ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 6

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#_InClass06.zip** For example, Group2A_InClass06.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 6 (100 Points)

In this assignment you will make HTTP requests and parse JSON data. Using the 500px API you will retrieve list of photos in JSON format and display them on your own app. The app is composed of three activities, namely **MainActivity**, **GalleryActivity** and **DetailsActivity**.

Important App Requirements:

1. Create a new android project called “In Class 6”.
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.

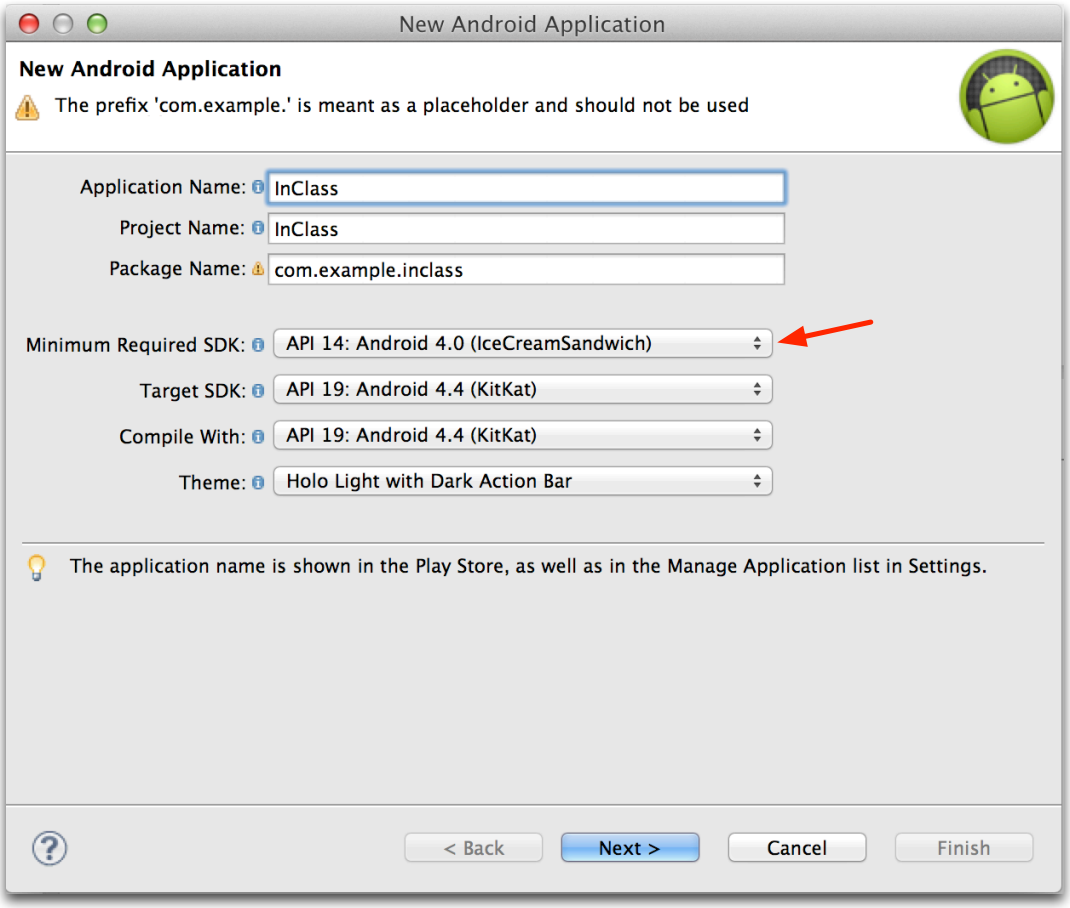

Figure 1. Choosing Minimum Required SDK to 14

Initial Setup

1. Go to https://developers.500px.com/ and create a new account.
2. After creating an account, you will receive an email from 500px you need to confirm your account by clicking on the link in the received email.
3. Go to https://developers.500px.com/ click on the *"**register your application**"* at the bottom of the page.
   a. Name your application "InClass06 App" and provide a short description.
   b. Set the application url to www.uncc.edu/~your-uncc-id
   c. Provide your email as the developer email.
4. The figure below shows the application information, copy the application "Consumer Key", which is required by the photo search api.

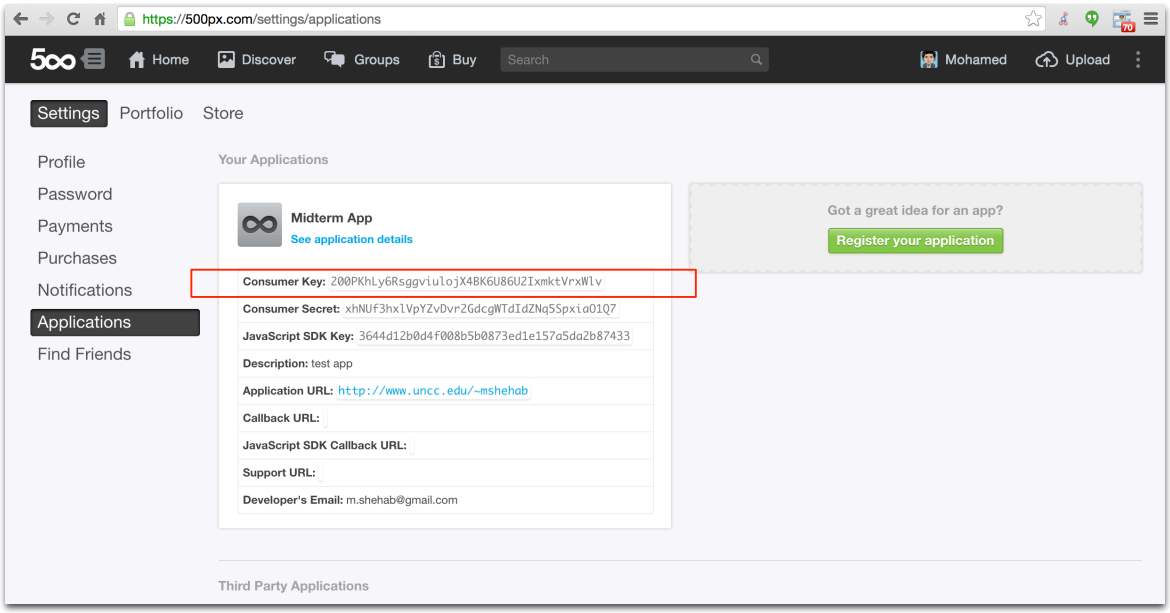

Figure 2. 500px Application Information

Part 1 (5 Points): MainActivity

The Main activity displays a EditText that enables the user to enter a keyword search term. Figure 3 shows the Main activity. The requirements include:

1. Upon selecting a keyword from the list, the EditText text should be set to the selected keyword, See Figure 3.
2. Upon clicking the submit button start the Gallery Activity. The intent should include the search term using extras. You will not be given any points if you use a static variable to share data between activities.
3. If the submit button is clicked and the EditText is empty display a Toast Message indicating that the field was empty and do not start the next activity.

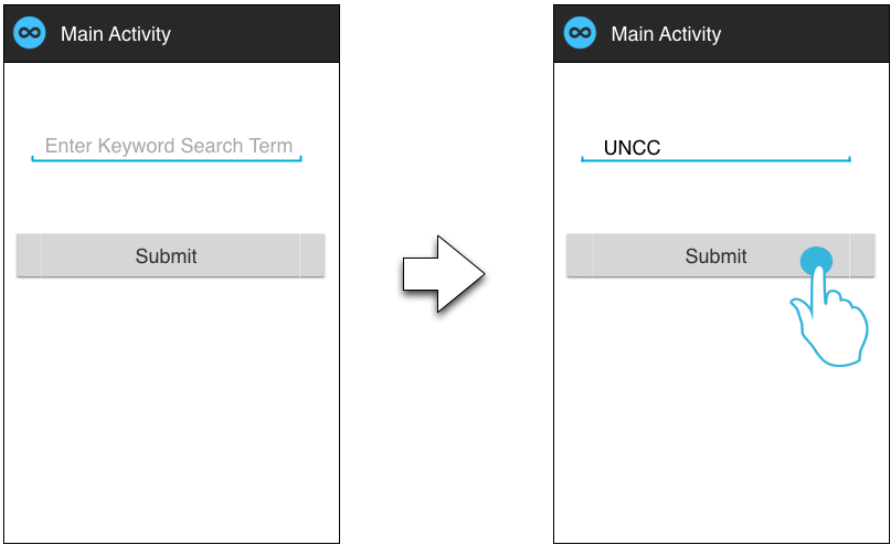

Figure 3. The Main Activity Wireframe

Part 2 (65 Points): Gallery Activity

The Gallery activity is responsible for the retrieving the photo search results based on the search keyword term sent from the Main Activity. The Gallery Activity wireframe is shown in Figure 4. The requirements include:

1. The 500px photos/search api should be used to retrieve the photos matching the requested keyword term:
    - Use the api documentation provided in https://github.com/500px/api-documentation/blob/master/endpoints/photo/GET_photos_search.md
    - You should set the "**consumer_key**" parameter based on your 500px app's consumer key.
    - You should set the "**term**" parameter to the search term keyword.
    - You should set the "**image_size**" parameter to 4.
    - You should set the "**rpp**" parameter to 50.
    - Use the api endpoint https://api.500px.com/v1/photos/search?
2. Send a GET request and retrieve the photo search results for the selected keyword term. 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 you should display a progress dialog as indicated in Figure 4.
3. Create a photo class that stores the photo title, photo url, and owner's name. The JSON parser should return a list of photo objects.
4. The progress dialog should be dismissed after the parsing is completed, and the ListView should be setup and displayed. Each list item should contain a TextView displaying the photo title, as shown in Figure 4.
5. Clicking a photo title should start the Details Activity. The intent should include the required photo 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.

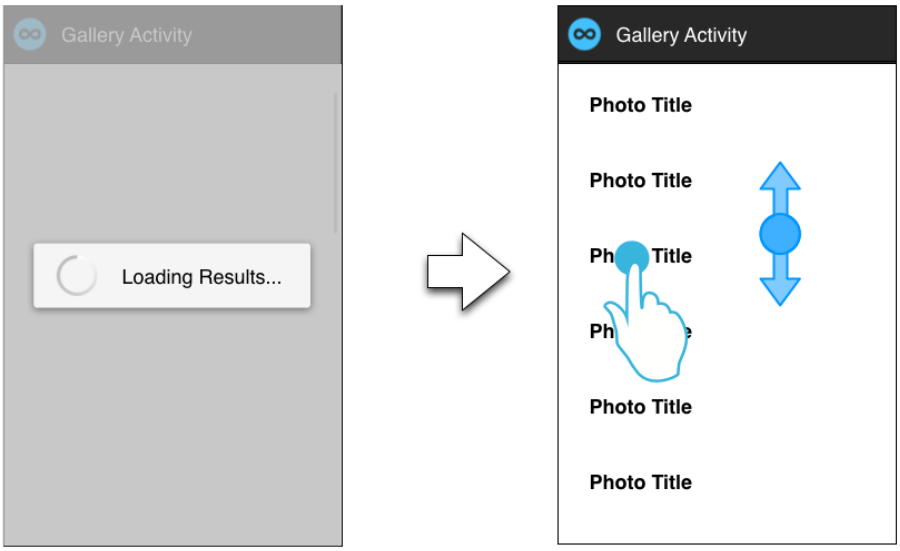

Figure 4. The Gallery Activity Wireframe

Part 3 (30 Points): DetailsActivity

This activity should display the photo details, which includes the photo title, owner name and the photo itself. The DetailsActivity should receive the photo information from the GalleryActivity. Image Download should be done in a child thread. Figure 5 shows the DetailsActivity. The implementation requirements include:

1. Use a thread pool (or AsyncTask) to retrieve the selected photo. Do not use the main threads to download the selected photo. If there is no photo image, use the provided "photo_not_found.png"
2. Once the selected photo is retrieved, display it as indicated in Figure 5
3. Pressing back should end this activity and go back to the Gallery activity.

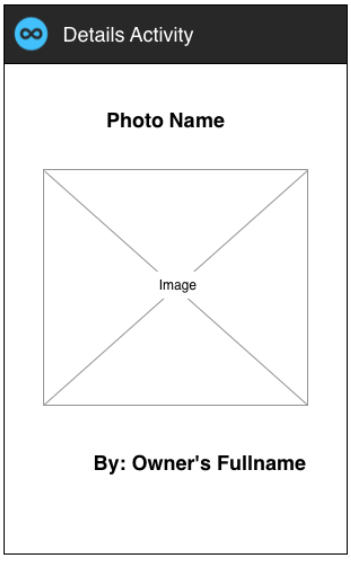

Figure 5. The Detail Activity Wireframe