Development Document

1. Team Information

1.1 Team Name

Misfits

1.2 Team Member

Semih Energin	se2302
Richard Michael Boyle	rmb2189
Ryan Jones	rlj2122
Jian Bao	jb3381

2. Development Process

The development process for team Misfits started with initial planning stage, where form and function of the design were first explored.

As a first step, we attempted to carefully look at the assignment guidelines, and to understand our requirements and constraints. One important design decision made at this point was to see the Facebook search bar as a tool which should be enhanced and not replaced. Rejecting the search bar completely would disregard one of the most common actions a Facebook user performs. At this point we considered how to describe the core of the search bar.

In considering Facebook search, we realize that it is a node-centric mechanism by which user leap to distant parts of the Facebook graph. The power and simplicity of this tool is it's rejection of edge viewing and traversal. Node's are found and viewed based solely upon their own data. With this in mind, we realized our goal would be to more comprehensively welcome nodes into the search mechanism to enhance the way the results can be viewed and compared- more nodes available, more data per node.

This theme guided our critique of Facebook's current search. One weakness we saw is the small number of results displayed. While this implementation is sufficient for your average Facebook website user, it is insufficient for our power-user target audience. The linear vertical orientation keeps results from using the whole space of the window, while pagination allows for only a small number of items to displayed without a new request for more. Also, extended information is provided either through links or hover-enabled windows. This increases the difficulty and time cost of comparing node information.

With these problems in mind, we shifted to the design phase and started sketching prospective solutions as a group. After sketching and evaluation, we decided upon our core initial design of a coverflow results viewer combined with a space for viewing more information. This point in time marked a shift in division of responsibilities that would consists for the duration of the project. At the highest level, coding tasks were separated from non-coding

tasks. One of the four team members had very little coding experience, he generally took responsibilities for early non-coding tasks. Coding tasks were generally divided by different GUI component. These were later compiled together to make the full application. Semih worked on the coverflow component, Ryan worked on the search component, Richard worked on the more information component, and Jian worked on the lists.

3. Target users

Since our application serves as a general purpose enhancement to Facebook search, almost any current user of Facebook holds a potential for interest. Searching is so important to Facebook because it allows people to jump to distant nodes without traversing the edge path. Our version of search maintains this focus of "nodes are king", as it is a familiar concept to users.

In this light, the target user is your average Facebook user: a web-savvy computer user who understands the social media sphere and has both the aptitude and the desire to navigate its treacherous pathways. Even a clumsy web user can feel safe using our app, as none of the potentially hazardous input fields of the online version are present.

The form factor of this app is a windowed desktop application on a personal computer. Because of the extensive audience of our app, we can generalize our average user as the mainstream Facebook user, and assume an age range from 14 to 50 of either gender as well as proficiency in written English. We will assume full mental and physical faculty, and a broadband equivalent Internet connection.

4. Design Decisions

4.1 Visibility of system status

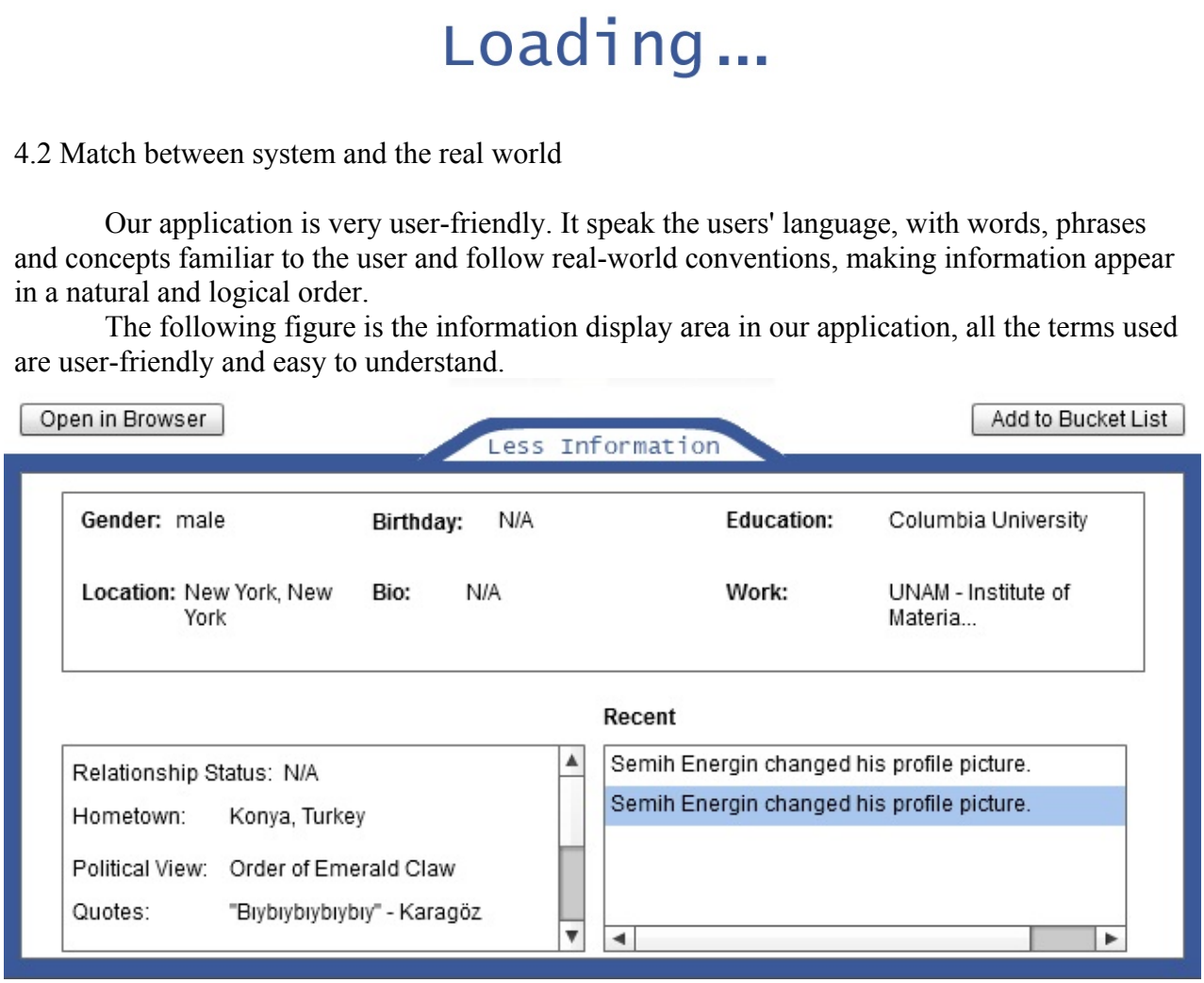
Our application always keep users informed about what is going on, through appropriate feedback within reasonable time.

4.1.1

When you entered the authentication information and click "log in", System will inform user as following.

4.1.2

While the searching is in progressing, the coverflow area will inform user as following.



4.2 Match between system and the real world

Our application is very user-friendly. It speak the users' language, with words, phrases and concepts familiar to the user and follow real-world conventions, making information appear in a natural and logical order.

The following figure is the information display area in our application, all the terms used are user-friendly and easy to understand.

4.3 User control and freedom

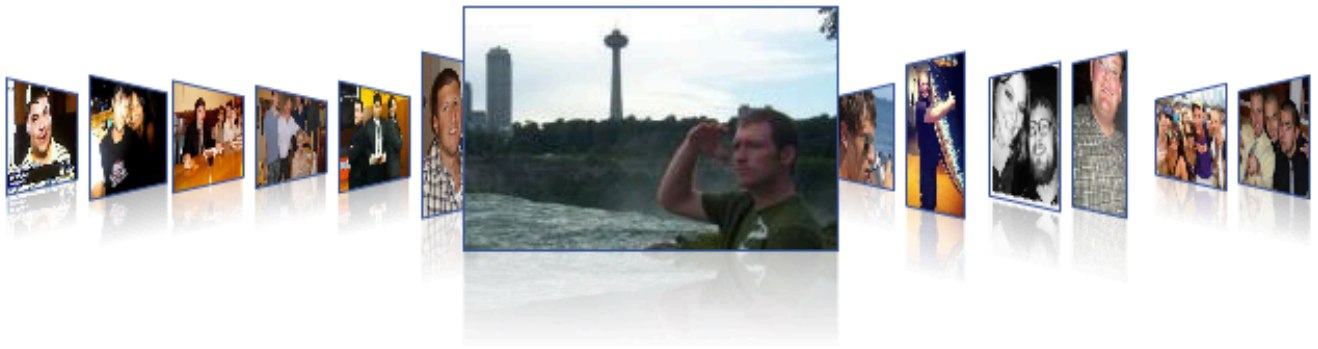
In our app, user control is streamlined by the lack of states and the fact that the main user window is mostly static. No Facebook graph data is modified, and objects are essentially read-only. If a user accidentally deletes a Bookmark item, it can be immediately re-searched and added again.

4.4 Consistency and standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Direct use of the Graph API entailed using Facebook objects such as people, groups, and events. These concepts are familiar to Facebook users and recognizable as search results, even using the same icons and ordering. The location of the search bar is at the top of the screen, and proximity of result object descriptors such as people, group, etc. allow the search bar to be recognized without being explicitly labeled.

The coverflow widget is also a familiar GUI component, and tactile navigation by clicking on adjacent images is recognized as standard.



Coverflow in FacebookSearch++

4.5 Error prevention

In our app we eliminate error conditions by deactivating GUI components when their use is inappropriate, and by pausing user interaction during changes in data or state that are error-prone. For example, the search bar is only active when a search query has been entered. No uncaught error occurs in the application.

4.6 Recognition rather than recall

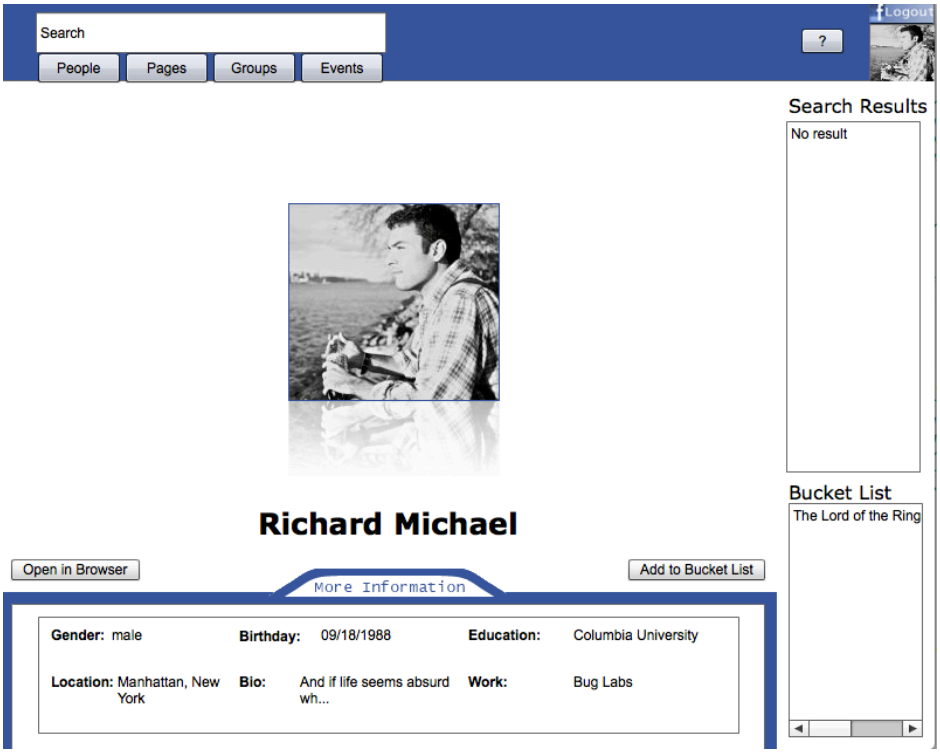
The application minimizes the user's memory load by only having one application state. The application does not leave a state after logging in. There is a constant coverflow area, a search result list, search bar and a more information section. This allows for the user to know where they are at all times.

4.7 Flexibility and efficiency of use

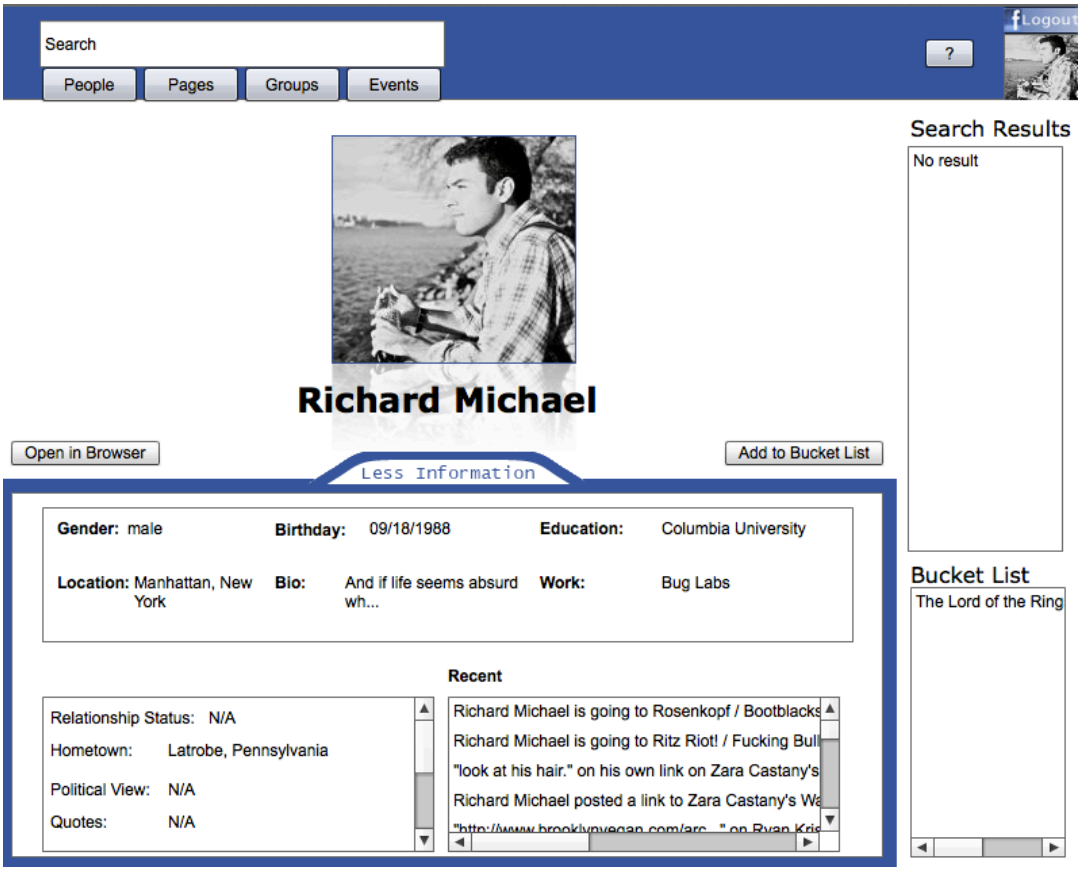
The flexibility of the application is shown by the number of ways for the user to scroll through the results. The options to move through results are using the left and right arrow keys, scrolling through the search result list, and clicking through the cover flow. An other efficient options that the application allows for is the bookmarking of objects. This allows the users to return to a state they were in previously.

4.8 Aesthetic and minimalist design

The design is minimalistic in that information is shown for single objects only. The application also hides detailed information in a toggle-able GUI component. This keeps the screen uncluttered allowing the user to search through results in a focused manner. Dialogs are also limited, the only ones that occur are whether or not a user would like to exit the application, allowing the user to focus on the search.



Less information minimalist state



More information state

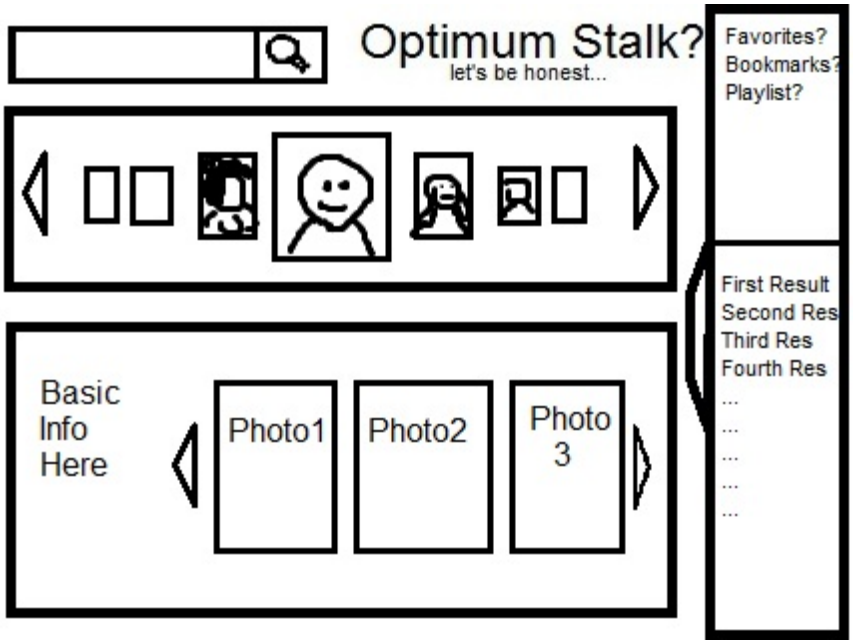
4.10 Help and documentation

The application has a help icon at the top of the application page which directs the user to the user manual. The user manual and documentation is an in depth document on how to use the FacebookSearch++ application. The document is split up into sections specific tasks the user might want to complete and a map of different parts of the application.

5. Prototyping and Testing Process

Prototyping:

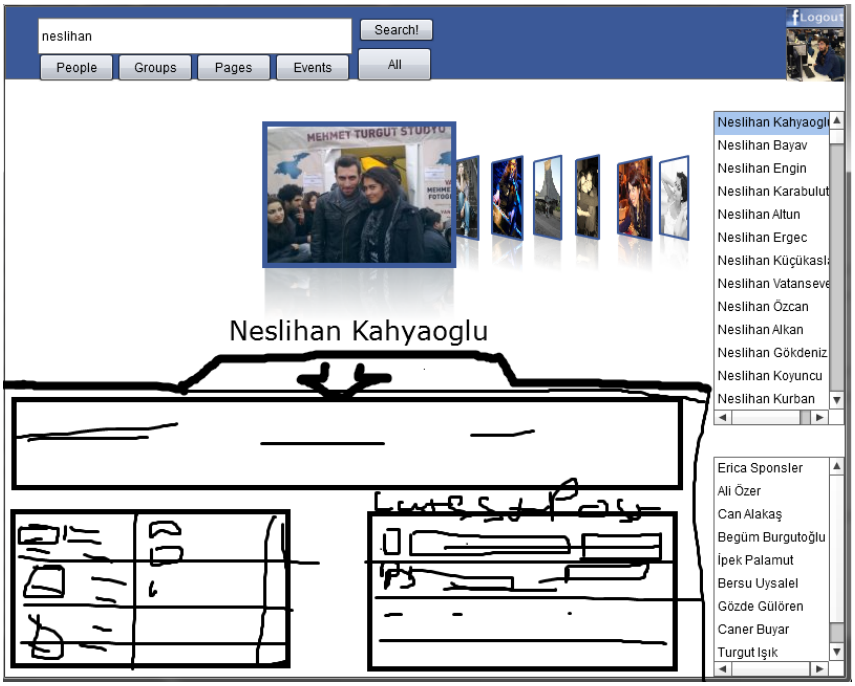
Initial Prototype: Login Screen



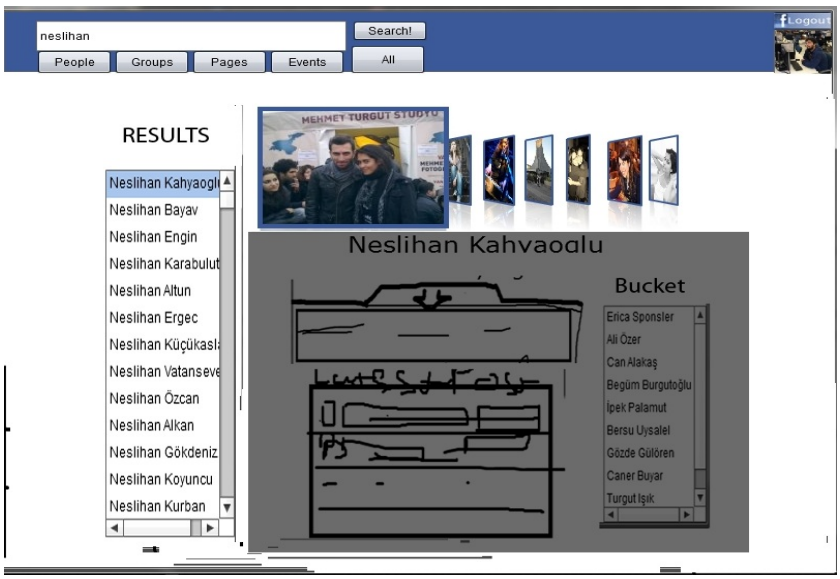
Initial Prototype: Application Window

The prototyping stage was completed by comparing a series of prototype mock-ups. The above diagrams show our initial design decisions. The FacebookSearch++ application window

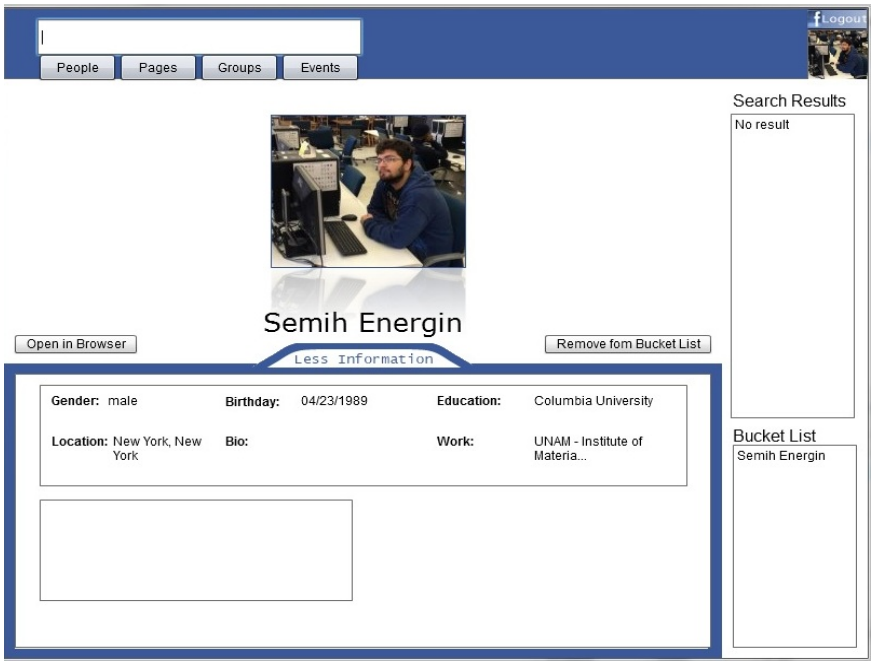
remains static other than the login screen and the main application window. The only thing that changes is the information within the different shelves. This lead us to decide that simple drawings would suffice in our prototyping stage. The prototype was tested with a series of story boards to understand the work flow.



Proposed Change on Information Display



One of the suggested ideas



Final decision reflected on prototype

After the initial sketch we started implementation and formed an initial prototype that can be discussed on. On this prototype we make some tests amongst ourselves and decided to make fundamental changes. The whole feel and purpose remained, but many of the components were re-defined during implementation and discussions. For example, sketches above reflect one of such discussions. We decided that for many search results users won't have permission to view all information and the allocated space would be too much for other results (pages, etc.). So we wanted to allow user to toggle between detailed and brief information viewing. A "drawer" concept was introduced and possible designs regarding it was discussed.

Similar discussions and sketches were developed either on computer or paper about search buttons, lists, the concept of bookmarking and the general integration of the components.

Testing and Feedback:

After an initial program was completed we tested its functionality using a small set of non-teammate users. Two different users were each asked to search for a friend, a group and add items to the Bookmarks. They were told to consider user friendliness and perception of speed and ease.

Our first test user searched for a friend of FacebookSearch++ and noted that the application while loading all of the images took a rather long time to populate so it was confusing to her whether or not to continue. This lead to the design decision to add a "Loading..." window to let the user know the current status of the application. This same user also noted that while the search results were being loaded they noticed that the buttons were still active and they tried to do another search which was impossible. This lead to greying out of buttons to let the user again know the current state.

While using the application, the second user was confused by a number of terms on the application window page. The term “Bucket List” meant nothing to him and he was confused what he was supposed to do with it. Instead the term “Bookmark” was used. This test subject also noted that the extra button click required from choosing a search category and then clicking search was too much work. The design was changed to allow for the category click to submit the query. They also mentioned clicking on the posts and directing to the post page in the browser. We explored this and could not find a good solution to get to the post page.

6. Software Engineering

Tools:

- FlashBuilder
- CogTool
- Photoshop

Widgets:

- DisplayShelf as3 widget
- TiltingPane supporting as3 component