

Final Project for SW Engineering Class

CSC 648-848 Summer 2018

Hat Box Photos
8/8/2018

Team 5

Amelia Rawlings (Amelia.Rawlings@gmail.com)

Front-End

Arnold Jiadong Yu

Dalu Li

Betty Reaney

Back-End

Keawa Rozet

U Keong Cheong (Ray Cheong)

Albert Du

Demo URL: <https://hatboxphotos.com/>

Product Summary

Welcome, everyone, to the open launch of HatBoxPhotos. We are delighted to offer you the opportunity to be one of the first to see our brand new free stock photo website. Features to look forward to include:

1. **Fast Search**

Need a high quality image for school, work, or just because? You can quickly find the perfect image for your needs at HatBoxPhotos.

2. **Uploading images**

Share your special moments with other users through our simple image uploading process!

3. **Admin moderation**

HatBoxPhotos enforces a family-friendly image policy. Our admin will either approve or reject uploaded images for live viewing within 24 hours.

Please visit us at <https://hatboxphotos.com/>. We look forward to your feedback.

SW Engineering CSC648/848 Summer 2018

Hat Box Photos

Team 5

Amelia Rawlings (Amelia.Rawlings@gmail.com)

Front-End

Arnold Jiadong Yu

Dalu Li

Betty Reaney

Back-End

Keawa Rozet

UKeongCheong (Ray)

Albert Du

Milestone 1

6/26/2018

Revision Table		
6/26/2018	1.0	Initial document

Executive Summary

We are looking to develop a website to quickly search, upload, and download photos for free. Online photos have become increasingly popular, coinciding with the growth of the photography industry. We figure people need methods and platforms in order to either share their work, store their work, or simply share photos of random parts of their lives. Such websites are an amazing and vastly underrated resource. Our website can serve as either a personal site or business site. Registered users with an account on our website will have full access to upload and download a variety of images. This site will also have a search function. We plan to develop a very clean software platform that will enable users to perform these functions.

Our site aims to serve a variety of people, anyone who needs to quickly find an image and photographers. This website will especially be perfect for photographers looking for images for inspiration. In addition, they will be able to store their portfolios by taking advantage of our image upload feature. We believe our website will be a great success.

Personas and Use cases

Mary is a college student.

- Very busy with college study.
- Has good computer and mobile skills.
- Not too much patience for searching needed material online.
- Worried about whether she can legally use the images she finds online.
- Basic research skill.

Mary's goal is to use images for her assignments.



Mary is a **college student**. She received a writing assignment about the city of San Francisco and has to use three landmark photos in her paper. When she gets home, she opens the home page of HatBoxPhotos. She searches the landmarks of San Francisco. Once she decides which landmark photos to use, she selects them for download, and is then prompted to register for a new account or log-in before she can download them to her computer. Then she proceeds to log out.

Personas and Use cases (cont.)

Henry is a photographer.

- Good photo shooting skill.
- Has good computer and mobile skills.
- Likes sharing lots of his photos online for free.
- Likes helping others.

Henry's goal is to find a good stock website so that people can use more of his photos.



Henry is a **photographer**. He visited San Francisco and took lots of photos in the city. Henry returns home and opens the home page of HatBoxPhotos. He logs in and chooses the photos that he wants to upload. Then, he writes a description about each photo and uploads them one by one for admin approval. He then proceeds to log out. Henry logs into his account the next day and gets a notification about whether his photos were approved.

Personas and Use cases (cont.)

Jennifer is the admin of the website.

- Basic management skill.
- Good Computer and mobile skills.
- Has basic experience in managing websites.
- Hard worker and fast learner.
- Good research skills.

Jennifer's goal is to manage the website well and help the company grow.



Jennifer is an **admin**. When she starts work, she opens the home page of HatBoxPhotos and logs into her account. Then, she navigates to the photo approval section. She reviews each photo and its description then decides whether to approve or reject the photo. She views the template email for approving/declining a photo and decides to add an additional section to further explain her decision. She finishes typing and submits her decision, sending an email to the picture's uploader.

Additional Use cases

Lucas is an aspiring **photographer**. He is planning on visiting the Golden Gate Bridge and wants to take some well-structured photos. For inspiration, Lucas navigates to hatboxphotos.com and searches for pictures of the Golden Gate Bridge in the search bar. Lucas finds a picture he wants and is planning to use it as a reference for when he goes himself, so he signs in and downloads the picture. After the picture finishes downloading, he logs off hatboxphotos.com.

John is an **admin**. He navigates to hatboxphotos.com to start looking for photos to approve and notices an inappropriate photo that should not have been approved for upload on the home page. John immediately logs in to his admin account and uses the search bar to find that exact photo he saw. He clicks on the photo and clicks an admin-only button called "Remove". John is taken to another page to edit the template email for removing a photo and decides to add a quick paragraph why it is an inappropriate photo. He clicks the "confirm" button and the photo is removed and the email is sent.

Data Definitions

Account: User account with username and password as credentials.

Admin: Registered user with the privilege to approve or reject uploaded images after review

- Admins shall log in to their account on hatboxphotos.com before reviewing images
- Admins shall review the content and description of all uploaded images before they go live on hatboxphotos.com
- Admins shall notify users in a message about why an uploaded image was not approved to go live
- Admins possess non-admin user privileges in addition to their admin privileges

Image: Full resolution image.

- **Description**: Text description of an image.
 - Image descriptions shall be limited to a certain length
 - Image descriptions shall be optional
- **Name**: Name of an uploaded image on hatboxphotos.com
- **Tag**: Category of an image (i.e. clothing, furniture, landmark, outdoor).
 - Image tags shall facilitate image search performed by users.
- **Thumbnail**: Preview of an image.
 - Thumbnails shall be displayed in search results
 - Thumbnails shall be displayed in place of full resolution images to users who are not logged in to their account
- **Upload Date**: Date an image was uploaded to hatboxphotos.com
- Registered users shall be able to upload or download images after logging in to their account.

Image Database: Database containing information about all images on hatboxphotos.com, such as image name, description, tags, thumbnail, and upload date.

Non-admin user: Registered user with the basic privileges of uploading and downloading images (i.e. students, photographers, etc.)

Registered User: User with an account on hatboxphotos.com

- Registered users shall be separated into admins and non-admin users
- Registered users shall log in to their account before uploading or downloading images
- Users shall provide their age, username, and password during the registration process
- Users shall have the option to provide their bio, occupation, and student status during the registration process

Data Definitions (cont.)

Unregistered user: User without an account on hatboxphotos.com

- Unregistered users shall only be able to view image thumbnails in place of full resolution images
- Unregistered users shall not be able to upload or download images

User: Anyone who visits hatboxphotos.com

- Users shall be separated into registered users and unregistered users
- Users shall be able to search for images
- Users shall be able to view image thumbnails

Functional Requirements

Unregistered Users:

- 1.1. Unregistered users shall be able to view photos on the home page
- 1.2. Unregistered users shall be able to sign up with an email and password
- 1.3. Unregistered users shall be able to use the search bar to search for specific photos
- 1.4. Unregistered users shall be able to view thumbnail image in place of full resolution images
- 1.5. Unregistered users shall be able to click on a thumbnail image and view it in medium resolution as well as its description
- 1.6. Unregistered users shall be prompted to sign up/in before downloading/uploading a photo

Registered Users:

- 2.1. Registered users shall have the same functionality as an unregistered user
- 2.2. Registered users shall be able to login with their email and password
- 2.3. Registered users shall be able to download an image in thumbnail, medium Resolution, or high resolution
- 2.4. Registered users shall be able to upload an image
- 2.5. Registered users shall be updated if their uploaded image was approved/declined
- 2.6. Registered users shall be able to view their profile
- 2.7. Registered users shall be able to view all their uploaded photos

Admin:

- 3.1. Admin users shall have the same functionality as a registered user
- 3.2. Admin users shall review the content and description of all uploaded images
- 3.3. Admin users shall click approve if photo reviewed is acceptable
- 3.4. Admin users shall click decline if photo reviewed is not acceptable
- 3.5. Admin users shall have the ability to edit the template approve/decline email that is sent to the registered user
- 3.6. Admin users shall be able to remove any previously approved photo

Photos:

- 4.1. Photos shall have the ability to be described by tags
- 4.2. Photos shall have a description
- 4.3. Photos shall show the date it was posted
- 4.4. Photos shall show the Registered user who posted it
- 4.5. Photos shall be viewed on the home page
- 4.6. Photos shall be viewed determined by search results
- 4.7. Photos shall be viewed in thumbnail and medium resolution for unregistered users.

Misc:

- 5.1. Search bar shall have fuzzy search functionality

Non-Functional Requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in Mo (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
3. Data shall be stored in the team's chosen database technology on the team's deployment server.
4. No more than 50 concurrent users shall be accessing the application at any time
5. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
6. The language used shall be English.
7. Application shall be very easy to use and intuitive.
8. Application shall render well on mobile devices (UI shall be responsive)
9. Google analytics shall be added
10. No e-mail clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2018. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

Competitive Analysis

Features:	Shutterstock	Alamy	YAY Images	hatboxphotos
Photo Library	BEST	BEST	GOOD	GOOD
Search & Filters, Ease of Use	BEST	GOOD	OK	GOOD
User Submissions Accepted	YES	YES	NO	YES
Pricing	HIGH	MEDIUM	LOW	FREE
No Account Required for Downloads	NO	NO	NO	YES

Stock photo sites are used for many purposes: from adding optimized, fast loading images to websites; to supplying enterprise advertising campaigns with customized, stylish photos and artwork. Many stock photo sites offer a large library of photos with many filters and search functions to help people find the perfect images for their purpose. Unfortunately, people will usually need to purchase a license in order to use those photos, and the license could be expensive. Moreover, people may need to create an account and pay for the images they want before they can download and use them, which could be annoying. However, at hatboxphotos, we are a completely free platform for people to get and share images. We not only offer a good number of photos with an easy to use interface, most importantly, there is no account required for downloading images. Helping people find and obtain their images in the easiest and fastest way, that's what sets hatboxphotos apart from the competition.

High-level system architecture

- Host: Amazon Web Services 1vCPU 1 GB RAM
- Operating System: Ubuntu v16.04 Server
- Database: MongoDB v2.6.10
- Web Server: NGINX v1.10.3
- Server-Side Language: JavaScript v8.11.3
- Web Framework: Reactstrap, Express
- IDE: NetBeans v8.2
- Web Analytics: Google Analytics

Team 5 Members

- Albert Du
- Amelia Rawlings
- Arnold Jiadong Yu
- Betty Reaney
- Dalu Li
- Keawa Rozet
- U Keong Cheong (Ray)

Checklist

External Meetings: DONE (Meetings set-up as needed)

GitHub Master: ON TRACK

SW tools and Server: DONE

Team Read M1 Submission: ON TRACK

SW Engineering CSC648/848 Summer 2018

Hat Box Photos

Team 5

Amelia Rawlings (Amelia.Rawlings@gmail.com)

Front-End

Arnold Jiadong Yu

Dalu Li

Betty Reaney

Back-End

Keawa Rozet

U Keong Cheong (Ray)

Albert Du

Milestone 2

Revision Table		
7/6/2018	1.0	Initial document
8/7/2018	2.0	Revised document

Data Definitions

Account: User account with email and password as credentials.

Admin: Registered user with the privilege to approve or reject uploaded images after review. This is done before images become live.

- Admins possess non-admin user privileges in addition to their admin privileges

Image: Full resolution image.

- **Category**: Category of an image (i.e. clothing, furniture, landmark, outdoor).
 - Image categories shall facilitate image search performed by users.
 - Images shall only have one category
- **Downloads**: The number of times an image is downloaded from hatboxphotos.com
- **File size**: Size of the image file
 - File size shall be limited to 16 MB
- **Image description**: Text description of an image.
 - Image descriptions shall be limited to a certain length
 - Image descriptions shall be optional
- **Image height**: Height of an image measured in pixels
- **Image file format**: Format of an image file
 - File formats shall be limited to jpeg, png, and gif.
- **Image name**: Name of an uploaded image on hatboxphotos.com
- **Image width**: Width of an image measured in pixels
- **Resolution**: The level of detail in an image
 - Image resolution shall be measured by image width and image height
 - Full image resolution shall be the image's original width and height
 - Medium image resolution shall be 256x256
 - Thumbnail resolution shall be 128x128
- **Status**: The approval status of an uploaded image.
 - Image status can be approved, pending, or rejected
- **Tag**: Keywords extracted from the image description using a keyword extractor
- **Thumbnail**: Preview of an image
 - Thumbnails shall be displayed in search results
 - Thumbnails shall be displayed in place of full resolution images to users who are not logged in to their account
- **Upload date**: Date an image was uploaded to hatboxphotos.com
- **Uploaded by**: Name of the registered user who uploaded an image to hatboxphotos.com

Data Definitions (cont.)

Image: (cont.)

- **Views**: The number of views an image receives from all users on hatboxphotos.com

Image database: Database containing information about all images on hatboxphotos.com, such as image name, description, tags, thumbnail, and upload date.

Non-admin user: Registered user with the basic privileges of uploading and downloading images (i.e. students, photographers, etc.)

Registered user: User with an account on hatboxphotos.com

- **Email**: A registered user's email address. Provided at registration.
- **Username**: A registered user's username. Provided at registration.
- **Password**: A registered user's password. Provided at registrations
- Registered users shall be separated into admins and non-admin users
- Registered users shall log in to their account before uploading or downloading images
- Users shall have the option to provide their bio, occupation, and student status during the registration process

Unregistered user: User without an account on hatboxphotos.com

- Unregistered users shall only be able to view image thumbnails in place of full resolution images
- Unregistered users shall not be able to upload or download images

User: Anyone who visits hatboxphotos.com

- Users shall be separated into registered users and unregistered users
- Users shall be able to search for images
- Users shall be able to view image thumbnails

Functional Requirements

Priority 1:

Unregistered Users

- 1.1 Unregistered users shall be able to view photos on the home page
- 1.2 Unregistered users shall be able to sign up with an email and password
- 1.4 Unregistered users shall be able to view thumbnail image in place of full resolution images
- 1.5 Unregistered users shall be able to click on a thumbnail image and view it in medium resolution as well as its description
- 1.6 Unregistered users shall be prompted to sign up/in before downloading/uploading a photo
- 1.7 Unregistered users shall be able to search for images

Registered Users

- 2.1 Registered users shall have the same functionality as an unregistered user
- 2.2 Registered users shall be able to login with their email and password
- 2.3 Registered users shall be able to download an image in thumbnail, medium resolution, or high resolution
- 2.4 Registered users shall be able to upload an image
- 2.6 Registered users shall be able to view their profile

Admin Users

- 3.1 Admin users shall have the same functionality as a registered user
- 3.2 Admin users shall review the content and description of all uploaded images before they go live on hatboxphotos.com
- 3.3 Admin users shall approve if photo reviewed is acceptable
- 3.4 Admin users shall decline if photo reviewed is not acceptable
- 3.6 Admin users shall be able to remove any previously approved photo
- 3.7 Admins shall log in to their account on hatboxphotos.com before reviewing photos
- 3.8 Admins shall notify users in a message about why an uploaded photo was not approved to go live

Photos

- 4.2 Photos descriptions shall be optional
- 4.5 Photos shall be viewed on the home page
- 4.6 Photos to be viewed shall be determined by search results
- 4.7 Photos shall be viewed in thumbnail and medium resolution for unregistered users.

Search

5.1 Search bar shall have fuzzy search functionality

Priority 2:

Users

2.5 Registered users shall be updated if their uploaded image was approved/declined

3.5 Admin users shall have the ability to edit the template approve/decline email that is sent to the registered user

Photos

4.1 Photos shall have the ability to be described by tags

4.3 Photos shall show the all image info including user id, but not user

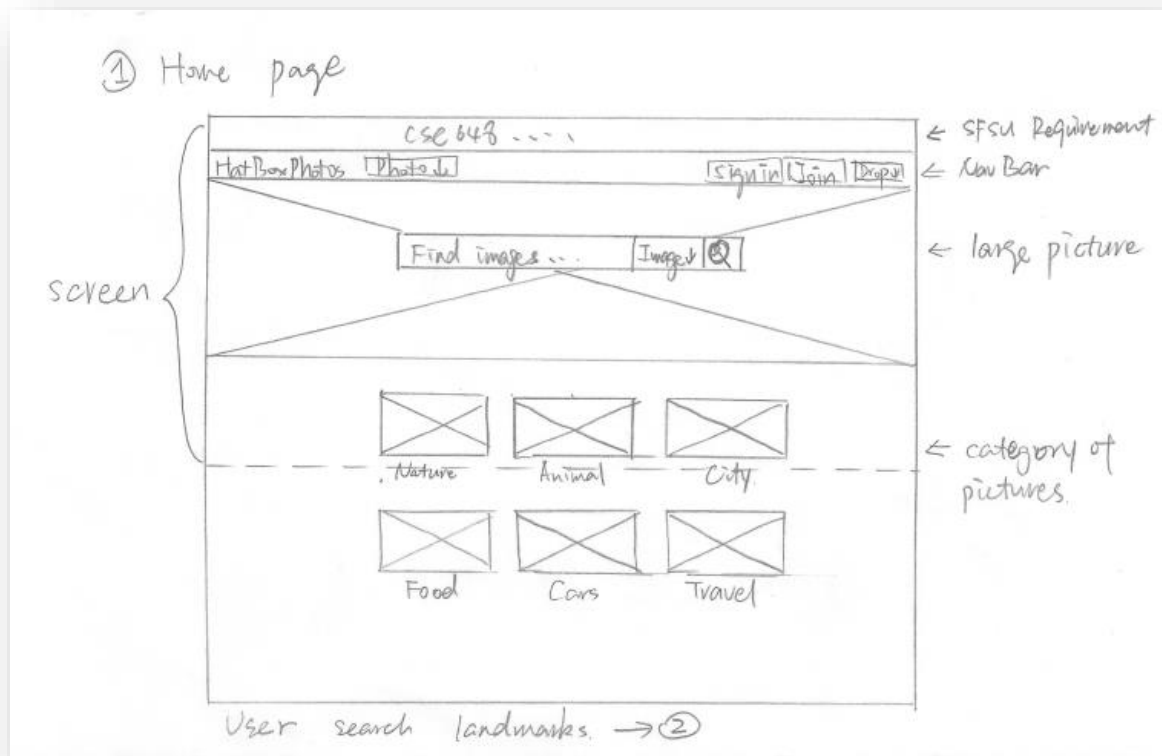
4.4 Photos shall show the Registered user who posted it

Priority 3:

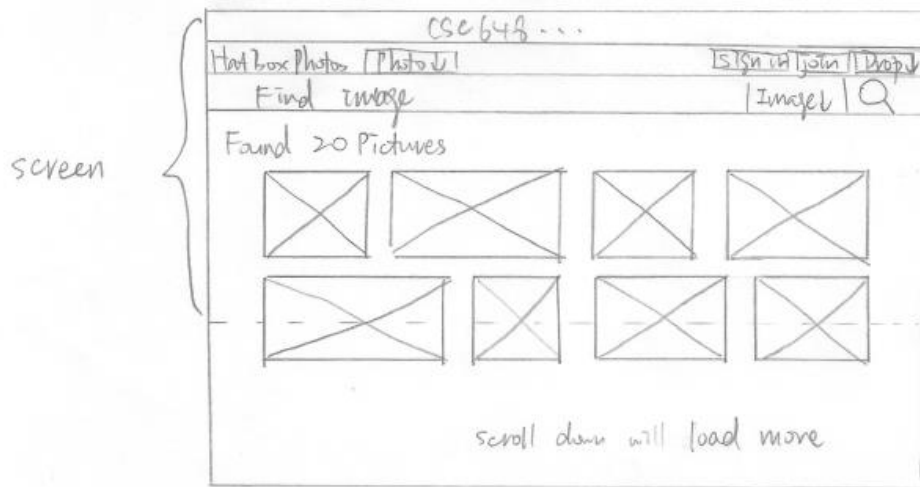
None at this time.

UI MOCKUPS AND STORYBOARDS

Use Case 1: Mary is a **college student**. She received a writing assignment about the city of San Francisco and had to use three landmark photos in her paper. When she gets home, she opens the home page of hatboxphotos. She searches the landmarks of San Francisco. Once she decides which landmark photos to use, she selects them, and is then prompted to register for a new account or log-in before she can download them to her computer. Then she proceeds to log out.

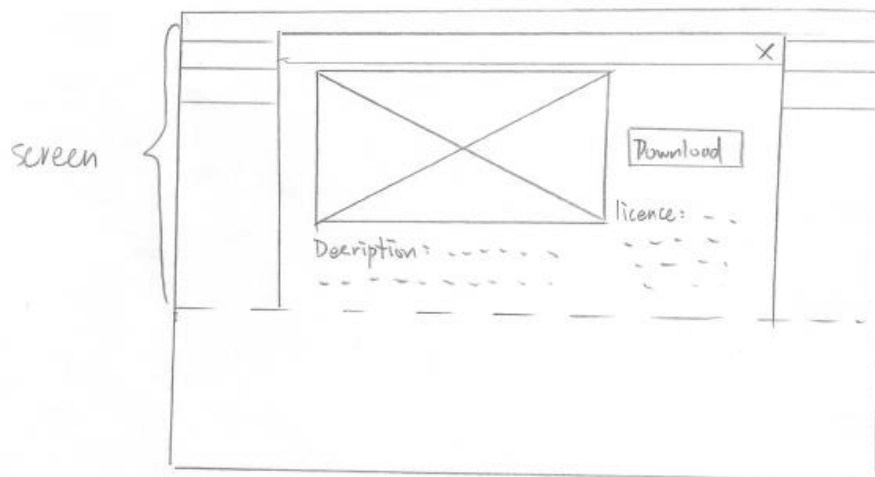


② Search Page.



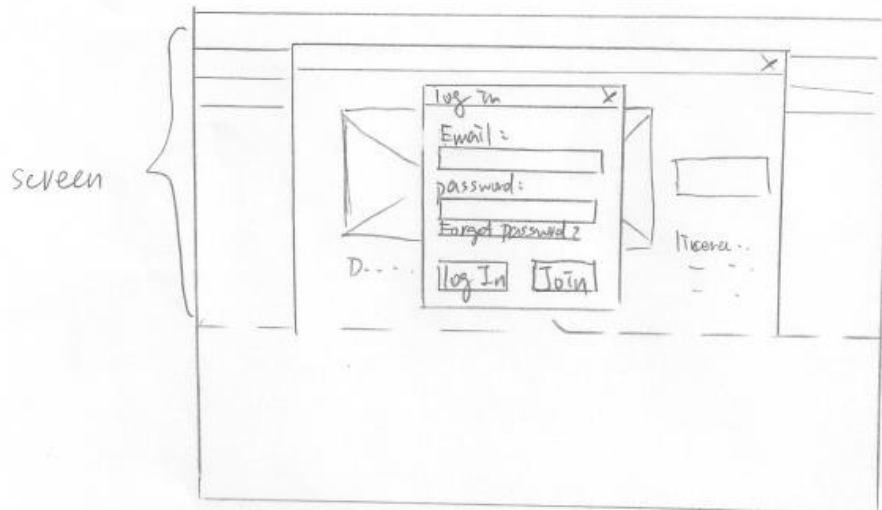
user click one photo → ③

③ Pop up photo details and download.



user clicks download button, system requires log in → ④

④ Request to log in



use clicks Join button, to sign up → ⑤

③ sign up Form:

Screen

CSC 648

HotBox Photos

Already a member? [sign in](#)

Email

Password

Retype Password

☐ I'm older

☐ Term ☐ Retail

☐ Not Robot

user joined and redirect user to → ④

④

screen

CSC 648

x

signed out

Description: - - -

- - -

licences: - - -

- - -

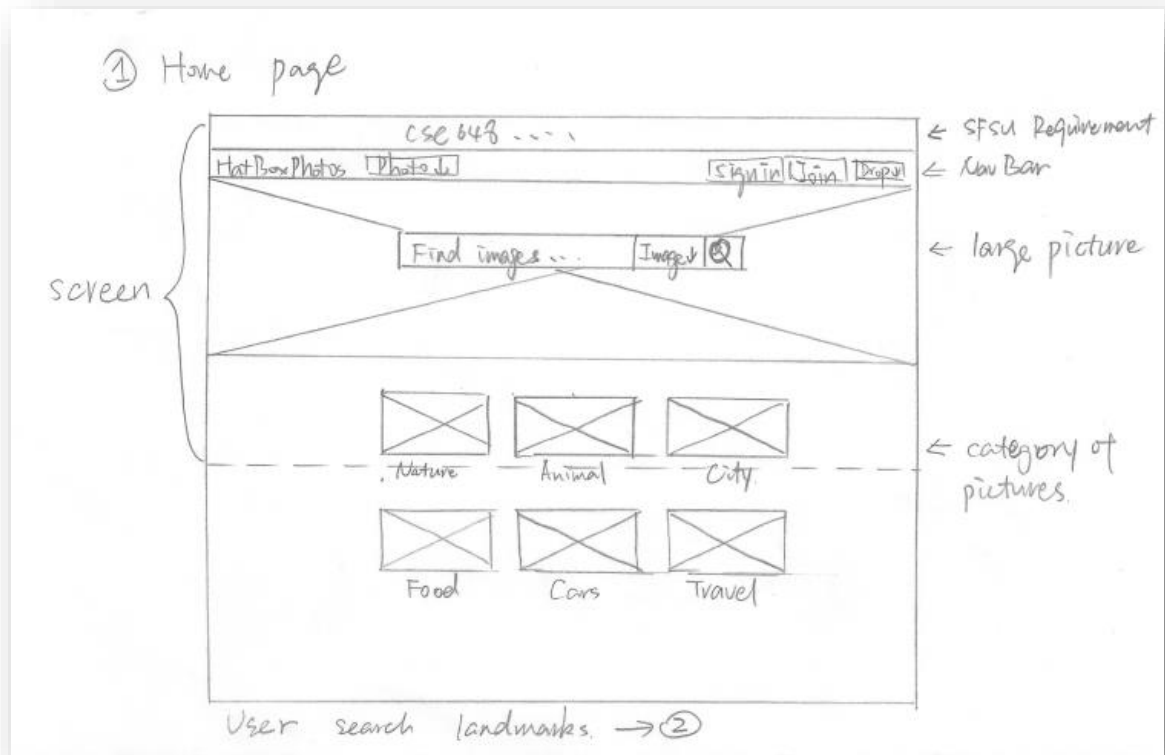
user downloads the photo and signed out.

← SFSN

← NavBar with signed in

← Search Bar

Use Case 2: Henry is a **photographer**. He had taken a trip to San Francisco and took lots of photos in the city. One Henry returns home, he opens the home page of hatboxphotos. He logs-in, chooses the photos that he wants to upload, writes a description about each photo, and uploads them one by one for approval. Then he proceeds to log out.



② sign in

CS648 ...

← SFSU Requirement

Nav Bar

Log in

Email:

password:

Forgot Password?

log in Join

pop up

After log in, then go to ③

③ Photographer Dash Board

CS648

Nav Bar

Id. Logout

Home

Profile

Upload

Message

History

Welcome to HatBoxPhotos

click upload → ④

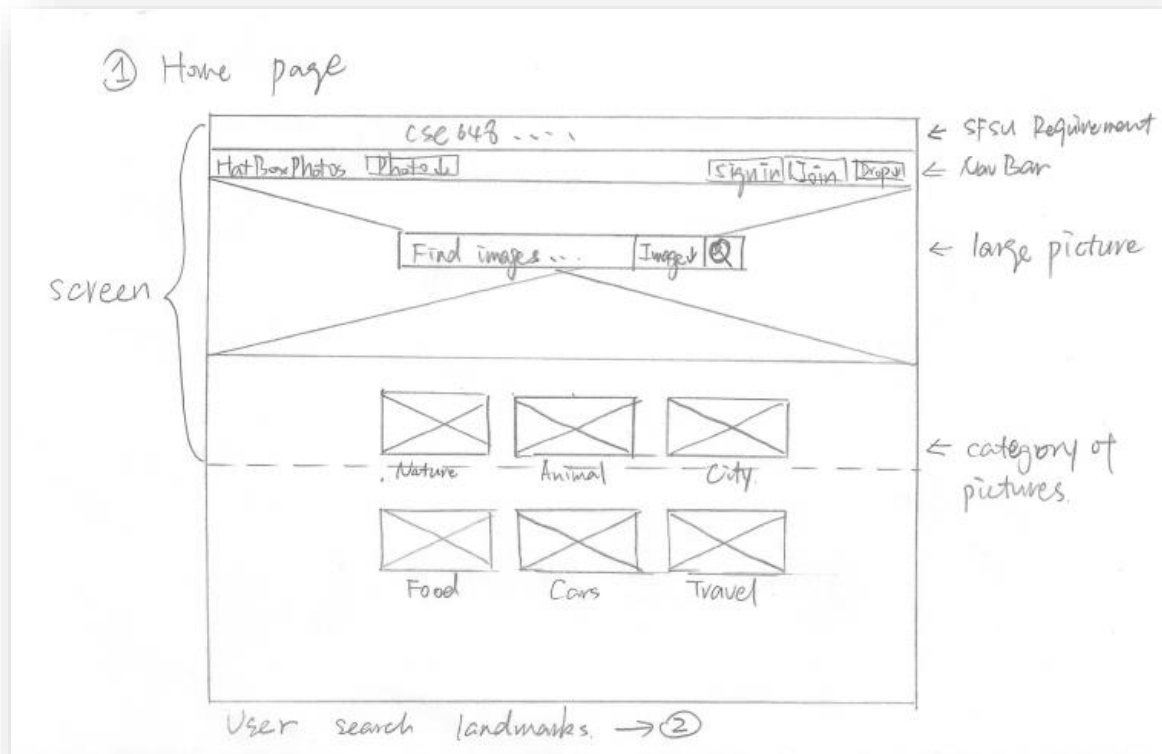
④ Upload

csc648	
NavBar	
<input type="checkbox"/>	Title: <input type="text"/>
<input type="checkbox"/>	Description: <input type="text"/>
<input type="checkbox"/>	Comment: <input type="text"/>
<input type="checkbox"/>	Link: <input type="text"/>
<input type="checkbox"/>	<input type="button" value="select"/>
	<input type="button" value="upload"/>

⑤

Csc648	
NavBar	Id Logout
<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	Thank you for choosing HatBoxPhotos. The Admin will get back to you In 24-48 Hours.

Use Case 3: Jennifer is an **admin**. When she starts work, she opens the home page of hatboxphotos and logs into her account. Then she navigates to the photo approval section. She reviews each photo and its description, then decides to approve or decline the request. She views the template email for approving/declining a photo and decides to add an additional section to further explain her decision. She finishes typing and submits her decision, sending an email to the picture's uploader.



② sign in

Hand-drawn sketch of a login page. The page has a header "CSC648" and a "Nav Bar". A "Log in" modal is centered, containing fields for "Email:", "password:", a "Forgot Password?" link, and "log in" and "Join" buttons. A magnifying glass icon is next to the password field. Annotations include "← SFSU Requirement" and "← Nav Bar" pointing to the header, and "popup" with an arrow pointing to the modal.

After log in, then go to ③

③ Admin Dashboard

Hand-drawn sketch of an Admin Dashboard. It features a sidebar with links: Home, Request, Message, History, and Setting. The main content area has a "Request" table with columns "id" and "Logout". The table lists "Arnold Upload" and "Betty Upload". A "Detail" section shows "Arnold Upload 20 picture that need review...". An annotation "④ Click Request" points to the "Request" link in the sidebar.

Admin Home Page → click Request → Review Request.
→ click Arnold Upload → ④

④

Admin Click each picture → ⑤

⑤

After approve, the system will send a message to the user. If decline, then comment and reason is also sent.

High level Architecture, Database Organization

DB organization:

- *Images Collection*
 - Image Document
 - ObjectID: image_id (auto generated by MongoDB)
 - GridFS: image_original
 - Base64: image_medium
 - Base64: image_thumbnail
 - String: image_file_name
 - String: image_file_format
 - String: category
 - String: uploaded_by
 - Date: upload_date
 - Int: file_size
 - Int: image_height
 - Int: image_width
 - Int: image_status
 - Int: views
 - Int: downloads
 - String: image_description
- *Registered Users Collection*
 - Account Document
 - Login Credentials Schema
 - ObjectID: _id (auto generated by MongoDB)
 - String: email
 - String: username
 - String: password
 - Permissions Document
 - Const: permissions
 - ADMIN
 - NON_ADMIN

Media storage:

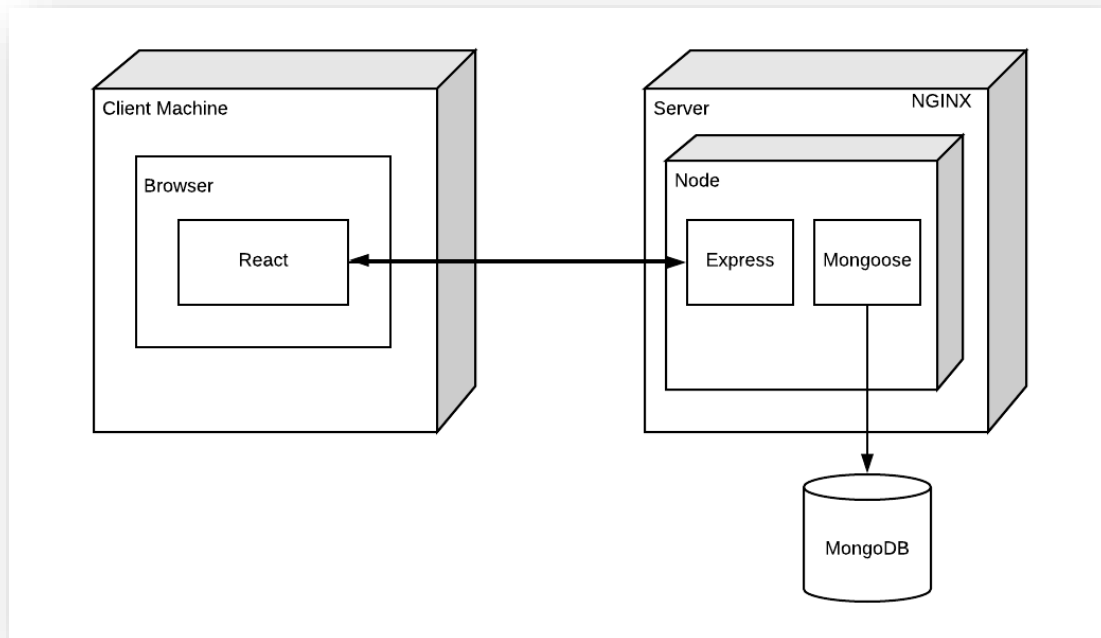
- Files
 - Base64 format
- GridFS
 - used for storing files above 16mb

Search/filter architecture and implementation:

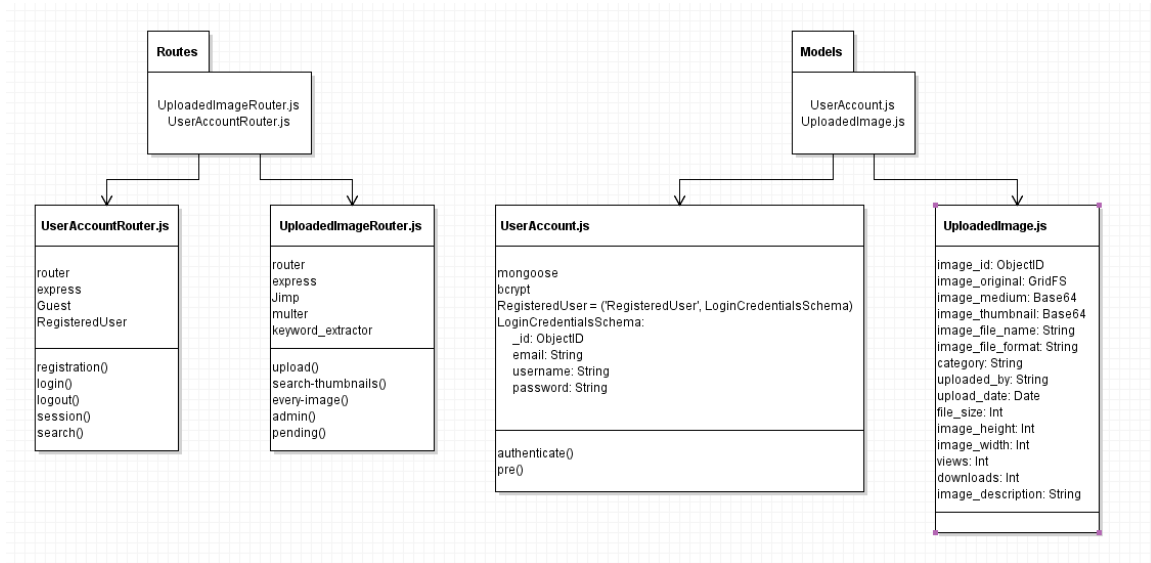
- Fuzzy search
 - Partial search using regex. Regex searches for image tags. Tags are generated by a keyword extractor which extracts key words from the image description

High Level UML Diagrams

Server UML:



Class UML:



Key Risks

Skills: Medium risk. We are still learning many of the tools that we are using. However, we are learning fast and sharing our knowledge, thus minimizing this risk.

Schedule: No risk at this time. All milestones thus far have been met on-time without the panic of the 11th hour.

Technical: No risk at this time. All previous technical issues have been resolved and measures put in place to prevent further problems.

Teamwork: No risk at this time. Teamwork is productive.

Legal: No foreseen risks at this time.

Milestone 3 review: Review Summary

Spring 2018

Summary and plans (to be filled out by the team after M3 review meeting and submitted with Milestone 5 documents)

Team: 5 Date: 7/27/2018 Number of students present: 6

1. *UI and functionality feedback (P1 functions only)*

- Home page
 - Make home page categories clickable
 - If a category is clicked, show all images in that specific category. Pulldown will also be on the same category
 - Include a little message/slogan. Show more of that and less of the banner. Ex: "We offer blah, blah, blah..."
 - Expand main menu across the top of the home page
 - Maybe add a logo
- Search (including search field validation)
 - Length of "about what we would type"
- Search results
 - Show other images if search yields no results. "Here are some other images"
 - Have page navigation if there is more than 1 page of search results.
 - No page navigation if only one page of search results
 - Results should be on a new page
- Filtering
 - Include ALL category in the pulldown menu
 - Default search filter should be the ALL category
- Search Details and maps
 - Clicking images from search results should show image details & a one-line image description under the image
- Messaging/contact seller
 - Messaging/contact seller has not been implemented at this time (7/27/2018)
- Listing Upload
 - Don't bury the upload button

- Seller dashboard
 - No seller dashboard at this time (7/27/2018)
- UI responsiveness (resize the browser)
 - Good UI responsiveness. Browser also responds at mobile size.
 - Fix alignment in login & registration forms
- Performance (e.g. display of results list)
 - Search results load quickly
- Join Us
 - Let user know that email will be used as userID
 - Have Terms of Agreement
 - Include password confirmation
 - Ask to re-enter password
 - Add a CAPTCHA
 - Add cancel to Sign In form
- Admin Page
 - Click image leads to two buttons: Accept and Reject
- Contributor
 - Too complicated right now
 - Menu should highlight which tab is selected
 - Fill in boxes are too long
 - Put title of page
 - Submit/Cancel is too small
 - Include a message such as “Upon submission, it will take up to 24 hours to approve” after contributor uploads an image
- Database
 - Add Category Schema
 - Have 5 images per category
- GitHub
 - GitHub comments should be more concise

2. *Feedback on coding style, github, database etc.*

- Git/Github organization (e.g. organization of branches)
 - GH organization looks good. Good use of branches.
- Git/Github usage: Comments on positing; Number of posting to github; Appr. even distribution of submissions among team members (check github post stats for all members)
 - Commit messages look OK. Some duplicated messages. Commit counts are slanted towards one member.
- Code documented (header, in code) with good coding style
 - Comments look good, Header comments exist.
- MVC/OO patterns followed up
 - Followed
- Frameworks (back end front end) deployed correctly
 - Installed and working
- Database organization (tables, naming...)
 - DB looks OK. Static table may be excluded given mongoDB is used.
- Blobs being used? If so, is it working?
 - Looked at Models, paths are used.
- Adherence to best practices of security (PW encrypted, search inputs verified etc.)
 - Passwords are encrypted (at least looks this way by inspecting models in code).
- Efficiency (proper use of image thumbnails, efficient search eytc.)
 - Looks good thumbnails used.
- Other
 - Looks like a lot of progress has been made over the past week and a half

3. *Feedback on Project status*

- *Teamwork*: is the team working out, any related issues
 - Teamwork has been mostly good. One member has been a little behind.
- *Risks*: all actual (not hypothetical) risks (schedule, technical, skills etc.) should be identified and either resolved or plans made how to resolve them asap.
 - Slight schedule risk. Milestone 3 was performed 2 days after other teams because we were already behind.
- *Coding practices*: by this time you should have decided which coding style to use and actually use it, so be ready to report on this. Actual code will be verified
 - Coding style decided but needs to be enforced
- *Usage of proper SE code management practices*: be ready to comment on how you manage code submissions, reviews, build and test etc. This will be verified by looking at actual code and server logs
 - Good code management
- *How did you address site security and safe coding practices*: PW management, use of secure protocols, checking for valid inputs etc.
 - Passwords encrypted
 - Need to add field validation for login & registration
- *Digital content* (e.g. images, video) please provide information on status and availability, chosen formats, resolution etc.
 - Formats: jpeg, png, and gif
 - Thumbnails and medium resolution images have been implemented
- Agree on final P1 list and from then on focus only on those
- Other

4. *List of committed P1 functions (agreed upon at the review meeting)*

- Image Search
- Image Upload
- Login and Registration
- Admin functionality

5. *Action items (plans based on feedback above)*

Front End Plans

- Spread main menu items across the top of the homepage
- Fix registration form and contributor form
- Remove Student and Photographer from the menu
- Add the ALL category to pulldown menu
- Adjust page navigation for image search results
- Adjust admin page to have 2 buttons: Accept and reject
- Unbury the upload button

Back End Plans

- Add Categories schema to the database
- Add input validation for login & registration
- Add more images to the database

SW Engineering CSC648/848 Summer 2018

Hat Box Photos

Team 5

Amelia Rawlings (Amelia.Rawlings@gmail.com)

Front-End

Arnold Jiadong Yu

Dalu Li

Betty Reaney

Back-End

Keawa Rozet

U Keong Cheong (Ray)

Albert Du

Milestone 4

Revision Table		
8/6/2018	1.0	Initial document

Product Summary

Welcome, everyone, to the beta launch of HatBoxPhotos. We are delighted to offer you the opportunity to be one of the first to see our brand new free stock photo website. Features to look forward to include:

1. **Fast Search**

Need a high quality image for school, work, or just because? You can quickly find the perfect image for your needs at HatBoxPhotos.

2. **Uploading images**

Share your special moments with other users through our simple image uploading process!

3. **Admin moderation**

HatBoxPhotos enforces a family-friendly image policy. Our admin will either approve or reject uploaded images for live viewing within 24 hours.

Please visit us at <https://hatboxphotos.com/>. We look forward to your feedback.

Usability Test Plan

Test Objective

This usability test will rate the effectiveness, efficiency, and satisfaction of the search function. The search function is the main function of our application. Therefore, we must ensure that users are able to find specific photos with ease. Things to consider are search results design, search bar and search button placement.

Our effectiveness test will measure task completion and count errors per task. Next, our efficiency test will measure the number of clicks, number of screens, and time to complete the task. Finally, our satisfaction test will measure user satisfaction using a Lickert Scale questionnaire.

Test Plan

Intended User: Student who needs pictures for Biology class.

Tasks to be accomplished:

1. Search for all pictures in the "Animal" category.
2. Search and download an image of a tree from the starting point.
3. Search and download an image of a lion from the starting point.
4. Search and download an image of a pizza from the starting point.

Starting Point: Home Page (<https://hatboxphotos.com/>)

System Setup 1: (Desktop)

Processor: Intel Core i7-6700k 4.00GHz
Memory: 32.0GB
System Type: Windows 10 64bit
Browser: Firefox Quantum 59.01 (64bit)

System Setup 2: (Laptop)

Processor: 1.8 GHz Intel Core i5
Memory: 4GB
System Type: macOS High Sierra
Browser: Google Chrome Version 67.0.3396.99

Effectiveness Test:

Desired time: less than 30 seconds.

Test/Use Case	% Completed	Errors	Comments
Search All animal category pictures	100%	None	Search result returned all animal pictures.
Search/Download a tree picture	80%	User search city category.	Search result returned 0 results for tree in city category.
Search/Download a lion picture	90%	User search nature category.	Search result returned 0 results for lion in nature category.
Search/Download a raccoon picture	100%	None	Search result returned 1 result for raccoon in all pictures.

Efficiency Test

Test/Use case	# of clicks	# of screens	Time
Search All animal category pictures	3	3	15 seconds
Search/Download a tree picture	5	5	20 seconds
Search/Download a lion picture	4	5	20 seconds
Search/Download a raccoon picture	5	4	25 seconds

Satisfaction Test

Answer the questions below according to your experience with search function.

1. The search bar was easy to use.

()	(X)	()	()	()
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

2. The search results page was clear.

()	()	(X)	()	()
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

3. I would be able to find any picture with search function.

()	()	(X)	()	()
Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

QA Test Plan

Test Objective

The QA objective is to test the image search and delete user function. Searching by image category, by image name, as well as delete an existing user shall be performed.

System setup

Processor: Intel Core i7-6700k 4.00GHz

Memory: 32.0GB

System Type: Windows 10 64bit

Browser: Firefox Quantum 59.01 (64bit), Google Chrome 67.0.3396.99 (64bit)

Features to be tested.

Search an image by category.

Search an image by name/description

Delete a user

Test Cases

1. Search an image by category.
 - a. From the home page search bar, select the 'Nature' category and have nothing in the search input. Click on 'Search' button. The result shall return 4 thumbnails.
2. Search an image by name/description
 - a. From the home page search bar, select the 'All' category and input "Man" in the search input. Click on 'Search' button. The result shall return 4 thumbnails.
3. Delete a user
 - a. Click on Admin button on Navigation bar. Then click on 'Manage User', select 'test5@gmail.com' and press delete button. Confirm by click delete button again on pop-up. The page will refresh and user with email 'test5@gmail.com' will not displayed.

Test Results

Test #	Test Title	Test Description	Test Input	Expected Output	Firefox	Chrome
1	Search an image by category.	Test category search functionality	"Nature" without input on search	4 thumbnails	Pass	Pass
2	Search an image by name/description	Test input search functionality	Input "Man"	4 thumbnails	Pass	Pass
3	Delete a user	Test delete user functionality	None	Delete user successfully	Pass	Pass

Code Review

Coding Style

- JSDoc documentation is used for class headers and function headers
- Inline comments are used to document non-headers

Emails:

Chosen Code for Peer Review Email:

Albert Du <adu1@mail.sfsu.edu>

To: Keawa.Rozet@yahoo.com

Hello Keawa,

I finished writing code for the search route in UploadedImageRouter.js. Could you check it out?

```
router.get("/search-thumbnails", function(req, res) {
  var noMatch = false;

  // Handles if the user enters something into the search bar
  if(req.body.search) {
    const regex = new RegExp(escapeRegex(req.body.search), 'gi');

    // Get all images and image thumbnails from image db
    // based on search and category
    Images.find({'image_category': req.body.image_category, tags: regex}, function(err, allImages) {
      thumbnails = allImages.map(function (allImages) {return allImages.image_thumbnail_path;});
      if(err) {
        console.log(err);
      } else {
        // if no images were found, return all images in selected category
        if (allImages.length < 1) {
          noMatch = true;
          // Get all images and thumbnails from db in only the selected category
          Images.find({'image_category': req.body.image_category}, function(err, restOfImages){
            thumbnails = restOfImages.map(function
(restOfImages)    {return restOfImages.image_thumbnail_path;});
            if(err){
              console.log(err);
            } else {
              res.json({images: restOfImages, thumbnails: thumbnails, noMatch: noMatch});
            }
          });
        }
      }
    });
  }
});
```

Thanks

Albert Du

Peer Feedback Email:

Keawa Rozet <keawa.rozet@yahoo.com>

To: Albert Du

Hey Albert!

The code looks really solid and is almost complete. I was testing it out using Postman and I was able to figure out some bugs that were preventing it from connecting.

I commented in the areas with "//FIX ME - " where you need to take a look.

```
router.get("/search-thumbnails", function(req, res) {
  var noMatch = false;

  // Handles if the user enters something into the search bar
  //FIX ME – don't use body because that is for x-www-form-urlencoded
  //instead use req.query since we are using a Form
  if(req.body.search) {
    //FIX ME – use req.query
    const regex = new RegExp(escapeRegex(req.body.search), 'gi');

    // Get all images and image thumbnails from image db
    // based on search and category
    //FIX ME – include finding by status: 1, that way we return only accepted images
    // use req.query
    Images.find({'image_category': req.body.image_category, tags: regex}, function(err, allImages) {
      thumbnails = allImages.map(function (allImages) {return allImages.image_thumbnail_path;});
      if(err) {
        console.log(err);
      } else {
        // if no images were found, return all images in selected category
        if (allImages.length < 1) {
          noMatch = true;
          // Get all images and thumbnails from db in only the selected category
          //FIX ME – req.query, status: 1
          Images.find({'image_category': req.body.image_category}, function(err, restOfImages){
            thumbnails = restOfImages.map(function (restOfImages) {return
restOfImages.image_thumbnail_path;});
            if(err){
              console.log(err);
            } else {
              res.json({images: restOfImages, thumbnails: thumbnails, noMatch: noMatch});
            }
          });
        }
      }
    });
  }
});
```

Send me what you have when you fix these areas, thanks Albert!

Best,

Keawa Rozet

Self-check On Best Practices for Security

- Major assets being protected include email addresses, usernames, and passwords
- Passwords in the database have been encrypted using the bcrypt hashing function
- Input validation is present in login, registration and the search bar
 - The Mongoose-unique-validator plugin prevents duplicate registration
 - Mongoose-unique-validator combined with the uniqueCaseInsensitive Mongoose schema validator perform case-sensitive validation for emails and usernames during login & registration
 - Username and password length is validated during registration by the Mongoose schema validators minlength and maxlength
 - Search bar only allows characters and spaces

Self-check: Adherence to Original Non-functional Specs

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). **(DONE)**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. **(ON TRACK)**
3. Data shall be stored in the team's chosen database technology on the team's deployment server. **(DONE)**
4. No more than 50 concurrent users shall be accessing the application at any time. **(DONE)**
5. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **(DONE)**
6. The language used shall be English. **(DONE)**
7. Application shall be very easy to use and intuitive. **(ISSUES)**
 - We are struggling to connect front-end and back-end. We cannot test fully on localhost because of "Access-Control-Allow-Origin" problem. Have tried many ways to solve it but failed.
 - We're having trouble with sign in and sign up pages.
8. Application shall render well on mobile devices (UI shall be responsive). **(DONE)**
9. Google analytics shall be added. **(ON TRACK)**
10. No e-mail clients shall be allowed. **(DONE)**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated. **(DONE)**
12. Site security: basic best practices shall be applied (as covered in the class) **(DONE)**

13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development (**DONE**)
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2018. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). (**DONE**)

Demo Sceenshots

Home Page

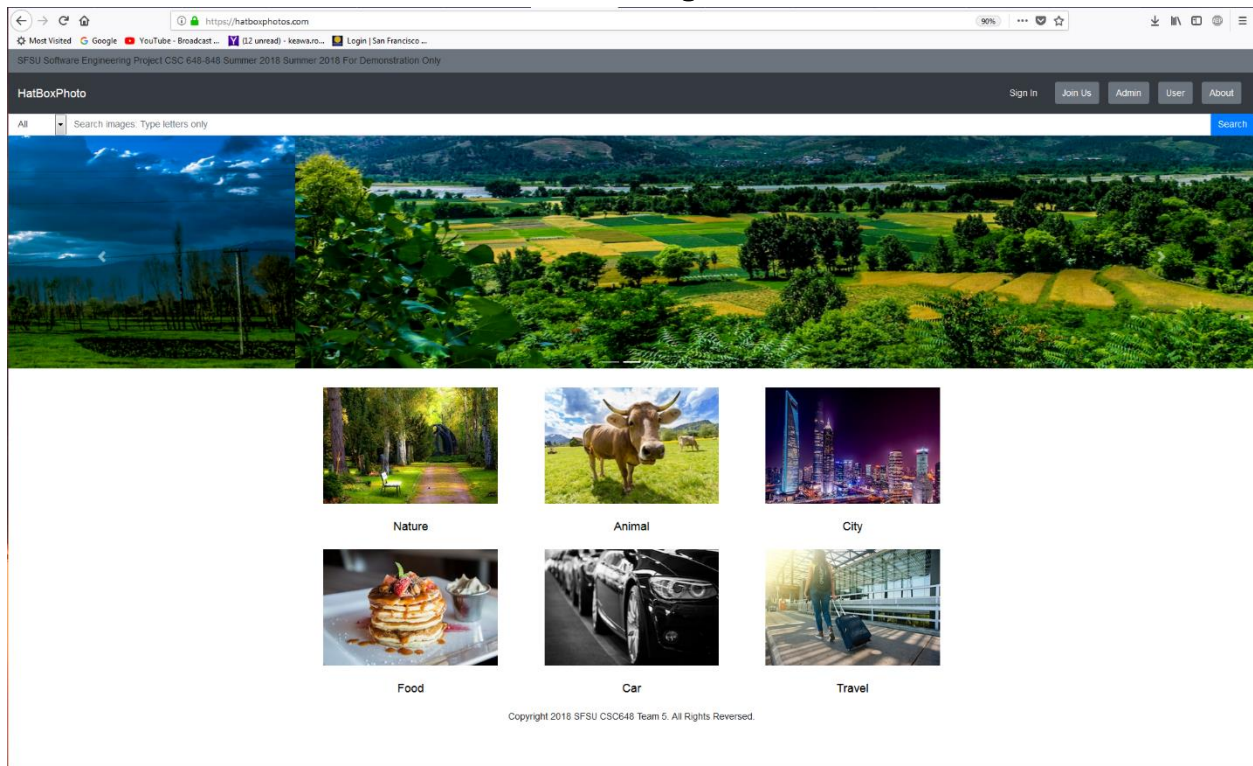
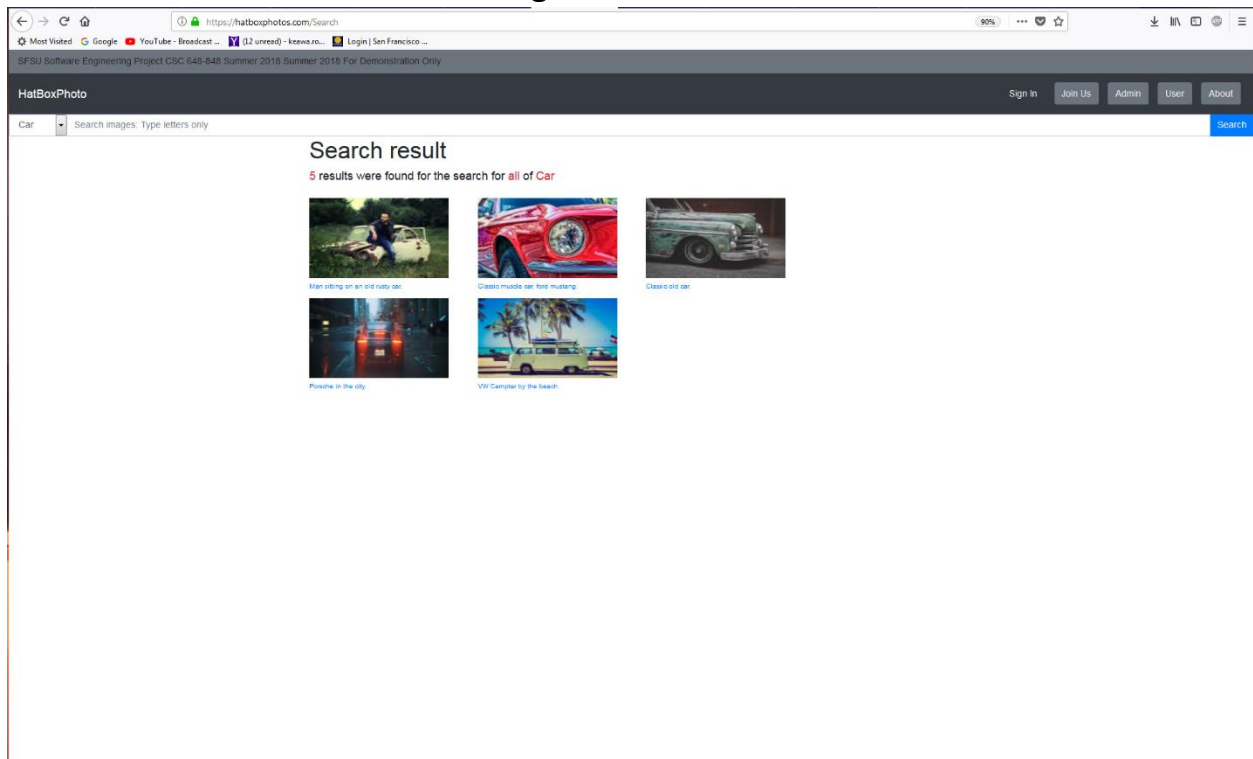


Image Search Results



Upload Image

← → ↻ 🏠 <https://hatboxphotos.com/user/upload> 90% *** ☆

Most Visited Google YouTube Broadcast ... 12 unread - keawa.m... Login | San Francisco ...

SFSU Software Engineering Project CSC 648-648 Summer 2018 Summer 2018 For Demonstration Only

HatBoxPhoto Sign In Join Us Admin User About

All Search images: Type letters only Search

Upload

Upload Photo

Category* (required)
Nature

Please choose image category:

Description* (required)
Description is used for search.

Please enter image description.

Comment (option)

Please enter comment about image.

Image File* (required)
Browse... No file selected.

Please choose one image you want to upload.

Cancel Confirm

Press Cancel will reload the page. Admin will approve or disapprove in 24-48 hours.

Sign In

← → ↻ 🏠 <https://hatboxphotos.com> 90% *** ☆

Most Visited Google YouTube Broadcast ... 12 unread - keawa.m... Login | San Francisco ...

SFSU Software Engineering Project CSC 648-648 Summer 2018 Summer 2018 For Demonstration Only

HatBoxPhoto Sign In Join Us Admin User About

All Search images: Type letters only Search

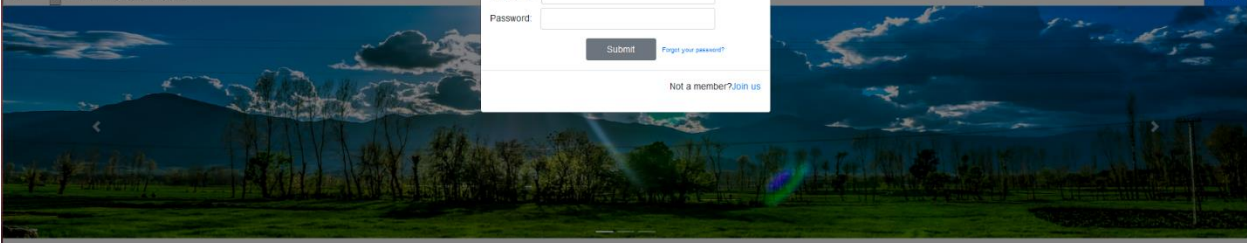
Sign In


Username:

Password:


Submit [Forgot your password?](#)

[Not a member? Join us](#)







Nature




Animal




City



Food



Car



Travel

Copyright 2018 SFSU CSC648 Team 5. All Rights Reserved.

Registration

Browser: <https://hatboxphotos.com>

SFSU Software Engineering Project CSC 648-648 Summer 2018 Summer 2018 For Demonstration Only

HatBoxPhoto

Sign In Join Us Admin User About

All Search images. Type letters only

Welcome to join us!

Email:

Username:


Password:


Confirm:


☐ I am at least 13 years of age

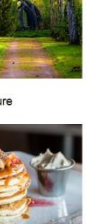
☐ I agree to the [Terms of Agreement](#)


☐ I am not a robot

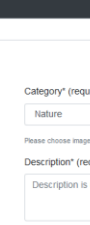
 Nature

 Animal

 City

 Food

 Car

 Travel

Copyright 2018 SFSU CSC648 Team 5. All Rights Reserved.

Registered User

Browser: <https://hatboxphotos.com/user/upload>

SFSU Software Engineering Project CSC 648-648 Summer 2018 Summer 2018 For Demonstration Only

HatBoxPhoto

Sign In Join Us Admin User About

All Search images. Type letters only

Upload Photo

Category* (required)

Please choose image category.

Description* (required)

Description is used for search.

Please enter image description.

Comment (option)

Please enter comment about image.

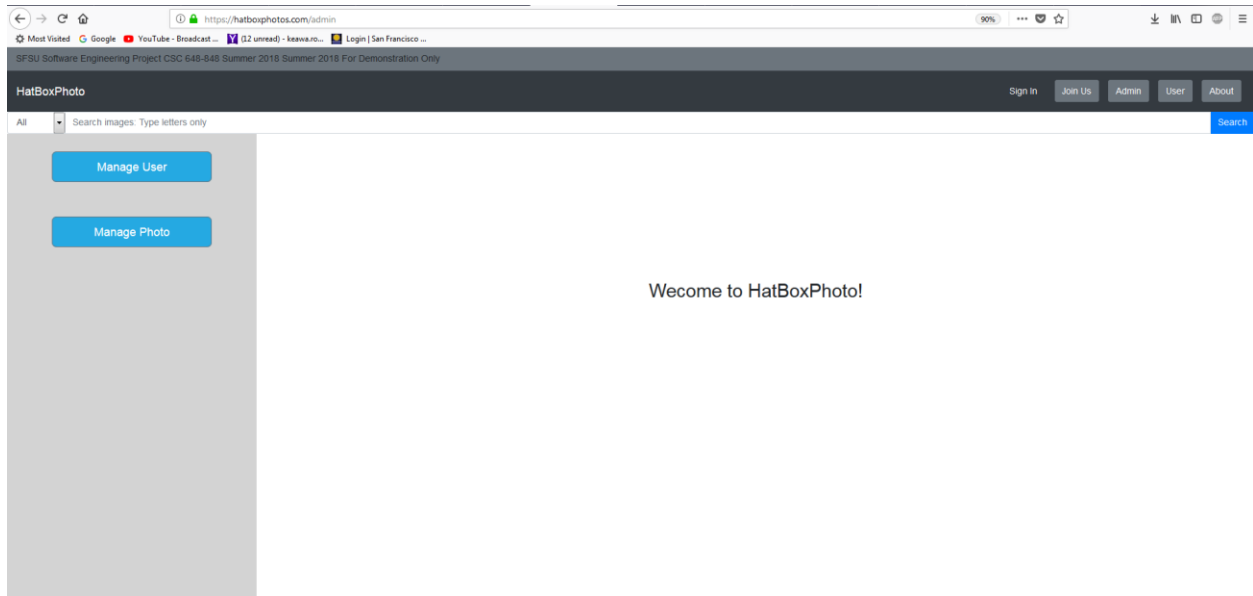
Image File* (required)

No file selected.

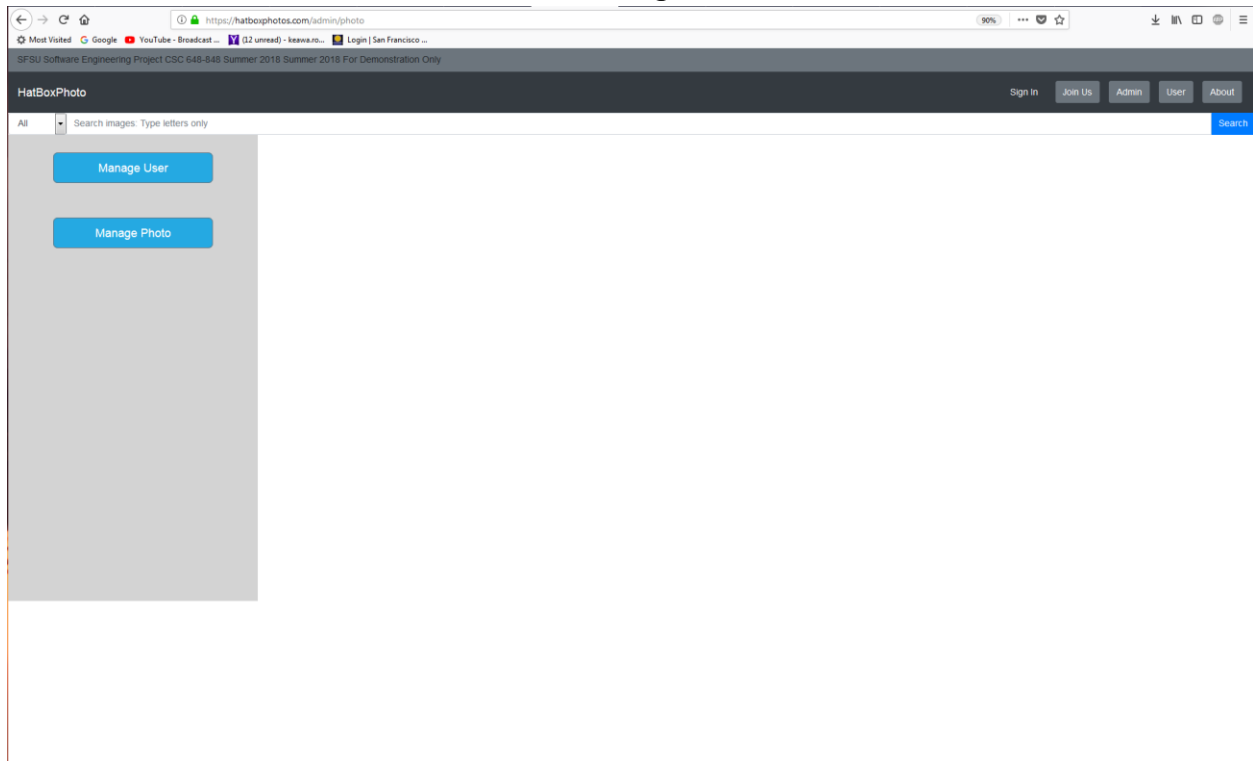
Please choose one image you want to upload.

Press Cancel will reload the page. Admin will approve or disapprove in 24-48 hours.

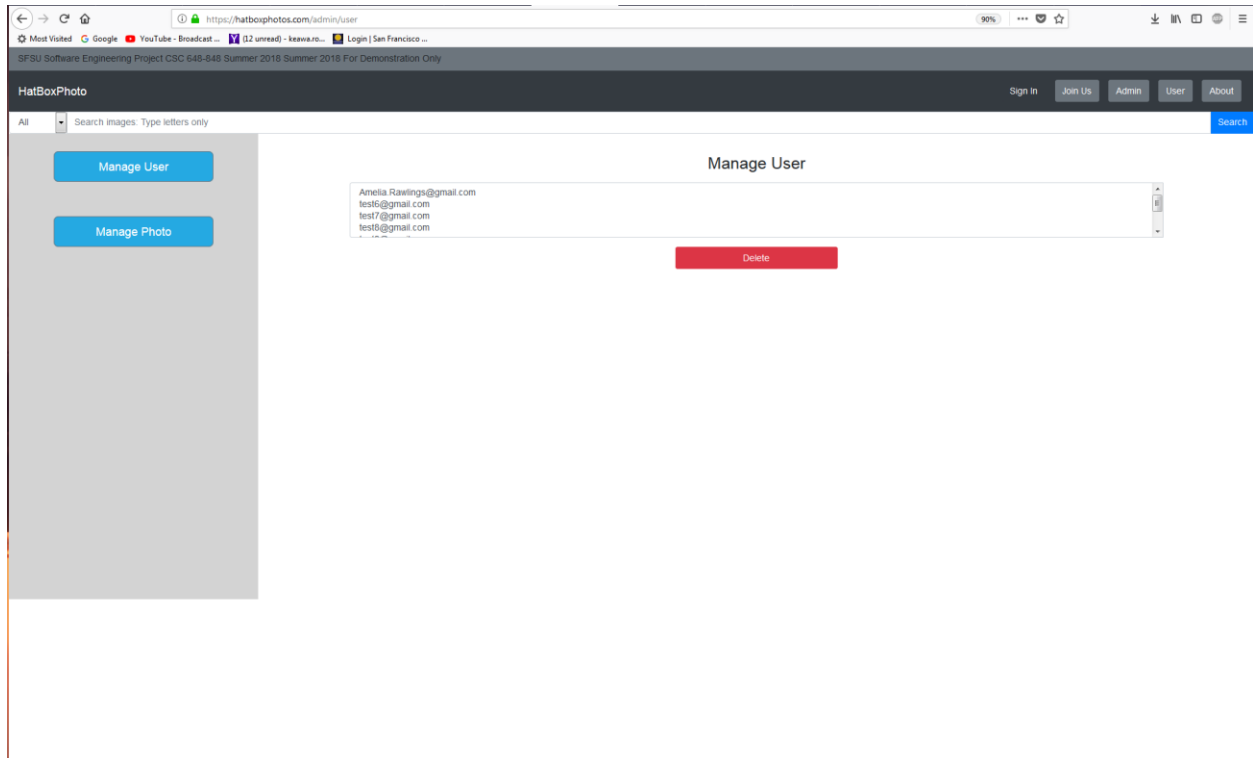
Admin Page



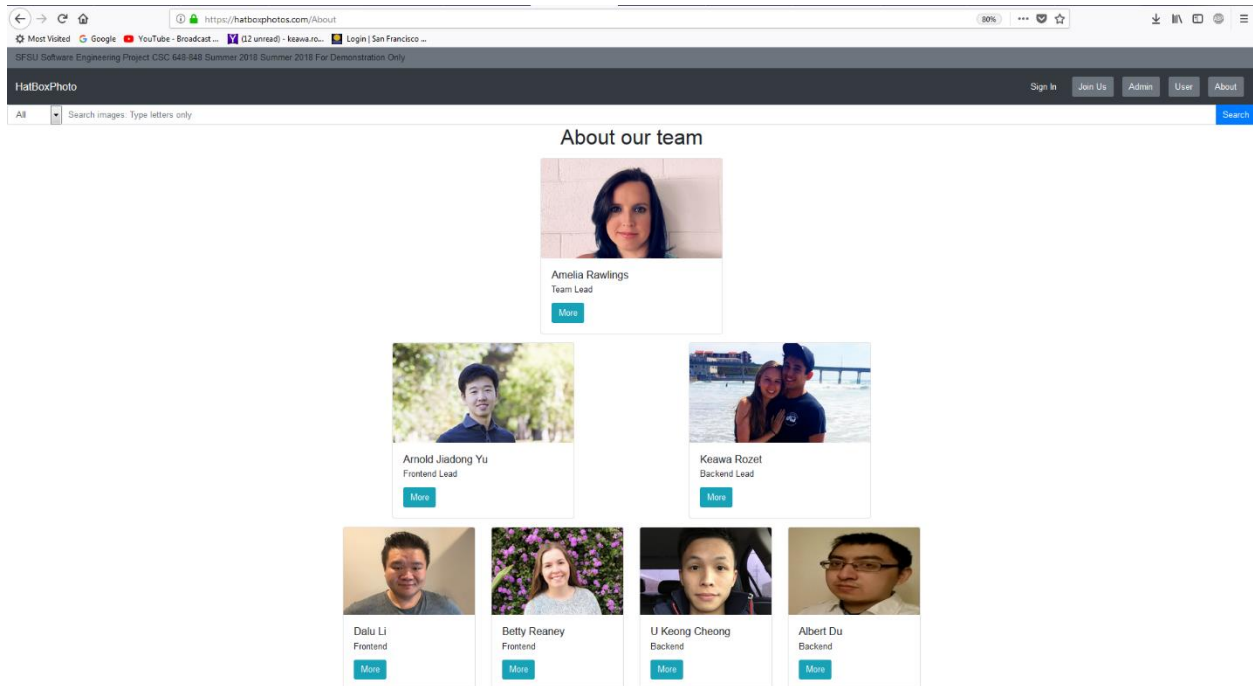
Admin Manage Photos



Admin Manage Users



About Page



Database Screenshots

HatBoxPhotosDB Collections

The screenshot shows the MongoDB Compass interface for the 'hatboxphotos.com:27020' cluster. The left sidebar lists the database 'hatboxphotosDB' and its collections: 'categories', 'registeredusers', 'sessions', and 'uploadedimages'. The main panel displays a table of collections with the following data:

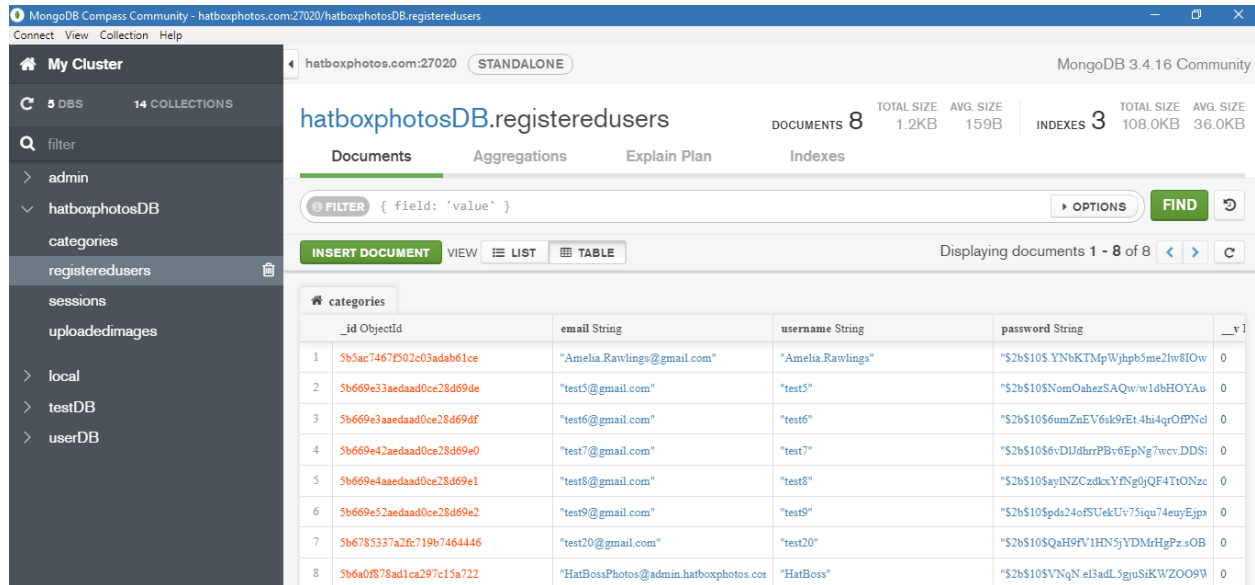
Collection Name	Documents	Avg. Document Size	Total Document Size	Num. Indexes	Total Index Size
categories	7	51.0 B	357.0 B	1	36.0 KB
registeredusers	8	158.6 B	1.2 KB	3	108.0 KB
sessions	19	190.0 B	3.5 KB	2	72.0 KB
uploadedimages	43	497.5 B	20.9 KB	1	36.0 KB

Categories Collection

The screenshot shows the MongoDB Compass interface for the 'hatboxphotosDB.categories' collection. The left sidebar lists the database 'hatboxphotosDB' and its collections: 'categories', 'registeredusers', 'sessions', and 'uploadedimages'. The main panel displays the 'categories' collection with 7 documents. The 'Documents' tab is selected, showing a table of documents with the following data:

_id	category	__v
5b5cbe4c5d54232a6e5e210c	"Nature"	0
5b5cbe745d54232a6e5e210d	"Animal"	0
5b5cbeaa5d54232a6e5e210e	"Car"	0
5b5cbebb15d54232a6e5e210f	"City"	0
5b5cbebb5d54232a6e5e2110	"Food"	0
5b5cbecc15d54232a6e5e2111	"Travel"	0
5b637a74766ae46704434ca3	"People"	0

RegisteredUsers Collection



hatboxphotosDB.registeredusers

DOCUMENTS 8 TOTAL SIZE 1.2KB AVG. SIZE 159B INDEXES 3 TOTAL SIZE 108.0KB AVG. SIZE 36.0KB

Documents Aggregations Explain Plan Indexes

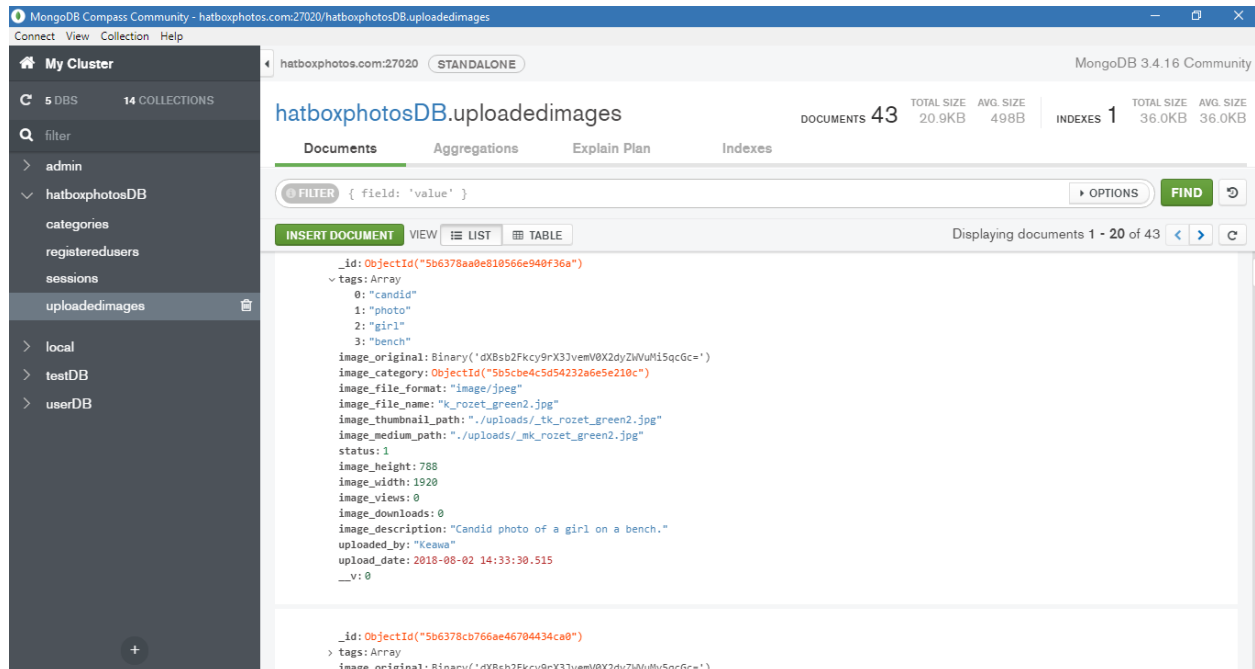
FILTER { field: 'value' } OPTIONS FIND

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 8 of 8

	_id ObjectId	email String	username String	password String	_v
1	5b5ac7467f502c03adab61ce	"Amelia.Rawlings@gmail.com"	"Amelia.Rawlings"	"\$2b\$10\$YNbKTMpWjhp5me2lw8fOw"	0
2	5b669e33aadaad0ce28d69de	"test5@gmail.com"	"test5"	"\$2b\$10\$NomOahezSAQw/w1dbHOYAu"	0
3	5b669e33aadaad0ce28d69df	"test6@gmail.com"	"test6"	"\$2b\$10\$6umZuEV6sk9rEt4hi4qrOPNcl"	0
4	5b669e42aadaad0ce28d69e0	"test7@gmail.com"	"test7"	"\$2b\$10\$6vDJdhrrPBv6EpNg7wcv.DDSI"	0
5	5b669e44aadaad0ce28d69e1	"test8@gmail.com"	"test8"	"\$2b\$10\$aylNZCzdcxYfNg0jQF4TiONzc"	0
6	5b669e52aadaad0ce28d69e2	"test9@gmail.com"	"test9"	"\$2b\$10\$pd24oISUekUv75iqu74euyEjpx"	0
7	5b6785337a2fc719b7464446	"test20@gmail.com"	"test20"	"\$2b\$10\$QaH9fV1HN5jYDMrHgPz.sOB"	0
8	5b6a0f878ad1ca297c15a722	"HatBossPhotos@admin.hatboxphotos.com"	"HatBoss"	"\$2b\$10\$VNqN_eI3adL5gju5iKWZOO9W"	0

UploadedImages collection



hatboxphotosDB.uploadedimages

DOCUMENTS 43 TOTAL SIZE 20.9KB AVG. SIZE 498B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Documents Aggregations Explain Plan Indexes

FILTER { field: 'value' } OPTIONS FIND

INSERT DOCUMENT VIEW LIST TABLE

Displaying documents 1 - 20 of 43

```
{
  "_id": ObjectId("5b6378aa0e810566e940f36a"),
  "tags": Array(
    0: "candid"
    1: "photo"
    2: "girl"
    3: "bench"
  ),
  "image_original": Binary("dXBsb2Fkcy9rX3JvemV0X2dyZWVudHl5cGc="),
  "image_category": ObjectId("5b5cbe4c5d54232a6e5e210c"),
  "image_file_format": "image/jpeg",
  "image_file_name": "k_rozet_green2.jpg",
  "image_thumbnail_path": "./uploads/_tk_rozet_green2.jpg",
  "image_medium_path": "./uploads/_mk_rozet_green2.jpg",
  "status": 1,
  "image_height": 788,
  "image_width": 1920,
  "image_views": 0,
  "image_downloads": 0,
  "image_description": "Candid photo of a girl on a bench.",
  "uploaded_by": "Keawa",
  "upload_date": 2018-08-02 14:33:30.515,
  "_v": 0
}
```

```
{
  "_id": ObjectId("5b6378cb766ae46704434ca0"),
  "tags": Array(
    0: "candid"
    1: "photo"
    2: "girl"
    3: "bench"
  ),
  "image_original": Binary("dXBsb2Fkcy9rX3JvemV0X2dyZWVudHl5cGc="),
  "image_category": ObjectId("5b5cbe4c5d54232a6e5e210c"),
  "image_file_format": "image/jpeg",
  "image_file_name": "k_rozet_green2.jpg",
  "image_thumbnail_path": "./uploads/_tk_rozet_green2.jpg",
  "image_medium_path": "./uploads/_mk_rozet_green2.jpg",
  "status": 1,
  "image_height": 788,
  "image_width": 1920,
  "image_views": 0,
  "image_downloads": 0,
  "image_description": "Candid photo of a girl on a bench.",
  "uploaded_by": "Keawa",
  "upload_date": 2018-08-02 14:33:30.515,
  "_v": 0
}
```


Google Analytics

Right now

4

active users on site

DESKTOP

100%

Top Referrals:

Source	Active Users	↓
There is no data for this view.		

Top Social Traffic:

Source	Active Users	↓
There is no data for this view.		

Top Keywords:

Keyword	Active Users	↓
There is no data for this view.		

Pageviews

Per minute

1:10

5.0

2.5

-26 min

-21 min

-16 min

-11 min

Top Active Pages:

Active Page			
1.	/admin/photo	3	75.00%
2.	/	1	25.00%

Top Locations:

Team Member Contributions

Amelia Rawlings (Team Lead)

CSC 648-848 Milestone 5: Team Member Contributions



Amelia Rawlings <amelia.rawlings@gmail.com>

Wed 8/8/2018 1:49 AM

To: ● Albert Du

Cc: Jiadong Yu; BettyReaney@gmail.com; Dalu.Li@hotmail.com; Keawa.Rozet@yahoo.com; RayCheong820@gmail.com



↻ Reply all | ▼

Good Evening,

I, Amelia Rawlings, have had the pleasure of being Team Lead for Team05.

As team lead, the majority of my contributions had to do with paperwork and team management. I organized team meetings so that we could all get on the same page. I also facilitated of-site meetings between the Leads to overcome major hurdles.

With respect to code, I created the Sign-In page as well as spent many hours converting the Search.js and SearchBar.js files to connect them to the back-end.

I less than 50 of my commits ended up in the end product, but I made sure that only the commits that were value-adding were merged.

I have really appreciated the experience of working with this team and hope to work with them again some day.

Sincerely,

Amelia Rawlings
Team 05 - Lead

Keawa Rozet (Backend Lead)



Keawa Rozet <keawa.rozet@yahoo.com>

Tue 8/7/2018 11:26 AM

To: ● Albert Du; Dalu.Li@hotmail.com; BettyReaney@gmail.com; RayCheong820@gmail.com; amelia.rawlings@gmail.com; Jiadong Yu ✉

👍 ↻ Reply all | ▼

Hello,

This is Keawa Rozet, the Backend Team Lead of Team05.

I was heavily involved in all stages of this project while actively designing our software architecture, producing well documented code, being a constant contributor in relevant conversations on slack, and proving clear goals and directions for my Backend team.

Hard Contributions to the project: creating our AWS server, setting up Nginx as a reverse proxy with SSL certification, linking our GitHub repo as our means of code deployment, being the sole handler of code deployment to the remote server, adding useful and readable code to all files involving anything incorporating the Backend, QA testing of all routes using Postman, managing MongoDB database, and fixing merge conflicts.

Soft Contributions to the project: using Trello to provide clear tasks and deadlines, being an active member on Slack where my team's specific channel, #backend, has over 1000 messages posted, learning how to use specific software ahead of time and providing my team with detailed instructions on how to use and recreate my steps, building interpersonal relationships with the majority of the team members, consistently attending in-person meetings outside of school, acting as a medium of communication between Backend and Frontend where they overlap, and proactively pushing for tasks to be completed.

At the time of writing this email, I am responsible for **276 commits** out of the total 575 commits made on the development branch.

As the Backend Lead, most of my duties separate from being a Backend SW are largely wrapped up in the Soft Contributions section listed above. My other duties as the Backend Lead come in the form of QA testing, code review for pull requests, and managing the server.

I have thoroughly enjoyed my time as Backend Lead and I largely attribute that to the **wonderful team members** I have had the pleasure to work with. I have learned more in this experience than I have in any other class I have taken at SFSU, and I am very grateful for that.

Best,

Keawa Rozet

Arnold Jiadong Yu (Frontend Lead)

CSC 648-848 Milestone 5: Team Member Contributions



Jiadong Yu

Wed 8/8/2018 1:01 AM

To: Ray Cheong <raycheong820@gmail.com>; Dalu Li <d.li@wdcine.com>; ● Albert Du;
Amelia Rawlings <amelia.rawlings@gmail.com>; BettyReaney@gmail.com; Keawa.Rozet@yahoo.com ✉



Reply all | ▼

Hello,

This is Arnold Jiadong Yu, the Frontend Lead of Team 05.

I was heavily involved in all stage of mockups, prototype, connecting frontend and backend, react coding as well as providing goals, tasks and directions for Frontend Team.

1. Design and code About Page (M0)
2. Personas and Use cases (M1)
3. Set up a meeting with the frontend team to finish the mockup design.
4. Remodify the mockup design received from other team members to keep the design consistent. (M2)
5. Prototype on Figma. (Design)
6. Navigation Bar, Homepage, most of the search bar, upload photo feature, delete user feature. Part of Sign in align and validation. (Coding)
7. Communication with backend lead in solving connection problems. (Coding)
8. Merge conflicts between frontend members. (Coding)
9. Wrote down part of comments Professor gave and sent to the group. (M3)
10. Usability test and QA test. (M4)
11. This contribution (M5)

At the time of writing this email, I am responsible for 185 commits for total 585 commits on the development branch as well as overall 69,646 ++ 52,833 – on GitHub insight contributors.

I have learned more from this experience of team and group work than any other classes I have taken.

Thank you all!

Best,

Arnold Jiadong Yu

Albert Du (Backend)

CSC 648-848 Milestone 5: Team Member Contributions



Albert Du

Wed 8/8/2018 2:17 AM

To: Amelia Rawlings <amelia.rawlings@gmail.com>; Dalu.Li@hotmail.com; RayCheong820@gmail.com; BettyReaney@gmail.com; Jiadong Yu; Keawa.Rozet@yahoo.com ✉



Reply all | ▾

Hello everyone,

This is Albert Du, a member of Team 5's backend team. Here are my contributions to the team project:

Technical Contributions:

1. Wrote the data definitions for Milestones 1 and 2. Many of these definitions became backend variable names, class names, or MongoDB schema fields.
2. Set up remote connection to our database
3. Contributed code to our uploaded image router, user account router, user account schema, and uploaded image schema
4. Tested most routes using Postman
5. Wrote code to convert images to thumbnails
6. Implemented password encryption and input validation
7. Reviewed code & comments.

Non-technical Contributions:

1. Wrote Milestone 3 Feedback Report
2. Got team started on Milestone 4 and 5 doc when team was having trouble
3. Compiled Milestone 4 and 5 documents
4. Active on Slack over the duration of this project
5. Proposed creating our first Trello board with backend lead Keawa Rozet

Best Regards

Albert Du



Albert Du

Wed 8/8/2018 2:22 AM

To: Amelia Rawlings <amelia.rawlings@gmail.com>; Dalu.Li@hotmail.com; RayCheong820@gmail.com; BettyReaney@gmail.com; Jiadong Yu; Keawa.Rozet@yahoo.com ✉



Reply all | ▾

Additional Note: As of this email, I have 40 contributions on the development branch

U Keong Cheong (Ray Cheong) (Backend)



Ray Cheong <raycheong820@gmail.com>

Wed 8/8/2018 12:43 AM

To: Dalu Li <d.li@wdcine.com>; Albert Du; Amelia Rawlings <amelia.rawlings@gmail.com>; Jiadong Yu; BettyReaney@gmail.com; Keawa.Rozet@yahoo.com ↗



Reply all | ▾

(Please discard the last email)

Hi everyone,

This is U Keong Cheong (Ray), the back-end team member of Team 5.

Here are my contributions to the team project:

1. Wrote Competitive Analysis in Milestone 1.
2. Contributed to defining DB organization and UML for image collection in Milestone 2.
3. Help finding resources for setting up the remote connection to our database.
4. Looked into MySQL as a backup plan while other team members were trying to get MongoDB to work.
5. Contributed code to uploading images to database, getting single image from database and display it on the browser.

Sincerely

Ray

Dalu Li (Damon Li) (Frontend)



Dalu Li <d.li@wdcine.com>

Tue 8/7/2018 10:00 PM

To: ● Albert Du

Cc: Amelia Rawlings <amelia.rawlings@gmail.com>; Jiadong Yu; BettyReaney@gmail.com; RayCheong820@gmail.com; Keawa.Rozet@yahoo.com ✉



Reply all | ▼

Hi,

This is Dalu Li, the front-end team member of Team 5.

Here are my contributions to the team project:

1. search approve
2. search results
3. search results popup and download
4. admin photo approve and decline photos
5. part of search bar function
6. debug the search bar and admin function

Sincerely

Dalu Li

Note: Damon has 9 commits on the development branch, with 10000++ and 6200-- lines of code.

Betty Reaney (Frontend)

CSC 648-848 Milestone 5: Team Member Contributions



Betty Reaney <bettyreaney@gmail.com>

Wed 8/8/2018 10:55 AM

To: Keawa Rozet <keawa.rozet@yahoo.com>

Cc: Albert Du; Dalu.Li@hotmail.com; RayCheong820@gmail.com; amelia.rawlings@gmail.com; Jiadong Yu



Reply all | v

Hello,

My contributions as a front end team member:

- Wrote Executive Summary
- Connected Sign In and Join us buttons on the Nav Bar to at first be popover windows
- Added 3 js files that created these buttons and coded their functionality.
- Created Sign In that included basic functionality
- Created Join in which includes various fields of input
- Removed drop down bar and realigned the search bar
- Hard coded the drop down menu to have the categories appear when clicked

Thank you!

Betty Reaney

Team Lead Post Analysis

As we come to the end of the semester we noticed we learned many lessons. This has been a great project developed by a great team, but as with everything there is always room for improvement. Be it doing additional research into tools to be used, learning a framework more before starting to use it, or knowing when, where, and how strongly to apply pressure to each other, there are plenty of things we wish we knew before that we know now.

When we first sat down to start developing HatBoxPhotos, we were all very ignorant about the tools available to help us develop the site. We did a little bit of research, but we didn't know exactly what we were looking for in a product. Therefore, we decided on Amazon's Web Services, AWS, to hold our server because it is very well known in the tech community. Some even considered it to be an industry standard. This was a poor decision on our part. We should have done more research to find a more user friendly hosting service. Although our back-end lead, Keawa, did an amazing job with learning and managing our AWS, it was still a very steep learning curve, and we missed a few deadlines due to this. Lesson learned, next time we will do more research instead of assuming something is the best because it's popular.

Along the same lines as selecting AWS prematurely, we selected frameworks that we were not familiar with. This caused issues, and many hours were spent, needlessly, trying to fix the connection between the front-end and the back-end. The problem was that none of us had a strong foundation on any framework. Lesson learned, next time we will spend more time learning about frameworks before deciding which one to choose.

One of the more interesting lessons learned, is more of a soft skill. We did our best not to micromanage one another. However, that resulted in miscommunications. Requests were sometimes made from one person to another without enough details. This resulted in many requests being completed with less than satisfactory results. Lesson learned, being specific in what we ask of others is not the same as micromanaging.

Hindsight is 20/20. There will always be things that we wish we had known then that we know now. We are proud of the work that Team05 did and the things that we were able to accomplish in these few weeks. Even if we should have spent more time learning about the tools before making a decision, learning the frameworks before deciding which one to use, and been a little more strict, clear and concise with our requests and communication with one another.