

CITS4232 Project 2012

Group C

10/30/2012

Group C

Carlo Zanotta Dumit	21203542
Dylan Nangle	220518079
Puneet Verma	21183182
Taiga Yano	20698782

Project Topic

A Simple Photo uploading site

Photo sharing sites make it easy for you to send photos around the world, which is great for keeping in touch with loved ones overseas.

What has been implemented

User registration

Users can register for an account by providing an unique user name as well as an unique e-mail address, along with a password for later authentication. The registration form will perform checks for invalid inputs and syntax client-side (using HTML5 and javascript) to ensure correct registration.

User login

Once the users are registered they can login to their account using their details (username or email) and the password they have set. The arguments provided in the login form are checked with the database. Upon successful authentication users are directed to their home page.

A persistent session is created upon login, and the user will remain logged in until he explicitly logs out.

The user can also set his avatar on his edit profile page, the default option is using gravatar (globally recognized avatar), if the user chooses to upload an image from his computer instead, then that will be his default avatar.

Photo upload

After users log in, they can upload photos on to their profile. Photos are saved onto the database with photo name, privacy setting and group name, for better organisation and the actual image file stored on the server file system using *paperclip*. User may edit the details

and grouping of the photos as well as removing the photos at any time. Upon uploading images are resized to the following sizes: thumbnail, small, medium, large, and real. However only the thumb and large sizes are actually used in the application, the user can select to “view sizes” on the pictures page if he desires to get a specific image size, this resizing wasn’t the focus of our application though.

Browse photo

A user’s photo can be viewed by visiting their page. By default all of user’s photos with privacy setting set to public are visible and all ‘private’ photos are only disclosed to the owner and the members in the owner’s friend list. Photos are displayed as thumbnails and upon clicking on the thumbnail it will enlarge to appropriate size.

All the images in the same group can be viewed in a slideshow format upon clicking on one of them and scrolling through them, also groups can be expanded and collapsed to better visualise the page.

The “Home” page features “recent uploads” gallery where some of the most recently uploaded photos are showcased to everyone. Again, private photos will only show up for “friend” users.

Friend list

Users may request to become “friends” with another user who will be given permission to view each other’s photos that are set to “private”. Users on both ends must agree to becoming friends and the result is a mutual relationship. At any time, either party can drop the request or remove a user from their friend list. When one removes a user from the friend list, the friend status will be removed from both users. If two users are already friends or the request is pending, interface will change accordingly to prevent multiple requests and/or spamming.

Private photos

Photos can be set as “private” which would then only be visible to yourself and those in the friend list. This attribute can be edited anytime within the edit picture page. Whilst a photo is private, the photo will not show up on public users’ queries.

Page Design

The website is HTML5 & CSS3 compliant and mostly the same as the Stage 1 files.

What has been left out

The Geo tagging system for the bonus mark has been left out.

Some of the AJAX functionality present in the first stage (mainly drag and drop functions) was dropped for the second part due to time constraints and unfamiliarity with using rails and javascript together while maintaining database consistency.

The search functionality present on the home page has not been implemented.

Development Notes

During the development of our project we used github for version control, project is available at www.github.com/felix-dumit/CITS4230.2

The website was deployed to mypix.herokuapp.com, however since we are only using a free account, after some time the images are deleted from the server file system. This could be fixed using Amazon S3.

The uploading to csse website works but the server has been experiencing high downtime.

Also there are a few relative paths inconsistencies due to the symbolic link created on the public folder, in particular the images to be displayed by lightbox (javascript image slideshow library) pop up on the wrong place. All of these errors are exclusive to the csse server, on local machine or heroku this problems are not an issue.

Reflection

With the group members all working on different projects it was difficult to arrange group meetings and coding session, infact we never had a full meeting with all of our members

present. Perhaps with more time we could have found a suitable time at which we can collectively discuss our ideas and come to definite conclusions rather than waiting for replies over web-mail.

We may have underestimated the Ruby on Rails syntax complications at first. The concept of having model, controller and view for each entity as well as the database schema which gets assembled on the main view took a lot longer than expected to understand.

This unit along with other units encouraged the use of code sharing, most notably Github, which was a nice practical experience for programming with multiple coders. It became apparent very quickly that regular communication augmented with the ability to concurrently work on a project results in much more efficient production.

Also due to time constraint and the necessity to first familiarize with rails syntax, we were not able to develop using TDD, or do much testing at all, but now that we have more experience a TDD approach would definitely be something worth doing on future projects.