

# **Midterm Report**

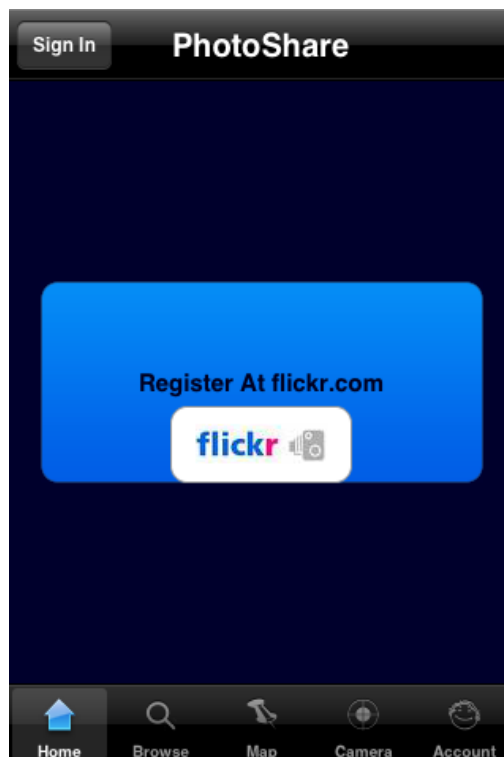
## **Photo Share (iPhotoShare)**

### **COMS E6998**

## **MOBILE COMPUTING WITH IPHONE AND ANDROID**

### **Authors**

**Waseem Ilahi (wki2001)**  
**McClain Braswell (mkb2014)**



<http://www.youtube.com/watch?v=6mNdxv5jFl8>

# Introduction

You are visiting a new city or just going to a place you like. You want to take a picture of what you see. It can be a monument or a masterpiece in a museum or anything that catches your eye. You launch photo share, take your picture, and it locates you using the phone's built-in GPS and automatically tags and uploads the picture to the web.

Now you choose to see pictures that other people have taken from the very spot where you are standing. You can see the Statue of Liberty from every angle, by day or night, under snow or sunshine, by only going there once! You can even see back in time to watch the progress of a famous new building as it is built from the ground up.

In the future, this application could even work with Microsoft's PhotoSynth technology to create large-scale 3D collages to which every user can make a contribution.

The key to such an application is having a large user base. Therefore, we have decided to develop using Apple's iPhone rather than Google's Android, since the iPhone has already become a commonplace device. We will utilize the iPhone's built-in GPS and data connection to tag every photo with geographic coordinates and upload them to the web. We will also use the SDK's mapping controls to provide a visual map of the user's location with pins to show where other users have taken photos. To host and query photos, we will use Flickr, which provides free hosting and a free API with support not only for tagging photos with geographic and timestamp information, but also for searching within a certain radius of a given location.

This is not, however, a Flickr application. We will require the use of a Flickr account for hosting and querying, but we will not provide a full featured Flickr interface. Users will not be able to organize their photos, browse other user's albums, or add or remove friends. While Flickr has not yet released an official full-featured iPhone application, there are third party projects in the works, and we do not feel the need to create another.

Instead, the purpose of our application is to change the way our user interacts with his environment. Rather than spending his time searching through a friend's latest updates, we want our user to spend a few moments to check out what other people thought were interesting about the very place he is standing at any particular moment. We want our user to be able to view the same landmark through the eyes of other visitors who have come before.

## Related Work

Flickr, PhotoBucket, WebShots, Picasa, and numerous other online photo sharing websites have become extremely popular on the desktop computer over the better part of this decade. Most of these services have released APIs and websites designed for mobile access. Even social networking sites, such as Facebook, MySpace, and Twitter now support “mobile uploads” of photos. The limitation most of these mobile services is that they require the use of a web browser. While the iPhone does implement perhaps the most robust and capable mobile web browser on a device of its class, there is nevertheless an inherent inefficiency associated with using Safari to interface with these services.

At the time of this writing, Flickr has not released an official iPhone Application, and when we began this project, there were no Flickr applications available on the AppStore. Since then, we have discovered two similar projects via web search: Darkslide<sup>1</sup> and Mobile Flickr<sup>2</sup>. Both projects aim to be full featured Flickr interfaces. Here is where our goals differ. We do not intend to be a full featured Flickr application. Instead, our goal is to provide a simple way to browse photos taken within a small radius of the user's current location and to quickly snap a photo and upload it complete with location tags. We do not intend to provide services to upload full albums or to browse a certain user's albums. Our application is strictly location based. We do not want our users to be sucked into long browsing sessions (“flickr stalking”) -- instead, we

---

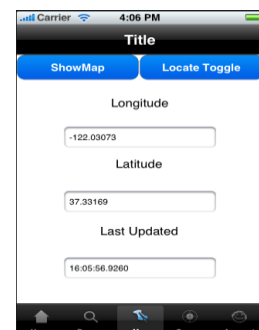
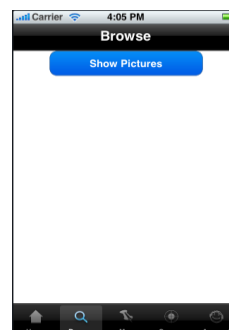
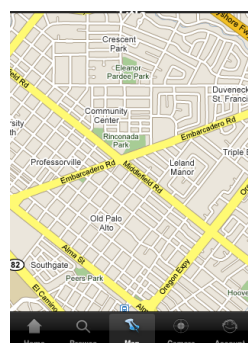
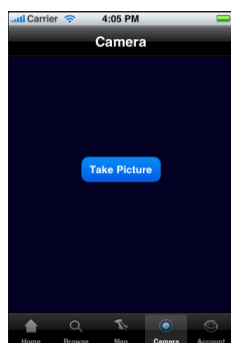
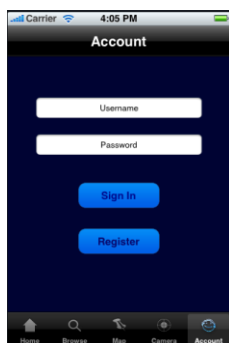
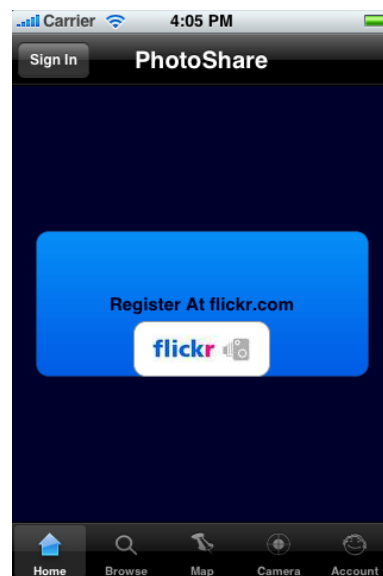
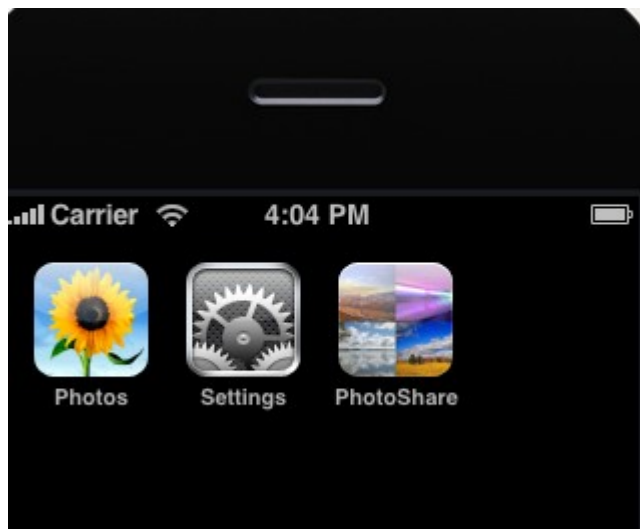
1 Darkslide Official Website, <<http://connectedflow.com/darkslide/>>, accessed March 12, 2009.

2 Apple iPhone School, <<http://www.appleiphoneschool.com/2008/04/15/mobile-flickr/>>, accessed March 12, 2009.

want our users to be able to share photos with others who have traveled the same path. We want our users to be able to travel back in time with a few flicks of their fingers to see what their surroundings looked like last week or last year, to see their surroundings from someone else's perspective. Our application is about interacting with our surroundings and sharing that experience with other users.

## Usage Model

The application launches with a splash screen giving version and author information. From here, the user may access 4 other screens via tabs at the bottom of the screen. In addition to the Home screen, the user may choose Take-a-Photo, Browse, Map, and Account. The account page houses all settings related to the user's Flickr account (required to use the application). Here, the user may sign in with his name and password. A link is provided to the flickr.com registration page for users who do not yet have an account.



The Take-a-Photo screen is perhaps the first thing the user will want to do, and is very simple and straight-forward. Once the user clicks the Take-a-Photo tab, the built-in camera becomes active, and the user can easily aim and shoot to take a photo. If the user is satisfied with the shot, he simply taps “upload,” and the photo is on its way to Flickr, complete with location tags and a timestamp. If the user is unhappy with his photo, he can choose to discard it and take a new one. The user may also choose “cancel” to give up and return to the Home screen.

The Browse screen is where the excitement begins. From here, the user can scroll through thumbnails of photos taken by other users standing in the same spot, only at different times. The user may choose to enter some search terms to refine his results, or tap a thumbnail to view it full screen. In full screen mode, the user can also zoom and pan the image using now-standard multi-touch gestures. From the full screen view, the user may also choose to save the image to his library or set it as his background. The user may return to the thumbnails by clicking the “Browse” button at the top left or click the “Map” button at the top right to jump to the Map screen.

The Map screen is another exciting view. In this view, the user is presented with a full screen map centered at his current location. The map is marked with pins representing photos taken nearby. The user may explore the map using more gestures, or tap on a pin to view full screen photos from the location. Again, from the full screen view, the user may choose to save the image or use it as his background.

The user may easily jump from one screen to another by simply clicking on the appropriate tab at the bottom of the screen. These tabs are always present for easy navigation, except when the user is taking a photo. On-screen dialogs will keep the user informed of every action, for example, if a file fails to upload, he may choose to try again or discard the image.

# Architecture

To use our application, we assume the user meets the following requirements: the user must have a Flickr account (our application provides a link to register online for free at flickr.com) and a reliable data connection, the user must be able to find an accurate reading (or semi-accurate estimation) of the current location via the iPhone's GPS, and in order to upload photos, the user must also have a fully functioning camera. We use the Flickr API for storage and query functionality, in particular, for its ability to store latitude and longitude coordinates, and to search for photos within a certain radius of a certain point. We also use the iPhone's built-in Map control to provide a visual representation of nearby photos.

Once a user has signed into his Flickr account, he may choose one of two basic modes: Take-a-Photo and Browse. In Take-a-Photo mode, the user simply points his camera at something interesting, snaps the photo, and after reviewing the photo, uploads it to Flickr, complete with location tags. The application automatically tags each photo with a latitude and longitude straight from the iPhone's GPS, as well as a timestamp, prior to uploading.

If the user prefers to see what other users have deemed interesting, he may instead choose Browse mode. In browse mode, the user may scroll through thumbnails of photos taken near where he stands, with the closest and most recent photos showing up first. A simple touch of the screen will bring up a full-size view of any image. If the user happens upon a photo he is particularly fond of, he may choose to download it or save it as his background image. Additionally, the user may browse via a Map view. Here, the user will see a map of his current location with pinpoints showing where other users have uploaded photos. The user may then choose to view these images full size. The idea here is to let the images of other users inform the user's decision on where to go next.

For example, suppose our user is exploring a large area, such as Yellowstone National Park, but he doesn't know which direction to go, or he is looking for a particular location that he will only know when he sees it. Rather than exhaust himself by wandering through the whole park, our user may simply browse through the Map view to find a photo of a landmark he remembers from the last time he visited the park, or he may find an area with a particularly large number of photos and decide there must be a reason so many have shared photos of this area. Once the user has found an interesting photo, he can simply orient himself on the map and find his way there.

The key to our application is the Flickr API. We use the Flickr API to tag and upload the user's images and to query other users' images. Specifically, we will use the following API methods:

*flickr.photos.search*

Query nearby photos, passing current GPS coordinates as latitude and longitude parameters of the query.

*flickr.photosets.addPhoto*

Upload photo with latitude, longitude, and timestamp tags passed as parameters. The photos will be added to a default iPhotoShare photoset created for the user.

*flickr.photosets.create*

If the user has not already created an iPhotoShare photoset, we will use this method to create one for them. This photoset will store all the photos they upload using our application.

## Project Status

Nils Hayat notified us about a week ago that he had dropped the class. Due to the departure of the teammate who came up with our application idea, we are not as far along as we intended to be. So far, we have fully implemented the different views and screens of our interface, but the functionality is not entirely complete. In particular, we have not fully implemented our use of

the Flickr API. We have, however, been able to use the iPhone's camera to snap a photo and the GPS to find a location. We are also able to use the Map control to show the user's current location. Functionality related to the Flickr API was originally the responsibility of our other team member, so we have not yet completed this work, however, we should be able to implement API-dependent features soon. We have been able to test our application using both the emulator and a real device.

## Evaluation

The main factor for how usable our application will be is the speed of the data connection. If a user cannot upload a photo or browse thumbnails of other photos in a timely manner, then this application will be of no interest to him. Unfortunately, we are at the mercy of the service providers on this point, but it is our experience that the data connection is certainly sufficient at least in heavily populated areas where our application will be most intriguing, and where there are more likely to be large numbers of photos to browse.

Our application also requires an accurate GPS reading, which again, is something we that is our of our hands. It has been our experience that the GPS is accurate most of the time.

We will judge our application's usability by timing how long it takes to upload a photo, how long it takes to browse through a few pages of photos, and how far off the GPS readings are.

Ultimately, there is not much we can do about these issues. We will judge the overall success of our project based on the number of photos that we help to share with the world.



# Team

The iPhotoShare team currently has two members:

Waseem Ilahi is a Master's Student in the second semester of Software Systems track in COMS. He took OS with the professor last semester. Implemented The basic UI for the app, and also the core functions so far.

McClain Braswell is a Master's student in his final semester of the Software Systems track. His interests include Computer Vision, Graphics, and interface design. He earned his undergraduate degree in English and Math at Columbia, and worked designing web applications prior to returning to school.

The only issue we have had is the departure of our third teammate. The application we chose to create was originally his idea, and thus it was he who had the best mental image of how our application would work. Since he left the project, we have had to reassign the parts of our project. McClain has now had to take over researching and implementing the Flickr API calls, which has turned out to be rather tricky. Originally, The work with Interface Builder and Xcode was split between Waseem and McClain, while researching how to use the Flickr API was to fall to the third member. Now, we have redistributed the work. Waseem is in charge of building our interfaces, while McClain focuses on using the API, with the actual coding work being split between both. We meet at least once a week on campus and work together on most aspects of the project.

# Schedule

Date	Goal	Completed
02/09/09	Define basic list of features.	02/09/09
02/23/09	Finalize mockup of interface	02/23/09
02/23/09	Apply for Flickr API key and familiarize with API	02/07/09
03/01/09	Implement skeleton interface using Interface Builder	03/01/09
03/09/09	Implement camera functionality	03/09/09
03/09/09	Implement GPS functionality	
03/09/09	Implement Map functionality	
03/13/09	First iteration of application	03/13/09
03/21/09	Implement uploading via Flickr API	
03/28/09	Implement browsing via Flickr API	
	Testing and reevaluation	
04/30/09	Final iteration	

## Acknowledgements

Flickr API Documentation:

We used Flickr's online API documentation and how-to pages to learn how to interface with Flickr's services via HTTP requests.

ObjectiveFlickr Project:

The ObjectiveFlickr project provided some demo applications with source code to help us understand how to use Objective-C to use HTTP requests to interface with the Flickr API.

# References

ObjectiveFlickr, <<http://lukhnos.org/objectiveflickr/blog/>>, accessed March 12, 2009

Flickr API, <<http://www.flickr.com/services/api/>>, accessed March 12, 2009

Google and all that there is on it ☺

# Application Flow Diagram

