Ben Hutchison

Spring 2010
Independent Study

For my Independent Study, I designed and created a web-based photo gallery. The goal of this project is to learn about and practice principles of good web user interfaces. Several books were very helpful for this endeavor, including Steve Krug's *Don't Make Me Think* and Smashing Magazine's *The Smashing Book*. The result can be seen at https://techhouse.org/cameratron[1].

Audience

The first and most important thing to consider when designing any sort of application is the body of users that the application must serve.

The users of this application are members of Technology House, a Program House at Brown University for students who share a common interest in science or technology. Most users are on-campus, but some users are also House alumni who live in other parts of the world.

In general, Tech House members are very technology savvy, and are good at using computers. However, this does not mean that a carelessly-constructed system will be acceptable. After experiencing a wide variety of websites and applications, Tech House members know what differentiates a good experience from a bad one. Also, since members are both good at using computers and somewhat discerning in their user experience, they have recent versions of modern web browsers.

One of the appointed positions in the House is Photographer, who is encouraged to take pictures at events. This person generally publishes the most photos, although any member is welcome to publish photos to the site.

Goals

The goal of this application is to let Tech House members view and share photos that are related the House. By archiving House events, Tech House can maintain a visual history.

There are already a dearth of photo-sharing sites on the Internet, including Facebook, Flickr, and Photobucket. Why does Tech House need to create an internal photo gallery, when these sites already exist? What makes an in-house system different?

An in-house system has the distinct advantage of being custom-tailored to Tech House's needs and strengths. While Flickr must work for anyone on the Internet, we can narrow the specification for our application. By focusing on this target audience, this application can become far more effective than a generally-available photo hosting site ever would.

Previous System

In 2008, I undertook a similar project of creating a photo gallery web application for Tech House. My goal was to replace the very old, slow, and hard-to-navigate Perl site (referred to henceforth as Mk.1) that had previously served as the House's photo gallery. The 2008 attempt (referred to henceforth as Mk.2), despite being faster and better than the Perl script, was rudimentary, unpolished, difficult, unintuitive, slow, and an awful user experience in general.

It did not handle photos of various sizes very well. They were displayed at 1:1 zoom, so small photos were centered in the screen and large pictures necessitated scrolling to see the entire picture. This was

[1] Users must be on-campus or members of Tech House to access this page.

especially bad for photos that were uploaded at full size from today's cameras, which can reach very high megapixel counts.

Navigation around the application was even worse. There were only two pages, a list of galleries and the photos of a particular gallery. However, there was no global navigation, site branding, or any sort of start-over link, so the site was confusing and very hard to navigate, especially if the user was linked into a gallery without seeing the front page first. Users who wanted to view a different galleries had to use the browser's back button. The site only had a title on the index page. There was no search feature. The list of galleries was not paginated, which resulting in a very tall page that took forever to load due to the number of thumbnails transmitted.

On the plus side, this site let you quickly view photos from a gallery. It used prefetching to cache images, resulting in very fast page loads. Keyboard shortcuts worked, and galleries displayed both a large picture and a scrolling list of thumbnails of other photos in the same gallery.

The upload feature also had pros and cons. In Mk.1, users had to manually SSH into the Tech House server, find the photo gallery directory, create a new folder to represent the gallery, SCP the photos into this directory, and finally run a command-line script to tag the photos. Obviously, this system was rarely used. One of the goals of Mk.2 was to make this process much easier by allowing uploads directly from the website. Users could either SCP the photos into a folder, as before, or put all of the photos into a ZIP file and upload it through the browser, after which the files were automatically copied to a new gallery. The website would then generate thumbnails and let the user add metadata to the photos while seeing previews of the photos. On the positive side, the Mk.2 upload system was a vast improvement over Mk.1, and the barrier to entry for learning the system was low. On the negative side, it was still cumbersome to SCP or ZIP files, the feature was hard to find, and metadata editing would overwrite all previous values.

Off-campus access of the Mk.2 system was poor because the site would always send full-size photos, and Tech House's outbound internet connection is limited to 60 KB/s.

Overall, there were two large flaws with the development of the Mk.2 photo gallery site. First, it was designed to be an incremental improvement on the Mk.1 system, which had an even more terrible interface. Any mistakes from this first system were usually copied over to the second one. Second, Mk.2 was designed with backwards-compatibility as a requirement. This meant that it needed to use the same data structures as Mk.1, which were flat text files that persistently stored photo and gallery metadata.

Requirements

To overcome these problems, I decided not to just make Mk.3 "like the old system, but with the bugs fixed." To really create a good web application, I needed to start with a blank sheet and imagine the best way to take a directory of about 5,000 photos and let Tech House members view and add to it. I needed to generate a longer, more specific list of requirements.

I drafted a list of requirements, ordered by importance. I shared this list with other Tech House members and asked for their feedback on this list and the faults of the old Mk.2 site. They helped me narrow down what sort of features House members want, and some ideas went back and forth a few times as I weighed their pros and cons. By listening to feedback from other House members, I was able to see faults with the Mk.2 system that were not apparent to me. After all, a creator tends to be blind to his own creation's faults, and seeing things from other users' perspectives is the only way to get past your jaded view of your project. Below, then, is the requirements list at the time that I began sketching:

MOST IMPORTANT
Show a big photo Show a list of photos in gallery Show information about current photo Photos fit in screen with manual override Edit photo metadata in viewer Navigate between photos more easily Fast Permalink to specific photos Make sense on Mobile Safari Search for people Video More image compression off-campus
LEAST IMPORTANT

Sections of the New System

PERSISTENT PAGE ELEMENTS

I first sketched the layout of different pages on paper, then build mockups in Photoshop. I went through several iterations for each, and I think the result is a very honed experience.

The design I chose was a clean, minimal interface. I wanted the site elements to be unobtrusive so the photos themselves – the stars of the show – would be the main focus. To do this, I used shades of gray and white for the color scheme, except for crimson elements that were interactive or important. This stood in stark contrast to the Mk.2 interface, which used a black background for viewing photos. The new colors helped separate the different parts of the page using subtle lightness changes, and resulted in a cleaner, fresher, and lighter appearance than a dark, heavy, claustrophobic black site.

Page elements use straight edges, solid colors, and appropriate whitespace. This makes them look more clean and professional, and they are sufficiently unobtrusive to let the photos stand out.

One consideration when I was designing both Mk.2 and Mk.3 was the prevalence of widescreen monitors among my audience, and its contrast with the non-widescreen nature of the photos in the gallery. In Mk.2, this lead me to make the list of thumbnails inside a gallery a vertical scrolling column on the right side of the screen. This made good use of empty screen space, but showed very few photos at a time. Additionally, it was harder to mentally keep track of how far through a gallery you were. In Mk.3, I chose a more conventional horizontal format for showing thumbnails inside a gallery, but I still took advantage of the extra space on both side of the picture by making the site navigation and action buttons a short vertical column in the top left of the page, where users expect to find site branding and navigation.

This column, called the Badge, contained the site's logo that linked back to the list of all galleries, where many users enter. Users expect this to be a "start-over" link, which is very important for any sort of interface. The subtle peace of mind that this sort of link provides lets users reset any mistakes they make in navigation, get back to a familiar place, and be more brave about looking around because they can always get back to where they started.

The Badge also contained links to the top level areas of the site, the list of All Galleries and the Upload tool, as well as a link back to a particular gallery's overview when the user is looking at a large picture. Below this navigation area, past a clear divider, is an area for performing different tasks, such as copying a permanent link to a page or editing metadata. All of these navigation and action links use both text and icons. This helps

both textual- and visual-based users scan the list quickly, understand each item easily, and later remember where they saw something.

In one of the earlier designs for the Badge, I used a lighter gray for the background, a smaller pixel font for the action links, and no icons. The result was tiny, hard-to-read links. By increasing the contrast, adding distinct icons that follow logical conventions for their associated behaviors or link destinations, the site navigation is much easier. Also, the navigation elements have a different appearance when you are on their destination page, giving the user a feeling that they know where they are.

The search box was placed in the upper-right corner of the page, to visually balance the placement of the badge and conform to the convention of search boxes in this location. It uses placeholder text that says Search and a button with a magnifying glass icon, so the purpose of this box is very intuitive for the user.

Next to the Badge is the title of the page in a large font. This is the most important way to tell users where they are in the site. Unlike Mk.2, a user can follow a deep link to any part of this site and still be able to look at the header in the upper left to find out where they are.

PHOTO VIEW PAGE

The focus of the photo view page is the currently selected image, displayed large and in the center of the screen. It never requires scrolling because the server sends a resampled version of the image depending on the browser's window size. This is an example of using Tech House's overpowered, underutilized server to perform additional work for a better, custom-tailored user experience. Also, the user can access the original sized image with a link in the Badge.

Transitioning between images is quick and easy, much like the Mk.2 system. The user can quickly scan through photos with the buttons or arrow keys, and the browser can display the images quickly because it prefetches two photos in advance. For off-campus users, the server applies more compression to the image, so it has a smaller file size.

One of the most significant improvements for Mk.3 is the in-place metadata editing. Users can edit the photo's description, people, date, time, location, and photographer simply by typing over the existing values and pressing the save button or Enter. They can batch edits by changing photos and making multiple changes before hitting save, and the site also lets users cancel pending changes in case they accidentally change something.

If a user wanted to save a link to a particular photo in a gallery, the Mk.2 system required the user to just bookmark the JPEG file. In Mk.3, the browser's location bar changes values when different photos are displayed, and the address shown there always links to the currently visible picture. There is also a link in the action links area that does the same thing for maximum obviousness.

GALLERY COLLECTION PAGE

This page is a good example of my blank-sheet approach to designing this site. In Mk.2, a user would be taken directly from the index page to the large view of a specific photo in a gallery. Since the thumbnail list was so short, it was very hard to tell where you were in the gallery, or to quickly get to a specific photo that you wanted to see.

For Mk.3, I implemented a new type of page, which shows a grid of thumbnails for each photo in the gallery. This let you get a sense of the size and content of the gallery, and it let you seek to any photo.

After testing this layout, I realized that it looked very similar to the site's index page of all galleries. I wanted each type of page to have a very distinct layout so users could recognize where they were and what they could do as easily as possible. I also became annoyed every time I arrived at this page because I was presented with lots of things to choose from, even though I usually wanted to do the same thing each time: view the gallery from start to finish. After applying Krug's squint test, I determined I needed something different from the All Galleries page that was large enough to be obvious. I decided to make the first

thumbnail four times as large as the rest. This helped greatly by making this page very distinct, obvious on a very coarse detail level, and it visually guided the user to click the first thumbnail to start at the beginning of the gallery, which doesn't make the user think.

UPLOADING PHOTOS

The Mk.2 photo upload facility was cumbersome and did report any status to the user. It was hard to get a group of files transferred, and it was almost impossible to add photos to an existing gallery without destroying the old metadata.

After talking to some Tech House members and former House webmasters, I decided to use a Yahoo YUI Flash/JavaScript utility to handle file transfers. This allows users to select multiple files at a time from their computer to upload, and it handles the file transfer process. Unlike similar Java utilities, the YUI Uploader does not trigger a certificate warning, and it can be seamlessly integrated into a page's interface.

The Mk.3 Uploader also makes it easy to add to existing galleries. The user can either type in the name of a new gallery or select an old gallery from an easy-to-scan dropdown menu, with autocomplete. Metadata will never be erased, and users can go back as many times as the like to add more photos.

As the upload process moves along, the user sees progress bars for each file, status messages reporting exactly what the Uploader is doing, and a pie chart of overall progress. There is a cancel button, and canceling the process will not leave the system in an inconsistent state.

What I Learned

Looking back on the difference between my old Mk.2 photo gallery site and the new Mk.3 version, I think I have made a substantial improvement in my design and creation of simple, easy, effective user interfaces. I cannot I believe some of the mistakes I made on Mk.2, and I understand why many users thought it was not very good.

Krug's book *Don't Make Me Think* was a great influence. I am impressed by the quality of his advice, and I tried to incorporate many of his good ideas into my design. Here are some of the points I liked the most:

- Page should be self-evident, obvious, and self-explanatory.
- When deciding between two designs, go with the one that makes the user think less.
- Use fewer words.
- Users should not have to look for what they want, it should be right there.
- Don't use weird names that people won't understand.
- Clickable things should be obviously clickable.
- Users should not have to solve any puzzles.
- Users should never have to ask where they are, where something is, or where to begin. They should never wonder what the most important thing on a page is, or where a name came from.
- A good interface explains itself, but the best interface doesn’t have to explain itself.

After thinking about these points, I revisited my mockups and analyzed them with a more critical eye. I found many places where I was leaving the user to guess where they were or how to do something. In many cases, I wanted to avoid clutter, so I was afraid to add more elements to my pages. However, after sketching and thinking and revisiting enough, I feel that I was able to come up with a design that optimizes both ease of use and aesthetic quality.