ProtoSearch :: Development Document

Team Prototypr

Team Members

Chatura Atapattu (cpa2116)

Muzi Gao (mg3194)

Xiaolong Jiang (xj2132)

Aditya Pipersenia (ap2904)

Overall Process

When presented with the project criteria, our team met and discussed how we currently used Facebook Search to find friends, acquaintances, groups, interests, etc. We realized that in addition to a linear list being unhelpful, search results should show why they are relevant and the connections between the user and the result. In a social network, where the emphasis is on connections and social relationships between users, the results must show that link.

With this in mind, we derived the two user scenarios presented prior. Specifically,

- Mark, who is searching for "Julia", whom he met at Nick's party the other night
- Diane, who is looking to get more involved with and learn more about Photography

Upon finalizing these user personas and scenarios, we then brainstormed on possible user interface designs that might help show relevant answers to these queries. After some initial sketches, our team gravitated towards a graph approach, where we display the end results and then connections (common interests, mutual friends, etc.) between the result and the user.

Our team then paper prototyped the initial sketches, forming a paper based UI that we were able to present to friends and demo ourselves, to obtain a feel of application flow. Using this feedback, we then embarked on developing the Adobe Air application, breaking down the roles and responsibilities as follows,

- Chatura Atapattu
    - Designing/developing the base application structure/view/tabs
    - Designing/developing the data classes, integrating with the Facebook API and handling data processing
- Muzi Gao
    - Designing and developing the search result/UI elements and components
    - Drawing of the final search results on the screen
- Xiaolong Jiang
    - Creating custom UI components to use in displaying search results
    - Dynamically determining the location and connection of search results
- Aditya Pipersenia
    - Developing weighing algorithm and processing of search results
    - Designing and developing data classes

After an initial version of the application was ready, we demoed with a couple friends who gave us feedback on the paper prototype, allowing them to test the application and provide further feedback on ease of use and intuitiveness. Using their feedback and determining the priority of the bugs/features discussed and the time frame available, we went about fixing bugs and making improvements to our application.

In the final iteration of the application, users can simply enter a search term in the search textbox and click the search button. The application accepts any search term and queries Facebook for

users, events, groups and pages that match that search result. We then takes those search results and evaluate how connected they are to the user, through mutual friends and interests. Based on these results, we score the search result and show the most relevant results on screen. The top 6 search results are displayed on screen, showing the connections between the user and the search result. Users can then click on either the search result or the connections to open a new browser tab to that Facebook page to view more information about it.

During the development of the application, we came across a lot of features and optimizations that we would like to implement. Ideally, the search algorithm would do a better job at determining which search results are more relevant. However given the UI design nature of this course, we emphasized the development of the UI over the application. Additionally, the quality of the search results is directly influenced by the data accessible to the user. However, the majority of user data on Facebook is private and not available for use and scoring the search results. Lastly, given the time, there are a lot of UI enhancements we wanted to implement, such as relative distance and size based on search result importance, dynamic sizing and location of search results on the screen and better UI finesse with regards to the components.

Target Users

The user class of ProtoSearch is generally widespread, but for this project, we are limiting the scope to that of typical college students who use Facebook for socializing, expanding on their social circle by meeting new people and getting involved in new activities.

Personas

Mark is a freshman college student currently declared as a finance major. He is familiar with using Facebook to keep in touch with his friends, although he is generally not technologically savvy. Mark is from a rural town and has only been using computers and the Internet extensively for the last two years.

Diane is a sophomore English major at Columbia University. After a rigorous freshman year, she has learned to use computers to search for information on the Internet. Diane has had access to a computer since she was ten and is able to perform all non-complicated and technologically advanced tasks.

Use Scenarios

Mark attended a party last night at his friend Nick's house. At the party he met Julia with whom he got along. A couple days later, he thought of sending Julia a friend request on Facebook, however, all he knows is her first name, what she looks like and that she is friends with Nick. Currently, searching

Julia on Facebook returns hundreds of people named Julia, some at his university, and others from all over the world. There are even some groups with Julia in the name, but he does not see the girl that he met in the first page of results. He tries filter by just people, but even then she is not listed. When using ProtoSearch, Mark will simply need to enter "Julia" in the search field and hit enter. The search results will show in a graph in which the mutual friends between each Julia and Mark are displayed, showing the ones with most mutual friends first. When he sees the correct Julia who has a connection through Nick, he will have found the right person.

Diane recently bought a DSLR camera on Black Friday. Equipped with her new toy, she wants to learn more about photography, perhaps see which of her friends share the same interest, go to a photography event her friends interested in photography are going to or join a group with her friends interested in photography. However, when she searches for "Photography", a bunch of groups are returned in the results, none of which tell her which of her friends are involved. Diane is left with no information on which group, event or friends are right for with regards to learning more about photography. When using ProtoSearch, Diane will type in "Photography" and click "Search". The search results will be presented in a graph, in which it will show which of her friends are interested in Photography, which photography groups her friends are parts of and what photography events her friends are attending.

Design Decisions

- To display more information about a search result, use a new browser tab within the application. This decision was made for consistency and to match between the system and the real world. While we could have opened a browser in the system's default internet browsing application, we wanted to keep the user experience consistent and contained within the application. However, the browser tab we open is similar to any browser that users may have prior experience with.
- To keep the design minimal and aesthetically pleasing, we decided to show a simple search text box and button. While we considered having a drop down which allowed users to narrow down their category of results, this made the UI unnecessarily complicated and cluttered the appearance.
- To keep the user informed of the system status, we utilized busy indicators and greyed out modal states that prevent the user from making changes while the application is processing. This state also keeps users informed on whether the application is loading, logging in or searching, etc.
- In addition to keep users informed of system status, we decided to display the logged in user's profile image, name and a link to the user's profile page. Initially we had the image be a link to the profile page, but testing showed that users were not aware they could click on the link. Thus we added a separate link to the user's profile page.

- To speed up processing and determining more appropriate search results, we decided to prefetch and locally cache relevant data. Users who had many friends, slow internet connections or computers can take a while to process the search results, thus to speed it up we locally cached the data. Upon loading, it will look for any new information. Users may also clear and refresh the data used for processing.

Prototyping and Testing Process

When prototyping the application, we chose to utilize a single iteration of paper prototyping. With prior experience of paper prototyping, we felt it would be the easiest, fastest and most beneficial to understanding the interaction between the user and the application. Furthermore, given the initial sketches of the application, it was easier to utilize pen, paper and scissors to create a prototype. The paper prototype was evaluated by the team members as well as friends of the team, (specifically Alex Binkley and Edward Kozek, individuals who have had a lot of experience with startups and numerous applications). Their feedback and our responses can be summarized as follows,

- Edward noted that it would be nice to be able to open his own profile page within the application. However, he was unaware that he could click on his profile image to open a tab, and found that quite unintuitive. We decided to add a link next to the name that a user could use to open their own profile page.
- Alex found it very useful to see the connections between the search results and himself, however, he wished that the more relevant results were more prominent, either by being bigger, thicker lines, closer to the center, etc. We felt this was a great idea and something we would like to implement, however given the time available and the priority of the feature, decided we would not work on this feature for this iteration.
- Edward added that having a drop down box in the search container cluttered and complicated the design, and a more minimalist approach would be better. We took his advice and removed the drop down.
- Alex also suggested that when there were a large number of connections between the user and the search result, the UI was very cluttered and it was hard to distinguish the connections. Thus we chose to limit the number of connections and consolidate the mutual friends into one scrollable list. This allowed showing all the mutual friends but not cluttering the UI too much.

The screens utilized in the paper prototype and their descriptions are available in Appendix A.

Software Engineering

In order to develop the application, we utilized the following software, libraries and code:

- Adobe Flash Builder 4.5 (http://www.adobe.com/products/flash-builder.html)
- Facebook ActionScript API 3.0 Modified (http://code.google.com/p/facebook-actionscript-api/)

  For our application that is very data centric, we heavily utilize batch requests. The unmodified version of the API allows 20 requests per batch, although Facebook allows 50. We modified the API to update it to allow up to 50 requests in a batch, as well as some minor bugs related to HTTP requests and data handling. We also discovered a bug we were unable to fix, that is caused by timeouts when retrieving data from Facebook. This bug is more prevalent on slower internet connections and processors that are unable to process the HTTP response data as it is received.

- Creating a Flex Facebook Application tutorial (http://www.adobe.com/devnet/facebook/articles/flex_fbgraph_pt1.html)

  We utilized this tutorial to learn about using the API and for the Facebook login/logout images.

- TerrificTabBar by Justin Shacklette (http://saturnboy.com/2010/08/terrifictabbar-custom-component/)

  We utilized TerrificTabBar to render tabs for our application.

- Web browser tabs (http://jeffreypalermo.com/blog/adobe-air-web-browser-with-flex-4-1-sample-application-available/)

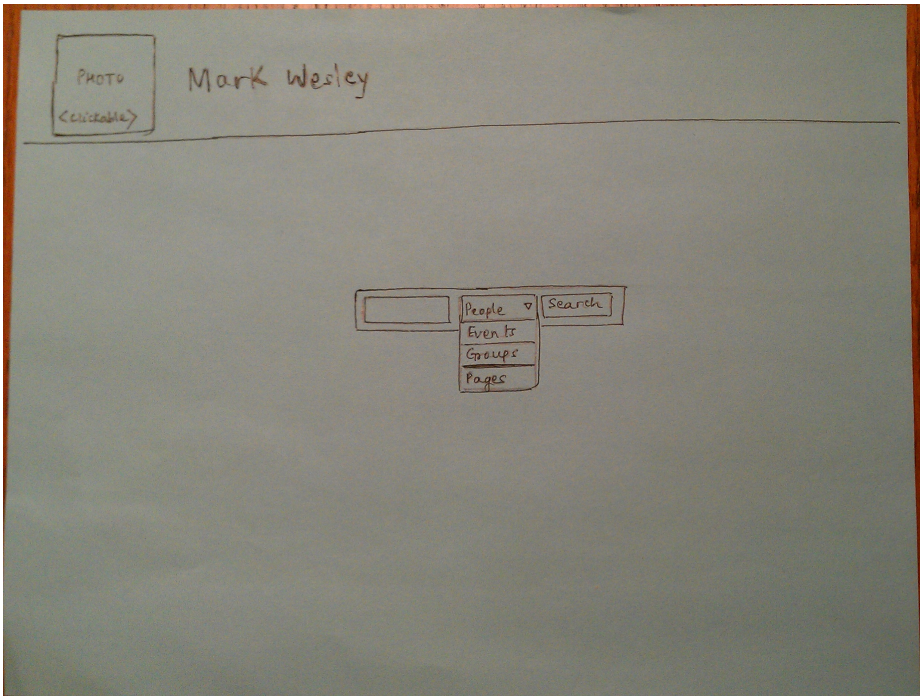
  We learned how to create browser tabs and adapted a lot of the design and code for rendering a browser tab from this website.

- Adobe Air Mobile Components Library

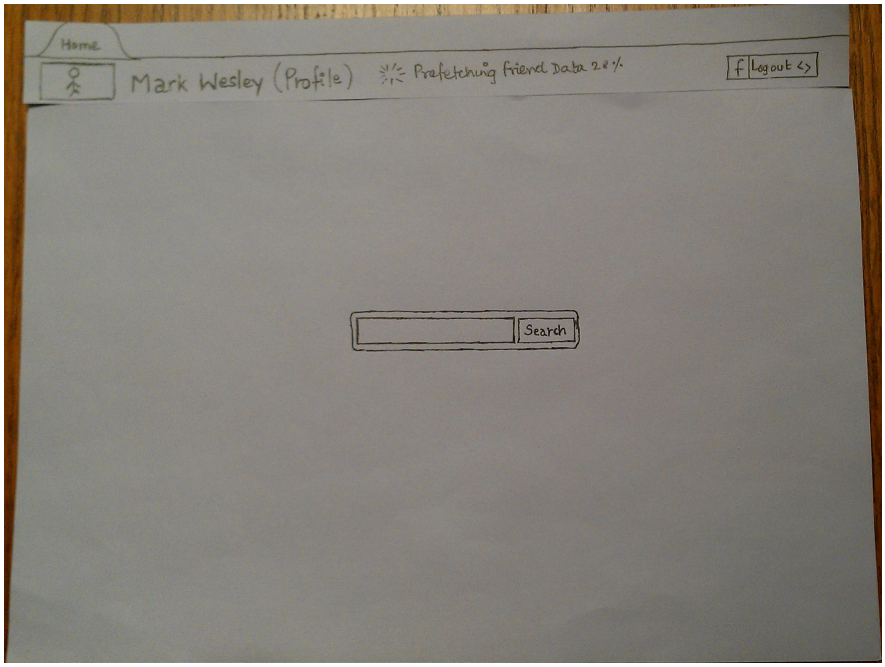
  This library contains the busy indicator component used within the application.

Appendix A

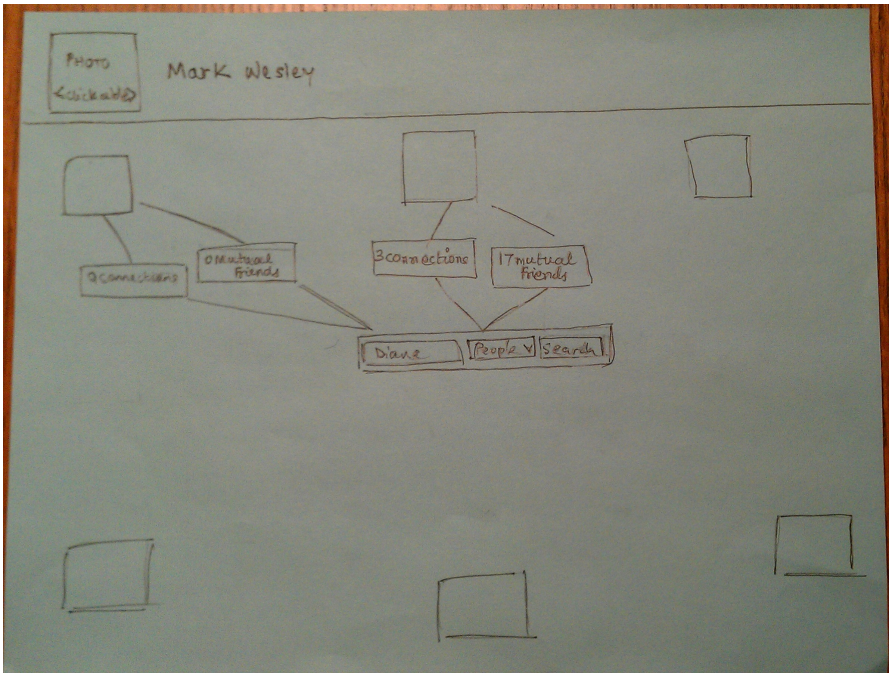
The first home screen prototype which we presented for user evaluation in which users suggested the removal of the drop down and simply search all types of results and add a separate link for the profile instead of having the image be a link because it was not intuitive at to the user.



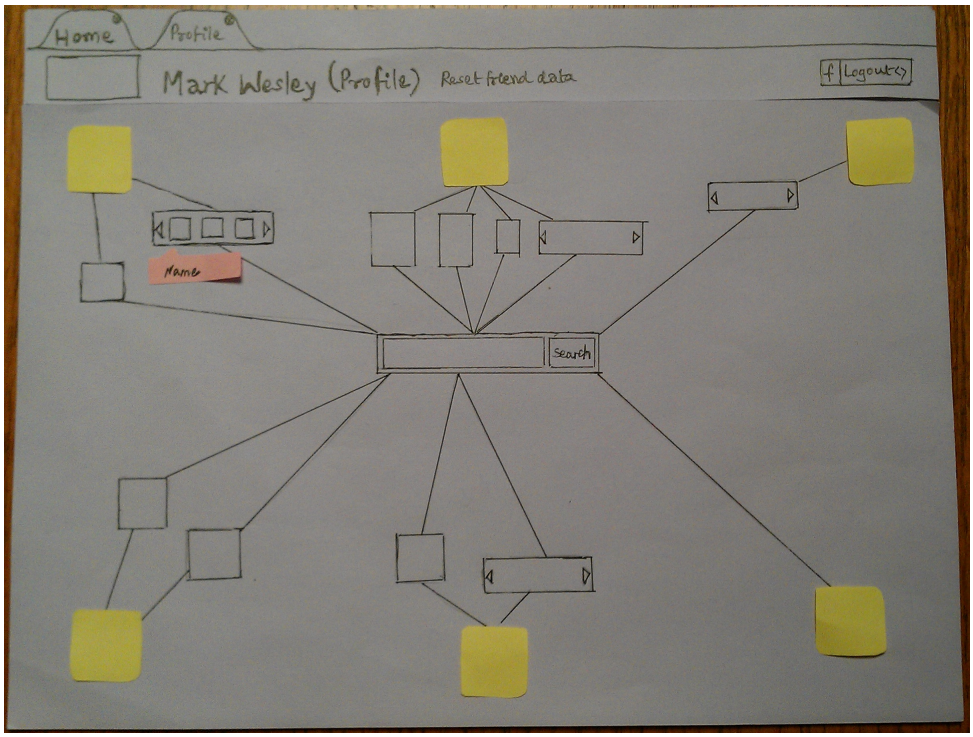
Upon receiving this feedback, we drew the following home screen to utilize in our final application.



Our initial design for the search results consisted of a simple graph and expanding forest of results.



Based on our own evaluations and user feedback, we determined it would be more user friendly to color code the results based on type and consolidate mutual friends into a list and show a maximum number of other connects as to not display too many.



Finally, we designed a simple tab based browser design to view search results and navigate to specific web pages. The initial design has a close button the Home tab but we decided we did not want that tab to be closed by the user. The browser tabs allow the user to go back, forward and refresh the webpage.