as a comma-separated pair (","), and each pair is separated by a semicolon (";"), as shown in the below example:

```
UNCC,https://drscdn.500px.org/photo/138881537/m%3D900/40289
9b651a8068ad5cddd48ca498a25;UNCC,https://drscdn.500px.org/p
hoto/107798503/m%3D900/99cc149d33f0479493aba8e4292ca8a4;UNC
C,https://drscdn.500px.org/photo/28044681/m%3D900/82e874490
a2f47ec8ad4b28e83f107e9;UNCC,https://drscdn.500px.org/photo
/57936218/m%3D900/e4d3105fff3ce410a87298a9cbd96138;UNCC,htt
ps://drscdn.500px.org/photo/119078659/m%3D900/58a409689aed0
05aa2d5b9e341aa6e5e;Android…
```

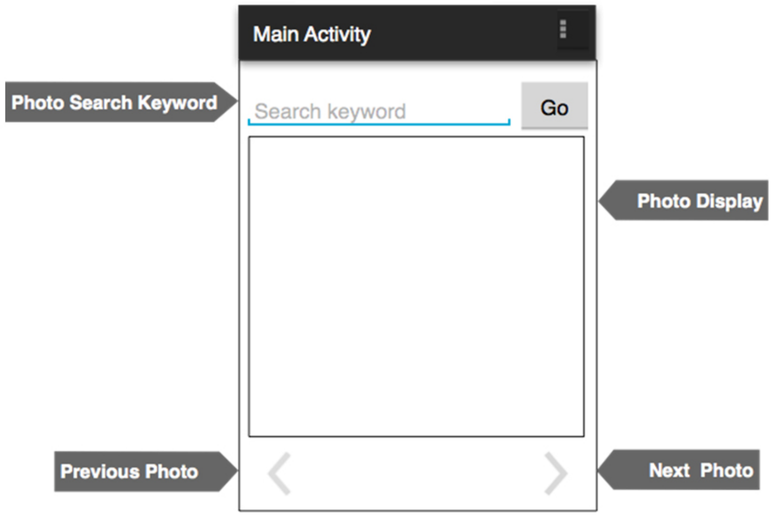

**Figure 1, Application Wireframe**

## Loading Images based on Keywords

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. Perform the following tasks:

1. Create a new android project called “In Class 05”.
2. Use AsyncTask or Thread to connect to the provided URL and retrieve the content of the dictionary. Use any data structure that may help you later retrieve keyword and URL efficiently.
3. While the dictionary is being retrieved display a progress dialogue indicating Retrieving dictionary as shown in Figure 2(a).
4. When the dictionary is retrieved and saved, dismiss progress dialogue so that user can interact with the activity.
5. The EditText will hold the search keyword to be used to search for photos.
6. The keywords are: ***UNCC, Android, winter, aurora borealis,*** *and* ***wonders.***
7. When the “Go” button is pressed, you should do the following:
    a. Use another AsyncTask/Thread to retrieve the first image associated with the keyword and display it in the ImageView.
    b. The AsyncTask/Thread class should be in a separate file/class not inner to the main thread. i.e. You should manage passing parameters to the class and then pass back the result image downloaded to the UI.
    c. While the image is being retrieved, you should display a Progress Dialog as indicated in Figure 3(b).
8. The Next and Previous photo icons should be disabled when the application is launched, and enabled after the first photo is displayed. The buttons will also remain disabled in the case there is only 1 image or there are no images corresponding to a keyword/ not existing keyword. Use icons provided in SupportFiles for setting the image icons (**`next.png, prev.png`**)
9. Do not store the photos, simply download and display the retrieved photos.
10. Upon clicking the “Next Photo” icon, you should download the next photo as stored in your data structure object (following the of order of appearance on the text file). You should call the AsyncTask/ Thread used to download the next photo.
11. If the currently displayed photo happens to be the last photo, you should download and retrieve the photo at index 0 (the first photo) when the Next icon is pressed.
12. Clicking the “Previous Photo” icon, you should download the previous photo. If the currently displayed photo happens to be the first photo, you should download and retrieve the photo at the last index position (last photo).
13. If the user enters some keyword other than those mentioned above, an error Toast message should be displayed saying “No Images Found” and the ImageView should be made blank.
14. Your application should download the requested photo only if there is an established internet connection. If there is no internet connection you should display a Toast message indicting that there is no internet connection and do not attempt to send the HTTP request.

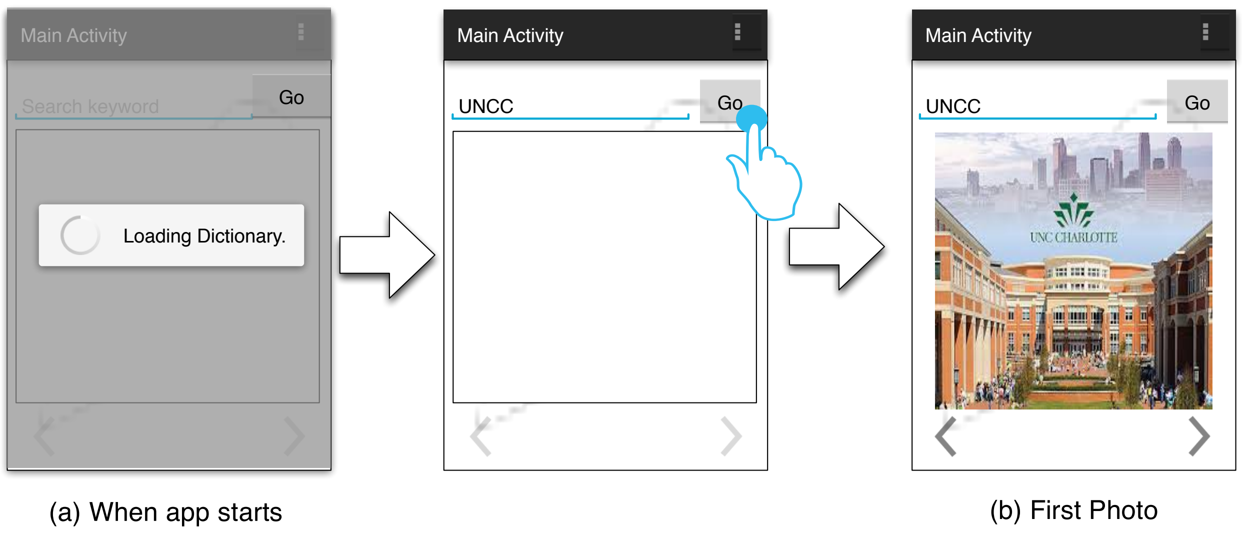

**Figure 2,Loading Dictionary & Displaying first photo**

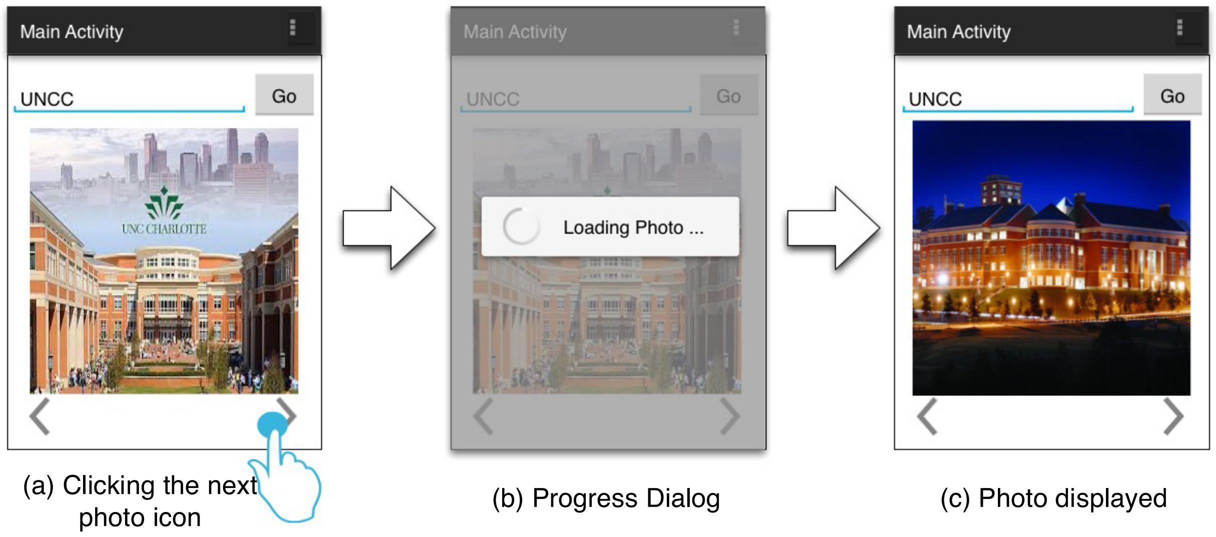

**Figure 3, Displaying next photo**