

JSquaredA

# Image Search Plus

---

## Design Document

**Joonki Chung, Justin Berger, Anthony Litchfield**

**1/17/2013**

Image Search Plus is the Fast and Easy Way to Search for Images Online.

## Table of Contents

Table of Contents .....	1
Executive Summary .....	3
Project Description.....	3
Components .....	4
Applications .....	4
Background and Technical Requirements.....	5
Background .....	5
Technical Requirements .....	6
System Requirements – Client Side.....	6
Requirements Analysis.....	7
System Architecture .....	7
Personnel.....	7
System Features .....	8
Required.....	8
Planned.....	8
Optional.....	8
Software Engineering Tools and Techniques .....	9
Software Engineering Tools.....	9
Software Engineering Techniques .....	9
Timeline .....	10
Alpha (Weeks 1-5) .....	10
Beta (Weeks 6-10).....	10
Production (Weeks 11-15).....	11

Appendix A: Use Cases .....	12
Use Case 1 – Using the Site.....	12
Use Case 2 – Registration .....	12
Use Case 3 – Returning User .....	13
Use Case 4 – Edit Profile .....	13
Use Case 5 – Change Settings .....	13
Use Case 6 – Save/Load Sketch.....	14
Use Case 7 – Using Advanced Tools .....	14
Appendix B: UI Sketches.....	15
UI 1 – Sketch Pane.....	15
UI 2 – Search Results.....	16
UI 3 – Selected Result.....	17
UI 4 – Registration.....	18
UI 5 – Login.....	19
UI 6 – Settings/Profile Menu .....	20
UI 7 – Change Profile .....	21
UI 8 – Change Settings .....	22
UI 9 – Save Sketch.....	23
UI 10 – Load Sketch .....	24

## Executive Summary

### Project Description

The purpose of this project is to help users quickly and intuitively search for images on the Internet. This project will exist as a website similar to a conventional image search engine. However, instead of providing keywords to search by, as is done by conventional search engines, users will provide a simple outline sketch. The image results for a user's query will be images which are most "similar" to their drawing.

Using Image Search Plus, users will be able to access and use this image search over the Internet, using Mozilla Firefox, Microsoft Internet Explorer, or Google Chrome, from PCs or mobile devices. Users will create sketches using their standard input devices such as their mouse, track-pen, or touch-pad.

Users will then click search to receive image results from their queries which correspond to "similar" images. If a user sketches the outline of the Rocky Mountains, images containing similar outlines will be returned. This would return images of the Rocky Mountains, but might also include images with a similar outline. Users can also register their email address with the project which will allow users to save their past sketches.

The image database that will be queried by this project will consist of links to images freely available on the Internet and image descriptors. The search results will need to be processed very quickly to minimize user wait-time. To achieve this, the image database will be populated by a pre-processing service that will create a 'descriptor' for the image. This descriptor will consist of the location of the image, the source page(s), and the gradient tensor field for the image.

We are storing only the descriptors because they will take significantly less storage space than the full images. This also avoids many copyright issues that stem from storing the images on our own servers. The number of images in Google's image search database was around 8 billion in 2005. This means we will either need to make Image Search Plus scalable to billions of images to be as powerful, or we need to have it pull only the most useful images. We will do this by storing only relevant images. An image is relevant to Image Search Plus if it has edges that can be detected by our algorithm and, contains at least 16 pixels per gradient tensor cell.

The image database will be populated by a web crawler. This crawler will be configured to start at a given page and look for <img> tags on the page. It will determine if an image is relevant to Image Search Plus. If it is relevant, it will be added to the list of images to pre-process. Any links on the page will be added to the list of pages to crawl.

The image database will need to be regularly updated by the crawler. For example, when images or web pages are removed, users will not be presented with dead links. This means the crawler will have to routinely check existing images / existing pages to determine if an image still exists.

## Components

- Custom web crawler – Will search pages for images to be added to the pre-processing queue.
- Image pre-processing – Will take images from the web crawler and generate the gradient tensor field for the image, which will be stored in the database. Images with the same gradient tensor field will be stored as the same image, with multiple links. Also determines if images are relevant.
- Image database – Will store the source URL(s) and the gradient tensor field for the image. The database will be able to return results quickly.
- The web interface and website design
  - Login / Registration Pages – Lets users log in or sign in the website.
  - Custom sketch pad interface will be built specifically for our application – Allows the user to draw sketches to find the image that the user want to get.
  - Search Results Page – Shows image results searched by the sketch from custom sketch pad.
- User database and sketch storage – Stores basic profile information and previously saved sketches for later retrieval. Might store some usage statistics as well.

## Applications

Sketch Retrieval and Reverse Image Look-Up have many real life applications most of which target businesses and people that deal with numerous images every day:

- Locating CAD designs in large collections.
- Locating a specific past design or piece of art done by graphic or technical artists.
- Forensic artists often make sketch drawings of suspects as described by a witness. One application then could be to easily search for similar looking people in police image databases.
- More easily locating images for presentations, blogs, web pages, and more, where the primary concerns is not “What keywords are associated with this image?” but rather, “What does the image look like? Is it visually what I want?”

## Background and Technical Requirements

### Background

Image search has been a key issue since the earliest days of computer stored digital image collections. The number of images stored in digital formats has increased and continues to increase at an ever higher rate. Approaches toward image search have been many and varied. An early approach was to give each image a unique filename and search for that filename. This approach requires the image archivist to associate keywords or information in a textual form at the time of saving the image. Unfortunately, if the image is hard to describe in short keywords or the archivist chose unusual keywords, a user may not be able to locate that image.

In the early 1990s, IBM developed a commercially available system called QBIC (short for “Query By Image Content”). Their system was expensive and multi-featured. It supported keyword search. It also supported “search by example”, meaning the user could make a drawing using a tool similar to Microsoft Paint (but much more limited) and search for images in the database with similar colors and shapes. For more information about QBIC, see this link: <http://www.cis.temple.edu/~vasilis/Courses/CIS750/Papers/qbic.pdf>

In 1995, the paper, “Fast Multiresolution Image Querying”, presented a way to do “search by example” relying on Harr Wavelet Decomposition of images as the similarity measure. Research demonstrated success on a database containing 10,000 images. This approach is used in the open source Python image library: ImgSeek. The paper is available here: <http://salesin.cs.washington.edu/abstracts.html>

A dissertation published in 2001 demonstrates that much progress has been made in the field of search by example. The writer discusses the problem in general terms and then reports his findings. He developed a system that makes use of 3 different shape-similarity measures which were the more popular approaches of the time. Those measures were Difference Chain Codes, Fourier Descriptors, and Co-Occurrence Matrices. The paper is available here: <http://mmis.doc.ic.ac.uk/pr-d.heesch-2001/sketch-retrieval.pdf>

This project will base its similarity measure on the approach used in a paper published in 2009, which was presented at EuroGraphics – Symposium on Sketch-Based Interfaces and Modeling. The similarity measure is based on Gradient Tensors. Their approach was adapted to sketch retrieval, meaning a user draws a simple outline sketch as the input without including colors. Their approach gave favorable results when searching through a database of over a million images collected from the Internet. The paper is available here: <http://perso.telecom-paristech.fr/~boubek/papers/SBIR/sbir.pdf>

At the present time, there are many image search solutions available which offer different features and capabilities. Most major search engines provide image search by keyword, for example: Google, MSN/Bing, Yahoo, and Alta Vista.

TinEye offers reverse image search. Users may visit the TinEye website and upload an image. TinEye will scan through its database of images taken from the web and informs the users where that image can be found on the Internet, even if the images are of a different size or resolution. Their approach is based on a very strict similarity measure, and color is a necessary factor.

Google offers a reverse image search capability. From the standard image search box, users may upload an image from their computer instead of choosing a keyword. Google then returns similar images based on color, shape, and possibly keyword. It likely makes use of color-histogram similarity as well as a machine-learning keyword tagging engine that Google has developed to assign likely keywords to new images that it finds.

Creating sketches online also has a background spanning many years. At present, this is a common application that is used by many companies, ranging from online signature pads for banks to cloud-based drawing systems. An example of a signature pad can be found here:

<http://thomasjbradley.ca/lab/signature-pad/accept/>

### Technical Requirements

We will utilize a Microsoft IIS webserver with a Microsoft SQL Server database to drive most of the website, including user profiles, saving sketches, loading sketches, and user settings.

The sketch pane will utilize existing examples using HTML5, and Javascript. Examples include <http://mudcu.be/sketchpad/> and <http://sketchpad.io/sketch/?newfile>.

The image search will determine sketch/image similarity based on Gradient Tensors that should produce good results. This algorithm requires us to create our own image search database. However we plan to use tools such as NVidia CUDA to make the pre-processing and searching as fast as possible.

The crawler will be based off of a crawler Justin created for a previous project. The previous crawler was designed to analyze statistics about how many of various tags were on a page (such as image tags). It did not obey crawler standards such as robots.txt or html meta noindex tags. It will need efficiency improvements, as well as integration with the pre-processing algorithm.

### System Requirements – Client Side

This project assumes users will have the following minimum system requirements:

- Internet-enabled desktop or laptop computer, or a mobile device (phone or tablet).
- A web browser which supports HTML 5 and Canvas, such as Chrome, Firefox, or Internet Explorer.
- An input device for drawing simple sketches, such as a mouse, a touchpad, a track pen, a TrackPoint®, etc.

### System Requirements – Server Side

Image Search Plus might require more powerful server hardware to scale to millions or billions of images, but should work adequately for our purposes on our currently available hardware.

## Requirements Analysis

### System Architecture

- The Image Pre-Processing
  - Off-line algorithm for performing Gradient Tensor decomposition.
  - Ability to process several million images (Multi-threaded or GPU-accelerated)
  - Exact Duplicate detection (perhaps with MD5 Message-Digest Algorithm)
- The Website and the Sketchpad
  - Easy to navigate, easy and intuitive to use
  - Fast access to image database for fast response times per query
  - Two website versions, one for desktop browsers and one for smaller screens such as mobile phones or tablets. The one for smaller screens will use larger icons and have less 'wasted' space.
  - Web Language (C# with ASP.Net preferred)
- The Web Crawler
  - C# based system for crawling the web
  - Visited pages will be tracked so that the same page does not get revisited
  - Web requests will be performed in parallel to offset connection latency
- Image Search Database
  - C# based for simplicity
  - Uses Gradient Tensor image/sketch similarity search algorithm
  - Image/Sketch similarity search algorithm will be easy to replace, update, or otherwise modify
- Registered User Database
  - MS SQL Database
    - Standard ASPNET Membership Provider
    - Also stores saved vector sketches

### Personnel

Joonki will be responsible for the Website and Sketch pane.

Justin will be responsible for the Crawler and User Database. He has created a simple crawler that can identify tags on a page, and has several years of professional experience working with databases.

Anthony will be responsible for the Pre-Processing algorithm, Image Database, Sketch/Image Similarity, and Search algorithm.



## System Features

### Required

- Sketch (Pencil only)
- Image Database / Pre-processing
- Search
- Search Results
- Simple Crawler (images on 'known' websites)

### Planned

- Crawler
- Sketch Eraser
- Sketch Clear
- Register / Login
- Paged Results
- Full-Screen results with details (link-back, page content, etc)
- Main Menu / Profile Menu
- Improved Pre-Processing - More efficient
- Improved Search - More efficient / More 'accurate' / Layered

### Optional

- Profiles
- Crawler can 'update' image database
- Save / Load Sketches
- Improve crawler efficiency and allow for multiple source types
  - Image Database Websites
  - Google Image Search Results
- Additional Filters for results AFTER search
  - Like Google image search and others (pre-dominant color, size, 'type')
- Option to choose different search methods or algorithms
- Sketch Color
- Sketch Line Width
- Pre-Processing/Search that handles color
- Pre-Processing/Search that handles line width

- Turn image into sketch, search off of this new sketch
- Sharing

## Software Engineering Tools and Techniques

### Software Engineering Tools

- SVN – Code Versioning
- Google Docs – Edit Documents Collaboratively
- Google Calendar – Schedule meetings
- Asana – Task Management

### Software Engineering Techniques

We will be using an agile approach for this project. We would like to have a working version ready to display at the end of every 5 week period, with improvements and additional features added during the first three to four weeks with testing during the remaining weeks.

Our initial version will contain the bare minimum necessary to have a working project, and we will add more features over time. We also plan to do significant testing every 5 week period.

- Build initial working prototype of major functionality
  - Unit Testing
  - Search Accuracy/Performance Analysis
- Build planned features to extend prototype
  - GUI usability / functionality testing
- Add optional features as time allows
  - Usability / Performance Testing

We will be using SVN for versioning of code, and Google Docs for documentation versioning and collaboration. Additionally we plan to use Asana for bug tracking and task management.

We will each be responsible for documentation over components we developed, with Justin pulling the final documentation into a unified document.

We are going to use Google Calendar to schedule meetings, most of which will be in-person. Any virtual meetings will be via Skype. Much of our communication will be via email and Asana, though we have the other members' phone numbers if we need a more immediate response. We will meet at least once a week on a day that fits the majority of our schedules, and plan to meet on the days class is held for the other section only.

## Timeline

### Alpha (Weeks 1-5)

	Joongi	Justin	Anthony
<b>Week 1</b>	Work on web-site framework. Placeholders for anything not built yet.	Design crawler objects. Create templates and interfaces where relevant.	Pre-processing stubs, image database stubs.
<b>Week 2</b>	Build basic website and basic sketch-pad that allows drawing sketches.	Code basic crawler, capable of queuing up links to new pages, and looking for img tags on a page.	Pre-processing algorithm, initial image database.
<b>Week 3</b>	Build webpages to do basic search and show results.	Code crawler to populate pre-processing algorithm.	Framework for testing
<b>Week 4</b>	Test website for bugs. Improve the design and visual effects if time allows.	Test crawler for bugs. Fix bugs. If time allows, add image size filters.	Test / Improve.
<b>Week 5</b>	Prepare demo of website.	Prepare Developer documentation and demo of crawler. Need a sample site to crawl with choice images.	Prepare Demo and documentation of image processing.

### Beta (Weeks 6-10)

	Joongi	Justin	Anthony
<b>Week 6</b>	Add Eraser/Clear functionality to sketch pad.	Fix any remaining bugs in crawler, and add any features that were desired during testing.	Improve Search Accuracy
<b>Week 7</b>	Build webpage that allows users to edit their profile. Create Main Menu.	Create the User Database for storing profiles, login information, and saved sketches. Begin coding interfaces for user database.	Improve Search Efficiency with hierarchical image database and optimized user sketch processing.
<b>Week 8</b>	Build webpage to show full-screen image results.	Create Register and Login pages.	Paged Results in DB
<b>Week 9</b>	Test website to find bugs.	Test user database, registration, and login.	Testing.
<b>Week 10</b>	Prepare demo of website. Add more visual effects if time allows.	Prepare Developer Documentation for user database, and demonstration of Login and Registration pages.	Prepare Demo and documentation of efficient image processing.

**Production (Weeks 11-15)**

	<b>Joonki</b>	<b>Justin</b>	<b>Anthony</b>
<b>Week 11</b>	Add Sketch color/line width Functionality to sketch pad.	Develop pages to save and load sketches.	Compare Search Algorithms and parameters.
<b>Week 12</b>	Add additional filters for search results.	Develop pages to allow sharing sketches.	Additional Search Filters (color, dimensions, file type)
<b>Week 13</b>	Testing of the entire website.	Testing of all components, and usability testing.	Image into Sketch into Search
<b>Week 14</b>	Bug Fixes and try to improve visual effects.	Fix bugs and continue Testing.	Testing
<b>Week 15</b>	Prepare Demo	Prepare final demo.	Prepare Demo

## Appendix A: Use Cases

### Use Case 1 – Using the Site

Upon visiting the website, users are presented with the sketch pad.

1. Visit website in browser
2. Begin drawing using the ‘pencil’ tool. This tool allows users to draw in the sketch pane as if they were using a pencil.
  - a. When the mouse moves over the sketch pane it will display a pencil cursor
  - b. Hold mouse button and move mouse within the sketch pane to draw curves/lines/shapes
  - c. Sketch pane will translate mouse movements into line segments drawn on the pane
  - d. Release mouse button to stop
  - e. If the “pencil” leaves the sketch pane while the mouse is still held down, it will end the current line segment where the mouse left the pane.
  - f. If the “pencil” re-enters the sketch pane while the mouse is still held down, it will begin a new line segment where the mouse entered the pane.
3. When the user clicks on the ‘search’ tool, they will be directed to the search results page, where they will see several possible matches for their sketch.
  - a. The user will be able to navigate through different pages of results without having to completely re-run the search algorithm.
4. When the user selects a result, it will bring them to a larger view of that image with some details about where the image was retrieved. From this view, they can close this view, navigate forward or backward through results, and click on a link to visit the site where the image was found (in a new tab).

### Use Case 2 – Registration

Users may register with the website, allowing them to save and load sketches.

1. Visit the website in browser
2. Click on “Sign In”
3. Click the “Sign Up” button
4. Fill in the registration form
  - a. Username
  - b. Email Address
  - c. Password and Confirmation
  - d. Name and Birthday
  - e. Profile Image

5. Once registration is complete, users will be logged in and redirected back to the sketch page without losing the sketch they were working on.
6. This will allow users to access settings, save sketches, and load sketches

### **Use Case 3 – Returning User**

Returning users can login using credentials given during registration.

1. Visit the website in browser
2. Click on “Sign-In”
3. Type in your username and password
4. Click the “Sign-In” button
5. Users will be logged in and redirected to the sketch page without losing the sketch they were working on.
6. This will allow users to access settings, save sketches, and load sketches.

### **Use Case 4 – Edit Profile**

Users who are logged in can edit their profile from the profile menu.

1. Assume the user is logged into the site
2. Click Profile image in the top right corner, this brings up the main menu.
3. Click Edit Profile, this brings the user to the Edit Profile page
  - a. Users may upload a profile image by clicking browse to select an image
  - b. Users may change their email address by typing in a new email address
  - c. Users may change their password by typing a new password
  - d. Users may change their name or birthday by typing in a new name or birthday.
4. Clicking save will commit any changes that were chosen

### **Use Case 5 – Change Settings**

Users who are logged in can edit various settings from the settings menu.

1. Assume the user is logged into the site
2. Click Profile image in the top right corner, this brings up the main menu.
3. Click “Settings”, this brings the user to the Settings page
  - a. Users may decide how many images they want to see per page by clicking the number in the slider.
  - b. Users may decide whether the result page opens in a new window or not by clicking the checkbox.
4. Clicking save will commit any changes that were made

### **Use Case 6 – Save/Load Sketch**

Users who are logged in can save and reload sketches.

1. Assume the user is logged into the site
2. Assume the user has a sketch (drawn before or after logging in)
3. Click “Save” which brings up a dialog to name the sketch
  - a. User may choose to type in a name
4. Click “Save” in the dialog to save the sketch.

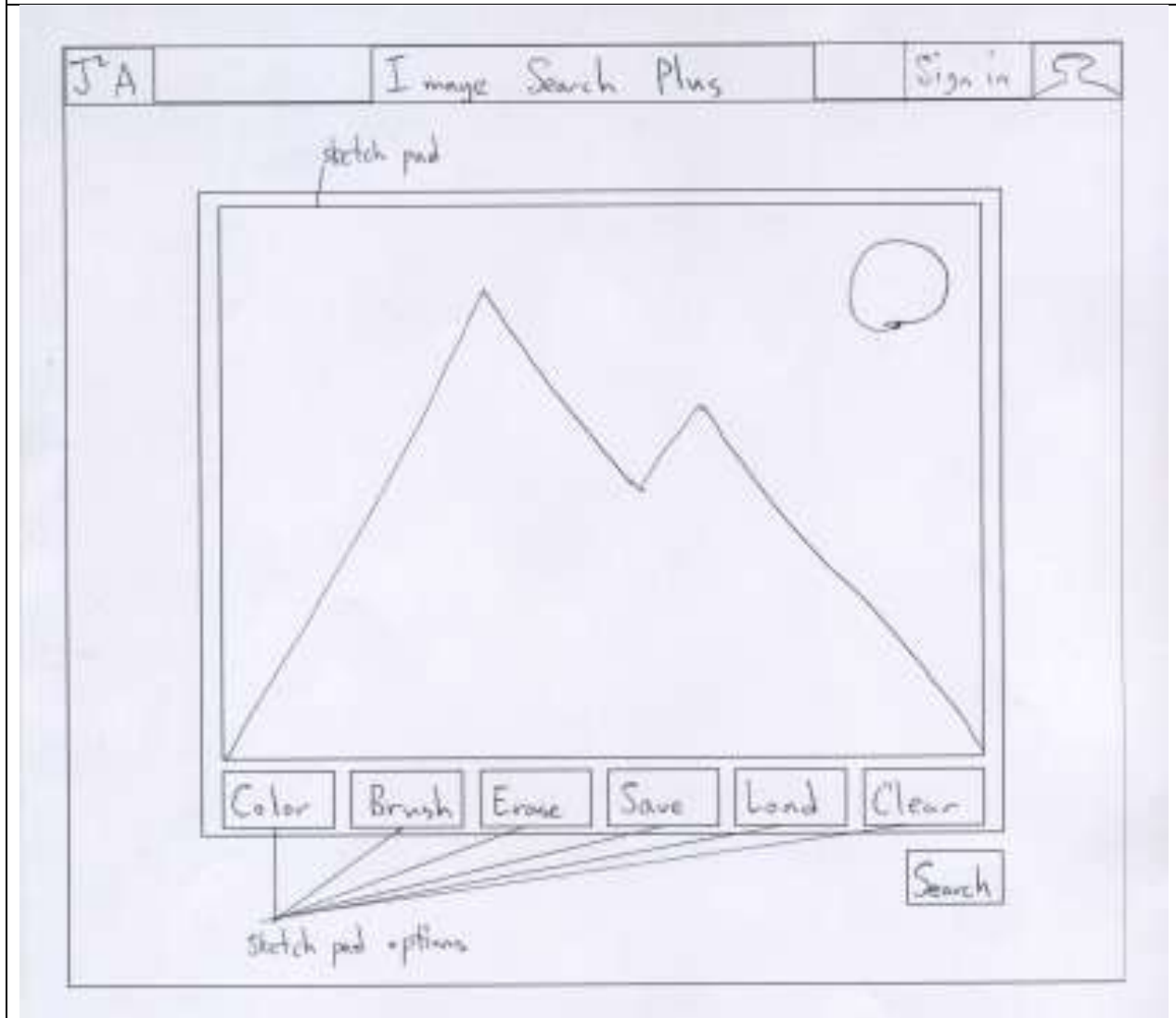
### **Use Case 7 – Using Advanced Tools**

Users can use various advanced tools to perfect their sketches.

1. Assume user has visited the site, and is working on a sketch
2. Several tools besides the pencil are available to create sketches
  - a. Eraser
    - i. If possible, changes the icon to an eraser while in the sketch pane
    - ii. User holds mouse button down inside the sketch pane to begin erasing.
    - iii. If the cursor crosses over a line segment it will be removed from the sketch.
  - b. Color
    - i. Drop down box allows user to choose a color
    - ii. Changes the color of future line segments drawn with the pencil
  - c. Line width (Brush size)
    - i. Drop down box allows user to choose a line-width
    - ii. Changes the width of future line segments drawn with the pencil
  - d. Clear
    - i. Clear the sketch pad
  - e. Undo
    - i. Undo the most recent line. Activated with Ctrl+z

## Appendix B: UI Sketches

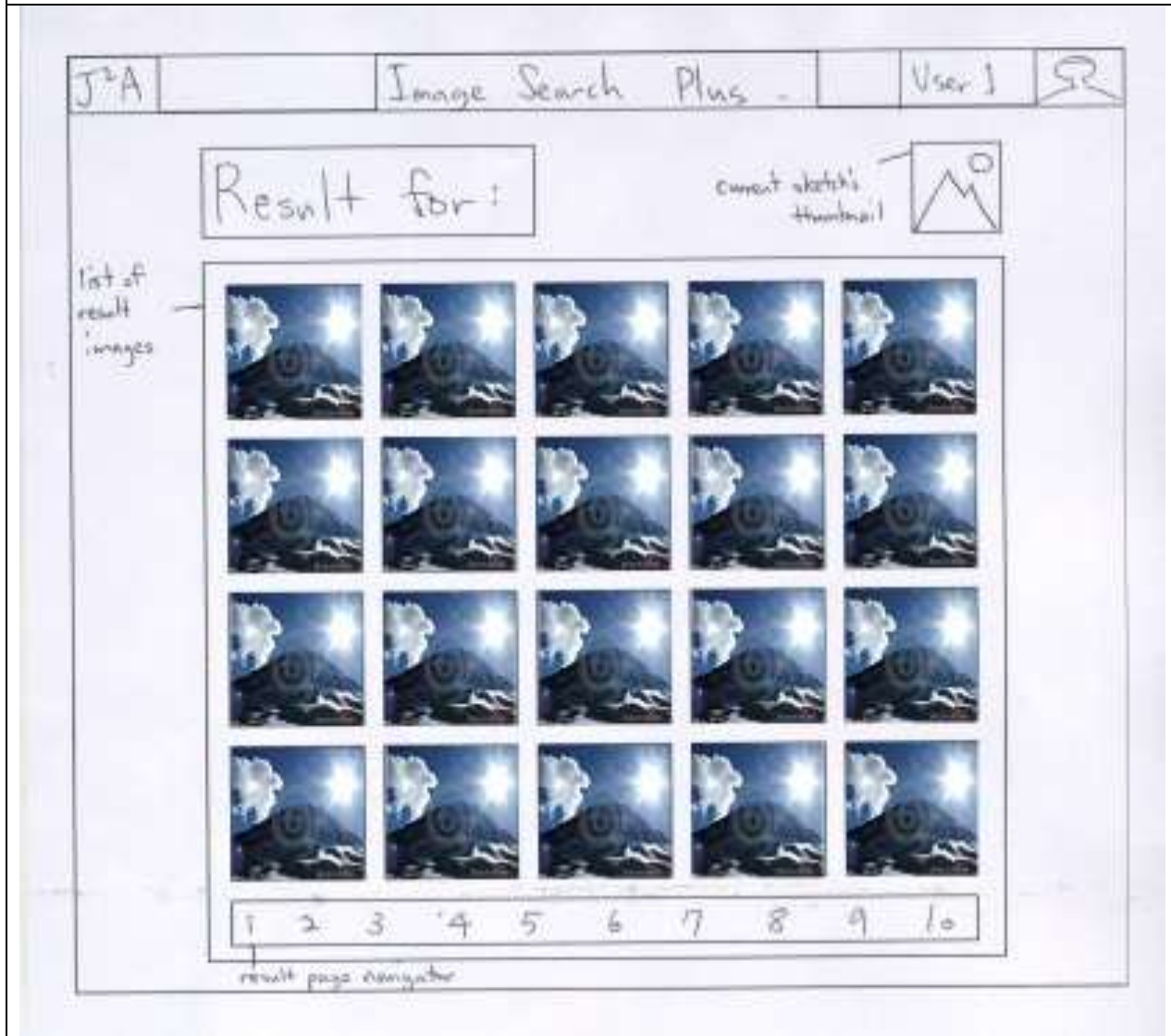
### UI 1 - Sketch Pane



*There are various options for user to perfect or save/load their sketches.*

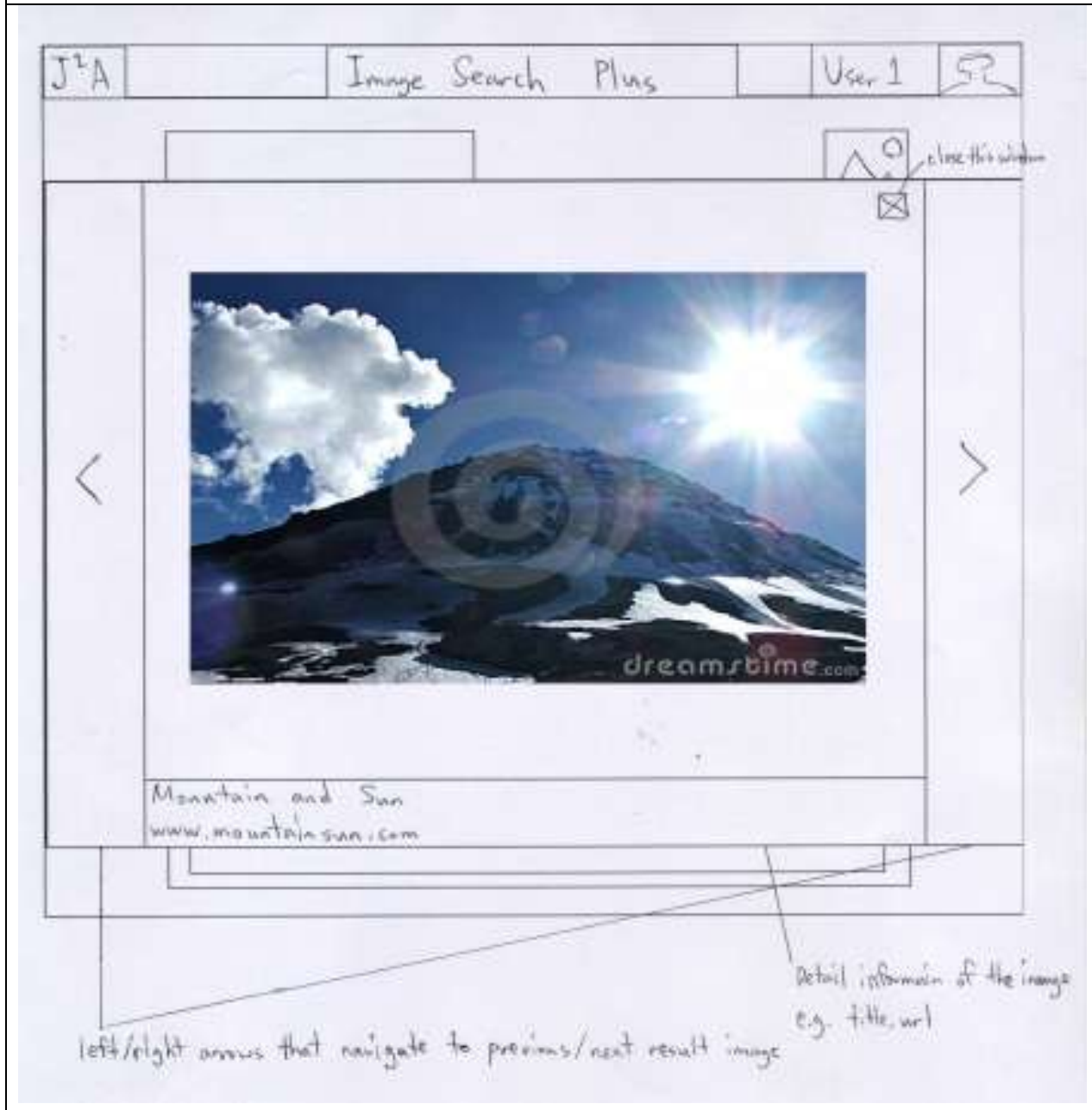


## UI 2 - Search Results



*User can go back to sketch pane page by clicking the sketch's thumbnail.*

## UI 3 - Selected Result



*User can save the image by right-clicking mouse. (Browser will handle saving)*

**UI 4 - Registration**

JA Image Search Plus Sign in

Create a new account

Username

Create a password

Confirm your password

Name

Email address

Birth day  Month  Day  Year

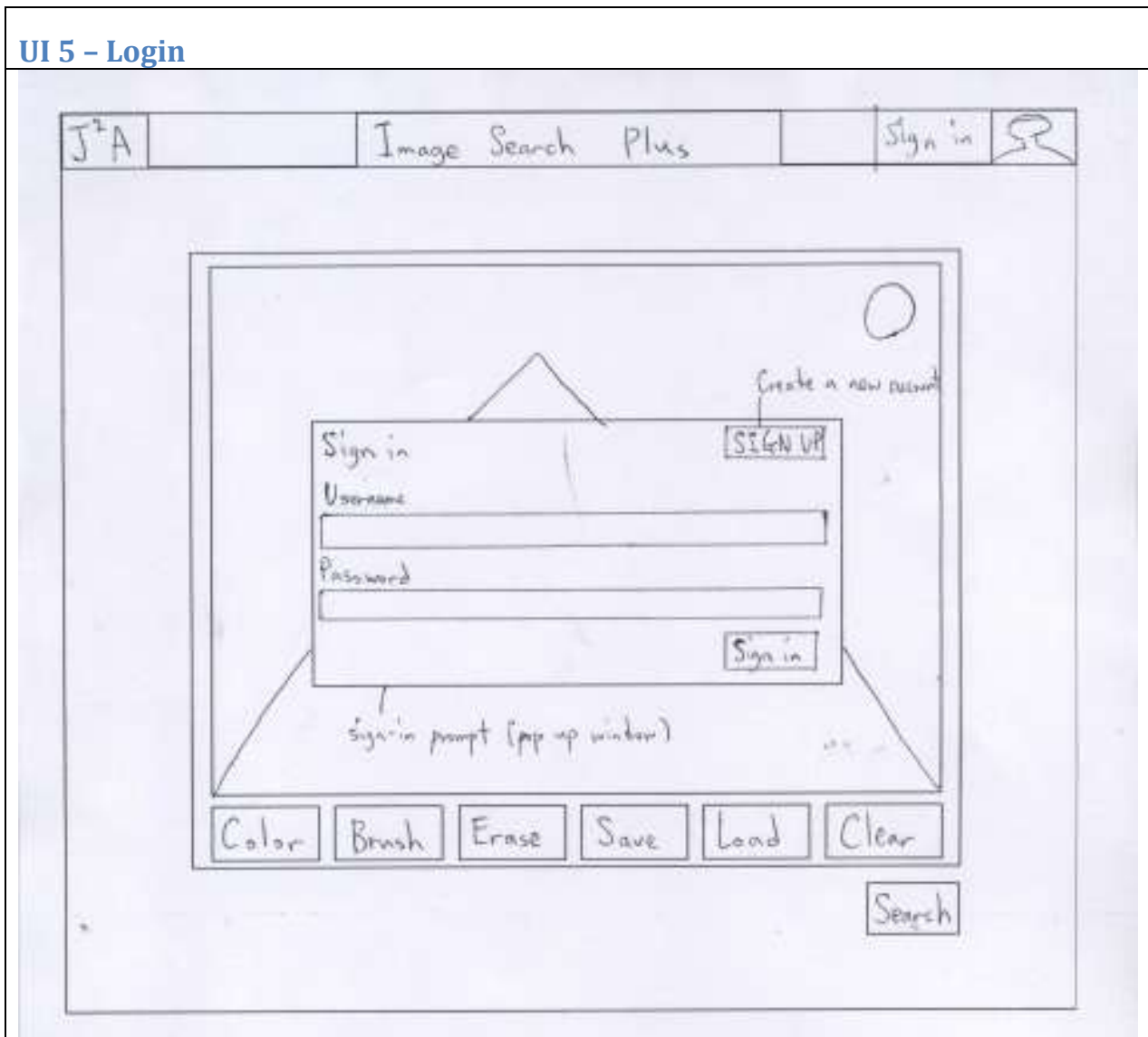
Profile image

User can insert the profile image by clicking “insert” button.

“insert” button will generate a file input dialog to select the image from user’s directory.

If the user is not logged in, Sign In button appears.

Otherwise, Sign In button is replaced by the label that shows the user’s name.

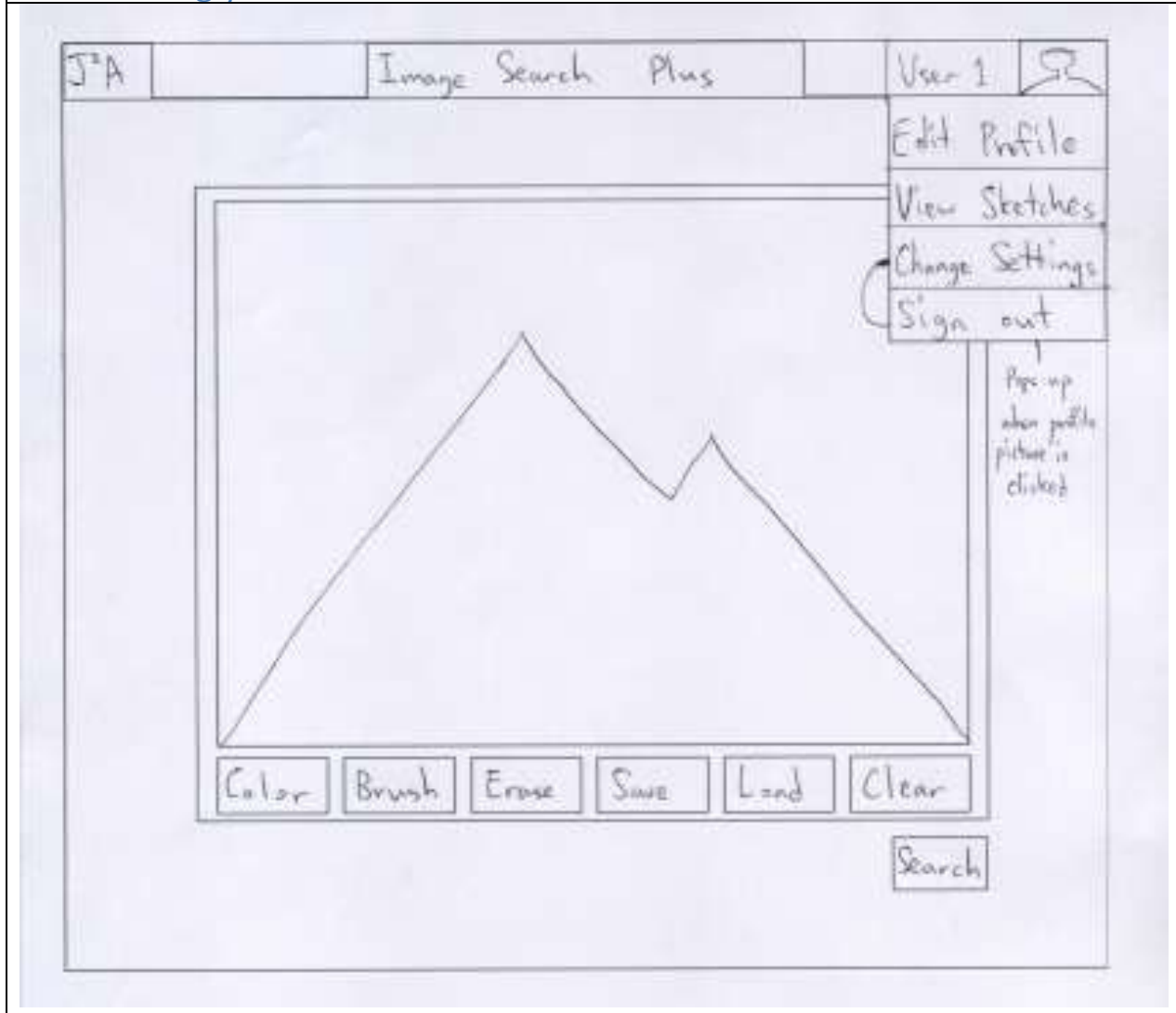
**UI 5 - Login**

*User can sign in by typing username and password and clicking Sign in button.*

*If the user doesn't have an account, user can click SIGN UP button to sign up.*

*If the user is not logged in, Sign In button appears.*

*Otherwise, Sign In button is replaced by the label that shows the user's name.*

**UI 6 - Settings/Profile Menu**


*Edit Profile lets the user modify his/her profile.*

*View Sketches lets the user see the list of saved sketches.*

*Change Settings lets the user change settings. (See UI 8 for details).*

*Sign Out lets the user sign out.*

**UI 7 - Change Profile**

J <sup>2</sup> A	Image Search Plus	User 1	
------------------	-------------------	--------	-------------------------------------------------------------------------------------


Edit Profile  
(Leave blank if you do not want to change)

Change password:

Change Name:


Change Email address:

Change Birthday:

Change Profile Image: 

*User fills in only the fields, text and/or picture, that the user wants modified.*

**UI 8 - Change Settings**

J <sup>2</sup> A	Image Search Plus	User 1	
------------------	-------------------	--------	-------------------------------------------------------------------------------------

Change Settings

- Results per page

10 15 20 25 30 35 40 45 50  
Faster slower

- Where results open

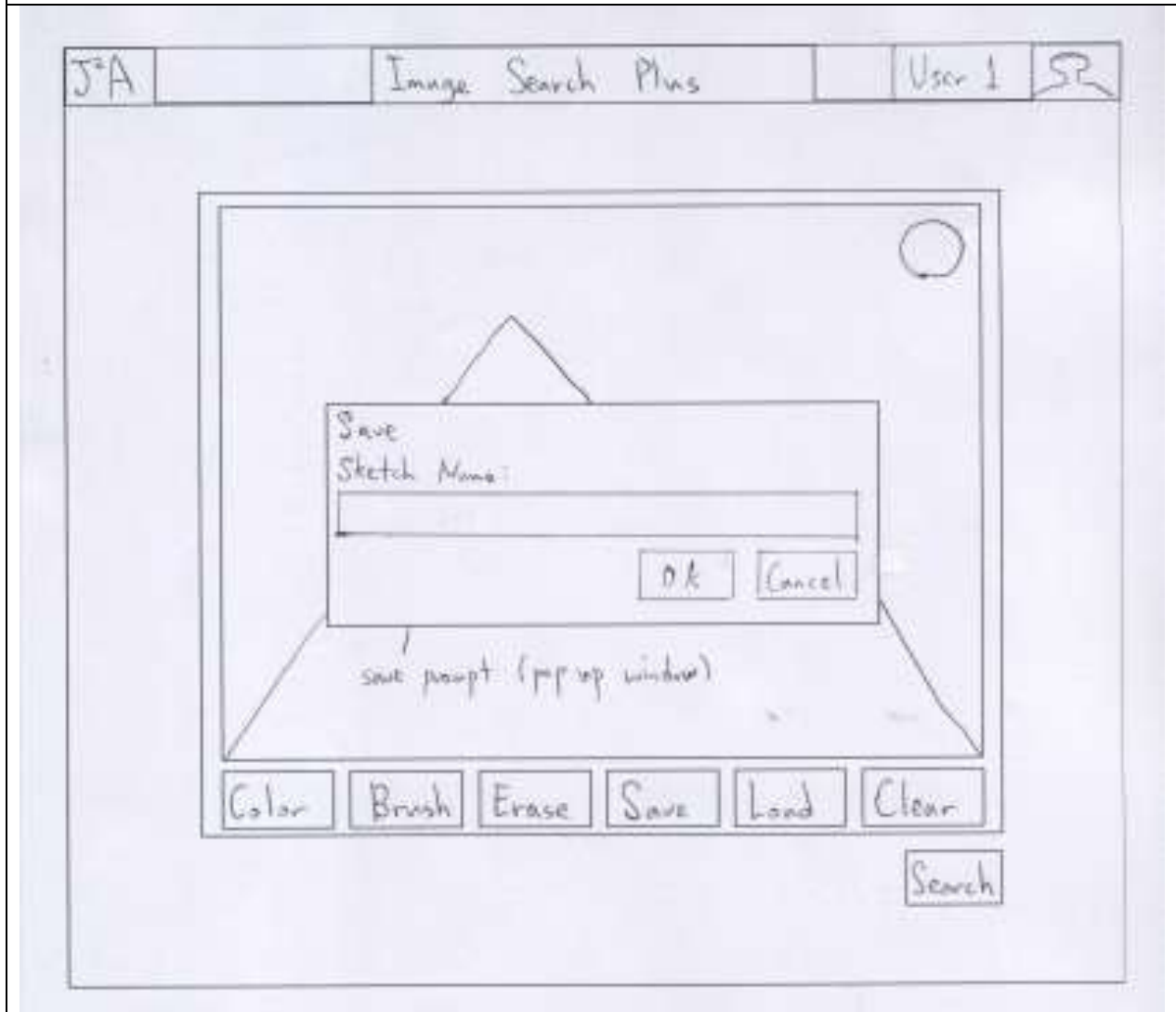
☒ Open in a new window

Save Cancel

*The more results to be shown, the longer it will take to show them.*

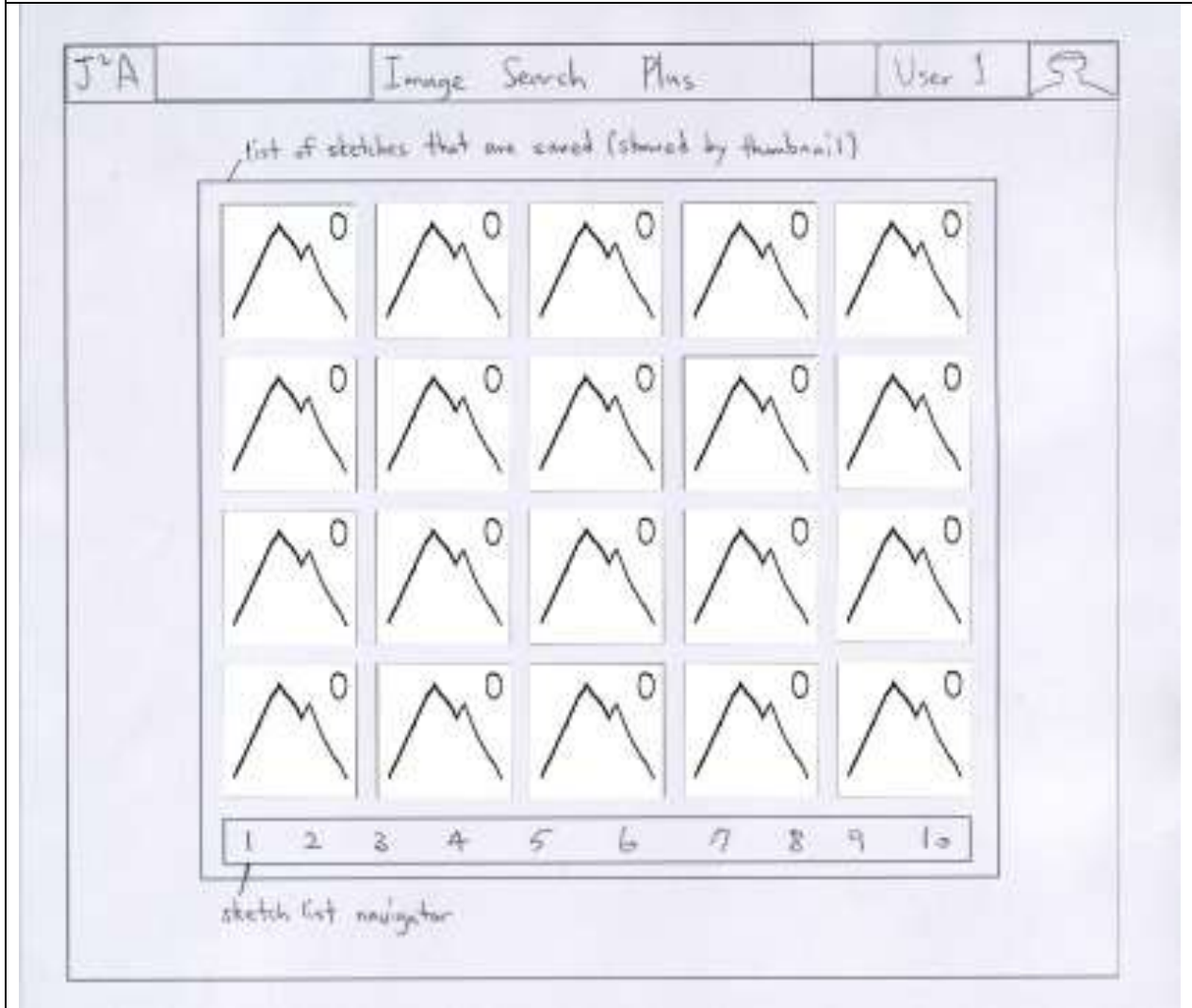


## UI 9 - Save Sketch



*Save prompt appears when the user clicks "Save" button.*



**UI 10 - Load Sketch**

*Each saved sketch is shown in thumbnail view.*

*There is a page navigator to navigate the sketch list.*