

searchbook

BY FUNNY BUNNIES

searchbook

BY FUNNY BUNNIES

searchbook is an Adobe AIR application created on Flash Builder 4.5 for Columbia's Computer Science User Interface Fall 2011 final project.

The member teams are

Ernesto Arreguin [eja2124]

&

Jon Bourdett [jab2279]

&

Chantal Galvez [cg2486]

&

Yiling Hu [yh2378]

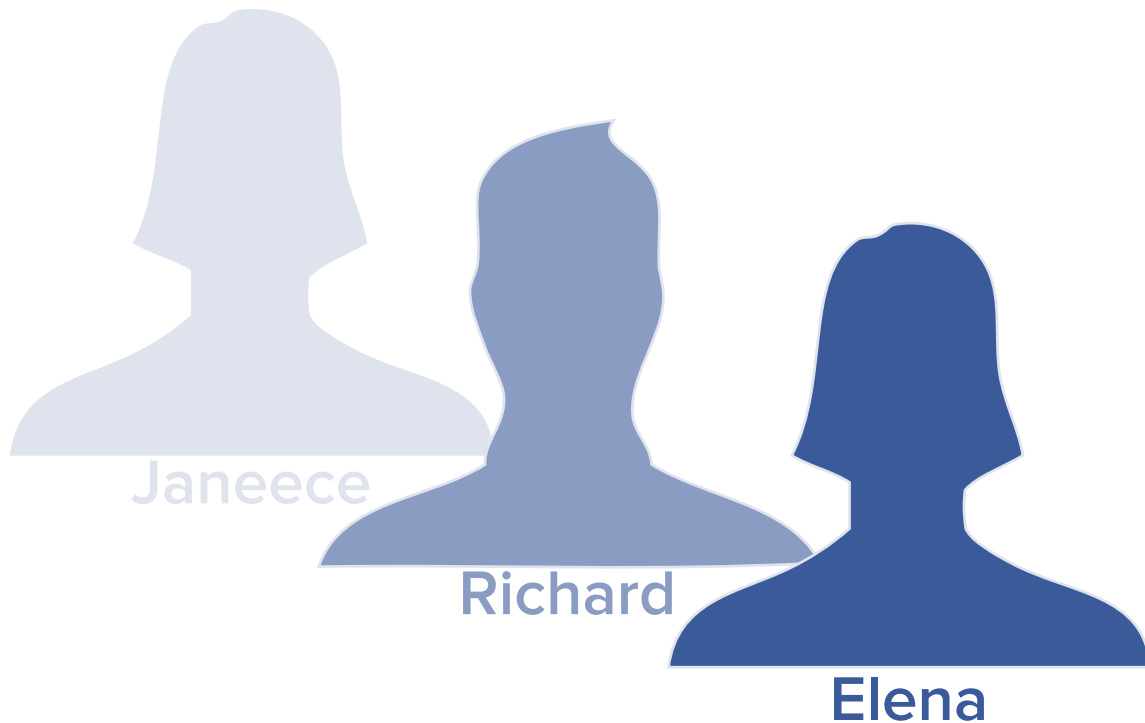
overall process

- GET FAMILIAR WITH GRAPH AND API
 - what can we do/ what can't we do?
- DECIDE ON HOW TO MAKE THE RESULTS' DISPLAY BETTER
 - Change the linearity of the result's display
 - Relevant results based on proximity to the user
 - Concentric circles
-
- DECIDE WHAT LIMITATIONS THE RESULTS ARE GOING TO HAVE
 - what are the users going to be able to search for?
 - what parts of the results are we going to be able to show?
 - what parts of the results are we going to show and not show?
- MAKE A PAPER PROTOTYPE
 - See how it is going to work
 - How we need to arrange the results and the wire frame for the whole application
- USER TESTING
 - with team members
- SET UP A VERSION CONTROL SYSTEM FOR WORKING ON THE PROJECT
 - Git @ <https://github.com/jonnyboy205/SearchBook>
- RESPONSIBILITIES OF EACH TEAM MEMBER
 - All design decisions were discussed by all members of the team.
 - While in the end all the team members were responsible for some part of all the deliverables, the initial distribution was:
 - Ernesto: Working Prototype, Search & API, Further results
 - Jon: Search results positioning, All search
 - Yiling: All search, tooltips
 - Chantal: Lo-fi Prototypes, Testing, Skinning, Documentation
- MAKE WORKING PROTOTYPE
 - Create a simple application that displays basic user information using the API
 - Build on simple application to create and display user searches decided upon
 - Add in-application information and external page links to search results
 - Arrange search result display as a visual representation of proximity

overall process

- MAKE DESIGN CHOICES FROM THE WORKING PROTOTYPE
 - General design choices
 - Design choices based on heuristics list
 - Design choices based on personas
- CODING & SKINNING
 - All coding and skinning process can be accessed through the commit history of the project in github
- USER MANUAL
- VIDEO

class of users



Our class of users consists of college students. College students tend to know how to use computers and spend a significant amount of time on Facebook. Thus it would be convenient to receive updates on things they like directly on Facebook. They are also aware of the search feature, and they know that doing a Facebook search will help them find results unique to Facebook. However, many of those users may not be sure about how to use the search filters or may not be happy with the “list” style of displaying results used by Facebook. Even if they tried to filter the results, they will most likely not know which filter to use, since the difference of pages, groups and events on Facebook is not very clear.

personas



Elena

Elena is a graphics art student about to finish her master's degree. She is 28 and originally from Slovakia. She lives with her boyfriend in an apartment in Manhattan and spends most of her time working on her art.

use scenario

Elena

Elena will be visiting her country soon. She has been very busy and has had no time to add her old friends to Facebook. She wants to quickly search for an old friend from her childhood with whom she hopes to spend some time during her visit. She opens Searchbook and after clicking the log in button and signing in she sees the different search choices offered by the application through radio buttons. She chooses the "people" option and enters her friend's name into the search bar. A series of concentric circles appear. Her face appears at the center of this circle and each circle groups a number of search pictures and results with each circle closer to her face showing the more relevant information. Elena immediately sees her friend in the circle closest to her. Unbeknown to Elena, Facebook has determined from the number of friends they have in common that this is one of the results she is likely looking for. She clicks on her friend's picture and an Internet browser opens displaying her friend's Facebook page where she can add her as a friend.

Delete Post

personas



Richard

Richard is an electrical engineering student at UCLA. He is 23 and trying to complete his bachelor's degree. He spends lots of time playing Call of Duty: Modern Warfare 3 and that has caused his grade to suffer a bit.

use scenario

Richard

Richard is looking for events where he can play Modern Warfare 3 with other people, as well as generally connect with other players to talk and discuss the game. He decides to use our Searchbook application. He logs in to the application using his Facebook credentials. Richard types in "Modern Warfare 3" into the search bar. He does not choose any specific category of search because he wants to see people and events, and possibly groups or pages. Our general search results page comes up, divided into four quadrants. Concentric circles surround the center of the screen, with more relevant results closer to the center. Richard sees that one of the events in the closest circle to the center is interesting(a tournament near LA, sponsored by GameStop). So he moves his mouse over to the event and hovers over it. This gives him some basic details about the event. He likes what he sees and decides he wants even further details. Richard clicks on the icon, which opens up Facebook on a browser, and takes you directly to that page associated with that event. After checking out that event for a few minutes, he realizes he wants to know which of his friends are interested in Modern Warfare 3. He goes back to the search results of "Modern Warfare 3" on the Searchbook application, and looks at all the people results displayed. He decides to check out the profiles of a few of the closest people to the center, because on hovering over these people, he notices they have several mutual friends. Eventually he friend requests some of the people.

[Delete Post](#)

personas



Janeece

Janeece is a music major. She plays flute. She is a 19 year old hipster who like bands and independent musicians that few people have ever heard of. She lives in Brooklyn and likes to come into Manhattan to try different coffee shops and meet new people.

use scenario

Janeece

Janeece just went to a new coffee shop at one of their weekly open mic nights. She heard a new musician named Buzz Armstrong who mentioned that he was "on Facebook" and she wants to follow his updates and information. However, he didn't mention if he had a Facebook Page, a public Profile, a specific Event for all his performances, or a Group for fans of him. She decides to go on Searchbook to find out more about him. She logs in with her Facebook credentials and enters in "Buzz Armstrong" in the search bar. She does not choose any specific search categories because she does not know which one is correct and it would be easier to show all the results at once. The results for People, Pages, Events, and Groups appear in the results circle as four different quadrants of equal size. The most relevant results (determined by Facebook's search) are closest to the center and less relevant results are farther away. Janeece sees that the correct result has a dedicated Facebook Page for his musician information. She decides to subscribe to the Page. She clicks on the result in the Pages quadrant which opens up her browser and takes her to the Facebook Page. After subscribing to the Page, she looks on the wall and sees another fan mention the Buzz Armstrong fan Group. Janeece goes back to the search results and sees that nothing in the initial Groups results are relevant. She then clicks on the arrow next to the outside ring of the Groups quadrant to show deeper results just in the Groups category. The circle transforms to only show Group results with each results' relevance reflected in its distance from the center with the most relevant results closes to the center. There were quite a few Groups with Buzz Armstrong for some reason so the correct group was a little distance away from the center. She then clicks on that result which takes her to the Facebook Group page where she requests to join the group.

Delete Post

design decisions

How the program works:

Facebook's design is very linear. The news feed, the new profile time lines and the search results are all order in a vertical linear mode.



Example of Facebook's Newsfeed, Timeline and Search results

The verticality of Facebook's design accommodates easily to the several sizes a browser can be support and translates better into smaller displays like mobile and tablets than another configuration might do.

Time lines and news feed fit particularly well with the linearity concept since they behave much like a stack- you add things to the front of the list, which makes them more visible. This reduces the visibility of the older items, but the nature of the feeds and the time line

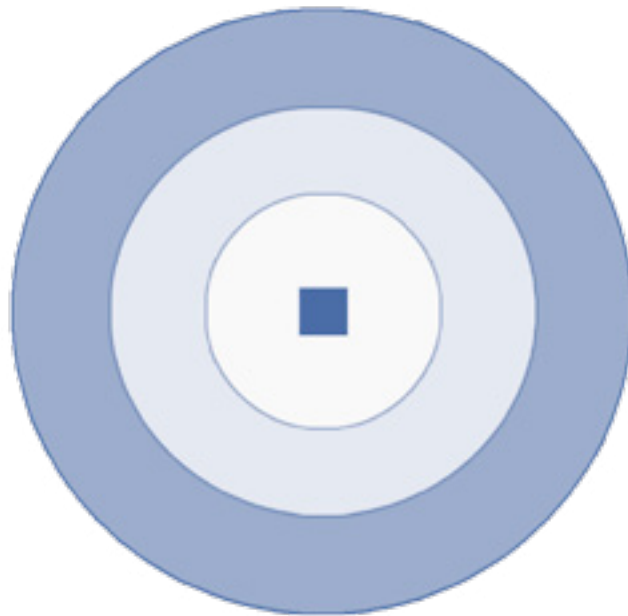
design decisions

inherently makes them less relevant as they become older.

This is not the same for results. When a user uses the search feature the results will be organized by relevance. While this proves useful for most searches, some users might want to have a more knowledge about the results that are given. Who does one result compare to its next immediate one? What is the distance between the last result and the first?

Taking from the API's name "Graph" we decided to explore the organization of the results. The first thing we were going to choose was some sort of graph, but since we had already studied the API, we knew most information for unfriended users was private, and the graph would not be useful as such.

The next choice was to use circles. Concentric circles with a user in the center would allow a clear representation of distance from the user to the result and a grouping of the results based on proximity (how the API feeds the results of a search to us).



Concentric circles used

One limit imposed by circles was that their representation takes much more plane space than a list, which can be scrolled in only one direction. If we were to display all results in several circles at a time would make the interface convoluted and hard to read, so we were going limit the results shown to only the amount that would fit the first set of circles. This, however we felt went against the idea of exploring search further, so we added more pages of results.

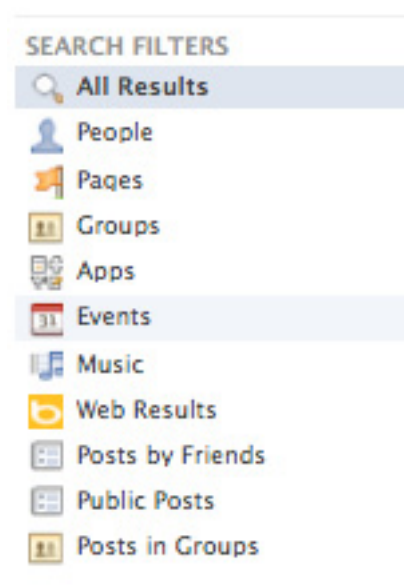
design decisions

Having different pages allows us:

- to have a standard size for the application layout
- reduce the number of results shown in a circle, to give each one more relevance.
- keep the layout clean
- explore the concept of adding/removing circles to show distance between users

Another limit imposed by the circles and the amount of space they provide, is that not all the identifiable information of each user can be presented upfront, as this would cause clutter.

Therefore, we selected what we, as frequent users of facebook and the case scenarios would find relevant. At this point we realized that facebook has many categories for search: people, pages, groups, apps, events, music, web results, posts by friends, public posts, posts in groups.

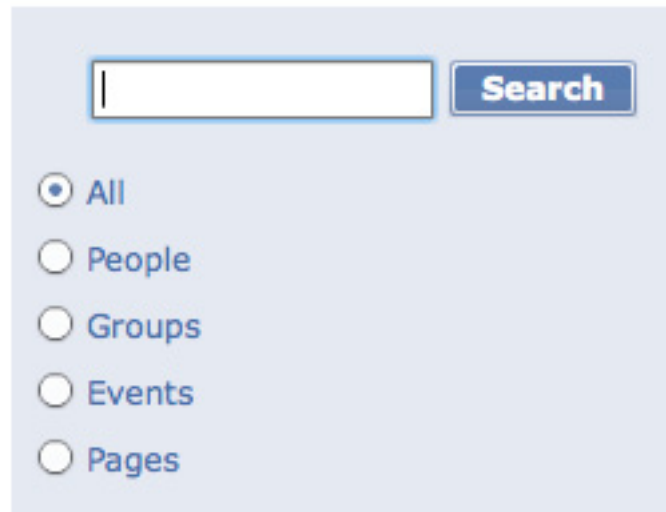


All types of search on facebook

Using the API documentation we found out that many of these categories weren't accessible or didn't retrieve much information.

So, we had to discriminate between the option what categories were the most relevant and what information on each category is relevant enough to be displayed and in what way. By means of an interview of friends of ours and, having in mind the case scenarios, we chose the following categories: People, Pages, Groups and Events, as well as an All Search that would include the most relevant results from each of these categories.

design decisions



Searches available on searchbook

Each of the results would have an image and a caption. The search retrieves a fixed sized image of 50x50 px. We realized the caption, could be of variable length, so we capped results at 22 characters. We placed the caption underneath the image to make better use of the space.



How each result is displayed.

For personas, the caption would be the name of a person.

For events, it would be the name of the event.

For groups it would be the name of the group.

For Pages it would be the name of the page.

While this is the most critical information, we saw it is not enough information for a result. We saw that facebook used tooltips to provide more information, and we decided to do the same.

The tooltips information for each type of results are:

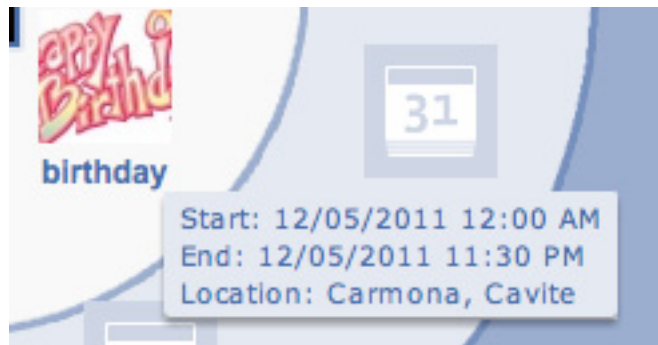
For personas: Full name, and network, if available (only allowed for friends)

design decisions

For groups: Description (which is also capped if it is significantly large)

For events: Start date and time, end date and time and location

For pages: Category and Bio



Event tooltip example

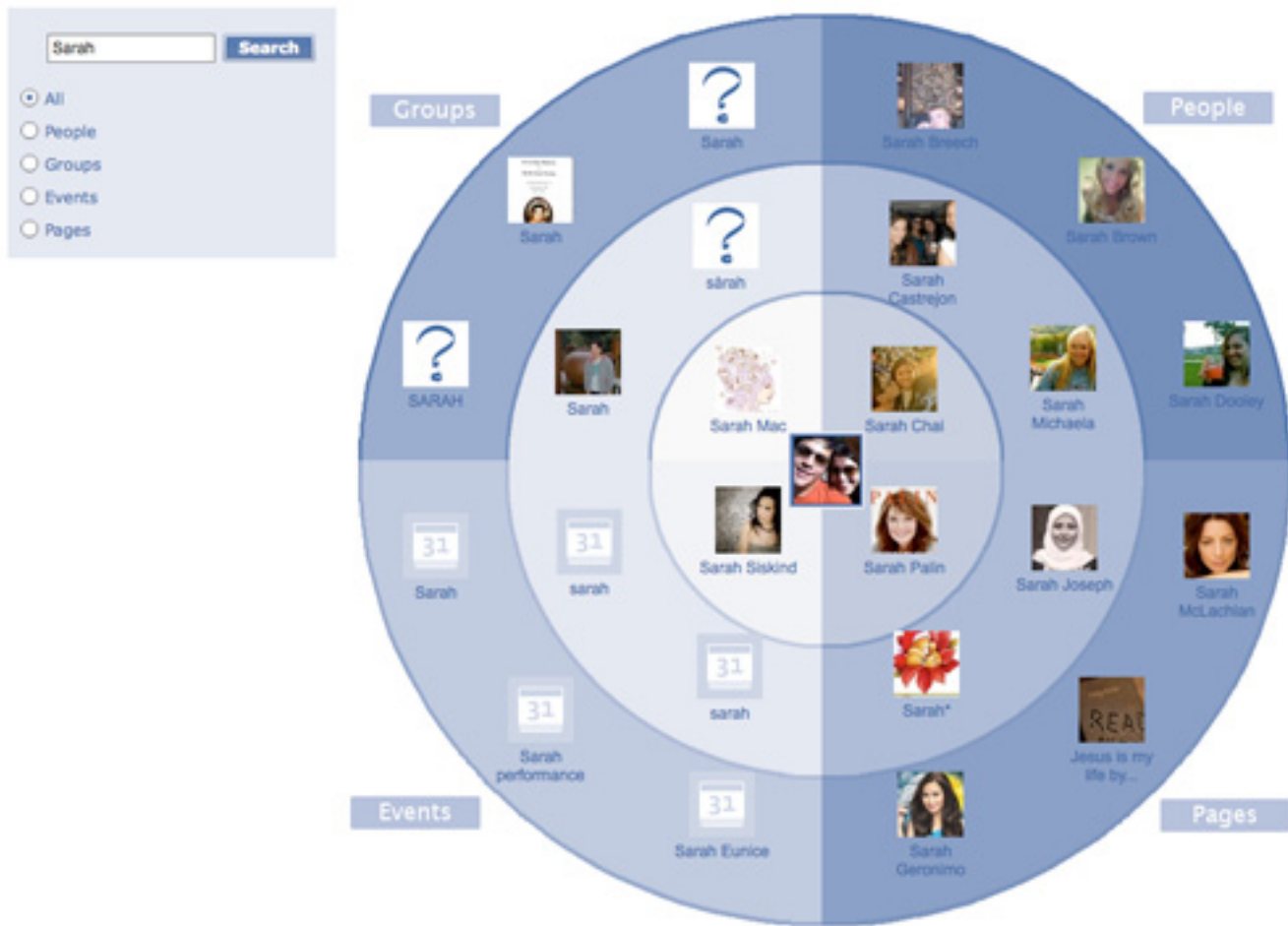


The program catches when any of these fields are missing and displays that the information is not available

To show the user which type of search they are performing and which other types are available, the search button, and search options are always available on the left hand side of the program.

If a user is uncertain about which type of search she/he wants to perform, they can do an All search that will show the most relevant results on all searches, separated by quadrant. From there the users can click on the specific type of search to get more results (note the labels). This is the way facebook search is implemented on their title bar. The advantage of our program is that right from the start, one can specify the category of the search they want to perform.

design decisions



All categories Search

The evolution of the layout configuration and the flow of the program can be found on the early prototypes.

Besides making the circles and the search query always visible the design was implemented to have the name, help and the logout option on a title bar, visible on all states.



Title bar

On the graphic side, the design borrows heavily on facebook's design. The main reason is that this is a product that is a complement of facebook, and therefore consistency helps the user navigate through the program faster. Facebook's Design team page makes a good case on why a redesign for a same product is useless and contradictory. As their design principles state,

design decisions

Clean

Our visual style is clean and understated, to create a blank canvas on which our users live. A minimal, well-lit space encourages participation and honest transparent communication. Clean is the not the easiest approach to visual style. To the contrary, margins and type scale, washes and color become more important as we reduce the number of styles we rely on.

Consistent

We invest our time wisely, by embracing patterns, recognizing that our usability is greatly improved when similar parts are expressed in similar ways. Our interactions speak to users with a single voice, building trust. Reduce, reuse, don't redesign.

design decisions

Design decisions based on heuristic list:

Visibility of system status

- Radial buttons and titles always show the user what search he/she is performing and which results are the ones visible.
- The user's profile picture is always visible in the center of the results, relating him/her to the search performed.
- When the user hovers over a result, the text label with the name gets bolded, to indicate the user is pointing to that particular result.
- When logged out, the user is shown a message that he/she must sign in in order to continue.

Match between system and the real world

- The language used for all the buttons and actions is familiar to the user of facebook, or to the user of any simple computer system.
- Words like "Next" or "Previous" are replaced with symbols (arrows) that complement the metaphor of adding/removing circles, making the program more comprehensible.

User control and freedom

- There is a logout button visible on the title bar on every page for the user to be able to logout at will.
- The arrows allow a user to go back and forth in a single category search without having to redo the search
- There is no need for undo/redo; the search is not going to change the data retrieved.

Consistency and standards

- Follow facebook platform conventions for naming the type of searches
- Layout similar to facebook
- the product is meant to be an addition to facebook, so we found it relevant to adhere the styles of facebook to our application.

Error prevention

- Radial buttons prevent uncategorized searches
- Not showing transition buttons between searches until they are done loading prevents the user from getting errors by sending another request while loading.
- When logged out, the search button/input and options are not visible to not allow users to perform searches.

design decisions

Recognition rather than recall

- The application shows the user's profile picture on the center of the concentric circles, always showing his/her proximity to the rest of the results.
- The user's name is always visible on the top right hand side.
- The search options are always available at the leftmost part of the application
- The search option maintains the search performed selected when showing the results, to remind the user which search they just performed.

Flexibility and efficiency of use

- Allow users to use the enter key to search instead of having to click search to not have to hover from keyboard to mouse.
- If a user is unsure which category their search belongs to, they can use all search, but direct category searching is available upfront (unlike facebook's search option)

Aesthetic and minimalist design

- The basic information for recognition of the result is displayed for the user not having to go through irrelevant information. when the user hovers over a result, extra information is shown on a tooltip.

Help users recognize, diagnose, and recover from errors

- The program has a default selection for "All" search to go to results on all categories available to not have a search without a category

Help and documentation

- The system is simple enough to be used without documentation. The labels on the results make the display clear; and the names on the action buttons and the search options.
- The User Manual is in a url that can be opened through the help link in all states of the program, it does assume that there is an Internet connection, but the program itself needs an internet connection to access facebook, so it is a fair assumption.
- All copyright and privacy issues (which are significant in facebook compared to other applications like flickr) are clearly stated in the User Manual.
- The user manual includes screenshots of the steps
- The user manual is divided into an about section and a "using searchbook" section, that goes through specific tasks of the application.
- If the user wants a more in depth knowledge of the app, the Development document (this) is available for download from the user manual.

prototyping & testing

Since we needed a fast turnaround, most of the prototyping was lo-fi. Before the prototyping, we had already settled on displaying the search results in concentric circles.

The first prototype was of the layout of the program [Figure 1.1]

We did a second prototype, [Figure 1.2], to do some testing, using a simple program flow of automatic login, and then just inputting a text for a search and getting the results.

Each of us was at least once an observer, a “computer” and the tester, and we all wrote feedback on the design anonymously on index cards. We did this so that when we discussed the results we would argue the idea itself, and not try to convince the owner of the idea personally. The most relevant points brought up in the testing were:

How do I know if I'm logged in or not? How can I log out?

How do I know which user I am logged in as?

How can I access help?

How can I go from search all to a specific category?

Why do I need to know the copyright?

How can I see more results? And go back and forth?

From these results we did a third prototype, which was a storyboard, showing how the flow of the program was going to work. [Figure 2] This is how the program was implemented.

The hi-fi prototype was a working model of some of the features we needed, like retrieval of search results, and placement of elements. The prototype was used as the foundation for the working program.

prototyping & testing

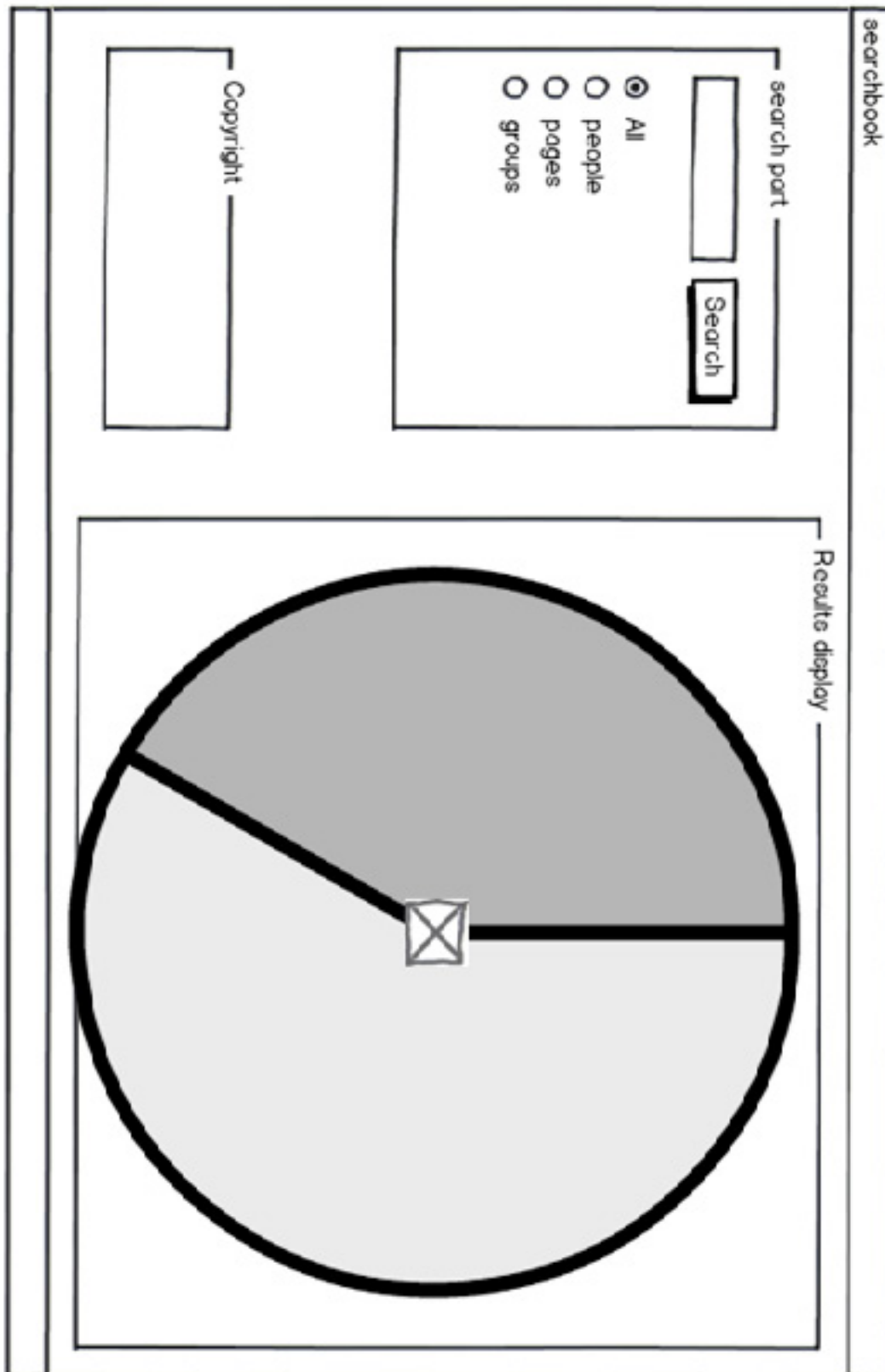


Figure 1.1: Layout Prototype

prototyping & testing

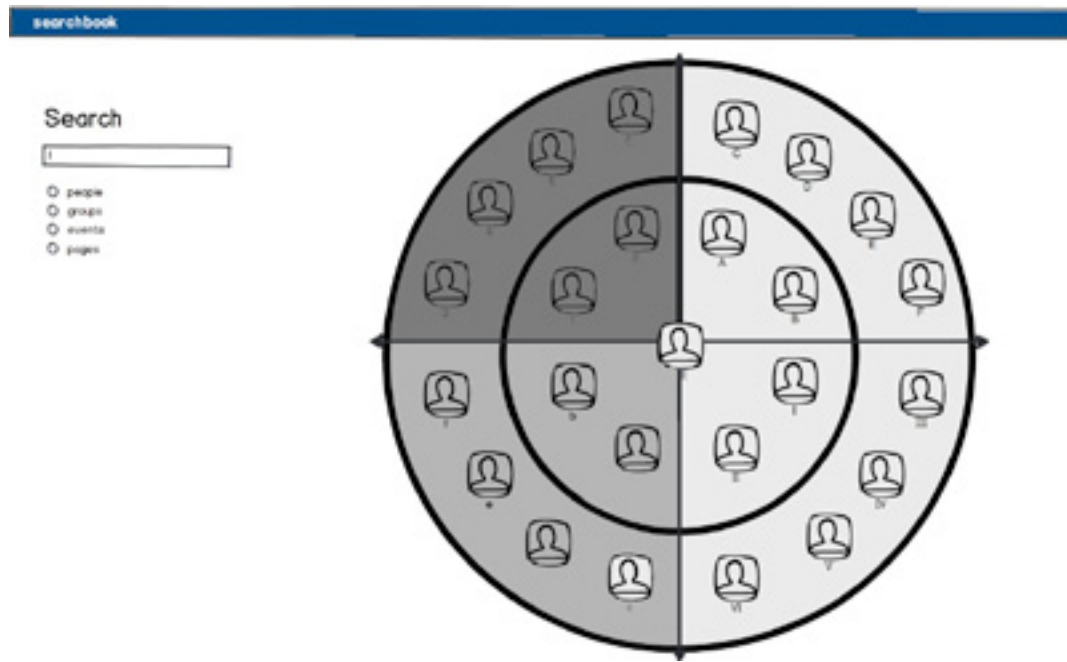


Figure 1.2: Paper Prototype used for testing

prototyping & testing

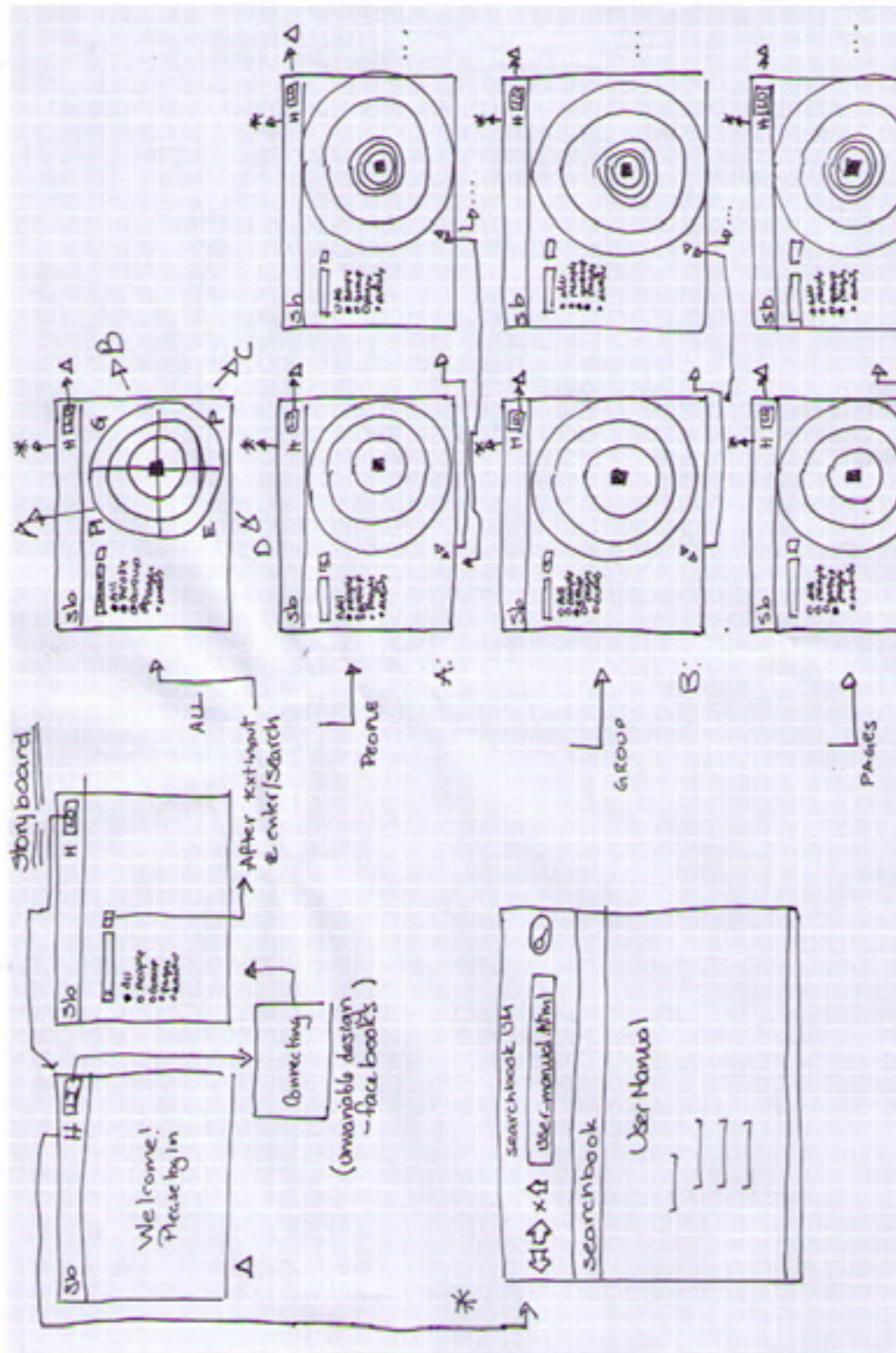


Figure 2: Storyboard

software engineering

This project was created for the AIR Desktop platform using Flex Builder 4.5. It is written in Action Script 3 and makes use of the Adobe ActionScript 3 SDK for Facebook Platform API which is a wrapper for the Facebook's Graph API.

For this project a combination of Spark components and MX components are used to obtain user search input and to display information on the screen. The Facebook Platform API log-in facilities are used at the beginning of the application to gather user log in information and permissions. These permissions include basic user information and friend location information.

Spark form elements such as radio buttons, text boxes and regular buttons are used throughout the program to specify search terms and types once a user has logged in. Once a user specifies a search term and a search type the Facebook API "api" function is called with the "search" parameter and the specified search term and types. Each search type returns an array of 0 to 25 results.

A class called InformationDisplay, which extends a skinnableContainer, has also been created to encapsulate the display of a picture/label pair which has a tool tip when highlighted and a link when clicked. To display search results a Spark group, which uses a basic layout, is employed to draw visual elements at discrete x, y coordinates.

InformationDisplay objects populated with the user search results are also added on to Display Group. The InformationDisplay objects are used to display picture/name pairs on the screen which are arranged around the circle graphics to approximate relevance of search results

software engineering

to the user. When the user makes an other search the Spark group container is cleared, the appropriate program state is called upon and the new information is displayed on the screen.

Finally, the Facebook Platform API facilities are also used to allow the user to log out when necessary.

references

User Manual

http://www.cs.columbia.edu/~cg2486/ui/searchbook/user_manual.html

Github Commit history

<https://github.com/jonnyboy205/SearchBook/commits/>

Facebook Design Page

https://www.facebook.com/note.php?note_id=118951047792

Facebook API

https://www.facebook.com/note.php?note_id=365915113919

<https://developers.facebook.com/docs/reference/fql/>

<http://www.adobe.com/devnet/facebook/samples.html>

<http://developers.facebook.com/docs/reference/api/>