

SW Engineering CSC648-848-05 Spring 2024

Application Name: **Swamp Study**

Seal Team One Members (Team 01)

Lennart Richter lrichter@sfsu.edu	Team Lead
Edmund Huang	Front-End Lead
Conrad Choi	Github Master
Julio Reyes	Docs-Editor
Rafael Fabiani	Backend-Lead

Version Table

Milestone 1	Feb 28, 2024
Milestone 1 V2	Apr 1, 2024
Milestone 2	Apr 4, 2024

Table of Contents

SW Engineering CSC648-848-05 Spring 2024.....	1
Table of Contents.....	2
Data Definitions.....	4
User.....	4
Account.....	4
Profile.....	4
Profile Picture.....	4
Match.....	4
Profile.....	4
Availability.....	4
Forum.....	4
Class.....	5
Thread.....	5
Question.....	5
Answer.....	5
Application.....	5
Rating.....	5
Functional Requirements.....	6
Priority 1: Must Have.....	6
Priority 2: Desired.....	7
Priority 3: Opportunistic.....	7
Use Cases.....	8
Priority 1: Must Have.....	8
Use case 1: Student using site needs a partner for a group project.....	8
Use case 3: Non-Traditional Student looking for study partner.....	14
Use case 3 Storyboard:.....	14
Use case 4: User looking to find a safe person to study with.....	18
Use case 4 Storyboard: Draft in progress.....	18
Use case 5: Finding a long term study partner.....	22
Use case 5 Storyboard (FROM DANIEL'S PERSPECTIVE):.....	22
Use case 6: Users looking for a group to work with.....	25
Use case 6 Storyboard.....	26
Use case 7: Introvert looking for someone with similar traits to study with.....	30
Use Case 7 Storyboard.....	30
Use case 8: Users ensuring the quality of matches by rating.....	34
Use Case 8 Storyboard.....	34
Use case 12: User getting his questions answered on the forum.....	37
Use Case 12 Storyboard.....	37
Database Architecture.....	40

Entity Relationship Diagram.....	42
Entity Establishment Relation Diagram (EER).....	43
DBMS System.....	43
Media Storage.....	44
Searching the Database.....	44
High Level API's and Main Algorithms.....	46
Login API:.....	46
Logout API:.....	46
Register API:.....	46
Match API.....	46
Like API.....	47
Forum API:.....	47
Like and Match Algorithm.....	47
Session Algorithm.....	48
Search Algorithm.....	48
System Design.....	50
UML Class Diagram:.....	50
Scalability Diagram.....	51
Microservices Diagram.....	52
High Level Application Network and Deployment Design.....	53
Deployment Diagram.....	54
Key Risks.....	54
Skills Risks:.....	54
Schedule Risks:.....	55
Project Management.....	55
Team Evaluation.....	56
Lennart (Team Lead) 10.....	56
Rafael (Backend Lead) 10.....	57
Edmund (Frontend Lead) 10.....	57
Julio (Docs-Editor) 10.....	57
Conrad (Github-Admin) 10.....	58

Data Definitions

User

A User is a student that has signed up with their email for the platform. The email has to have an “@sfsu.edu” ending.

Account

Each User has an account which correlates to their email and password. The password is stored hashed in the Database as a string

Profile

A Profile contains a User's description and profile picture. This data will be used to facilitate better matches

Profile Picture

An image used to identify users on the platform. Represented by a .jpg extension can have a maximum size of 5 MB.

Match

A Match is a set of 2 or more users that have both liked each other. This Match is then used to facilitate a meeting and communication between the users.

Profile

A profile is a public display of a user's information, such as availability, subject and many more.

Availability

Availability is the times available a user can attend a study session. Represented by a Date and Time field, typically showing days and times in AM/PM that the user is available

Forum

The Forum is a part of the Application where Users can ask and answer questions related to specific classes. The Forum is organized into threads by class

Class

A Class is the representation of an SFSU course. Represented by a string Department and Course Name, as well as an integer representing the Class Number

Thread

A Thread is used to organize the Forum. Each thread contains one question and many answers. Each thread belongs to a specific class

Question

A Question is a text written by a User, the Author. Each Question automatically creates a thread

Answer

An Answer is a text written by a User in response to a question. Each answer belongs to a thread.

Application

The application holds all the components and necessary data for the user to interact with and engage with the platform.

Rating

Each user will have ratings which will get averaged to get their rating within the system as a study partner

Functional Requirements

Priority 1: Must Have

1. User Account Management

- 1.1. A User shall be able to create an account
- 1.2. A User shall only have one account
- 1.3. An Account shall have one email associated with it

2. User Profile Management

- 2.1. A User shall be able to view other users' accounts in the system
- 2.2. A User shall be able to set their availability
- 2.3. A User shall be able to establish their profile
- 2.4. A User shall be able to accept or decline study partner invitations
- 2.5. A Profile shall be owned and edited by one and only one user
- 2.6. A Profile shall be allowed one picture of the owner
- 2.7. A Profile shall be allowed a short biography
- 2.8. A Profile shall have many subject attributes

3. User Interface

- 3.1. The Interface shall allow a User to navigate through a potential Match's profile
- 3.2. The Interface shall allow a User to access/edit their Profile
- 3.3. The Interface shall allow a User to access/edit their Account

4. Matching System

- 4.1. A Set of Users shall become matched if they Like each other
- 4.2. A Matched pair shall be able to contact each other
- 4.3. A Matched pair shall tell the system after their first meeting

5. Forum

- 5.1. The Forum shall allow one main question per Thread
- 5.2. The Forum shall allow many answers until marked done
- 5.3. The Forum shall allow each class its section
- 5.4. The Forum shall allow users to post in their class Forum
- 5.5. The Forum shall allow many questions
- 5.6. A User shall be able to ask many questions in the Forum
- 5.7. A User shall be able to answer many questions in the Forum

6. Additional Features

- 6.1. A Matched Pair shall be able to unmatch each other
- 6.2. A User shall be able to flag another user for review

Priority 2: Desired

- 1. User Account Management**
- 2. User Profile Management**
 - 2.1. A Profile shall be able to set a filter for their study partner preferences
 - 2.2. A Profile shall be allowed to be invisible when not wanting to study
- 3. User Interface**
 - 3.1. The Interface shall allow a user to swipe right or left on a potential study partner
- 4. Matching System**
 - 4.1. The System shall highlight common attributes users have with each other
 - 4.2. The System shall notify users when they have a Match
 - 4.3. The System shall send a notification to the user's device
 - 4.4. A High Rating shall make the user more visible
 - 4.5. The System shall show a user their ideal match
- 5. Forum**
 - 5.1. Only Users who have passed/are taking the class shall be able to access the Forum
 - 5.2. A Professor shall be able to access the Forum
 - 5.3. A Professor shall be allowed to answer questions in the Forum
 - 5.4. A Professor shall be allowed to mark questions for removal in the Forum
- 6. Additional Features**
 - 6.1. A Matched Pair shall be able to decide whether they want to meet in person or online
 - 6.2. A Matched Pair shall automatically be given an appointment
 - 6.3. A Matched Pair shall automatically be made a Zoom Meeting
 - 6.4. A Profile shall be marked Student or Professor
 - 6.5. A User shall be able to block another user from seeing their Profile

Priority 3: Opportunistic

- 1. User Account Management**
 - 1.1. An Account is allowed to log in from only one device at a time
- 2. User Profile Management**
 - 2.1. A User shall be able to video call with their partner

Use Cases

Priority 1: Must Have

Use case 1: Student using site needs a partner for a group project.

Use case 1 Storyboard:

Sebastian is worried about his grades and is stressed about midterms. He wants to find a study partner but has trouble.



Remembering hearing about SwampStudy, Sebastian signs up and starts to look for a study partner.



ooo

Swamp Study

Welcome!

Finding your Study partner, made EASY

Login or create an account to get started!

Email: [REDACTED]

Password: [REDACTED]

Remember Me

[Forgot Password?](#)

Login

[Don't have an account?](#)

Register 

We have:

- a matching system that takes user preferences into account
- class forums to ask/answer questions about topics

The screenshot shows a web browser window with a purple header bar. The header contains three white circles on the left, a search bar in the center, and the text "Swamp Study" on the right, accompanied by a cartoon alligator icon. Below the header is a yellow oval button labeled "About". A red arrow points from the top-left towards the "About" button. Another red arrow points down to the "Register" button. A third red arrow points to the "Register & Login" button at the bottom.

Register

Email: * - this field is required
* sebas1@sfsu.edu

Password: *
* *****

Confirm Password: *
* *****

*Password must be 8-16 characters long
*Password must contain AT LEAST 1 uppercase letter
*Password must contain AT LEAST 1 number

First Name: * Sebastian

Last Name: * Stan

I comply with the terms and conditions.

Register & Login ← Click

ooo

SWAMP STUDY

Profile 

START MATCHING **FORUM**

General chat

- o Vegetal125 : who can beat Gator?
- o Shagville-oatmeal: what is your favorite smell?
- o Something123: whats the wifi password?

Ask a question:

Search questions:

General chat

Select Course :

No courses found

Add courses in profile

ooo

Swamp Study 

Matching **Forums**

Profile

Picture:
No Image Found
Click to upload

Study:
Courses:
- Math 226

Availability:

Mon:	4:00pm - 6:00pm
Tue:	10:00am - 11:00am
Wed:	1:00pm - 3:00pm
Thu:	11:00am - 12:00pm
Fri:	
Sat:	
Sun:	

Hide Account:

* If checked, your profile will be invisible to other users.

Save

ooo

About Swamp Study Profile Logout

Matching Forums

Match with people here! 

Drag to (or click) the side to match or decline

Name	Rating
	

Availability:

Mon :
Tue :
Wed :
Thu :
Fri :
Sat :
Sun :

Courses Studying:
- CSC510

Notifications

New You matched with [Name]

ooo

About Swamp Study Profile Logout

Matching Forums

You matched with [Name]

Name	Rating
	

Availability:

Mon:
Tue:
Wed:
Thu:
Fri:
Sat:
Sun:

Studying
Class

Click 

Confirm Decline

You matched with [Name]
NEW Meeting details with [Name]

ooo

About Swamp Study Profile Logout

Matching

Forums

Meeting details with [Name]

Date: 04/02/24

Time: 01:00 PM - 03:00 PM

Location: SFSU Library First Floor

Study Course: CSC510

Study Partner: [Name]

Notifications

Meeting details with [Name]

You matched with [Name]!

After matching with a study partner and getting automatically scheduled a time and place for the study session, Sebastian feels much less worried.



Use case 3: Non-Traditional Student looking for study partner

Use case 3 Storyboard:

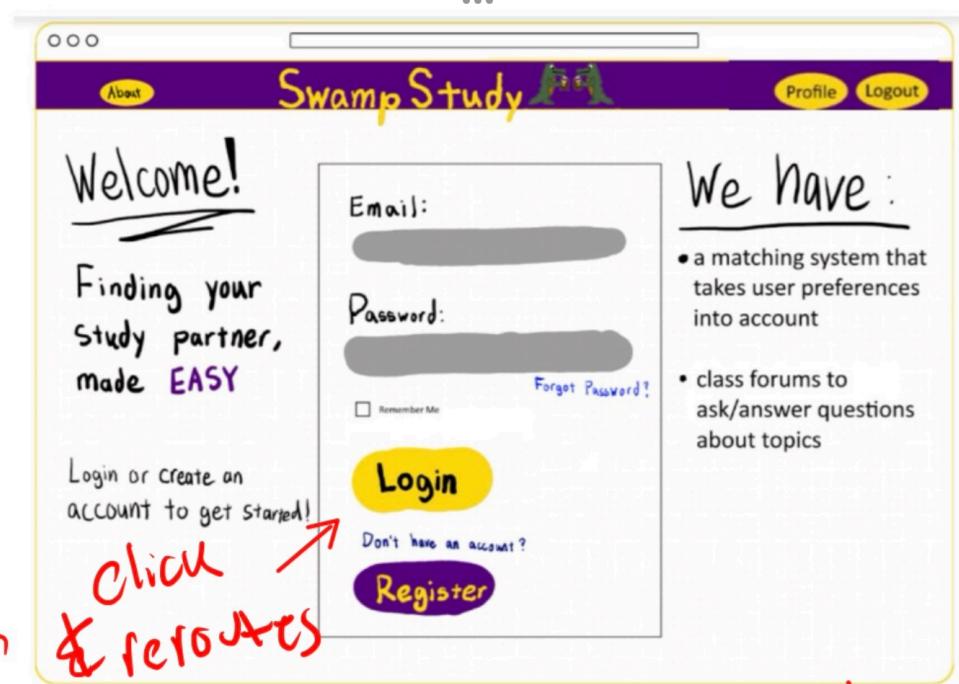
Richard has a lot on his plate, he has a family, goes to school full time and has a full time job.

Due to not being able to find someone to study with that also has his rigid schedule, his home life, school work and his job are suffering



One day he discovered Swamp Study and is able to find a study partner that matched his free time slot. Now he can stop worrying so much about school and focus on his job and family. Everyone is happier.

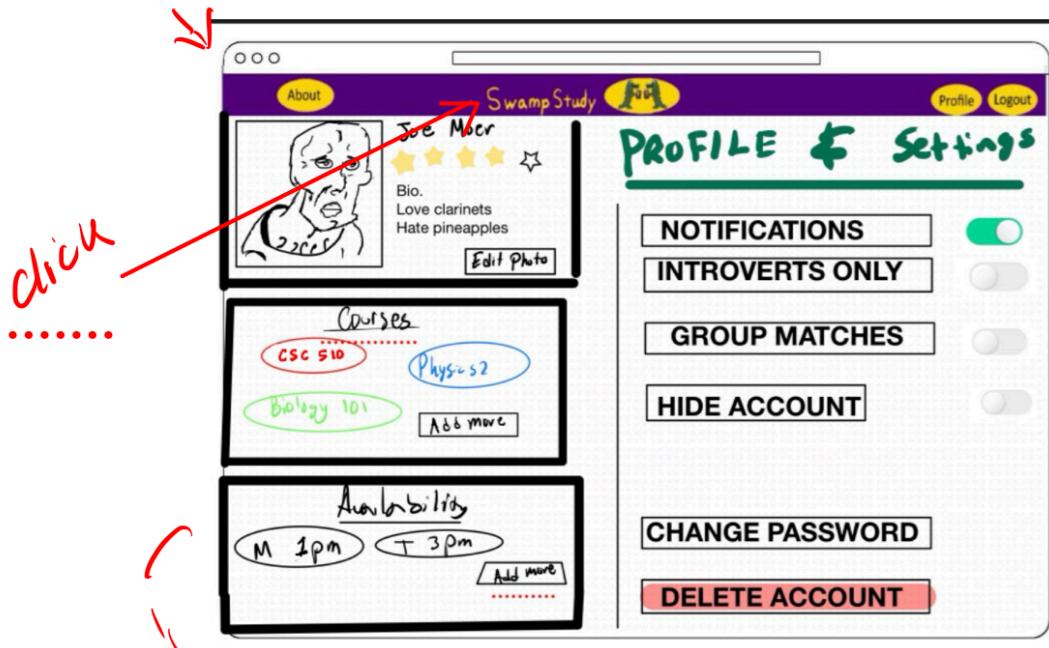
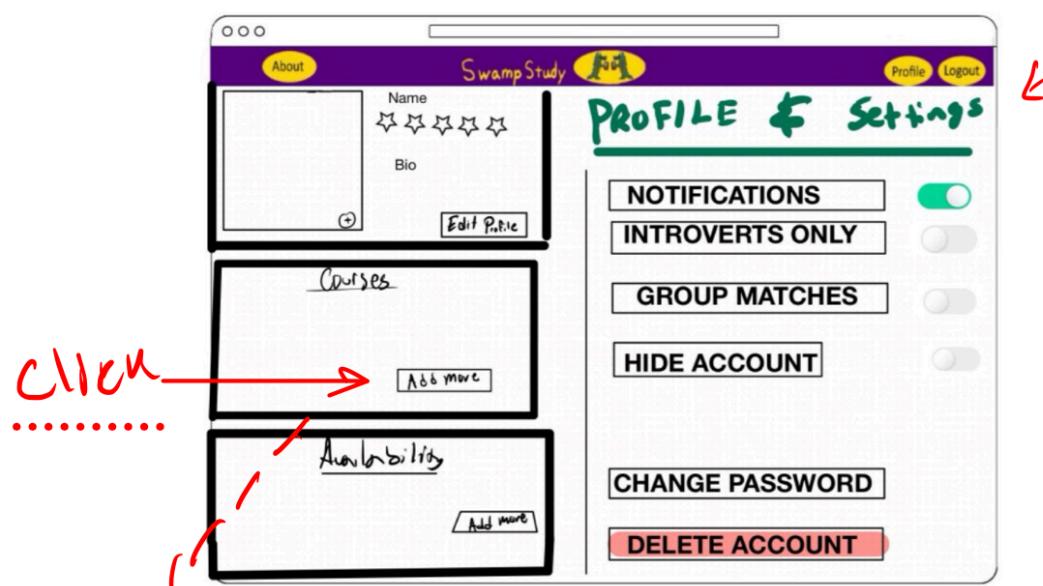


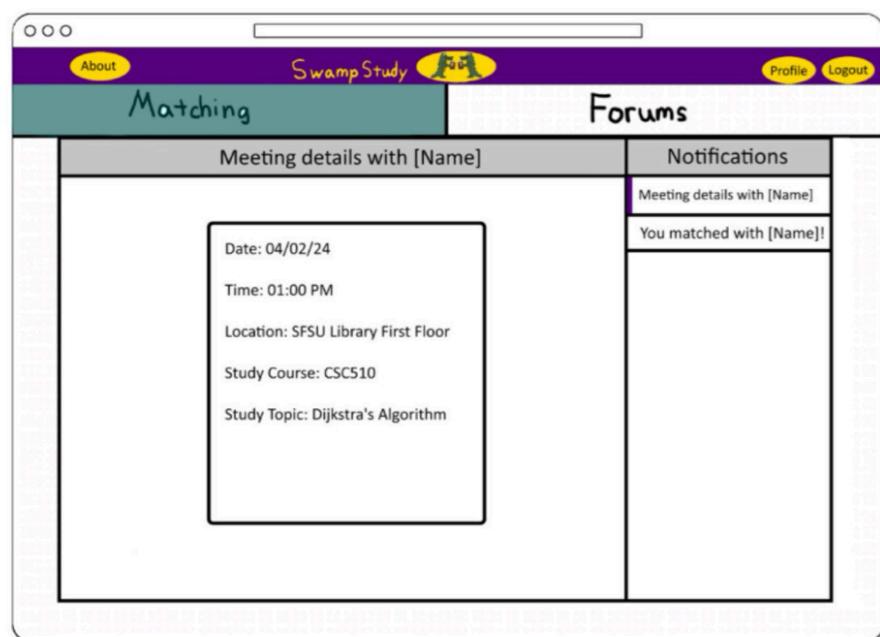
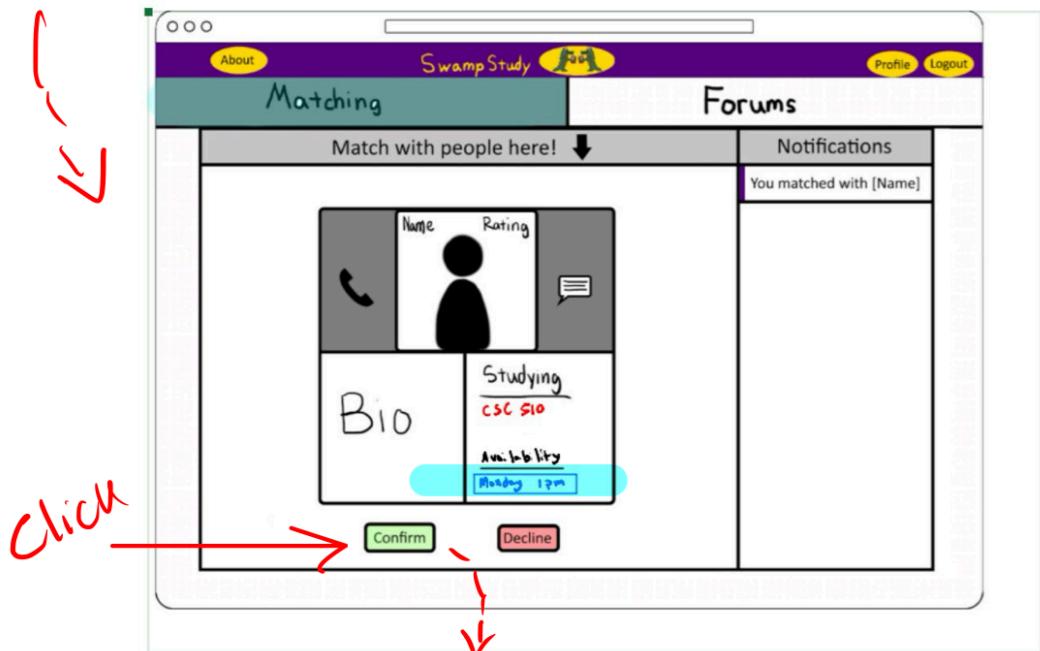


A hand-drawn wireframe of a web page titled "Swamp Study". The header includes "About", "Profile", and "Logout" buttons. The main content area has tabs for "Matching" and "Forums", with "Matching" currently selected. On the left, a sidebar titled "General Chat" lists various course options: "Select Course:", "math 101", "Science 101", "History 102", "Biology 101", "Calc 2", "CSC 600", "Business 201", and "Art 100". The right side shows a list of forum posts in a green-bordered box:

- Vegeta123: Who can beat Goku? (3 comments, 7 likes)
- Shaggy1000-Oatmeal: What is your favorite smell? (4 comments, 4 likes)
- Something123: What's the best password? (3 comments, 7 likes)
- illidozTumorrow: What's your favorite soap? (4 comments, 5 likes)
- HairyPotter: Do you believe in Aliens? (1 comment, 6 likes)

Below the posts are two input fields: "Ask a question:" and "Search questions:". Red annotations include a red arrow pointing to the "Matching" tab and another red arrow pointing to the "Forums" tab. A large red arrow points from the "Matching" sidebar down to the "Ask a question:" field.





Use case 4: User looking to find a safe person to study with

Use case 4 Storyboard: Draft in progress

This is Melanie. She gets
distracted from studying easily
and her grades suffering.



Melanie gets on her SwampStudy account and matches with a potential study partner. Melanie appreciates the pre-confirmation contact methods available before meeting in real life.



ooo

Welcome!

Finding your Study partner, made EASY

Login or create an account to get started!

Email: mel3@sfsu.edu

Password: *****

Remember Me

[Forgot Password?](#)

Login *click*

[Don't have an account?](#)

Register

We have:

- a matching system that takes user preferences into account
- class forums to ask/answer questions about topics

ooo

SWAMP STUDY

[Home](#) [About Us](#) [Profile](#) [Account](#) [Logout](#)

START MATCHING *Click*

FORUM

General chat

o Vegetal25 : who can beat Gollu ?

o Shaquille-oatmeal: what is your favorite smell?

o Something123: whats the wifi password?

Ask a question:

Search questions:

General chat

Select Course :

No courses found

Add courses in profile

ooo

Matching

Match with people here!

Drag to (or click) the side to match or decline

Name	Rating

Availability:

Mon :	
Tue :	
Wed :	
Thu :	
Fri :	
Sat :	
Sun :	

Courses Studying:
- CSC510

Profile Logout

NEW You matched with [Name]!

ooo

Matching

You matched with [Name]

Name	Rating

Availability:

Mon:	
Tue:	
Wed:	
Thu:	
Fri:	
Sat:	
Sun:	

Studying
Class

Profile Logout

NEW You matched with [Name]!

Meeting details with [Name]

Confirm Decline

click to send call request

click to send text request

confirming creates meeting only if other user confirmed as well

ooo

About Swamp Study Profile Logout

Matching Forums

Meeting details with [Name]

Date: 04/02/24

Time: 01:00 PM - 03:00 PM

Location: SFSU Library First Floor

Study Course: CSC510

Study Partner: [Name]

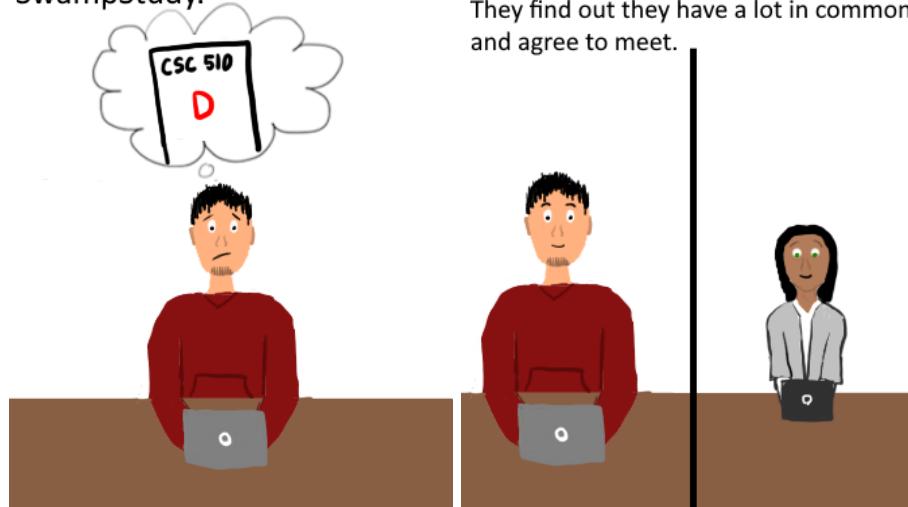
Meeting details with [Name]!

You matched with [Name]!

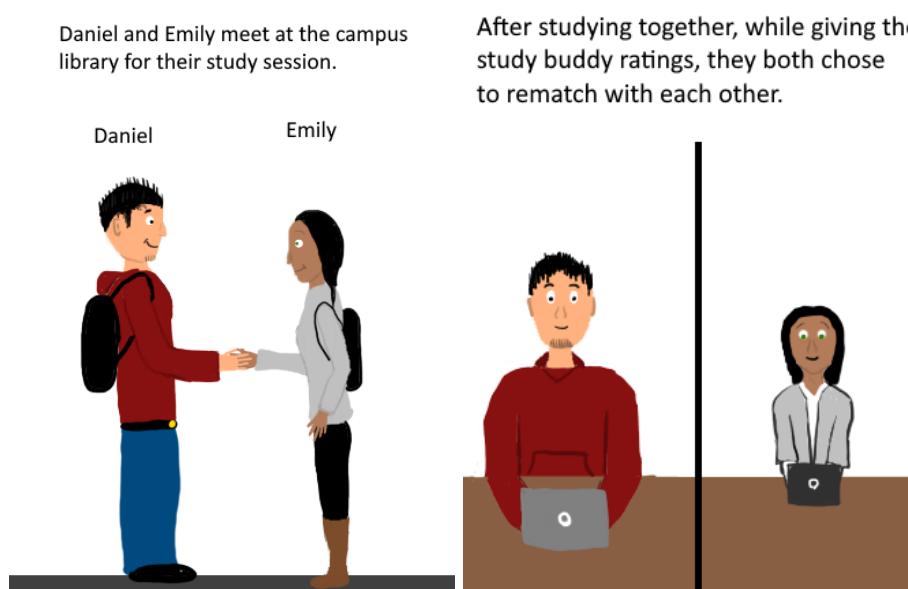
Use case 5: Finding a long term study partner

Use case 5 Storyboard (FROM DANIEL'S PERSPECTIVE):

Daniel is struggling with CSC510 and wants to find a study partner so he logs into SwampStudy.



Daniel matches with Emily and they discuss their study habits and etc. They find out they have a lot in common and agree to meet.



Daniel and Emily meet at the campus library for their study session.

After studying together, while giving the study buddy ratings, they both chose to rematch with each other.

Finding your Study partner made EASY

ooo

Swamp Study



Welcome!

Email: dan3@sfsu.edu

Password: *****

Remember Me

[Forgot Password?](#)

Login

Don't have an account?

Register

ooo

ooo

SWAMP STUDY


Home
About Us
Profile
Account
Logout

START MATCHING
← Click
FORUM

General chat

- Vegetal25: Who can beat Gollum?
- Shaquille-oatmeal: What is your favorite smell?
- Something123: Whats the WiFi password?

General chat

Select Course :

No courses found

Add courses in profile

Ask a question:

Search questions:

The screenshot shows the Matching section of the Swamp Study app. At the top, there's a purple header bar with yellow buttons for "About", "Profile", and "Logout". Below the header, the word "Matching" is displayed in a teal bar. A large green button with a downward arrow says "Match with people here!". To the left of this button is a red "X" and a note: "Drag to (or click) the side to match or decline". On the right, a green checkmark indicates a successful match: "You matched with Emily!". The central area contains a user profile card with fields for Name (empty), Rating (empty), Availability (empty), and Courses Studying ("- CSC510").

This screenshot shows the Matching section after a successful match with Emily. The purple header bar and teal "Matching" bar are visible. A green banner at the top of the main area says "You matched with Emily!". The central profile card now displays a user picture of a woman named Emily with a 5-star rating, availability from Monday to Sunday, and the course "- CSC510" listed under "Studying". At the bottom of the card are "Confirm" and "Decline" buttons. To the right, the Notifications panel shows a new message: "You matched with Emily" and "Meeting details with Emily". Red arrows indicate the path from the "Match with people here!" button through the matching process to the confirmation screen.

ooo

[About](#) SwampStudy  [Profile](#) [Logout](#)

Matching

Forums

Meeting details with Emily	Notifications
Date: 04/02/24 Time: 01:00 PM - 03:00 PM Location: SFSU Library First Floor Study Course: CSC510 Study Partner: Emily	Meeting details with Emily You matched with Emily

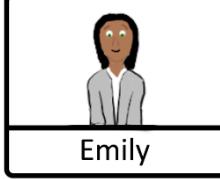
ooo

ooo

[About](#) SwampStudy  [Profile](#) [Logout](#)

Matching

Forums

Rate your study partner	Notifications
How would you rate your study partner?  Emily 	Rate your study partner

Feedback

Would you like to partner with this person again?

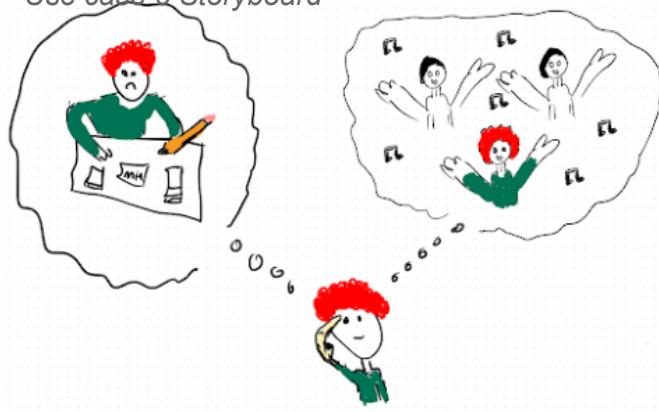
Yes No

Submit

ooo

Use case 6: Users looking for a group to work with

Use case 6 Storyboard



Kyle is a social butterfly, he craves company. He knows he has to study, and is happily surprised to find a "group section" in the SwampStudy application! He can now have the best of both worlds!



The screenshot shows a mobile application interface titled "Forums". At the top, there are navigation buttons for "About", "Profile", and "Logout". Below the title, there is a section titled "Matching" with a yellow header. A sidebar on the left lists various courses under "Select Course": "math 101", "Science 101", "History 102", "Biology 101", "Calc 2", "CSC 600", "Business 201", and "Art 100". The main content area displays several posts from users:

- o Vegetal125: Who can beat Goku? (3 comments, 7 likes)
- o Shaggydoggeman: What is your favorite smell? (4 comments, 4 likes)
- o Something123: Whats the wifi password? (3 comments, 7 likes)
- o illdoitTommorrow: Whats your favorite song (4 comments, 5 likes)
- o Harry Potter: Do you believe in aliens? (1 comment, 6 likes)

Below the posts are two input fields: "Ask a question:" and "Search questions:". A red arrow points from the text "click" to the "Ask a question:" field.

The screenshot shows a mobile application profile settings screen. At the top, there are buttons for "About", "Profile", and "Logout". On the left, there is a user profile card for "Joe Moor" featuring a caricature photo, a 5-star rating, and bio information: "Bio. Love clarinets Hate pineapples". There is also an "Edit Photo" button. Below the profile card are sections for "Courses" and "Availability".

Courses: Calc 2 (circled in red), Physics 2, Biology 101 (circled in green). There is a "Add more" button.

Availability: M 2pm, T 3pm. There is an "Add more" button.

The right side of the screen contains the "PROFILE & Settings" section with the following options:

- NOTIFICATIONS (switched on)
- INTROVERTS ONLY (switched off)
- GROUP MATCHES** (switched on, circled in red)
- HIDE ACCOUNT (switched off)
- CHANGE PASSWORD
- DELETE ACCOUNT (button highlighted with a red border)

A red arrow points from the text "click" to the "DELETE ACCOUNT" button.

Click →

The screenshot shows a mobile application interface for "SwampStudy". At the top, there are tabs for "About", "Matching", "Forums", "Profile", and "Logout". The "Matching" tab is active. On the left, a sidebar titled "General Chat" lists various courses under "Select Course": "math 101", "Science 101", "History 102", "Biology 101", "Calc 2", "CSC 600", "Business 201", and "Art 100". To the right, a list of forum posts is displayed:

- Vegetable125: Who can beat Goku? (3 views, 7 comments)
- Shaggydog-dogment: What is your favorite smell? (4 views, 4 comments)
- Something123: Whats the wifi password? (3 views, 7 comments)
- illidiot-Tomorrow: What's your favorite soap? (4 views, 5 comments)
- HairyPotter: Do you believe in Aliens? (1 view, 6 comments)

Below the posts are two input fields: "Ask a question:" and "Search questions:".

Click ↴

The screenshot shows the same "SwampStudy" interface. The "Matching" tab is active. On the left, a user profile for "Bio" is shown with fields for "Name" (Studying), "Rating" (Class - Topic), and "Bio". Below the profile are "Confirm" and "Decline" buttons. On the right, a sidebar titled "Notifications" displays a message: "You matched with [Name]". Below this are links: "View matches", "View Groups" (which is highlighted in yellow), "Appointments", and "Reported Users".

Meeting details with [Name]

Date: 04/02/24

Time: 01:00 PM

Location: SFSU Library First Floor

Study Course: CSC510

Study Topic: Dijkstra's Algorithm

Group

Notifications

Meeting details with [Name]

You matched with [Name]!

View matches

View Groups

Appointment

Reported Us

Use case 7: Introvert looking for someone with similar traits to study with

Use Case 7 Storyboard



Taylor is an introvert, she is shy around others and prefers to keep to herself. But unfortunately she needs help in school and is anxious that the partner will talk about nonsense and end up distracting her.

She is happy to find that SwampStudy has a “introvert” filter that fixes her issue.

She is now free to study with other just like her, and avoid the uncomfortable small talk extroverts tend to do.



ooo

About SwampStudy Profile Logout

Matching

Forums

General Chat

Select Course :

- math 101
- Science 101
- History 102
- Biology 101
- Calc 2
- CSC 600
- Business 201
- Art 100

Vegetal125: Who can beat Goku? 3 7

Shaggydog-oatmeal: What is your favorite smell? 4 4

Something123: What's the wifi password? 3 7

illidoT+Tumorrow: What's your favorite soap? 4 3 5 7

HarryPotter: Do you believe in Aliens? 1 6

Ask a question:

Search questions:

A screenshot of a web application titled "SwampStudy". The interface includes a navigation bar with "About", "Profile", and "Logout" buttons. On the left, there's a sidebar titled "Matching" with a "General Chat" section and a list of course names. The main area is titled "Forums" and shows several user posts in a list format, each with a timestamp and interaction counts. Below the posts are input fields for asking a question or searching for questions.

Click

ooo

About SwampStudy Profile Logout

PROFILE & Settings

Joe Moer Introvert

★★★☆☆ Bio.
Love clarinets
Hate pineapples

Edit Photo

Courses

Calc 2 Physics 2

Biology 101 Add more

Availability

M 2pm T 3pm Add more

NOTIFICATIONS

INTROVERTS ONLY

GROUP MATCHES

HIDE ACCOUNT

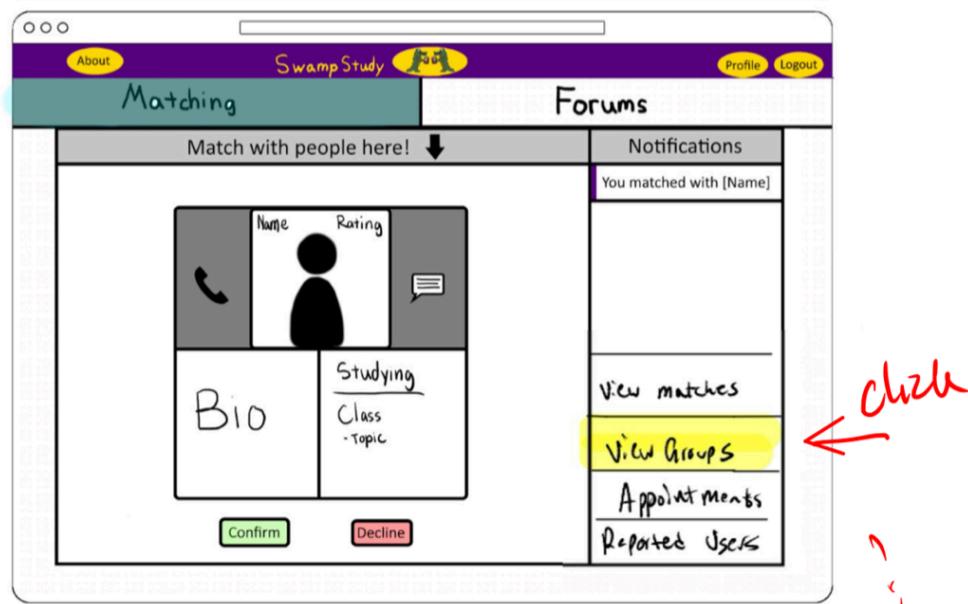
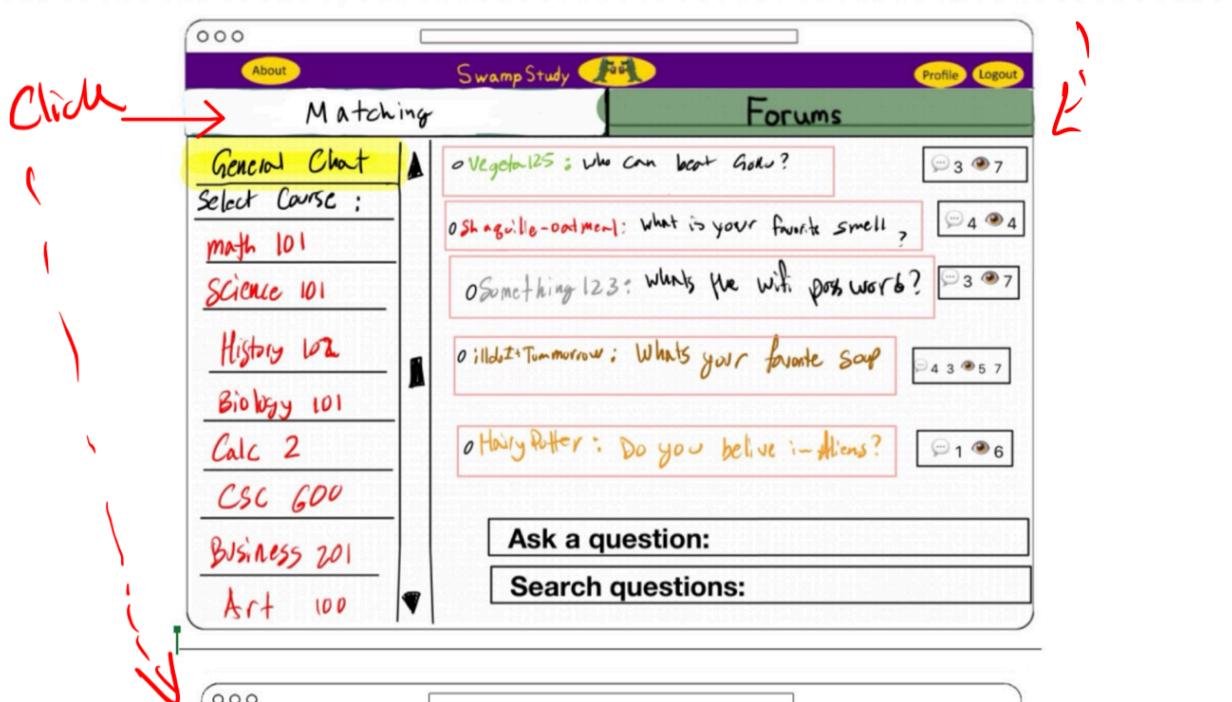
CHANGE PASSWORD

DELETE ACCOUNT

A screenshot of the "Profile & Settings" page. It displays a user profile for "Joe Moer" with an introvert status, a bio, and an edit photo button. Below this are sections for "Courses" (with "Calc 2", "Physics 2", and "Biology 101" listed) and "Availability" (with "M 2pm" and "T 3pm"). To the right are several toggle switches for account settings: "NOTIFICATIONS" (on), "INTROVERTS ONLY" (on), "GROUP MATCHES" (off), "HIDE ACCOUNT" (off), "CHANGE PASSWORD" (button), and "DELETE ACCOUNT" (button). Red arrows and the word "Click" highlight the "INTROVERTS ONLY" switch and the "DELETE ACCOUNT" button.

Click





ooo

About SwampStudy Profile Logout

Matching Forums

Match with people here! ↴

Calc 2

Biology 201

Notifications

You matched with [Name]

[View matches](#)

[View Groups](#)

[Appointments](#)

[Reported Users](#)

Click ↴ ↴ ↴

ooo

About SwampStudy Profile Logout

Matching Forums

Meeting details with [Name]

Date: 04/02/24
Time: 01:00 PM
Location: SFSU Library First Floor
Study Course: CSC510
Study Topic: Dijkstra's Algorithm

Introverts

Notifications

Meeting details with [Name]
You matched with [Name]!

[View matches](#)

[View Groups](#)

[Appointment](#)

[Reported Us](#)

ooo

About SwampStudy Profile Logout

Matching Forums

Meeting details with [Name]

Date: 04/02/24
Time: 01:00 PM
Location: SFSU Library First Floor
Study Course: CSC510
Study Topic: Dijkstra's Algorithm

Introverts

Notifications

Meeting details with [Name]
You matched with [Name]!

[View matches](#)

[View Groups](#)

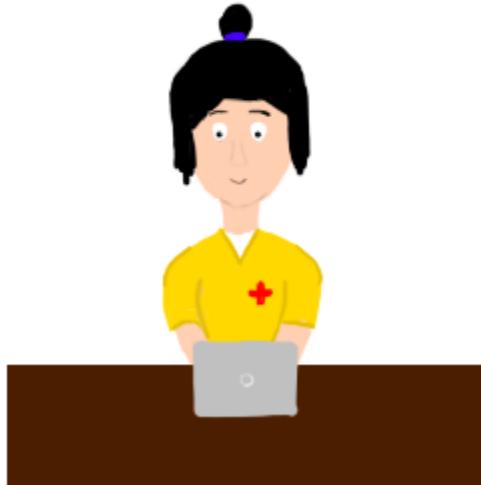
[Appointment](#)

[Reported Us](#)

Use case 8: Users ensuring the quality of matches by rating

Use Case 8 Storyboard

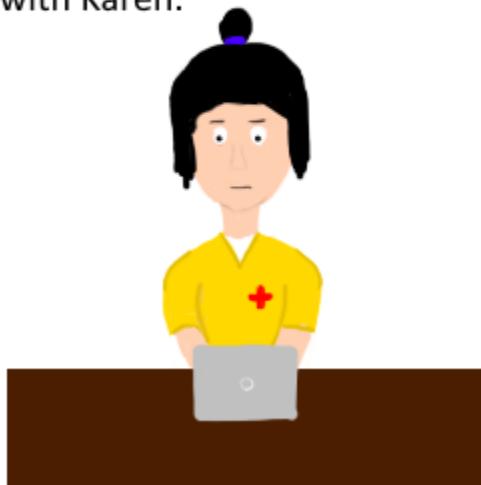
This is Michelle, she matches with Karen and they schedule a session.



Michelle is finding Karen to be intollerable and is losing patience.



Michelle is giving Karen a bad rating. She ends this session hoping no one else has to deal with Karen.



ooo

About SwampStudy Profile Logout

Matching Forums

You matched with Karen

Karen ★★



Availability:
Mon:
Tue: 1:00pm - 4:00pm
Wed:
Thu:
Fri:
Sat:
Sun:

Studying
- Biol 210

Click → Confirm Decline

Notifications

You matched with Karen

NEW Meeting details with Karen

Click



ooo

About SwampStudy Profile Logout

Matching Forums

Meeting details with Karen

Date: 04/02/24

Time: 01:00 PM - 03:00 PM

Location: SFSU Library First Floor

Study Course: Biol 210

Study Partner: Karen

Notifications

Meeting details with Karen

You matched with Karen

ooo

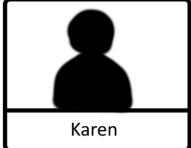
SwampStudy 

Profile Logout

Matching Forums

Rate your study partner

How would you rate your study partner?

 Karen



Reasons For Your Rating:

Karen is loud, rude, and condescending. She goes on off-tangent rants too often.
I had a terrible time studying.
I hope no one else gets matched with her.

Would you like to partner with this person again?

Yes No

Submit

Notifications

Rate your study partner

Use case 12: User getting his questions answered on the forum

Use Case 12 Storyboard



Geronimo missed the first week of an online class, he finds out there is a textbook that is required, but can't find which one is right.

He panics as the professor doesn't answer emails, and he does not have anyone else's contact information.



He tried to search the conventional places, but had no luck. Until he found the Swamp Study Website, here he was able to Ask in the class specific forum and get his answer.



ooo

About SwampStudy Profile Logout

Matching Forums

General Chat

Select Course :

math 101
Science 101
History 102
Biology 101
Calc 2
CSC 600
Business 201
Art 100

Vegeta123: Who can beat Goku? 3 7

Shaggydog-dogmen: What is your favorite smell? 4 4

Something123: What's the wifi password? 3 7

Illdiz-Tomorrow: What's your favorite soap? 4 3 5 7

HarryPotter: Do you believe in aliens? 1 7

Ask a question:

Search questions:

Click →

ooo

About SwampStudy Profile Logout

Matching Forums

General Chat

Select Course :

math 101
Science 101
History 102
Biology 101
Calc 2
CSC 600
Business 201
Art 100

Cart_Lady172: What class is this? 3 7 ← click

BigfootRept: Anyone have notes for Tuesday? 4 4

MomSpaghetti: How do I do #7 on HW5? 3 7

Ask a question:

Search questions:

)

|

{

ooo

About SwampStudy Profile Logout

Matching Forums

General Chat

Select Course :

math 101

Science 101

History 102

Biology 101

Calc 2

CSC 600

Business 201

Art 100

o cat-body172 : what class is this?

↳ Shagule - oatmeal : This is Calc 2

↳ momSpaghetti : Im not sure

Back

3 7

Red annotations:

- A red box surrounds the text "cat-body172 : what class is this?".
- Two red arrows point from the right margin towards the red box.

Database Architecture

1. Database Requirements

1.1. Registered User (Strong)

- 1.1.1. A User shall be able to make many matches
- 1.1.2. A User shall be able to post many questions
- 1.1.3. A User shall be able to answer many question
- 1.1.4. A User shall have at most one account

1.2. Account (Weak)

- 1.2.1. An Account shall belong to one and only one Registered User
- 1.2.2. An Account shall be in many matches
- 1.2.3. An Account shall have many attributes
- 1.2.4. An Account shall have one and only one profile
- 1.2.5. An Account shall have many Ratings

1.3. Match (Weak)

- 1.3.1. A Match shall contain two users
- 1.3.2. A Match shall have a class associated with it
- 1.3.3. A Match shall have one meeting
- 1.3.4. A Match shall have a meeting link
- 1.3.5. A Match shall have a meeting time

1.4. Question (Weak)

- 1.4.1. A Questions shall be posed by one Account
- 1.4.2. A Question shall have many answers
- 1.4.3. A Question shall have a class associated with it
- 1.4.4. A Question shall be in one and only one thread

1.5. Answer (Weak)

- 1.5.1. An Answer shall belong to one and only one Thread
- 1.5.2. An Answer shall have one account author
- 1.5.3. An Answer shall have a rating

1.6. Rating (Weak)

- 1.6.1. A Rating shall belong to one and only one question

1.7. Thread (Weak)

- 1.7.1. A Thread shall contain one and only one question
- 1.7.2. A Thread shall contain many answers
- 1.7.3. A Thread shall be marked with its status

1.8. Class (Strong)

- 1.8.1. A Class shall belong to many questions
- 1.8.2. A Class shall belong to many matches
- 1.8.3. A Class shall be associated with many users

1.9. Profile (Weak)

- 1.9.1. A Profile shall belong to one account
- 1.9.2. A Profile shall have many attributes

1.10. Rating (Weak)

- 1.10.1. A Rating shall belong to one and only one Account

2. Entity Attributes, Relationships, and Domains
 - 2.1. user (Strong)
 - 2.1.1. id: PK, Numeric, Generated
 - 2.1.2. email: unique, alphanumeric
 - 2.1.3. password: alphanumeric, hashed
 - 2.1.4. created: datetime, generated
 - 2.2. account (Weak)
 - 2.2.1. id: PK, Numeric, Generated
 - 2.2.2. first_name: alphanumeric
 - 2.2.3. last_name: alphanumeric
 - 2.2.4. profile_picture, upload (S3 Link)
 - 2.2.5. userFKId: FK, numeric, user_id
 - 2.3. match (Weak)
 - 2.3.1. id: PK, Numeric, Generated
 - 2.3.2. classId: FK, Numeric, class_id
 - 2.3.3. meetingTime: datetime
 - 2.3.4. meetingLink: Alphanumeric
 - 2.4. match_users_account (Weak)
 - 2.4.1. matchId: PK, Numeric, Generated
 - 2.4.2. accountId: FK, Account, user_id
 - 2.5. class (Strong)
 - 2.5.1. id: PK, INT
 - 2.5.2. name: Alphanumeric
 - 2.5.3. number: INT
 - 2.5.4. department: Alphanumeric
 - 2.6. question (Weak)
 - 2.6.1. id: PK, Numeric, Generated
 - 2.6.2. accountId: FK, Numeric, account_id
 - 2.6.3. question: alphanumeric
 - 2.7. answer (Weak)
 - 2.7.1. id: PK, Numeric, Generated
 - 2.7.2. accountId: FK, Numeric, account_id
 - 2.7.3. answer: alphanumeric
 - 2.7.4. threadId: FK, Numeric, thread_id
 - 2.8. thread (Weak)
 - 2.8.1. id: PK, Numeric, Generated
 - 2.8.2. classId: FK, Numeric class_id
 - 2.8.3. questionId: FK, Numeric, question_id
 - 2.8.4. title: alphanumeric
 - 2.9. like (Weak)
 - 2.9.1. id: PK, Numeric, Generated
 - 2.9.2. likerId: FK, Numeric, Account ID
 - 2.9.3. likedId, FK, Numeric, Account ID
 - 2.9.4. timestamp: Date time, Generated

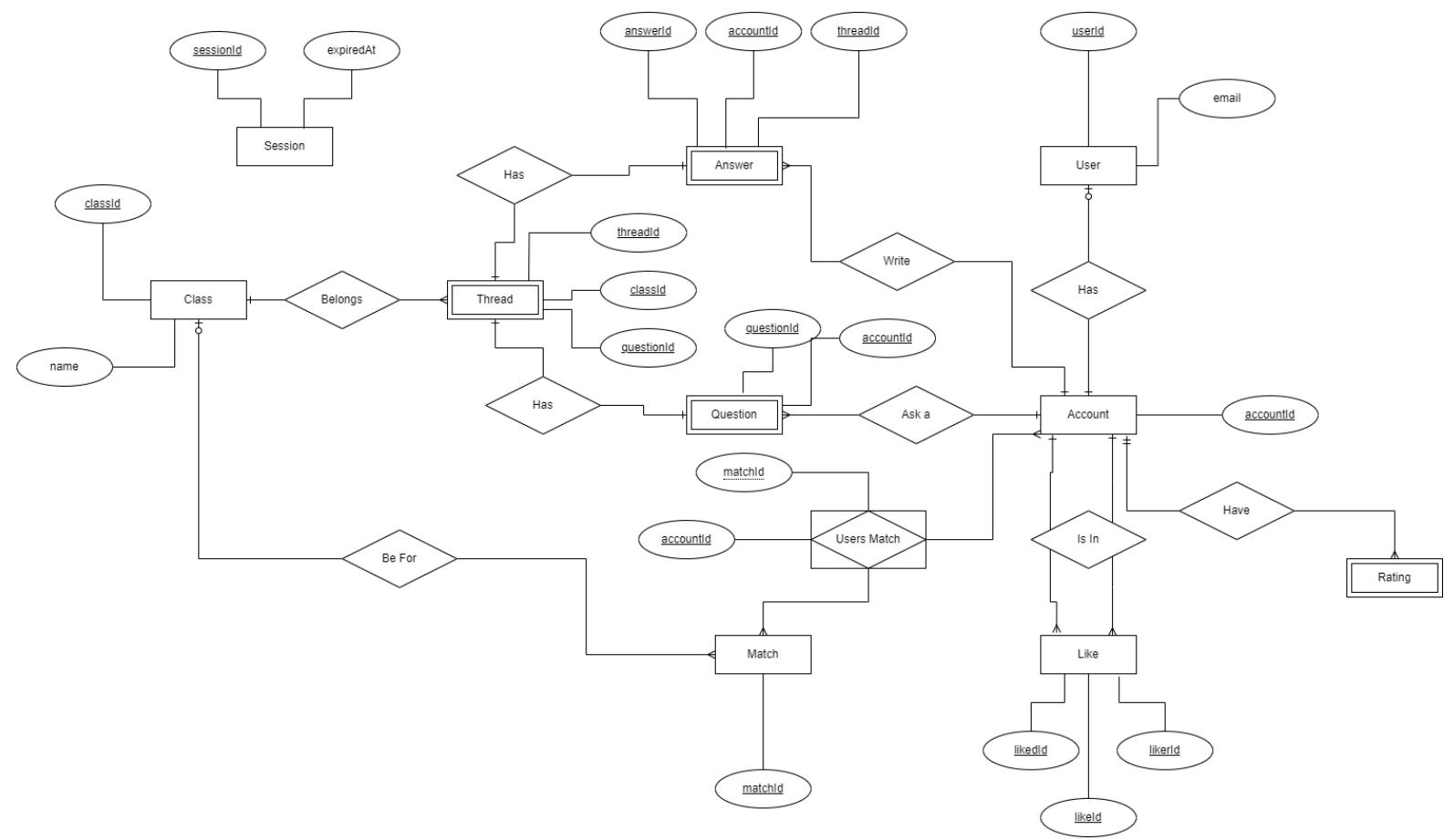
2.10. session (Strong)

- 2.10.1. id: PK, VarChar, Generated
- 2.10.2. expiredAt: Big Int
- 2.10.3. json: text
- 2.10.4. deletedAt: datetime

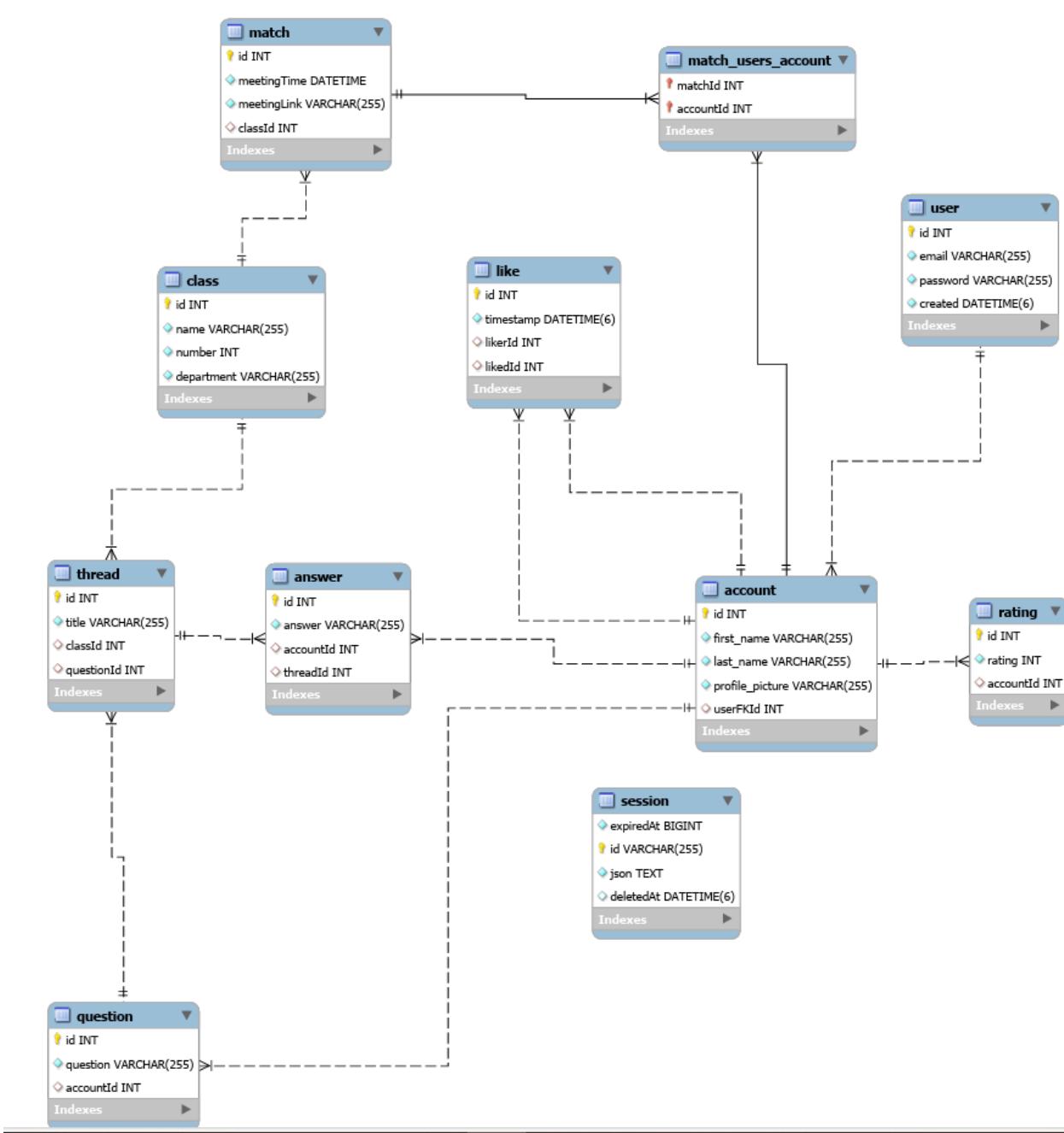
2.11. rating (weak)

- 2.11.1. id: PK, Numeric, Generated
- 2.11.2. value: Numeric

Entity Relationship Diagram



Entity Establishment Relation Diagram (EER)



DBMS System

The DBMS system we chose is MySQL. We made this decision since it's the system we have the most experience with and there are existing integrations. We also integrated

TypeORM into our application which is a TypeScript ORM library which allows us to define our database within our application code.

Media Storage

For our media storage we are using an AWS S3 bucket and storing the links to the images within our database. The integration is done using the AWS SDK which integrates into our express application.

Searching the Database

All the searching is done through the TypeORM library.

There are 2 mechanisms for searching, EntityManager, and Repository, our application utilizes the Repository feature.

```
const result = await myDataSource
  .getRepository(Account)
  .findOneBy({ id: userId as unknown as number });
```

The above example is using the Repository, which is meant for grabbing single tables/entities from the database. The one above is an example of getting an Account based on the User Id

We can also build more complex search queries by combining the Repository with the QueryBuilder

```
const matches = await myDataSource
  .getRepository(Match)
  .createQueryBuilder("match")
  .innerJoin("match.users", "account")
  .where("account.id = :userID", { userID: id })
  .getMany();
```

With the QueryBuilder we are able to join tables and build virtually any SQL query that we need.

To Optimize searching we can mark tables with relations as eager, which then automatically loads the tables marked in the relation which makes for faster and more efficient searching.

```
@ManyToMany((type) => Category, (category) => category.questions, {
  eager: true,
})
```

High Level API's and Main Algorithms

Login API:

The login API lets the existing users access their accounts by verifying their email and passwords with the database. The API checks if the email and password fields are provided and will respond if there is anything missing. Should there be no user found it will return a not found response. The API will check the database for the user name then check if the password stored matches using bcrypt's compare function adding a layer of protection. A session for the user is generated with userId.

Logout API:

The logout api handles user logout by ensuring that the session is properly destroyed and removed from the database.

Register API:

The Register API processes a new user to SwampStudy. It starts by validating the essential fields like first name, last name, and etc. If there were to be a missing field then the return will be a bad request response. The API will check for existing users in the database. When all the checks are successful it will hash the password, and create a new user entity with corresponding account entity.

Match API

The match API retrieves a given user's matches and returns the information for the match in JSON format. The API returns an array of a user's matches or an empty array

if there are none. On error it returns a negative status code along with a response message.

Like API

The Like API sends a request to create a like, based on the current user and the profile the current user wishes to like. Requests are sent with the two profiles, the liker and the liked. Once the request is sent in the background the matching algorithm determines whether there is a match. On success returns a positive status code and response message. On failure returns a negative error code and error message.

Forum API:

The forum API facilitates the creation, retrieval, and management of questions, answers, and discussion threads within the forum. It supports operations such as posting new questions and answers to questions, fetching questions along with their corresponding answers, and organizing discussions by classes and departments. This ensures a structured and interactive environment for users to engage in topic specific discussions, leveraging data integrity and validation to maintain a robust platform.

Like and Match Algorithm

The Matching algorithm gets run every time a user likes another user. As soon as this action takes place, the algorithm checks if there is a reciprocal like in the system. If that exists, a match is formed and put into the system for the user to interact with. If not the like gets inserted and can be used to create a potential match if the other person ever likes this user.

Session Algorithm

The session management algorithm in SwampStudy encompasses session creation , validation, and destruction across both frontend and backend areas. This begins at user login, where existing users pass an email, and password to the backend. When the backend receives the request to login, it is routed to the user controller, where in the email is used to query for the corresponding users information stored in the db provided that the password passes a comparison check using bcrypts compare function against the stored hash. This user information is then used to set key session data used for validation at different junctions of our web application. In our backend, sensitive routes are protected using a session validation middleware, which ensures the session is valid by checking that the appropriate data is present in the session. This middleware is attached to the request and calls next() upon successful validation of the session indicating to continue to the next function or it'll return an error response indicating the session is invalid. At different points in the web application, the frontend will check that a session is authenticated to determine whether to grant or prevent a user from accessing certain site resources and or updating the appearance of the UI dynamically. This check for session validity is primarily handled by the authentication controller, which sends back a status upon checking that the session is populated with the correct data. This data can only be accessed from the db itself and is set at the login phase.

Search Algorithm

The Search algorithm is implemented using the ORM database connection. In order to search for Forum Posts using title keywords, we take the Keyword placed into the

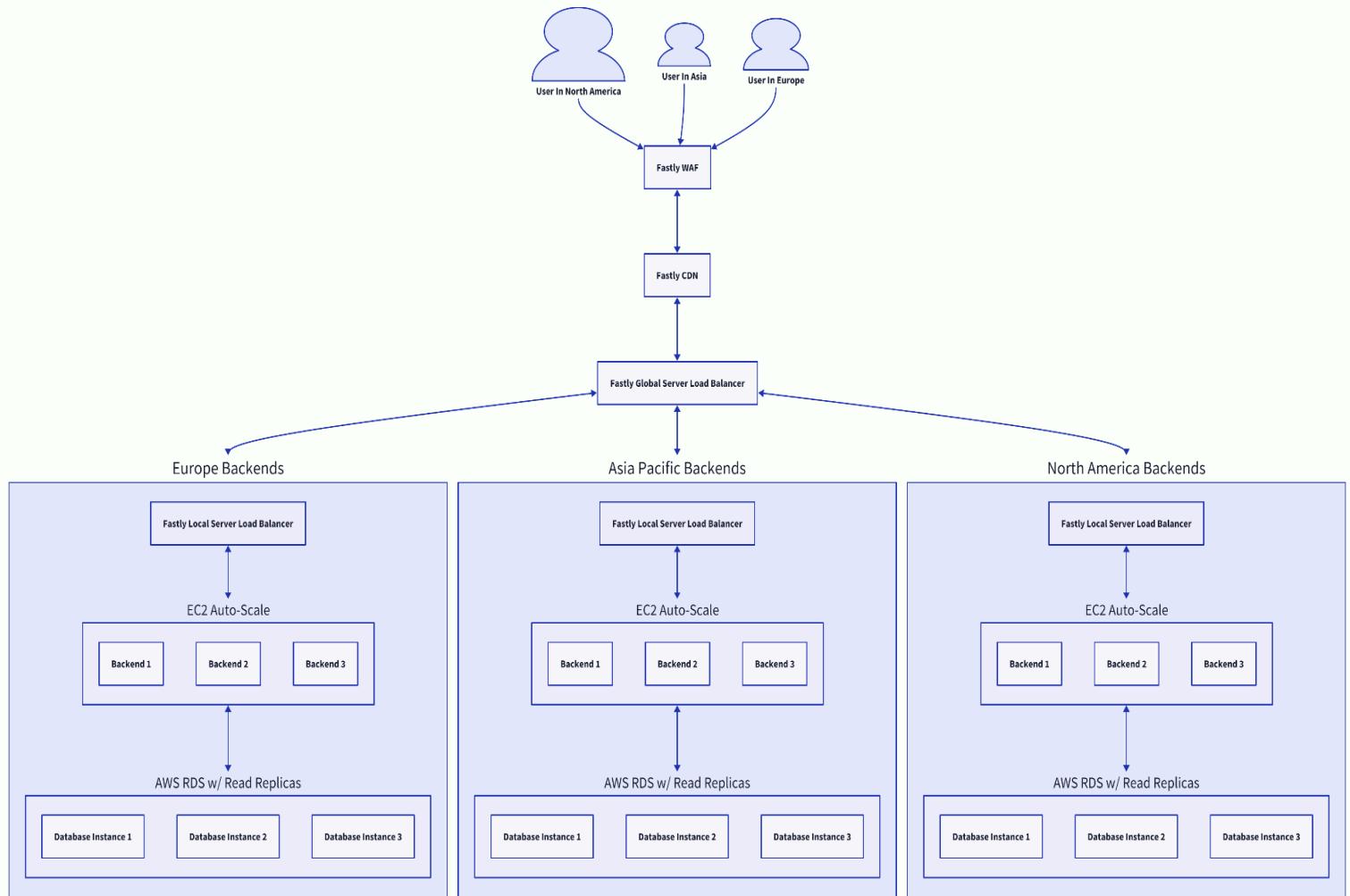
search box and then use a query with the ‘where’ keyword and regex wildcards to determine which posts fit the given criteria and then filter and display the results based on that keyword. We also scan the answers of each question as well in order to give a more comprehensive search. What gets returned are any threads with question/titles/answers that contain the requested keywords. This gives the customers a more comprehensive result given that it also takes into account the answers provided by other users.

System Design

UML Class Diagram:

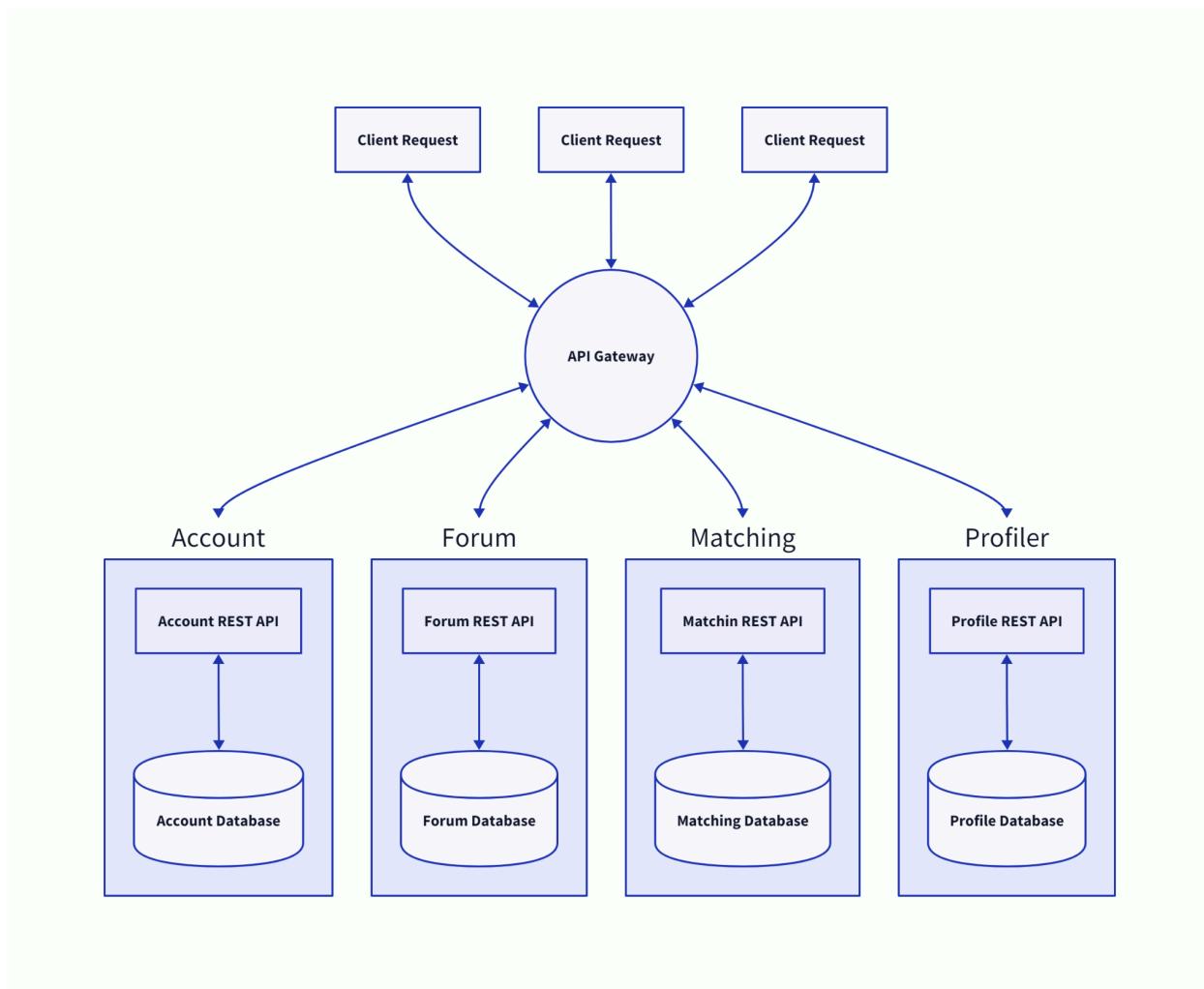
See the end of the Document for full size diagram

Scalability Diagram



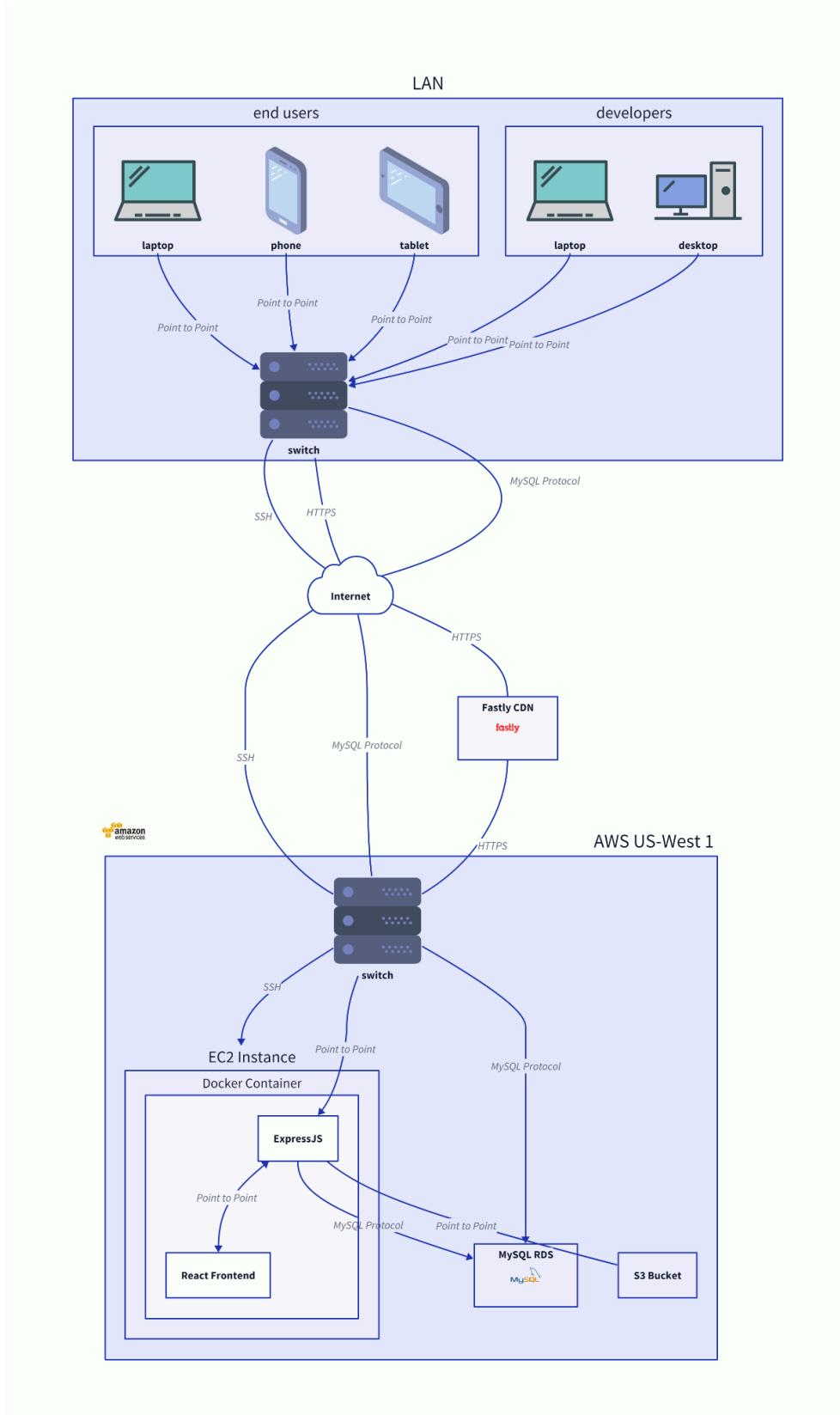
To let our service scale with ease, we first introduce a WAF at the top of the chain. This will allow us to filter malicious requests and bots trying to gain access to the system. We then run all the requests through our CDN which allows for request collapsing and to deliver cached content far quicker from the edge. After that requests hit the global load balancer which decides which server is the fastest and most responsive to handle that traffic. After being sent to a local region there is another load balancer which checks the availability of the local servers and determines the most appropriate server. The servers then have multiple Read Replicas of the Database they can access in order to make the reading quicker. All the servers/EC2 instances have a Auto-Scale setup which means if the traffic ever exceeds the current resources more will be created to handle with the increased demand.

Microservices Diagram

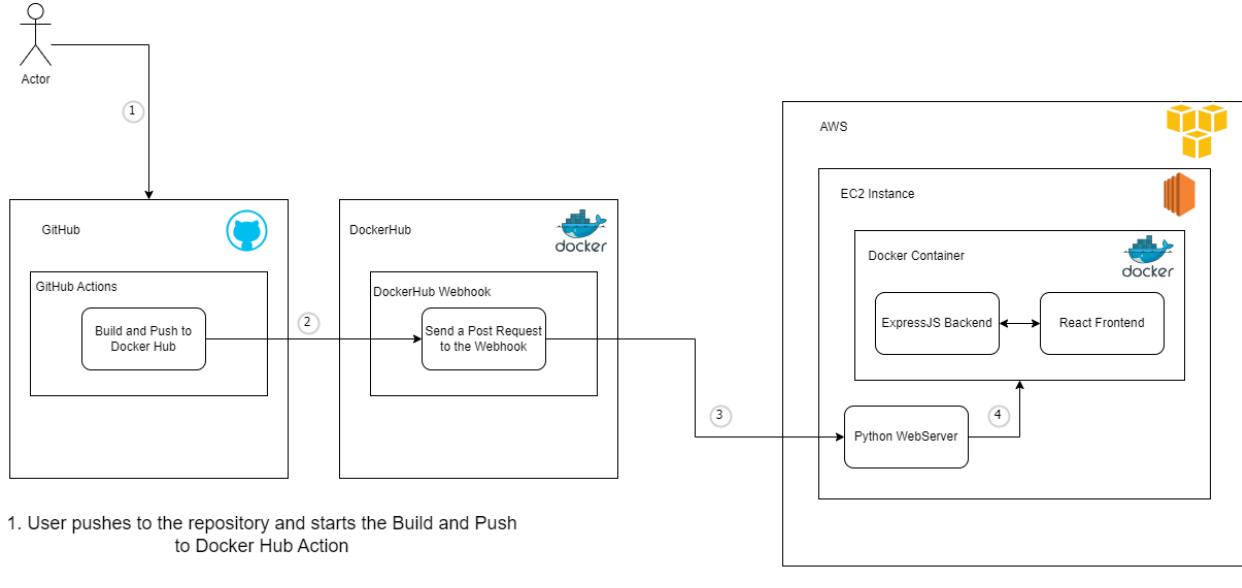


High Level Application Network and Deployment Design

Network Diagram



Deployment Diagram



1. User pushes to the repository and starts the Build and Push to Docker Hub Action

2. DockerHub notices a new push and initiates a post request sent to the webhook

3. Python WebServer receives the Post request notifying of a new version of the application

4. Python WebServer downloads the new Docker image, and then restarts the application with the new image

Key Risks

Skills Risks:

Even though we chose a Tech Stack we were all mostly comfortable with. Working across a Full Stack project with many integrations and interactions is still a challenge at times. We have been taking these skills risks head on by discussing frequently on Discord meetings in person and making videos to explain tricky pieces of code and integrations that might not be 100% intuitive.

Schedule Risks:

With 7 weeks remaining in the Semester we will be definitely pushing the limits with our full suite of requirements. I have full faith in our ability to get the application into a workable state. One of our risks also includes the passion of the team to add more features and ideas that pop up along the way. We have to be very selective in admitting new ideas and they have to be fully thought out to gain admission. All team members have also been very proactive to make meetings and complete their work which gives us the best chance of finishing this project in a timely manner.

Project Management

For Project Management we have been using Notion as our tool of choice. Using Notion has allowed us to create a new sub project for each Milestone. By doing this we are able to see at a glance what our progress is and where we need to make more progress. Every team meeting we go through our Notion Board/Tasks and evaluate deadlines and priorities, shifting as needed. Each task in our Notion has an assigned contributor and deadline. For our future Milestones we are going to continue using Notion, so far it has served us very well and allowed us to have a great overview with fine detail control. We also have internal documentation within Notion that everyone can contribute to which allows us to have a centralized location for all our notes. Along with Notion we use Discord for our team communication and quick exchanges. And Zoom as our meeting platform where we host our weekly team meetings, as well as other shorter brainstorming sessions.

Team Evaluation

Lennart (Team Lead) 10

- Project Planning
- Developed Backend API for Matching
- Developed Backend API for Likes
- Various Bug Fixes
- Network Diagram
- Deployment Diagram
- Scalability Diagram
- Microservice Diagram
- Attended all meetings
- Responsive on Discord

Rafael (Backend Lead) 10

- Session API
- Session Implementation
- Bug Fixes in Forum
- Assisted in Front-End Development
- The Connection between front and back end
- Session API Description
- Reviewed Diagrams for the Document
- Bug Fixes on the front end
- Implemented the Search bar and algorithm
- Attended all meetings
- Responsive on Discord

Edmund (Frontend Lead) 10

- Drew Mockups
- Drew Storyboards
- Login Component
- Logout Component
- Registration Form
- Terms and Conditions
- Attended all meetings
- Worked with UX Researcher
- Responsive on Discord

Julio (Docs-Editor) 10

- Drew Mockups
- Drew Storyboards
- Aided in organization of Document
- M1V2 Submission
- Nav Bar on Website
- Registration Form on Website
- Forum implementation on Website
- Meeting Summaries
- Attended all meeting
- Worked with UX Researcher
- Responsive on Discord
- Reserved Library rooms

Conrad (Github-Admin) 10

- Managed GitHub Conflicts
- UML Diagram for Document
- API Description for Forum
- API Implementation for Forum
- Forum Entities
- Reviewed and improved all Entities
- Attended all meeting
- Responsive on Discord

