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 student in the group is required to submit the assignment on moodle.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. **Export your project as follows:**
   a. From eclipse, choose *"Export…"* from the File menu.
   b. From the Export window, choose *General* then *File System*. Click *Next*.
   c. Make sure that your project for this assignment is selected. Make sure that all of its subfolders are also selected.
   d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
   e. When exporting make sure you select *Create directory structure for files*.
   f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
   a. All the group members should submit the same zip file.
   b. The file name is very important and should follow the following format: **Group#_InClass05.zip** For example, Group2A_InClass05.zip
   c. You should submit the assignment through Moodle: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 5 (100 Points)

In this assignment you will get familiar with HTTP connections and XML parsing. You will develop a photo viewer application that downloads and displays online photos.

Important App Requirements:

1. Create a new android project called “In Class 5”.
2. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 19**. The app should display correctly on Nexus 5. Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
3. You will be using layout files, and strings.xml to create the required user interfaces. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse.

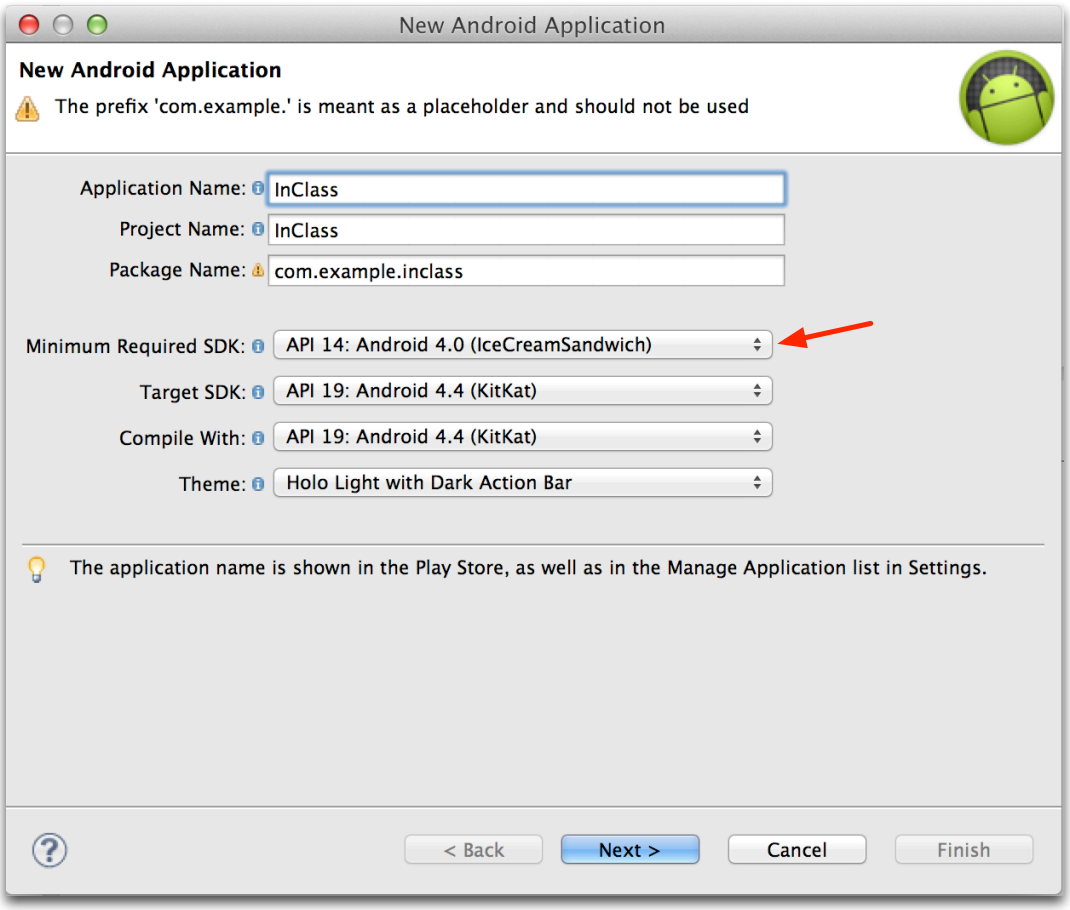


Fig 1. Choosing Minimum Required SDK to 14

Initial Setup and API Description

First you should create an account on Flickr, and then select create an new app at https://www.flickr.com/services/apps/create/. In this assignment we are going to use the Flicker Photos Search api for retrieving the photos related to a search keyword. For information related to the api please check https://www.flickr.com/services/api/flickr.photos.search.html. The API details is as follows:

- Endpoint: https://api.flickr.com/services/rest/?
- Arguments (GET Method)
  - method: should be set to flickr.photos.search
  - api_key= should be set to your API key.
  - text: this is the search keyword for example, UNCC.
  - extras: should be set to “url_m” to retrieve the photo URL.
  - per_page: should be set to “20” to retrieve the first 20 photos.
  - format: should be set to “rest” in order to retrieve XML.

Parsing XML Document using XML SAX and Pull Parser

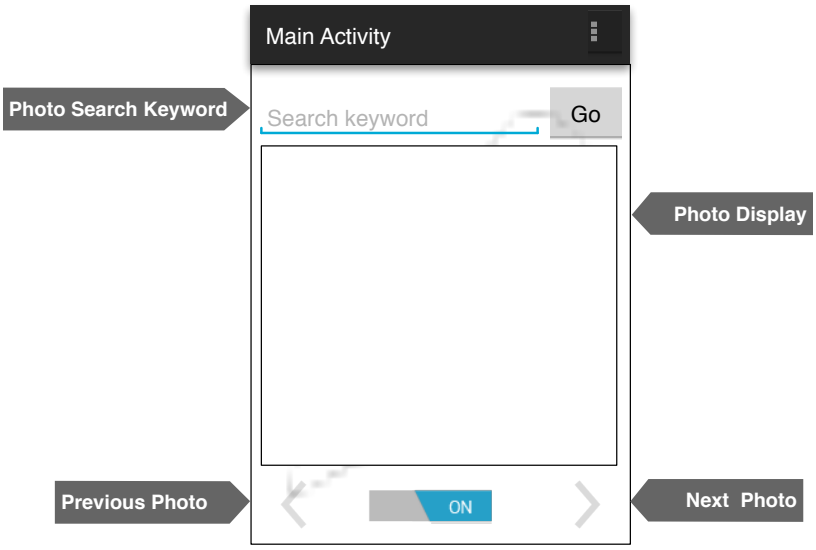
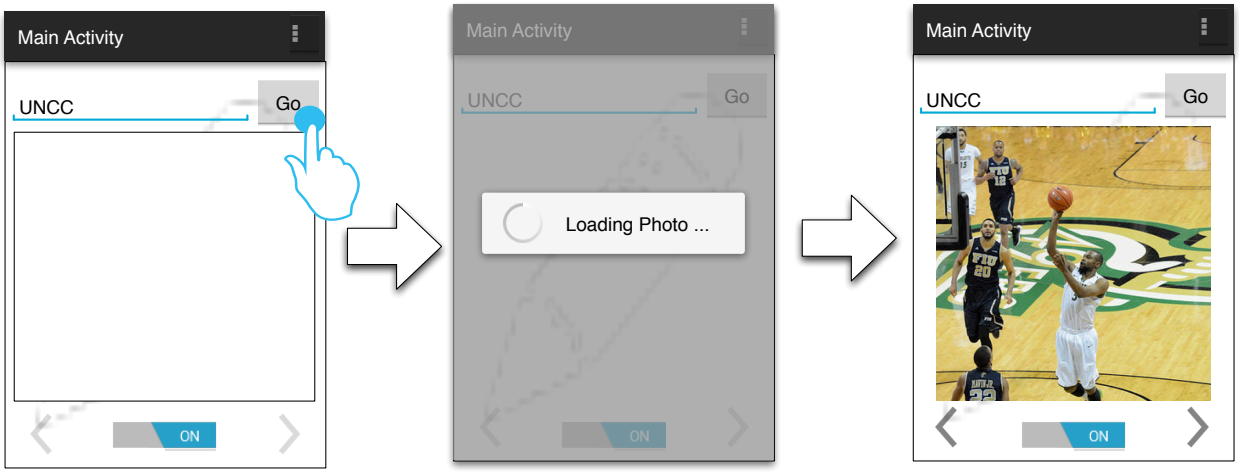


Figure 2, Application Wireframe

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 2. Below are the requirements:

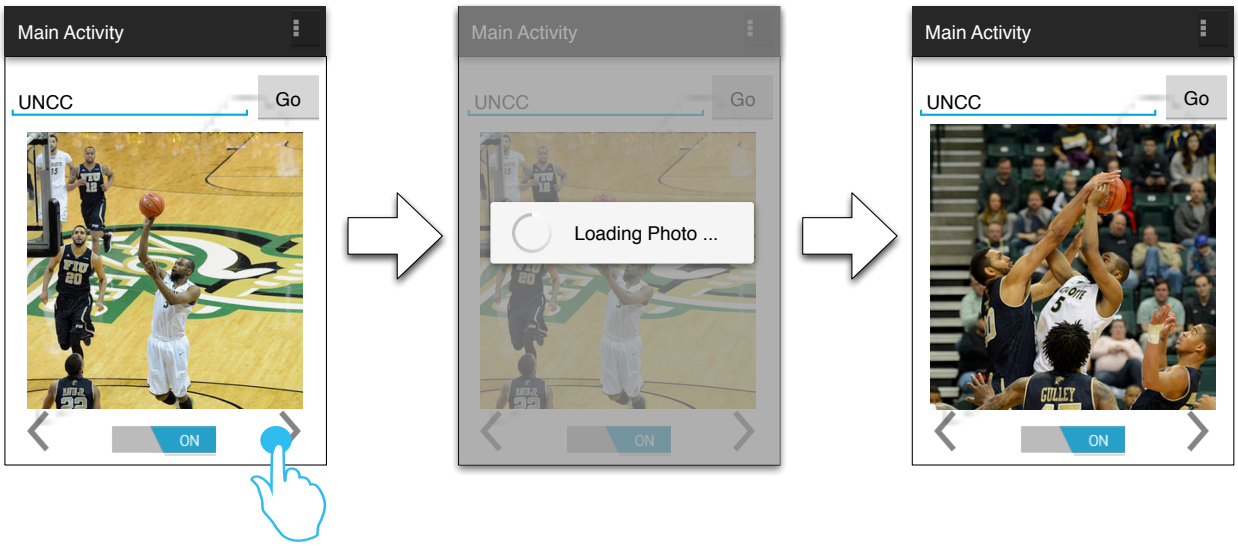
1. Create a new android project called “In Class 5”.
2. The EditText will hold the search keyword to be used to search for photos on Flickr.
3. The Switch at the bottom of the app is used to select the type of parser to be used. If it is set to ON then a SAX parser should be used. If the switch is set to OFF then an XMLPull parser should be used.
4. When the “Go” button is tapped, the corresponding URL should be constructed and the query request should be sent to the server to retrieve the XML document stream.

5. You should use a separate thread to perform data retrieval from the server and data parsing. Do not use the Main Thread to perform these tasks. Use an AsyncTask or a Thread/Handler.
6. Implement the XML parser indicated by the Switch at the bottom of the app and pass the document stream to the parser. Parse the photos URLs and store photo URLs in an Array List.
7. While the data is being retrieved and parsed, and the first photo is being downloaded, 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.
9. You should use a child thread or AsyncTask to perform the downloading of the photos. Do not store the photos loaded, simply download and display the retrieved photos. All UI operations should be performed by the Main Thread.
10. Upon clicking the “Next Photo” icon, you should download the next photo. For example, if the currently displayed photo has index 5 in the array list, then you should download and display the photo at index 6. If the currently displayed photo has index 19, which is the last photo, you should download and retrieve the photo at index 0.
11. Upon clicking the “Previous Photo” icon, you should download the previous photo. For example if the currently displayed photo has index 5 then you should download and display the photo at index 4. If the currently displayed photo has index 0, which is the first photo, you should download and retrieve the photo at index 19.
12. While the photo is being downloaded you should display a Progress Dialog as indicated in Figure 3(b). The Progress Dialog should be dismissed when the worker thread or AsyncTask is done retrieving the photo.
13. 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.



(a) Entering keyword (b) Progress Dialog (c) First Photo displayed

Figure 3, Displaying first photo



(a) Clicking the next photo icon

(b) Progress Dialog

(c) Photo displayed

Figure 4, Displaying next photo