

Final Project for SW Engineering Class CSC 648-848 Spring 2023

Team 5

Piqued

Jose Avila, Team Lead, javila6@mail.sfsu.edu

Andy Shi, Frontend Lead, ashi2@mail.sfsu.edu

Leo Saeteurn, Backend Lead, lisaeteurn@mail.sfsu.edu

Nishit Pachchigar, Github Master, npachchigar@mail.sfsu.edu

Joshua Hayes, Database Master, jhayes10@mail.sfsu.edu

Gautami Gudla, Document Editor, gkollolusrinivasa@mail.sfsu.edu

<http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000>

Milestone 5

May 25th, 2023

History Table

Version	Submission Date
M5V1	May 25, 2023
M4V2	May 25, 2023
M4V1	May 18, 2023
M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 19, 2023
M2V1	March 30, 2023
M1V2	April 16, 2023
M1V1	March 5, 2023

1. Product Summary

1.1. Name Of The product:

Piqued

1.2. Final P1 Functions:

1.1.1. All Users

- 1.1.1.1. All users shall be able to view the website's information
- 1.1.1.2. All users shall be able to contact the site moderators with questions
- 1.1.1.3. All users shall be able to view the privacy policy
- 1.1.1.4. All users shall be able to view the terms and conditions

1.1.2. Guest Users

- 1.1.2.1. Guest users shall be able to create an account
- 1.1.2.2. Guest users shall be able to use a unique username to create an account
- 1.1.2.3. Guest users shall be able to only view public posts
- 1.1.2.4. Guest users shall be able to view the contents of public posts
- 1.1.2.5. Guest users shall be able to view a profile's public information
- 1.1.2.6. Guest users shall be able to search for public posts
- 1.1.2.7. Guest users shall be able to discover other public users
- 1.1.2.8. Guest users shall be able to view comments on public posts
- 1.1.2.9. Guest users shall be able to view reactions on public posts

1.1.3. Registered Users

- 1.1.3.1. Registered users shall be able to log in
- 1.1.3.2. Registered users shall be able to log out
- 1.1.3.3. Registered users shall be able to sign in with either their email or username.
- 1.1.3.4. Registered users shall be able to add a profile picture
- 1.1.3.5. Registered users shall be able to edit their profiles
 - 1.1.3.5.1. Registered users shall be able to change their profile picture
 - 1.1.3.5.2. Registered users shall be able to change their displayed username
 - 1.1.3.5.3. Registered users shall be able to change their email address

- 1.1.3.6. Registered users shall be able to search for other users by username
- 1.1.3.7. Registered users shall be able to create blog posts.
- 1.1.3.8. Registered users shall be able to add photos on blog posts.
- 1.1.3.9. Registered users shall be able to search for posts.
- 1.1.3.10. Registered users shall be able to delete their posts.
- 1.1.3.11. Registered users shall be able to add comments on post.
- 1.1.3.12. Registered users shall be able to delete their own comments.
- 1.1.3.13. Registered users shall be able to like/dislike to other people's comments

1.1.4. Administrator Users

- 1.1.4.1. Administrator users shall be able to create authorized users on account

1.1.5. Site moderators - creators(us)

- 1.1.5.1. Site moderators shall be able to delete posts/articles
- 1.1.5.2. Site moderators shall be able to ban accounts
- 1.1.5.3. Site moderators shall be able to temporarily block accounts for review
- 1.1.5.4. Site moderators shall be able to edit the layout of the platform
- 1.1.5.5. Site moderators shall be able to add additional features
- 1.1.5.6. Site moderators shall be able to answer users' questions and concerns
- 1.1.5.7. Site moderators shall be able to create an account
- 1.1.5.8. Site moderators shall be able to post on the platform's main page
- 1.1.5.9. Site moderators shall be able to make site announcements
- 1.1.5.10. Site moderators shall be able to update website
- 1.1.5.11. Site moderators shall be able to temporarily suspend the website for maintenance
- 1.1.5.12. Site moderators shall be able to reinstate accounts
- 1.1.5.13. Site moderators shall be able to review accounts

1.1.6. User's Profile Page

- 1.1.6.1. User's profile page shall have the user's picture
- 1.1.6.2. User's profile page shall have the user's username
- 1.1.6.3. User's profile page shall have published public posts
- 1.1.6.4. User's profile page shall have a biography
- 1.1.6.5. User's profile page shall have contact information that are made public by user

1.1.7. All pages

- 1.1.7.1. All pages shall have a navigation bar/header
- 1.1.7.2. All pages shall have a menu with links to informational pages

1.1.7.3. All pages shall have our company's logo

1.1.8. User's Logged-In Interface

1.1.8.1. User's logged-in interface shall have a link to the user's account management page in the navigation bar

1.1.9. Blog Posts

1.1.9.1. Blog posts shall contain an original author

1.1.9.2. Blog posts shall contain a date and time of creation

1.1.9.3. Blog posts shall contain a title

1.1.9.4. Blog posts shall contain a photo(s)

1.1.9.5. Blog posts shall contain a category(ies)

1.1.9.6. Blog posts shall contain a tag(s)/hashtag(s)

1.1.9.7. Blog posts shall contain a comment sections with comments from other users

1.1.10. Exploration Newsfeed

1.1.10.1. Exploration newsfeed shall contain random public posts

1.1.10.2. Exploration newsfeed shall have a search bar

1.1.10.3. Exploration newsfeed shall have blog posts that can be clicked on to view full posts

1.1.11. Search Bar

1.1.11.1. Search bar shall have a text box to search up public blog posts by categories, tags, and users

1.1.12. Navigation Bar

1.1.12.1. Navigation bar shall have a link to the user's profile when logged in

1.1.12.2. Navigation bar shall have a link to the home page

1.1.12.3. Navigation bar shall have a link to the user's account management page when logged in

1.1.12.4. Navigation bar shall have a post button

1.1.12.5. Navigation bar shall have a log out button when logged in

1.1.13. Menu (Footer)

1.1.13.1. Menu shall have a link to the company's information page

1.1.13.2. Menu shall have a link to contact the company by emailing

1.1.13.3. Menu shall have the company's contact information

1.1.13.4. Menu shall have a link to "about us" page

1.1.13.5. Menu shall have a copyright logo

1.1.13.6. Menu shall have a link to the terms and conditions

1.1.13.7. Menu shall have a link to the privacy policy

1.1.14. Chat Box

1.1.14.1. Chat box shall contain user's username for both sender and recipient(s)

1.3. What is Unique In Our Product:

Piqued is a unique blogging website that offers a safe and social platform for individuals to express themselves while sharing common interests. Unlike other blogging sites, Piqued allows users to interact with their friends and other like-minded individuals, creating a sense of community around shared interests. The platform is perfect for both extroverted and introverted individuals, as it offers a safe space to share without fear of scrutiny. Piqued also allows users to find new interests and experiences, making it a great platform for exploration and self-discovery. The ability to connect and build a community of like-minded individuals sets Piqued apart from other blogging sites and makes it an excellent choice for those looking to express themselves and connect with others.

What makes Piqued truly special is the ability to connect with individuals who share the same interests. Through engaging and detailed blog posts, you can delve into your favorite topics such as travel experiences, culinary adventures, life hacks, and more. Discover new ideas, gain inspiration, and add exciting activities to your bucket list, all while building a vibrant community of like-minded individuals.

1.4. URL Of the Product:

<http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000>

2. Milestone documents – M1-M4

SW CSC648/848 Spring 2023

Piqued

Team 5

Jose Avila, Team Lead, javila6@mail.sfsu.edu

Andy Shi, Frontend Lead, ashi2@mail.sfsu.edu

Leo Saeteurn, Backend Lead, lisaeteurn@mail.sfsu.edu

Nishit Pachchigar, Github Master, npachchigar@mail.sfsu.edu

Joshua Hayes, Database Master, jhayes10@mail.sfsu.edu

Gautami Kollolu Srinivasa, Document Editor, gkollolusrinivasa@mail.sfsu.edu

Milestone 1

February 23, 2023

History Table

Version	Submission Date
M1V2	April 14, 2023
M1V1	March 5, 2023

Table of Contents

Cover Page.....	
1	
Table of Contents.....	
2	
1. Executive Summary.....	
3	
2. Main Use Cases.....	
4	
2.1 Searching for Travel Destinations	
2.2 Restaurant Experience Blog	
2.3 Foodie Browsing for Restaurants	
2.4 Administrators	
2.5 Aspiring Content Creator	
2.6 Connecting Users	
2.7 Social Suggestions	
2.8 Advertisement	
2.9 Business Account	
2.10 Chatter	
3. List of Main Data Items and Entities.....	
14	
3.1 Data Description	
4. List of Functional Requirements.....	
15	
4.1 Guest Users	
4.2 Registered Users	
4.3 Administrative Users	
4.4 Site Moderators	
5. List of Non-Functional Requirements.....	
17	
6. Competitive Analysis.....	
18	
7. High-level System Architecture and Technologies Used.....	
20	
8. Checklist.....	
21	
9. List of Team Contributions.....	

1. Executive Summary

Have you ever wanted a platform where you can express yourself while sharing common interests? Well look no further! Introducing, Piqued! Piqued is the ultimate blogging website that allows you to express yourself and your interests freely online while connecting with your friends, families, your friendly neighbors, and even make new friends. It's no ordinary blogging site because you can socialize and interact with your group of friends who you are connected with. It's already hard enough to be able to talk to someone about your interests that you want to share but the people around you are not listening! This site is not only for extroverted individuals who do not shy away from reflecting on their activities, this is a great site for introverted individuals as well because you can share your daily Pique without having to speak it out loud - some of us know the nerve wrecking feeling. While blogging your interests online, you can find numerous individuals who share the same interests as you! It's important to be able to share without being scrutinized, so this platform will create a safe space for all individuals to share what they're into. Users can also find things they may be interested in and want to look more into it, so Piqued is a great platform where you can find details of someone else's experience so you can decide if you want to take that trip, or try cooking new food, or finding the easiest way to manage your calendar - who knows, maybe you'll find someone else's interest that may be intriguing to you but you didn't know it! You connect and build a community where you can discuss commonalities.

Piqued allows for you to share your favorite interests in full detail. For example, you traveled to Spain where you visited many places and tried new foods, and when your friends ask you about it, you can say "hey, I'm glad I "Piqued" your interest, why don't you check out my recent blog post where I shared my itinerary, photos, and my experience!". You can share a plethora of interests such as travels you've gone on, food you've tried or cooked yourself, and even life hacks - who wouldn't love a quick tip! You can explore other people's interests and may stumble upon a blog that you might want to add onto your bucket list then blog about it from your perspective. Most of all, Piqued allows you to connect to other like minded individuals and create a space where you can share things that you think are important. Make it your world whether you're sharing with the world or blogging to keep this as an online journal where you can come back to and just reminisce on your experience or make it a place where you can explore other people's interests so that you can add it to your "must try" list. Whatever it is, you can use the platform however you want to.

There are several blogging sites, but is there one like Piqued where you can blog to share with your connections and interact with them? What's unique about this platform is that it allows users to socialize, talk to each other about interests and share each other's experiences, and your connections can react to your daily Pique! What's also special about this site is it creates a community where users can find individuals who share the same interests and make new

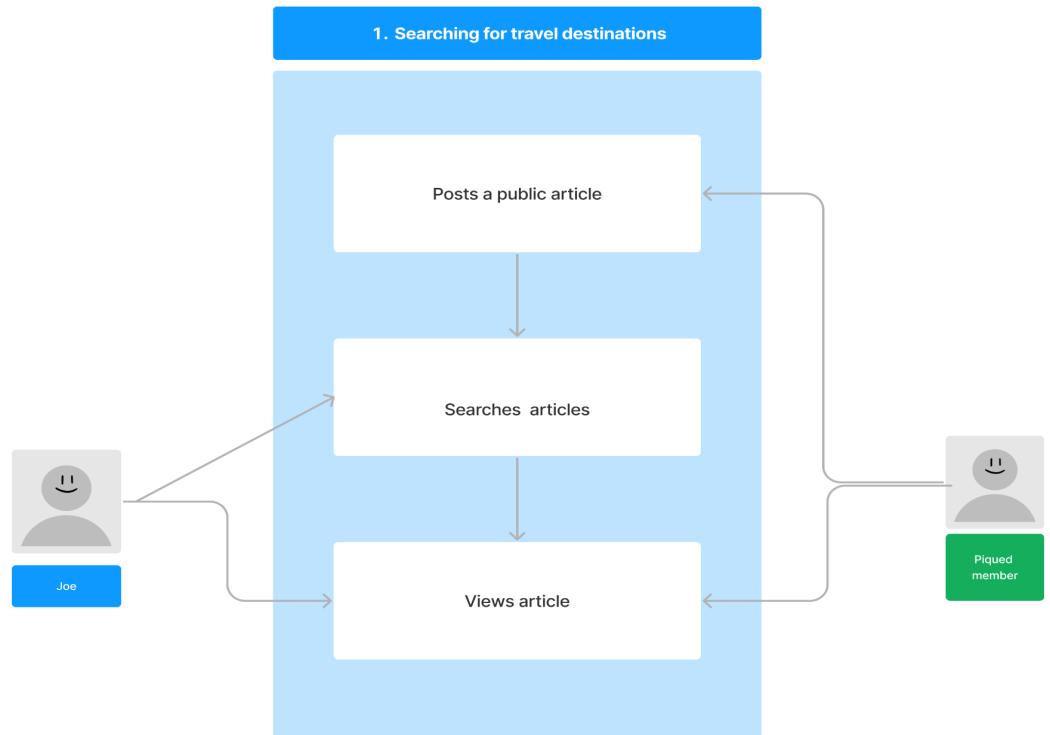
connections across the country. You'll never know if you meet your next soulmate, right? So, have I Piqued your interest?

2. Main Use Cases

2.1. Searching for Travel Destinations

Actors: General User: can search and view public blog posts

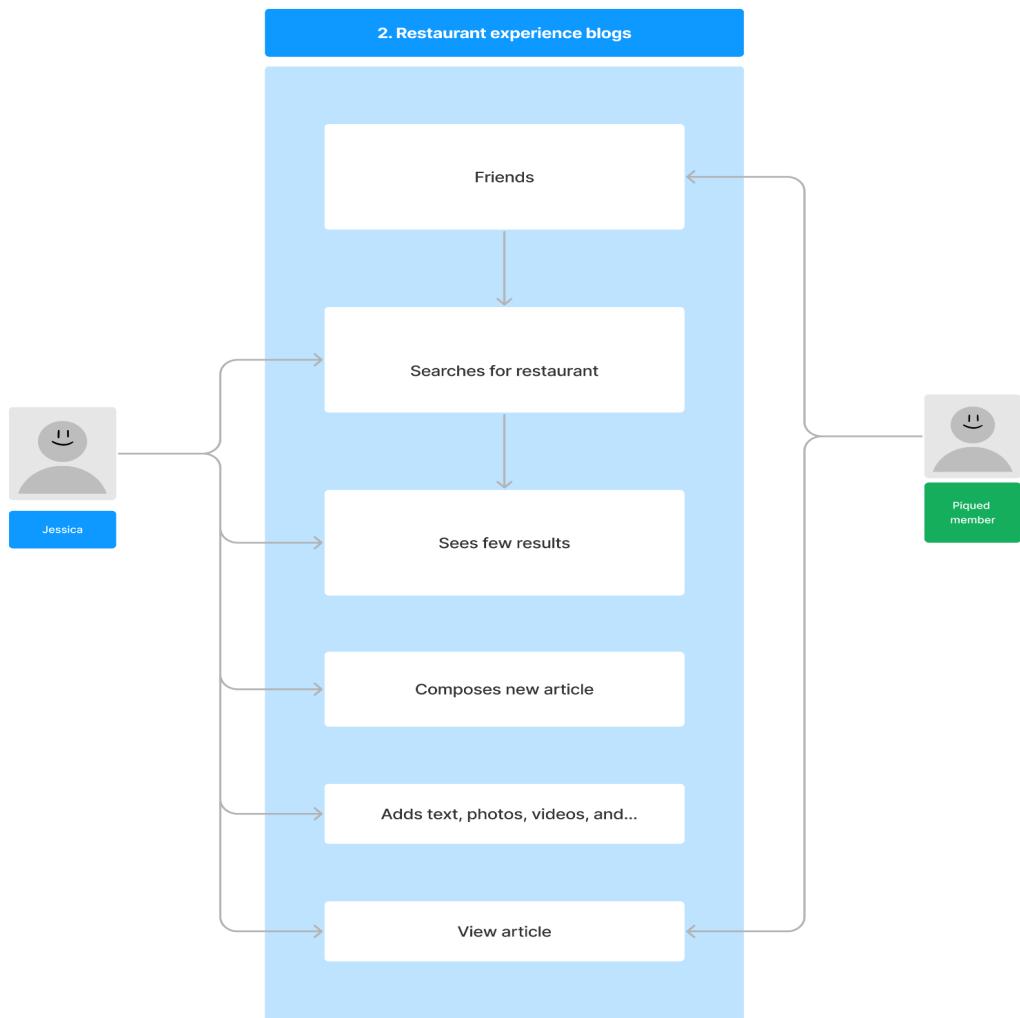
Description: Joe is a young adult planning a trip to Los Angeles with his friends, but doesn't know what are some good places to visit as this will be his first time traveling there. While he has never used the platform before, he uses the internet and social media on a daily basis, so he is familiar with technology but does not like registering on accounts due to growing data privacy concerns. He quickly learns how to use the search functionality and performs a search for "places to visit in Los Angeles". This brings up a list of blogs from other users that have made posts about his search query and their recommendations on places to check out. He quickly scans through the listing, finds one that interests him, and is able to learn more about places to check out while in the city.



2.2. Restaurant Experience Blogs

Actors: Registered User: has an account and can post private/public blogs.

Description: Jessica works as a food journalist for a local news company. She has no trouble writing and publishing review articles online. In her free time, she also likes to use Piqued to write blogs and share with her thousands of followers about the restaurants she has visited. Recently, Jessica came across a restaurant that recently opened and wasn't getting much traction just yet on Piqued, despite the food being in her opinion top tier. In an effort to help spread awareness, she decides to log on to her Piqued account to create a new public post and begin drafting a new page. She also adds photos and videos of some of the menu options to her post. Once she is done, she can click a button to publish the post publicly,

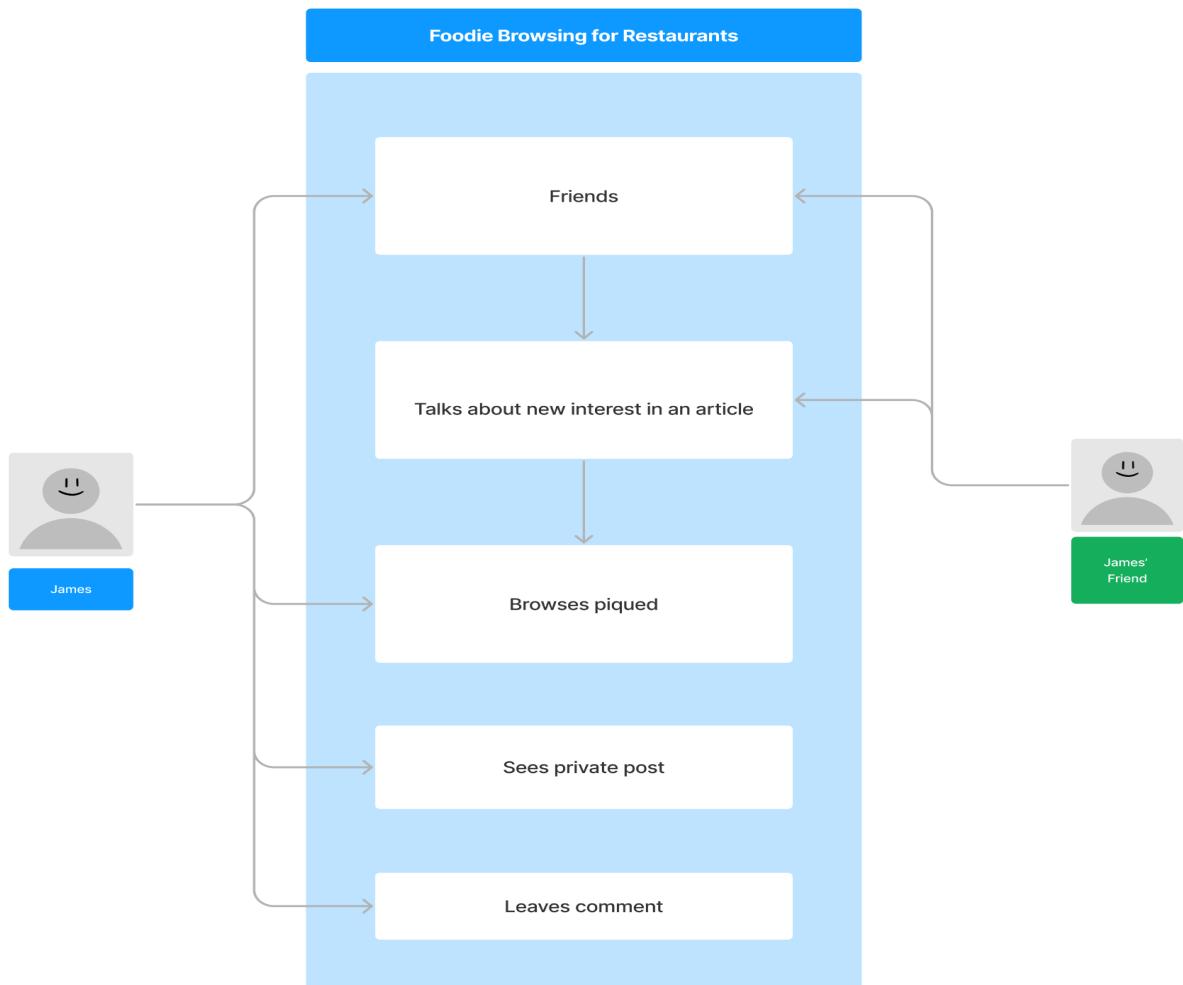


and other users, both signed-in and signed-out, can view her blog and decide whether or not to check out the new restaurant.

2.3. Foodie Browsing for Restaurants

Actors: Registered User: can browse and view private blog posts.

Description: James has recently moved to San Francisco and loves to try new restaurants whenever he has the chance to. To help him discover local hidden gems in the city, James uses Piqued to get dining recommendations from other Piqued foodies. As a registered user, James can: create blog posts, follow other Piqued users, and access private blogs written by friends, in addition to general user capabilities. While browsing the site in his spare time, he sees a recent article about a new restaurant in town that was posted by another SF foodie that he is friends with and gets frequent private food recommendations from. After trying

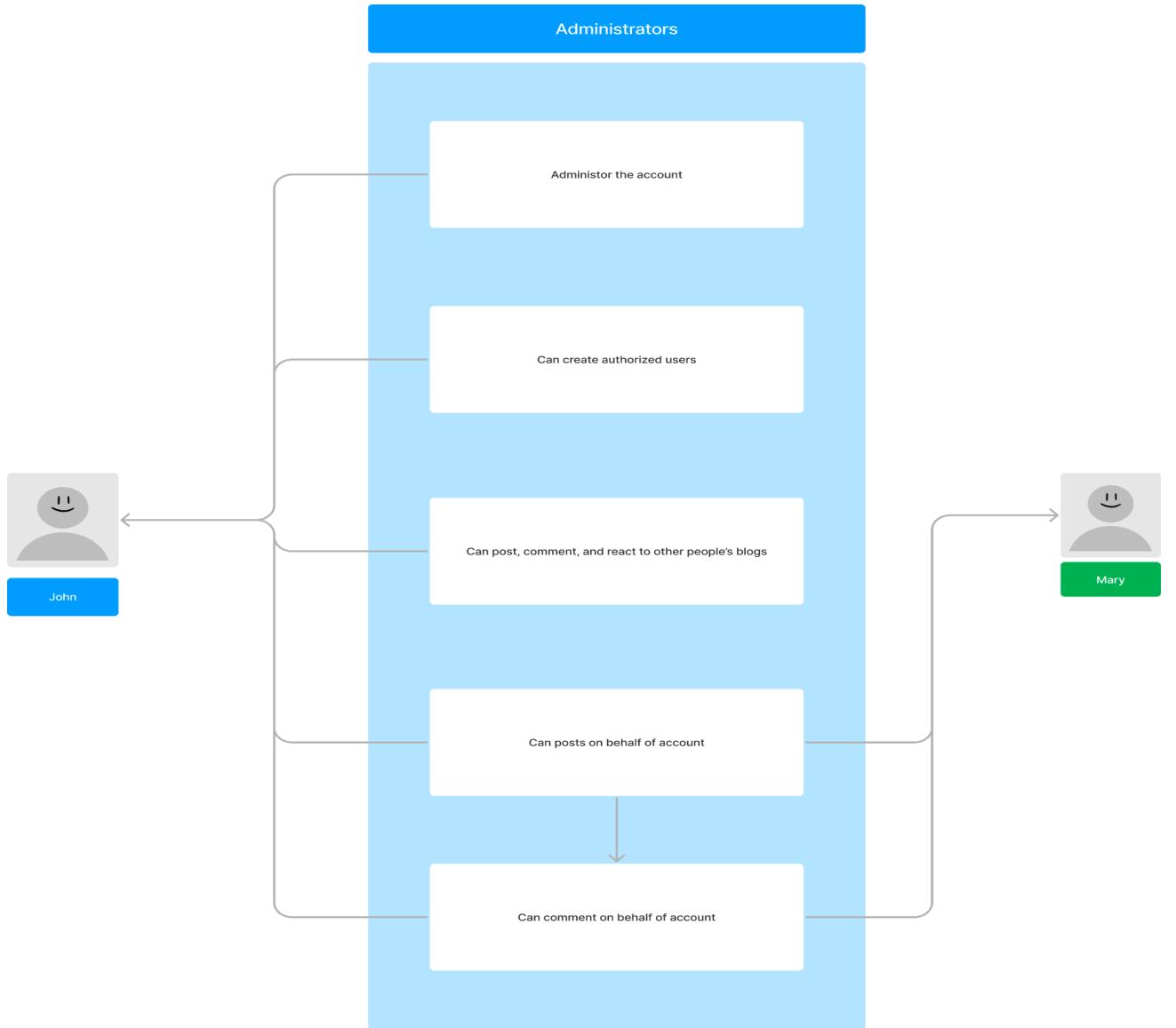


out the new restaurant, James goes back to the blog and thanks his friend for the great recommendation.

2.4. Administrators

Actors: Administrator users: manage and maintain accounts. Registered user: has the ability to post and comment on blogs.

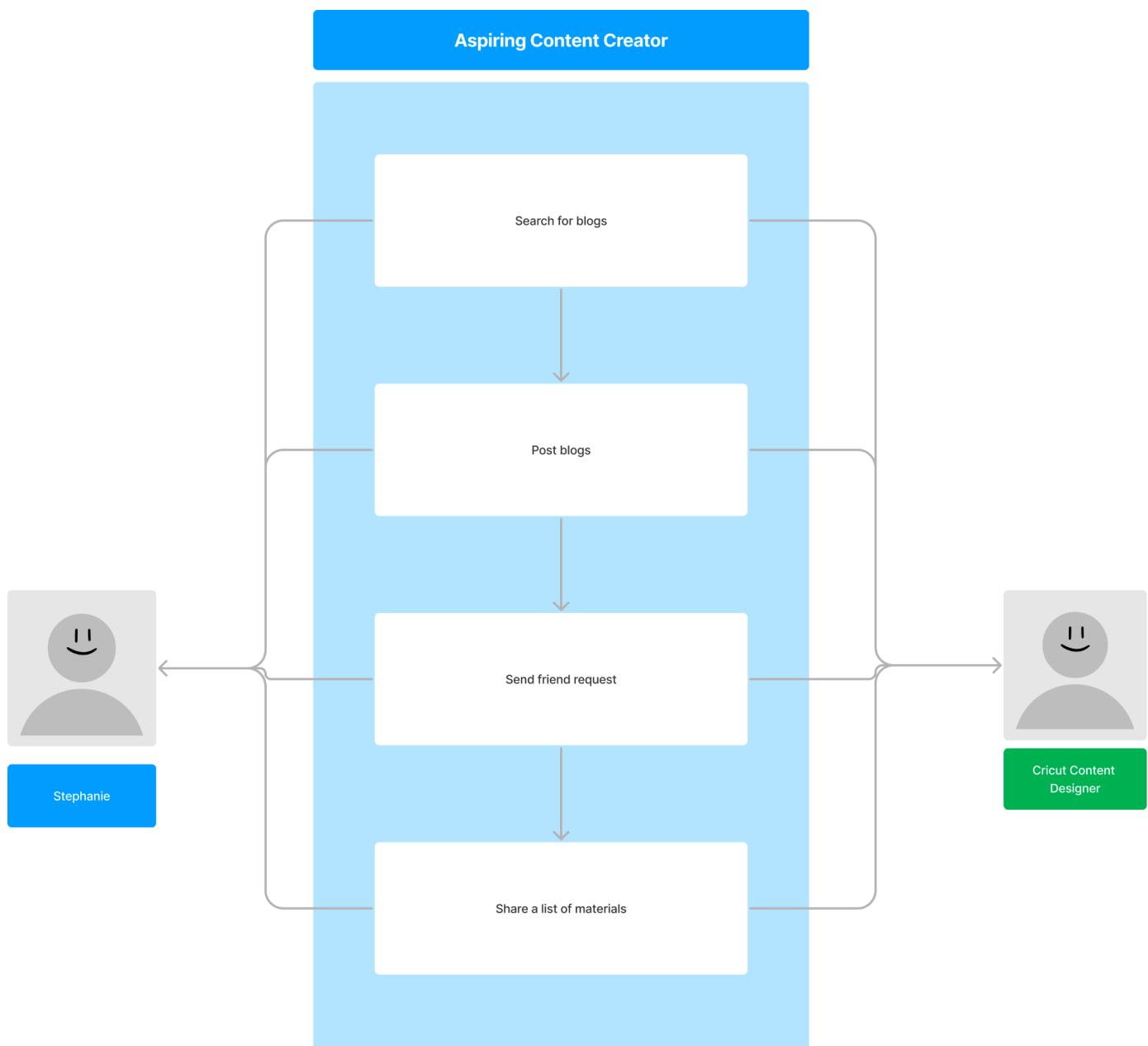
Description: John is the administrator of his business account on Piqued. He has the ability to manage the account, create authorized users, give access to authorized users to post on his behalf, and he can monitor what's going on with his account. He can also post, comment, react, and chat on the account as well. John can authorize one of his employees, Mary, who is also a registered user of Piqued to post and comment on his behalf using his account when he is busy working on other things, but only to the account's blog. With an extra user on his account, John is able to have contents posted on his account every week.



2.5. Aspiring Content Creator

Actors: General User: can search and view public blog posts for inspirations.

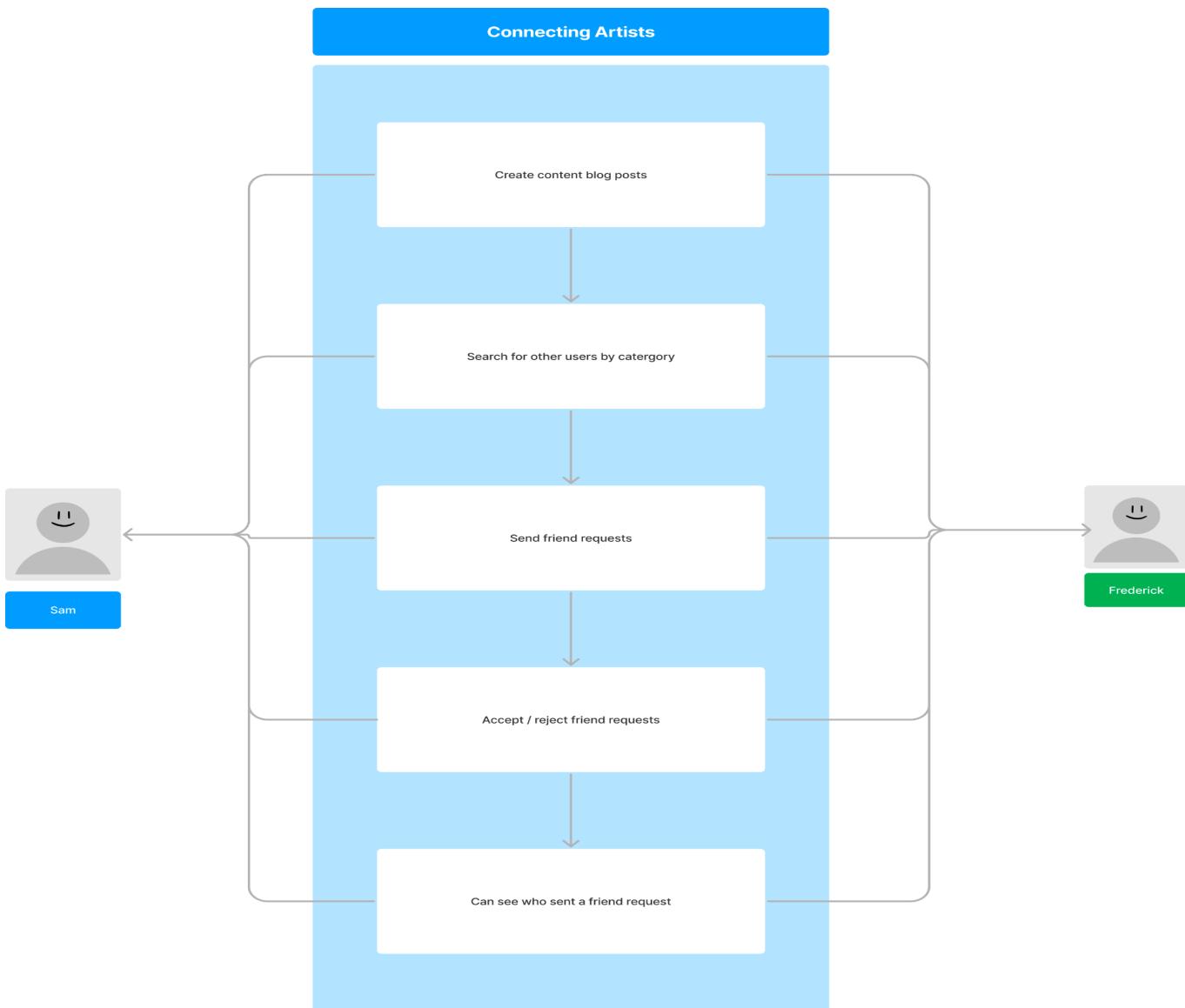
Description: Stephanie loves to create things using her Cricut machine but seems to only think of a few ways to make designs using her vinyls. To get some inspiration, she hops on Piqued website, logs in, and searches for Cricut designs. Lo and behold, she finds numerous blog posts from various Cricut content designers who post about their tips and tricks for creating new designs as well as provide a list of materials needed.



2.6. Connecting Artists

Actors: Registered User: has an account and can connect to other users by sending friend requests.

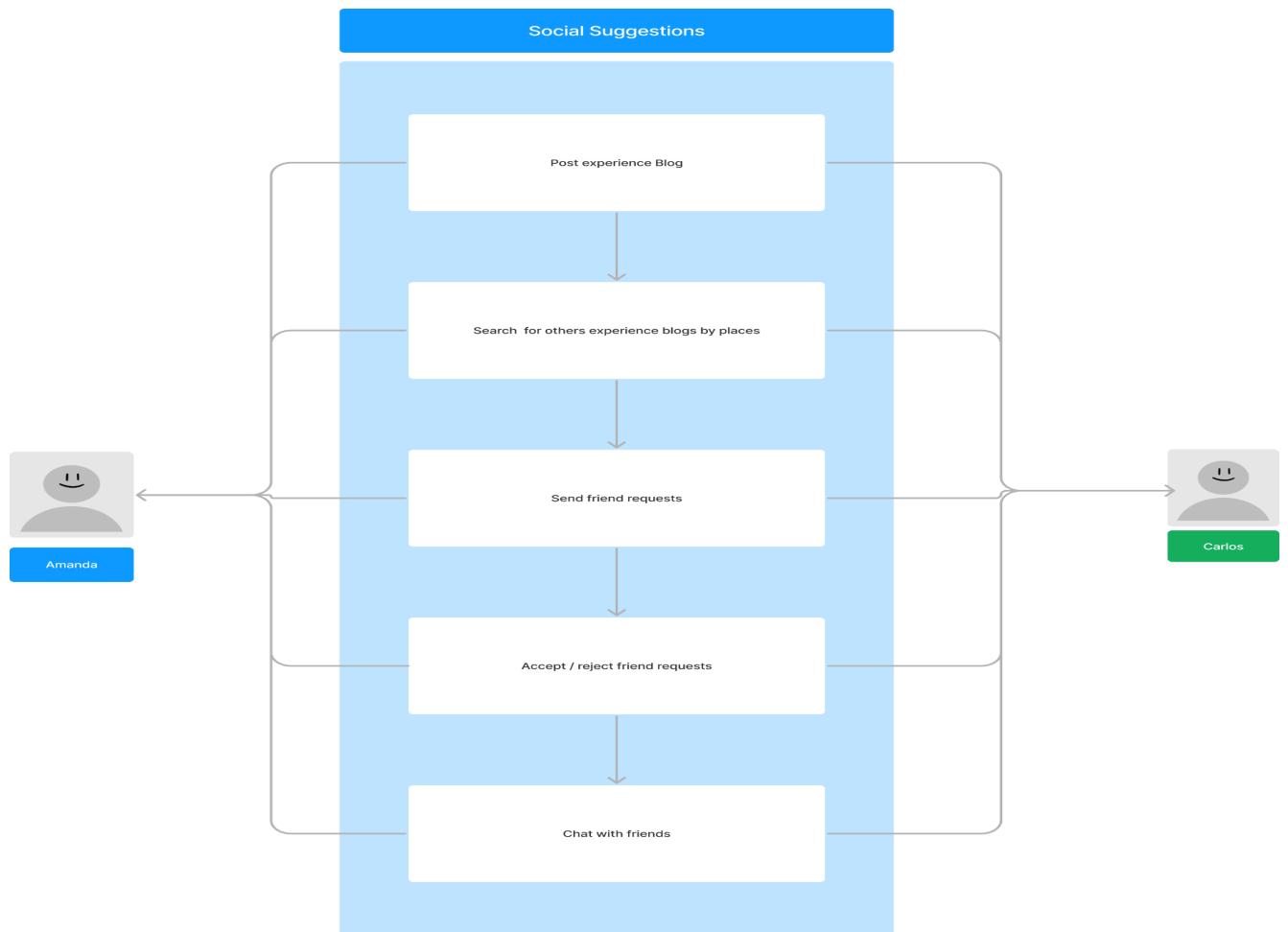
Description: Sam is a registered user and is passionate about making macaroni art. While browsing through Piqued for other art inspiration, she comes across an account of another user, Frederick, who also shares the same passion. Eager to find more friends with an interest in macaroni art, Sam sends a friend request to the other user. Frederick checks and sees that he has a friend request and is able to accept or reject Sam's request, which he accepts.



2.7. Social Suggestions

Actors: Registered User: has an account and can see friends' posts.

Description: Amanda and Carlos are friends from the same hometown. Amanda has been traveling the world and has visited New York, Florida, Italy, and Croatia, whereas Carlos has not traveled much and has never been to Italy before. Carlos is planning to travel to Italy and wants to seek Amanda's advice on where to go, what to eat, and how to spend his time in Italy. Unfortunately, he cannot get in touch with Amanda, and he is lost and not having a good experience in Italy. If Carlos had registered with the Piqued web browser, he would be able to see all of Amanda's experience and suggestions in Italy without having to come in contact with her. Through Piqued, Carlos could access Amanda's travel experiences, recommendations, and insights on Italy. He could find information about the best places to eat, the best things to do, and the most interesting places to visit. This would have made his experience in Italy much better, as he could have followed

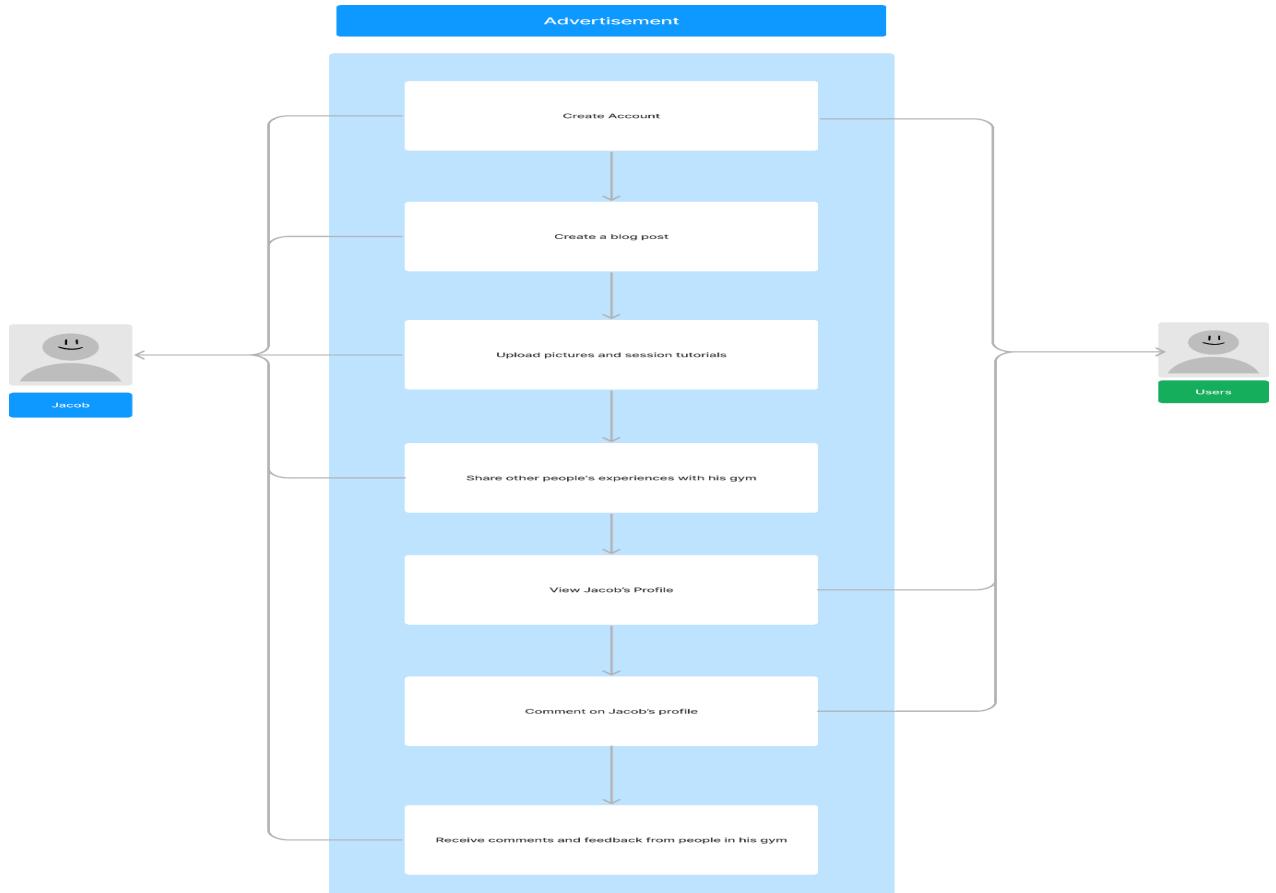


Amanda's suggestions and discovered the best of Italy without having to go through the trouble of trying to reach out to her.

2.8. Advertisement

Actors: Registered User: has an account and can post private/public blogs to promote his business.

Description: Jacob is a gym instructor based in New York City, and he wants to showcase his skills and abilities to people all around the world. To achieve this, he learns about a new app called Piqued, which is a platform that allows users to blog about their experiences, share their ideas and knowledge, search for people's suggestions, and perform many more activities. After creating a new account on Piqued, Jacob uploads pictures and session tutorials on the website, along with a summary of how he helps people get into their best physical shape. Additionally, he shares other people's experiences with his gym, which helps him reach out to more people and attract them to join his gym. The app also allows people from Jacob's gym to comment on his blog and share their perspectives about how he trains people. This feedback can help Jacob improve his services and provide better training to his clients. Using Piqued to showcase his skills and abilities as a gym instructor allows Jacob to reach a wider audience, attract more people to his gym, and receive valuable feedback from his clients. It is an effective way to

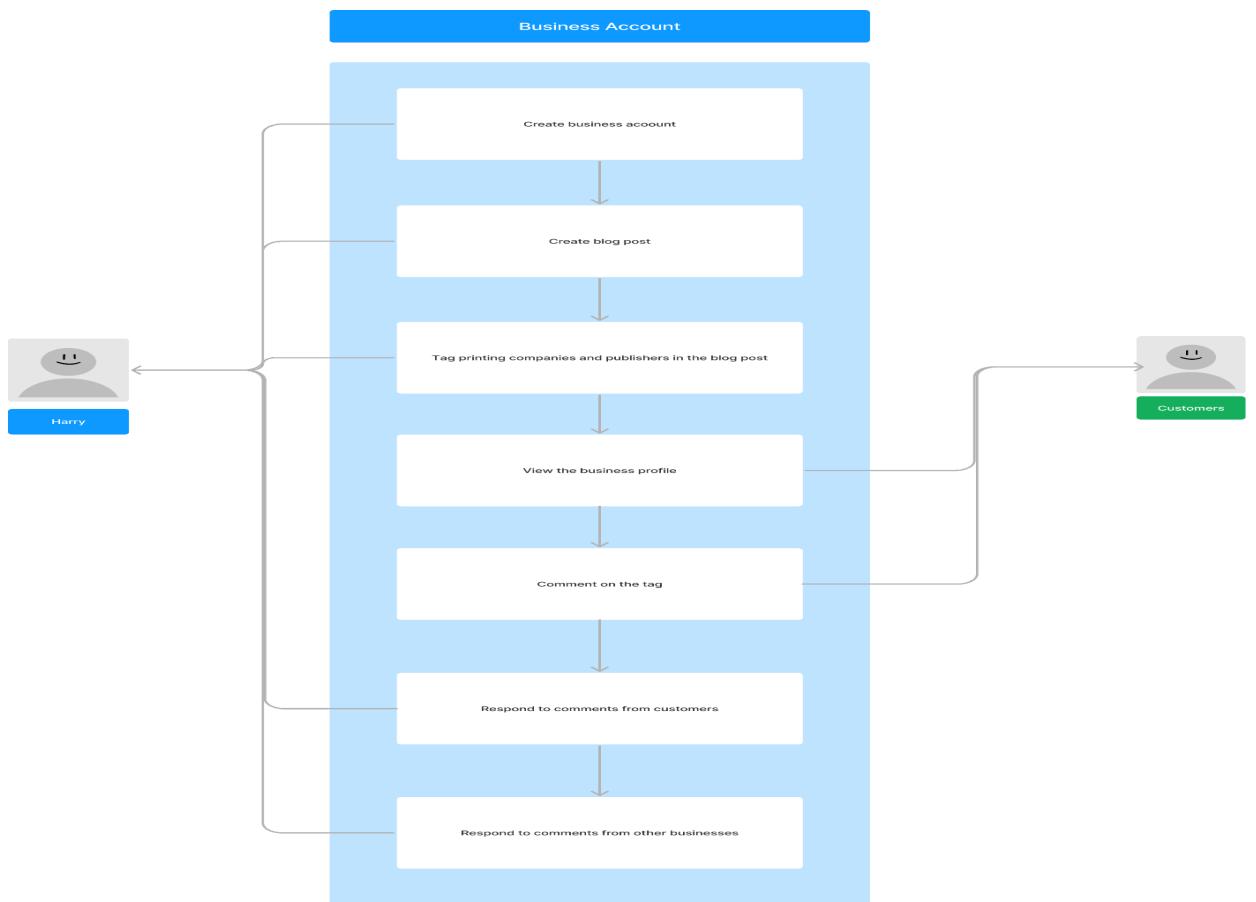


promote his business online and build a community of people who are interested in fitness and well-being.

2.9. Business Account

Actors: Registered User: has an account and can post private/public blogs to promote his business.

Description: Harry runs a paper company and he uses a marketing tool called Piqued to promote his business online. Piqued is a platform that allows Harry to create and share blog posts about his business, which can reach a wider audience on the internet. In his blog posts, Harry updates people on how he works with some good printing companies and publishers. He can also tag these businesses in his blog posts, which helps people see how these businesses are connected with each other. This allows Harry to broadcast his business to a wider audience and showcase the partnerships he has established. Furthermore, Piqued allows Harry to see comments from people all around the world who are interested in his business. This means that Harry can interact with potential customers and respond to their questions or concerns. Additionally, Piqued also allows Harry to see comments from businesses who are interested in working with him. This can help

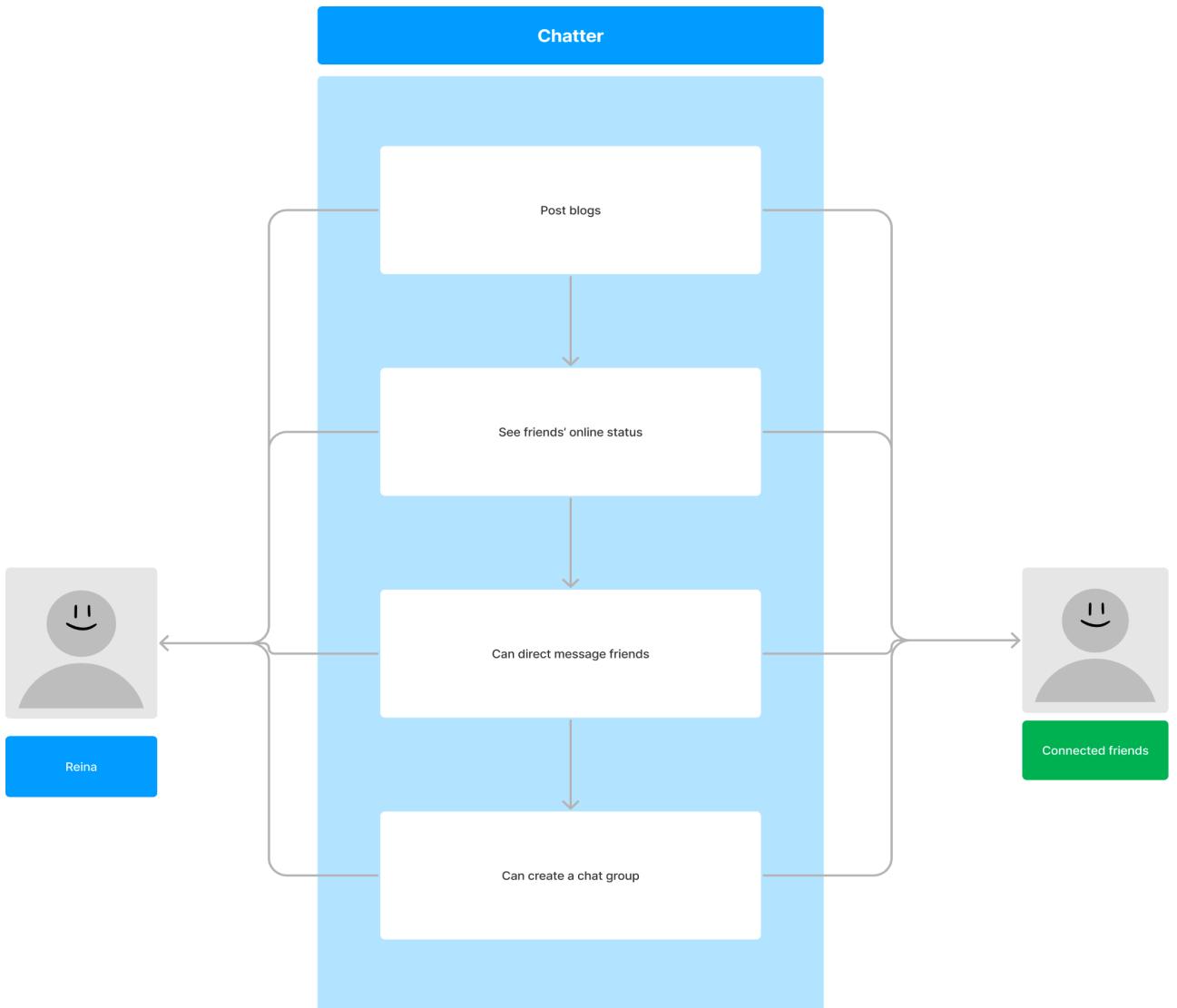


Harry expand his network and potentially establish new partnerships in the industry.

2.10. Chatter

Actors: Registered User: has an account and can chat with connected friends privately.

Description: Reina loves social media, and one thing she loves the most about social media is the ability to connect with her friends and family online as well as interact with them. Reina spends a lot of time blogging about her interests and telling her friends about it. Reina spends at least 4 days out of her week blogging about her interesting life, whether it's a new food that she tried, a new place she discovered in her hometown, or a new restaurant that just opened up. Once she posts a blog, she likes to reach out to her friends to let them know what's new in



her life and to encourage them to check out her blog. She loves interacting with them to get their opinion on things and to hear that she had Piqued their interests.

3. List of Main Data Items and Entities:

3.1. Data Description

- 3.1.1. General user: Individuals who do not have a registered account but can search and view public posts on the website. They can also view comments and reactions on public posts.
- 3.1.2. Registered user: Individuals who can log into their account, create, and publish posts on the website. They have the ability to edit and delete their own posts and comments, manage their account information, and view site statistics.
- 3.1.3. Site Moderators: Administrators who manage the frontend and backend of the website.
- 3.1.4. Posts/Articles: Main content pieces on the website, written by users/authors. They contain text, images, and/or videos and can be categorized and tagged.
- 3.1.5. Comments: Responses to posts made by users on the website.
- 3.1.6. Reactions: Emotional responses on posts.
- 3.1.7. Company's "about us" page: Information about each of the developers of the website.
- 3.1.8. Company's description: Information about the website and its purpose.
- 3.1.9. Categories: Groupings of posts based on topic or theme
- 3.1.10. Search bar: A search function that allows users to search blogs.
- 3.1.11. Images: Visual content that can be used to enhance posts and improve engagement.
- 3.1.12. Advertising Banners: Display ads that generate revenue for the website.
- 3.1.13. Social Media Links: Links to the website's social media profiles.
- 3.1.14. Website logo: Display the company's logo on the website to represent the company
- 3.1.15. Notification: Represents a notification to be sent to a user.
- 3.1.16. Navigation bar: A section where links are grouped together for easy navigation.
- 3.1.17. Profile link: A link where the user can click on to go to their account page to edit their account.
- 3.1.18. Home page link: A link where the users can navigate back to the main domain.

4. Initial List of Functional Requirements

4.1. All Users

- 4.1.1. All users shall be able to view the website's information
- 4.1.2. All users shall be able to contact the site moderators with questions
- 4.1.3. All users shall be able to view the privacy policy
- 4.1.4. All users shall be able to view the terms and conditions

4.2. Guest Users

- 4.2.1. Guest users shall be able to create an account
- 4.2.2. Guest users shall be able to use a unique username to create an account
- 4.2.3. Guest users shall be able to only view public posts
- 4.2.4. Guest users shall be able to view the contents of public posts
- 4.2.5. Guest users shall be able to view comments on public posts
- 4.2.6. Guest users shall be able to view reactions on public posts
- 4.2.7. Guest users shall be able to view a profile's public information
- 4.2.8. Guest users shall be able to search for public posts
- 4.2.9. Guest users shall be able to discover other public users

4.3. Registered Users

- 4.3.1. Registered users shall be able to log in
- 4.3.2. Registered users shall be able to log out
- 4.3.3. Registered users shall be able to sign in with both their email and username.
- 4.3.4. Registered users shall be able to request to reset a new password if they forget it
- 4.3.5. Registered users shall be able to retrieve their username if they forget it
- 4.3.6. Registered users shall be able to add a profile picture
- 4.3.7. Registered users shall be able to customize profile using the standard format template
- 4.3.8. Registered users shall be able to edit their profiles
- 4.3.9. Registered users shall be able to change their profile picture
- 4.3.10. Registered users shall be able to change their displayed username
- 4.3.11. Registered users shall be able to change their email address
- 4.3.12. Registered users shall be able to change their DOB
- 4.3.13. Registered users shall be able to change what information is displayed publicly to general users.
- 4.3.14. Registered users shall be able to add contact information
- 4.3.15. Registered users shall be able to make contact information private or public
- 4.3.16. Registered users shall be able to make their information private
- 4.3.17. Registered users shall be able to make some of their information private
- 4.3.18. Registered users shall be able to make some of their information public

- 4.3.19. Registered users shall be able to change their password
- 4.3.20. Registered users shall be able to search for friends by username or email
- 4.3.21. Registered users shall be able to see pending friend requests
- 4.3.22. Registered users shall be able to accept friend requests
- 4.3.23. Registered users shall be able to reject friend requests
- 4.3.24. Registered users shall be able to remove friends
- 4.3.25. Registered users shall be able to create blog posts.
- 4.3.26. Registered users shall be able to search for posts
- 4.3.27. Registered users shall be able to update their posts
- 4.3.28. Registered users shall be able to delete their posts
- 4.3.29. Registered users shall be able to disable comments for their posts
- 4.3.30. Registered users shall be able to add comments on post
- 4.3.31. Registered users shall be able to edit their own comments
- 4.3.32. Registered users shall be able to delete their own comments
- 4.3.33. Registered users shall be able to like/react to other people's comments
- 4.3.34. Registered users shall be able to view friends' posts
- 4.3.35. Registered users shall be able to comment on friends' posts
- 4.3.36. Registered users shall be able to react to friends' posts
- 4.3.37. Registered users shall be able to repost public posts
- 4.3.38. Registered users shall be able to send private messages.
- 4.3.39. Registered users shall be able to receive private messages.

4.4. Administrator Users

- 4.4.1. Administrator users shall be able to monitor their account
- 4.4.2. Administrator users shall be able to create authorized users on account
- 4.4.3. Administrator users shall be able to limit authorized users access on account

4.5. Site moderators

- 4.5.1. Site moderators shall be able to delete posts/articles
- 4.5.2. Site moderators shall be able to reinstate accounts
- 4.5.3. Site moderators shall be able to ban accounts
- 4.5.4. Site moderators shall be able to review accounts
- 4.5.5. Site moderators shall be able to temporarily block accounts for review
- 4.5.6. Site moderators shall be able to edit the layout of the platform
- 4.5.7. Site moderators shall be able to add additional features
- 4.5.8. Site moderators shall be able to change template layouts for profiles
- 4.5.9. Site moderators shall be able to remove spam accounts
- 4.5.10. Site moderators shall be able to answer users' questions and concerns
- 4.5.11. Site moderators shall be able to create an account
- 4.5.12. Site moderators shall be able to post on the platform's main page
- 4.5.13. Site moderators shall be able to make site announcements

- 4.5.14. Site moderators shall be able to update website
- 4.5.15. Site moderators shall be able to temporarily suspend the website for maintenance

5. List of Non-Functional Requirements

5.1. Performance:

- 5.1.1. The website should be available 24/7 with a minimum of downtime for maintenance and updates.
- 5.1.2. Scalability: The website should be able to handle a large number of users and posts without slowing down or crashing.

5.2. Security:

- 5.2.1. Passwords must be encrypted in the database.
- 5.2.2. Each account should have a unique email.
- 5.2.3. Each account should have a unique username.
- 5.2.4. Logged in users must stay logged in while navigating unless they log out.

5.3. Legal:

- 5.3.1. To complete registration, users must accept T&C.
- 5.3.2. Privacy Policy.

5.4. Compatibility:

- 5.4.1. The website should be compatible with a range of devices, browsers, and operating systems.
- 5.4.2. The UI should be responsive to all screen sizes and device form factors

5.5. Usability:

- 5.5.1. The website should be user-friendly, with a clear interface that is easy to navigate.

5.6. Marketing:

- 5.6.1. The website will have a logo to represent the company.
- 5.6.2. The website should have a description to explain what the website is about.

6. Competitive Analysis

Feature/Company	Medium	Tumblr	Twitter	Facebook	Blogger
Strengths	Easy to get started writing articles. Weekly user stories that staff recommend.	Wide net of starting topics to follow, Upcoming changes/feature/wip transparency. Easy to post. Diversity of content	Platform used for concise writing and sharing thoughts across the internet	Social media platform with the highest number of users than any other platforms, Post's word limit almost equal to 200 page book	Personal url for easy content sharing. Ability to generate passive income based on sharing interests/passion via Adsense
Weaknesses	3 free member only articles /mo. Only active on twitter + blog. Searching, No password/ easy to change email	Moderation against harmful speech, filtering, and sorting. Confusing layout (ie: 1 vs 2)	Limited characters used for each post(280 to be precise) To use advance features like editing or deleting posts users need to use paid version Users cannot link other social medias on posts	Weak protection for users data Accused of political manipulation and addiction of social media	Can only register with gmail (owned by alphabet). Lack of a community.
Pricing	\$5/mo or \$50/yr for membership(unlimited quality articles, no ads)	\$4.99/mo or \$39.99/yr for ad free	\$11/month or \$114.99/year for twitter blue Users can also use free service for limited	Not yet but will be coming in the future with a price range of \$11.99 a month on the web and \$14.99 a	N/A

			features.	month on iOS.	
Social Media	Blog , Facebook , Instagram , Twitter , Youtube	Blog , Twitter , Instagram , Facebook , Youtube	Instagram Twitter Facebook Youtube	Youtube Instagram	Blogger
Onboarding Experience	Short; only email, name, and at least 3 topics and then left to learn how to navigate site	Decent amount of steps to get started.	Easy to set up, choose among wide variety of topics interested in	Need an email account or phone number to create an account Few steps to get started using the platform	Account set up in 3 steps. No info/tips/guide on managing a website.

Feature	Medium	Tumblr	Twitter	Facebook	Blogger	Piqued
Search	+	+	+	+	-	+
Browse	+	+	+	+	-	+
Content Management	+	+	+	+	+	++
Social Group (ie: friends)	-	+	+	+	-	+
Private Messaging	-	+	+	+	-	+

7. High-level system architecture and technologies used

- Server Host: **Amazon Web Services**
- Server Operating System: **Ubuntu/Linux**
- Supported Browsers : **Safari, Chrome**
- Server Host: **AWS EC2**
- Web Server: **NGINX**
- Server-Side Language: **JavaScript**
- Front-end Language: **HTML, CSS, JavaScript**
- Database: **MySQL**
- Additional Technologies:
 - Web Frameworks: **Svelte**
 - IDEs: **IntelliJ and VSCode**
 - Other Frameworks : **NodeJS, ExpressJS**

8. 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
 - Done
- 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

9. List of Team Contributions

Name	Duties	Contribution Score
Nishit	Set up AWS, set up our databases, wrote some use cases, table of contents, and competitive analysis	9
Gautami	Functional requirements, non-functional requirements, entities, and use cases	8
Joshua	Competitive analysis, functional requirements, and use case	6
Leo	Executive summary, title page, table of contents, use cases, functional requirements, document formatting, and entities.	9
Andy	Use cases, functional requirements, index page for the about us page, and set up Svelte	8
Jose	Delegated task, scheduled meetings, set reminders for the team, help come up with use cases, reviewed the work, helped with the system and technology stack, answered questions, and helped a hand when needed.	

SW Engineering CSC648/848 Spring 2023

Piqued

Team 5

Jose Avila, Team Lead, javila6@mail.sfsu.edu

Andy Shi, Frontend Lead, ashi2@mail.sfsu.edu

Leo Saeteurn, Backend Lead, lisaeteurn@mail.sfsu.edu

Nishit Pachchigar, Github Master, npachchigar@mail.sfsu.edu

Joshua Hayes, Database Master, jhayes10@mail.sfsu.edu

Gautami Kollolu Srinivasa, Document Editor, gkollolsrunivasa@mail.sfsu.edu

Milestone 2

March 30, 2023

History Table

Version	Submission Date
M2V2	April 19, 2023
M2V1	March 30, 2023
M1V2	April 16, 2023
M1V1	March 5, 2023

Table of Contents

Cover Page	1
Table of Contents	2
Data Definitions	3
Priority Functional Requirements	5
Priority 1	5
Priority 2	8
Priority 3	9
UI Mockups and Storyboards (high level only)	10
Homepage(Logged Out)	10
Homepage(Logged In)	11
Homepages(Mobile)	12
Various Navigation Bar Ideas	13
Log In/Sign Up Page (Idea 1)	14
Log In/Sign Up Page (Idea 2)	15
Create Post (Idea 1)	16
Create Post (Idea 2)	17
Create Post (Idea 3)	18
Post (Logged Out)	19
Post (Signed In)	20
Search	21
Log In/Sign Up User Flow	22
Create Post User Flow	23
Search and View Post User Flow	24
High level database architecture and organization	25
High Level APIs and Main Algorithms	29
High Level UML Diagrams	31
High Level Application Network and Deployment Diagrams	32
Identify <i>actual</i> key risks for your project at this time	34
Project management	35
Detailed list of contributions	36

1) Data Definitions

- a) Guest user: Individuals who do not have a registered account but can search and view public posts on the website. They can also view comments and reactions on public posts.
- b) Registered user: Individuals who can log into their account, create, and publish posts on the website. They have the ability to edit and delete their own posts and comments, manage their account information, and view site statistics.
- c) Site Moderators: Administrators who manage the frontend and backend of the website.
- d) Blog Posts/Articles: Main content pieces on the website, written by users/authors. They contain text, images, and/or videos and can be categorized and tagged.
 - i) Author: The user who originally posted the article.
 - ii) Published timestamp: A timestamp that represents when an article was originally posted.
 - iii) Updated timestamp: A timestamp that represents when an article was last updated by the author.
 - iv) Images: Visual content that can be used to enhance posts and improve engagement.
 - v) Categories: Groupings of posts based on topic or theme
 - vi) Tags/Hashtag: Labels which can be added to a post to define its category and determine how it is found through search or recommended to other users.
 - vii) Comments: Rich text responses to posts made by users on the website.
 - viii) Reactions: Emotional responses on posts. This may include likes, remote responses, or any other form of instant feedback.
 - (1) Like: A positive response to a post by a registered user initiated by a simple button click.
 - (2) Reaction: Any other form of response by a registered user that can be either positive or negative.
 - (3) Reaction count: A number which indicates the amount of reactions to a post.
- e) Repost: A link to another user's post that appears on a user's profile.
- f) Recommendations: Algorithmically generated list of posts based on the user's interests.
- g) Feed/For You Page: A landing page for logged-in users which shows a scrollable list of recommended posts.
- h) Search: A search function that allows users to search blogs.

- i) Search filter: Function to filter out results based on certain criteria. This may be based on the aforementioned categories.
- ii) Search sort: Function to sort the results based on a certain order.
- i) Editor: UI for editing a blog post. This can feature a rich text editor where users can apply formatting to text and insert images, video, and other types of media.
- j) Advertising Banners: Display ads that generate revenue for the website.
- k) Notification: Represents a notification sent to a user. This can be a new follower, a comment on a post, etc.
- l) Profile Page: A customizable page which contains a user's public information and posts.
 - i) Biography/Bio: A short, user-provided paragraph about the registered user.
- m) Navigation bar/Header: A section where links are grouped together for easy navigation.
 - i) Website logo: Display the company's logo on the website to represent the company
 - ii) Profile link: A link where the user can click on to go to their account page to edit their account.
 - iii) Home page link: A link where the users can navigate back to the main domain.
 - iv) Navigation links: Links to various primary pages or pages commonly accessed by users.
 - v) Hamburger Menu: A menu which contains navigation items to pages. This may be used on mobile for containing primary navigation links, or to allow users to quickly access followed pages.
- n) Footer: An section at the bottom of the page which contains informational links.
 - i) Company's "about us" page: Information about each of the developers of the website.
 - ii) Company's description: Information about the website and its purpose.
 - iii) Social Media Links: Links to the website's social media profiles.
 - iv) Privacy policy and Terms of Service: Links for legal purposes.

2) Prioritized Functional Requirements

a) Priority 1 Requirements :

i) All Users

- (1) All users shall be able to view the website's information
- (2) All users shall be able to contact the site moderators with questions
- (3) All users shall be able to view the privacy policy
- (4) All users shall be able to view the terms and conditions

ii) Guest Users

- (1) Guest users shall be able to create an account
- (2) Guest users shall be able to use a unique username to create an account
- (3) Guest users shall be able to only view public posts
- (4) Guest users shall be able to view the contents of public posts
- (5) Guest users shall be able to view a profile's public information
- (6) Guest users shall be able to search for public posts
- (7) Guest users shall be able to discover other public users
- (8) Guest users shall be able to view comments on public posts
- (9) Guest users shall be able to view reactions on public posts

iii) Registered Users

- (1) Registered users shall be able to log in
- (2) Registered users shall be able to log out
- (3) Registered users shall be able to sign in with both their email and username.
- (4) Registered users shall be able to request to reset a new password if they forget it
- (5) Registered users shall be able to retrieve their username if they forget it
- (6) Registered users shall be able to add a profile picture
- (7) Registered users shall be able to edit their profiles
- (8) Registered users shall be able to change their profile picture
- (9) Registered users shall be able to change their displayed username
- (10) Registered users shall be able to change their email address
- (11) Registered users shall be able to change their DOB
- (12) Registered users shall be able to change what information is displayed publicly to general users
- (13) Registered users shall be able to add contact information
- (14) Registered users shall be able to make contact information private or public.
- (15) Registered users shall be able to change their password
- (16) Registered users shall be able to search for friends by username
- (17) Registered users shall be able to see pending friend requests

- (18) Registered users shall be able to accept friend requests
 - (19) Registered users shall be able to reject friend requests
 - (20) Registered users shall be able to remove friends
 - (21) Registered users shall be able to create blog posts.
 - (22) Registered users shall be able to add photos on blog posts.
 - (23) Registered users shall be able to search for posts.
 - (24) Registered users shall be able to update their posts.
 - (25) Registered users shall be able to delete their posts.
 - (26) Registered users shall be able to disable comments for their posts.
 - (27) Registered users shall be able to add comments on post.
 - (28) Registered users shall be able to edit their own comments.
 - (29) Registered users shall be able to delete their own comments.
 - (30) Registered users shall be able to like/react to other people's comments
 - (31) Registered users shall be able to comment on posts
 - (32) Registered users shall be able to react to posts
 - (33) Registered users shall be able to send private messages
 - (34) Registered users shall be able to receive private messages.
 - (35) Registered users shall be able to make their information private
 - (36) Registered users shall be able to make some of their information private
 - (37) Registered users shall be able to make some of their information public
- iv) Administrator Users
- (1) Administrator users shall be able to monitor their account
 - (2) Administrator users shall be able to create authorized users on account
 - (3) Administrator users shall be able to limit authorized users access on account
- v) Site moderators - creators(us)
- (1) Site moderators shall be able to delete posts/articles
 - (2) Site moderators shall be able to ban accounts
 - (3) Site moderators shall be able to temporarily block accounts for review
 - (4) Site moderators shall be able to edit the layout of the platform
 - (5) Site moderators shall be able to add additional features
 - (6) Site moderators shall be able to change template layouts for profiles
 - (7) Site moderators shall be able to remove spam accounts

- (8) Site moderators shall be able to answer users' questions and concerns
 - (9) Site moderators shall be able to create an account
 - (10) Site moderators shall be able to post on the platform's main page
 - (11) Site moderators shall be able to make site announcements
 - (12) Site moderators shall be able to update website
 - (13) Site moderators shall be able to temporarily suspend the website for maintenance
 - (14) Site moderators shall be able to reinstate accounts
 - (15) Site moderators shall be able to review accounts
- vi) User's Profile Page
 - (1) User's profile page shall have the user's picture
 - (2) User's profile page shall have the user's username
 - (3) User's profile page shall have published public posts
 - (4) User's profile page shall have a biography
 - (5) User's profile page shall have contact information that are made public by user
 - vii) All pages
 - (1) All pages shall have a search bar
 - (2) All pages shall have a navigation bar/header
 - (3) All pages shall have a footer with links to informational pages
 - (4) All pages shall have our company's logo
 - viii) User's Logged-In Interface
 - (1) User's logged-in interface shall have a notification icon that shows if there's any notification
 - (2) User's logged-in interface shall have a link to the user's account management page in the navigation bar
 - ix) Blog Posts
 - (1) Blog posts shall contain an original author
 - (2) Blog posts shall contain a date and time of creation
 - (3) Blog posts shall contain a date and time of modification to posts
 - (4) Blog posts shall contain a title
 - (5) Blog posts shall contain a photo(s)
 - (6) Blog posts shall contain a category(ies)
 - (7) Blog posts shall contain a tag(s)/hashtag(s)
 - (8) Blog posts shall contain a comment sections with comments from other users
 - (9) Blog posts shall contain a reaction(s) from other users
 - (10) Blog posts shall be able to be reposted by other users

- x) Exploration Newsfeed
 - (1) Exploration newsfeed shall contain random public posts
 - (2) Exploration newsfeed shall show recommended posts based on user's interests
 - (3) Exploration newsfeed shall have a search bar
 - (4) Exploration newsfeed shall have blog posts that can be clicked on to view full posts
- xi) Search Bar
 - (1) Search bar shall have a filter function
 - (2) Search bar shall have a sorting option
 - (3) Search bar shall have a text box to search up public blog posts by categories or tags
- xii) Navigation Bar
 - (1) Navigation bar shall have a link to the user's profile when logged in
 - (2) Navigation bar shall have a link to the home page
 - (3) Navigation bar shall have a link to the user's account management page when logged in
 - (4) Navigation bar shall have a link to the exploration page
- xiii) Footer
 - (1) Footer shall have a link to the company's information page
 - (2) Footer shall have a link to contact the company by emailing
 - (3) Footer shall have the company's contact information
 - (4) Footer shall have a link to "about us" page
 - (5) Footer shall have a copyright logo
 - (6) Footer shall have social media links
 - (7) Footer shall have a short description of the company
 - (8) Footer shall have a link to the terms and conditions
 - (9) Footer shall have a link to the privacy policy

b) Priority 2 Requirements :

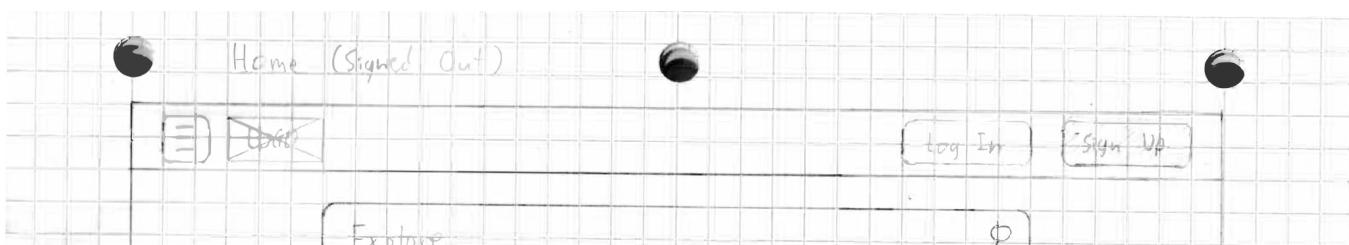
- i) Registered Users
 - (1) Registered users shall be able to customize profile using a standard format template
 - (2) Registered users shall be able to post videos on their blog post.
 - (3) Registered users shall be able to save their profile photos to make it easier for them to change back and forth
 - (4) Registered users shall be able to report other spam accounts to site moderators via a report link
- ii) Blog Posts

- (1) Blog posts shall contain videos
- iii) Chat Box
 - (1) Chat box shall contain user's username for both sender and recipient(s)
 - (2) Chat box shall contain group chat with more than two users
 - (3) Chat box shall contain all users' online status
 - (4) Chat box shall contain messages from all users within chat
 - (5) Chat box shall contain date and time of messages sent
- c) Priority 3 Requirements:
 - i) Guest users
 - (1) Guest users shall be able to send a private message to a user
 - ii) User's Profile Page
 - (1) User's profile page shall have their online status
 - (2) User's profile page shall have a standard interface template that users can use to create a profile
 - iii) Blog Posts
 - (1) Blog posts shall have a standard interface template that users can use to create a blog post
 - iv) All Pages
 - (1) All pages shall have a advertisement banner
 - v) Chat box
 - (1) Chat box shall contain emojis
 - (2) Chat box shall contain reactions

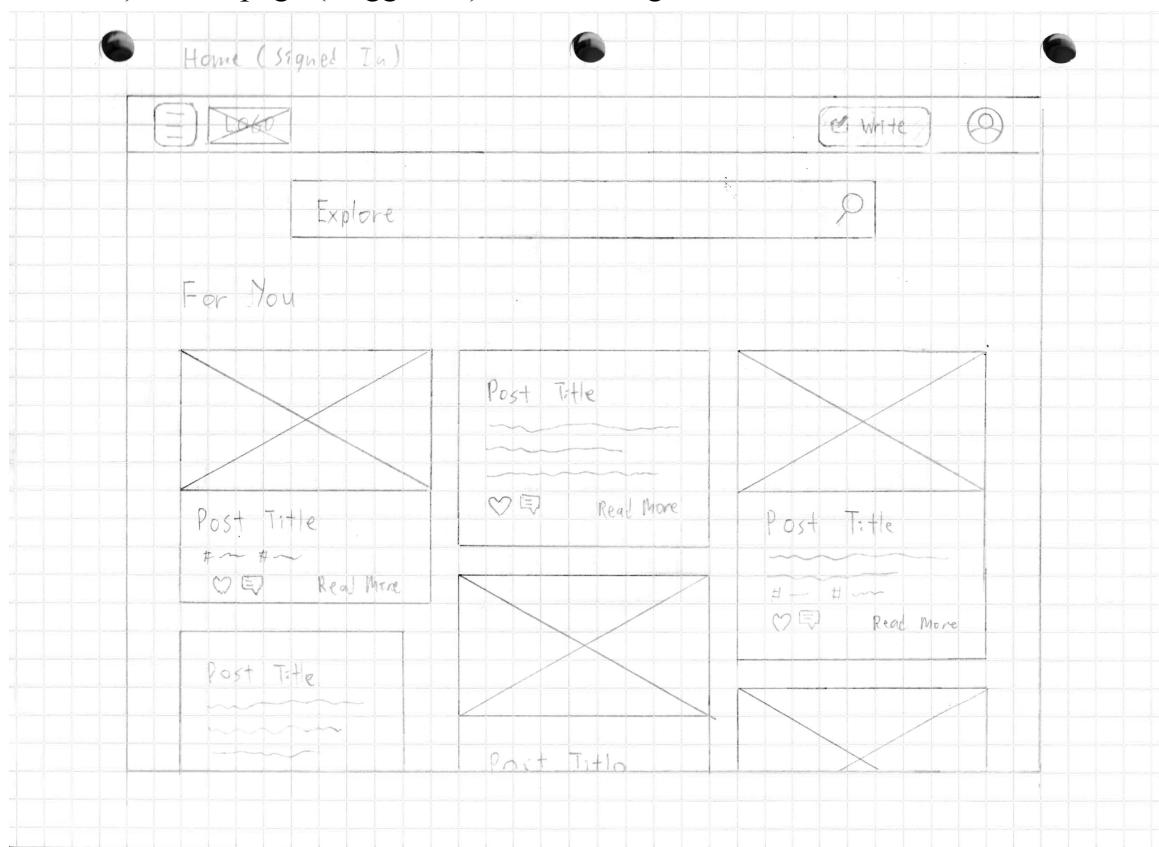
3) UI Mockups and Storyboards (high level only)

Note: First 12 images are single page wireframes that show pages in detail. 13-16 shows user flow/storyboards.

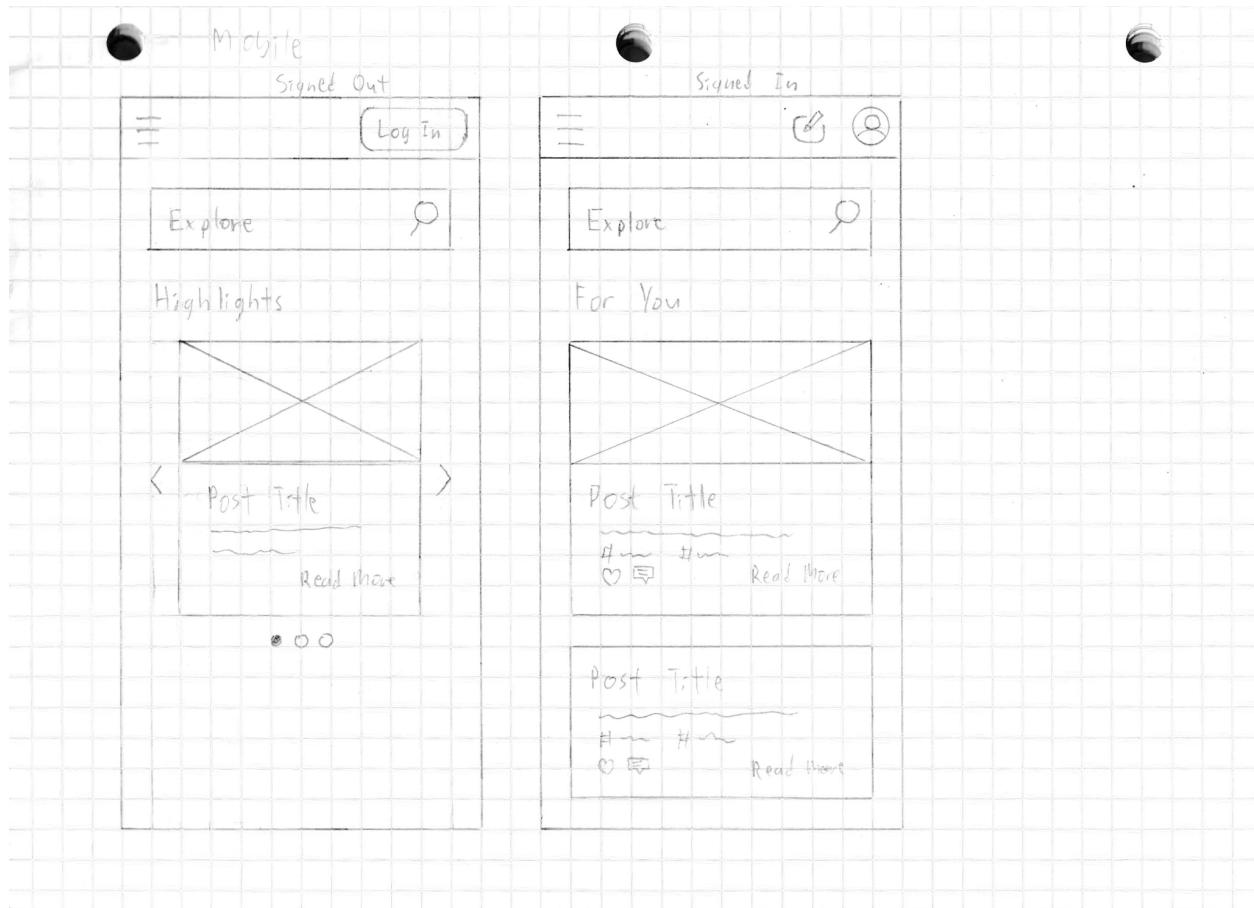
a) Homepage (Logged Out) - Wireframe



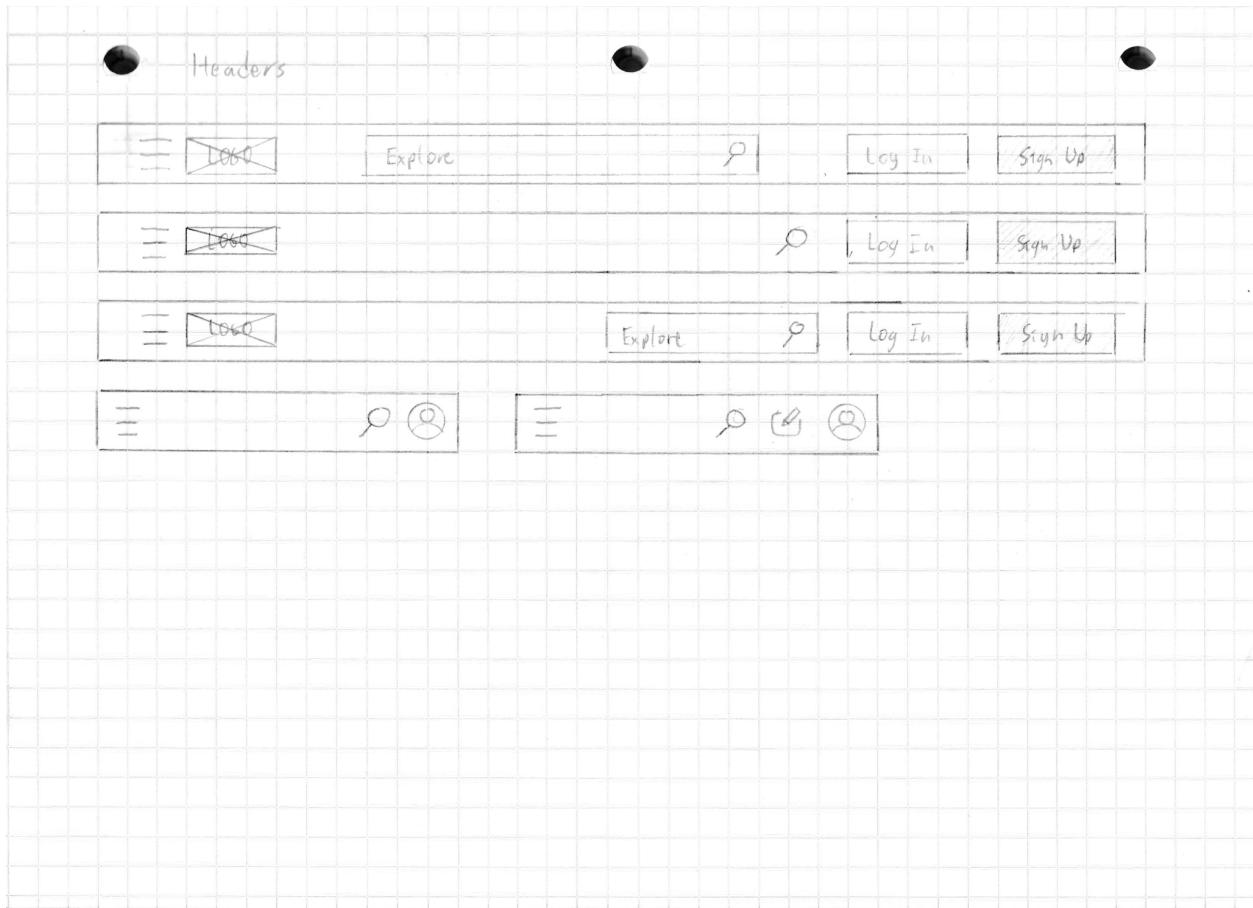
b) Homepage (Logged In)/For You Page - Wireframe



c) Homepages (Mobile) - Wireframe



d) Various Navigation Bar Ideas - Wireframe



e) Log In/Sign Up Page (Idea 1) - Wireframe

Sign Up / Log In V1

Home

Sign Up

Already have an account?
Log In

Username

Email

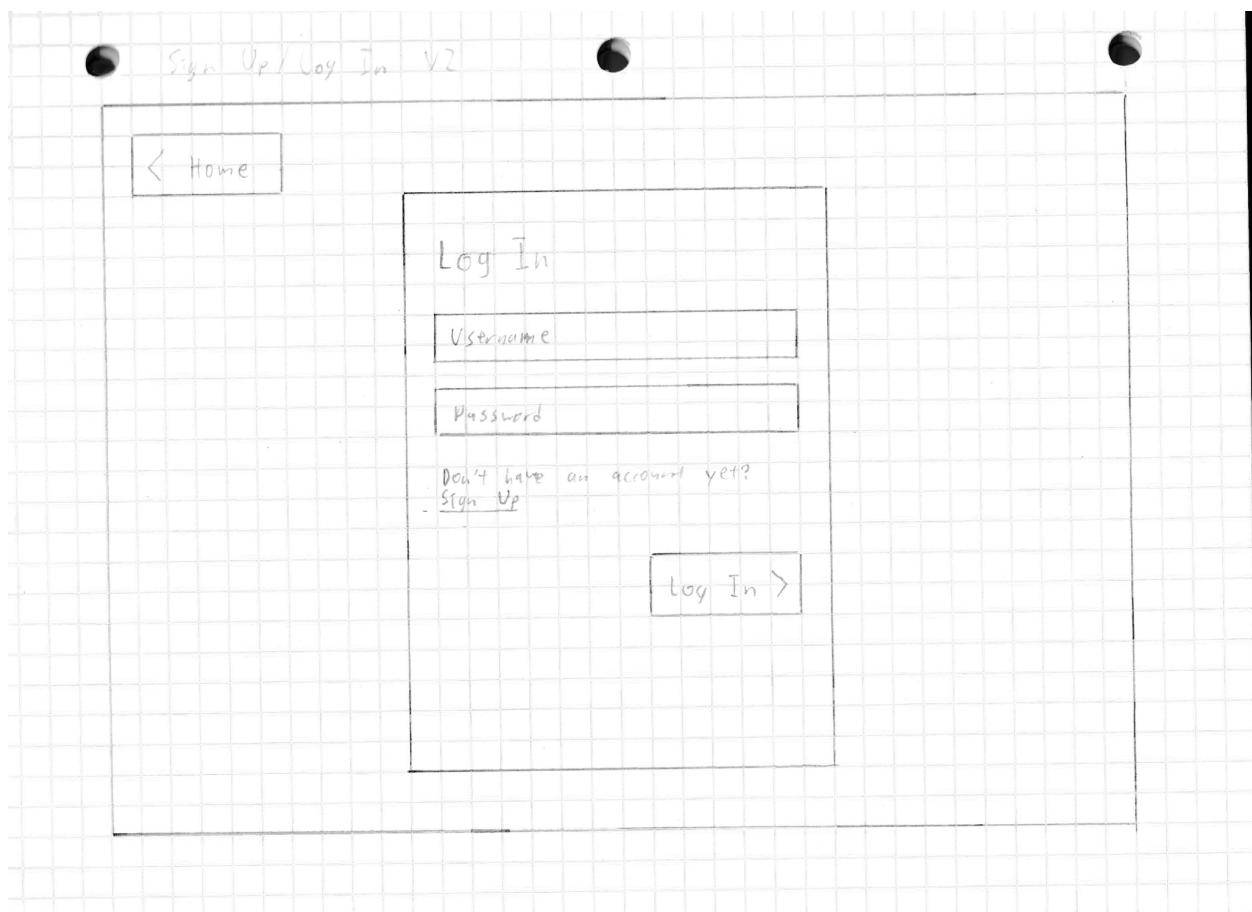
Password

Confirm Password

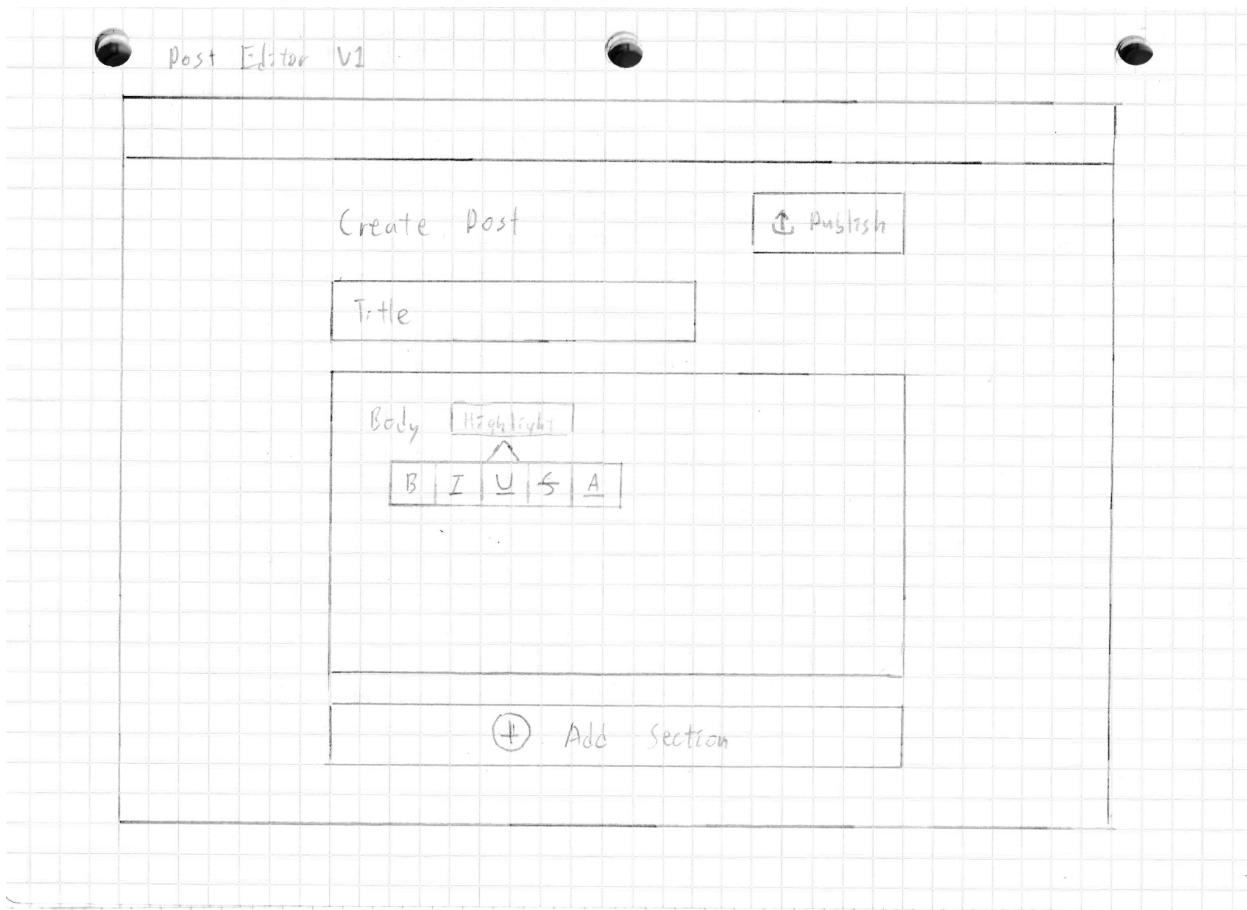
By signing up, you are
agreeing to our TOS.

Sign Up >

f) Log In/Sign Up Page (Idea 2) - Wireframe



g) Create Post (Idea 1) - Wireframe



h) Create Post (Idea 2) - Wireframe

Editor Idea (Alt)

Title	<input checked="" type="checkbox"/> Private
B I U S A A ½ ¾ ½ ¾	
Body	

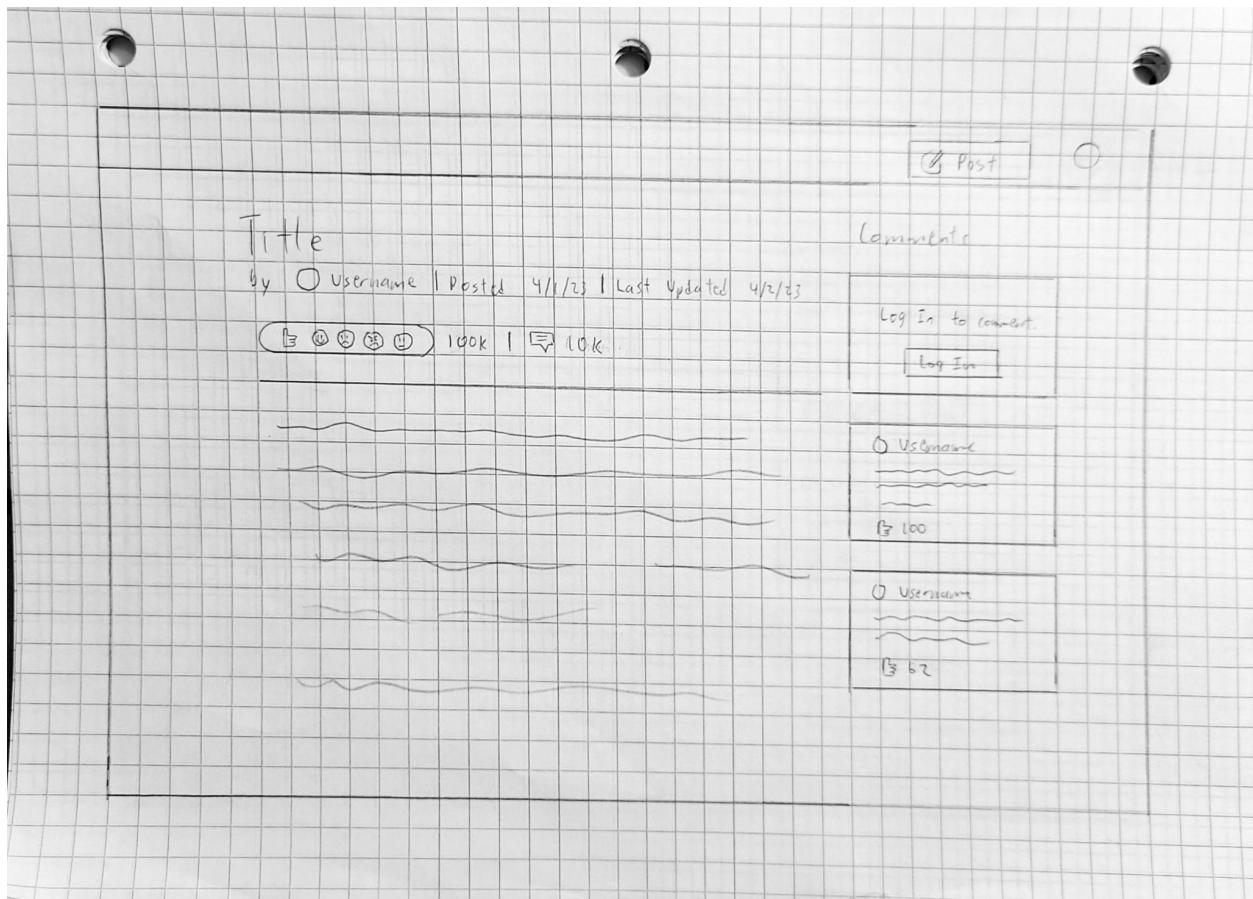
i) Create Post (Idea 3) - Wireframe

A hand-drawn wireframe for a post creation interface, likely a mobile application, on a sheet of graph paper.

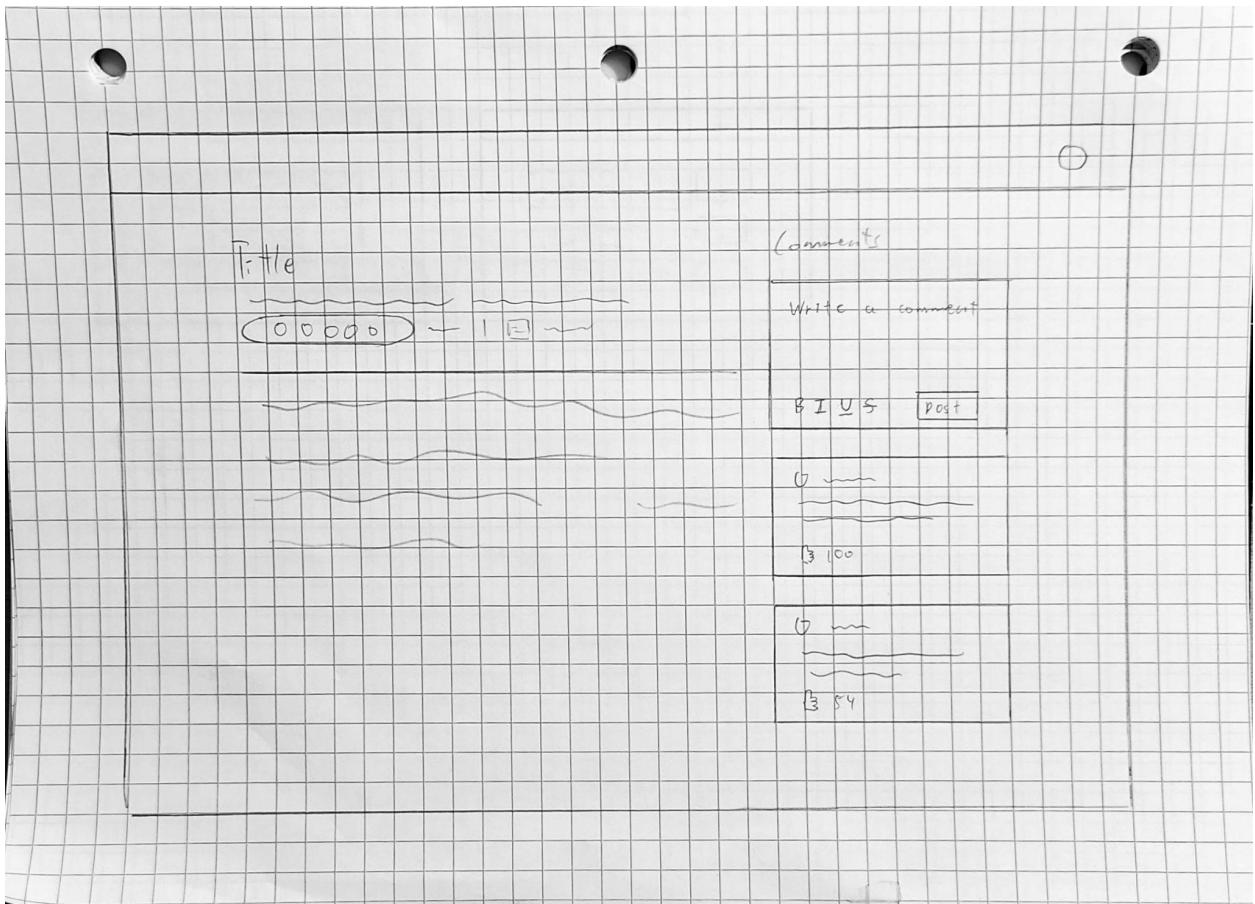
The wireframe consists of several rectangular boxes:

- A small rectangular box at the top left labeled "Title".
- A large rectangular box below it labeled "Body".
- A vertical rectangular column on the right side containing three checkboxes:
 - Make private
 - Disable comments
 - Disable likes
- A rectangular button at the bottom of the column labeled "Publish" with an upward arrow icon.

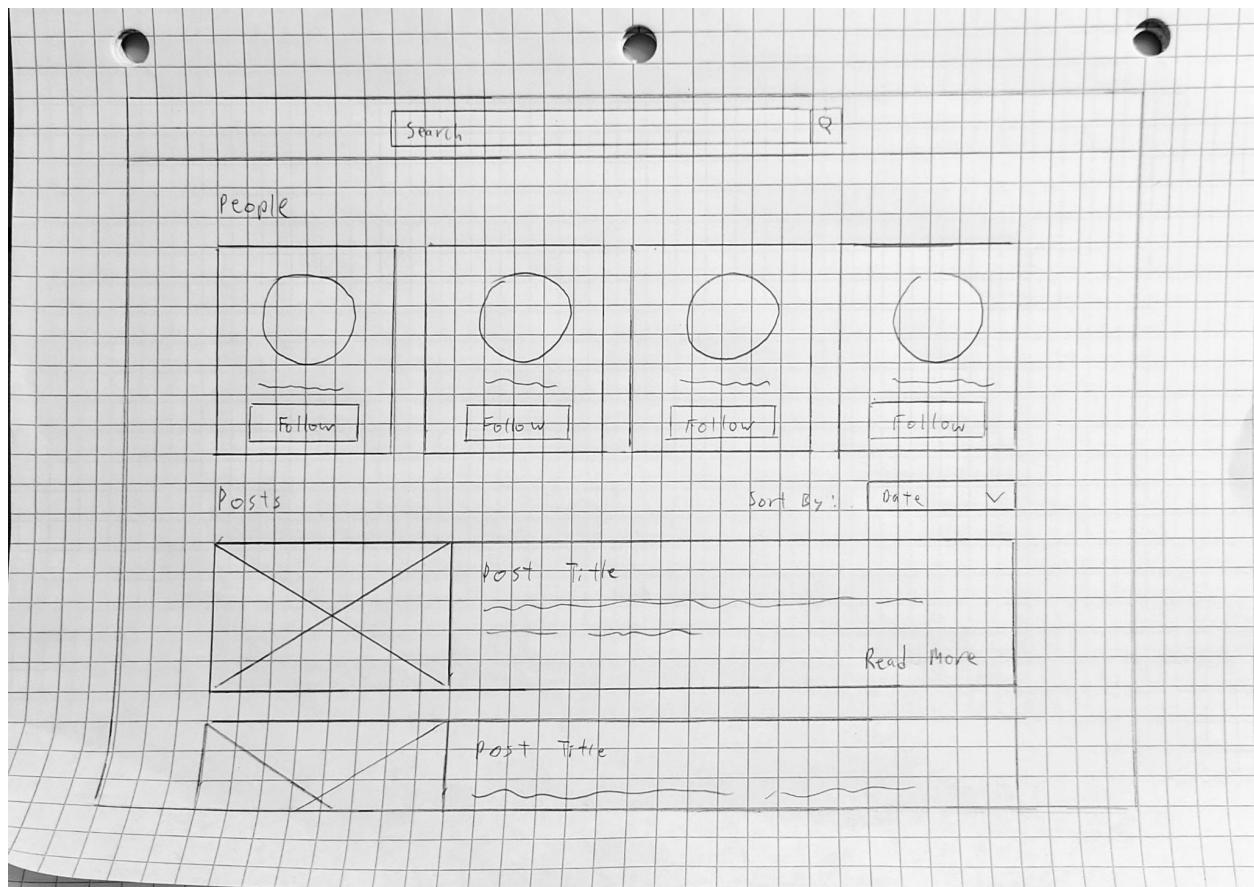
j) Post (Logged Out) - Wireframe



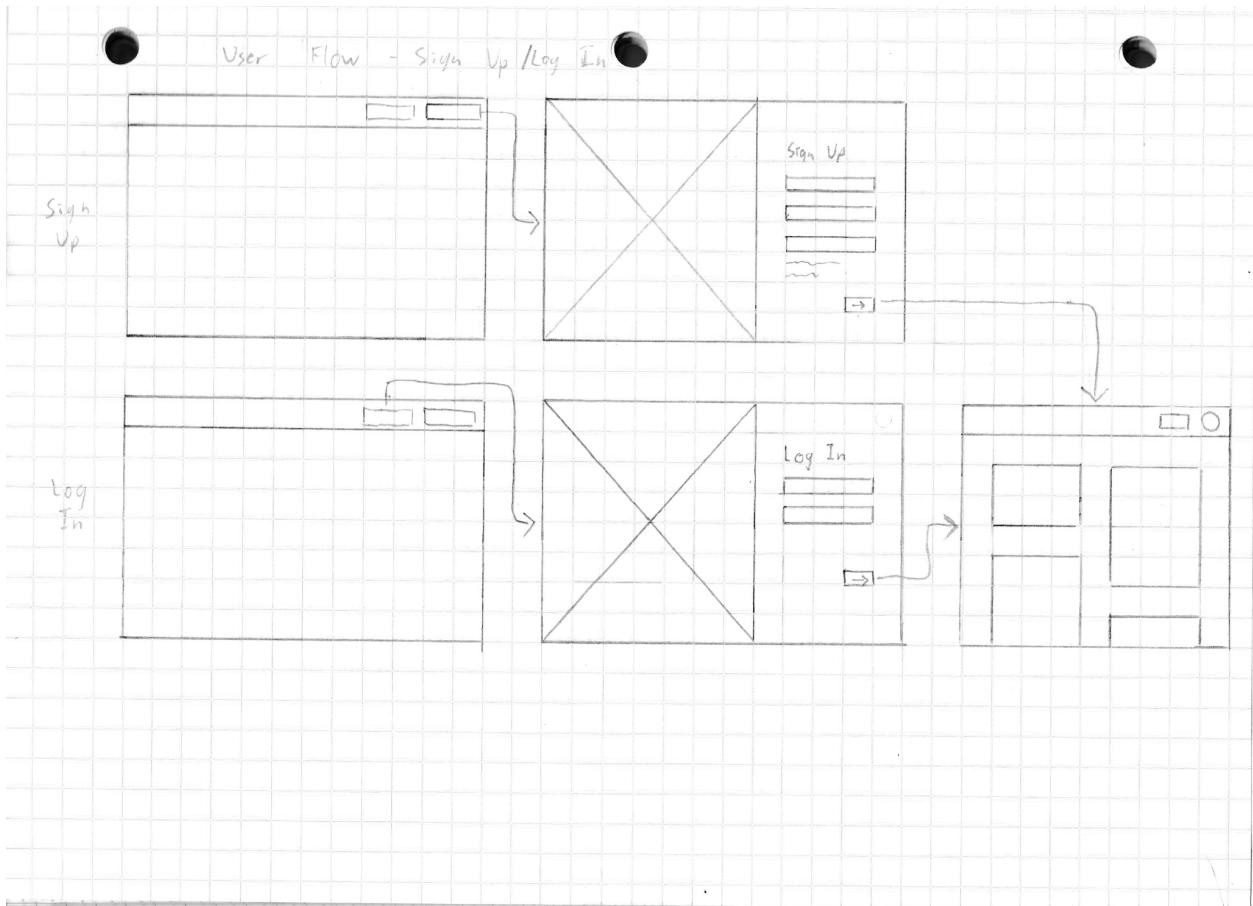
k) Post (Signed In) - Wireframe



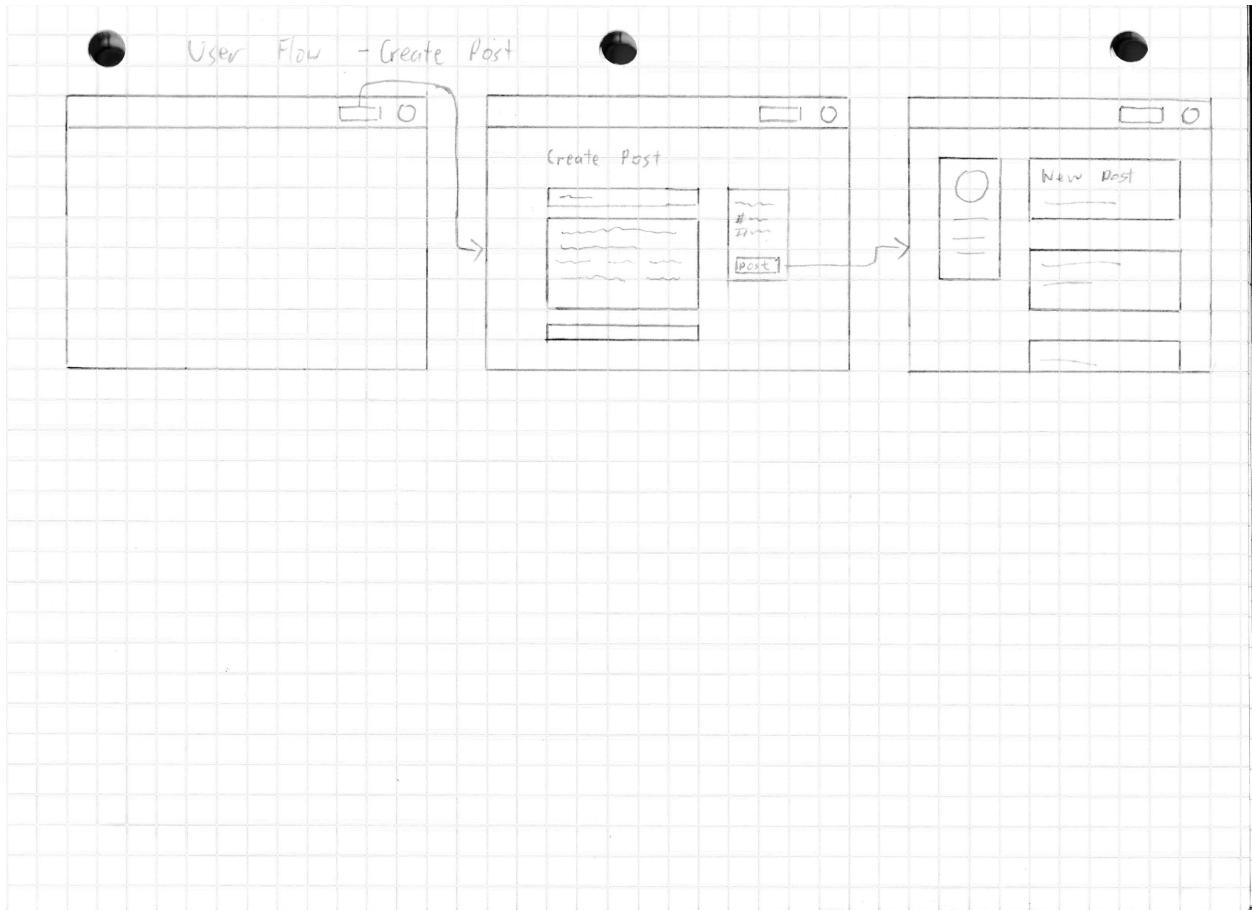
1) Search - Wireframe



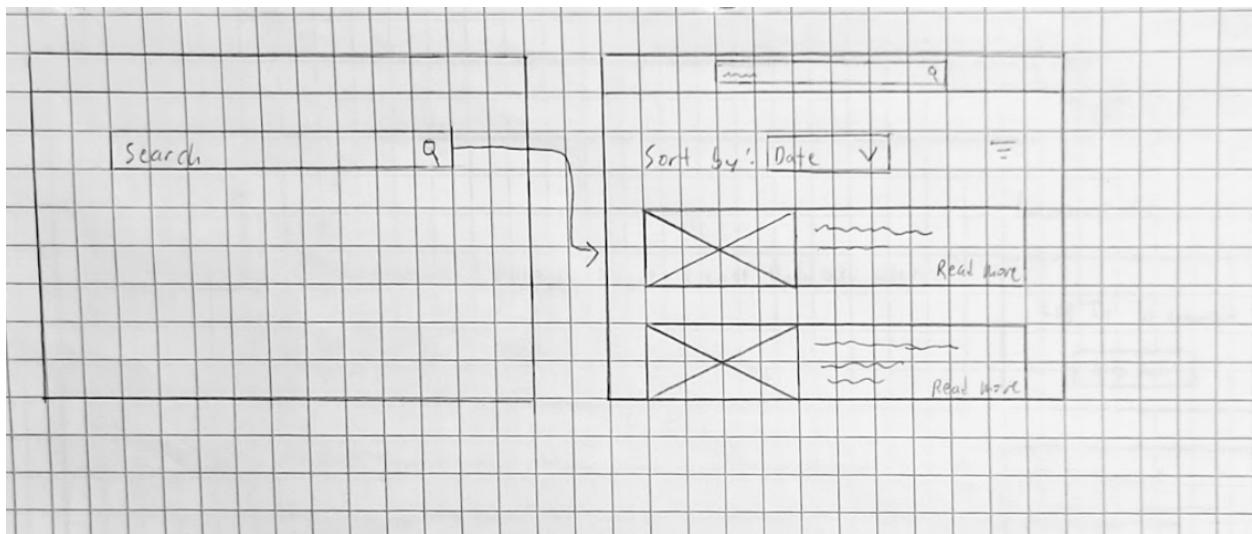
m) Log In/Sign Up User Flow - Storyboard



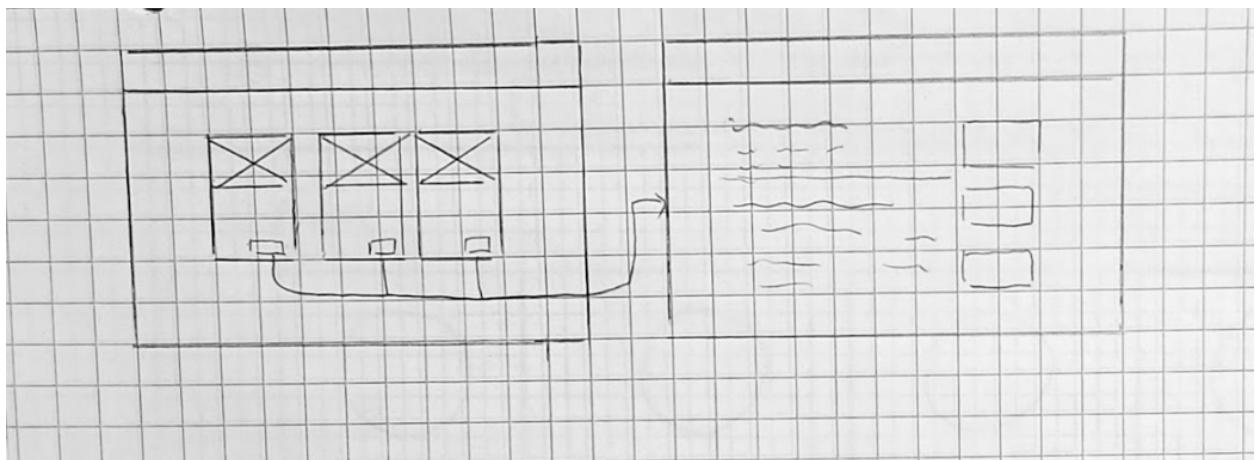
n) Create Post User Flow - Storyboard



o) Search User Flow - Storyboard



p) View Post User Flow - Storyboard



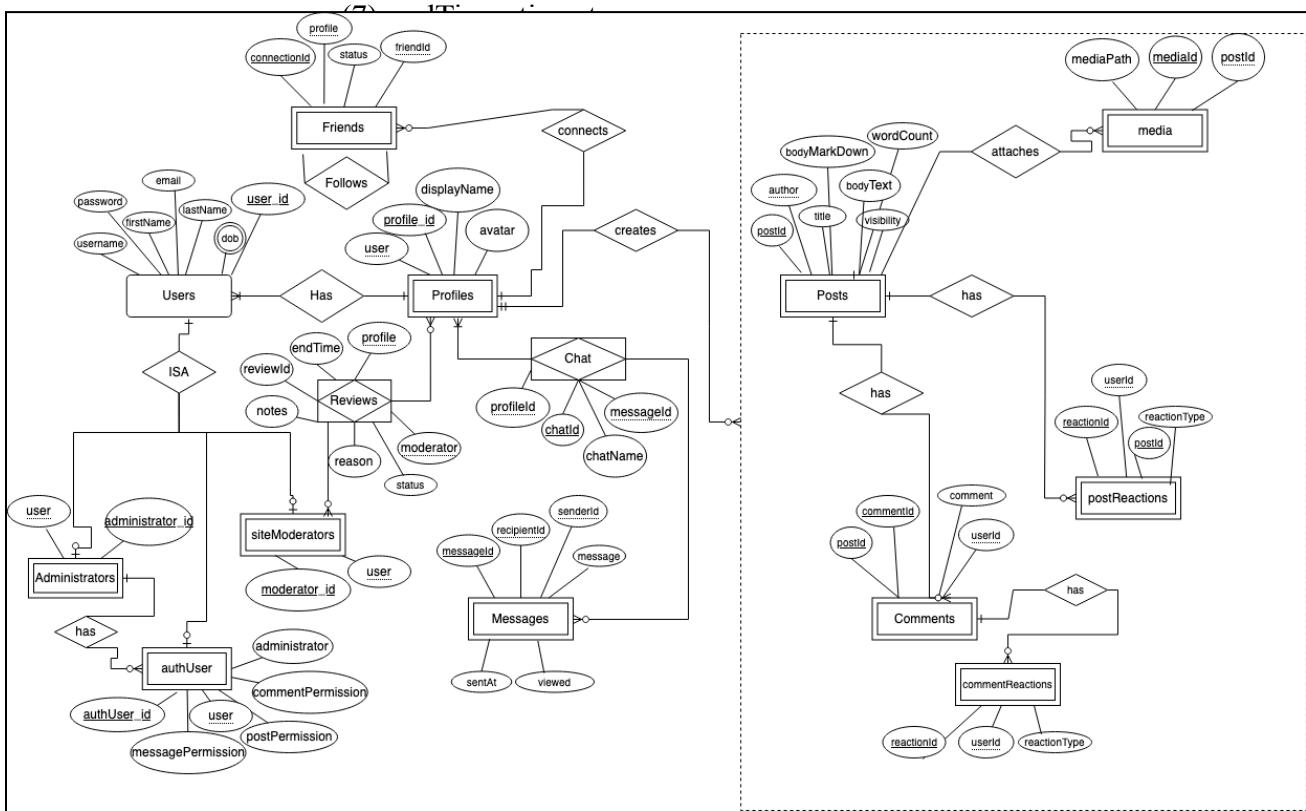
4) High level database architecture and organization

- a) Database requirements
 - i) Users shall be able to create many posts.
 - ii) A post shall be created by only one user.
 - iii) Users shall be able to create many comments.
 - iv) A comment shall belong to only one user.
 - v) A post shall have many comments.
 - vi) A comment shall belong to only one post.
 - vii) A user shall be able to have many friends.
 - viii) A friend shall belong to only one user.
 - ix) A user shall be able to send many private messages
 - x) A private message shall be sent to only one user.
 - xi) A post shall have many reactions.
 - xii) A reaction shall belong to only one post.
 - xiii) A post shall have many many photos.
 - xiv) A photo shall belong to only one post.
- b) Entities and Attributes
 - i) Users: Strong
 - (1) userId: key, numeric
 - (2) username: alphanumeric, unique
 - (3) name: composite, multi-value, alphanumeric
 - (4) email: composite, alphanumeric, unique
 - (5) password: alphanumeric
 - (6) dob: multi-value, date
 - (7) lastLogin: multi-value, timestamp
 - (8) createdAt: multi-value, timestamp
 - ii) Profiles: Weak
 - (1) profileId: key, numeric
 - (2) avatar:
 - (3) displayName: alphanumeric, unique
 - iii) Posts: Weak
 - (1) postId: key, numeric
 - (2) authorId: key, numeric
 - (3) title: alphanumeric
 - (4) bodyText: alphanumeric
 - (5) bodyMarkdown:
 - (6) visibility: alphanumeric
 - (7) wordCount: numeric
 - (8) commentsAllowed: alphanumeric
 - (9) createdAt: multi-value, timestamp

- (10) lastModified: multi-value, timestamp
- iv) Reactions (posts): Weak
 - (1) reactionId: key, numeric
 - (2) postId: key, numeric
 - (3) userId: key, numeric
 - (4) reactionType: alphanumeric
 - (5) lastModified: multi-value, timestamp
- v) Comments: Weak
 - (1) commentId: key, numeric
 - (2) authorId: key, numeric
 - (3) postId: key, numeric
 - (4) comment: alphanumeric
 - (5) createdAt: multi-value, timestamp
 - (6) lastModified: multi-value, timestamp
- vi) Reactions (comments): Weak
 - (1) reactionId: key, numeric
 - (2) commentId: key, numeric
 - (3) userId: key, numeric
 - (4) reactionType: alphanumeric
 - (5) lastModified: multi-value, timestamp
- vii) Friends: Weak
 - (1) friendId: key, numeric
 - (2) profileId: key, numeric
 - (3) friend: key, numeric
 - (4) status: alphanumeric
 - (5) createdAt: multi-value, timestamp
 - (6) lastModified: multi-value, timestamp
- viii) Chat: Weak
 - (1) chatId: key, numeric
 - (2) profileId: key, numeric
 - (3) chatName: alphanumeric
 - (4) messageId: key, numeric
- ix) Moderators: Weak
 - (1) moderatorId: key, numeric
 - (2) userId: key, numeric
- x) Reviews: Weak
 - (1) reviewId: key, numeric
 - (2) moderator: key, numeric
 - (3) status: numeric
 - (4) profile: key, numeric

(5) reason: alphanumeric

(6) notes: alphanumeric



c) Entity Relationship Diagram (ERD)

- d) We will be using MySQL as our DBMS for its ease of use and flexibility for setting up/operating on data.
- e) Media Storage:
 - i) Media(images) shall be stored on the file system to reduce overhead on CRUD operations and improve scalability.
- f) Search/filter architecture and implementation:
 - i) In the backend, there would be an API function, GET, to render the input from the user, the user can be a guest user or a registered user. In the search bar, the users can search up keywords, which will be retrieved from frontend and will be parsed and saved according to the context of the posts such as post ID, user ID, title, body, etc. The keywords are essentially categories, such as places like California or Spain, and restaurants like McDonald's. With the database, there will be a table that lists all public blog posts contents that are separated by specific parts of the content. Each new blog post that a registered user posts as public posts, they will be stored based on the specific category that they choose, and they can choose multiple categories for one post.

5) **High Level APIs and Main Algorithms**

Some of our APIs will include a user login authentication function, a new user authentication function, a user information storage function, a list of blogs storage function, a search bar function, and an API function for updating passwords. The user login authentication function is a function that retrieves the login information by intaking the information that are inputted in the user interface login page then the API function authenticates the information by comparing the information stored in MySQL database to make sure the username and password matches. The key is the username and the password is the value.

New User Authentication :

The new user authentication function is a function that will utilize the open authorization feature to enable Google API and/or Facebook API that will be used to authenticate a new user trying to register for an account. New users will be able to create an account by connecting to either their Google account or Facebook account by routing the page to Google or Facebook API and allowing them to login. Username and email are collected from their Google/Facebook account.

User Information Storage :

The user information storage function is a function that stores user information, such as name, username, password, and contact, after registering for an account into MySQL database. User will input information on the register for an account page, the function will store the information into variables and insert into MySQL database.

Blog Storage :

The list of blogs storage function is a function that stores each new blog that is posted into MySQL database. When a registered user creates a new blog post, a new page is created which essentially is a sublink to the user's page. The links to the blog post will be stored in a string array. There would be multiple hashmaps of string arrays that will store the blog links based on keywords.

Search Function :

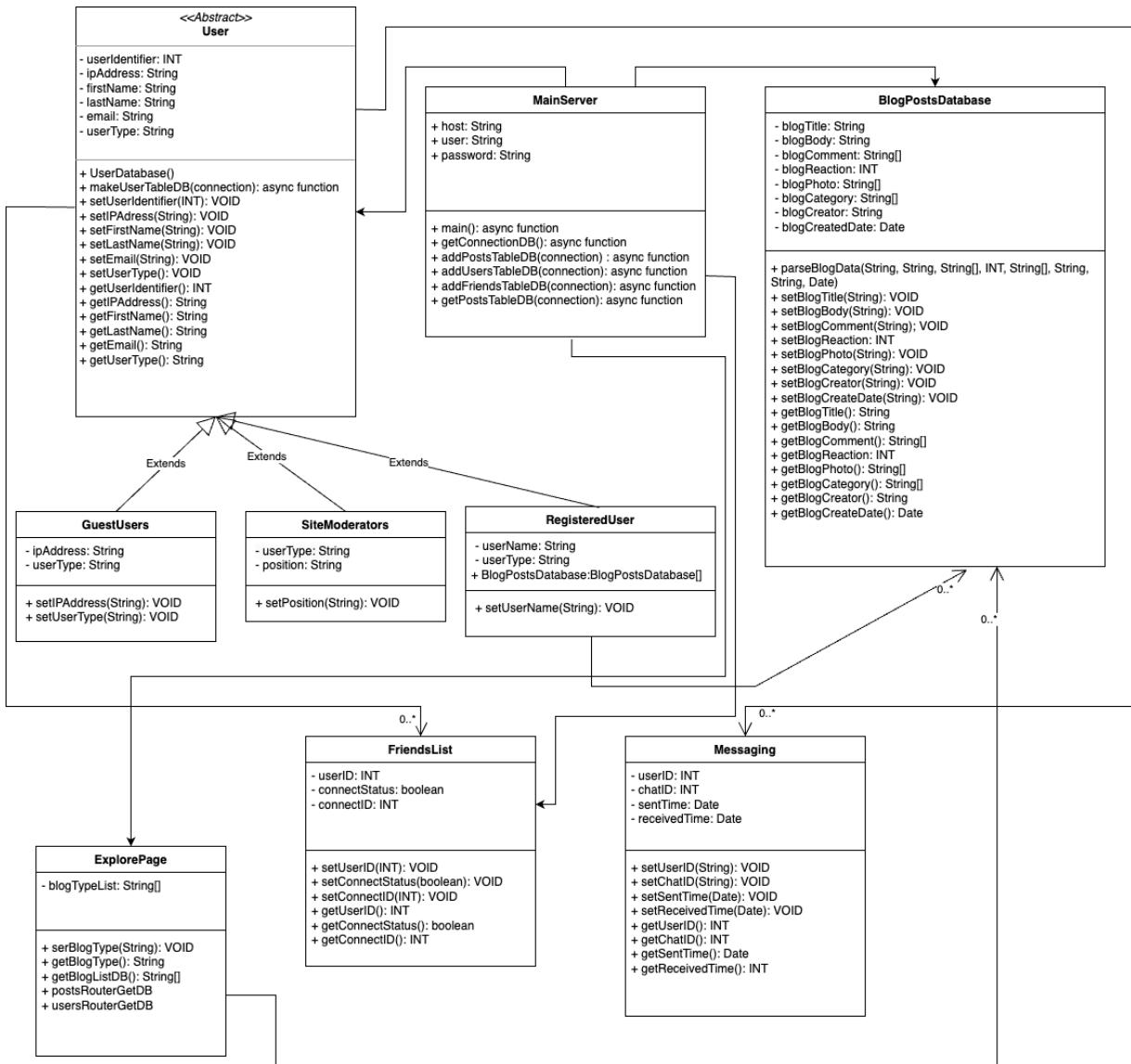
The search bar function is a function that populates a list of data from MySQL database based on the specific keyword input from the user in frontend. When a specific keyword is searched in the search bar, the function will use the key to search through MySQL database and return an array of links to blog posts relating to the keyword searched.

Updating Password :

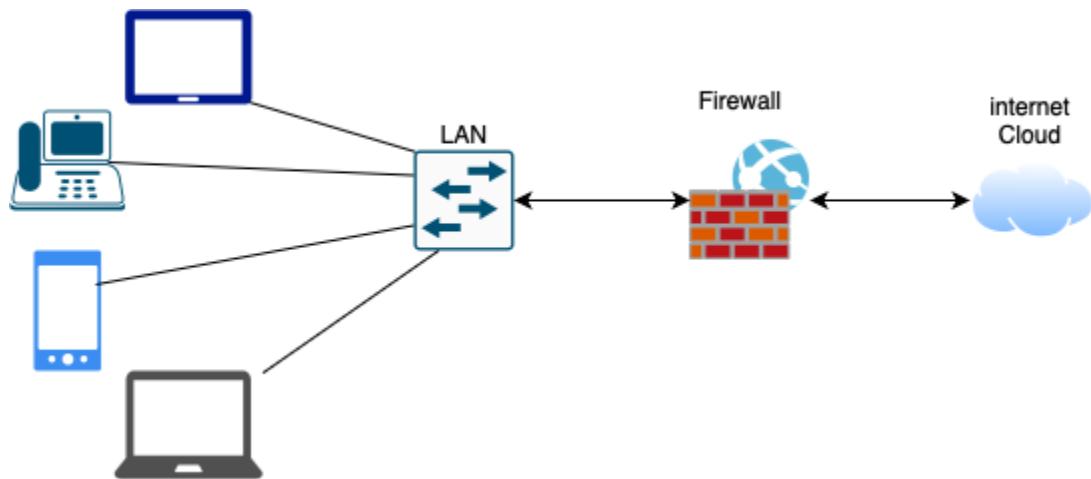
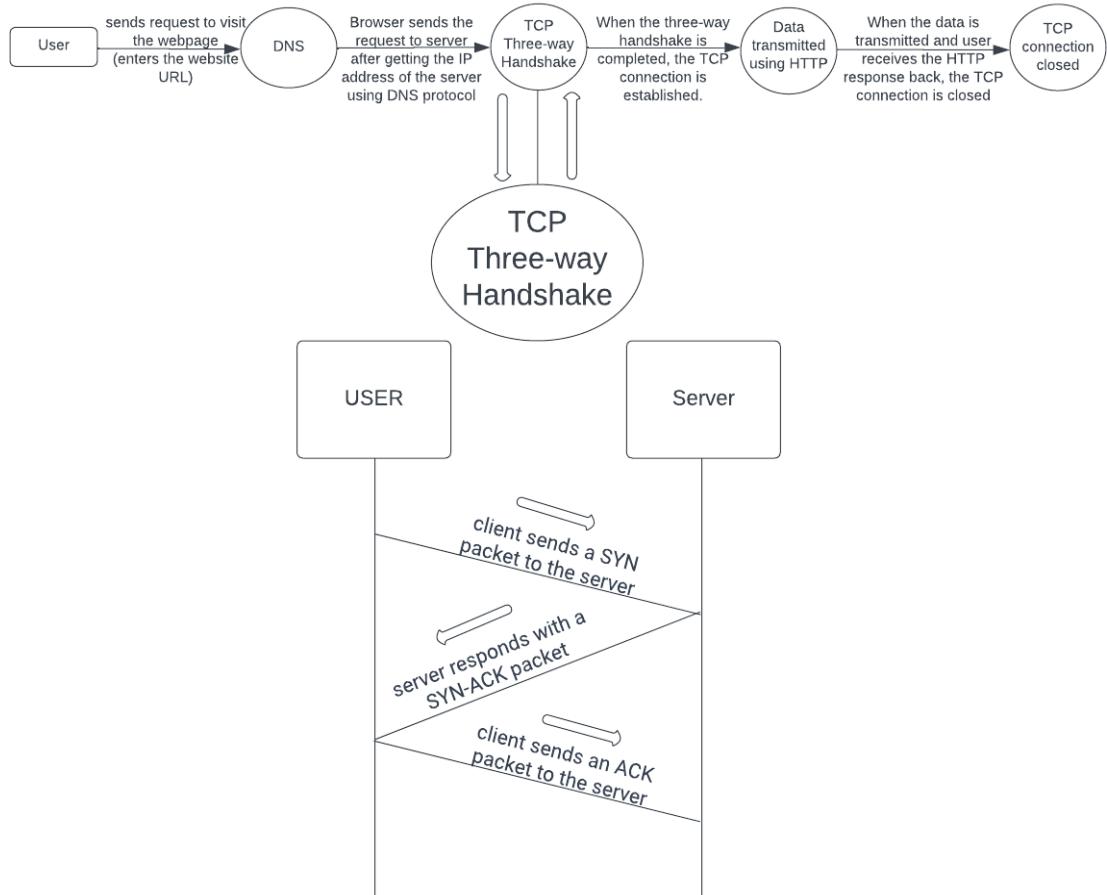
The API function for updating passwords is a function that will allow the registered user to change their password. When the user clicks on a button to change their password, it will direct them to a page where they need to enter their current password and the new

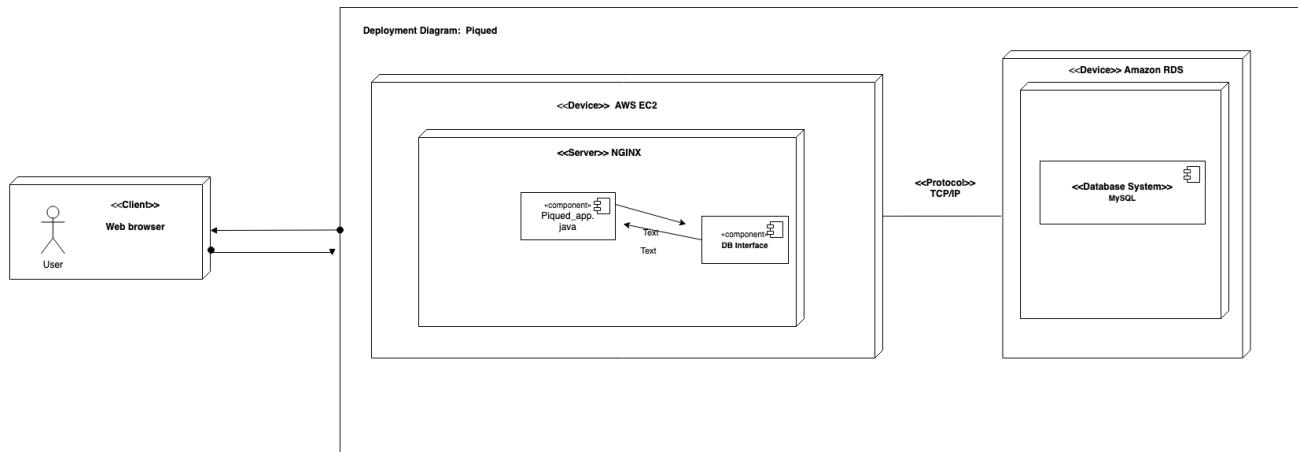
password twice in order to update it. The function will then check the current password in MySQL database by searching for the key, username, and comparing the value, password, to make sure they match. If the current password matches, it will then take the new password and store it into the database by replacing the current value, password, with the new value, new password.

6) High Level UML Diagrams



7) High Level Application Network and Deployment Diagrams





8) Identify actual key risks for your project at this time

- a) **Skill risks:** There is a risk that we may not have enough experience with high-level APIs, which could hinder our ability to develop certain features.

Solution: To address this, we plan to conduct extensive research and learn more about the relevant APIs. We will also consider reaching out to experts in the field for guidance and advice.

- b) **Schedule risks:** There is a risk that we may not be able to complete the project on time, given our commitments and available resources.

Solution: To mitigate this risk, we will prioritize tasks based on their importance and assign them to team members with the appropriate skills and experience. We will also regularly monitor progress and adjust our plan as needed to ensure that we stay on track.

- c) **Technical risks:** There is a risk of encountering technical issues related to our use of AWS, which could impact our ability to deploy and maintain the project.

Solution: We will conduct extensive testing and research to identify and address any technical issues that arise. We will also consider consulting with experts in the field for guidance and advice.

- d) **Teamwork risks:** There is a risk of issues related to collaboration, participation, task assignment, and meeting attendance, which could impact our ability to work effectively as a team.

Solution: We will establish clear communication channels and expectations, resolve conflicts quickly and proactively, and hold each other accountable for meeting commitments and attending meetings.

- e) **Legal/content risks:** There is a risk of encountering copyright issues related to the content and software we use in the project.

Solution: We will ensure that all content and software used in the project are properly licensed and free of any copyright infringement.

9) Project management

For Milestone 2, we used Trello as our primary tool for task management. Tasks were allocated based on team members' skills and represented by cards that could be moved between different lists on the board. Trello allowed us to track progress in real-time and identify bottlenecks or issues that needed addressing.

We also had frequent team meetings on zoom and in-person. Used Discord for communication and to discuss progress and challenges. For future milestones, we plan to continue using Trello, creating a new board for each milestone and adding lists for the different stages of the project.

Our approach to project management will be based on effective communication, clear task allocation, and regular progress tracking to ensure timely completion with high quality. We will continue to have frequent team meetings and use Discord as our primary communication channel, but explore other tools if needed for specific requirements.

10) Detailed list of contributions

Name	Role	Task Contribution	Scores
Jose Avila	Team Lead	Delegated task, scheduled meetings, set reminders for the team, gave input about the aesthetics and functionality of the project, answered questions and lent a hand when needed.	
Andy Shi	Frontend Lead	Write data definitions. Lead group effort to decide on color palette and typefaces. Created mockups and user flows storyboards on paper. Worked on frontend homepage code and created UI components. Collaborated with the backend lead, github master, and database master to code backend and connect database with backend and frontend for vertical SW prototype.	9
Leo Saeteurn	Backend Lead	Prioritized functional requirements, high level database architecture and organization, high level APIs and main algorithms, high level UML diagrams, high level application network diagram, collaboration on logo, collaboration on UI for color and font themes. Collaborated with the frontend lead, github master, and database master to code backend and connect database with backend and frontend for vertical SW prototype.	9
Nishit Pachchigar	Github Master	Collaborated with the backend lead, frontend lead, and database master on how to implement backend and connect with the frontend for vertical SW prototype. Worked on AWS and managed Trello. Managed the GitHub files.	7
Joshua Hayes	Database Master	Prioritized functional requirements, high level database architecture and organization, deployment diagram. Created database from diagram for backend virtual	8

		prototype. Collaborated with the backend lead, github master, and frontend lead to code backend and connect database with backend and frontend for vertical SW prototype.	
Gautami Kollolu Srinivasa	Document Editor	Identified actual key risks and solutions for the project. Wrote project management and data definitions. Collaboration on UI for color and font themes. Proofread the complete milestone 2 document to meet the requirement. Collaborated with the front end team for the vertical SW prototype.	6

Piqued

Team 5

Jose Avila, Team Lead, javila6@mail.sfsu.edu

Andy Shi, Frontend Lead, ashi2@mail.sfsu.edu

Leo Saeteurn, Backend Lead, lsaeteurn@mail.sfsu.edu

Nishit Pachchigar, Github Master, npachchigar@mail.sfsu.edu

Joshua Hayes, Database Master, jhayes10@mail.sfsu.edu

Gautami Gudla, Document Editor, gkollolusrinivasa@mail.sfsu.edu

Milestone 3

April 27th, 2023

History Table

Version	Submission Date
M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 19, 2023
M2V1	March 30, 2023
M1V2	April 16, 2023
M1V1	March 5, 2023

Table Of Contents

Cover Page	1
Table Of Contents	2
1. Data Definitions V3	3
2. Functional Requirements V3	5
# Priority 1 Requirements :	5
# Priority 2 Requirements :	8
# Priority 3 Requirements:	8
3. Wireframes Based on Your Storyboards/Mockups V2	10
4. High Level Database Architecture and Organization V2	11
5. High Level Diagrams V2	13
6. List of Contributions :	15

1. Data Definitions V3

- 1.1. Guest user: Individuals who do not have a registered account but can search and view public posts on the website. They can also view comments and reactions on public posts.
- 1.2. Registered user: Individuals who can log into their account, create, and publish posts on the website. They have the ability to edit and delete their own posts and comments, manage their account information, and view site statistics.
- 1.3. Site Moderators: Administrators who manage the frontend and backend of the website.
- 1.4. Blog Posts/Articles: Main content pieces on the website, written by users/authors. They contain text, images, and/or videos and can be categorized and tagged.
 - 1.4.1. Author: The user who originally posted the article.
 - 1.4.2. Published timestamp: A timestamp that represents when an article was originally posted.
 - 1.4.3. Updated timestamp: A timestamp that represents when an article was last updated by the author.
 - 1.4.4. Images: Visual content that can be used to enhance posts and improve engagement.
 - 1.4.5. Categories: Groupings of posts based on topic or theme
 - 1.4.6. Tags/Hashtag: Labels which can be added to a post to define its category and determine how it is found through search or recommended to other users.
 - 1.4.7. Comments: Rich text responses to posts made by users on the website.
 - 1.4.8. Reactions: Emotional responses on posts. This may include likes, emote responses, or any other form of instant feedback.
 - 1.4.8.1. Like: A positive response to a post by a registered user initiated by a simple button click.
 - 1.4.8.2. Reaction: Any other form of response by a registered user that can be either positive or negative.
 - 1.4.8.3. Reaction count: A number which indicates the amount of reactions to a post.
- 1.5. Recommendations: Algorithmically generated list of posts based on the user's interests.
- 1.6. Feed/For You Page: A landing page for logged-in users which shows a scrollable list of recommended posts.
- 1.7. Search: A search function that allows users to search blogs.
 - 1.7.1. Search filter: Function to filter out results based on certain criteria. This may be based on the aforementioned categories.

- 1.7.2. Search sort: Function to sort the results based on a certain order.
- 1.8. Editor: UI for editing a blog post. This can feature a rich text editor where users can apply formatting to text and insert images, video, and other types of media.
- 1.9. Profile Page: A customizable page which contains a user's public information and posts.
 - 1.9.1. Biography/Bio: A short, user-provided paragraph about the registered user.
- 1.10. Navigation bar/Header: A section where links are grouped together for easy navigation.
 - 1.10.1. Website logo: Display the company's logo on the website to represent the company
 - 1.10.2. Profile link: A link where the user can click on to go to their account page to edit their account.
 - 1.10.3. Home page link: A link where the users can navigate back to the main domain.
 - 1.10.4. Navigation links: Links to various primary pages or pages commonly accessed by users.
- 1.11. Menu: An section at the bottom of the page which contains informational links.
 - 1.11.1. Company's "about us" page: Information about each of the developers of the website.
 - 1.11.2. Company's description: Information about the website and its purpose.
 - 1.11.3. Privacy policy and Terms of Service: Links for legal purposes.

2. Functional Requirements V3

Priority 1 Requirements :

2.1. All Users

- 2.1.1. All users shall be able to view the website's information
- 2.1.2. All users shall be able to contact the site moderators with questions
- 2.1.3. All users shall be able to view the privacy policy
- 2.1.4. All users shall be able to view the terms and conditions

2.2. Guest Users

- 2.2.1. Guest users shall be able to create an account
- 2.2.2. Guest users shall be able to use a unique username to create an account
- 2.2.3. Guest users shall be able to only view public posts
- 2.2.4. Guest users shall be able to view the contents of public posts
- 2.2.5. Guest users shall be able to view a profile's public information
- 2.2.6. Guest users shall be able to search for public posts
- 2.2.7. Guest users shall be able to discover other public users
- 2.2.8. Guest users shall be able to view comments on public posts
- 2.2.9. Guest users shall be able to view reactions on public posts

2.3. Registered Users

- 2.3.1. Registered users shall be able to log in
- 2.3.2. Registered users shall be able to log out
- 2.3.3. Registered users shall be able to sign in with either their email or username.
- 2.3.4. Registered users shall be able to add a profile picture
- 2.3.5. Registered users shall be able to edit their profiles
 - 2.3.5.1. Registered users shall be able to change their profile picture
 - 2.3.5.2. Registered users shall be able to change their displayed username
 - 2.3.5.3. Registered users shall be able to change their email address
- 2.3.6. Registered users shall be able to search for other users by username
- 2.3.7. Registered users shall be able to create blog posts.
- 2.3.8. Registered users shall be able to add photos on blog posts.
- 2.3.9. Registered users shall be able to search for posts.
- 2.3.10. Registered users shall be able to delete their posts.
- 2.3.11. Registered users shall be able to add comments on post.
- 2.3.12. Registered users shall be able to delete their own comments.
- 2.3.13. Registered users shall be able to like/dislike to other people's comments

2.4. Administrator Users

- 2.4.1. Administrator users shall be able to create authorized users on account
- 2.5. Site moderators - creators(us)
 - 2.5.1. Site moderators shall be able to delete posts/articles
 - 2.5.2. Site moderators shall be able to ban accounts
 - 2.5.3. Site moderators shall be able to temporarily block accounts for review
 - 2.5.4. Site moderators shall be able to edit the layout of the platform
 - 2.5.5. Site moderators shall be able to add additional features
 - 2.5.6. Site moderators shall be able to answer users' questions and concerns
 - 2.5.7. Site moderators shall be able to create an account
 - 2.5.8. Site moderators shall be able to post on the platform's main page
 - 2.5.9. Site moderators shall be able to make site announcements
 - 2.5.10. Site moderators shall be able to update website
 - 2.5.11. Site moderators shall be able to temporarily suspend the website for maintenance
 - 2.5.12. Site moderators shall be able to reinstate accounts
 - 2.5.13. Site moderators shall be able to review accounts
- 2.6. User's Profile Page
 - 2.6.1. User's profile page shall have the user's picture
 - 2.6.2. User's profile page shall have the user's username
 - 2.6.3. User's profile page shall have published public posts
 - 2.6.4. User's profile page shall have a biography
 - 2.6.5. User's profile page shall have contact information that are made public by user
- 2.7. All pages
 - 2.7.1. All pages shall have a navigation bar/header
 - 2.7.2. All pages shall have a menu with links to informational pages
 - 2.7.3. All pages shall have our company's logo
- 2.8. User's Logged-In Interface
 - 2.8.1. User's logged-in interface shall have a link to the user's account management page in the navigation bar
- 2.9. Blog Posts
 - 2.9.1. Blog posts shall contain an original author
 - 2.9.2. Blog posts shall contain a date and time of creation
 - 2.9.3. Blog posts shall contain a title
 - 2.9.4. Blog posts shall contain a photo(s)
 - 2.9.5. Blog posts shall contain a category(ies)
 - 2.9.6. Blog posts shall contain a tag(s)/hashtag(s)
 - 2.9.7. Blog posts shall contain a comment sections with comments from other users
- 2.10. Exploration Newsfeed

- 2.10.1. Exploration newsfeed shall contain random public posts
 - 2.10.2. Exploration newsfeed shall have a search bar
 - 2.10.3. Exploration newsfeed shall have blog posts that can be clicked on to view full posts
- 2.11. Search Bar
- 2.11.1. Search bar shall have a text box to search up public blog posts by categories, tags, and users
- 2.12. Navigation Bar
- 2.12.1. Navigation bar shall have a link to the user's profile when logged in
 - 2.12.2. Navigation bar shall have a link to the home page
 - 2.12.3. Navigation bar shall have a link to the user's account management page when logged in
 - 2.12.4. Navigation bar shall have a post button
 - 2.12.5. Navigation bar shall have a log out button when logged in
- 2.13. Menu (Footer)
- 2.13.1. Menu shall have a link to the company's information page
 - 2.13.2. Menu shall have a link to contact the company by emailing
 - 2.13.3. Menu shall have the company's contact information
 - 2.13.4. Menu shall have a link to "about us" page
 - 2.13.5. Menu shall have a copyright logo
 - 2.13.6. Menu shall have a link to the terms and conditions
 - 2.13.7. Menu shall have a link to the privacy policy
- 2.14. Chat Box
- 2.14.1. Chat box shall contain user's username for both sender and recipient(s)

Priority 2 Requirements :

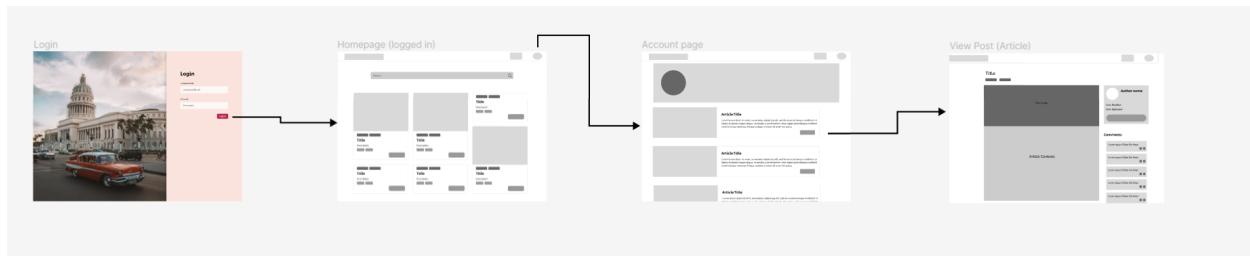
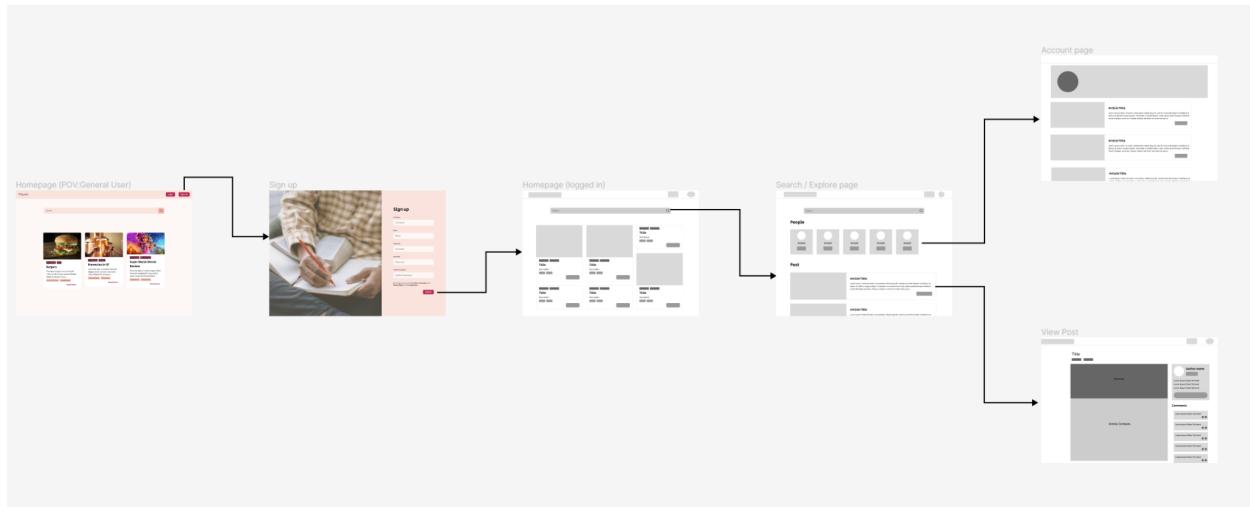
- 2.15. Registered Users
- 2.15.1. Registered users shall be able to customize profile using a standard format template
 - 2.15.2. Registered users shall be able to post videos on their blog post.
 - 2.15.3. Registered users shall be able to save their profile photos to make it easier for them to change back and forth
 - 2.15.4. Registered users shall be able to report other spam accounts to site moderators via a report link
 - 2.15.5. Registered users shall be able to react to posts
 - 2.15.6. Registered users shall be able to send private messages
 - 2.15.7. Registered users shall be able to receive private messages
 - 2.15.8. Registered users shall be able to add contact information

- 2.15.9. Registered users shall be able to make contact information private or public.
- 2.15.10. Registered users shall be able to update their posts.
- 2.16. User's Logged-In Interface
 - 2.16.1. User's logged-in interface shall have a notification icon that shows if there's any notification
- 2.17. Blog Posts
 - 2.17.1. Blog posts shall contain videos
 - 2.17.2. Blog posts shall contain a reaction(s) from other users
 - 2.17.3. Blog posts shall be able to be reposted by other users
 - 2.17.4. Blog posts shall contain a date and time of modification to posts
- 2.18. Chat Box
 - 2.18.1. Chat box shall contain group chat with more than two users
 - 2.18.2. Chat box shall contain messages from all users within chat
 - 2.18.3. Chat box shall contain date and time of messages sent
- 2.19. Search Bar
 - 2.19.1. Search bar shall have a filter function
 - 2.19.2. Search bar shall have a sorting option

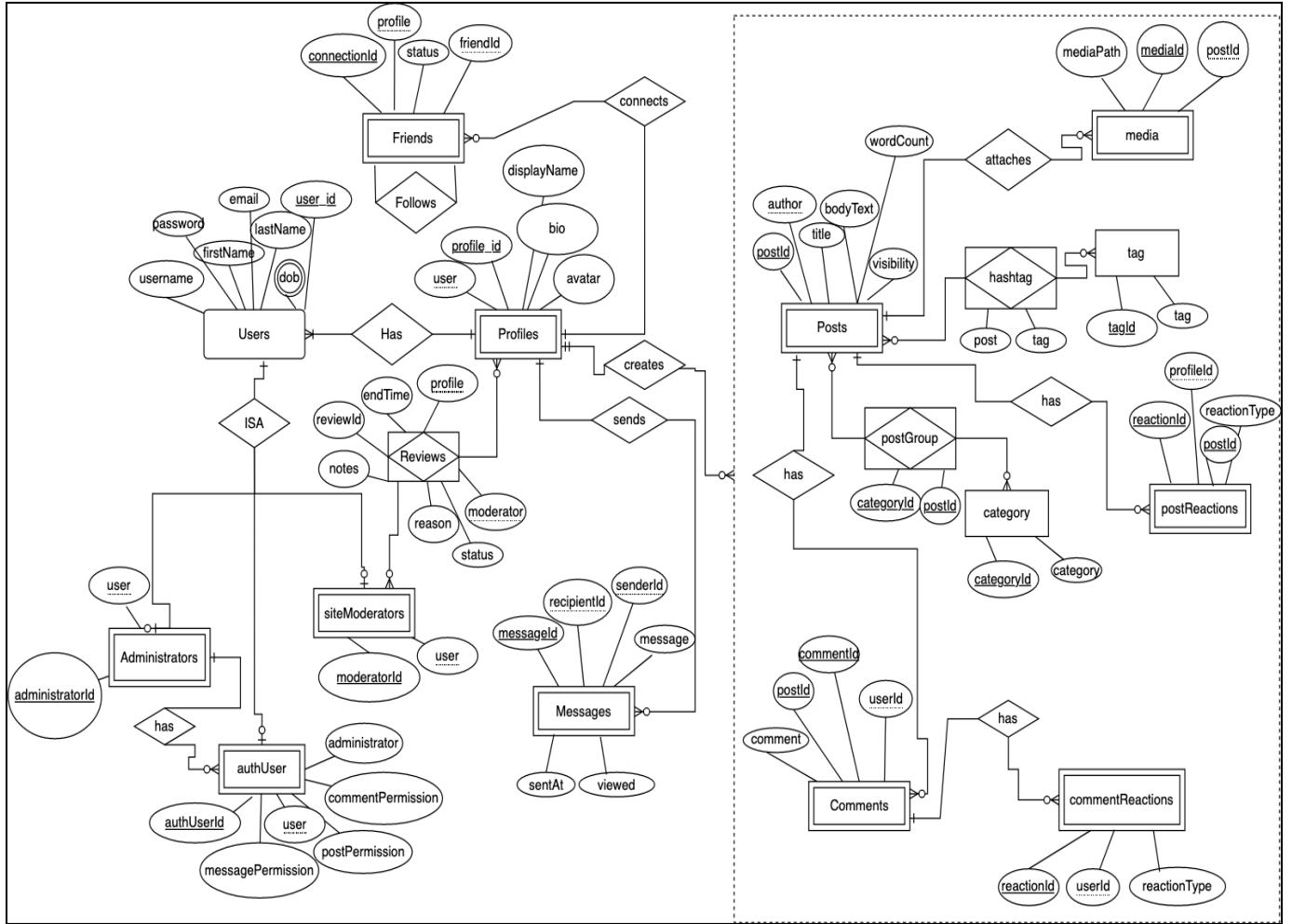
Priority 3 Requirements :

- 2.20. Guest users
 - 2.20.1. Guest users shall be able to send a private message to a user
- 2.21. Registered Users
 - 2.21.1. Registered users shall be able to disable comments for their posts. [P3]
- 2.22. User's Profile Page
 - 2.22.1. User's profile page shall have their online status
 - 2.22.2. User's profile page shall have a standard interface template that users can use to create a profile
- 2.23. Blog Posts
 - 2.23.1. Blog posts shall have a standard interface template that users can use to create a blog post
- 2.24. All Pages
 - 2.24.1. All pages shall have a advertisement banner
- 2.25. Menu
 - 2.25.1. Menu shall have social media links
- 2.26. Chat box
 - 2.26.1. Chat box shall contain emojis
 - 2.26.2. Chat box shall contain reaction

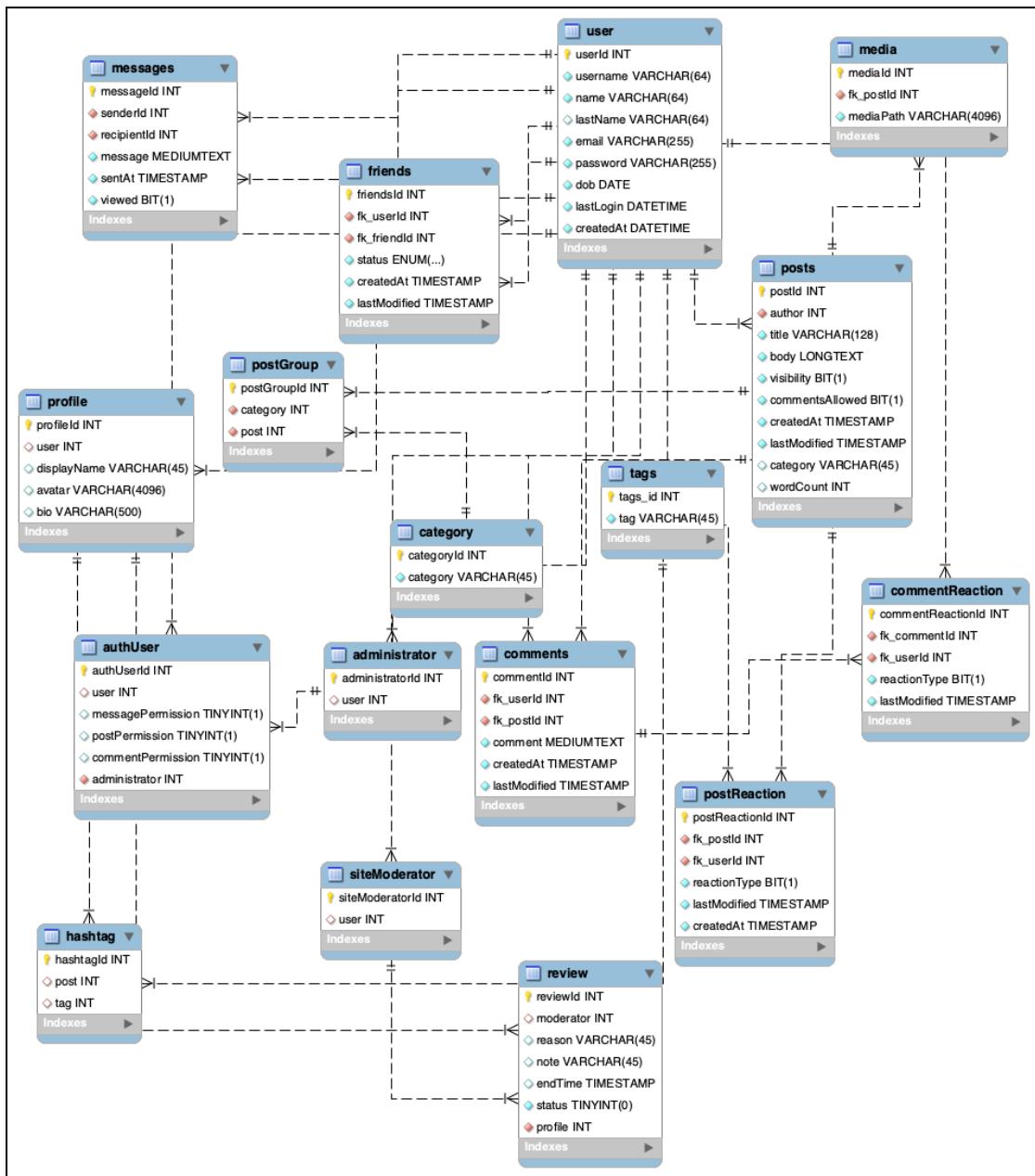
3. Wireframes Based on Your Storyboards/Mockups V2



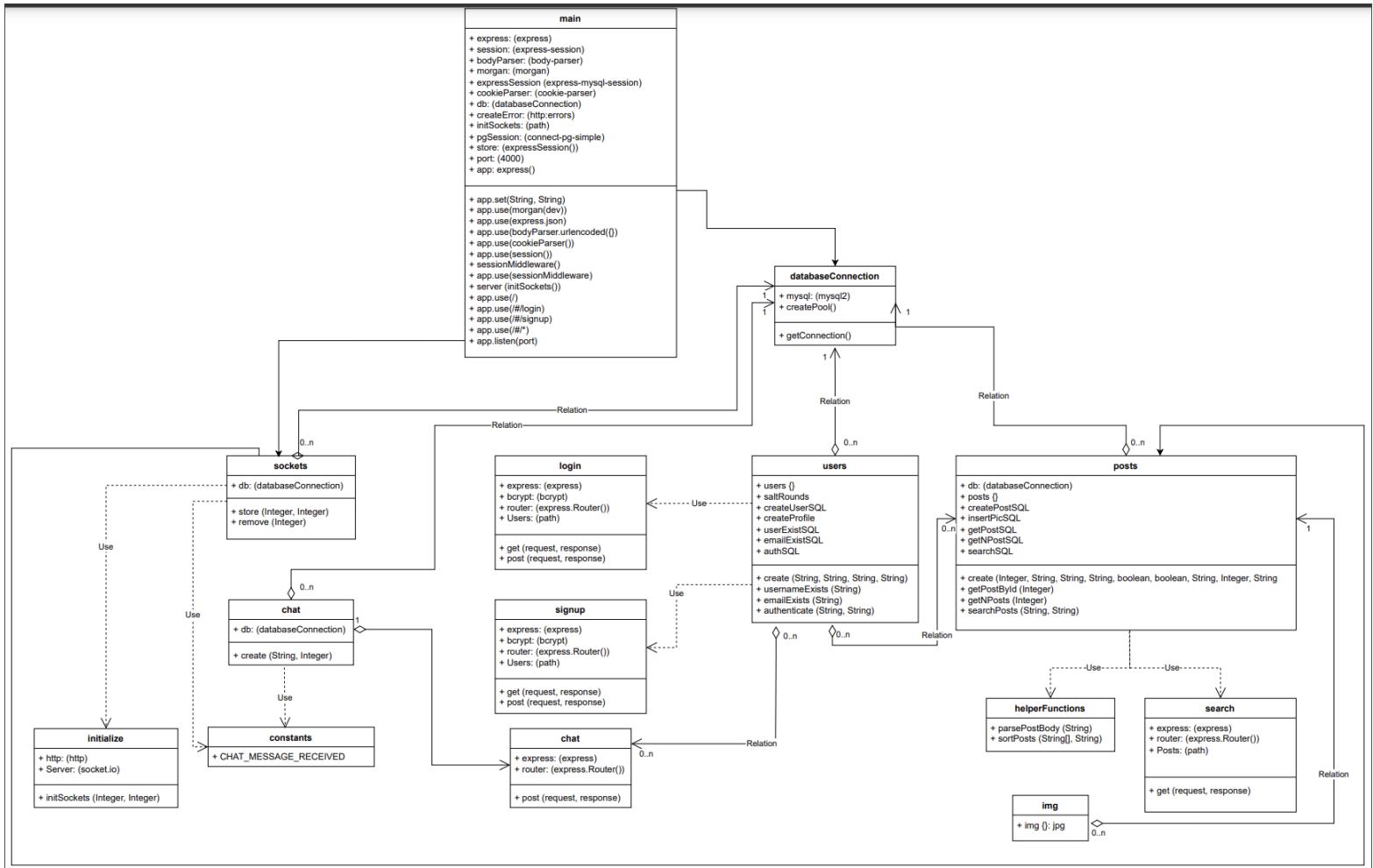
4. High Level Database Architecture and Organization V2



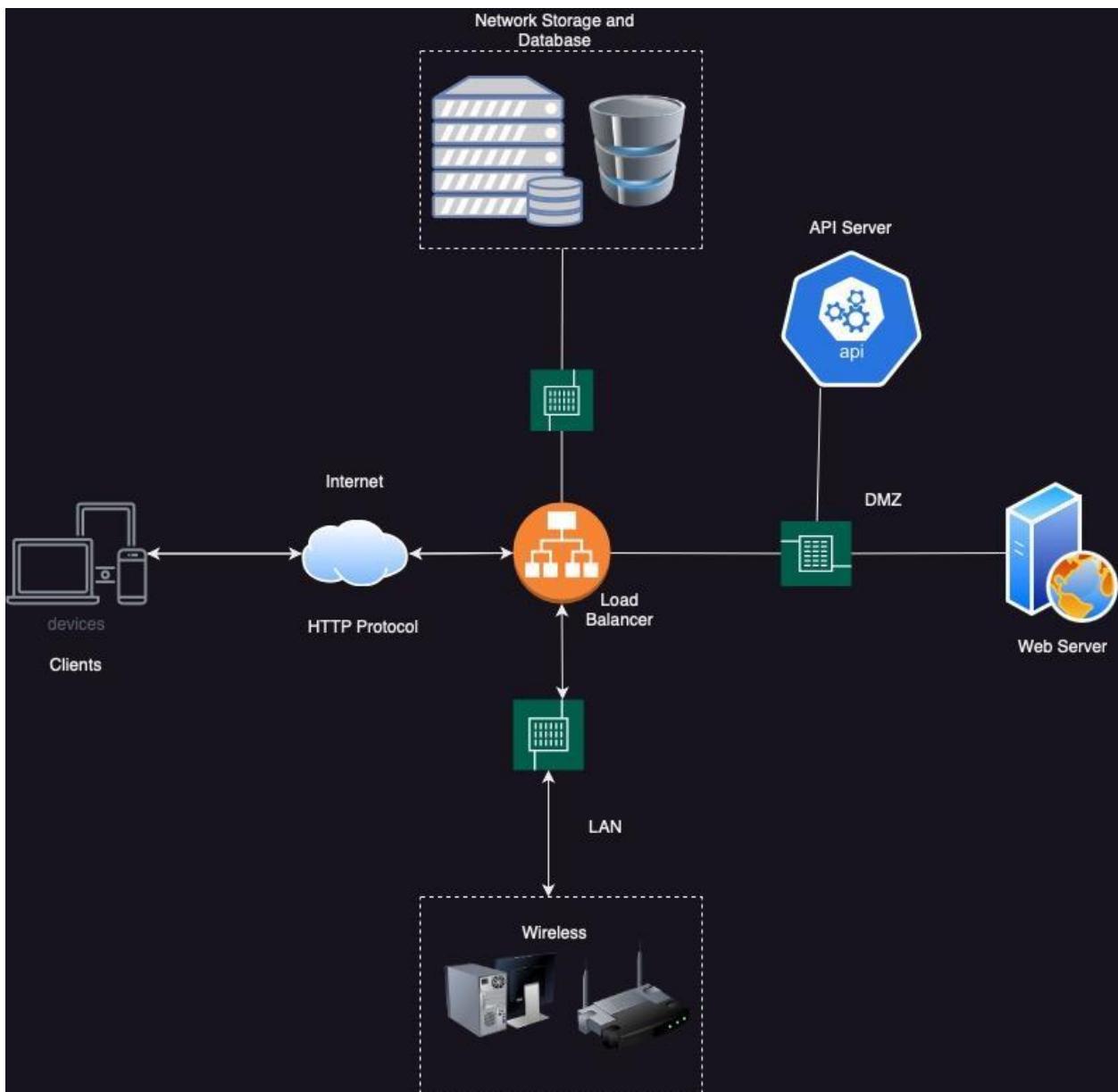
Organization V2



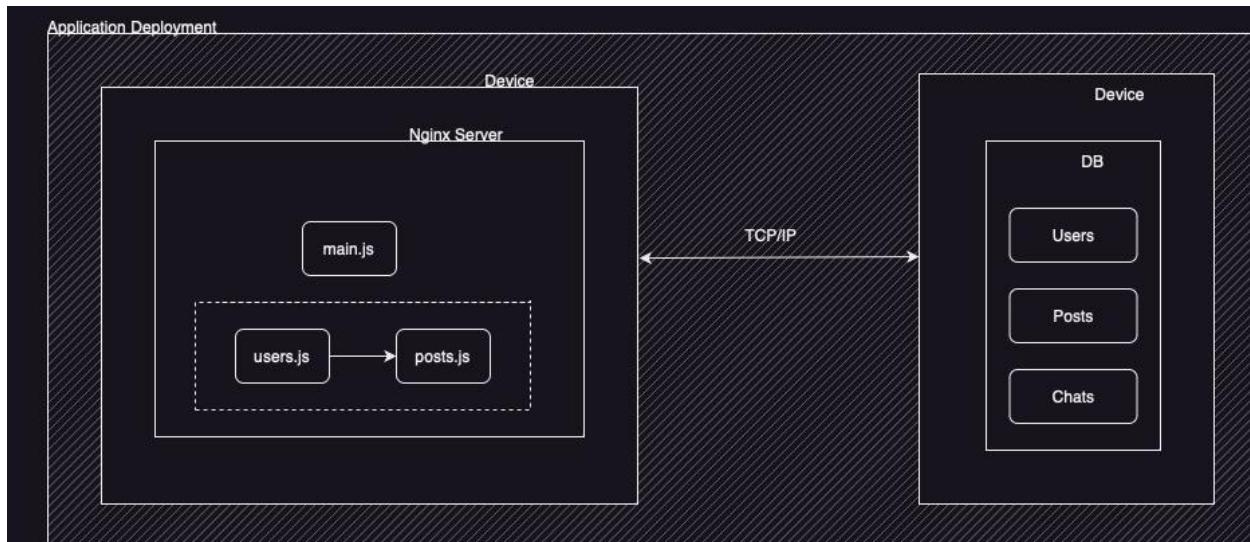
5. High Level Diagrams V2



Network Diagram :



Deployment Diagram :



6. List of Contributions

Role	Name	Contributions	Score
Team Lead	Jose Avila	Delegated task, scheduled meetings, set reminders for the team, gave input about U/I, created wireframes using figma, gave back feedback from CTO, motivated the team, and resolved internal issues.	
Front End Lead	Andy Shi	Coded the frontend U/I, set up routes, schedule meeting with front end members, created wireframes and components, wrote data definitions and collaborated with the back end.	9
Back End Lead	Leo Saeteurn	Schedule meetings with back end members, coded back end, created the uml diagram and wrote/revised functional requirements.	8
Database Master	Joshua Hayes	Created all databases , created the high level database diagram, wrote/revised functional requirements, set up routes, collected data and coded back end.	9
GitHub Master	Nishit Pachchigar	Created High Level Network and Deployment diagrams, worked with the back end routing systems together, pushing the work to the github, deploying website to AWS instance	8
Document Editor	Gautami Gudla	Edited & updated the document,wrote data definitions, worked on the frontend, created components, revised P1 requirements,gave us reminders, and proofread the document.	6

Piqued

Team 5

Jose Avila, Team Lead, javila6@mail.sfsu.edu

Andy Shi, Frontend Lead, ashi2@mail.sfsu.edu

Leo Saeteurn, Backend Lead, lisaeteurn@mail.sfsu.edu

Nishit Pachchigar, Github Master, npachchigar@mail.sfsu.edu

Joshua Hayes, Database Master, jhayes10@mail.sfsu.edu

Gautami Gudla, Document Editor, gkollolusrinivasa@mail.sfsu.edu

Milestone 4

May 18th, 2023

History Table

Version	Submission Date
M4V2	May 25, 2023
M4V1	May 18, 2023
M3V2	May 18, 2023
M3V1	April 27, 2023
M2V2	April 19, 2023
M2V1	March 30, 2023
M1V2	April 16, 2023
M1V1	March 5, 2023

Table Of Contents

1. Product Summary	3
1.1. Name Of The product:	3
1.2. Final P1 Functions:	3
1.3. What is Unique In Our Product:	6
1.4. URL Of the Product:	6
2. Usability Test Plan	7
3. QA Test Plan	8
3.1. Performance:	8
3.2. Security:	8
3.3. Legal:	8
3.4. Compatibility:	8
3.5. Usability:	8
3.6. Marketing:	8
4. Code Review	10
4.1. Team 5 Code review by Saru of Team 4 - reviewed users.js file	10
4.2. Code Review By Nishit Pachchigar and Gautami Gudla - Reviewed Signup.svelte	10
5. Self-Check On Best Practices For Security :	14
5.1. List major assets you are protecting	14
5.2. Confirm that you encrypt PW in the DB :	14
5.3. Confirm Input data validation :	14
6. Self-check: Adherence To Original Non-Functional Specs	15
6.1. Performance:	15
6.2. Security:	15
6.3. Legal:	15
6.4. Compatibility:	15
6.5. Usability:	15
6.6. Marketing:	15
7. List Of Contributions	16

1. Product Summary

1.1. Name Of The product:

Piqued

1.2. Final P1 Functions:

1.1.1. All Users

- 1.1.1.1. All users shall be able to view the website's information
- 1.1.1.2. All users shall be able to contact the site moderators with questions
- 1.1.1.3. All users shall be able to view the privacy policy
- 1.1.1.4. All users shall be able to view the terms and conditions

1.1.2. Guest Users

- 1.1.2.1. Guest users shall be able to create an account
- 1.1.2.2. Guest users shall be able to use a unique username to create an account
- 1.1.2.3. Guest users shall be able to only view public posts
- 1.1.2.4. Guest users shall be able to view the contents of public posts
- 1.1.2.5. Guest users shall be able to view a profile's public information
- 1.1.2.6. Guest users shall be able to search for public posts
- 1.1.2.7. Guest users shall be able to discover other public users
- 1.1.2.8. Guest users shall be able to view comments on public posts
- 1.1.2.9. Guest users shall be able to view reactions on public posts

1.1.3. Registered Users

- 1.1.3.1. Registered users shall be able to log in
- 1.1.3.2. Registered users shall be able to log out
- 1.1.3.3. Registered users shall be able to sign in with either their email or username.
- 1.1.3.4. Registered users shall be able to add a profile picture
- 1.1.3.5. Registered users shall be able to edit their profiles
 - 1.1.3.5.1. Registered users shall be able to change their profile picture
 - 1.1.3.5.2. Registered users shall be able to change their displayed username
 - 1.1.3.5.3. Registered users shall be able to change their email address

- 1.1.3.6. Registered users shall be able to search for other users by username
- 1.1.3.7. Registered users shall be able to create blog posts.
- 1.1.3.8. Registered users shall be able to add photos on blog posts.
- 1.1.3.9. Registered users shall be able to search for posts.
- 1.1.3.10. Registered users shall be able to delete their posts.
- 1.1.3.11. Registered users shall be able to add comments on post.
- 1.1.3.12. Registered users shall be able to delete their own comments.
- 1.1.3.13. Registered users shall be able to like/dislike to other people's comments

1.1.4. Administrator Users

- 1.1.4.1. Administrator users shall be able to create authorized users on account

1.1.5. Site moderators - creators(us)

- 1.1.5.1. Site moderators shall be able to delete posts/articles
- 1.1.5.2. Site moderators shall be able to ban accounts
- 1.1.5.3. Site moderators shall be able to temporarily block accounts for review
- 1.1.5.4. Site moderators shall be able to edit the layout of the platform
- 1.1.5.5. Site moderators shall be able to add additional features
- 1.1.5.6. Site moderators shall be able to answer users' questions and concerns
- 1.1.5.7. Site moderators shall be able to create an account
- 1.1.5.8. Site moderators shall be able to post on the platform's main page
- 1.1.5.9. Site moderators shall be able to make site announcements
- 1.1.5.10. Site moderators shall be able to update website
- 1.1.5.11. Site moderators shall be able to temporarily suspend the website for maintenance
- 1.1.5.12. Site moderators shall be able to reinstate accounts
- 1.1.5.13. Site moderators shall be able to review accounts

1.1.6. User's Profile Page

- 1.1.6.1. User's profile page shall have the user's picture
- 1.1.6.2. User's profile page shall have the user's username
- 1.1.6.3. User's profile page shall have published public posts
- 1.1.6.4. User's profile page shall have a biography
- 1.1.6.5. User's profile page shall have contact information that are made public by user

1.1.7. All pages

- 1.1.7.1. All pages shall have a navigation bar/header
- 1.1.7.2. All pages shall have a menu with links to informational pages

1.1.7.3. All pages shall have our company's logo

1.1.8. User's Logged-In Interface

1.1.8.1. User's logged-in interface shall have a link to the user's account management page in the navigation bar

1.1.9. Blog Posts

1.1.9.1. Blog posts shall contain an original author

1.1.9.2. Blog posts shall contain a date and time of creation

1.1.9.3. Blog posts shall contain a title

1.1.9.4. Blog posts shall contain a photo(s)

1.1.9.5. Blog posts shall contain a category(ies)

1.1.9.6. Blog posts shall contain a tag(s)/hashtag(s)

1.1.9.7. Blog posts shall contain a comment sections with comments from other users

1.1.10. Exploration Newsfeed

1.1.10.1. Exploration newsfeed shall contain random public posts

1.1.10.2. Exploration newsfeed shall have a search bar

1.1.10.3. Exploration newsfeed shall have blog posts that can be clicked on to view full posts

1.1.11. Search Bar

1.1.11.1. Search bar shall have a text box to search up public blog posts by categories, tags, and users

1.1.12. Navigation Bar

1.1.12.1. Navigation bar shall have a link to the user's profile when logged in

1.1.12.2. Navigation bar shall have a link to the home page

1.1.12.3. Navigation bar shall have a link to the user's account management page when logged in

1.1.12.4. Navigation bar shall have a post button

1.1.12.5. Navigation bar shall have a log out button when logged in

1.1.13. Menu (Footer)

1.1.13.1. Menu shall have a link to the company's information page

1.1.13.2. Menu shall have a link to contact the company by emailing

1.1.13.3. Menu shall have the company's contact information

1.1.13.4. Menu shall have a link to "about us" page

1.1.13.5. Menu shall have a copyright logo

1.1.13.6. Menu shall have a link to the terms and conditions

1.1.13.7. Menu shall have a link to the privacy policy

1.1.14. Chat Box

1.1.14.1. Chat box shall contain user's username for both sender and recipient(s)

1.3. What is Unique In Our Product:

Piqued is a unique blogging website that offers a safe and social platform for individuals to express themselves while sharing common interests. Unlike other blogging sites, Piqued allows users to interact with their friends and other like-minded individuals, creating a sense of community around shared interests. The platform is perfect for both extroverted and introverted individuals, as it offers a safe space to share without fear of scrutiny. Piqued also allows users to find new interests and experiences, making it a great platform for exploration and self-discovery. The ability to connect and build a community of like-minded individuals sets Piqued apart from other blogging sites and makes it an excellent choice for those looking to express themselves and connect with others.

What makes Piqued truly special is the ability to connect with individuals who share the same interests. Through engaging and detailed blog posts, you can delve into your favorite topics such as travel experiences, culinary adventures, life hacks, and more. Discover new ideas, gain inspiration, and add exciting activities to your bucket list, all while building a vibrant community of like-minded individuals.

1.4. URL Of the Product:

<http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000>

2. Usability Test Plan

Functions	Task	Effectiveness	Efficiency	Satisfaction	Errors	Comments
Create	Create a post	90%	95%	95%	No errors	Intuitively accessible and effortlessly discoverable.
Search	Search a post	40%	75%	50%	Nothing popped up when search	The search function provided a user-friendly interface, allowing seamless input, although it failed to generate any relevant results.
Write Comments/ Leave a reaction	Leaving feedback	80%	70%	85%	No errors	Although the comment section was initially challenging to locate, it proved to be user-friendly and intuitive once discovered.
Editing profile	Updating user credentials	95%	95%	90%	Users are able to use invalid data or avatar	Unable to update user avatar
Chat	Chat with chatbox	90%	90%	70%	Messages aren't loaded in real time	Site must be reloaded

Create Function:

Test objectives:

The objective of this test is to ensure that the users can effortlessly locate the "Create a Post" button and seamlessly navigate the process of creating a post without any confusion.

Test description:

- System setup: The user is registered in db and wants to create a post. They are able to click links and create a post through the UI.

- Environment setup : Mobile device, laptop, or desktop.
- Starting point: Home page
- Intended Users: Users who are looking to create a new article

URL: <http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/#/post>

Search Function:

Test objectives:

The objective of this test function is to test the user experience when searching for a variety of items such as searching for another user, category, post, or hashtag. This is being tested to validate that the correct results are being sent as a response to the user. Users can send search requests for various data on the site. The user may send invalid data or data which does not exist on the site's database. In this case, it would be expected for the function to return some useful suggestions and also notify the user that their search did not yield any exact matches. In the case that a user searches for an entity that exists in the database, all relevant results should be returned.

Test description:

- System setup: Entities exist within the database that can potentially match their search. When a search is entered, relevant results should be returned.
- Environment setup :User using their mobile device, laptop, or desktop.
- Starting point: Home page
- Intended Users: Users searching for a topic, keyword, or other user

URL: <http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/>

Writing comments/reaction Function:

Test objectives:

The objective of this test function is to test the site's usability for when a user would like to leave feedback for a post. Feedback can be done in different ways such as leaving a comment or leaving a reaction. This function is being tested because it is a key feature for the social functionality of the site. This is being tested also to ensure that users can only leave one form of a reaction but also leave many comments. Input validation should also be tested to verify that comments of varying length from invalid input all the way to full responses.

Test description:

- System setup: A user is registered within the db and posts exist. The user is looking to leave a comment on a post. They are able to click a post and leave a reaction or comment.
- Environment setup :User using their mobile device, laptop, or desktop.
- Starting point: Home page
- Intended Users: Users that want to leave feedback for an article

URL: <http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/#/article>

Editing Profile Function:

Test objectives:

The objective of this function is to measure the usability of the site when a user would like to edit their profile. This is being tested to measure how many steps and how long it takes a registered user to reach their account details page and edit their personal information. This function is also being tested to make sure that 1) The user is shown the correct personal information when accessing their profile. 2) This is also another area where input validation must be checked. 3) Testing to ensure that non-functional security requirements such as ensuring that password validation are still upheld and are updated on the backend.

Test description:

- System setup: The user is registered and is looking to edit their personal information. The link to edit account details exist and data should be modifiable.
- Environment setup : User using mobile device, laptop, or desktop.
- Starting point: Home page
- Intended Users: Registered users looking to update their user information

URL: <http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/#/account>

Chat Function:

Test objectives:

The objective of this test function is to test the user experience when chatting with other registered users. The function is being tested to make sure that messages are sent in real time and are valid input.

Test description:

- System setup: Users exist in the db and are looking to communicate with other users. Chat and send button works as well as sending/receiving messages in realtime
- Environment setup :User using their mobile device, laptop, or desktop.
- Starting point: Home page
- Intended Users: Registered users that want to chat with other registered users.

URL: <http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/>

3. QA Test Plan

3.1. **Test objectives:** The objective of the quality assurance test is to verify that the functionality of the software builds, executes, and runs according to the specifications of the non-functional requirements listed. The test should verify and identify any bugs or defects so that the developers of the software can debug and fix the issues.

3.2. HW and SW setup:

3.2.1. The testing requires either a laptop, desktop, or mobile device with internet access and a website browser.

Website link on AWS server:

<http://ec2-18-224-63-127.us-east-2.compute.amazonaws.com:4000/>

3.3. Feature to be tested are non-functional requirements:

3.3.1. QA Test Plan 1: Password Encryption

3.3.1.1. **Case 1:** Enter your information on the signup page. Check that the database does not show the password you entered

3.3.1.2. **Case 2:** Enter your username and password on the login page. Check that the database does not show the password you entered.

3.3.1.3. **Case 3:** Enter new information with a new password. Check that the database does not show the password you entered.

3.3.2. Test input:

3.3.2.1. **Case 1:** Unique password that contains alphanumeric characters with at least one symbol and minimum length of eight characters.

3.3.2.2. **Case 2:** New unique password that contains alphanumeric characters with at least one symbol and minimum length of eight characters.

3.3.2.3. **Case 3:** New unique password that is not the same as case 1 and 2.

3.3.3. Expected test results:

3.3.3.1. **Case 1:** pass

3.3.3.2. **Case 2:** pass

3.3.3.3. **Case 3:** pass

3.3.4. QA Test Plan 2: Unique Username

3.3.4.1. **Case 1:** Enter your information on the signup page. Check that the database does not contain the same username as the one you created.

3.3.4.2. **Case 2:** Enter your information on the signup page. Check that there are no error messages mentioning a duplicate username.

3.3.4.3. **Case 3:** Enter username and password in the login page. Check to make sure there is no error message that mentions a duplicate username.

3.3.5. **Test input:**

3.3.5.1. **Case 1:** Unique username

3.3.5.2. **Case 2:** Unique username

3.3.5.3. **Case 3:** Unique username

3.3.6. **Expected test results:**

3.3.6.1. **Case 1:** pass, no error

3.3.6.2. **Case 2:** pass, no error

3.3.6.3. **Case 3:** pass, no error

3.3.7. **QA Test Plan 3:** Cookie for logging in

3.3.7.1. **Case 1:** Enter your username and password in the login page. Go into the home page. Check to make sure you are still logged in by checking for your username at the top right of the page.

3.3.7.2. **Case 2:** Enter your username and password in the login page. Go into the home page then to the search page. Check to make sure you are still logged in by checking for your username at the top right of the page.

3.3.7.3. **Case 3:** Enter your username and password in the login page. Go into the search page then to the account page. Check to make sure you are still logged in by checking for your username at the top right of the page.

3.3.8. **Test input:**

3.3.8.1. **Case 1:** Unique username and password

3.3.8.2. **Case 2:** Unique username and password

3.3.8.3. **Case 3:** Unique username and password

3.3.9. **Expected test results:**

3.3.9.1. **Case 1:** pass

3.3.9.2. **Case 2:** pass

3.3.9.3. **Case 3:** pass

3.3.10. **QA Test Plan 4:** UI Responsiveness

3.3.10.1. **Case 1:** Open up your preferred browser and login into the website with your username and password. Minimize the page, resize it by hovering over the edge of the browser and adjusting the size in and out. Check to make sure that there are no glitches with the display.

3.3.10.2. **Case 2:** Go to the website without logging in. Go into the search page and adjust the size of the website window. Check to make sure that there are no glitches with the display.

3.3.10.3. **Case 3:** Go to the home page without logging in. Adjust the window size of the browser. Check to make sure the display is not distorted as you adjust the window.

3.3.11. **Test input:**

3.3.11.1. **Case 1:** Username and password

3.3.11.2. **Case 2:** No input, adjust browser window.

3.3.11.3. **Case 3:** No input, adjust browser window.

3.3.12. **Expected test results:**

3.3.12.1. **Case 1:** pass, no error

3.3.12.2. **Case 2:** pass, no error

3.3.12.3. **Case 3:** pass, no error

3.3.13. **QA Test Plan 5:** Website navigation through each page

3.3.13.1. **Case 1:** Log into your account. Click on the account page by clicking on the menu bar then the account management option. Check to make sure you were directed to the account management page.

3.3.13.2. **Case 2:** Create an account in the signup page by entering the required information. Search a post by entering a keyword in the search bar, such as food. Check to make sure the page lists blog posts based on the word you search.

3.3.13.3. **Case 3:** Log into your account in the login page with your username and password. Click on one of the blog posts that shows up in the home page. Check to make sure you were directed to the full blog post.

3.3.14. **Test input:**

3.3.14.1. **Case 1:** Username and password

3.3.14.2. **Case 2:** Full name, username, email, and password

3.3.14.3. **Case 3:** Username and password

3.3.15. **Expected test results:**

3.3.15.1. **Case 1:** pass, no error

3.3.15.2. **Case 2:** pass, no error

3.3.15.3. **Case 3:** pass, no error

Test #	Test Title	Test Description	Test Input	Expected correct output	Google Chrome Test Results	Safari Test Results
1	Password encryption	Ensure that inputted user passwords are encrypted when stored in the database.	Fullscreen, username, email, password	Passwords are encrypted with a hash which produces random characters which get stored into the database.	PASS	PASS
2	Unique username	Ensure that the username input in the frontend displays an error if a username is already taken	Fullscreen, username, email, password	There are no duplicate usernames shown in the database.	PASS	PASS
3	Cookie for logging in	Make sure that session cookies are stored and persist across browser sessions.	Fullscreen, username, email, password	User is able to navigate between pages and the browser does not log the user out.	PASS	PASS
4	UI Responsiveness	Ensure that the page reflows and resizes content based on the available window/screen area	Username and password	The web browser window should be able to be adjusted and the website should respond correctly without displaying any distortion.	PASS	PASS
5	Website navigation through each page	Make sure that users can visit each page given the correct authentication conditions and routing.	Fullscreen, username, email, password	All users shall be able to go from one page to another, no matter the page they're on, and the pages should be redirected correctly based on the page they intended to go into.	PASS	PASS

4. Code Review

4.1. **Code Style :**

- 4.1.1. We are using svelte for our frontend coding structure. It is very easy to use as we can create each individual component and attach it with other components.
- 4.1.2. We created individual pages using svelte which involves HTML and CSS for styling and called individual components like buttons, cards onto the page and added contents inside it. As shown in the below screenshot, we created Searchbar and Card components which have their own styling and structure. We imported them in other pages and those styling will be applied to the contents automatically.

```
<Searchbar placeholder={"Browse"} />
</div>
<h1>Highlights</h1>
<div class="card-container">
  <Card {...cardData.card1} />
  <Card {...cardData.card2} />
  <Card {...cardData.card3} />
</div>
```

- 4.1.3. We are using Node and Express for backend, which renders the data from the database and passes it along to the Frontend.

4.2. **Team 5 Code review by Saru of Team 4 - reviewed users.js file**

- 4.2.1. Naming Convention :
 - The standard naming convention has been followed.
- 4.2.2. Code Header :
 - The name of the file is descriptive to reflect the content inside the file.
- 4.2.3. Commenting :
 - Comments are missing. Adding comments is recommended as it helps understand code better.

4.2.4. Limited Use of Global Variables and Functions:

- There is limited use of global variables and functions indicating good coding practice.

4.2.5. Indentation:

- The indentations are placed properly increasing readability of the code.

4.2.6. Exception Handling:

- Every function contains an exception handler which is great.

4.2.7. Statements:

- The statements are clear and comprehensive.

4.3. **Code Review By Nishit Pachchigar and Gautami Gudla - Reviewed Signup.svelte**

4.3.1. Naming Convention:

- The standard naming convention is followed well which makes it easy to understand the code.

4.3.2. Code Header:

- The minimal code header is not adequate. It is important to understand the purpose of the code function, which is why minimal code header is required.

```
        }
    }

    function validateUsername() {
        usernameElem.value = usernameElem.value.trim();
        const regex = /^[^A-Za-z0-9]/;
        let test1 = usernameElem.value.trim() != "";
        let test2 = !usernameElem.value.trim().includes(' ');
        let test3 = !regex.test(usernameElem.value);

        if (test1 && test2 && test3) {
            usernameValid = true;
        }
    }
}
```

4.3.3. Commenting

- Necessary amount of comments not present in the file. Comment helps other team members to understand the functionality of the code.

```

function validateName() {
    const regex = /^[^@#$%^&*()_-\+\=\{\}\[\]\]\\`~<>?]+$/;
    fullnameElem.value = fullnameElem.value.trim();
    let test1 = fullnameElem.value != "";
    let test2 = regex.test(fullnameElem.value);
    console.log("Full name", test1, test2);
    if (test1 && test2) {
        fullnameValid = true;
    }
    else if (!test1) {
        fullnameValid = false;
        fullnameError = "Full name must not be blank.";
        showFullscreenError = true;
    }
    else if (!test2) {
        fullnameValid = false;
        fullnameError = "Full name must not contain special characters.";
        showFullscreenError = true;
    }
    else {
        fullnameValid = false;
        fullnameError = "Full Name is invalid.";
        showFullscreenError = true;
    }
}

```

4.3.4. Limited Use of Global Variables and Functions:

- Limited use of global variables and functions is present which serves the purpose of encapsulation. It is a good practice to make the code better.

4.3.5. Indentation:

- The functions are guiding the users to put the correct input and in case of invalid prompt by the user, it will help the users to enter valid input by providing the proper indentation.

```

if (passwordElem.value == "") {
    passwordError = "Password cannot be empty";
} else {
    passwordError = "Password must meet the following conditions.";
}

```

4.3.6. Exception Handling:

- Exception handling is present in the code in the form of if and else statements. It is making sure to handle any unexpected hurdle.

4.3.7. Statements:

- The statements are following consistent styling and appropriate indication but only comments are missing for easy understanding of the code.

5. Self-Check On Best Practices For Security :

5.1. List major assets you are protecting :

- 5.1.1. We're protecting the user's private information such as logins and passwords.
- 5.1.2. We are restricting un-registered users to react and respond with comments. In order to do that, they need to create an account.
- 5.1.3. Users need to have an account in order to follow another user's account.

5.2. Confirm that you encrypt PW in the DB :

- 5.2.1. We have PW encrypted in the db using bcrypt using 10 salt rounds.
- 5.2.2. Here is a screenshot of PW encryption :

password
\$2b\$10\$/BPF61cLo6qN.Ciu3G5HNOhPkdgShVVbmxNa3D6qeR7RKZxN8ulvG
\$2b\$10\$4Vx7R5J6pZ8QcHKNfb15KOXfk61uHBJB7hOpq/rAVXPwgGAXglAPu
\$2b\$10\$8g3ZjnrtH3.3qtwLVBQlejNtpQSr.opYSB3SyaJGsxtMr9OvwALy
\$2b\$10\$HcWnFluUx8e.Z97A42GLLu2mNEQw.eYQFkWnZ6vaa81Z3QXk7yaCa
\$2b\$10\$LmZZKczd.7BGYM2ZJThpQ.z3rzC9W.oQTBtBGeR/adoHOx/JEvre
\$2b\$10\$ZwbcLMT91raPCeTfTqCbpeEWfetzUKRtcdXgdYJYWigrM2cHXRICS

5.3. Confirm Input data validation :

- 5.3.1. While creating an account, users should meet certain requirements like for the password and username without any space.
- 5.3.2. Users cannot have special characters for their username and the backend will also have a validation function to check if that username already exists or not. And the same thing will also be checked for the email.
- 5.3.3. Search box will parse the request into the backend if no results match then it will show suggestions.

6. Self-check: Adherence To Original Non-Functional Specs

6.1. Performance:

- 6.1.1. The website should be available 24/7 with a minimum of downtime for maintenance and updates.
- 6.1.1.1. ISSUE: the AWS server is a free trial and we have all the instances running on one of the team's accounts. We keep having to restart the instances for the program to work on the site using the URL.

- 6.1.2. Scalability: The website should be able to handle a large number of users and posts without slowing down or crashing.

6.1.2.1. ON TRACK

6.2. Security:

- 6.2.1. Passwords must be encrypted in the database.

6.2.1.1. DONE

- 6.2.2. Each account should have a unique email.

6.2.2.1. DONE

- 6.2.3. Each account should have a unique username.

6.2.3.1. DONE

- 6.2.4. Logged in users must stay logged in while navigating unless they log out.

6.2.4.1. DONE

6.3. Legal:

- 6.3.1. To complete registration, users must accept T&C.

6.3.1.1. DONE

- 6.3.2. Privacy Policy.

6.3.2.1. DONE

6.4. Compatibility:

- 6.4.1. The website should be compatible with a range of devices, browsers, and operating systems.

6.4.1.1. ON TRACK

- 6.4.2. The UI should be responsive to all screen sizes and device form factors

6.4.2.1. ON TRACK

6.5. Usability:

- 6.5.1. The website should be user-friendly, with a clear interface that is easy to navigate.

6.5.1.1. DONE

6.6. Marketing:

- 6.6.1. The website will have a logo to represent the company.

6.6.1.1. DONE

6.6.2. The website should have a description to explain what the website is about.

6.6.2.1. ON TRACK

7. List Of Contributions

Role	Name	Contributions	Score
Team Lead	Jose Avila	Delegated task, scheduled meetings, set reminders for the team, gave input about U/I, reviewed frontend and backend code, reviewed documents, worked on usability test plan, asked questions to the CTO on behalf of the team, and gave back feedback from CTO to the team.	
Front End Lead	Andy Shi	Coded the frontend UI, set up routes, scheduled meetings with front end members, created wireframes and components, helped with selecting our final P1s and collaborated with the back end.	9
Back End Lead	Leo Saeteurn	Schedule meetings with back end members, coded algorithms, helped with QA Test Plan, helped select final P1 requirements, and Adherence to original non -functional specs.	8
Database Master	Joshua Hayes	Created all database tables, set up routes, collected data and coded back end. Worked on the	9

		usability test plan, and self checked on best security.	
GitHub Master	Nishit Pachchigar	Helped Andy code the frontend, help with routing and creating components, participated in the code review, and helped Josh with the self check on best security. Help select finalP1s and kept Github up to date.	9
Document Editor	Gautami Gudla	Edited & updated the document, wrote the product summary, participated in the code review with Nishit, worked on the frontend, coded components, revised P1 requirements, gave us reminders, and proofread the document.	8

Team Contributions

Name	Role	Contributions	Score
Jose Avila	Team Lead	Delegated task, scheduled meetings, set reminders for the team, gave input about the aesthetics and functionality of the project, answered questions and lent a hand when needed. gave input about U/I, created wireframes using figma, gave back feedback from CTO, motivated the team, and resolved internal issues. Reviewed frontend and backend code, reviewed documents, worked on usability test plan, and asked questions to the CTO on behalf of the team	
Andy Shi	Front End lead	Use cases, functional requirements, index page for the about us page, and set up Svelte. Wrote data definitions. Lead group effort to decide on color palette and typefaces. Created mockups and user flows storyboards on paper. Worked on frontend homepage code and created UI components. Collaborated with the backend lead, github master, and database master to code backend and connect database with backend and frontend for	10

		vertical SW prototype. Set up routes, schedule meeting with front end members, created wireframes and components, wrote data definitions, helped with selecting our final P1s.	
Leo Saeteurn	Back End Lead	Executive summary, title page, table of contents, use cases, functional requirements, document formatting, and entities. Prioritized functional requirements, high level database architecture and organization, high level APIs and main algorithms, high level UML diagrams, high level application network diagram, collaboration on logo, collaboration on UI for color and font themes. Collaborated with the frontend lead, github master, and database master to code backend and connect database with backend and frontend for vertical SW prototype. Schedule meetings with back end members, coded back end, created the uml diagram and wrote/revised functional requirements. Coded algorithms, helped with QA Test Plan, helped select final P1 requirements, and Adherence to original non-functional specs	9
Gautami Gudla	Document Editor	Functional requirements, non- functional	7

		<p>requirements, entities, and use cases</p> <p>Identified actual key risks and solutions for the project. Wrote project management and data definitions.</p> <p>Collaboration on UI for color and font themes.</p> <p>Proofread the complete milestone 2 document to meet the requirement.</p> <p>Collaborated with the front end team for the vertical SW prototype.</p> <p>Edited & updated the document, wrote data definitions, worked on the frontend, created components, revised P1 requirements, gave us reminders, and Wrote the product summary,</p> <p>participated in the code review with Nishit,</p> <p>worked on the frontend, coded components,</p> <p>revised P1 requirements,</p> <p>and gave us reminders.</p>	
Joshua Hayes	Database Master	<p>Competitive analysis, functional requirements, and use case Prioritized functional requirements, high level database architecture and organization, deployment diagram. Created database from diagram for backend virtual prototype.</p> <p>Collaborated with the backend lead, github master, and frontend lead to code backend and connect database with backend and frontend for vertical SW prototype.</p> <p>Created all databases ,</p>	9

		<p>created the high level database diagram, wrote/revised functional requirements, set up routes, collected data and coded back end.</p> <p>Worked on the usability test plan, and self checked on best security.</p>	
Nishit Pachigar	GitHub Master	<p>Set up AWS, set up our databases instance, wrote some use cases, table of contents, and competitive analysis. He also collaborated with the backend lead, frontend lead, and database master on how to implement backend and connect with the frontend for vertical SW prototype. Worked on AWS and managed Trello. Managed the GitHub files created High Level Network and Deployment diagrams, Helped Andy code the frontend, help with routing and creating components, participated in the code review, and helped Josh with the self check on best security. Help select final PIs and kept Github up to date.</p>	8

Nishit Pachchigar	Jose Avila	Andy Shi		Joshua Hayes	Gautami Gudla
-------------------	------------	----------	---	--------------	---------------

Post analysis – lessons learned

This project has been full of ups and downs all semester. Mainly because for many of us this was the first time we had to get out of our comfort zone and work with other people or even work on language/technologies that we have never used before. But breaking through comfort zones is what helps us grow as a person and will help us with our careers in the long run. Since the beginning we had one clear goal and that was to create not just a school project but an application that could potentially be a company. Of course we still have a lot of work to do to reach that goal but we would probably be further ahead than where we are now if we would've started with a better game plan with a clearer path. One of the mistakes that we ran into was picking the wrong language for the back end. We initially started off using Java Spring Boot but many of the teammates spent a lot of time trying to learn it which set us back and we had to pivot so, we ended up changing the language to Javascript. Another problem we ran into was time management. We oftentimes would start on the milestones late which created a sense of panic and caused us to rush through it. Time management is certainly a key factor in creating an optimal project and that's something we all learned while doing this project. Not everything was a headache and set back though, there were plenty of times where we felt that the task was impossible to do and we persevered and more times than not we made it possible, it was an amazing feeling. Perseverance and a positive attitude was another great lesson learned and it's needed to get through all the frustrations and sores. Overall, we had a great experience working on this project. We learned many lessons, gained new skills, and met great people along the way.

