CS 646 iPad/iPhone Mobile Application Development
Fall Semester, 2013
Assignment 5

San Diego State University -- This page last updated 10/29/13

Assignment 5
Photo Sharing

Goals

Accessing data via network
Table view and Naviagation
Using JSON
Database access
Interacting with Photos

App description

The app will download and display photos. The course server is at http://bismarck.sdsu.edu/photoserver/. Anyone can access the photos on the server. Only registered users can upload photos to the server. Your app should first show a list of users. When a user selects a user the app shows them a list of photo names from that user. When the user selects a photo name, then the app shows the photo. In this assignment we will do this with three different activities. The app is targeted for a phone, not a tablet.

Downloading a photo is a three step process. First one needs to get a list of current registered users. This is done using the following http GET request:

http://bismarck.sdsu.edu/photoserver/userlist

The server response is a JSON array. The array contains a JSON object for each registered user that has photos on the server. Users that have not uploaded photos are not listed. The JSON object for a users contains a "name" field and an "id" field. Here is an example response.

[{"name":"Roger Whitney","id":"1"},{"name":"Mystery Man","id":"2"}].

To get a list of the photo of a user one uses the user's id in an HTTP GET request. Here is how you request the list of photos for user with id "2". To get the list of photos for a different user you replace the "2" with the user's id.

http://bismarck.sdsu.edu/photoserver/userphotos/2

The server responds with a JSON array. The array contains a JSON object for each photo the given users has on the server. The JSON object for a photo contains a "name" field and an "id" field. Here is an example response.

[{"name":"dog","id":"23"},{"name":"dots","id":"24"},{"name":"moonWeeds","id":"31"}]

To download a photo one uses another GET request. Here is how you get the photo with id "24". Again you change the id to get a different photo.

http://bismarck.sdsu.edu/photoserver/photo/24

The server responds with a document of type "image/jpeg" and the bytes that make up the photo. That is the server sends back the contents of the photo.

Comments about the HTTP GET commands

1. Since these are HTTP ULRs you can and should try them in a web browser. This can help you debugging your app as you will be able to see the server response.

2. The HTTP URLs are case sensitive. Note that all the URLs above are all lowercase.

3. When there is a problem with a request the server responds with an HTTP response code of 404 and a message trying to describe the problem. The response code of 404 will cause the HTTP clients to throw an exception. You should handle this case. The exception will contain the error message. The first point above may be useful when this happens.

4. The ids for the users and photos are for internal use in your app. The ids have no meaning to users so they should not be displayed to users of your app.

Extra Credit

Will include uploading photos, caching photos on devices, showing thumbnails of photos with list of photo names and using gestures to move between photos.

Uploading photos. Allow the user to upload photos from their phone to the server. The user should be able to select a photo from their photo library on the phone and upload it. To upload a photo you need to use the following POST url:

http://bismarck.sdsu.edu/photoserver/postphoto/username/userpassword/photoname

Where:

> **username** is the user name of the account on the server that you are uploading the photo to
>
> **userpassword** is the password of the username account
>
> **photoname** is the name of the photo you are uploading

The bytes of the photo are to be sent in the body of the post in a File Entity (that is not streaming). Your account name is your first name as given in Blackboard. If your first name as more than one word it is the first word in your first name. Your password is the last for digits in your RedID. If successful the server returns a JSON object contain the id of the photo just added. For example:

{"id":"12"}

Saving Photos. Save photos to permanent storage to allow off-line usage and to provide faster viewing. When the app needs to display a photo you first check to see if the photo is already saved on the device.

Saving User and photo lists. Save the list of users and list of photos in the device database to allow off-line usage.

Thumbnail view. When you display the list of photo names add a thumbnail view of the photo.

Swipe gesture. When displaying a photo use a swipe left or right to display the next or previous photo from the same person.

Thumbnail Photo. In the list of photos show a thumbnail image of the photo.

Grading

Points	Item
15	Download list of users using non-UI thread
15	Display list of users
15	Download list of photos for a user using non-UI thread
15	Display list of photos
15	Download photo as a photo, not as a web page
15	Display photo using separate view than the list of photos
10	Swipe gestures (Extra Credit)
35	Saving user and photo lists in database (Extra Credit)
10	Uploading photos (Extra Credit)
10	Saving photos (Extra Credit)
5	Thumbnails

What to Turn in

Create a Xcode project for the assignment. Xcode places the project in its own directory. Place the directory (and all its contents) into a zip file. This assignment we will not use blackboard to turn in assignments. We will use my older course portal (http://bismarck.sdsu.edu/CoursePortal).