ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 4

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#_InClass04.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.**

<u>In Class Assignment 4 (100 Points)</u>

The application consists of a single activity that enables the user to download and view online photos. The interface should be created to match the user interface (UI) presented in Figure 1. You are required to perform the following tasks:

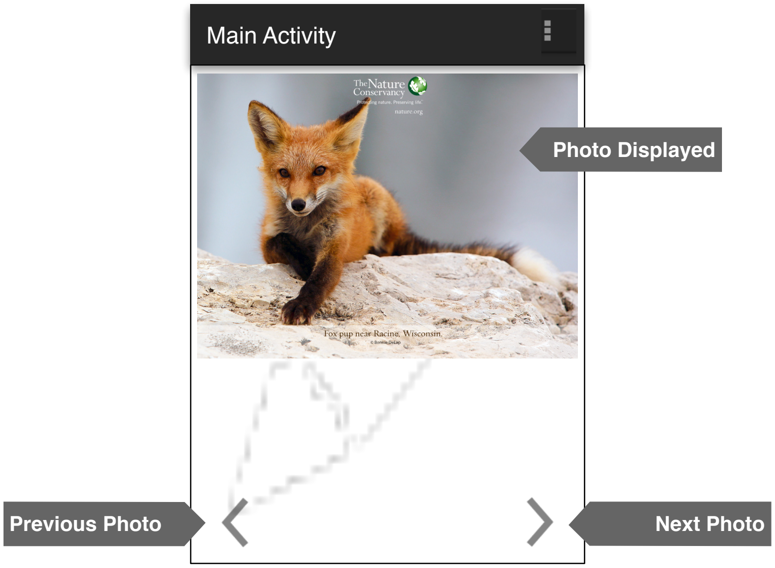

Figure 1, Application Wireframe

1. We created two web services which can be accessed using GET method. The Get photo list API returns a list of Photos IDs. The Get Photo API expects a GET parameter "pid", which is the photo id.
   a. Get Photo List : http://dev.theappsdr.com/lectures/inclass_photos/index.php
   b. Get Photo : http://dev.theappsdr.com/lectures/inclass_photos/index.php?pid=XXX
2. When the app first loads you should retrieve the list of Photos IDs by calling the Get Photo List API. Having the list of Photos IDs, you should retrieve and display the first photo. You need to call the Get Photo and pass the first photo ID (pid) from the retrieved photo list.
3. You should use a child thread or AsyncTask to perform the loading of the image IDs and images. Do not store the images loaded, simply download and display the retrieved images. All UI operations should be performed by the Main Thread.
4. Upon clicking the "Next Photo" icon, you should download the next photo. If the currently displayed photo is the last photo, you should download and retrieve the first photo again.
5. Upon clicking the "Previous Photo" icon, you should download the previous photo. If the currently displayed photo is the first photo, you should download and retrieve the last photo.
6. While the photo is being downloaded you should display a Progress Dialog as indicated in Figure 2 The Progress Dialog should be dismissed when the worker thread or AsyncTask is done retrieving the photo.
7. Your application should download the requested image only if there is an established internet connection. If there is no internet connection you should display an Toast message indicting that there is no internet connection and do not attempt to send the

HTTP request.

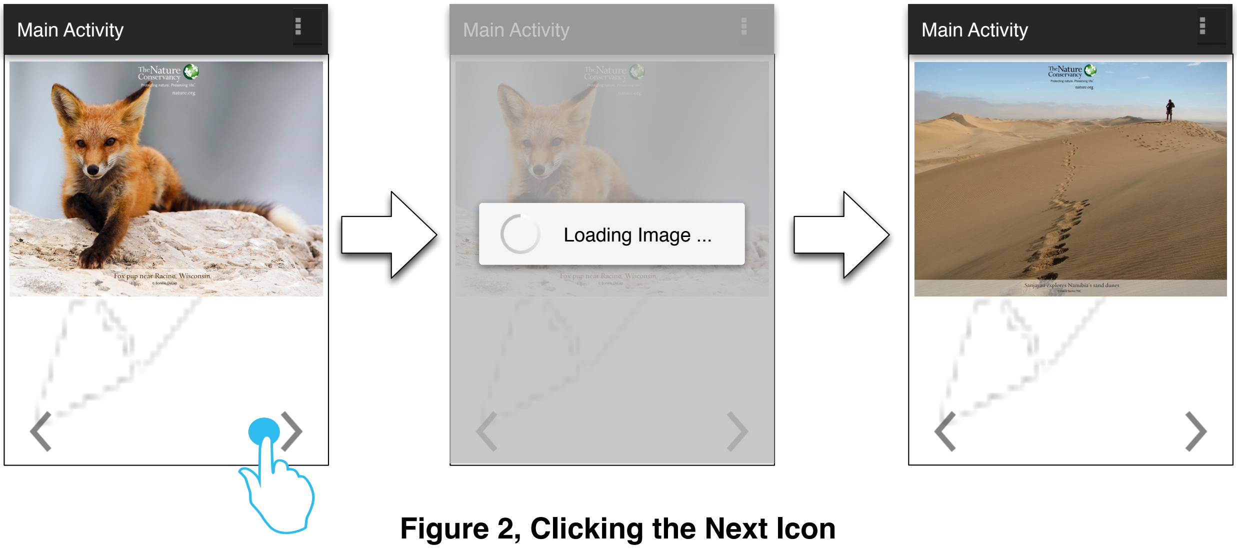

Figure 2, Clicking the Next Icon