COMSE 6998-7

Assignment 2 – Due 11:55pm March 5th

In this assignment you will learn basic Android programing and get ideas about REST API calls, which are commonly used in many web services.

You are going to build a 'Photos of This Name' app. You will let user choose a contact from the phone's contact list or input a name, then search on Flickr to get recent photos tagged with that contact's first name or the inputted name, and display search result.

Prerequisites & Resources:

1. Set up Android development environment. Download Android SDK from here http://developer.android.com/sdk/index.html and install it. The default bundle includes eclipse (the IDE) and you're recommended to use this bundle. If you already have eclipse and you wish to use it, choose 'USE AN EXISTING IDE' from the list and follow its instruction. Also you might use Android Studio or other IDE (no tech support from us for this option). Note that we are going to test your app on Android 4.4, API level 19.

2. Create a Flickr API key here http://www.flickr.com/services/apps/create/noncommercial/, and read about REST format http://www.flickr.com/services/api/request.rest.html. The API we will use is http://www.flickr.com/services/api/flickr.photos.search.html. URL format of Flickr images is described at http://www.flickr.com/services/api/misc.urls.html.

3. Read official Android training if you wish: http://developer.android.com/training/index.html.

4. You may use any third-party libraries, as long as you obey their licenses.

Steps:

1. Build UI of your main activity. It might include: a button ‘search’, a button ‘pick a contact’, and a text field ‘name’ (just for your reference. You can do a different design, on condition that your design supports similar functionality).

2. If user wants to pick a contact, use intent to show a contact picker and get result from it. Or user can choose to input a name. Make your app support both ways.
Hint: Intent parameters: Intent.ACTION_PICK, Contacts.CONTENT_URI

3. Code the REST API call.
When user clicks ‘search’, make a desired REST API call. In this case, it’s simply a HTTP GET request. Use the following URL and parameters.
http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api key here}&tags={name here}&extras=date_taken,owner_name,description
Hint: For example,
http://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={api key here}&tags=Emma&extras=date_taken,owner_name,description
It will return data in XML format. Since this is a HTTP GET request, you can try it in your browser.

4. Make your code understand returned data.
After response XML is returned to your app, parse it.
Hint: http://developer.android.com/training/basics/network-ops/xml.html

5. Design & build another activity containing a list view (or if you have a better way to display data, feel free to get rid of that list view)
When your app finishes parsing returned data, it should display all returned photos in a list, along with owners’ names, dates taken, and descriptions. Photos should be loaded asynchronously.
See **Prerequisites** about how to convert photo id to actual URL. You should use small sized versions of photos here, e.g.,
http://farm8.static.flickr.com/7409/12403432974_597f04aef7_m.jpg

6. Use intent to open browser upon photo selection.

If user clicks on a photo, use intent to invoke a browser to show the medium sized version of it, e.g.

http://farm8.static.flickr.com/7409/12403432974_597f04aef7_c.jpg

Hint: Intent.ACTION_VIEW, Uri.parse(url)

7. (Challenge Step – You don't have to do this step but if you complete it correctly you will get bonus points)

You can see that there are multiple pages of search results available for popular names. Implement 'Pull Up to Load More' in your list view.

Hint: Do more HTTP requests with correct paging parameters and update your display accordingly.

Deliverable

Make a main project folder named uni_first_last_hw2. Within that folder there should be the project folder (including all third party library files you use, if any), an .apk file of the built app (you don't need to sign it), and a README file containing one paragraph describing which steps you have done and which you have not. If you use any third party libraries, list their names and your usage in README as well.

```
              [uni_first_last_hw2]
                /           |            \
[project folder]   README.txt     [.apk]
```

Submit this folder as a zip file named uni_first_last_hw2.zip to CourseWorks before the deadline.

Grading:

The assignment will be graded based on:

a. The final output, based on the requirements.

b. The look and feel.

c. Code quality.

d. Understanding of the code written.

The point (out of 50 + 10 bonus) breakdown:

Step 1: 5

Step 2: 5

Step 3: 5

Step 4: 10

Step 5: 15 (including 5 points for loading image asynchronously)

Step 6: 5

Step 7: 10 (bonus)

Coding style, document: 5

Your code will be tested on a virtual device (Nexus 4, Android 4.4 API level 19, 1907 RAM with VM heap 64).

Academic Honesty

You may discuss with others. However you should NOT copy others’ code nor give your code to others, doing so will result in 0 point of this assignment for both copyer and copyee.

Please follow SEAS policy on conduct and discipline:
http://bulletin.engineering.columbia.edu/policy-conduct-and-discipline