

Final Project for SW Engineering CSC648-848 Fall 2018

Gator Exchange

Team 11

Matthew Ho: mho1@mail.sfsu.edu

Jason Chow

Dian Zhu

Riza Shrestha

Justin Shee

Ghulam Khan

Juan Valdez

“Milestone 5”

December 18 2018

<https://gator-exchange.netlify.com/>

2). Product Summary:

Gator Exchange is a website that serves as a buy and sell hub that caters towards students at San Francisco State University. Our website is unique in its ability to allow students to find what they need and sell what they want with ease of use. Postings on *Gator Exchange* are a great way to showcase items by student sellers, and provide an easy way of contact for potential buyers.

I. All Users

- A. All users are able to view recent postings
- B. All users are able to register to create an account which is used for contacting sellers and making new item posts.
- C. All users are able to search through items by category or name.

II. Registered Users

- A. Registered users are allowed to contact sellers to inquire about item postings
- B. Registered users are presented with a new posting form which allows them to create a new listing that includes a description, title, image, and price of the item.
- C. Registered users have access to a seller dashboard which contains a list of postings they've made and comments that have been made on their posts.
- D. Registered users can delete or modify their item listings

III. Administrators

- A. Administrators can browse the website and have the same functionality as a registered user.
- B. Administrators have their own admin dashboard which contains a list of postings that are pending for approval.
- C. Administrators can approve or disapprove postings.

URL: <https://gator-exchange.netlify.com/>

3). Milestones

SW Engineering CSC 648/848 Fall 2018

Gator Exchange

Team 11

Matt Ho
Justin Shee
Ghulam Khan
Riza Shrestha
Juan Valdez
Jason Chow
Dian Zhu

MileStone 1

9/26/18

1. Executive Summary:

Living as a student is not a walk in a park. As you may or may not know, being a student is heavy task due to an inconsistent schedule, dealing with huge workloads, and most importantly lack of financial stability. On top of tuition, students find themselves losing a considerable amount of money on living costs, books, and other study materials. As semesters finish, and living situations change, students are finding themselves funnelling money consistently changing expenses. What if there was a service out there that allows students to buy items they need at a cheaper price and sell things they don't need anymore? Introducing Gator Exchange.

Gator Exchange is the premier buy and sell website which features an array of features catering toward students. Students at SFSU are offered a unique and special place to advertise, sell, and buy goods from other students. Gator Exchange features a dynamic front page that features a what's hot section, which students could market their items on for a fee. Gator Exchange allows for sellers to upload a picture and description of what they're selling and buyers can browse through these items and chat with buyers in order to set up meet ups. Gator Exchange also features a customer review system and a way for SFSU students to verify themselves in order to maintain high level security. With these features, students will finally be able to have a way to advertise their products, and buy from trusted sellers from their own school.

Gator Exchange is developed by a startup team of seven passionate students at SFSU that understand the difficulties that students have to go through on a day to day basis. We are passionate about improving the student experience at SFSU and we find that finances are one of the biggest issues that we constantly deal with here. With Gator Exchange, students worry less about finances, recycle their cash, and can take one stress out of their already stressful lives.

2. Persona and Use Cases:

a.) Customer-Seller: Joseph was a SFSU graduate who is still living around the area, and wants to get rid of some books and supplies that he doesn't need anymore. He works at a daycare and takes part time classes to supplement his jobs needs in child development, so he is a very busy person. He has basic experience when it comes to using computers and websites. Joseph first heads over to the website because he wants to post his textbook he is finished using for sale. He searches up the book on the website in order to figure out a general pricing of the book, in order to not overprice or underprice his item. Next, he posts his item and proceeds to input a name for the product, the category of the produce, an image of the product, a price for the product, and a description of the product. Afterwards he submits his post and is prompted to register because he doesn't have an account. Once he finishes registration his application is put up for review to be approved by an admin and will be notified once his item has been published. The notification is sent that his post does not contain anything in the way of some inappropriate post, and when checking his dashboard he notices that his post is put under approved.

b.) Customer-Buyer: Lauren is a student who attends class overseas in another country but during her breaks she comes back to San Francisco to be with her family. At times she would like to quickly and cheaply acquire some books or some school supplies so as to be prepared when going back. Throughout her time with her family, she is very busy and just wants to quickly acquire some items without having to take the time to head over to a local stationary store. She has all the basic knowledge when it comes to computers and online shopping. Being told about the website she heads over to take a look at a chance to quickly restock on notebooks and pens. She heads over to the notebooks categories and begins to scroll through the available notebooks up for posting. After clicking on the post Lauren is able to read the description, checkout some photos of the item, and decides this is the one she wants. With her decision, she contacts the person through the website which leads her to a messaging page where she can direct message the seller through the website and see if the item is still for sale and figure out how the transaction can be done.

c.) Admin: Nelson was hired to be an admin for the buy and sell website, and has gone through some training to understand how to administer the website. He has been through many different jobs regarding computer and administration work, so he is very well skilled when it comes to this type of job. After a quick week of training he is good to go with his new job. The website just launched and he does not have very many posts that he has to go through everyday, but one person decided that it would be a good idea to play a joke and submitted a picture that is not fit for viewing as well as a title and description that has curse words that Nelson hasn't even heard before. He goes through the post disapproval process which will reject the posting because it

goes against user policies. Then he acquires the users email in order to send a email stating that the post was rejected due to a reason, and letting them know they will be banned for what they have posted. He then proceeds to go to the users profile and temporarily bans him for the offense, and if the person decides to continue with the misbehavior they will eventually get permanently banned.

d.) Customer - Buyer: Bill a standard San Francisco State University student who is looking to get some cheap used textbooks, but unfortunately he had some problems and was unable to get his hands on books that he would need for the semester. He was told about this website and decides to go have a quick look in hopes that some books that he needed would be sold by some other students. Having standard computer knowledge like any other student going to San Francisco State University in 2018, he begins to browse the online store through class code and is able to find the textbook he is looking for and for a fairly cheap price. He clicks on the item to and goes through the direct message process the person to get in contact and tells him that he is interested and want to set up a meeting point for them to make the trade.

e.) Customer - Seller: Julie is a very popular seller on Gator Exchange and has a thorough understanding of technology and using the website. She holds the largest share of items sold on the website and has a very good reputation amongst the buyers. Being that she is a school student, her life is very busy and she needs to get the postings up very quickly. Today she wants to post up a textbook, a computer mouse, and a mini laptop desk. So she heads over to her dashboard where it shows all of the items that are still up and items which people have messaged her that they want to buy. She decides it would be better of her time to get back to the buyers and post up the items later. She starts by answering to the posting which sends her to a screen where she is able to contact the person. Afterwards, while waiting to answer to a reply, she checks out the other postings and messages them as well. Finishing up with responding to everyone, she then proceeds to delete all of the items which she has already sold. This was done very quickly and she begins to post up her new merchandise before still having to make it to her lunch appointment.

3. Data Definitions:

Registered User: These users are users

who are registered to our site, they have a login name, and a password that allows them to do things Unregistered Users cannot. Registered Users can read and view posts, comment or contact a seller, post a selling item, and login.

Unregistered User: are users who do not

have a login name or password that allows them to do somethings registered users can. This type of user will have the ability to read and view posts, fill out posting information, but not be able to post until signing up for the website. Will have the ability to begin posting, however when they go to submit they will be prompted to login or register to post the item.

Admins: are the people in charge of managing posts and making sure a post is appropriate for the website. They can do everything Registered Users can do, as well as accepting posts, deleting posts, banning users, and deleting users.

Post function: Will allow registered users to post items that they want to sell. If a user is logged in the user will have a window where he or she will insert title of the post, add image or images, put price, optional description, and contact information.

Registration- This feature will allow an Unregistered User to register. If nobody is logged in and a user is about to post, contact a seller, or etc. a registration will prompt up at the last possible moment.

Search- A search field will be present on top of our dashboard page next to a category selection option. This function will allow Registered or Unregistered Users to search items categorically. Once the search button has been clicked, the items closes fitting the category and words put in by

4. Functional Requirements:

I. User (Unregistered) functional requirements

- 1 - Unregistered users shall be able to search using item type categories
- 1 - Unregistered users shall be able to view items per page up to 48
- 1 - Unregistered users shall be able to view postings
- 1 - Unregistered users shall be able view website and items without an account
- 1 - Unregistered users shall create an account to sell or buy
- 2 - Unregistered users shall be able to view sellers' basic information (Name, General area of preferred transaction)
- 2 - Unregistered users shall be able to search for products with search bar

II. User (Registered) functional requirements

- 1 - Registered users shall be able to communicate between buyer and seller
- 1 - Registered users shall be able to create a selling post
- 1 - Registered users shall be able to view their account, includes order history, current selling, and profile information (account dashboard)
- 1 - Registered users shall be able to buy item, by agreeing to price and setup with buyer for transaction place and time
- 1 - Registered users posting shall not be altered by other entities
- 1 - Registered users shall agree to terms and conditions before posting
- 1 - Registered users shall follow guidelines or user's post shall be disabled from being public
- 1 - Registered users shall be able to edit an existing post
- 1 - Registered users shall be able to upload a maximum of 5 photos in the post
- 1 - Registered users shall add a title, description, category, and price in the post
- 2 - Registered users additional information shall be able to revealed by user after agreed transaction (Preferred contact information, and meet up location)
- 3 - Registered users shall be able to comment on the post in the comments section

IV. Admin functional requirements

- 1 - Admins shall be able to disable postings that violate guidelines
- 1 - Admins shall be able to monitor any postings
- 1 - Admins shall be able to access all data and content and modify the database

- 1 - Admins shall approve or deny registered users' post before publication
- 2 - Admins shall be able to ban users who violate guidelines

5. Non-functional Requirements:

High Level specifications:

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).
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. Selected application functions must render well on mobile devices
4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
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-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only*" at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive Analysis:

Categories	Ebay	Craigslist	Touch Of Modern	Gator Exchange
Search and Filter	✓	✓	✓	✓
Prices	✓	✓	✓	✓
Display pictures by categories	✓	✓	✓	✓
Pictures Caption/Description	✓	✓	✓	✓
Upload Pictures	✓	✓	✓	✓
Shopping Cart	✓	✓	✓	

✓ = Feature exists

The top 3 websites that control most of the market have a lot of features that allows them to compete, but they have one big down fall is their pricing is very expensive which is good for us because we can drive a lot of user to our platform by providing students to buy and sell with a reasonable price until we have a strong user base. Also, we can compete with them on launch by having most of their features like Search, registration, Organize pictures by categories, upload pictures option and also adding items in the shopping cart. Furthermore, when going to these websites that don't show any pictures on their homepage, but they show categories. We can do top 20 sales of the week on our main page which will incentive more sellers to our website to have their items featured by us on the main page.

7. High-Level System Architecture:

This following section mentions all the tools, frameworks, and APIs being used in the development of this project.

Our application will be developed following these environments that consists of the following:

- **Linux**
 - i. We are deploying our application on the Ubuntu 16.04 Linux platform.
 - ii. Hosted on Heroku Paas 512MB ram, the cloud-computing platform.
- **Web Server**
 - i. Our application will be served through the ExpressJS 4.16.3 web server to allow users to access our application over the web.
- **Database**
 - i. The database backend will be managed with MongoDB 4.0 to store information.
- **Server-Side Language**
 - i. Using JavaScript language allows us to create a very dynamic and content-driven web application.

Frameworks and APIs:

As for the front-end and back-end of our application, we'll be utilizing the following frameworks to assist in our development:

- Bootstrap
- Express
- Node JS
- React

Supported Browsers:

Our application shall be able to run smoothly in the following browsers:

- Google Chrome: Version 69.0.3497.100, 68.0.3440
- Mozilla Firefox: Version 62.0.2, 61.0.2

8. Team:

Matt Ho - Team Lead

Jason Chow - Developer

Justin Shee - Back-end Lead

Riza Shrestha - Developer

Ghulam Khan - Front-end Lead

Juan Valdez - Developer

Dian Zhu - Developer

9. Checklist:

1. Team found a time slot to meet outside of class : **Wednesdays at 2pm.**
2. Github Master: **Matt Ho**
3. Team decided and agreed together on using the listed SW tools and deployment server:
DONE
4. Team ready and able to use the chosen back and front-end frameworks and those who need to learn it: **ON TRACK**
5. Team lead ensured that all team members read the final M1 and agree/understand it before submission: **DONE**

SW Engineering CSC648/848 Fall 2018 Milestone 2

Gator Exchange

Team 11

Matthew Ho

Justin Shee

Jason Chow

Dian Zhu

Riza Shrestha

Juan Alvarez

Mustafa Khan

Team Lead Email: mtthwbh@gmail.com

October 18 2018

Version	Revision Date	Description
1.0	October, 18 2018	Submitted for review
2.0	October, 22 2018	Revised according to feedback

1.Data Definitions V2:

Registered User: These users are users

who are registered to our site, they have a login name, and a password that allows them to do things Unregistered Users cannot. Registered Users can read and view posts, comment or contact a seller, post a selling item, and login.

Unregistered User: are users who do not

have a login name or password that allows them to do somethings registered users can. This type of user will have the ability to read and view posts, fill out posting information, but not be able to post until signing up for the website. Will have the ability to begin posting, however when they go to submit they will be prompted to login or register to post the item.

Admins: are the people in charge of managing posts and making sure a post is appropriate for the website. They can do everything Registered Users can do, as well as accepting posts, deleting posts, banning users, and deleting Users.

Post function: Will allow registered users to post items that they want to sell. If a user is logged in the user will have a window where he or she will insert title of the post, add image or images, put price, optional description, and contact information.

Registration- This feature will allow an Unregistered User to register. If nobody is logged in and a user is about to post, contact a seller, or etc. a registration will prompt up at the last possible moment.

Search- A search field will be present on top of our dashboard page next to a category selection option. This function will allow Registered or Unregistered Users to search items categorically. Once the search button has been clicked, the items closes fitting the category and words put in by

Contact Seller- A button that anyone can press in order to be taken to the posts information that a Registered user has posted and added their contact information.

2. Functional Requirements V2:

I. User (Unregistered) functional requirements

#	priority	requirement
1.01	1	Unregistered users shall be able to search using item type categories
1.02	1	Unregistered users shall be able to view items per page up to 48
1.03	1	Unregistered users shall be able to view postings
1.04	1	Unregistered users shall be able view website and items without an account
1.05	1	Unregistered users shall create an account to sell or buy
1.06	2	Unregistered users shall be able to view sellers' basic information (Name, General area of preferred transaction)
1.07	2	Unregistered users shall be able to search for products with search bar

II. User (Registered) functional requirements

#	priority	requirement
2.01	1	Registered users shall be able to communicate between buyer and seller
2.02	1	Registered users shall be able to create a selling post
2.03	1	Registered users shall be able to view their account, includes order history, current selling, and profile information (account dashboard)
2.04	1	Registered users shall be able to buy item, by agreeing to price and setup with buyer for transaction place and time
2.05	1	Registered users posting shall not be altered by other entities
2.06	1	Registered users shall agree to terms and conditions before posting
2.07	1	Registered users shall follow guidelines or user's post shall be disabled from being public
2.08	1	Registered users shall be able to edit an existing post
2.09	1	Registered users shall be able to upload a maximum of 5 photos in the post
2.10	1	Registered users shall add a title, description, category, and price in the post
2.11	2	Registered users additional information shall be able to revealed by user after agreed transaction (Preferred contact information, and meet up location)

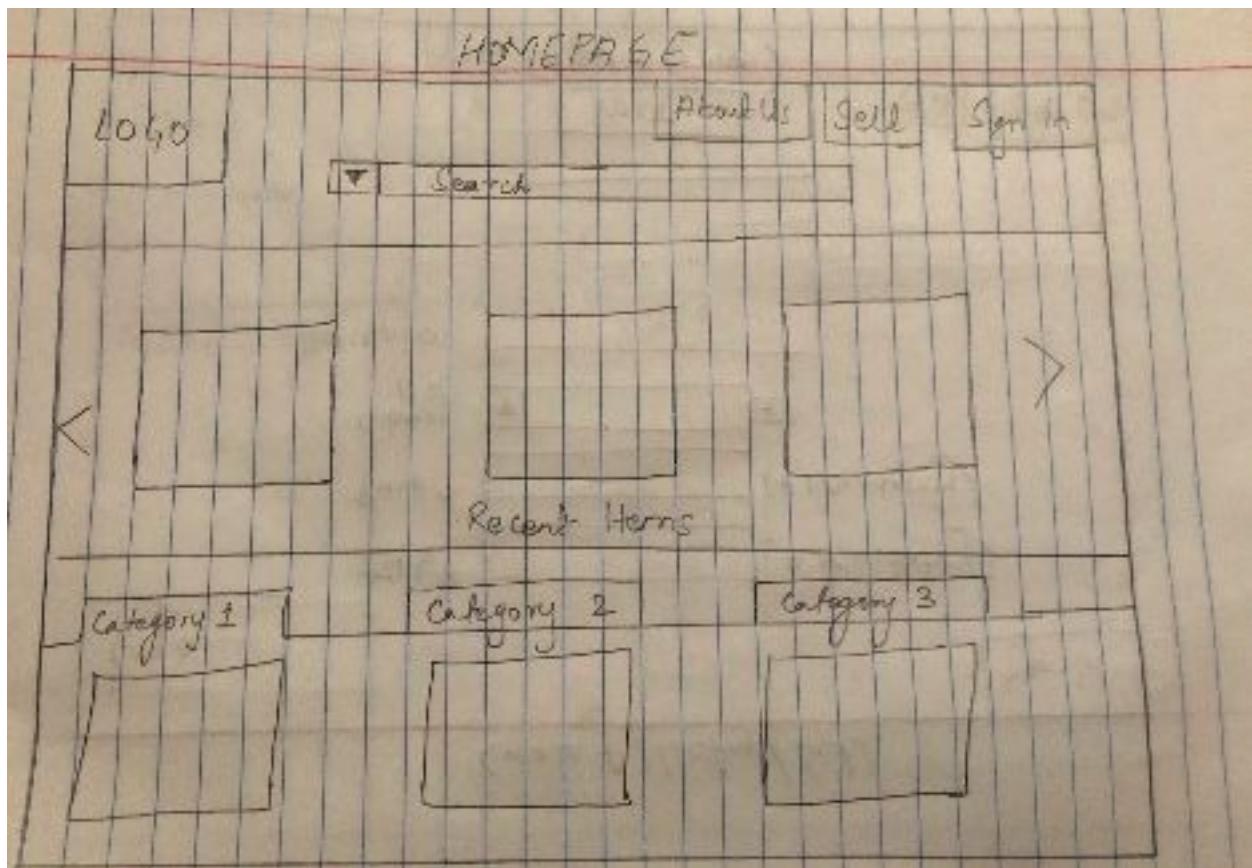
2.12	3	Registered users shall be able to comment on the post in the comments section
------	---	---

III. Admin functional requirements

#	priority	requirement
3.01	1	Admins shall be able to disable postings that violate guidelines
3.02	1	Admins shall be able to monitor any postings
3.03	1	Admins shall be able to access all data and content and modify the database
3.04	1	Admins shall approve or deny registered users' post before publication
3.05	2	Admins shall be able to ban users who violate guidelines

3. UI MockUp and Storyboard (high level Only)

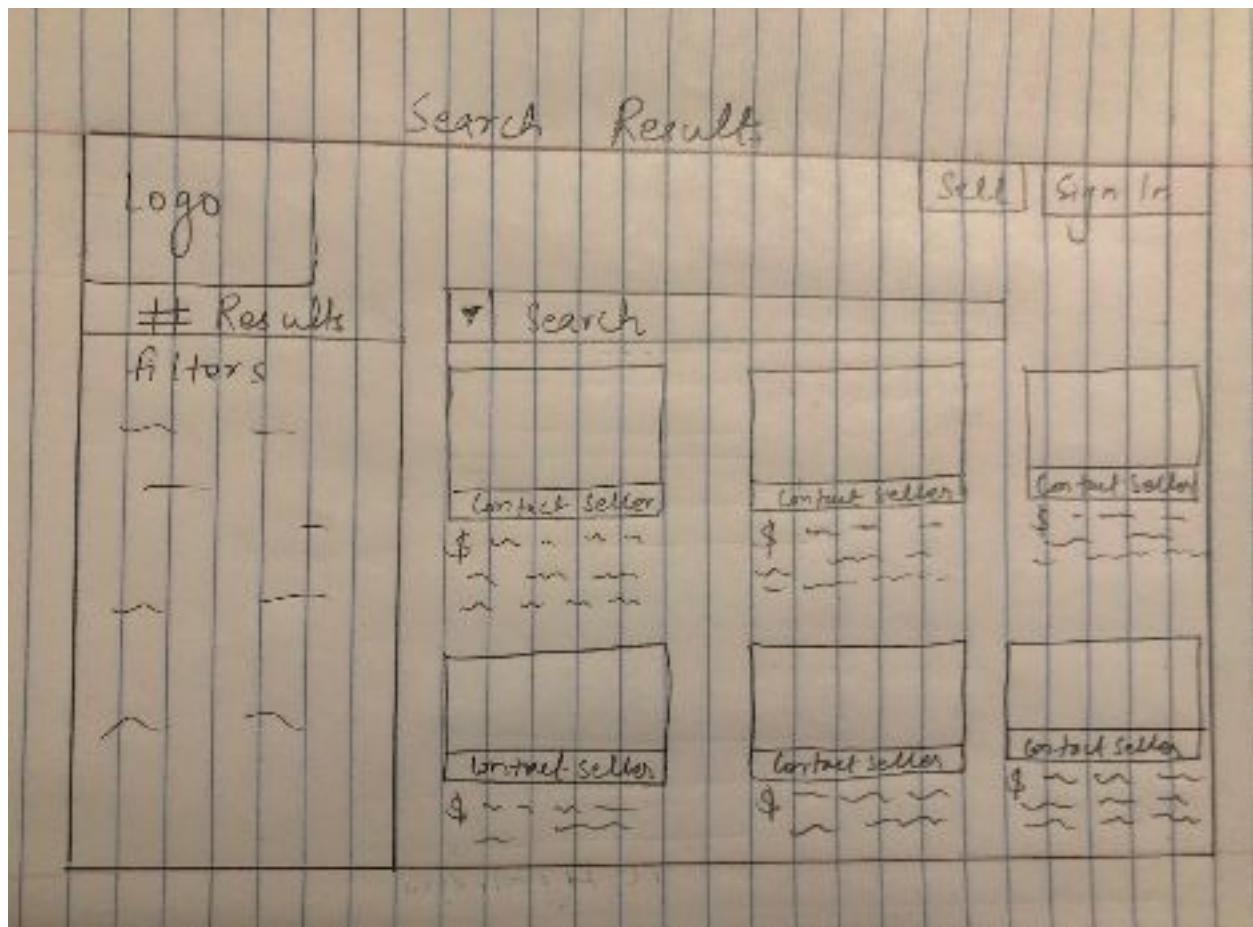
A. Home Page



- The homepage is the landing page for a user.
- The homepage features a navigation bar that includes a home button logo titled GatorExchange.
- Users have the option to click on sell; however it is exclusive to registered users, so a registration form will appear.
- Users have access to a search bar which navigates through all items.
- The search bar has a category selector which can narrow down the search
- Users are presented with a dashboard of recent items underneath the search bar.

- Underneath the dashboard are a list of items currently on the website and are separated by category.
- Items are shown in a tab style.
- Items can be clicked on, and viewed by any user. There are different options for a registered and unregistered user.

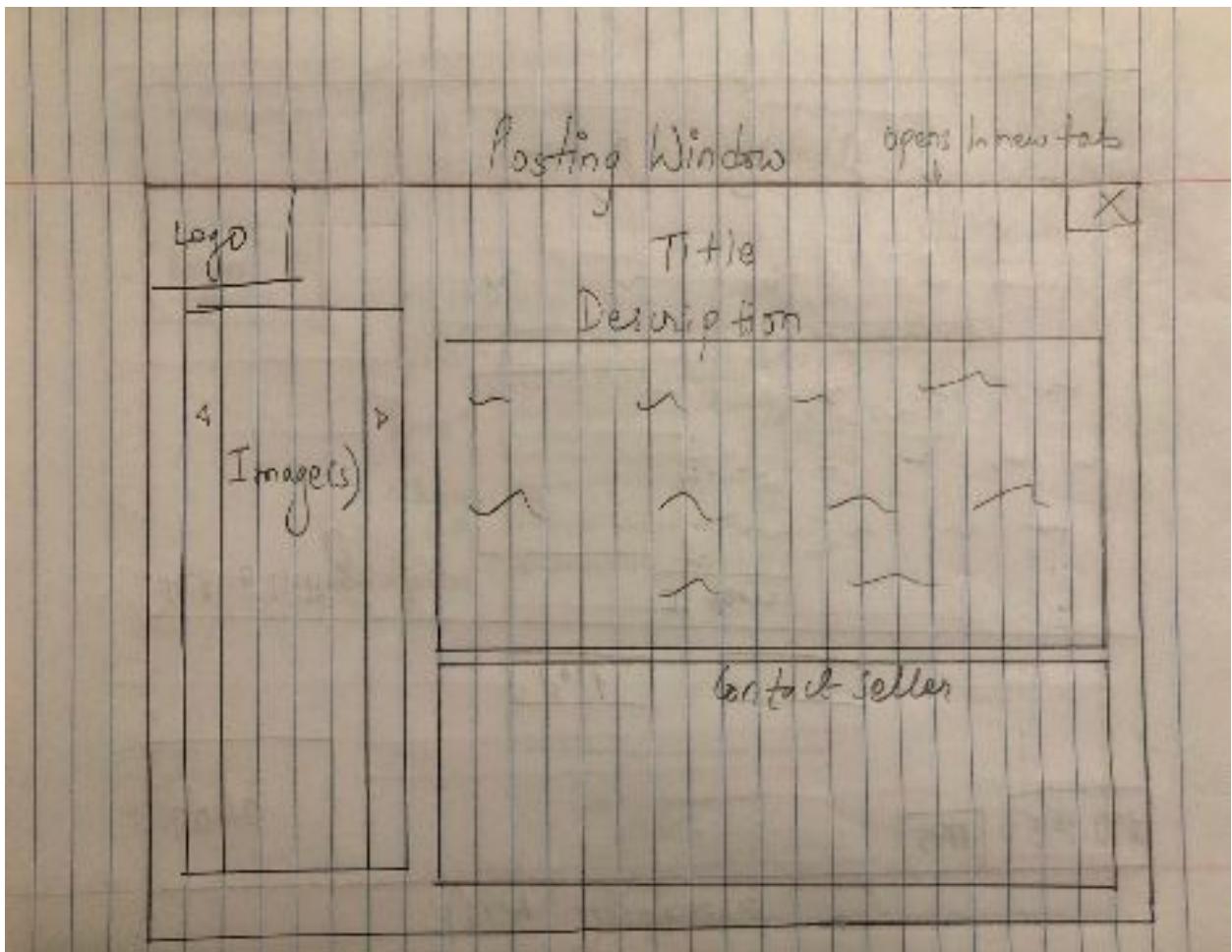
B. Search Results



- After a User searches, the user is presented with a search results page.
- The search results page has a list of results of items
- These items include an image, a description, and a price.
- Items are clickable and a new window will show up accordingly.
- Items are displayed based on category specified.
- Items are displayed on the right portion of the page.
- The left portion of the page is going to be a list of filters.
- Filters allow the user to further narrow down their search.

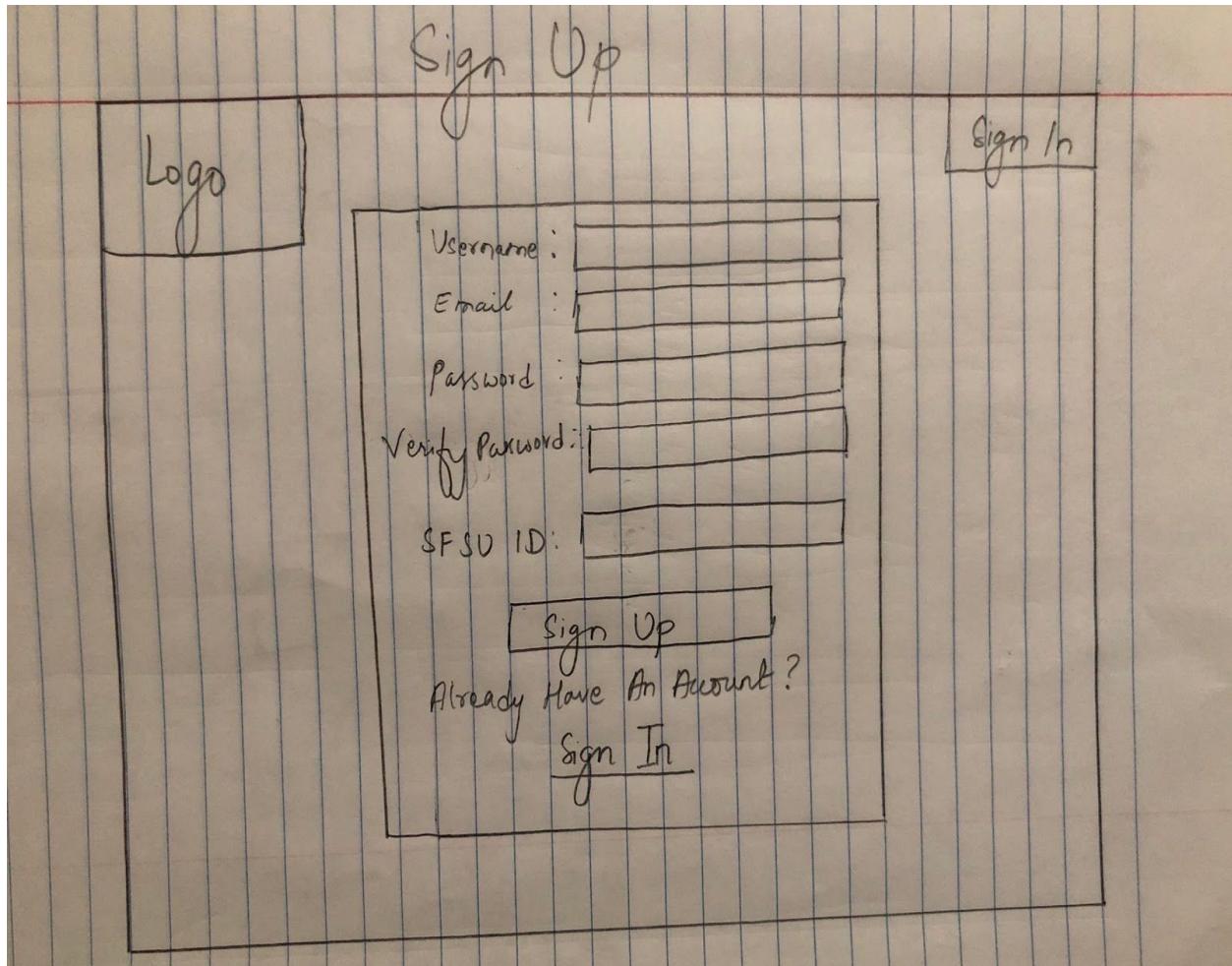
- On top of the filters area, is a number indicating how many results have been found.
- Users have the option to return to the homepage through the navigation bar on the top of the page.
- Users have the option to search again

C. Posting



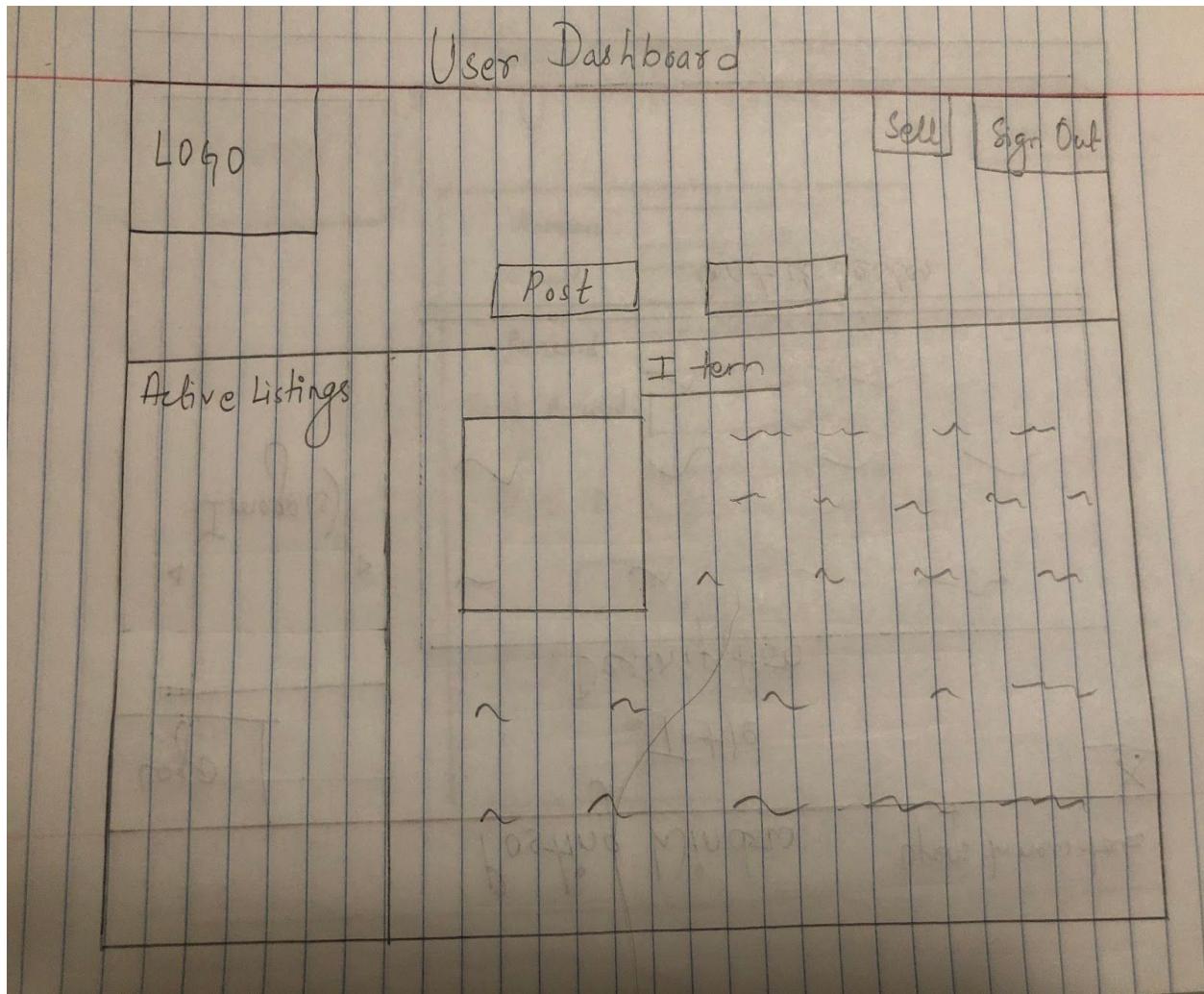
- This is the posting window that pops up for a user if they click on an item.
- The item's gallery will be shown on the left. It consists of user uploaded images of their item.
- There will be an area for a description of the item being sold.
- Following the description, there is an area for a registered user to contact the seller.
- Users can contact the seller by providing a message and their email information.

D. Sign Up



- The register form is an overlay form that will pop up over the current page on the website.
- Users can use this form to register here.
- There will be mandatory forms, being username, email, and password.
- There is an option for existing users to sign in.

E. User Dashboard



- A registered users dashboard shows up to a user who signs in.
- The registered user has a large area on the right which displays their previously viewed items.
- There will be a section on the left for active listings that they have posted themselves.

- Registered users will have an option to post an item through their dashboard.
- Registered users can also search the website through their dashboard

F. Sell/ Create a Listing

Create A Listing/Sell

<i>Logo</i>	<i>Title</i> *	(* Mandatory)
	<i>Price</i> *	(* Mandatory)
	<i>Category</i>	<input type="button" value="▼"/> <input type="button" value="+"/>
<i>Description</i>		
<input type="button" value="POST"/> <i>(Effective Time)</i>		

- This is the page for a user to create a listing.
- This page is navigated to through the sell button on the navigation bar.
- Listings have mandatory fields title, price, and category.
- Category is a drop down menu which displays a list of categories a user can pick from
- A description form is also provided.
- The post button serves two purposes. A registered user can use this to make a new post. An unregistered user will be prompted to either sign in or register.

4. DB Organization

The main DB tables will include:

- Users - which will allow users to login and register
 - Username: String
 - Password: String
 - (PRIMARY) UserId: Integer
 - numOfPosts: Integer
 - pendingNumOfPosts: Integer
 - approvedNumOfPosts: Integer
- Admin - which is the same as a User but categorized differently as not to mix, will probably not allow them to buy or sell.
 - Username: String
 - Password: String
 - (PRIMARY) AdminId: Integer
 - numOfPostsWaiting: Integer
 - numOfPostsApproved: Integer
- Posts - this way the items that are posted will be logged and kept track
 - (FOREIGN) UserId: Integer
 - (PRIMARY) PostId: Integer
 - Title: String
 - Description: String
 - ImageAddress: String or Image:Blob
 - Price: Float
 - Category: String
 - Status: Boolean
- Messages - log users that have attempted to reach each other and allows the messages to be saved for future use.
 - (FOREIGN) UserSellerID: Integer
 - (FOREIGN) UserBuyerID: Integer
 - (PRIMARY) MessageID: Integer
 - Log: String

Media Storage

Media will be kept in storage via MongoLab.

Search/filter architecture and implementation:

Posts from the user will be split up into categories, and each item will be marked with a category as marked in the Posts DB Table. Users will be able to choose from a simple drop down category menu in order to narrow down their search.

A secondary search bar will also be provided which should allow the user to type in exactly what they are looking for possibly through the use of SQL's select command and %like. This way if the user doesn't fully type what they want into search %like will be able to pop with results that are similar. Possible implementation of allowing users to input tags for their merchandise that, despite not being in the title if the tag is searched for that item will show up as well.

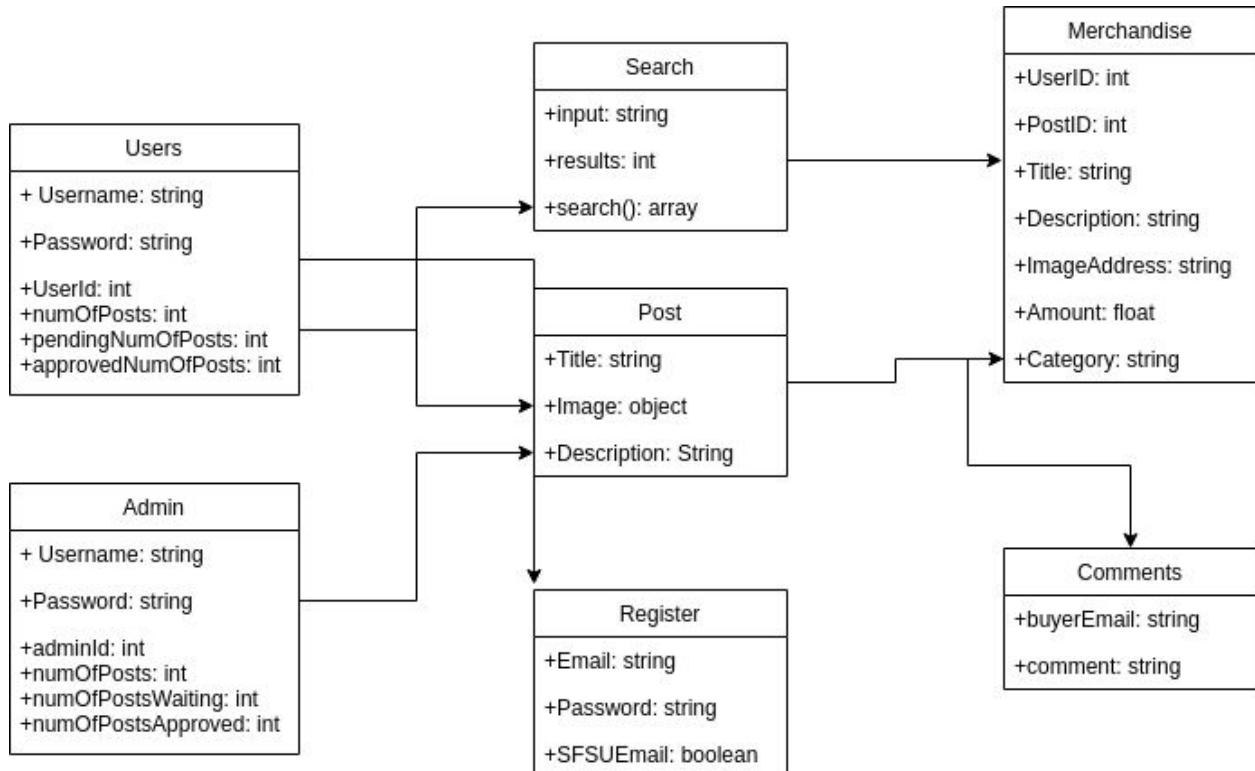
Both search features can also be used together for the user to minimize the range in order to find what they are looking for, but for early tests using both may be excessive.

Processing Algorithm

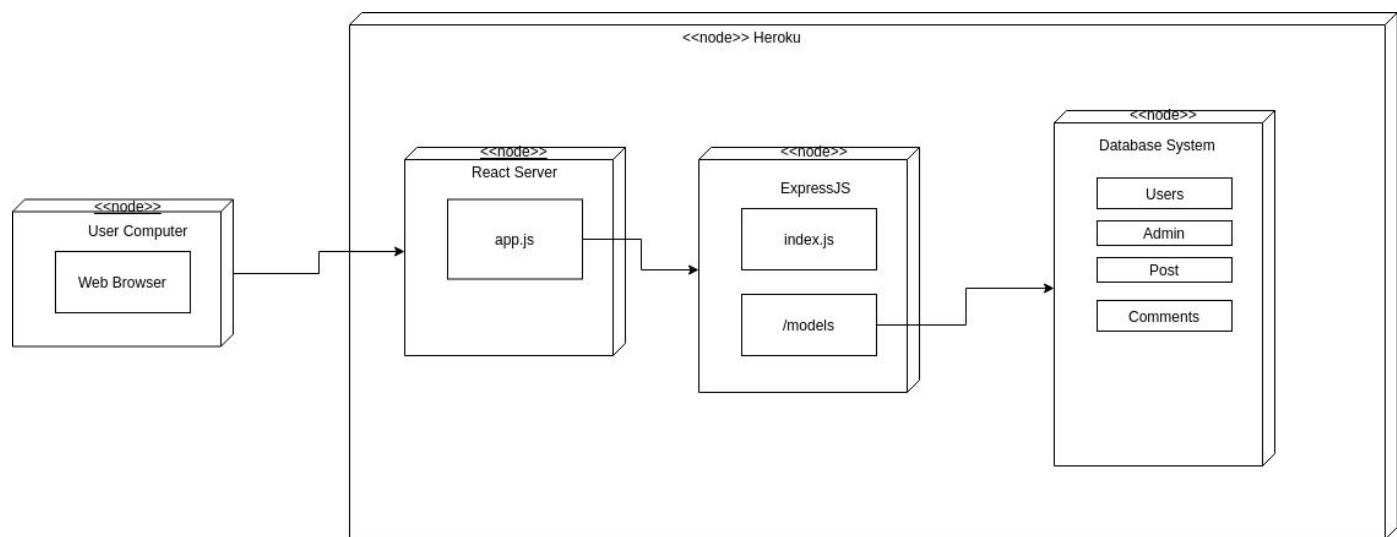
We will use SQL for category and %like for text to search for items.

5. UML Diagrams

High-level UML class diagrams



UML Component and Deployment Diagram:



6. Key Risks:

- **Skill Risks:**

React

Most of the team members are not familiar with react and have not worked with it. Learning React is very important mostly to the front-end team as all our website design will be done through it. To learn React all the members were provided with full course on it. Also, other online resources like w3school.com were used to clarify any doubts.

Mongo DB

Most of the team members have never worked with MongoDB. Learning from the tutorial online members was able to understand it. Since, designing a schema is a vital task inexperienced members were guided by experienced members to build it.

Node JS

Some of the members of the team had worked with it before but rest who did not have familiarity with it was provided tutorials to learn. Also any doubt or difficulty understanding any topic were cleared by experienced members.

Github:

Most of the team members were familiar with GitHub. But in order to prevent crashing of whole site due to a small code error, particular person was given responsibility for pushing everything to the master.

- **Schedule Risks:**

Work division and time are very important factors that help to create a good website. For the betterment of the team and keeping in consideration that members have work and other classes. So, we decided to meet every Wednesday afternoon for team meeting. This meeting is a scrum where we discuss work division, progress and future task to complete. If any member needed assistance than everyone would solve the issue together.

- **Technical Risks:**

The technical risks involved are that the website needs to open on different browsers but due to different visual elements, resolution and graphics. To ensure GatorExchange works on every browser, our team is going to keep the code clean and select styles that are cross platform compatible. Also outlining the workflow prior to starting work has been a key in ensuring we are all on the same page.

- **Team Risks:**

The front-end and back-end should always be in an agreement to produce a good result. Our team works very well together as members are aware about their role for each milestone. Also streamlined communication between backend, frontend and leads has helped ensure everyone is on the same page.

- **Legal Risks:**

When the GatorExchange website is deployed publically we could run into risk of users posting images that can violate the terms and conditions of the website. So, in order to prevent that we have Admin who will be able to monitor any posting, disable postings and ban users who violate our guidelines. Also we will be careful on ensuring we choose images that we are legally allowed to use.

SW Engineering CSC 648/848 Section 02 Fall 2018 HouseHunter Team 11

Milestone 3

Matthew Ho
Justin Shee
Jason Chow
Juan Valdez
Riza Shrestha
Dian Zhu

November 18th, 2018

(Instructor Live feedback)

Home page

- Buttons to buy items were not there
- Coloring could be better
- Alignment needs to be fixed
- Don't have texts as links
- Readability needs improvement
- Logo on left
- Login/Register needed

Search (including search field validation)

- Search bar was too long
- Make the categories aligned

Search results

- Add filtering
- Lazy registration

Filtering

- Filter by category
- Linking should only done through image
- Contact seller

(Instructor Offline Feedback)

* Unfortunately due to misinformation on domain name, feedback was not given offline for our website. Here is what was given and here was our response to the feedback.

Instructor to Check and comment below:

* **Git/Github organization** (e.g. organization of branches)

GH organization looks fine. Have some branches but not really a feature branch scheme or a person branch scheme.

* **Git/Github usage:** Comments on posting; Number of posting to github; Appr. even distribution of submissions among team members (check github post stats for all members)
Commit messages could be better, some commits are small and could be combined.

Commit counts :

52 Matt Ho

18 Matthew Ho

8 nagasnow

7 Jason Chow

4 dianzhu

3 csc648848Instr

1 Anthony

1 Anthony J Souza

1 Juan Valdez

1 Mustafa Khan

1 sthriza

Note the top 2 are the same person.

*** Code documented (header, in code) with good coding style**

Little to no comments

*** MVC/OO patterns followed up**

followed

*** Frameworks (back end front end) deployed correctly**

Deployable but website is down

Db access / instructions were not give

Team Lead Response:

Hi Anthony,

We just saw your feedback. There were a couple of concerns that I wanted to address. Firstly, the backend is the only thing deployed on heroku while the front-end is deployed on netlify at :
<https://gator-exchange.netlify.com/>

Database access can be accessed at mlab:

<https://mlab.com/login/>

Username: csc648team11

Password: 648team11

My apologies for not updating the google drive. All changes and access will reflect this email.

Let me know of any issues/concerns.

Best,

Matthew Ho

HOME PAGE

Logo

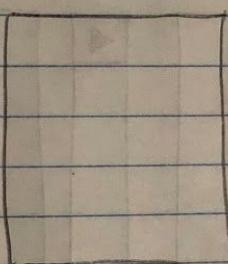
▼ Search

About Us

Sell

Sign In

<



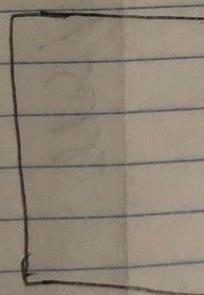
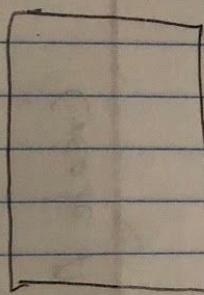
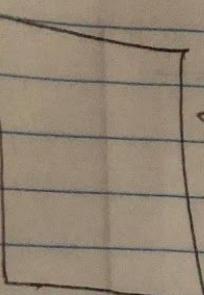
>

Recent Items

Category 1

Category 2

Category 3



HOME PAGE

LOGO

Search

About Us

Sell

Sign In

<

>

Recent Items

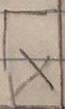
Category 1

Category 2

Category 3

Posting Window

opens in new tab



Logo

Title

Image(s)

Description

Contact seller

Create A Listing/Sell

Sign In

Logo

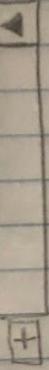
Title*

(* Mandatory)

Price*

(* Mandatory)

Category



Description

POST

(Effective Time)

Search Results

Logo

Results

filters

▼ Search

Sell

Sign In

Contact seller

Contact seller

Contact seller

Contact seller

\$ ~ ~ ~

\$ ~ ~ ~

Contact seller

Contact seller

Contact seller

Contact seller

\$ ~ ~ ~

User Dashboard

1040

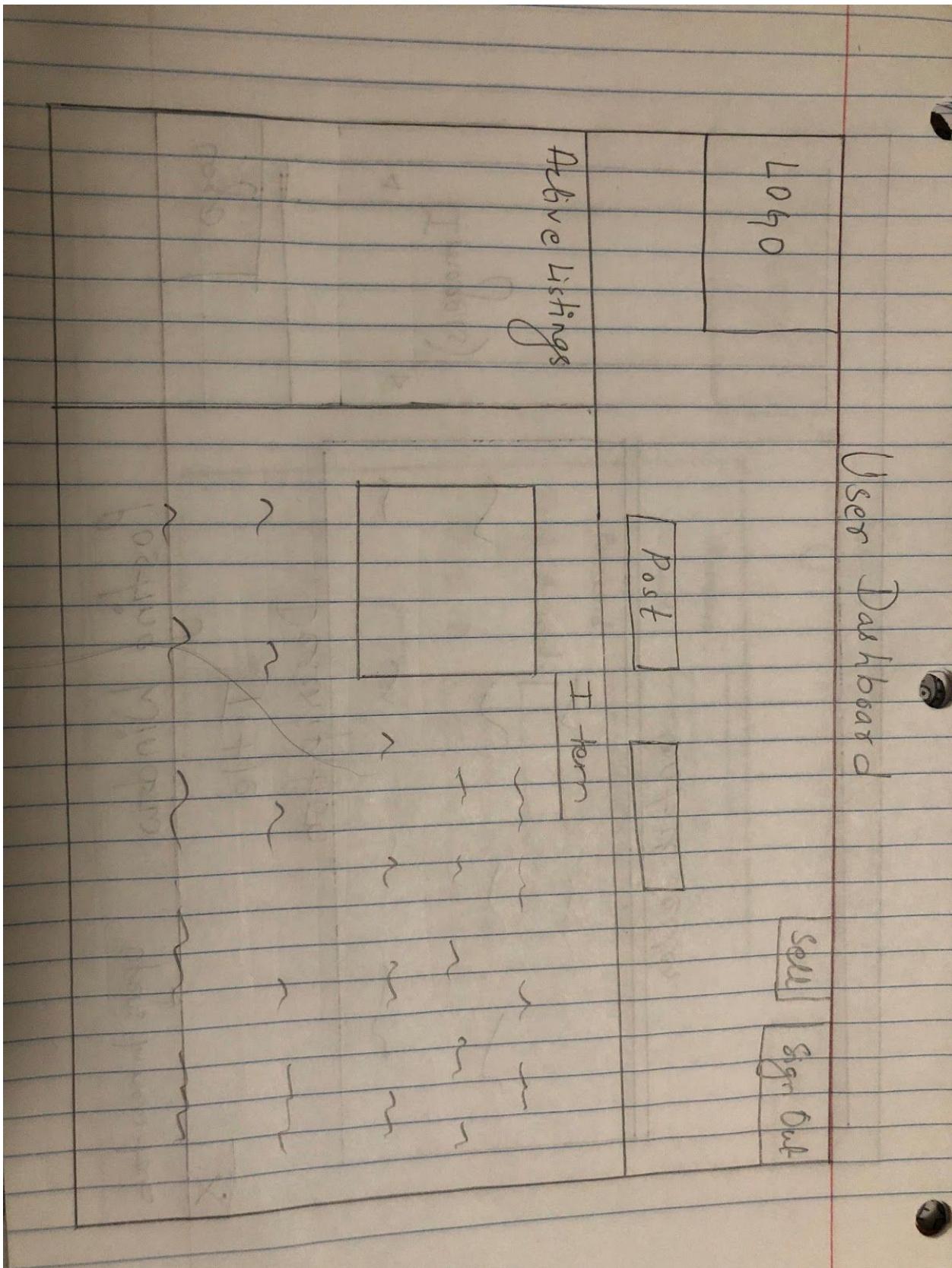
Above Listings

Post

Item

SELL

Sign Out



SW Engineering CSC648/848 Fall 2018

Gator Exchange

Team 11

Matthew Ho

Jason Chow

Dian Zhu

Riza Shrestha

Justin Shee

Ghulam Khan

Juan Valdez

“Milestone 4”

December 02 2018

Revision	Comment	Date
1	Revised according to feedback	Dec 07, 2018

Product Summary:

Gator Exchange is a website that serves as a buy and sell hub that caters towards students at San Francisco State University. Our website is unique in its ability to allow students to find what they need and sell what they want with ease of use. Postings on *Gator Exchange* are a great way to showcase items by student sellers, and provide an easy way of contact for potential buyers.

I. All Users

- A. All users are able to view recent postings
- B. All users are able to register to create an account which is used for contacting sellers and making new item posts.
- C. All users are able to search through items by category or name.

II. Registered Users

- A. Registered users are allowed to contact sellers to inquire about item postings
- B. Registered users are presented with a new posting form which allows them to create a new listing that includes a description, title, image, and price of the item.
- C. Registered users have access to a seller dashboard which contains a list of postings they've made and comments that have been made on their posts.
- D. Registered users can delete or modify their item listings

III. Administrators

- A. Administrators can browse the website and have the same functionality as a registered user.
- B. Administrators have their own admin dashboard which contains a list of postings that are pending for approval.
- C. Administrators can approve or disapprove postings.

URL: <https://gator-exchange.netlify.com/>

Usability Test Plan: Search

Test Objectives

Our test objective is to find out how usable is the search function through usability tests. These tests will identify if the search function is effective, efficient, and satisfactory. These three keys will be used as a metric to understand the usability of the search function. The function of search is to populate the page with similar to close results as the keywords inputted on the search bar. First, the search function must be able to consistently provide accurate results. Second, the search function must be efficient in time, effort, and has an easy to understand interface. Time is calculated as duration from start to finish of using the search function. Effort is calculated by ease of accessibility. Finally, satisfaction will be defined by the user on how they feel about the overall usability of the search function. The objective is to define where our search function feature is lacking and how we can improve on it with the information we gather from usability testing.

Test Plan

URL of system to be tested: <https://gator-exchange.netlify.com/>

System Startup

Our search function is usable on most operating systems and web browsers. To test simply use the URL above to be directed to gator exchange. After visiting Gator Exchange, use the search bar to search for items.

Starting point

When visiting Gator Exchange, users may enter an item they are looking for. For example, Books, Clothes, White Shirt, batteries, etc. Search results will then be generated.

Intended User

The intended user of our search function is mostly SFSU students visiting our website, Gator Exchange. However, any user visiting the website is allowed to use our search function. This allows for users to actively search for items they would like to buy. Making an easy and effective way to search our extensive catalog of items.

Task to be accomplished

Task	Description
Search input	Search item
Search	Search function running
Successful completion criteria	Results displayed on page
Benchmark	Completed in 30 seconds

Questionnaire

Place an X on one choice per question, Provide optional comments about your usability.

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to find the search bar					

Comments:

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It was easy to search for an item					

Comments:

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The results of searching for a book was accurate					

Comments:

3.) QA test plan

Test Objectives

Our test objective for QA testing is to be able to be able to test our search function performs to the specs of our software, or in this case our website. The idea is that we want to be able to test whether our search function to do certain things given specific parameters. First thing to test for is if our search results yield results despite the search parameter being empty as we wouldn't want people to be given no result in order to persuade them to take a look at some products. The second thing we want to test is that when we type in parts of what is being searched for we get some similar results that way if a word is misspelled the function doesn't give us a blank screen once again. Lastly, we want to be able to check if the search result is yielding proper results, we wouldn't want something along the lines of searching for shirt and having the first few searches come up as pants. This would be a rather large bug, but this could happen and if not tested for would make the website unusable. Events that are not tested for through the search function will be how the results are displayed, despite making use of the search function this is another function in itself. Another event not tested would be the checking if the category works as that in itself is another function which is combined with the search function in order to display results. In this test we want to make sure search works fine and without major bugs that could make the the website not perform properly.

HW & SW setup

URL of system to be tested: <https://gator-exchange.netlify.com/>

Hardware

Windows Laptop 8GB RAM, intel core i5

Mac Laptop 8GB RAM, intel core i5

Software

Using Latest 2 version of Firefox and Chrome

The website frontend will be run through Netlify

The website backend will be run through Heroku

Coded through VS Code through the use of React & Javascript

Feature to be tested: Search function

Test Plan

Test DB

Item 1: category = "clothes"; "title": "Bape Hoodie",

Item 2: category= "books"; "title": "CSC 648 Book",

Item 3: category= "books"; "title": "CSC 600 Book",

Item 4: category="clothes"; "title": "Plain White Shirt",

Item 5: category="miscellaneous"; search = "title": "4 AA Batteries",

Test

Number	Title	Description	Test Input	Expected Output	Pass/Fail
1	Empty Search Bar Test	Test empty string in search to check if all items will display	"" into the search bar on the main page then press the enter key	Get all 5 results from the test DB	Pass
2	Search Working Test	Test that % like will work by putting in a string that belongs in multiple posts and check if both results appear	"Book" into the search bar on the main page then press the enter key	Get the 2 results with books in the title namely: CSC 648 Book and CSC 600 Book	Pass
3	Partial Search Test	Another test for %like to check if someone doesn't complete the word will their result still come up or if autofill exists	"Shir" into the search bar on the main page then press the enter key	Bring up the result of the plain white shirt	Pass

4) Code Review

- a) The coding style we chose is ES6 Javascript following Google's coding convention on the front and backend.
- b) We used Github for our peer review and we communicated code over Discord and Slack.

A screenshot of a GitHub pull request review thread. The main message is from user `khanmustufa294`, commented 24 minutes ago, requesting a review from `mattbho`. The pull request details show commits by `sthriza` and others, including styling changes for login and register pages. A comment from `mattbho` requests changes, specifically noting that a text area tag must be changed to an input tag with type = text. The right sidebar shows the review status as "Requested changes must be addressed to merge this pull request". Other sections include Reviewers, Assignees, Labels, None yet, Projects, None yet, Milestone, None, and Notifications, which states "You're not receiving notifications from this thread".

khanmustufa294 commented 24 minutes ago • edited

Pull request to Matt for code Review

sthriza and others added some commits 14 days ago

- login and register styling added
- Update login.css
- updated login page style
- updated register page style
- updated register page style
- Edited and styled the Sell-form with similar format as login and register
- Fixed container typo

khanmustufa294 requested a review from mattbho 24 minutes ago

mattbho requested changes 21 minutes ago • edited

Text area tag must be changed to input tag with type = text

View changes

Reviewers: mattbho

Requested changes must be addressed to merge this pull request

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: None

Notifications: You're not receiving notifications from this thread.

Add more commits by pushing to the **front-end-styling** branch on CSC-648-SFSU/csc648-fa18-Team11.

5 participants

This branch has not been deployed
No deployments

Changes requested
1 review requesting changes by reviewers with write access. [Learn more.](#)

mattbho requested changes [See review](#) [Dismiss review](#)

All checks have passed
2 neutral and 3 successful checks

Header rules - gator-exchange Completed in 2m — No header rules processed [Details](#)

Pages changed - gator-exchange Completed in 2m — 10 new files uploaded [Details](#)

Mixed content - gator-exchange Successful in 2m — No mixed content detected [Details](#)

Redirect rules - gator-exchange Successful in 2m — 3 redirect rules processed [Details](#)

netlify/gator-exchange/deploy-preview — Deploy preview ready! [Details](#)

This branch has conflicts that must be resolved
Use the [web editor](#) or the [command line](#) to resolve conflicts. [Resolve conflicts](#)

6 client/src/App.js

[Copy path](#) [View file](#)

#	Line	Content
	6	@@ -6,6 +6,10 @@ import "./styling/navbar.css"; 6 import "./styling/errorpage.css"; 7 import "./styling/homepage.css"; 8 import "./styling/details.css"; 9 + import "./styling/register.css"; 10 + import "./styling/login.css"; 11 + 12 +
	13	
	14	10 import NavBar from "./pages/components/navbar";
	15	11 import SearchBar from "./pages/components/searchbar";
	16	@@ -17,7 +21,7 @@ import PostDetails from "./pages/PostDetails"; 16 import SearchResults from "./pages/components/search-results"; 17 import SellForm from "./pages/components/sell-form"; 18 import Login from "./pages/components/login"; 19 - import Register from "./pages/components/register" 20 + import Register from "./pages/components/register";
	21	21 class App extends Component { 22 render() {
	23	23 }

5) Self-check on best practices for security :

1. Buyer account information
2. Seller account data
3. Server security
4. Database protection
5. Secure internet protocols

Best Practices:

Password encryption in database is done by hashing with bcrypt. Users are able to register an account or login if they have existing account, which information gets stored in the database.

For registration, combination of frontend html code and passport is being used to validate the data. User emails are being checked by using Javascript to ensure it matched the correct email format. For password, user needs to confirm the password so that there is no error.

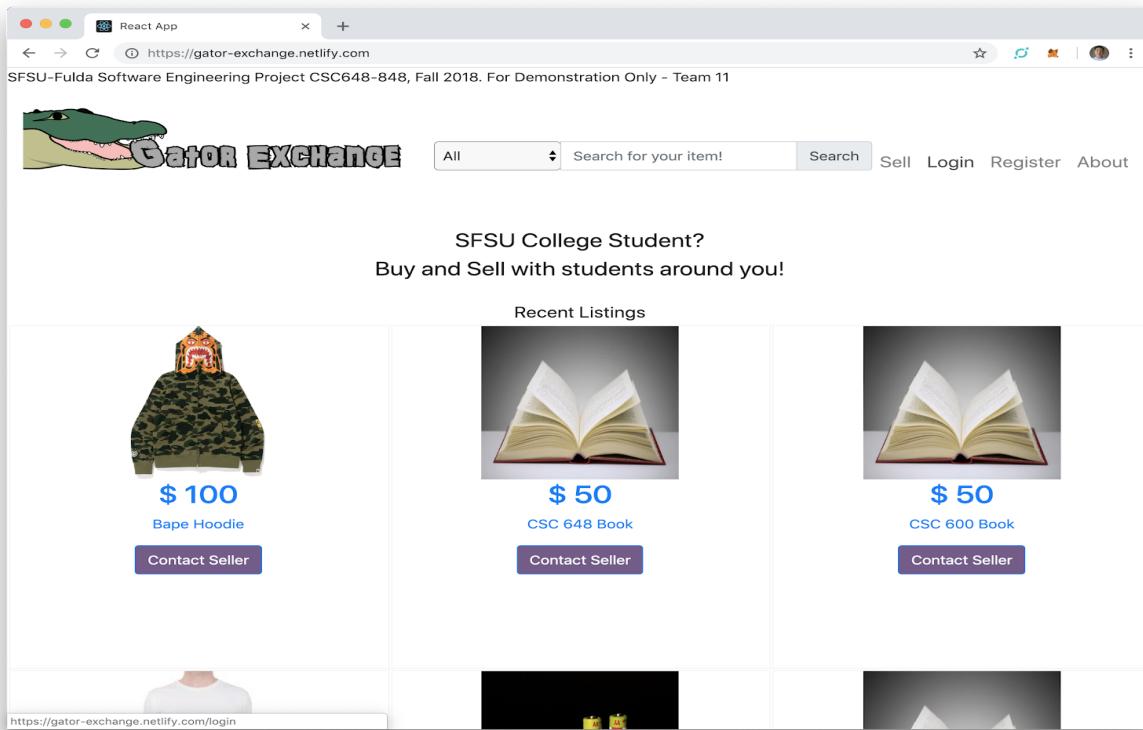
Search bar input is also being validated using frontend html codes. The search bar input is also being checked by % like algorithm which compares whether the input information matches the database or not.

6) Self-check: Adherence to original Non-functional specs

High Level specifications:

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. -**Done**
3. Selected application functions must render well on mobile devices -**Done**
4. Data shall be stored in the team's chosen database technology on the team's deployment server.-**On Track**
5. No more than 50 concurrent users shall be accessing the application at any time-**Done**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.-**Done**
7. The language used shall be English.-**Done**
8. Application shall be very easy to use and intuitive.-**Done**
9. Google analytics shall be added -**Done**
10. No email 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-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only*" at the top of the WWW page. (Important so as to not confuse this with a real application).-**Done**

4.) Product Screenshots



React App

https://gator-exchange.netlify.com/searchresults

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

 Gator EXCHANGE

All Search Sell Login Register About

 \$ 50 CSC 648 Book Contact Seller	 \$ 50 CSC 600 Book Contact Seller	 \$ 50 CSC 600 Book Contact Seller
---	---	---

React App

https://gator-exchange.netlify.com/sell

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

 Gator Exchange

All Search for your item! Search Sell Login Register About

Create Listing/Sell

Select Category

Item Name:

Item Description:

Price:

Upload Picture: Choose File No file chosen

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

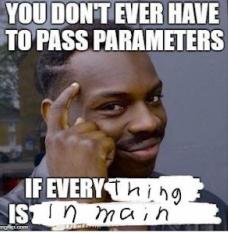
 Gator Exchange

All Search for your item! Search Sell Dashboard About Logout

My Dashboard

Messages My Postings Pending Posts

YOU DON'T EVER HAVE TO PASS PARAMETERS



IF EVERYThing IS IN main

Meme \$ 3 Memes

Approve Remove

React App

https://gator-exchange.netlify.com/register

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

 **Gator Exchange**

All Search for your item! Search Sell Login Register About

Register

Username

Email Address

SFSU ID

Password

Confirm password

REGISTER

[Sign In](#)

React App

https://gator-exchange.netlify.com/login

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

 **Gator Exchange**

All Search for your item! Search Sell Login Register About

Login

Username

Password

Login

Need an Account? [Register](#)

React App

https://gator-exchange.netlify.com/about

SFSU-Fulda Software Engineering Project CSC648-848, Fall 2018. For Demonstration Only - Team 11

 GATOR EXCHANGE

All Search for your item! Search Sell Login Register About

About Us

Matt Ho



Hey Guys, this is Matt Ho. I'm the team lead for this project. I love music, fitness, and sports. I'm an aspiring full stack developer.

Justin Shee



Backend Lead for team11 of CSC648. I'm working towards a computer science major at San Francisco State University. I've had many experiences with java but recently have started learning many other language such as ruby and swift. My dream job would be anything that has to do with servers because it's something I have worked with before even thinking about being a computer major

Jason Chow



Dian Zhu



5.) Key DB Tables

Posts

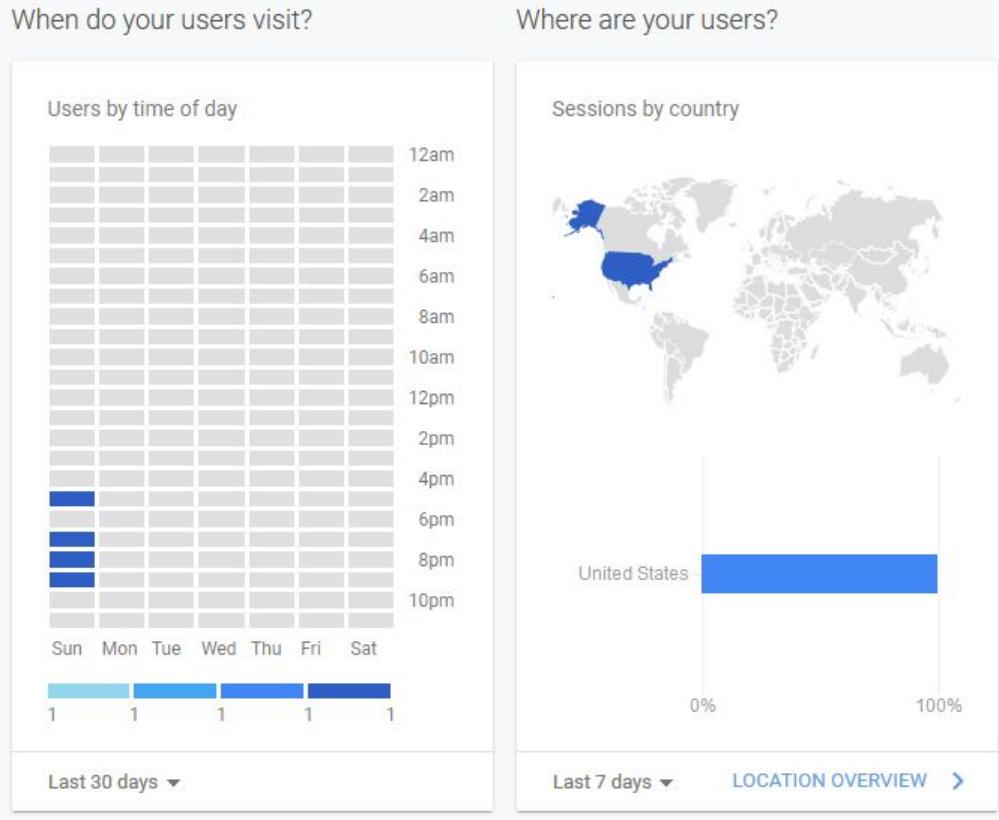
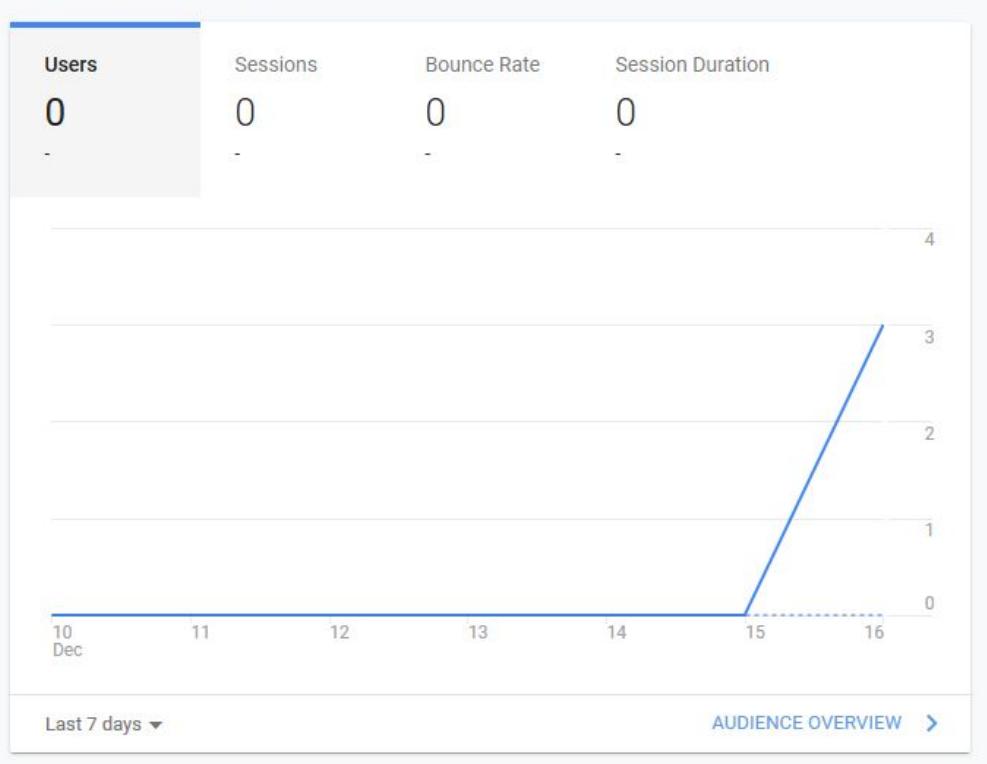
records / page <input style="width: 40px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="10"/> <input style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="▼"/>		[1 - 10 of 20] next > last >>				
Description	Price	Sellerid	Title	Category	ID	
Awesome Bathing Apes hoodie for sale 100 OBO	100		Bape Hoodie	Clothes	5bd0cb01745edb3c4e9de032	
CSC 648 Solutions guide for sale	50		CSC 648 Book	Books	5bd0cb15e13b493c657d8f2d	
CSC 600 Slides and activities	50		CSC 600 Book	Books	5bd0d5aeabbdd55070992b3f	
Size medium, white shirt	15		Plain White Shirt	Clothes	5bd2523b70825ffcef53aa78	
4 Standard AA Batteries Full Charge	5.99		4 AA Batteries	Miscellaneous	5bd25668797b4bf9b4acda5c	
CSC 600 Slides and activities	50		CSC 600 Book	Books	5bdcf3537edb497c6cc4703e	
Used Good Condition, wiped data, ready-to-use	120		500gb ssd	Miscellaneous	5c16b9a14d9250001688ea05	
test	1		test	Car	5c16e92ec896a0782fd1e8b1	
test	111		test	Select Category	5c16f6997c230e0ee3b46b86	
Haircut by ya boy	50		Haircut	Clothing	5c16fedc0b0644137727ebc5	
records / page <input style="width: 40px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="10"/> <input style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="▼"/>	[1 - 10 of 20] next > last >>					

Users

records / page <input style="width: 40px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="10"/> <input style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="▼"/>		[1 - 5 of 5]		
messages		username	email	sfsuld
[]		test	test@test.com	0
[]		bob	bob@bob	1
[]		t	t@t.com	2222222222
["5c14a32868cf4543418c075", "5c14a34868cf4543418c076", "5c153e35b9e867001652373b", "5c16a65c4aa90a27d9f83c36", "5c16c81eaa5ac324a8fe3a63", "5c16c83baa5ac324a8fe3a64", "5c16c8bcaa5ac324a8fe3a65"]		Matt	test	1
[]		nagasnow	lozphreak@gmail.com	913091548
records / page <input style="width: 40px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="10"/> <input style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px;" type="button" value="▼"/>	[1 - 5 of 5]			

6.) Google Analytics Plot

Google Analytics Home



7). Team Member Contributions



Matthew Bryan Ho

Mon 12/17/2018 2:25 PM

To: justinshee2@gmail.com; Dian Zhu; rizashrestha01@gmail.com; Jason Chow; Juan Daniel Valdez; ghulamk2@gmail.com



Reply all | v

My Contributions:

Set up meetings,
Assigned tasks,
Suggested software stack,
Main point of contact for professor and team,
Quality control,
Back end and Front-end development: Including search, user login/registration, user dashboard, admin dashboard
User Login,
Database management/implementation,
github manager,
Milestone Documentation,
Manage deployed servers

Github Commits:

91 commits



Justin Shee <justinshee2@gmail.com>

Mon 12/17/2018 12:07 PM

To: Matthew Bryan Ho; Juan Valdez <juanvaldez.college@gmail.com>; ghulamk2@gmail.com; Dian Zhu; rizashrestha01@gmail.com; Jason Chow



Reply all | v

Contributions- Justin:

- Backend loading postDetail data
- PostDetail simple no styling frontend creation
- Backend loading storing sell form data
- Creating and maintaining cloudinary image storage
- handling image upload
- sell form validation

6 commits.



Dian Zhu

Mon 12/17/2018 10:40 AM

To: Juan Valdez <juanvaldez.college@gmail.com>; ghulamk2@gmail.com; Matthew Bryan Ho; justinshee2@gmail.com; rizashrestha01@gmail.com; Jason Chow



Reply all | v

Required: Dian Zhu; Juan Valdez <juanvaldez.college@gmail.com>; ghulamk2@gmail.com; Matthew Bryan Ho; justinshee2@gmail.com; +2 more



When: Mon 12/17/2018 11:00a - 11:30a

Where:

Accept

Tentative

Decline

Propose new time

This invitation was updated after this message was sent. Open the update or open the item on the calendar.

This event occurs in the past.

I am happy that I have been working with everyone. My contributions are about front end style, and I also complete some parts of the milestones.
Commits: 3

Juan Valdez <juanvaldez.college@gmail.com>

  Reply all | ↻

Mon 12/17/2018 10:06 AM

To: ghulamk2@gmail.com;  Matthew Bryan Ho; justinshee2@gmail.com; rizashrestha01@gmail.com;  Dian Zhu;  Jason Chow 

I contributed to styling and making the website look visually appealing in certain areas such as the sell-form page, some of the dashboard details for users. I contributed on the milestones finishing the given parts that were assigned to me, and offered to help when there was work to be accomplished. I tried my best to involve people in styling to help around and be involved with the team. Attempting to do my best quality of work when it came to making the website visually appealing, though it felt like I did not do much at all, and the back-end team did the majority of the hard work. I do believe that when compared to the back end I did not do much, but at least I managed to finish what was assigned to me in styling.

Commits made: About 6-8.

-Juan Valdez

ghulamk2@gmail.com

  Reply all | ↻

Mon 12/17/2018 10:18 AM

To: Juan Valdez <juanvaldez.college@gmail.com>

Cc:  Matthew Bryan Ho; justinshee2@gmail.com; rizashrestha01@gmail.com;  Dian Zhu;  Jason Chow 

First of all, this has been a great team. I really enjoyed working with everybody in the team.

My contributions have been with mostly front end "about me" page as well as finishing my assigned part of the milestone. I also was very much involved suggesting different ways to the team on tackling some of the problems.

Commits: 8-10

RS

Riza Shrestha <rizashrestha01@gmail.com>

  Reply all | ↻

Mon 12/17/2018 11:02 AM

To: ghulamk2@gmail.com;  Matthew Bryan Ho; Justin Shee <justinshee2@gmail.com>; Riza Shrestha <rizashrestha01@gmail.com>;  Dian Zhu;  Jason Chow; juanvaldez.college@gmail.com 

Mostly contributed on styling login and registration page so that it looks good. I also contributed on milestones for the parts that were assigned to me.

Commits: 1-2

Team Contribution - Jason Chow



Jason Chow

Mon 12/17/2018 10:15 AM

To: Juan Valdez <juanvaldez.college@gmail.com>; ghulamk2@gmail.com;  Dian Zhu; rizashrestha01@gmail.com; justinshee2@gmail.com 

Contributions (Jason)

- React Router
- Navigation Bar
- Front page styling
- Dashboard styling
- Post detail styling

12 Commits in Github team repo

8) Post Analysis

Post Analysis

 Matthew Bryan Ho
Today, 12:55 PM
rizashrestha01@gmail.com; Dian Zhu; ghulamk2@gmail.com; justinshee2@gmail.com; Jason Chow; Juan Daniel Valdez

  Reply all | 

This project was a great learning tool for teaching me and my team modern software engineering processes. Among the many topics I've learned, I think that these tools will help me in the real world more than any other skills that I've learned in any other classes. As a Team Lead, I had the responsibility of learning modern SE processes as well as learning to manage a team of seven. These responsibilities had many challenges and it has taught me how to approach them better in my future.

One of the biggest challenges was learning to manage a large team. We had a pretty inexperienced team when it came to the stack that we chose so learning the technology was a challenge within itself. The issue with this was that it gave us very limited time to work because everyone was still learning. There were also language barrier constraints among one of our members so it was difficult to teach them as well as have them do some work. Most members contributed as much as they could but there were some members who barely contributed at all. The most difficult part about these members were keeping them up to date with the current code. Being that they were barely participating, they fell behind even more to the point where they did not know the stack, and thus stopped contributing past the second month of the semester. I think the biggest issue that I had with some members were the lack of reciprocation when it came to effort, leaving many of my delegations untouched.

In terms of things that I could have done better to address these challenges, there were things that came to mind. I definitely think I could have educated myself on the stack better in order to further teach everyone else. I should have definitely dispersed the load much more than I did, as well as enforced stricter communication between some members. Some members did not even show up to class, let alone communicate to the team the entire semester so I could have definitely been more strict when reaching out to them. Those members that did help as much as they could however, I could have done them a much better service by commenting out my code better and being more available. Overall, this was a great learning experience and everything came together at the end.

Transcript:

This project was a great learning tool for teaching me and my team modern software engineering processes. Among the many topics I've learned, I think that these tools will help me in the real world more than any other skills that I've learned in any other classes. As a Team Lead, I had the responsibility of learning modern SE processes as well as learning to manage a team of seven. These responsibilities had many challenges and it has taught me how to approach them better in my future.

One of the biggest challenges was learning to manage a large team. We had a pretty inexperienced team when it came to the stack that we chose so learning the technology was a challenge within itself. The issue with this was that it gave us very limited time to work because everyone was still learning. There were also language barrier constraints among one of our members so it was difficult to teach them as well as have them do some work. Most members contributed as much as they could but there were some members who barely contributed at all. The most difficult part about these members were keeping them up to date with the current code. Being that they were barely participating, they fell behind even more to the point where they did not know the stack, and thus stopped contributing past the second month of the semester. I think the biggest issue that I had with some members were the lack of reciprocation when it came to effort, leaving many of my delegations untouched.

In terms of things that I could have done better to address these challenges, there were things that came to mind. I definitely think I could have educated myself on the stack better in order to further teach everyone else. I should have definitely dispersed the load much more than I did, as well as enforced stricter communication between some members. Some members did not even show up to class, let alone communicate to the team the entire semester so I could have definitely been more strict when reaching out to them. Those members that did help as much as they could however, I could have done them a much better service by commenting out my code better and being more available. Overall, this was a great learning experience and everything came together at the end.