

S W A M P

SW Engineering CSC648/848 Spring 2020

TEAM 04 Global - Milestone 04

May 21, 2020

Team Lead: Benjamin Lewis - email:

benjaminlewis984@gmail.com

Git Master: Weerachai Poorakkiat

Front-End Lead: Dang Le

Back-End Lead: William Lew

Front-End Dev: Onubulachi-Abigail Wami

Back-End Dev: Kevin Huynh

Swamp url - <http://18.191.184.143:3000/>

-Product summary-

Swamp, an online marketplace for the SFSU community by the SFSU community.

Our product contains a unique function that caters directly to SFSU students and faculty by means of utilizing a search feature that allows for site content to be searched by a class filter. This class filter allows users of Swamp to find/post items based on a specific SFSU class (i.e CSC648).

Our product performs the following functions:

1. Unregistered users **shall**:
 - I. Be able to register
 - II. Be able to browse site content
2. Registered users **shall**:
 - I. Be able to remove listing
 - II. Be able to post media for sale
 - III. Be able to logout
 - IV. Be able to login
 - V. Be able to message media seller
 - VI. Be able to request to purchase
 - VII. Be able to approve purchase request
 - VIII. Be able to deny purchase request
3. Administrative users **shall**:
 - I. Be able to login
 - II. Be able to logout
 - III. Be required to approve posting
 - IV. Be able to delete posting
 - V. Be required to view user reports
 - VI. Be able to ban user
 - VII. Be able to unban user

Product url - <http://18.191.184.143:3000/>

SWAMP

SW Engineering CSC648/848 Spring
2020

*TEAM 04 - Milestone 01
FREEZE VERSION*

March 12, 2020

Executive Summary

Swamp is a media marketplace made by San Francisco State University students for other SFSU community members that offers trade and exchanging of media. Developed by gators, the creators of Swamp found that currently there is not a multimedia marketplace that caters only to SFSU students and faculty. Using modern software web application technology, our team is able to ensure quick, easy and reliable use of our product so that our users can have exclusive access to a reliable resource of media assets for academic or extracurricular usage.

Swamp offers a fluid user experience that is user-friendly for the SFSU community members that may be familiar or unfamiliar with web application usage, while also offering features that tailor to needs of the student body and SFSU faculty. Our search system offers a feature allowing for students to filter listings posted by area of study. Upon entering the site, users are able to browse the site's marketplace. Once an unregistered user commits to a transaction, they will be prompted to register ensuring that the buying and selling feature is exclusive to registered users. Admin acts as a filter for content by approving or denying user listings so that all visible content is maintained to be appropriate.

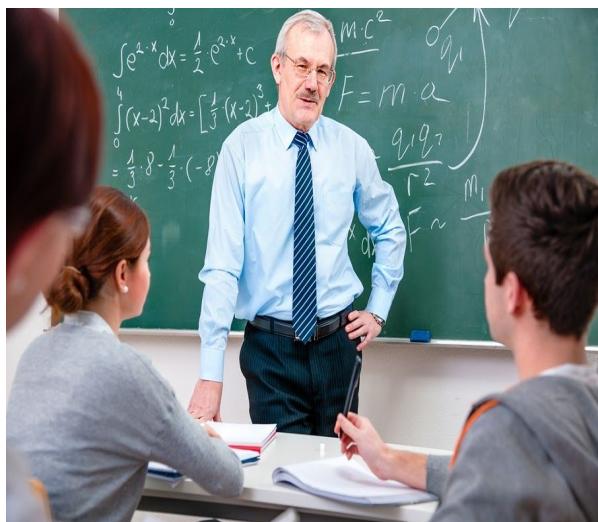
Swamp's team is composed of upper division San Francisco State University computer science students and operates out of San Francisco. Each team member adds a unique perspective to the Swamp product while being able to effectively work as a team in order to create a product that delivers an efficient exchange of media between SFSU community members. Since Swamp is created by students, our team is able to shape the product toward the needs of the demographic we are serving.

Personas



Registered - Emma SFSU Student

- She wants to use a website that can provide easy usage and friendly user interface
- She wants to help out students who are selling media with Swamp
- She wants to sell her media in the future



Registered - Michael SFSU Faculty

- He needs to post class material that will be easy for his students to access
- He wants to be able to have his classmates find this material quickly with a easy search function
- He wants students to be able to download the content quickly



Unregistered - Robert SFSU Student

- He wants an easy search engine to use
- Website should not have too much confusion to use for him
- He needs less clicks to get what he wants



Admin - William SFSU Student

- Experienced programmer
- Can ban suspicious behavior
- Works part time on campus
- Knows school posting policy

Use Cases

1 - Registered User

1.1 Emma, a sophisticated student from San Francisco State, used to only buy required books from popular sites such as Amazon, Ebay, etc. These sites always sell digital book content overpriced as they do not cater their content toward students. Since being introduced to what Swamp's Services offers, she learned about how reliable and easy-to-use Swamp is. It is also a lot easier to search the website for SFSU specific content. Now, when searching for a required digital content, she is able to find e-books sold at noncommercial prices (cheaper than our competitors). Once being prompted to sign up she is able to easily receive assigned homework from her professors, while she is also able to download other types of digital media.

1.2 Michael is a professor at SFSU. He is teaching a class that requires his students to download slides. Since he wants all his students in each section of his classes to have access to the SFSU digital content. He knows that Swamp is the best place to post involving SFSU material. Since he tags his content through his specific class name she knows that her students will be able to download the content after a simple search. Swamp makes this transaction especially easy for his students with only a few clicks.

2 - Unregistered User

2.1 Robert, who works in construction, has little knowledge on how to use a website. He wants to go back to school. He realises that he needs to buy media content required by a class before the semester starts. He hears about Swamp Services from a professor. Swamp provides him with an easy browsing engine that allows him to find class content posted by his professors. All he has to do in order to download the posted item is register

through a very easy registration page that simply asks for email and password.

3 - Admin User

3.1 William is an experienced programmer who monitors Swamp's services. If the admin sees or is notified about inappropriate content he can either ban a user account or delete user posted content. Admin is in charge of approving items to be able to sell on Swamp. Every item has to have approval from administrators like William to be able to sell on Swamp.

Main data items and entities

Registered user	Users that own an account and are able to log in	Able to buy & sell products
Admin	Administrative account that acts as a filter for listings	Approve or reject listings to ensure appropriate marketplace; ban users
Unregistered user	Users that do not own an account and but are able to register	Only able to browse all listings
Item listing	Listed item that can be an image, audio, video, or document file	Entity to be exchanged between users through the marketplace
Metadata	File name, size, type, author, item description, owner ID, category, price, licensing, pointer to raw media file, and date created	Used for keeping track of items as well as for filter/ searching purposes
Messaging	Users, upon checkout of listings, can message the owner in order to complete payment	Allows for ease of purchasing

Functional Requirements

1. Unregistered users **shall**:
 - I. Be able to search for Swamp data
 - II. Be able to filter their searches
 - III. Be able to register
 - IV. Be able to add item from cart
 - V. Be able to view from shopping
2. Registered users **shall**:
 - I. Be able to do everything unregistered user can except for being able to register
 - II. Be able to logout
 - III. Be able to login
 - IV. Be able to edit password
 - V. Be able to post media for sale
 - VI. Be able to message media seller
 - VII. Be able to edit their own existing listing
 - VIII. Be able to view their own listings, requests, approvals
 - IX. Be able to request to purchase
 - X. Be able to approve purchase request
 - XI. Be able to deny purchase request
 - XII. Be able to remove listing
 - XIII. Be able to report listing
3. Administrative users **shall**:
 - I. Be able to login
 - II. Be able to logout
 - III. Be required to approve posting
 - IV. Be required to delete posting
 - V. Be required to view user reports
 - VI. Be able to ban user

Non-functional requirements

- I. 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).
- II. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.
- III. Selected application functions must render well on mobile devices.
- IV. Data shall be stored in the team's chosen database technology on the team's deployment server.
- V. Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner.
- VI. No more than 50 concurrent users shall be accessing the application at any time.
- VII. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
- VIII. The language used shall be English (no localization needed).
- IX. Application shall be very easy to use and intuitive.
- X. Google analytics shall be used.
- XI. No e-mail clients shall be allowed.
- XII. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
- XIII. Site security: basic best practices shall be applied (as covered in the class) for main data items.
- XIV. Media formats shall be standard as used in the market today
- XV. Media material shall be either free or for sale, as determined by media owner.
- XVI. Each media material shall have its license info as one of the following: a) free use and modification; b) free but only allowed for SFSU related projects; c) for sale.
- XVII. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
- XVIII. The website shall prominently display the following exact text on all pages "*SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only*" at the top of the WWW page. (Important so as to not confuse this with a real application).

Competitive Analysis

Competitive Analysis	Amazon	Craigslist	Swamp
Search	++	-	++
Support	+	+	+
Mobile	+	-	+
User Friendly	++	-	+
SFSU class filter	-	-	++

Competitive Analysis Summary -

Swamp is able to surpass our competitors in more than a few ways. One way, by offering a easy searching function as well as filter, Swamp is able to give its users a much more accurate and easier to use searching experience than some of our competitors. Users will find that Swamp provides a user-friendly experience that allows browsing, buying, selling and registering easy and quick to do for any website or mobile usage experience level. Lastly, Swamp offers SFSU students and faculty a custom SFSU class searching feature that caters directly to students and professors that is completely original and not offered in any other website.

System architecture and technologies used

Server Host	Amazon Web Services
Operating System	Ubuntu v18.04
Database	MySQL v14.14
Web Server	Amazon EC2
Server-side Language	NodeJS v12.14.1

Front-end Framework	ReactJS v16.9.0
Back-end Framework	Express v4.16.1

Team and roles

Team Lead: Benjamin Lewis

Git Master: Weerachai Poorakkiat

Front-End Lead: Dang Le

Back-End Lead: William Lew

Front-End Dev: Onubulachi-Abigail Wami

Back-End Dev: Kevin Huynh

Checklist

- Team found a time slot to meet outside of the class - DONE
- Github master chosen - DONE
- Team decided and agreed together on using the listed SW tools and deployment server - DONE
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing - IN PROGRESS
- Team lead ensured that all team members read the final M1 and agree/understand it before submission - DONE
- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) - DONE

SWAMP

SW Engineering CSC648/848 Spring
2020

TEAM 04 Global - Milestone 02

March 19, 2020

1. Functional Requirements

-Priority 1-

1. Unregistered users **shall**:
 - I. Be able to register
 - II. Be able to browse site content
2. Registered users **shall**:
 - I. Be able to remove listing
 - II. Be able to post media for sale
 - III. Be able to logout
 - IV. Be able to login
 - V. Be able to message media seller
 - VI. Be able to request to purchase
 - VII. Be able to approve purchase request
 - VIII. Be able to deny purchase request
3. Administrative users **shall**:
 - I. Be able to login
 - II. Be able to logout
 - III. Be required to approve posting
 - IV. Be able to delete posting
 - V. Be required to view user reports
 - VI. Be able to ban user
 - VII. Be able to unban user

-Priority 2-

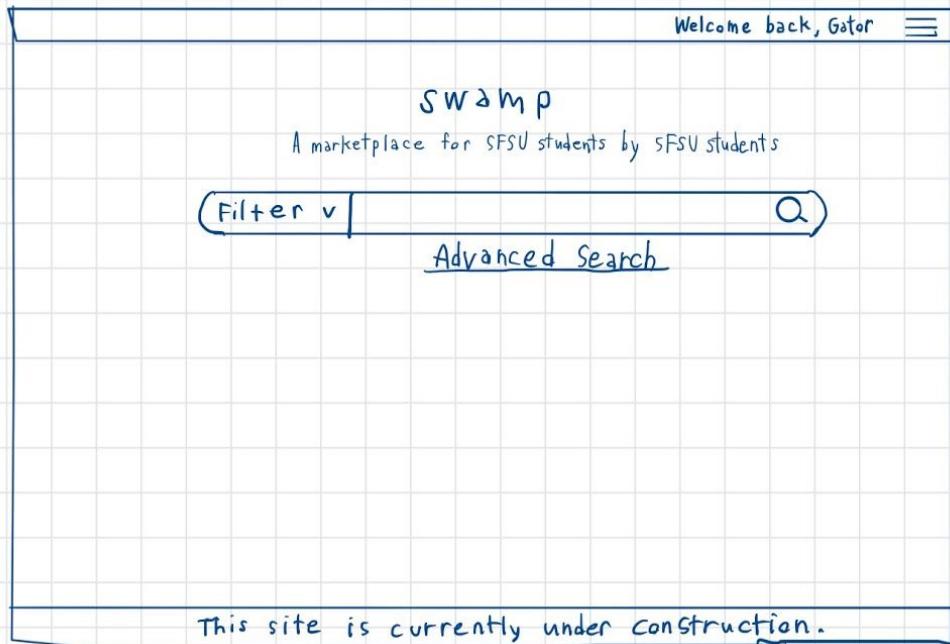
4. Unregistered users **shall**:
 - I. Be able to search for Swamp data
 - II. Be able to filter their searches
 - III. Be able to add item to cart
5. Registered users **shall**:
 - I. Be able to do everything unregistered user can except for being able to register
 - II. Be able to edit password
 - III. Be able to edit their own existing listing
 - IV. Be able to view their own listings, requests, approvals
 - V. Be able to report listing

2. Main data items and entities

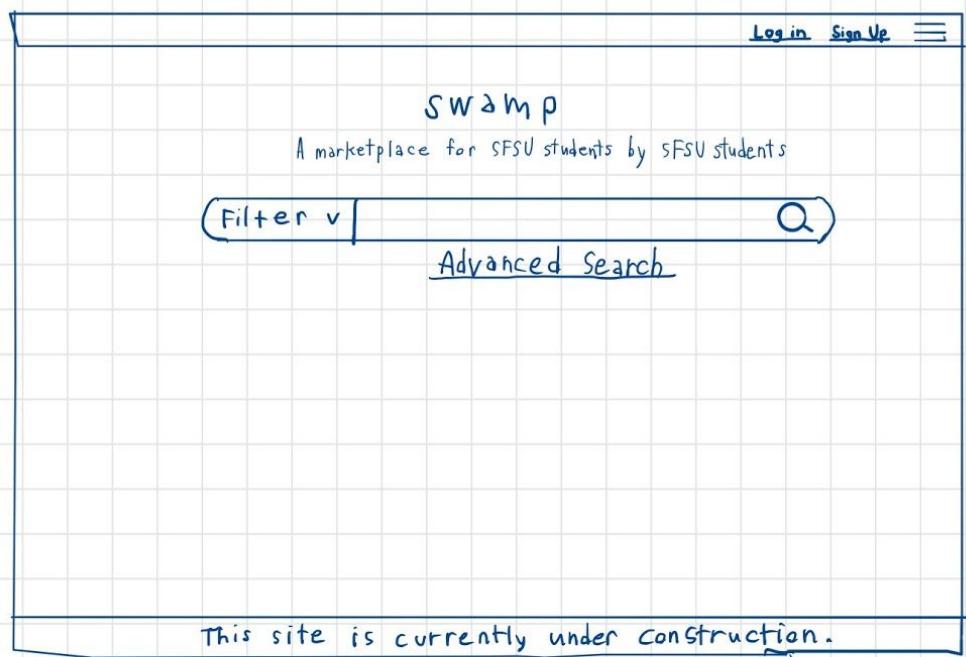
Registered user	Users that own account and are able to log in	Able to buy & sell products
Admin	Administrative account that acts as a filter for listings	Approve or reject listings to ensure appropriate marketplace; ban users
Unregistered user	Users that do not own an account and but are able to register	Only able to browse all listings
Item listing	Listed item that can be an image, audio, video, or document file	Entity to be exchanged between users through the marketplace
Metadata	File name, size, type, author, item description, owner ID, category, price, licensing, pointer to raw media file, and date created	Used for keeping track of items as well as for filter/ searching purposes
Messaging	Users, upon checkout of listings, can message the owner in order to complete payment	Allows for ease of purchasing

3. UI Mockups and Storyboard

1a



1b



- 1a. Homepage when logged in
- 1b. Homepage when logged out

Welcome back, Gator



SWAMP

A marketplace for SFSU students by SFSU students

Filter v



Advanced Search

This site is currently under construction.

Welcome back, Gator



SWAMP

A marketplace for SFSU students by SFSU students

Filter ^

banana



- All
- Documents
- Images
- Music
- Videos

Apply

Advanced Search



This site is currently under construction.

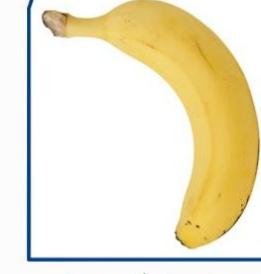
SWAMP Filter v banana Welcome back, Gator

All	<input type="checkbox"/>			
Documents	<input type="checkbox"/>	FREE	FREE	FREE
Images	<input type="checkbox"/>			
Music	<input type="checkbox"/>	FREE	\$10	\$15
Videos	<input type="checkbox"/>			

This site is currently under construction.

Flag

SWAMP Filter v banana Welcome back, Gator

All	<input type="checkbox"/>			
Documents	<input type="checkbox"/>	File: banana.jpg	Description: An image of a banana to inspire your artistic soul	X
Images	<input type="checkbox"/>	from gatorbay 2020	 Save	 Add
Music	<input type="checkbox"/>			 Flag
Videos	<input type="checkbox"/>			

This site is currently under construction.



age of a banana
stic soul

 Save  Add

SWAMP Filter v banana Welcome back, Gator

Showing items 1 to 10 of 30

All Documents Images Music Videos

<input type="checkbox"/>	FREE	FREE	FREE
<input type="checkbox"/>	FREE	\$10	\$15
<input type="checkbox"/>			

This site is currently under construction.

SWAMP Filter v banana Welcome back, Gator

Edit Profile View History Report User Saved Posts Upload New Post Checkout

All Documents Images Music Videos

<input type="checkbox"/>	FREE	FREE
<input type="checkbox"/>	FREE	\$10
<input type="checkbox"/>		

This site is currently under const About

swamp

Checkout (1 item)



Review items



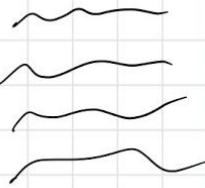
File: banana.jpeg

Description: An image of a banana
to inspire your artistic soul

by gatorbay2020



Finish



This site is currently under construction.

swamp Welcome back, Gator 

All
Documents
Images
Music
Videos


FREE


FREE


FREE


\$10




This site is currently under const [About](#)

Edit Profile
View History
Report User
Saved Posts
Upload New Post 
Checkout

swamp Welcome back, Gator 

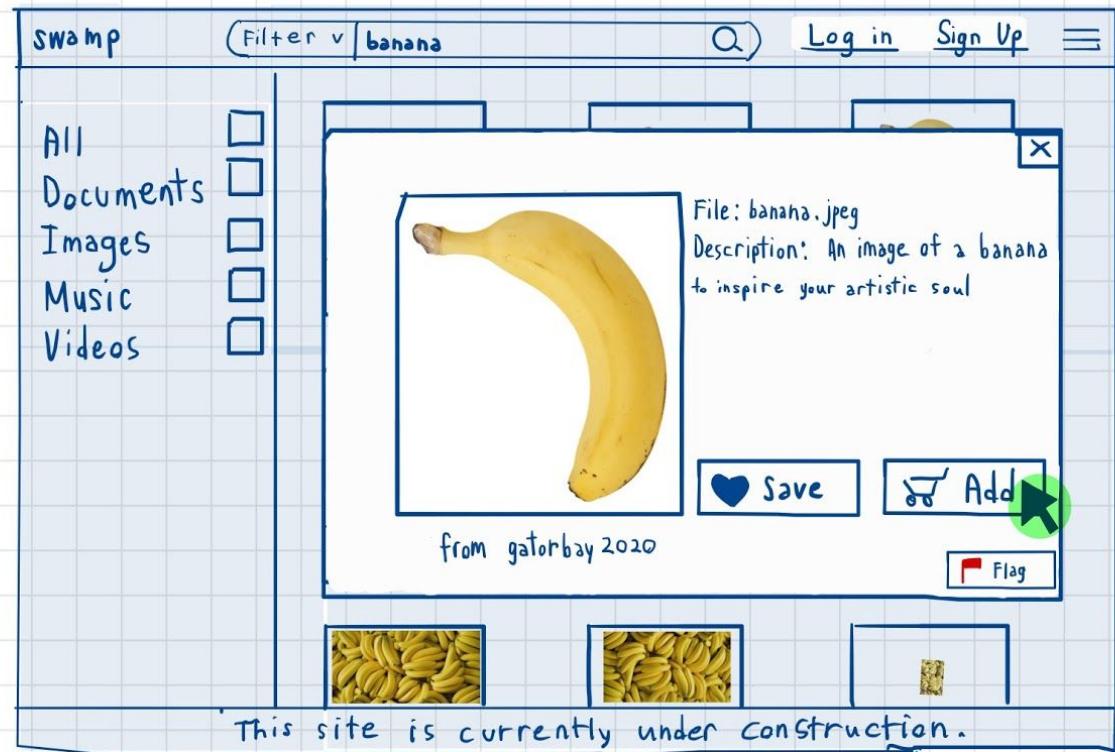
Title

Description: Please be informative about what you are uploading.

Upload : No file chosen.

Price:

This site is currently under construction.



SWAMP Register Here

←

Full Name
private

Username
public, cannot be changed

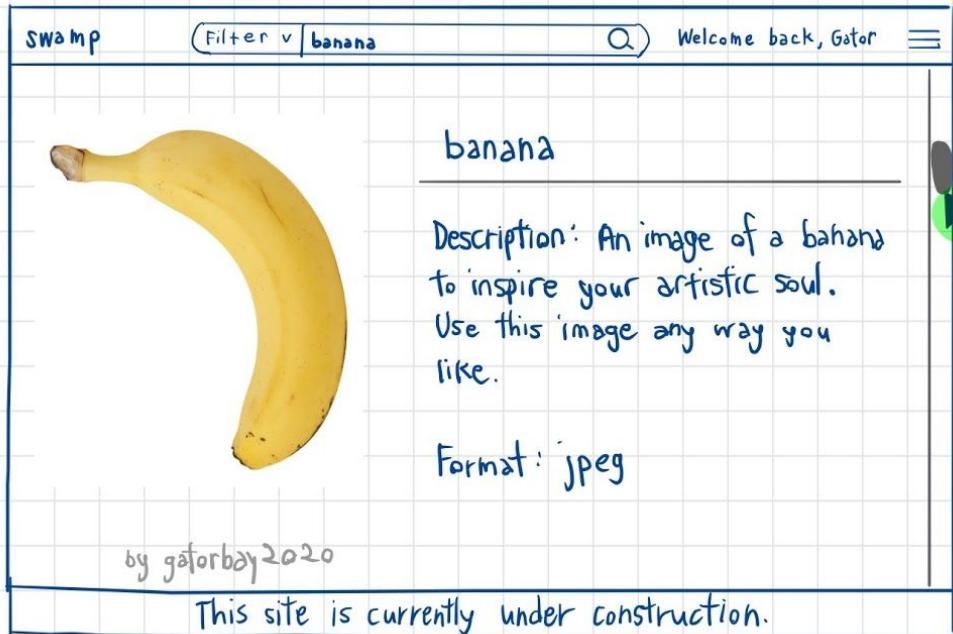
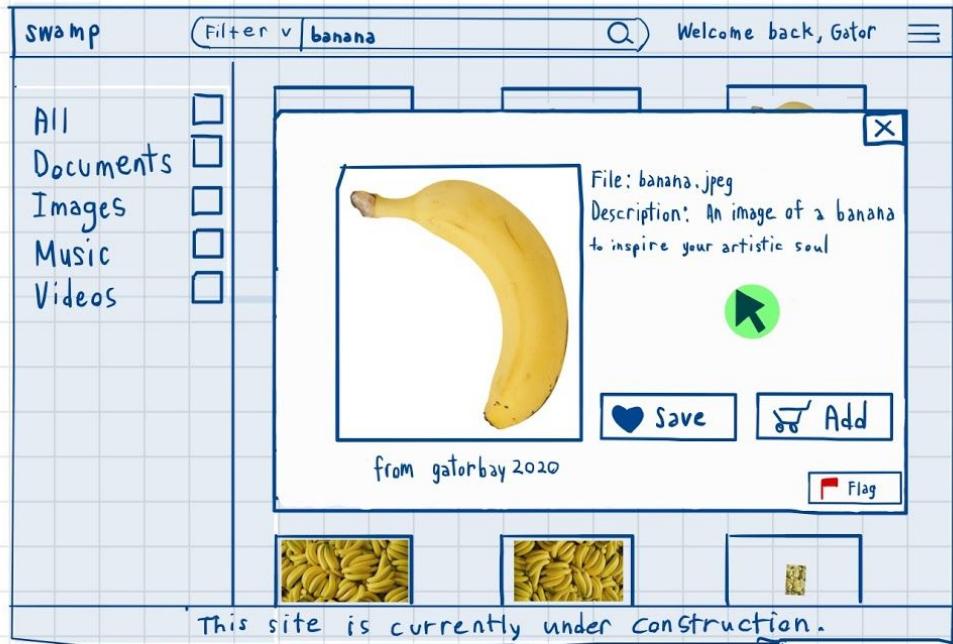
SFSU Email
cannot be changed

Password

Confirm Password

X Cancel ✓ Submit 
* Continue to checkout

This site is currently under construction.



Priority 4: Flag user/post

Swamp (Filter v banana) Welcome back, Gator

Use this image any way yo like.

Format: jpeg

by gatorboy2020

Flag 

Save Add

Similar items

This site is currently under construction.



Swamp (Filter v banana) Welcome back, Gator

Use this image any way yo like

Inappropriate post
Copyrighted material
Post does not match description
Other

Report

by gatorboy2020

Flag 

Similar items

This site is currently under construction.

Report

swamp	(Filter v)	<input type="text"/>	<input)<="" td="" type="button" value="Q"/> <td>Admin</td> <td></td>	Admin	
Pending		Banned	User List		

This site is currently under construction.



swamp	(Filter v)	<input type="text"/>	<input)<="" td="" type="button" value="Q"/> <td>Admin</td> <td></td>	Admin	
Pending		Banned	User List		
Title: ~~~~	User: ~~~~	<input type="checkbox"/> X	<input type="checkbox"/> ✓		
File Type: ~~~~			<input type="checkbox"/> X	<input type="checkbox"/> ✓	
Title: ~~~~	User: ~~~~	<input type="checkbox"/> X	<input type="checkbox"/> ✓		
File Type: ~~~~			<input type="checkbox"/> X	<input type="checkbox"/> ✓	
Title: ~~~~	User: ~~~~	<input type="checkbox"/> X	<input type="checkbox"/> ✓		
File Type: ~~~~			<input type="checkbox"/> X	<input type="checkbox"/> ✓	
Title: ~~~~	User: ~~~~	<input type="checkbox"/> X	<input type="checkbox"/> ✓		
File Type: ~~~~			<input type="checkbox"/> X	<input type="checkbox"/> ✓	
Title: ~~~~	User: ~~~~	<input type="checkbox"/> X	<input type="checkbox"/> ✓		
File Type: ~~~~			<input type="checkbox"/> X	<input type="checkbox"/> ✓	
F'in Type: ~~~~		<input type="checkbox"/> X	<input type="checkbox"/> ✓		

This site is currently under construction.

swamp (Filter v) Admin

Pending	Banned	User List
Title: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~ User: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
This site is currently under construction.		

swamp (Filter v) Admin

Pending	Banned	User List
Title: ~~~~~	User: ~~~~~	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~	User: ~~~~~	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Title: ~~~~~	User: ~~~~~	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
File Type: ~~~~~		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
		<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
This site is currently under construction.		

swamp Admin

Pending	Banned	User List
Title: ~~~~ File Type: ~~~~	User: ~~~~	<input type="button" value="X"/> <input checked="" type="button" value="✓"/>
Title: ~~~~ File Type: ~~~~	User: ~~~~	<input type="button" value="X"/> <input checked="" type="button" value="✓"/>
Title: ~~~~ File Type: ~~~~	User: ~~~~	<input type="button" value="X"/> <input checked="" type="button" value="✓"/>
Title: ~~~~ File Type: ~~~~	User: ~~~~	<input type="button" value="X"/> <input checked="" type="button" value="✓"/>

This site is currently under construction.

swamp Admin

Pending	Banned	User List
User: ~~~~		<input type="button" value="Ban"/> <input type="button" value="confirm"/>
User: ~~~~		<input type="button" value="Ban"/> <input type="button" value="confirm"/>
User: ~~~~		<input type="button" value="Ban"/> <input type="button" value="confirm"/>

This site is currently under construction.



4. Architecture and Database Organization

1. Database Organization:
 - I. Accounts
 - a. Acc_id: PK
 - b. Username: UK
 - c. Password
 - d. Email
 - II. Registered Users
 - a. Reg_id: PK
 - b. Acc_id: UK
 - c. First_name
 - d. Mid_initial
 - e. Last_name
 - f. Listing_req
 - g. Listing_app
 - h. Listing_den
 - III. Admin

- a. Admin_id: PK
 - b. Acc_id: FK, UK
- IV. Banned Users
- a. Ban_id: PK
 - b. Acc_id: FK, UK
 - c. Banned_by: FK
 - d. Reason
 - e. Ban_date
 - f. Unban_date
 - g. Ban_active
- V. Media Content
- a. M_id: PK
 - b. Title
 - c. Description
 - d. Preview_path: UK
 - e. Raw_path: UK
 - f. Category
 - g. Price
 - h. Acc_id: FK
 - i. Academic
 - j. Status
 - k. Status_by: FK
- VI. Digital Media
- a. Dm_id: PK
 - b. M_id: FK, UK
 - c. sold
- VII. Physical Media
- a. Pm_id: PK
 - b. M_id: FK, UK
 - c. Bought_by: FK
- VIII. Message Box
- a. Message_id: PK
 - b. Sender_id: FK
 - c. Acc_id: FK
 - d. Message
 - e. Date
 - f. Time
 - g. Read
- IX. Reported Users List

- a. Ru_id: PK
 - b. Reporter_id: FK
 - c. Reportee_id: FK
 - d. Date
 - e. Time
 - f. Reason
- X. Reported Media List
- a. Rm_id: PK
 - b. Reporter_id: FK
 - c. M_id: FK
 - d. Date
 - e. Time
 - f. Reason
- XI. Shopping Cart
- a. Item_id: PK
 - b. Approved_id: FK
 - c. Price: FK
 - d. Reg_id: FK
- XII. Wish List
- a. M_id: PK, FK
 - b. Reg_id: PK, FK
- XIII. Bought Digital Media
- a. Bought_id: PK
 - b. M_id: PK, FK
 - c. Reg_id: PK, FK
- XIV. Checkout
- a. Order_id: PK
 - b. M_id: FK, UK
 - c. Reg_id: FK, UK
2. Media Storage: All media content will be kept in file systems with their paths stored in the attributes: raw_path and preview_path.
3. Search/Filter Architecture and Implementation: We will be using a combination of the mySQL LIKE operator and SOUNDEX function. The terms that will be searched will be the titles and description of the Media Content table, but ONLY if the M_id is present in the Approved Media table. The Media Content items can be filtered into their own categories (ex. All, Documents, Videos, ...) and beyond that will be

whether the media content is academic or not, but further filtering has yet to be decided.

4. Our API: We will be creating our own API to abstract away the MySQL connector.

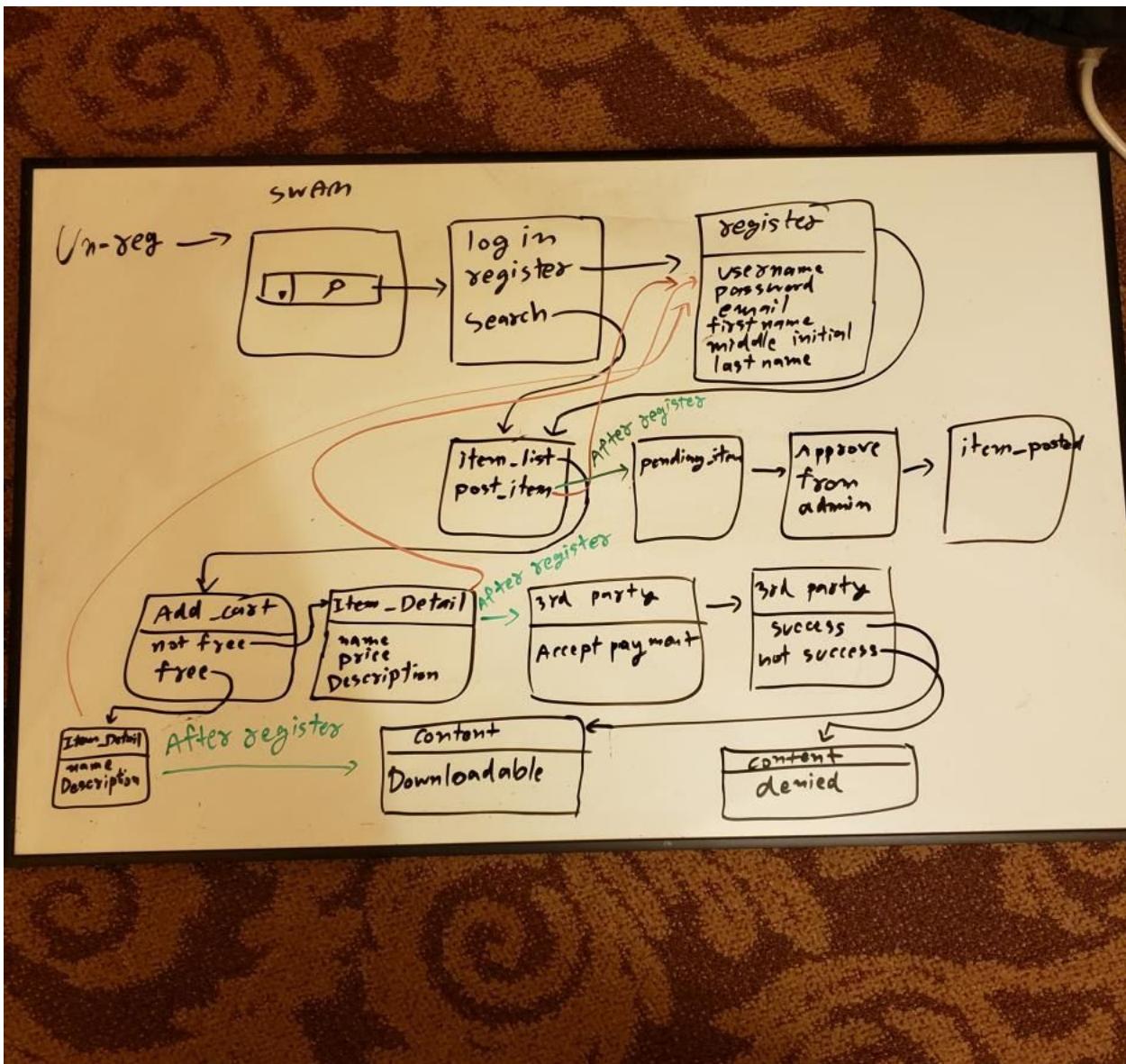
Server Host	Amazon Web Services
Operating System	Ubuntu v18.04
Database	MySQL v14.14
Web Server	Amazon EC2
Server-side Language	NodeJS v12.14.1
Front-end Framework	ReactJS v16.9.0
Front-end Framework	Redux v4.0.5
Back-end Framework	Express v4.16.1
Database Server Host	Google Cloud

5. We haven't changed any SW tools or Frameworks, but it has come to our attention that we had forgotten to include that our database is hosted on Google Cloud

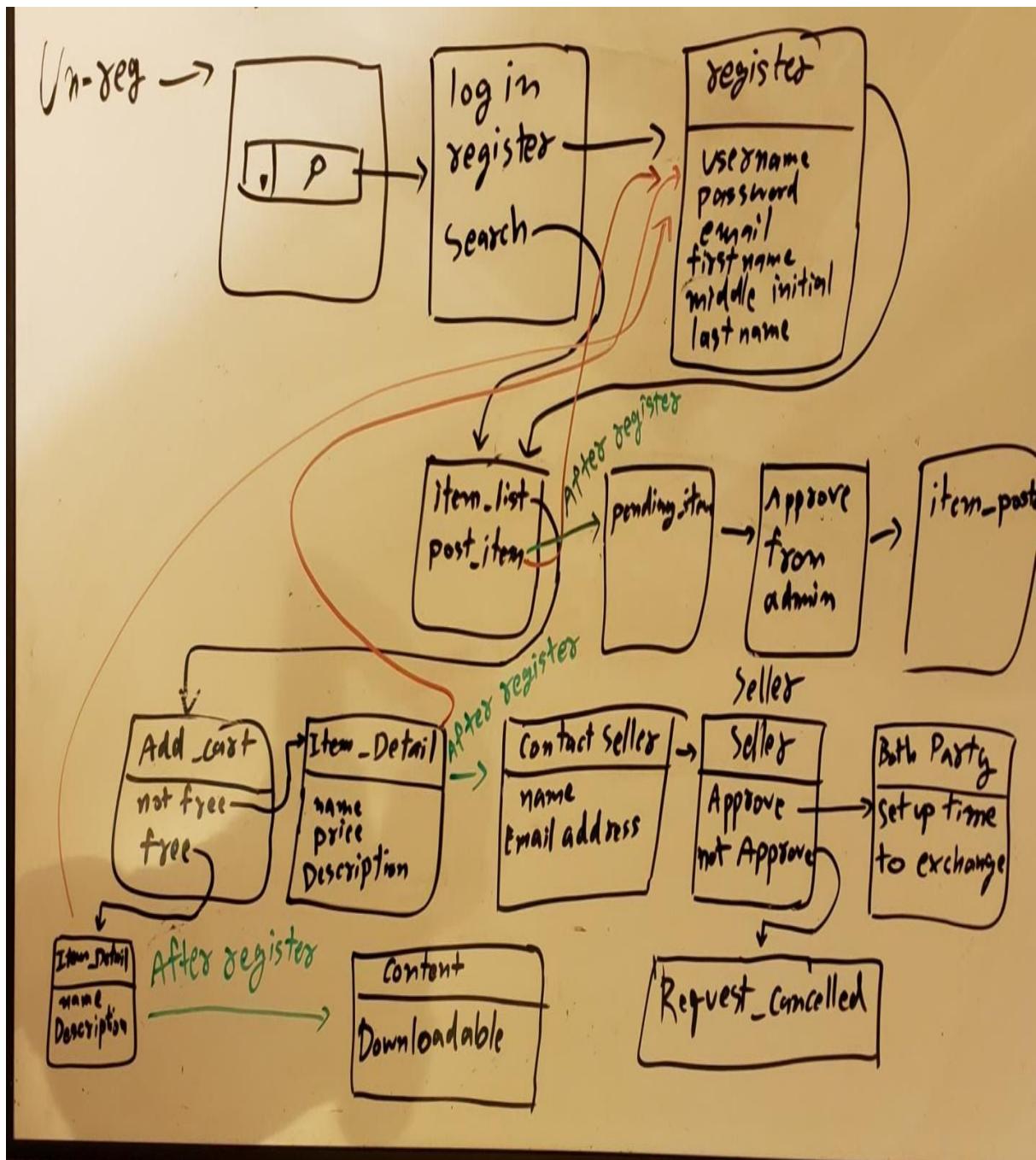
5. UML Diagrams

5 a) Class Diagram for each cases

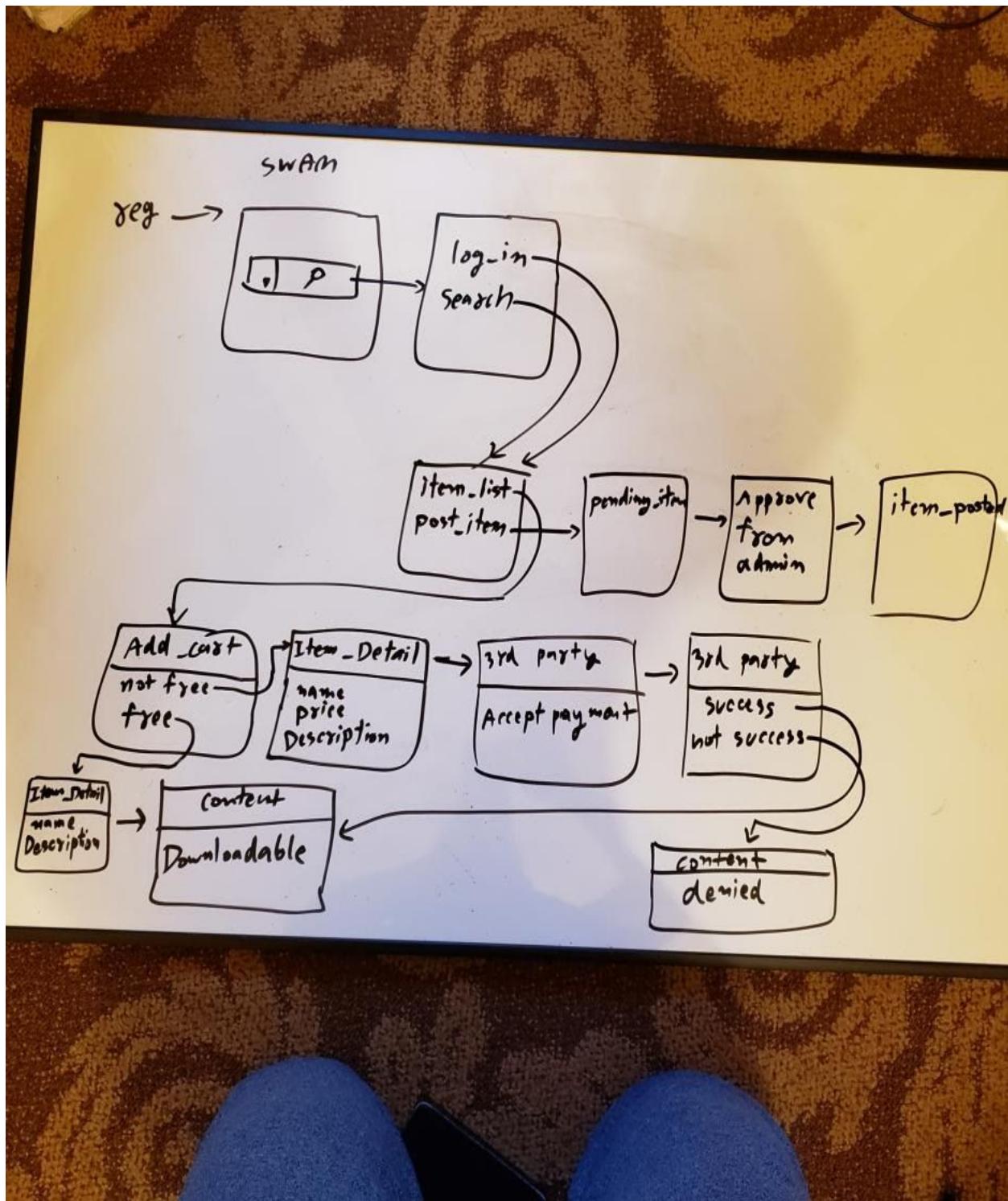
For unregistered users to buy digital media content



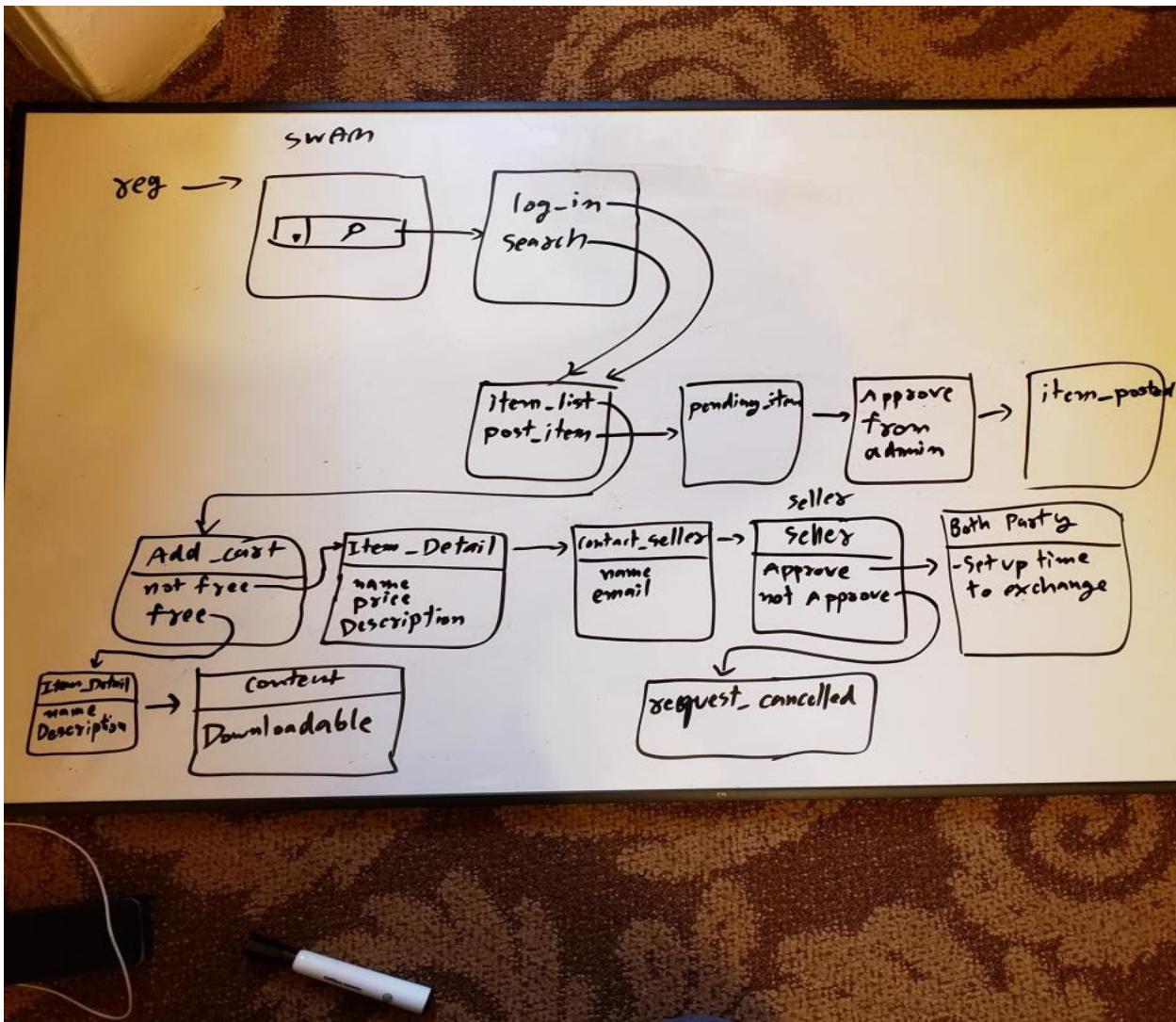
For unregistered users to buy physical media content



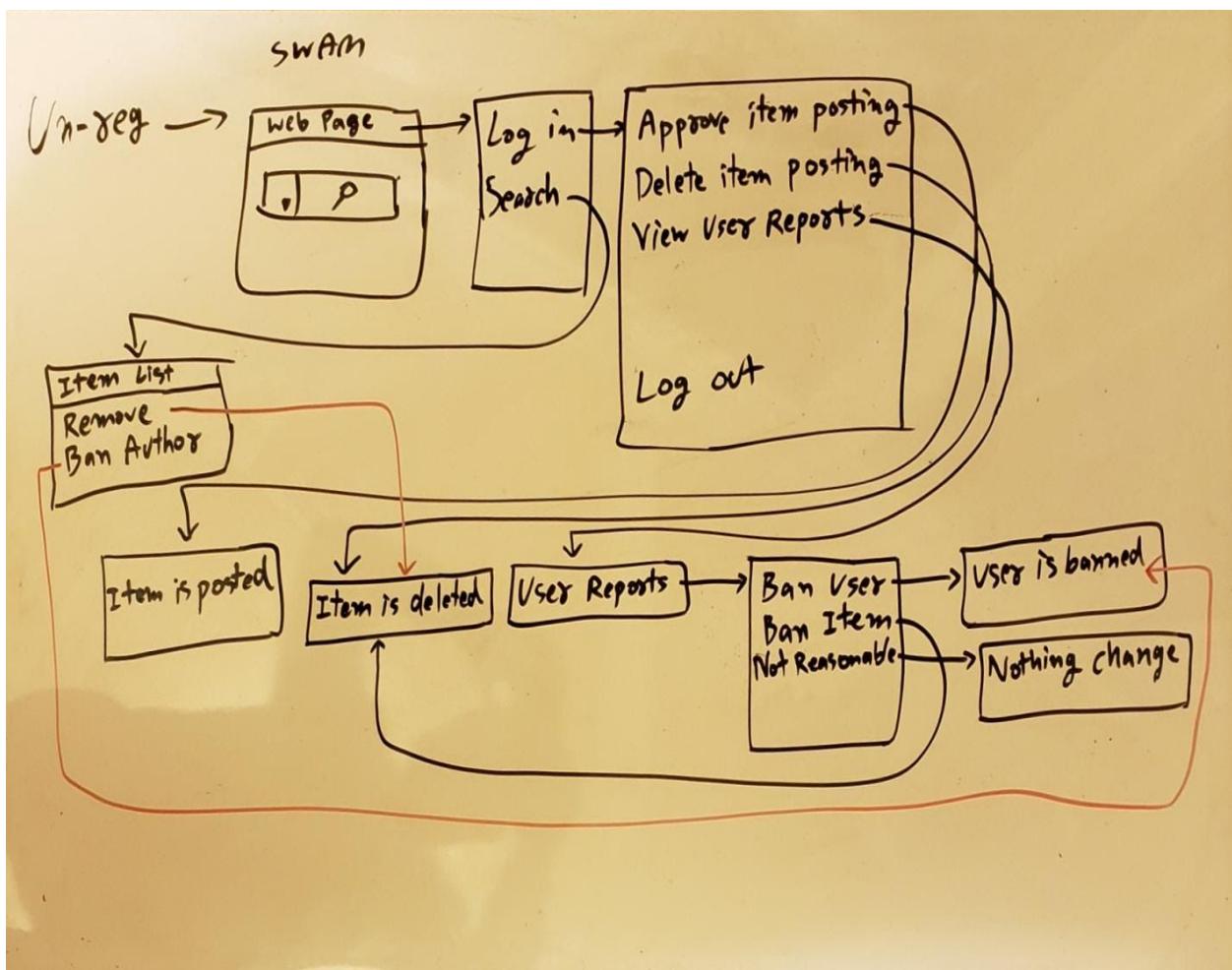
For registered users to buy digital media content



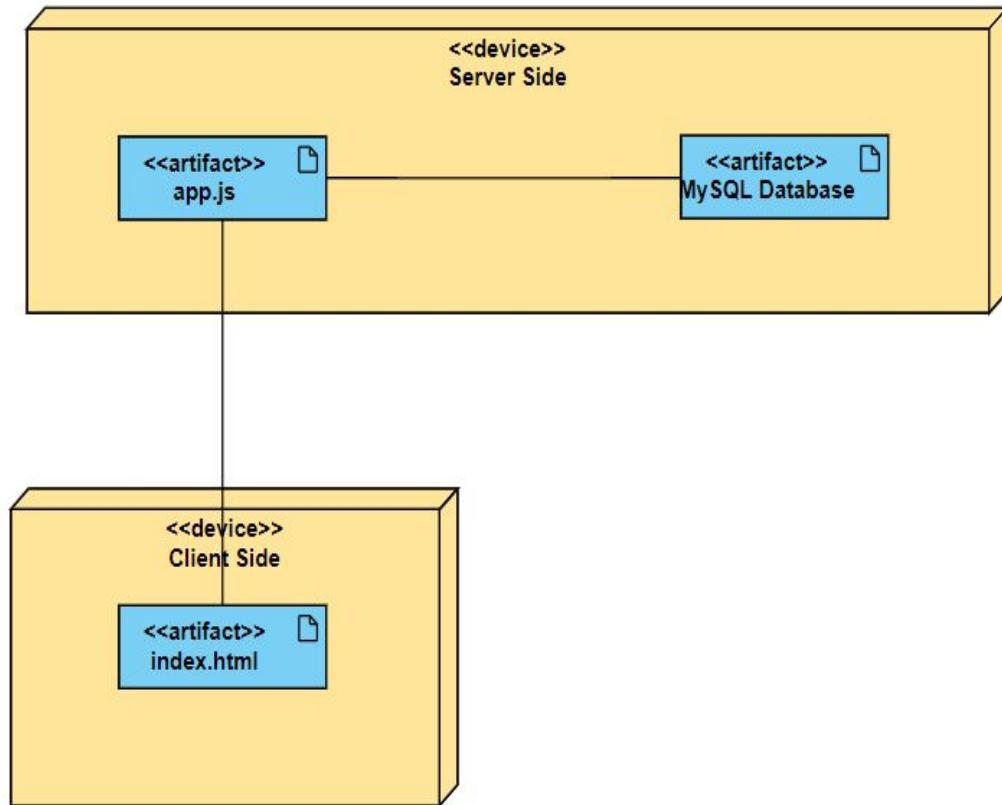
For registered users to buy physical media content



For Admin



5 b) Component and Deployment Diagram



6. Key Risks

Skills risks (do you have the right skills) - Everyone is qualified for their position.

Schedule risks (can you make it given what you committed and the resources) - Currently ON TRACK. No schedule risks.

Technical risks (any technical unknowns to solve) - Front end design and implementation maybe a bit unknown but currently no issues.

Teamwork risks (any issues related to teamwork) - Coronavirus

Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright) - No, Swamp uses free software. Admin take care of any inappropriate content.

7. Project Management

Currently, the entire Swamp team is connected through Discord text and voice chatting. Swamp also meets up over Zoom sessions once a week for class lectures. We also are split up into a front-end and back-end team. Each of those teams has leaders who understand what is currently the highest priority to work on. Each of the teams also has their own chatting channel for them to stay connected. Discord also gives the team the ability to screen share in order to have a presenter amongst the meetings.

Swamp also uses the application Trello to keep tasks organized by describing those tasks with cards. These cards can be organized in many ways such as priority, due date, etc. This way the team stays connected and up-to-date with what the project state currently is.

SWAMP

SW Engineering CSC648/848 Spring
2020

TEAM 04 Global - Milestone 03 - Feedback summary

March 19, 2020

-Feedback Summary-

-All of the below issues have been fixed-

Search bar must be accessible to all pages and stateless.

Logo should act like a button to home screen

There needs to be 1 to 3 lines of text on the site describing what the site is about

Unregistered users should not be able to download or post if they are not logged in.

Need to be showing the amount of items on the browsing page.

Colors and text on the site need to be easy to read.

If a posted item is free there needs to be a button called “download”.

If a posted item is not free then there needs to be a button called “Contact Seller”.

There needs to be a posting button on the user dashboard.

Login and sign up functions need alignment.

Sign up function needs a forgot password function and a terms and agreement checkbox (dummy link).

Commit messages need to be more descriptive such as say which file was changed, why, was the issue fixed?

Comments in code need more comments describing functions (only for things that are not intuitive).

SWAMP

SW Engineering CSC648/848 Spring
2020

TEAM 04 Global - Milestone 04

May 18, 2020

-Product summary-

Swamp, an online marketplace for the SFSU community by the SFSU community.

Committed functions:

1. Unregistered users **shall**:
 - I. Be able to register **DONE**
 - II. Be able to browse site content **DONE**
2. Registered users **shall**:
 - I. Be able to remove listing **IN PROGRESS (BACKEND DONE)**
 - II. Be able to post media for sale **DONE**
 - III. Be able to logout **DONE**
 - IV. Be able to login **DONE**
 - V. Be able to message media seller **IN PROGRESS (BACKEND DONE)**
 - VI. Be able to request to purchase **IN PROGRESS (BACKEND DONE)**
 - VII. Be able to approve purchase request **IN PROGRESS (BACKEND DONE)**
 - VIII. Be able to deny purchase request **IN PROGRESS (BACKEND DONE)**
3. Administrative users **shall**:

- I. Be able to login **DONE**
- II. Be able to logout **DONE**
- III. Be required to approve posting **DONE**
- IV. Be able to delete posting **DONE**
- V. Be required to view user reports **IN PROGRESS**
- VI. Be able to ban user **DONE**
- VII. Be able to unban user **DONE**

Our product contains a unique function that caters directly to SFSU students and faculty by means of utilizing a search feature that allows for site content to be searched by a class filter. This class filter allows users of Swamp to find/post items based on a specific SFSU class (i.e CSC648).

URL to product - <http://18.191.184.143:3000/>

- Usability test plan -

- Test objectives:

USE CASE: We are using Emma's use case (defined in earlier milestones) as a means of testing our user friendly site. We are trying to make sure that the site, upon entering, is easy to traverse so there is no confusion for the user. First when the site is loaded the user should be allowed to browse the page but to download an item the user should be taken to a registration page.

USER FRIENDLY: When creating our UI we made sure that our site was easy on the eyes (so the user can see and read everything with ease) so we picked out a color palate that would work and made sure all words on the website were a readable size font. This was a bit different than our UI mockups since those were a little crowded making it harder for a user to see what to do.

REGISTERING: Once signed in to the website the user will be registered and then taken to the login screen. Once logged in the user will be able to download content and contact the seller if the item they want to purchase is

not free. Also the registered user will now have a user dashboard to view their items that they posted as well as user data like email, name, username. Also once the user is logged in the login and sign up button disappears from the nav-bar. Replacing login and sign up on the nav-bar is the user dashboard and a logout button.

SITE INFORMATION: Also throughout the site we display messages that allow the user to understand what is going on. For example, if the user tried to sign up with an email that isn't ending with @mail.sfsu.edu they will be prompted to put in a proper email. Also upon successfully logging in there will be a message that tells the user that they were logged in successfully.

Test background and setup -

When starting the creation of the Swamp e-commerce site the team got together to discuss things we all found to be of vital importance to our site. We had many SCRUM meetings that resulted in us creating use cases for inevitable reasons and actions people might do on our site most commonly.

separate paragraphs each covering: System setup, starting point, who are the intended users, URL of the system to be tested ad what is to be measured (for M4 focus only on user satisfaction evaluation e.g Likert tests). Up to 1 page

Usability Task description: When going through usability testing, our team simulated a new user entering the site in order to download an item of their choosing. Upon entering the site the user is brought to the main page which displays our site name and explains briefly what our site is about. From there our users went directly to the browsing page where they were able to look through our sites posted items. They would also use the search bar in order to narrow their search for finding the right digital content they wanted. They will

need to register to our site if they want to download some of our site's content.

These are the instructions to be given to the tester: Write them in a separate paragraph in the format of instructions for the usability tester what to do e.g. describe the task testers do before filling out the Likert questionnaire and do the assessment.

	Rating 1-5 (1=Strongly Disagree, 3=Neutral, 5=Strongly Agree)				
Statements	User Responses				
I found the signup process easy to follow.	5	5	5	3	5
It is simple to browse the website.	5	5	5	5	5
Uploading content to the website is a simple process.	3	1	5	3	3
Downloading content from the website is easy.	5	5	5	3	5
The website is easy to use.	5	3	5	3	5
Overall, the website is built with an effective user interface.	5	5	5	3	5
Your comments are greatly appreciated. Please tell us what you like and how to improve our product.					

Our team gave site testing permissions to five different people in order to see what their responses were. Each person picked did not know a lot of computer science so that there was any bias.

- QA test plan -

Test objectives: - Search bar feature. Users shall be able to search for items from the search bar located in nav.

HW and SW setup (including URL):

Feature to be tested - The search bar should be able to filter the site's content from the user's input.

QA Test plan:

				Browser Results		
#	Description	Input	Expected output	Firefox	Chrome	Safari
1	Test %like in search bar	Type “yee” in search bar	Get 1 result, with all having the word “yee” in the title	PASS	PASS	PASS
2	Test specific type of media, images	Type “grass” in search bar and select “images” in dropdown	Get 2 results, all of image type having the word apple in the title	PASS	PASS	PASS

		menu				
3	Test whether item searched matches details	Type “grass” in search bar and select “images” in dropdown menu. Click the first image.	Get the details page of the item clicked. Title should match the card clicked in search. (Grass Field)	PASS	PASS	PASS

				Browser Results		
#	Description	Input	Expected Output	Firefox	Chrome	Safari
1	Test ability to upload to the website	Navigate to the Dashboard and Click the “+” icon to upload. Fill out the upload form.	Pending request for newly uploaded content will appear in Dashboard	PASS	PASS	PASS
2	Test ability to delete a post from the website (IN PROGRESS)	Navigate to the Dashboard view “Current Posts.” Click on the Delete button of	Current Posts field in Dashboard will be empty.	FAIL	FAIL	FAIL

		the post that was uploaded in previous test				
3						

- Code Review -

Coding Style:

Our style consists of a few different things:

1 - All of our variables, class names, function names, and file names are descriptive to what their purpose is. Our team follows an OOP (Object-Oriented Programming) style meaning each function, class, file, and variable is consistent to what its purpose is. Each function encapsulates what the purpose of the feature is to the corresponding classes, files and variables that pertain to that feature. For example, our login feature will have nothing to do with browsing the site though it will define variables that pertain to a wider scope. This OOP style makes it easy for our team to traverse our code so that the team is able to easily identify the purpose of certain sections. This allows the team to maximize the attempts to pinpoint areas the team works on like finding bugs, usability testing, adding features, etc.

2 - Syntax: All of our variables, files, functions and classes follow a camel-case syntax (exampleVariable = true).

-Code review Sample-

Issue with search bar not displaying site content



Benjamin Dunbar Lewis

Mon 5/18/2020 9:23 PM

To: oawami@gmail.com
Cc: Benjamin Dunbar Lewis

↶ ⚡ ⏪ ⏴ → ⋮

Hi Onu,

When I was testing our search bar feature I noticed that it is no longer displaying the content of the site upon user searching. I believe the issue is somewhere inside of the navbar file that I will attach.

While you're reviewing the issue it would be useful if you could also leave some comments explaining what certain sections of the code are doing. Also, leave some comments on what you change. This way the issue won't happen again

Starting at line 93 of Navbar.js -

Best,
Benjamin Lewis

...

```
<SearchBar className="container justify-content-center">
    <div className="input-group mb-2">
        {/* <DropDown className="input-group-prepend"> */}
        <select>
```

```
        id="category"
        onChange={(e) => setCategory(e.target.value)}>
        <option value="all" selected>All</option>
        <option value="document">Documents</option>
        <option value="image">Images</option>
        <option value="audio">Audio</option>
        <option value="video">Video</option>
    </select>
    {/* </DropDown> */}

    <input
        id="userInput"
        class="form-control"
        type="text"
        aria-label="Text input with dropdown button"
        placeholder="Search by title..."
        value={query}
        onChange={(e) =>setQuery(e.target.value)}
    ></input>

    <button
        className="btn btn-dark"
        onClick={() => {
            value.setProducts(category, query);
            history.push("/result");
        }}
    >
        <i class="fas fa-search"></i>
    </button>
</div>
</SearchBar>
```

Issue with search bar not displaying site content



Onubulachi Wami <oawami@gmail.com>

Mon 5/18/2020 9:45 PM

To: Benjamin Dunbar Lewis



Hello Ben,

Thank you for notifying me about the issue with the search bar. I will take a look into it as it is a critical issue. The issue was most likely produced when redesigning the navigation bar, and moving it out of the home page. I went ahead and added some comments to the code.

Please let me know of any other issues you find.

```
``` /* Search bar */
/* Search bar is using bootstrap so the size will shrink when the browser
shrinks in order to accommodate mobile devices (flex-shrink-1). The search bar
also follows common practice in websites by always being present in the
navbar and being centered. The dropdown menu allows the user to select
a specific category for searching content on the website. */
{ navSearch ?
<ProductConsumer className="flex-shrink-1 align-content-center">
{value) => (
<SearchBar className="container justify-content-center">
<div className="input-group mb-2">
<select
```

```
id="category"
onChange={(e) => setCategory(e.target.value)}>
<option value="all" selected>All</option>
<option value="document">Documents</option>
<option value="image">Images</option>
<option value="audio">Audio</option>
<option value="video">Video</option>
</select>

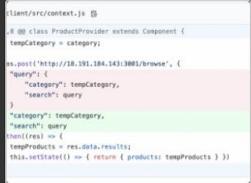
<input
id="userInput"
class="form-control"
type="text"
aria-label="Text input with dropdown button"
placeholder="Search by title.."
value={query}
onChange={(e) => setQuery(e.target.value)}
></input>
/* Search results are generated through setProducts which is located
in src/context.js. This takes the query from the searchbar and generates an array of items
related to the query
using %like on the title of the items posted to the website. The
generated array will display
```

```
on the /results page when redirected.*/}
<button
 className="btn btn-dark"
 onClick={() => {
 value.setProducts(category, query);
 history.push("/result");
 }}
>
 <i class="fas fa-search"></i>
</button>
</div>
</SearchBar>```
...

```

Issue with search bar not displaying site content

OW Onubulachi Wami <oawami@gmail.com>  
Mon 5/18/2020 11:07 PM  
To: Benjamin Dunbar Lewis



Hello Ben,  
It turns out that some lines were edited in context.js that was preventing the %like feature in search from working properly. These changes were added and the search function is now working as intended.  
See attached.

...

## -Security-

### -Protecting user records

**Password encryption:** In order to make sure our users are protected we used bcryptjs in our backend. Bcrypt is a tool for allowing credentials like passwords to be secured in our database. It turns the user's password into an encrypted hash value that is essentially generated bytecode that can only be reversed through its decryption. The hash value created is sort of like a virtual

fingerprint that is unique so even if two passwords are the same the hash value will be different.

- **Threats:** Just there are people who create code for the good of others there are always those who try to exploit it. The hackers have many methods of exploiting protected things like passwords called attacks. Our encryption algorithm will protect against attacks like rainbow table attack, brute force, Man in the Middle, Drive-by attack, Cross-site scripting, etc. The list goes on and on but with encryption our team is able to keep even well equipped hackers at bay.

**Named Placeholders:** To prevent our database from being the victim of a malicious attack and putting our users' information at risk, we've adopted the usage of named placeholders in our sql queries. Using named placeholders is akin to replacing all the parameters in the brackets with their respective positions in the query string and escaping them.

### **-Protecting media records**

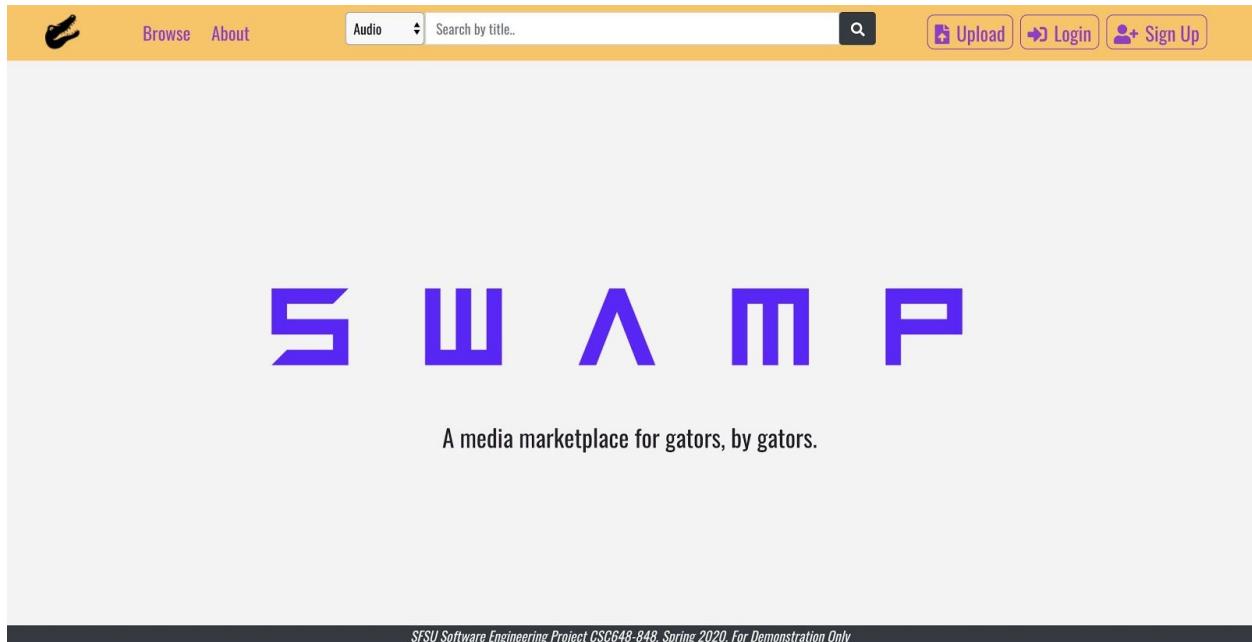
**Search bar validation:** Inside off our search bar we only allow users to be able to type up to 40 characters. Among these 40 characters only alphanumeric characters are allowed to be entered. This ensures that only properly formed data is entering the search bar. This will prevent malformed data from entering our database that could potentially give our site malfunctions. Only allowing alphanumeric data to enter our site is a form of syntactic validation which forces inputs from the user to enter correct syntax. We correct any data before it gets too deep into our code so that the site can spot errors early before causing damage.

## **-Non-functional specs-**

- I. 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). **DONE**
- II. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers. **DONE**
- III. Selected application functions must render well on mobile devices. **ON TRACK**
- IV. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**
- V. Full resolution free media shall be downloadable directly, and full resolution media for selling shall be obtained after contacting the seller/owner. **DONE**
- VI. No more than 50 concurrent users shall be accessing the application at any time. **DONE**
- VII. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
- VIII. The language used shall be English (no localization needed). **DONE**
- IX. Application shall be very easy to use and intuitive. **DONE**
- X. Google analytics shall be used. **DONE**
- XI. No email clients shall be allowed. **DONE**
- XII. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**
- XIII. Site security: basic best practices shall be applied (as covered in the class) for main data items. **DONE**
- XIV. Media formats shall be standard as used in the market today **DONE**
- XV. Media material shall be either free or for sale, as determined by the media owner. **DONE**
- XVI. Each media material shall have its license info as one of the following: a) free use and modification; b) free but only allowed for SFSU related projects; c) for sale. **DONE**
- XVII. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. **DONE**
- XVIII. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). **DONE**

## -Project Demo-

### Home Page -

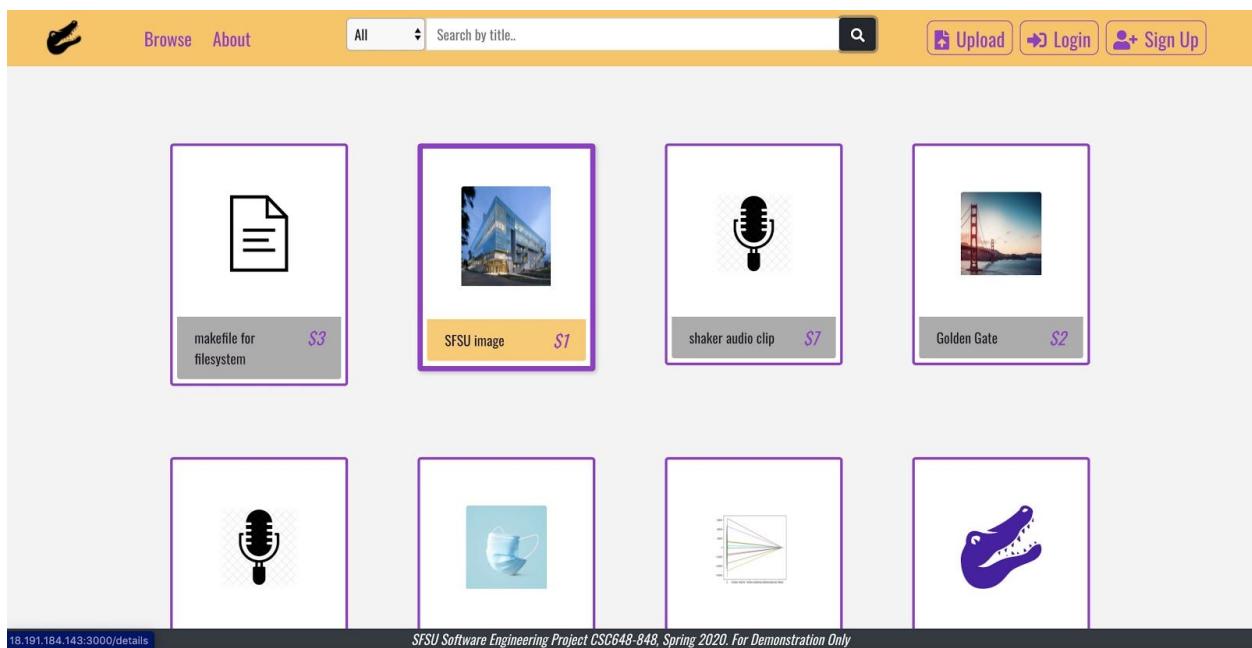


The screenshot shows the SWAMP home page with a yellow header bar. From left to right, it contains: a logo icon, 'Browse' and 'About' links, a dropdown menu set to 'Audio', a search bar with placeholder 'Search by title..', a magnifying glass icon, 'Upload', 'Login', and 'Sign Up' buttons.

The main content area features large purple letters spelling 'SWAMP'. Below them is the tagline 'A media marketplace for gators, by gators.'

A dark grey footer bar at the bottom contains the text 'SFSU Software Engineering Project CSC648-848, Spring 2020. For Demonstration Only'.

### Browse Page -



The screenshot shows the SWAMP browse page with a yellow header bar identical to the home page's.

The main content area displays four items in a grid:

- Icon: document, Title: 'makefile for filesystem', Status: '\$3'
- Icon: building, Title: 'SFSU image', Status: '\$1'
- Icon: microphone, Title: 'shaker audio clip', Status: '\$7'
- Icon: bridge, Title: 'Golden Gate', Status: '\$2'

Below this grid, there are two more items:

- Icon: microphone, Title: '' (empty)
- Icon: cup, Title: '' (empty)

A dark grey footer bar at the bottom contains the text '18.191.184.143:3000/details' and 'SFSU Software Engineering Project CSC648-848, Spring 2020. For Demonstration Only'.

## Sign Up Page -

The screenshot shows a sign-up form titled "Sign-Up". The form includes fields for Username, Password, Confirm Password, SFSU e-mail address, First Name, and Last Name. There is also a checkbox for "Terms and Conditions" and two buttons at the bottom: "Submit" and "Cancel".

SFSU Software Engineering Project CSC648-848, Spring 2020. For Demonstration Only

## User Dashboard -

The dashboard displays the user's profile information: Status: Student. Below the profile picture, there are links for Posts, Requests, and Purchases. The user's details are listed as: Username: ben, Full Name: Benjamin Lewis, Email: ben@mail.sfsu.edu. A large blue plus sign icon indicates a new post can be created. To the right, there is a thumbnail image of a modern glass building and a card featuring the SF State Gator logo with the text: Title: do not approve, Price: \$2, Status: rejected, and a red Delete button. Another card features the SF State Gator logo with the text: Title: Gator, Price: \$1, Status: approved, and a red Delete button.

Status: Student

Posts Requests Purchases

Username: ben  
Full Name: Benjamin Lewis  
Email: ben@mail.sfsu.edu

+ 

Title: do not approve  
Price: \$2  
Status: rejected

Delete

  
Title: Gator  
Price: \$1  
Status: approved

Delete

## **-Backend Database Tables-**

### **All tables used:**

```
mysql> show tables;
+-----+
| Tables_in_mydb |
+-----+
| accounts
| admins
| approved media
| approved requests
| banned users
| category
| digital media
| media content
| message box
| message requests
| physical media
| registered users
| rejected media
| rejected requests
| reported media
| reported users
| shopping cart
| wish list
+-----+
18 rows in set (0.09 sec)

mysql> █
```

### **Users table:**

acc_id	username	password	email	first_name	last_name	profile_path
1	bob	\$2a\$10\$.AALiy9gudohxjnMktnbc.3t5vtCMo8dcItRrRcyR7zhztknvr8fo	bob@mail.sfsu.edu	bob	bob	profile/default/profile.png
2	koala	\$2a\$10\$hXA2628QfUlf0qjsz5U3.ovpxL5s3mjGktD7.mruPA45/f/ZC	koala@mail.sfsu.edu	koala	koala	profile/default/profile.png
3	williamlew	\$2a\$10\$NngZG0ahc/sqj0wbsuwlM.STrnchbzgh2reM9RSRW9lj42Zxe	lew@mail.sfsu.edu	william	lew	profile/default/profile.png
4	ben	\$2a\$10\$fqkPbEAkBvYlt1BygYKq7ejo57.NmQegmzf/RG85NfairPKau6mBm	ben@mail.sfsu.edu	Benjamin	Lewis	profile/default/profile.png
5	neb	\$2a\$10\$kaVaycZihIu5nfEOQuwi/g9tF4e.WjDaSRyxlignZwGxIK	neb@mail.sfsu.edu	neb	neb	profile/default/profile.png
7	yee	\$2a\$10\$ENLQDXMMmtEzbF3AXnNoaStoebEsrJd930bf6wCrnA1tfRbvq	yee@mail.sfsu.edu	yee	yee	profile/default/profile.png
8	yeet	\$2a\$10\$RUIELS5XrDoesOx4xFkyeh1GN3z1BK8R5UOvuueEVQAFNgxmrvu	yeet@mail.sfsu.edu	yeet	yeet	profile/default/profile.png
9	kevin	\$2a\$10\$5n.tbmB8Pi30vMzrSHn8uGzjnr3ACUMBu.g4khw8uk.tMwazEs7a	something@mail.sfsu.edu	Kevin	Huynh	profile/default/profile.png
10	onu	\$2a\$10\$FVGWstH/xmU73gr51aeGz9ctxs2k7Ny1dTfTMFaGezQZz6RWe	owami@mail.sfsu.edu	Onubulachi	Wami	profile/default/profile.png
11	test1	\$2a\$10\$N.AqWffaafhb1A1RMe.Qun.XNe/dvMmlrFCFBG0hnz/Duick7pGm	test@mail.sfsu.edu	tester	testing	profile/default/profile.png
23	abc	\$2a\$10\$VrhUb1raMzr9fQmgsfed01iMwdctepDnkP83F4a2TPholZD2	abc@mail.sfsu.edu	abc	abc	profile/default/profile.png
24	signup	\$2a\$10\$N4ZpSnc9cnHwm7Dli0t2u6Cq7uc5Mt0k8/DBkgR8J0rvzRaV1Z	signup@mail.sfsu.edu	signup	signup	profile/default/profile.png
25	random	\$2a\$10\$6..9MBH2sMNndbvuldp6D.zcaVeB1519Mh40vJBMEQcxSmzFLt9e	random@mail.sfsu.edu	random	random	profile/default/profile.png
26	genji	\$2a\$10\$8/H19Te8VR01Dr1eKw9.VesAaRa60/umXh4q1kD1KXjf7AMTs	genji@mail.sfsu.edu	genji	khan	profile/default/profile.png
27	hanzo	\$2a\$10\$9lC1s8sDmgfRkw5b/EJB50poEQxshgyjC68uTGE0h148z.pFH.	hanzo@mail.sfsu.edu	hanzo	ninja	profile/default/profile.png
28	abcd	\$2a\$10\$NDuxcoa0x1N9rOzzsNNOAG.ZH8.f22m9yrtvF5jov8DQz6RWe	abcd@mail.sfsu.edu	abcd	efjk	profile/default/profile.png
29	efg	\$2a\$10\$WBWkdkh4kUzVS1mJSJ9twe92Saasagg9sielhIA3UifMRBRctf.7K	efg@mail.sfsu.edu	efg	hij	profile/default/profile.png
30	abcde	\$2a\$10\$5VrhUb1raMzr9fQmgsfed01iMwdctepDnkP83F4a2TPholZD2	abcde@mail.sfsu.edu	Abcde	Abcde	profile/default/profile.png
31	tired	\$2a\$10\$Jj824.y3m3j1kng0ejR3eu2w.TUhLsb/0893jyv0R1ibKb.QYQnW	tired@mail.sfsu.edu	tired	af	profile/default/profile.png
32	ffff	\$2a\$10\$30F5rYnjpu33oYcJaMo.emJlitxnc21AA2XL.4Lu4coWwyCilY	ffff@mail.sfsu.edu	ffff	gggg	profile/default/profile.png
33	gggg	\$2a\$10\$5VrhUb1raMzr9fQmgsfed01iMwdctepDnkP83F4a2TPholZD2	gggg@mail.sfsu.edu	gggg	gggg	profile/default/profile.png
34	anotherone	\$2a\$10\$9ryjyOH4t2Ms5MdFdjtuf9wzXp5vb7jfbHToID.PqM01AGkRq	anotherone@mail.sfsu.edu	anotherone	one	profile/default/profile.png
35	abigail	\$2a\$10\$3/pDCDVfSyxd107zuNjq.3379p1u5SF5t7WM/MKpao.Glo.	abigail@mail.sfsu.edu	abigail	abi	profile/default/profile.png
36	wami	\$2a\$10\$6DN80geq9adHOwIC107zuNjq.3379p1u5SF5t7WM/MKpao.Glo.	wami@mail.sfsu.edu	wami	abi	profile/default/profile.png
37	aaron	\$2a\$10\$5UK2XoZ.qdgY9ePRg7H.YejzaqSwXzvWutvKhWqqb201mOnQicMIDLE	aaron@mail.sfsu.edu	aaron	carter	profile/default/profile.png
38	www	\$2a\$10\$mbkUjaf6.Y229Sgl1t1B/unCgD0mRpgsMc08KPFfTMdA8grumW	www@mail.sfsu.edu	www	xxxx	profile/default/profile.png
39	jjjj	\$2a\$10\$5zvusNDD0Qd3vshug8wtsuKLJCLpb1uvyEvMwGssPlJvxbetSQtI	jjjj@mail.sfsu.edu	jjjj	jjjj	profile/default/profile.png
40	kkkk	\$2a\$10\$dpPxvunuzHuxgR.fladx90LN3zJzWzdrEsRnbp4n4NbvnjekYHa	kkkk@mail.sfsu.edu	kkkk	kkkk	profile/default/profile.png
41	ak	\$2a\$10\$773UMPxvXFdeuUuXnjq0UNfNsBaP7DPGcaVF1szF91Rw6sdzmcQm	ak@mail.sfsu.edu	ak	ak	profile/default/profile.png
42	akk	\$2a\$10\$5Se11HfdAawsLdgv8.lhUoJopph9cu60k2npEPa98KCh0e.p6e8W	akk@mail.sfsu.edu	akk	akk	profile/default/profile.png
43	aite	\$2a\$10\$c1bzcISAbhqeqKtsJ0UmngemJ6nXklGDks5s5v4p3ypUte1ZmDTRY	aite@mail.sfsu.edu	aite	aite	profile/default/profile.png
44	pppp	\$2a\$10\$wssoz/qWlnx/5/JM0j//ek8d9j0aREu/aGjUB3WCmIk3QMT/k/.	pppp@mail.sfsu.edu	pppp	dodo	profile/default/profile.png
45	dddd	\$2a\$10\$5AV0Sqbk7xd3q8uKDeumf47UgFcVjhGlo18pnMhWuqxOp8ole	ddd@mail.sfsu.edu	ddd	ddd	profile/default/profile.png
46	1111	\$2a\$10\$tpzaGfm62QlBB1gZ5eeUuxn0pnaEug/pOs.JUBUe161hFlduNG	1111@mail.sfsu.edu	1111	1111	profile/default/profile.png
47	oooo	\$2a\$10\$5t.Ot.D1f0PA/w2v7VQ0unxJndnsttndjD8371giDakfLNKnZ	oooo@mail.sfsu.edu	oooo	oooo	profile/default/profile.png
48	ooo	\$2a\$10\$08ZsXAPj0zGftQ0s6Fne.7VQhrMVUPk2zqXapAUj9u22K5Ed.cm	ooo@mail.sfsu.edu	ooo	ooo	profile/default/profile.png
49	uuu	\$2a\$10\$7rl77B0/b3XnID1hpJtw.74dgGxjv22ZfFQor02LA0Mmg.dk7p1u	uuu@mail.sfsu.edu	uuu	uuu	profile/default/profile.png
50	dang	\$2a\$10\$RVgAgOucCmBlh1dkIAkz.sThfvPGhs1d0gvf5u5HFPUwAK6Usy	dang@mail.sfsu.edu	dang	le	profile/default/profile.png
51	boo	\$2a\$10\$dfxLa7cjbvixhksiegau4027Ft6tKCTTVQ8gnvswGTwq.SMiQw	boo@mail.sfsu.edu	boo	lee	profile/default/profile.png
52	joe	\$2a\$10\$K.x1jJ1hmTSJTxJ78ULTeJt9218ruyNcbYhNqcpMTTzgeFywqGqq	xyz@mail.sfsu.edu	weerachai	poorakkiat	profile/default/profile.png
53	pokemon	\$2a\$10\$/lnCKS6GOX0x3mbc/cHy.sgh9n0IREvt7d6n04L5dBCCL15Mn44C	asdfr@mail.sfsu.edu	Kevin	Huynh	profile/default/profile.png

## Categories table:

```
mysql> select * from category order by category_id asc;
```

category_id	category
1	document
2	image
3	video
4	audio

```
4 rows in set (0.09 sec)
```

```
mysql> |
```

## Administrators table:

```
mysql> select * from admins;
+-----+-----+
| admin_id | acc_id |
+-----+-----+
| 1 | 2 |
| 2 | 9 |
+-----+-----+
2 rows in set (0.09 sec)

mysql>
```

### Approved Media table:

```
mysql> select * from `approved media`;
+-----+-----+-----+-----+
| approved_id | m_id | status | status_by |
+-----+-----+-----+-----+
| 3 | 3 | approved | 1 |
| 4 | 4 | approved | 1 |
| 5 | 5 | approved | 1 |
| 6 | 6 | approved | 2 |
| 9 | 9 | approved | 2 |
| 10 | 10 | approved | 2 |
| 11 | 13 | approved | 2 |
| 12 | 14 | approved | 2 |
| 13 | 15 | approved | 1 |
| 14 | 11 | approved | 1 |
| 15 | 12 | approved | 1 |
| 16 | 16 | approved | 1 |
| 17 | 17 | approved | 1 |
+-----+-----+-----+-----+
13 rows in set (0.08 sec)
```

### Rejected Media table:

```

mysql> select * from `rejected media`;
+-----+-----+-----+-----+
| rejected_id | m_id | status | status_by |
+-----+-----+-----+-----+
| 1 | 18 | rejected | 2 |
| 2 | 19 | rejected | 2 |
+-----+-----+-----+
2 rows in set (0.09 sec)

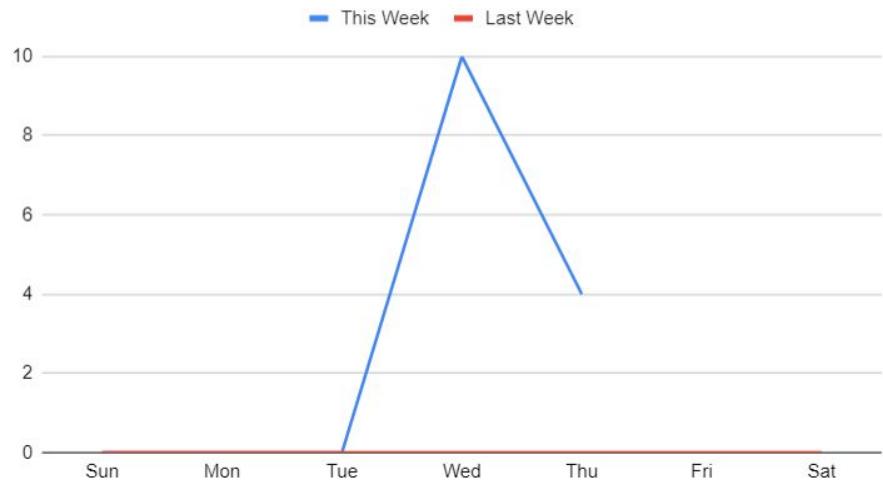
```

mysql>

## -Google Analytics-

The screenshot shows the Google Analytics interface. On the left, there's a sidebar with navigation links like Home, Customization, Reports, Realtime, Audience, Acquisition, Behavior, Conversions, Attribution (Beta), Discover, and Admin. The main area displays real-time metrics: 6 users, 10 sessions, 10% bounce rate, and a session duration of 11m 44s. Below this is a line chart showing user count over time from May 14 to May 20. A tooltip for May 20 indicates 6 users. To the right, there's a blue box for 'Active Users right now' showing 0, and another for 'Page views per minute' showing 1. A callout box highlights 'No data available' for 'Top Active Pages'. At the bottom, there are links for 'AUDIENCE OVERVIEW' and 'REAL-TIME REPORT'.

## This Week and Last Week



## -Project Management-

### Trello:

A screenshot of a Trello board titled "Milestone 2". The board has five columns: To Do, Doing, Code Review, Testing, and Done. Each column contains several cards with task details. The board is set to public.

- To Do:**
  - Create "My listings" back-end points
  - Add website cosmetics such as image sizing, colors, fonts and formats
  - Test new login/register front end page with back-end db
  - forgot password link
  - + Add another card
- Doing:**
  - Create "My purchases" back-end points
  - M4 documentaion
  - + Add another card
- Code Review:**
  - Dashboard for User Profile
  - Persistent login
  - + Add another card
- Testing:**
  - M2 Document Creation (Mar 19)
  - Revamp Front End
  - Create Login/ Register feature
  - Database/ Architecture
  - + Add another card
- Done:**
  - Add Redux
  - Implement Local Cart (4/4)
  - Convert BE team's client to React
  - UI Mockups and Storyboards (Mar 19)
  - Risks for part 6
  - UML for part 5
  - + Add another card

This board is set to public. You can change its visibility at any time. [Learn more here](#)

**Milestone 2**

**Design**

- Dashboard for User Profile
- + Add another card

**To Do**

- Create "My listings" back-end points
- Add website cosmetics such as image sizing, colors, fonts and formats
- Test new login/register front end page with back-end db
- Export Searchbar into NavBar to show up on every page
- Persistent login
- forgot password link
- + Add another card

**Doing**

- Create "My purchases" back-end points
- don't have account link on login
- Fix /browse post to allow for multiple pages of results
- Add checkbox for "terms and conditions" on signup. Create link for terms page
- + Add another card

**Code Review**

- Create Login/ Register feature
- Revamp Front End
- + Add another card

**Testing**

- M2 Document Creation
- Database/ Architecture
- + Add another card

**Milestone 2**

**Doing**

- Adding upload to dashboard and clean up download
- + Add another card

**Code Review**

- + Add a card

**Testing**

- Dashboard for User Profile
- Working on download and message seller
- Add website cosmetics such as image sizing, colors, fonts and formats
- Fix /browse post to allow for multiple pages of results
- + Add another card

**Done**

- Add Redux
- Create "My listings" back-end points
- Create "My purchases" back-end points
- Take cookies out of front end
- M4 documentaion
- forgot password link
- Part 6 (Self check) of M4 progress... Change anything to either DONE, IN PROGRESS, or ISSUE
- Database/ Architecture
- Persistent login
- + Add another card

## -Team 04 Self Assessment-

### Benjamin Lewis:

 Benjamin Lewis <benjaminlewis984@gmail.com>  
to m2tour, oawami, khuynh2, WilliamMilesLew, dang.le730 ▾

2:21 PM (11 minutes ago)



Hi Team,

I am sending this in regards to our Milestone 5 self-assessment. Below I have described my contributions, GitHub submission amount number, difficulties I have had, and what I would have done differently.

a) As the team lead, I have contributed to the project by:

- Organizing meetings
- Focusing the team on the task at hand or internal and external requirements
- Coordinating team logistics
- Communicate the team's progress, status, accomplishments, and project direction
- Organize project management for a common vision
- Use of humor/ be firm but understanding
- Balance workloads

b) Total commit to GitHub - 43

c) Some difficulties in contributing to the project I had:

- Figuring out accurate priorities
- Working on front end communicating with the backend
- Working on creating the milestone documentation to be accurate and uniform.

d) If I were to do something different next time I would have gotten to work quicker on the actual coding of the project itself. I would also have stuck to the priorities ones closer from the beginning.

Thank you all for your wonderful contributions to Team 04 and all of the countless hours you have spent working on the vision. I could not have asked for a better team to work with. As difficult as this process was at some points I feel that our rewards for hard work made it all worth it. Swamp is nothing without all of us.

Best,  
Benjamin Lewis

### William Lew:

#### Self Assessment - William Lew

 William Lew <williammileslew@gmail.com>  
to dang.le730, benjaminlewis984, m2tour, oawami, khuynh2 ▾

5:19 PM (0 minutes ago)

Hello Team,

I am sending this in regards to our Milestone 5 self-assessment. Below I have described my contributions, GitHub submission amount number, difficulties I have had, and what I would have done differently.

a) As the back-end lead, I have contributed to the project by:

- Assigning my top back-end engineer tasks
- Organizing back-end meetings to discuss back-end content
- Organizing back-end meetings with the front-end to try to integrate them together
- Created the foundation to the back-end server
- Created a documentation page containing the server's API to allow for the easy integration with the front-end
- Helped front-end with back-end integration
- Maintained the AWS virtual machine

b) Total commits to GitHub - 157

c) Some difficulties in contributing to the project I had:

- It was difficult navigating the client directory and trying to determine what file managed what entity on the website. Trying to understand the front-end's Redux was confusing due to the complexity of Redux itself.

d) If I were to do something different next time, I would try to create a foundation for the front-end right after I created the foundation to the back-end just to simplify the integration of the back-end for the front-end team.

Thank you all for the great experience.

William Lew

## Weerachai Poorakkiat (Joe):

### Self Assessment. Weerachai Poorakkiat (Joe)



Ordinary Boy ...

Thu 5/21/2020 5:02 PM

To: dang.le730@gmail.com; WilliamMilesLew@gmail.com; khuynh2@mail.sfsu.edu; benjaminlewis984@gmail.com; oawami@gmail.com



Hi Swamp,

I am sending this regarding to our Milestone 5 self-assessment.

Below I have described my contributions, GitHub submission amount number, difficulties I have had, and what I would have done differently.

a) As the git master, I have contributed to the project by:

- Organizing zoom meeting to help each other to work on project
- Helping git (respiratory) issue when it happens,
- Organizing branches to help for each team
- Helping team lead for any tasks like his secretary
- Helping frontend tasks

b) Total commit to GitHub – 45

c) Some difficulties in contributing to the project I had:

- Working on tasks on frontend. Some function has to spend hours to figure it out to make it work. Sometimes we accidentally spend most of time on p2 which is not important to do like p1. So, I would say sometimes we focus on the wrong priority.

d) If I were to do something different next time, I would try to focus on the most important thing first (P1) other than P2. Study more technologies to implement and make the progress faster.

Thank you, guys. Without you, SWAMP won't be happened.

Respectfully,

Weerachai Poorakkiat (Joe)

## Kevin Huynh:



Kevin Huynh

Thu 5/21/2020 5:58 PM

To: williammileslew@gmail.com; oawami@gmail.com; m2tour@hotmail.com; Benjamin Lewis (via Google Docs) <benjaminlewis984@gmail.com>



Hi Swamp,

I am sending this regarding to our Milestone 5 self-assessment.

Below I have described my contributions, GitHub submission amount number, difficulties I've had, and what I would have done differently.

a) As a back-end team member, I have contributed to the project by:

- Designing the ERD and the database model, and the creation of the database
- Maintaining and creating the database whenever needed
- Implementing back-end endpoints
- Providing comments and documentation on various endpoints
- Refactored the entire back-end to keep up to date with ES6 standards and to prevent SQL injection
- Finding and fixing a number of problems that arose during implementation and testing
- Providing help to the front end when trying to integrate with the back-end

b) Total commits to Github - 117

c) Some difficulties in contributing to the project I had:

- It was difficult juggling the project with my other classes, especially when all my other classes have projects of their own. Without any knowledge of React and Redux, I was unable to provide much help to the front-end team, especially when they're struggling. Perhaps the most problematic of all was trying to integrate the front-end and back-end.

d) Looking back, if I were to do something differently, I would have tried to learn React and Redux. If I had experience and knowledge of React and Redux, I would have helped out in the front-end by creating some priority one pages and features instead of waiting for the front-end to finish and try to debug them.

It was a pleasure working with all of you,

Kevin Huynh

# Onubulachi-Abigail Wami:

The screenshot shows an email thread in Google Mail. The first message is from Onubulachi Wami to a team, dated 9:45 PM (9 minutes ago). It contains a self-assessment for Milestone 5, detailing contributions, difficulties, what was done differently, and GitHub statistics. The second message is a reply from Onubulachi Wami, dated 9:54 PM (1 minute ago), expressing gratitude for everyone's contributions.

**Self Assessment - Onu Wami**

**Onubulachi Wami** <owami@gmail.com>  
to williammileslew, m2hour, Benjamin, khuyenh2, dang.le730, Onubulachi-Abigail

Hi Team,

I am sending this for the completion of Milestone 5 - Self Assessment.  
Below I will describe my contributions to the success of our team this semester, as well as some difficulties that arose.

a) Contributions

- Provided mockup for the website
- Provided storyboard for navigation through the site
- Added bootstrap elements to ease mobile browsing
- Connected front-end functionality to back-end functions
- Add in visual design

b) Difficulties

- Communication was frequent but sometimes chaotic
- Representation of achieved functions without accountability
- FT Work-School Imbalance
- Establishing a consistent front-end structure with essential technologies

c) What I Would Differently?

- Proactively checking changes to the project
- Never take representation of tasks for face value
- Check and ask questions whenever needed, team members help each other
- Fix bad habits early
- Create a global Bug Table so known bugs don't reach later in development
- Create front-end mockup pages even without functionality. Back-end team was ready and established a structure early on and the time spent waiting was unnecessary.
- Spend more time testing

d) Git Hub

- 41 commits
- Additions: 2,776 ++
- Deletions: 1,195 --

**Onubulachi Wami**  
to williammileslew, m2hour, Benjamin, khuyenh2, dang.le730, Onubulachi-Abigail

I want to thank everyone for their contributions this semester. It was a pleasure working with you all.

## Dang Le:

11:11

VoLTE



### Self Assessment - Dang Le Add label ☆



Dang Le 3:51 PM

to benjaminlewis984, m2tour, oa... ↻



Hi Team,

I am sending this in regards to our Milestone 5 self-assessment. Below I have described my contributions, GitHub submission amount number, difficulties I have had, and what I would have done differently

a) I have contributed to this project by:

- Mapping out UX concepts
- Establishing front end structure
- Establishing visual design of the site with Bootstrap
- Provided mock-ups of functionality to assist other members

b) GitHub Stats: 18 commits, 32,097 additions, 17,490 deletions

c) Some difficulties in contributing to the project I had as the front-end lead:

- Proactively communicating which days I was occupied / free
- Being more specific about what technologies the front-end is using early on
- Handling workload imbalance

d) If I were to do something different next time I would have communicated with my CEO much sooner about bad habits I am noticing on my team so we can get them resolved early on.

Thank you for your hard work, best of wishes for your future.

Best,  
Dang Le

Reply

Reply all

Forward