

SW Engineering

CSC648/848 Spring 2020

Gator Trade

By:

Team 03

Samuel Bahlibi (Team Lead) samuel.bahlibi@gmail.com

Qian Hu (Front-end Lead)

Tingfeng Wan (Back-end Lead)

John Joshua Gutierrez (Github Master)

Eric Ngo (Full-Stack Developer)

Marvin Nguyen (Front-end Team)

Local Team

Milestone 2

Table of Contents

Page 1	Functional Requirements
Page 4	List of main data items and entities
Page 6	User Interface Mockups and Storyboards
Page 12	High Level Architecture, Database Organization
Page 15	High Level UML Diagrams
Page 16	Identifying key risks
Page 17	Product Management

1. Functional Requirements

Priority 1:

Unregistered User:

- 1.1 - Shall be able to browse media being posted or sold.
- 1.2 - Shall be able to browse for media.
- 1.3 - Shall be able to register.
- 1.4 - Shall be able to filter media by SFSU specific classes.
- 1.5 - Shall be able to search for media.

Registered User:

- 2.1 - Shall inherit everything an unregistered user can do.
- 2.3 - Shall be able to download free digital media.
- 2.4 - Shall be able to message another user for their contact info.
- 2.6 - Shall be able to accept a meetup location.
- 2.7 - Shall have a dashboard to view his/her postings and messages.
- 2.8 - Shall be able to upload digital media to the website.
- 2.12 - Shall be able to login.
- 2.13 - Shall have a dashboard to view the list of media they posted and messages.

Administrator:

- 3.1 - Shall inherit everything a Registered User can do.
- 3.2 - Shall be required to moderate(approve) all media posted on the site before they go live.
- 3.3 - Shall be able to delete any post on the site.
- 3.4 - Shall be able to ban users from the site.

Priority 2:

Unregistered User:

N/A

Registered User:

- 2.5 - Shall be able to ping for a meetup location.
- 2.9 - Shall be able to delete their digital media from the website.

Administrator:

3.5 - Shall have a login protected admin-panel to facilitate

Priority 3:

Unregistered User:

N/A

Registered User:

2.10 - Shall have a profile page showing their current listings.

2.11 - Shall be able to view purchased items.

Admin:

N/A

2. List of Main Data Items and Entries

Unregistered User:

- User who does not have an account on the website.
- They can register for an account.
- They are able to browse the website, search, and view any of the media being posted or sold.

Registered User:

- They have the same privileges of an unregistered user.
- User who has created an account on the website.
- They are able to purchase media and download free media
- They can create posts for free media or purchasable media.
- They can contact the seller.

Admin:

- User with full admin privileges.
- Admins must approve all posted content before it goes live on the website.
- Privileges include the ability to ban users and remove posts if it violates any rules.

Post:

- Post is an item media with the following information:
 - Owner ID such as username/email
 - Price if the item is not free
 - Download link if item is free
 - Category
 - Media Type (Video, photo, book, etc.)
 - Description (background information about the media)
- Pictures shall be low res.
- Download will be available if the media is free.

User Registration Record:

- Contains the following information of a registered user:
 - Email address
 - Password (Encrypted)
 - First Name
 - Middle Name (Optional)
 - Last Name
 - Student or Faculty

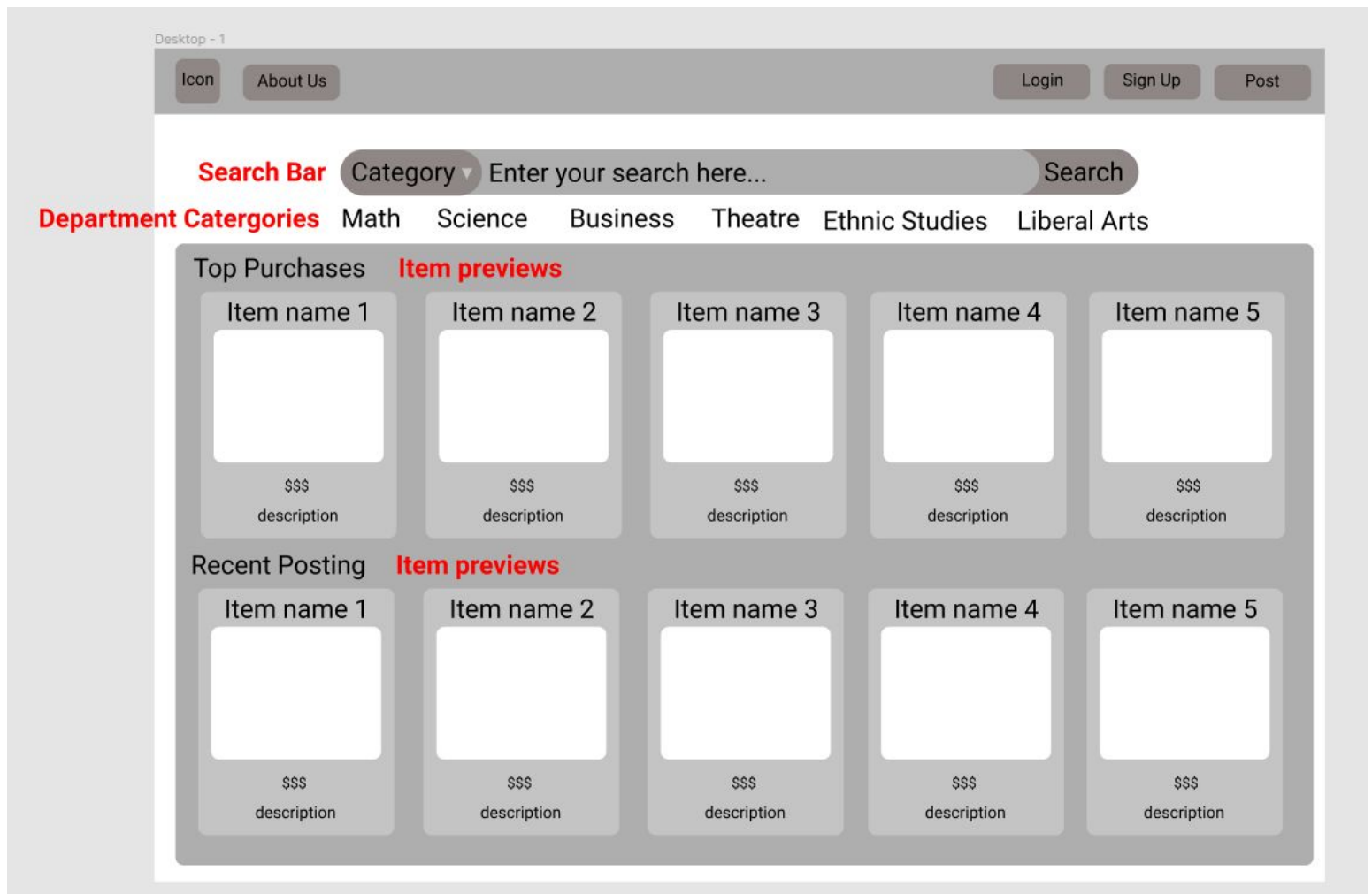
Message:

- Can be used by registered users.
- Contains text that a registered user can create to send to a seller.
- Contains the time the message is sent and the sender's information such as their name.
- Sellers can send messages back to a registered user that contacted them.

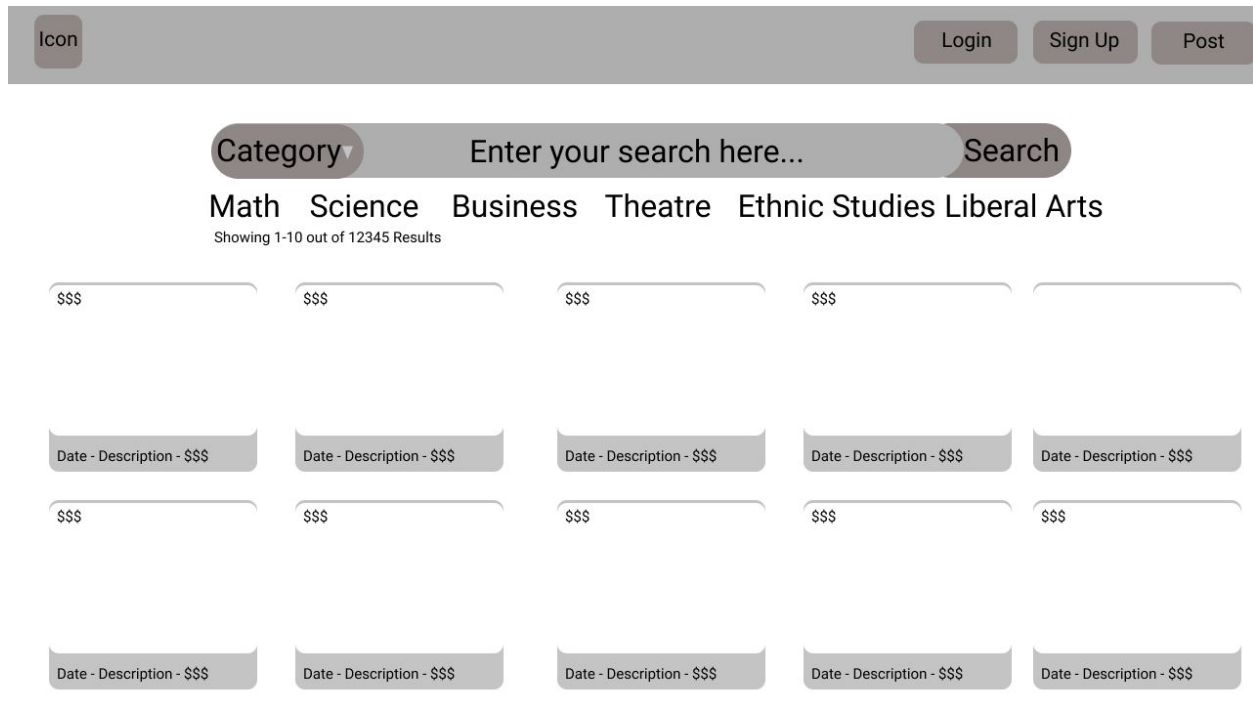
3. User Interface Mockups and Storyboards

Pages we need to make on FIGMA:

- **Home Page**
 - A search bar
 - Items, like a catalogue
 - Somewhere in the corner is register or login
 - Can browse



- **Search Results Page**
 - Main purpose is to browse



- Item Detail Page



- Register Page

Gator Trade
Register

1 First Name:*

2 Last Name:*

3 Email:*

4 School Role:

5 Password:*

6 Confirm Password:*

7 ☐ I agree to the [terms and conditions](#)*

8

Already registered?
[Sign in](#)

Input 1-6 information

*** marks mandatory fields**

Check off 7

Click when finished →

- Login Page

Gator Trade

Login

Login
Information
Inputs

Username

Password

Press to
submit
information

Submit

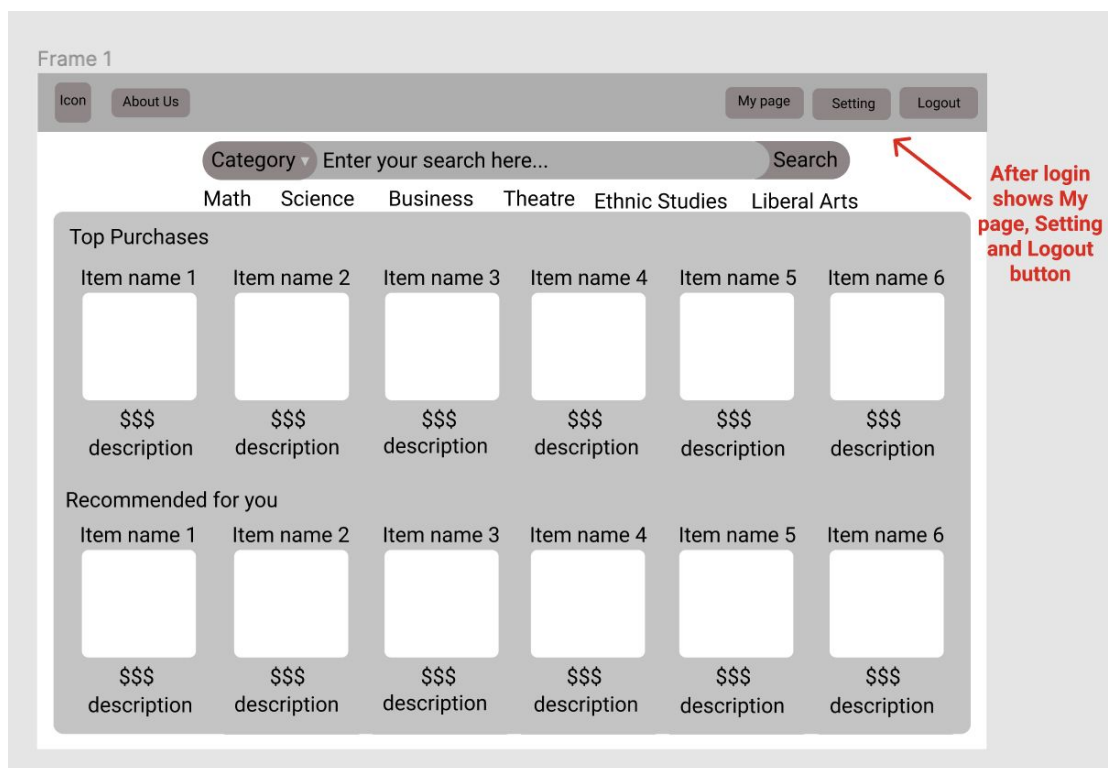
Link to sign
up page

Need an account?

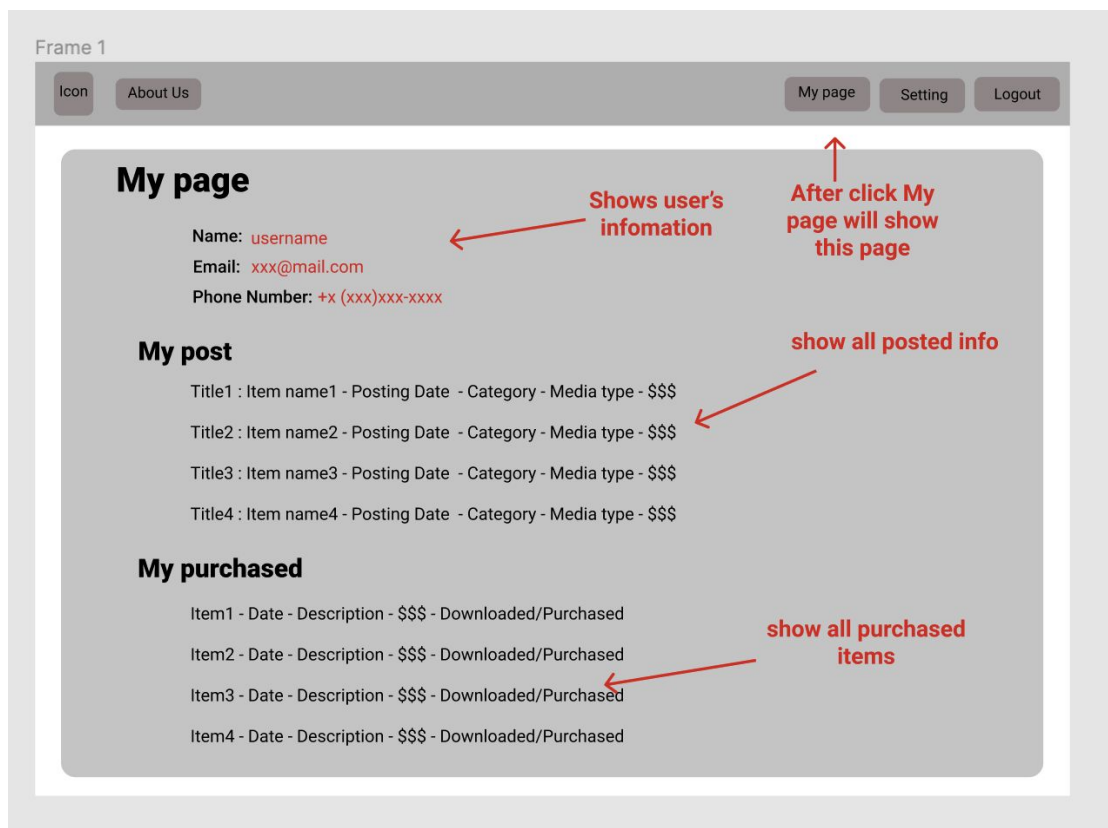
Sign up!

Forgot password?

- Account Page:



Show's account info that shows posting and downloads



- **Priority 3: Settings Page**

Frame 1

Icon About Us My page Setting Logout

Setting

Username:
XXXXXX Edit

Email:
xxx@mail.com Edit

Phone Number:
+x (xxx)xxx-xxxx Edit

Password:
***** Edit

Notefication:
Email feedback ☒

After click Setting will show this page to set up personal info

- Create a Posting Page

Create a Post

1 Title*

2 Category* ← Drop down menus

3 Media Type* ←

4 Price* ← Typed or incremented

5 Description...

6 Upload file... Upload file if it is not a physical object (Picture, Video, etc.)

7 ☐ Physical object? Check off if the object is physical(Book)

Image preview

8 Insert Picture

Note: The image inserted is used for the post. It is not for uploading a picture.

9 Create Post

Click when finished

Input all information (1-4) required

5-8 are optional

Your Post has been created. It has been sent for approval

You will be redirected...

[Click Here](#) if you have not been redirected

**Landing page after creating the post.
Redirects back to home page**

4. High Level Architecture, Database Organization

DB Organization:

User table: Contains information of registered users as well as administrators.

Column	Type	Default
ID	ObjectId	
First Name	String	
Last Name	String	
Email Address	String	
School Role	String (either “student” or “faculty”)	Student
Admin	Boolean(T is Admin, F, not admin)	False
Password	String	
Phone Number	String	“”

Post Table: Contains information of posts

Column	Type	
ID	ObjectId	
Author ID	ObjectId (foreign key by User)	
Title	String	
Media Preview	BLOB	None
Media Path	String (filepath)	
Media Type	String	
Physical	Boolean	
Cost	Float	Dollar amount. 0.00

Approver ID	ObjectId (foreign key by User)	
Description	Text	""
Status (approved or not yet approved)	Boolean	False

Purchases Table: contain logs of individual's access to post's media

Column	Type	
user_email	String(foreign key by Users)	
post ID	ObjectId (foreign key by Posts)	

Post Category Table: contains individual post's categorizations

Column	Type	
post ID	ObjectId (foreign key by Posts)	
category	String	

Media Storage:

- Keep maintenance simple by using DB BLOBs.

Search/filter architecture and implementation:

- Use SQL to perform search queries.
- Do simple search for browse by classes and free text (title) search (typeahead).
- Search items can be ordered alphabetically or by date.

APIs: APIs will follow REST patterns

Users API:

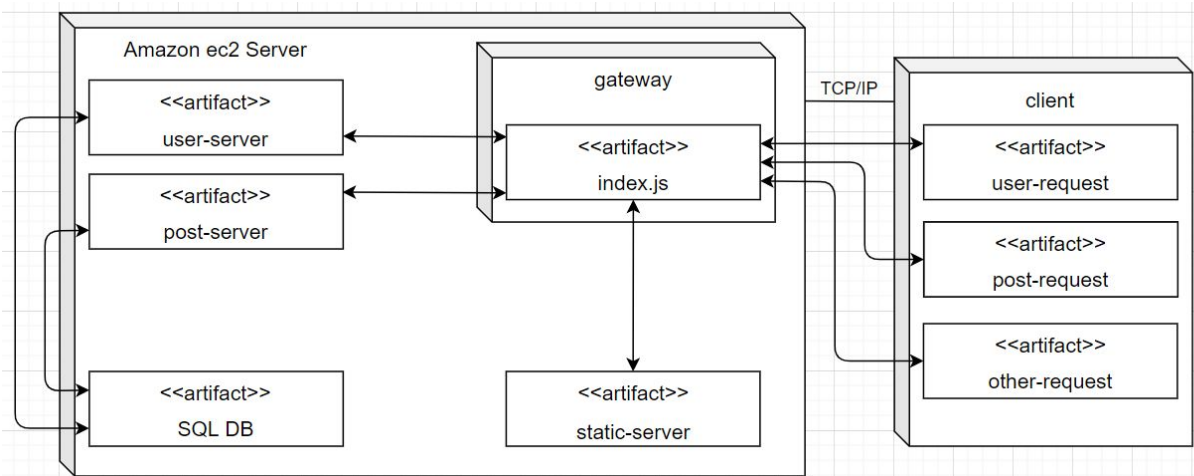
- GET /user : Get user information
- GET /user/posts : Get user's authored posts
- GET /user/posts/purchased : Returns an array of user's purchased posts.
-
- POST /user : Create new user
 - Body: {Firstname, LastName, email, password, school role}

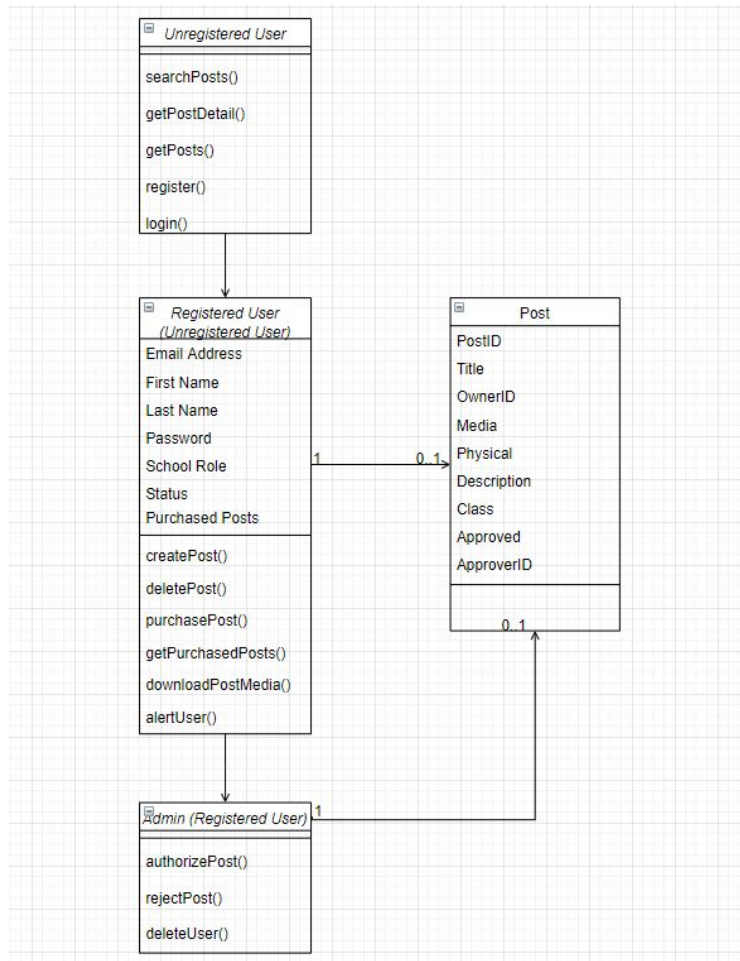
- PUT /user : Update user information
- DELETE /user: Delete user

Post API:

- GET /post/all : Return all approved posts
- GET /post?id={id} : Return a post's information.
- GET /post/approval : returns a list of posts to be approved (used by admin).
- GET /post/requests : returns a list of buyers/post pairs for current user/seller to approve (purchase confirmation).
- POST /post : Create a new post
 - BODY: Multipart data format containing {title,media preview,media BLOB(if applicable),description,physical, class}
- POST /post/purchase : Update user's purchased items. If post is not free, requests confirmation from user.
 - BODY: {post_id, purchase:true/false, approve:true/false}.
- POST /post/requests : Allows seller to approve a buyer's purchase
 - BODY: {buyer_id, post_id}
- PUT /post/approval : Admin to approves or rejects a post.
 - BODY: {post_id, approve: true/false}.
- DELETE /post?id={id} : Deletes a post

5. High Level UML Diagrams





6. Identifying key risks

Skills risks: We have no skill risks. As a preventative measure we have discussed the idea of everyone doing a quick boot camp to understand the languages we are using.

Schedule risks: We currently have no schedule risks, everyone is meeting their goals discussed in our meetings. We hold three meetings per week practicing Scrum. In the future I plan to move into an Agile team development.

Technical risks: As far as any technical issues, we seem to be well off. Eric is a great asset to our team. Having industry experience, as well as, knowing good programming practices he is a source we will look towards for any technical risks.

Teamwork risks: In the beginning we had issues with team members not showing up to meetings. As of now we have ironed out those wrinkles and are proceeding without any issues. We had our first team bonding experience after our meeting on March 20th.

Legal/content risks: As far as any legal risks there are none. We do not plan to take any copyrighted materials and use it in our project.

7. Product Management

For product management, I used Trello to show the current flow of work that needed to be done. It helps to have a team that's very active and involved with the whole project. We had a rocky start with Milestone 1, but things quickly shifted after we began Milestone 2. We're currently dealing with the COVID-19 pandemic, but this has actually turned into a positive for us. Our previous Milestone we were only meeting once a week. Our weekly scrum meetings have tripled and I feel that has increased our productivity. These three scrum meetings per week turn into miniature checkpoints before we hit the Milestone.

