

SW Engineering CSC648-848-05 Spring 2024

Application Name: **Swamp Study**

Seal Team One Members (Team 01)

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

Project URL: <https://swamp-study.global.ssl.fastly.net/>

Video URL: <https://youtu.be/vUsOvnIQRsl>

Version Table

Milestone 1	Feb 28, 2024
Milestone 1 V2	Apr 1, 2024
Milestone 2	Apr 4, 2024
Milestone 2 V2	Apr 24, 2024
Milestone 3	Apr 24, 2024
Milestone 3 V2	May 15, 2024
Milestone 4	May 15, 2024
Milestone 5	May 23, 2024

Table of Contents

SW Engineering CSC648-848-05 Spring 2024.....	1
Table of Contents.....	2
Product Summary.....	4
Committed Application Functions.....	4
Summary/Unique Brag.....	5
Post Analysis.....	6
Team Evaluation.....	7
Lennart (Team Lead) 8.....	7
Rafael (Backend Lead) 9.....	7
Edmund (Frontend Lead) 8.....	8
Julio (Docs-Editor) 7.5.....	8
Conrad (Github-Admin) 8.....	9

Product Summary

Name of the product: Swamp Study

URL: <https://swamp-study.global.ssl.fastly.net/>

Committed Application Functions

1. User

- 1.1. A User shall be able to create an account
- 1.2. A User shall only have one account
- 1.3. A User shall be able to view other users' accounts in the system
- 1.4. A User shall be able to set their availability
- 1.5. A User shall be able to establish their profile
- 1.6. A User shall be able to ask many questions in the Forum
- 1.7. A User shall be able to answer many questions in the Forum
- 1.8. A User shall be able to like other Users
- 1.9. A User shall be able to match with other Users

2. Profile

- 2.1. A Profile shall be owned and edited by one and only one user
- 2.2. A Profile shall be allowed one picture of the owner
- 2.3. A Profile shall be allowed a short biography
- 2.4. A Profile shall have many subject attributes
- 2.5. A Profile shall be able to set a filter for their study partner preferences
- 2.6. A Profile shall be marked Student or Educator
- 2.7. A Profile marked Educator shall be able to access the Forum
- 2.8. A Profile marked Educator shall be allowed to answer questions in the Forum

3. Account

- 3.1. An Account shall have one email associated with it

4. User Interface

- 4.1. The Interface shall allow a User to navigate through a potential Match's profile
- 4.2. The Interface shall allow a User to access/edit their Profile
- 4.3. The Interface shall allow a User to access/edit their Account

5. Match

- 5.1. A Match shall be provided a meeting link
- 5.2. A Match shall be provided a meeting time

5.3. A Match shall be made based on different criteria

6. Forum

- 6.1. The Forum shall allow one main question per Thread
- 6.2. The Forum shall allow each class its section
- 6.3. The Forum shall allow users to post in their class Forum
- 6.4. The Forum shall allow many questions

Summary/Unique Brag

The crowded field of educational applications is one of little breadth. With the COVID pandemic and teachers' reliance on online tools, the educational process is more intertwined with technology than ever before. The "Legacy" applications in the study space focus on providing a platform in which to ask and answer homework questions. This has led to a rise of unethical behavior to the educational process. These platforms want to sell a subscription that lets you get homework answers quickly for a price. This takes out the most important aspect of education, the process of learning and collaboration.

Swamp Study aims to be the leader in bringing collaborative learning to the online space. A common quandary faced by college students and those of all ages is the process of finding a study partner, someone with whom to share and enhance the learning experience. This is where Swamp Study comes in, we provide a state of the art matching system to give students the access to study partners at their learning institution. We provide users with their most likely matches first ensuring that you get quality partners which will lead to better grades. By letting the students put in their class schedule and availability we can make sure that you'll find people with commonalities that will surely make the study experience an enjoyable one.

Milestone Documents

SW Engineering CSC648-848-05 Spring 2024

Application Name: **Swamp Study**

Seal Team One Members (Team 01)

Lennart Richter <u>lrichter@sfsu.edu</u>	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	Mar 31, 2024

Table of Contents

SW Engineering CSC648-848-05 Spring 2024.....	1
Table of Contents.....	2
Executive Summary.....	4
Use Cases.....	5
Use case 1: Base - need partner.....	5
Use case 2: Notifications.....	7
Use case 3: Availability.....	9
Use case 4: Safety.....	11
Use case 5: Long-Term.....	13
Use case 6: Group matches.....	15
Use case 7: Introverts.....	17
Use case 8: Rating.....	19
Use case 9: Monetization/ Premium.....	21
Use case 10: Reward program.....	23
Use case 11: Educator.....	25
Use case 12: Forum.....	27
Use case 13: Quality Control.....	29
Data Items and Entities.....	31
User.....	31
Premium user.....	31
Matched Pair.....	31
Introvert.....	31
Blocked user.....	31
Educator.....	31
Tutor.....	31
Group Cluster.....	31
Profile.....	31
Rating.....	31
Availability.....	31
Proficient Subject.....	31
Challenged Subject.....	32
Forum.....	32
Reward.....	32
Warned User.....	32
Account.....	32
Application.....	32
Call center.....	32
Video call center.....	32
Appointment center.....	32
Functional Requirements.....	33

User Account Management:.....	33
User Profile Management:.....	33
User Interface:.....	33
Matching System:.....	33
Forum:.....	34
Additional Features:.....	34
Non-Functional Requirements.....	35
1. Usability.....	35
2. Reliability.....	35
3. Scalability.....	35
4. Maintainability.....	35
5. Utility.....	35
6. Performance.....	35
7. Privacy.....	35
8. Data Integrity.....	36
9. Security.....	36
10. Storage.....	36
11. Deployment.....	36
12. Coding Standards.....	36
13. Environmental Sustainability.....	36
Competitive Analysis.....	37
Competitive Analysis Table.....	37
Competitive analysis Cont.....	39
Competitive Features Table:.....	40
Summary.....	41
Technical Architecture.....	42
1. Server Host:.....	43
2. Operating System.....	43
3. Database.....	43
4. Web Server.....	43
5. CDN.....	43
6. Server Side Language.....	43
7. Additional Technologies.....	43
8. Note on Technical Architecture Change.....	43
Checklist.....	44
Team Evaluation.....	45
Lennart (Team Lead) 10.....	45
Rafael (Backend Lead) 10.....	45
Edmund (Frontend Lead) 10.....	45
Julio (Docs-Editor) 10.....	46
Conrad (Github-Admin) 10.....	46

Executive Summary

SwampStudy came about solving the question: “How do you find people to study with in the world of hybrid schooling and diminished social interaction?” Finding a study partner can be a daunting task than it otherwise should be. Social interaction is a foundational component in fostering learning by being a means for people to exchange what they know, explore new perspectives, and create a sense of community among the students. SwampStudy can provide a platform for students to exchange information in an open forum and provide a space for meaningful interaction between students. We believe that this can revolutionize how students form partnerships with each other and build tight-knit campus communities.

SwampStudy will look similar to other platforms geared towards social connection. There will be a tab where you can swipe to find potential study partners and another feature will be an open forum for inter-student discussion.

When creating a profile, users can set preferences by tuning the partner, finding functionality by tailoring their potential study partners by area of study and course. Some of the things users will be able to tune are the subject/specific class they're interested in, what kind of studying they would like to do, and even how much conversation they want while studying. This way, users can fine-tune and streamline the process of finding compatible study partners that fulfill their needs and studying styles. The Forum section will be broken up into classes. Users can ask questions in each class's forum based on the classes they take. This enables students to share knowledge even when they don't have the same professor, effectively casting a wider net on the knowledge that students can gain.

Use Cases

Use case 1: Base - need partner

Actors: Sebastian (Student), Potential Match (Student), SwampStudy (Company)

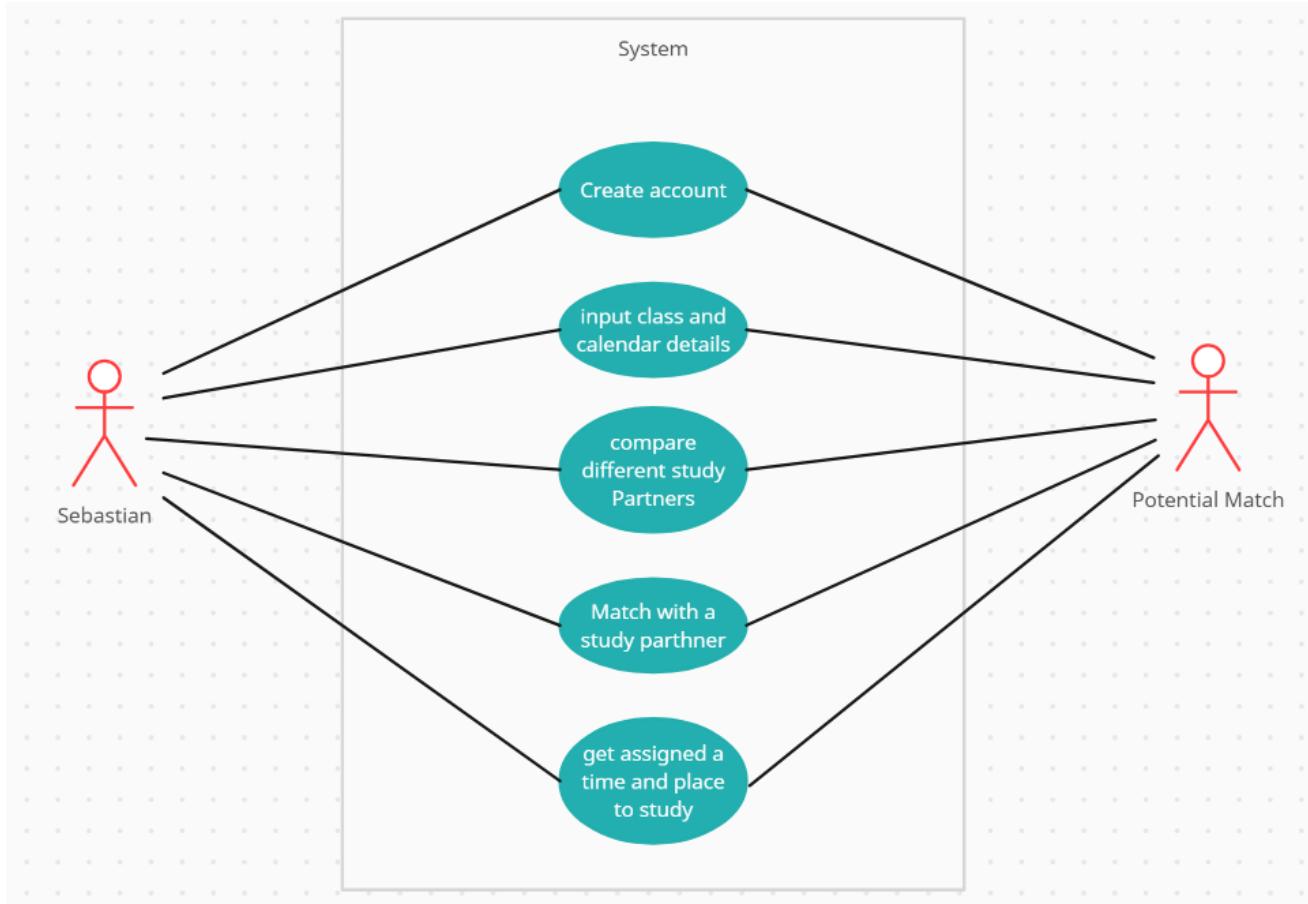
Assumptions:

- The student has access to the internet
- The student has an email

Use Case: Sebastian is a student at SFSU. Midterms are coming up and he's very worried about his grades and stressed about the pressure he'll face during the tests. He wants to find someone to study with, but his class being a lecture, he doesn't have the chance to connect with people and find someone to study with. Sebastian hears about SwampStudy and decides to give it a try. After creating an account and deciding which class he would like a study partner for, he received an array of people studying for the same class or in a similar subject at SFSU. He's relieved to see like-minded people studying the same subject. After swiping on a few people he receives a match. The platform automatically assigns a meeting place and time with a calendar link for him to add. After adding it to his Calendar, Sebastian feels much less worried knowing he has someone to study with now at a determined time. All the stressful parts of finding someone to study with have been taken out of his hands.

Benefits Sebastian received:

- **Time-saving:** He found a partner from his own home
- **Convenience:** Not a lot of his time was spent on finding a partner
- **Online platform:** Bypasses awkwardness of being rejected in person
- **Networking:** Users find new friends



Use case 2: Notifications

Actors: Cynthia (Student), Potential Match (Student), StudySwamp (Company)

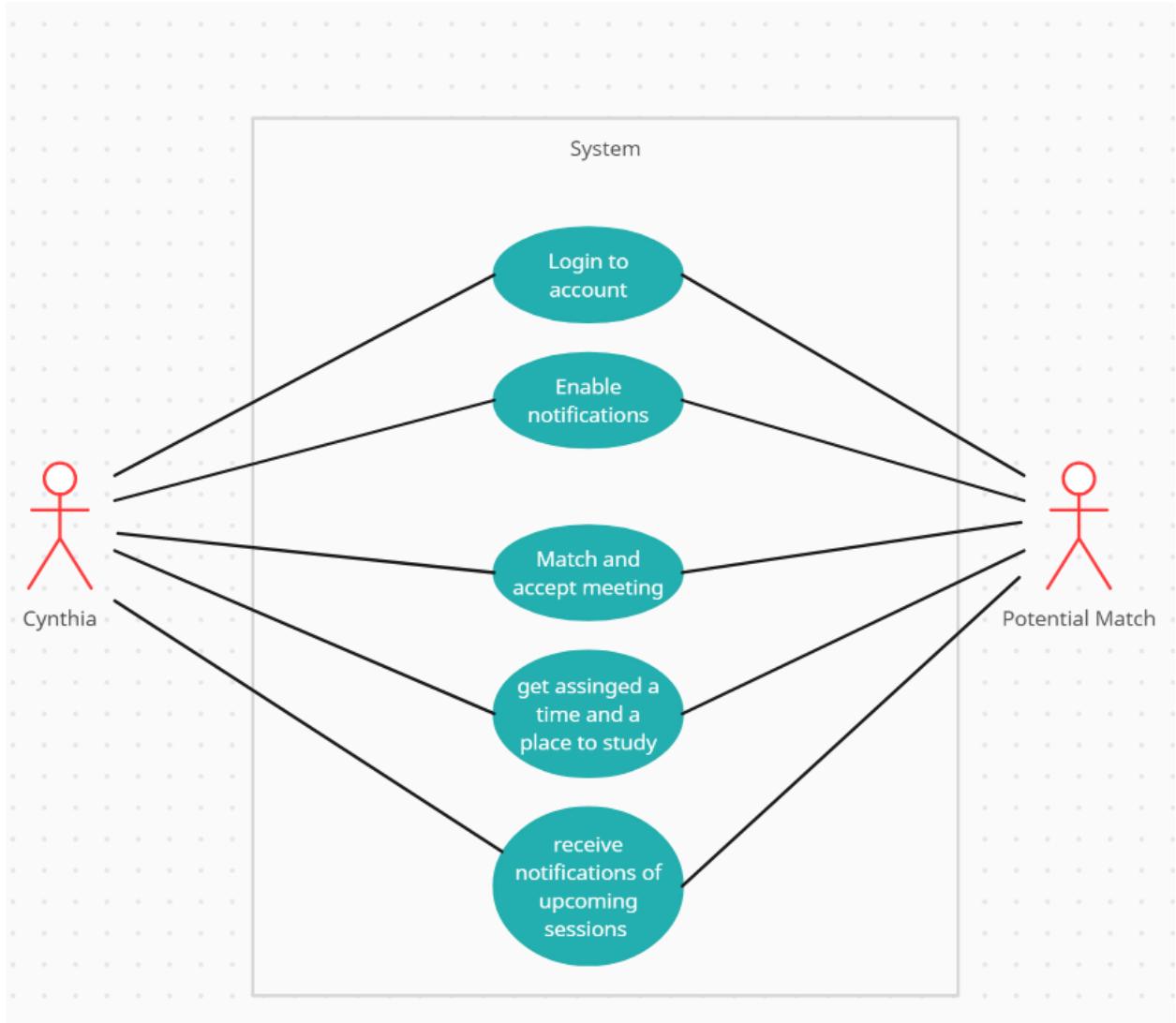
Assumptions:

- Student has a StudySwamp account
- Student has notifications turned on

Use Case: Cynthia is a very forgetful student at SFSU, her grades are suffering as she keeps forgetting deadlines and forgets to put time aside to study. She finally receives a match on the StudySwamp app, and although happy she hopes she doesn't let her partner down as this might be her last chance to pass her classes. Going about her week she finds an email reminding her about the upcoming meeting. She realized that without the notification she would have completely forgotten about it. She is thankful for the feature and makes it to the meeting on time.

Benefits Cynthia received:

- **Reminders:** The system is able to give Cynthia email/text/phone pop-up notifications
- **Peace of mind:** Cynthia's forgetful nature was not a factor anymore



Use case 3: Availability

Actors: Richard (Student), Potential Match (Student), StudySwamp (Company)

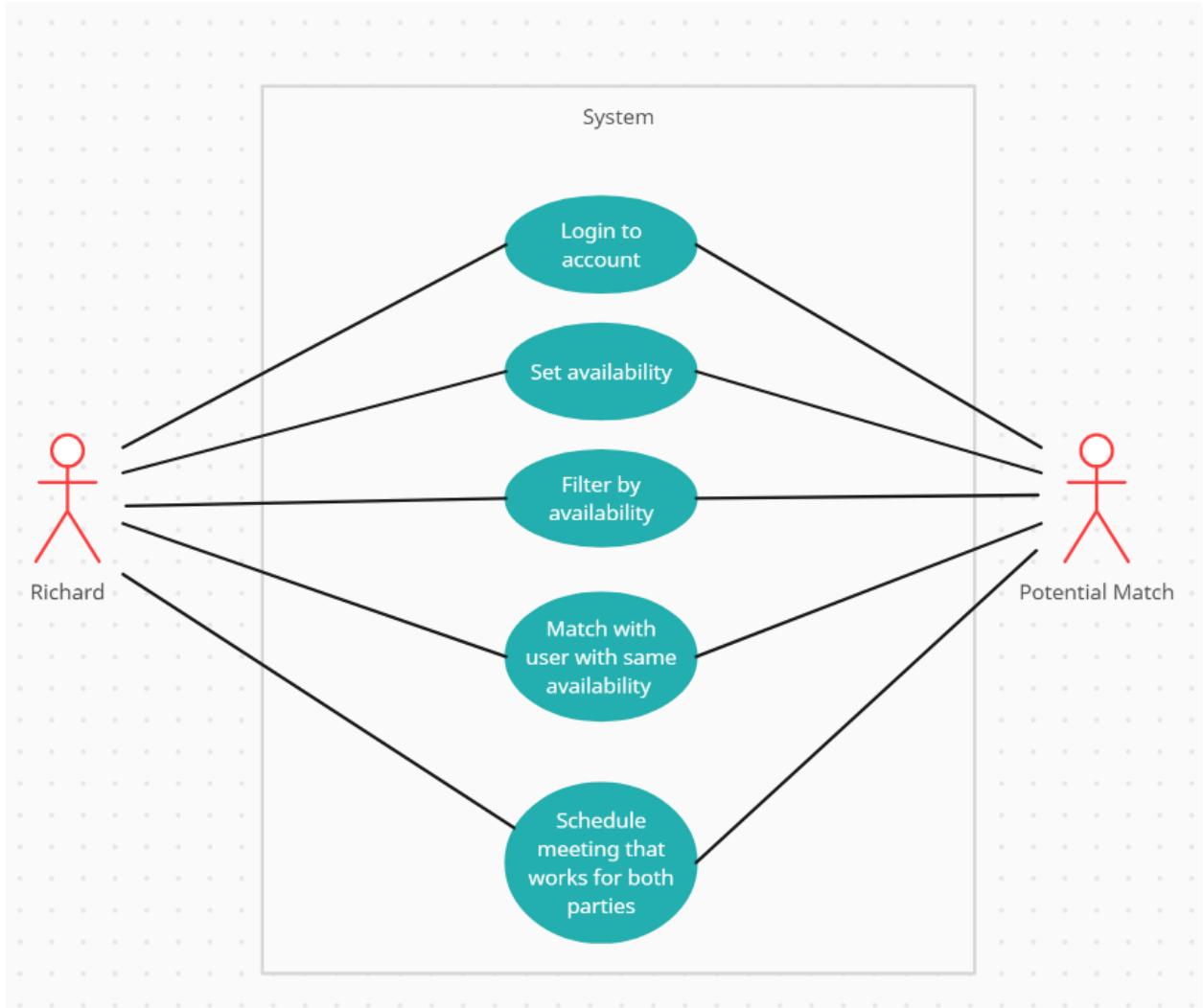
Assumptions:

- Student has a StudySwamp account
- Student sets their availability

Use Case: Richard is a non-traditional student. He has a wife and daughter at home. He is a full-time student and has full-time work. He is drowning in responsibilities and his grades are suffering. He is determined to graduate by any means so he swallows his pride and asks for help, he is referred to use StudySwamp to find someone to help him study. As his schedule is packed he doesn't seem too hopeful. Upon downloading the app he finds that there is a section where he puts out his entire availability. Although the window for his free time is small, the system is able to find a student that matches his free hours. Pleasantly surprised, Richard is able to find a new study partner without sacrificing time away from his work or family.

Benefits Richard received:

- **The system works within his hours:** The system only matches with partners that also have his free time open
- **He maintains family and work balance:** Because he did not have to change his schedule around he was able to continue his prior commitments



Use case 4: Safety

Actors: Melanie (Student), Potential Match (Student), StudySwamp (Company)

Assumptions:

- Matches have video calls available on their device

Use Case: Melanie, a student at SFSU, struggles with maintaining focus while studying alone. She is stressed about finding a compatible study partner due to scheduling conflicts. Upon a friend's recommendation, Melanie discovers StudySwamp. Intrigued, she signs up, completes her profile, and specifies her study preferences. Initially wary about safety, Melanie is reassured by SwampStudy's video call feature, which allows her to meet potential study partners virtually before in-person meetings. After selecting a study partner, Melanie joins a video call to get to know them. Once comfortable, she is automatically assigned a convenient time and place on campus to meet to study. Melanie looks forward to productive study sessions with her new partner.

Benefits Melanie received:

- **Time Savings:** StudySwamp eliminates the need for Melanie to search for a study partner, matching her with someone who shares her study topic and availability.
- **Reduced Stress:** StudySwamp's automated scheduling feature alleviates the stress of coordinating meeting times and places.
- **Enhanced Safety:** Melanie feels secure using StudySwamp's video call feature and vetting process for in-person meetings.



Use case 5: Long-Term

Actors: Daniel (Student), Emily(Student), StudySwamp (Company)

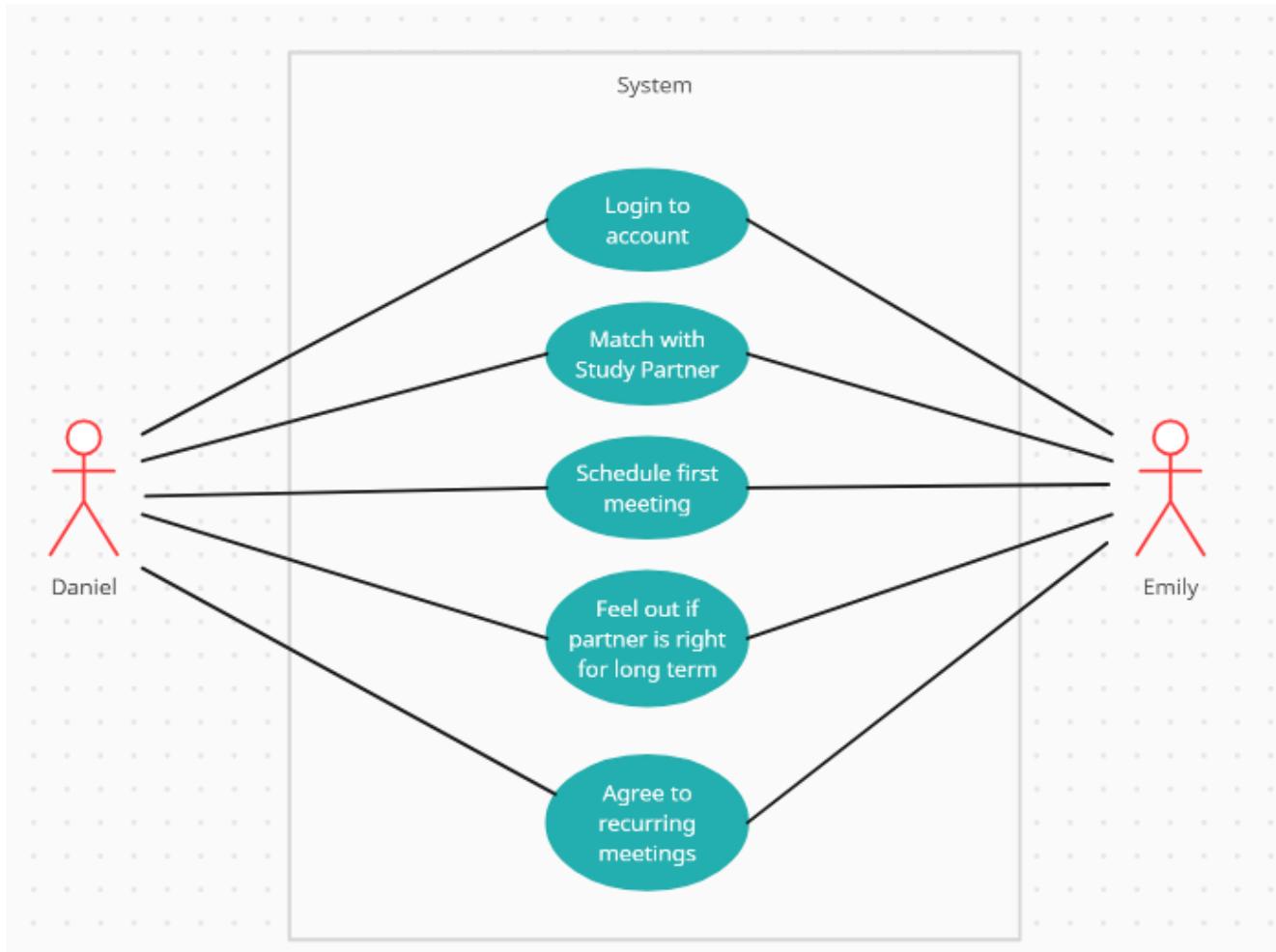
Assumptions:

- StudySwamp requires students to sign up using their student email addresses.
- All participants are students at the same university.

Daniel is a student at SFSU majoring in Computer Science. He is struggling with his Algorithms course. He wishes to find a study partner but he has no time due to his packed schedule. Daniel heard about StudySwamp from a friend and decided to give it a try. He sees a recommendation for a student named Emily, who is also taking Algorithms and has availability that matches his. Daniel and Emily decide to video call to get to know each other before meeting in person. They discuss their study habits, goals for the course, and availability. They find that they have a lot in common and agree to meet up to study together. After the video call, Daniel and Emily use StudySwamp to schedule their study session. StudySwamp suggests a time and place based on their availability and location. They agreed to meet at the campus library on Saturday afternoon. Daniel and Emily meet at the library as scheduled. They find that they work well together and can help each other understand the course material better. At the end of the meeting, the StudySwamp application asks both of them if they would like to schedule their next meeting. Seeing how they got along so well, they decide to continue studying together for the rest of the semester.

Benefits Daniel and Emily received:

- **Time-Saving:** Using StudySwamp, Daniel was able to save time that would be used to find a study buddy traditionally.
- **Safe and Convenient Matching:** StudySwamp's video call feature allowed Daniel and Emily to meet and get to know each other safely and conveniently before meeting in person. This ensured that they were comfortable with their study partner before committing to study together.
- **Ease of Convenience:** Neither had to actively look for a study partner that matches topics and schedules.



Use case 6: Group matches

Actors: Kyle(Student), Ike (Student), Cartman(Student), StudySwamp (Company)

Assumptions:

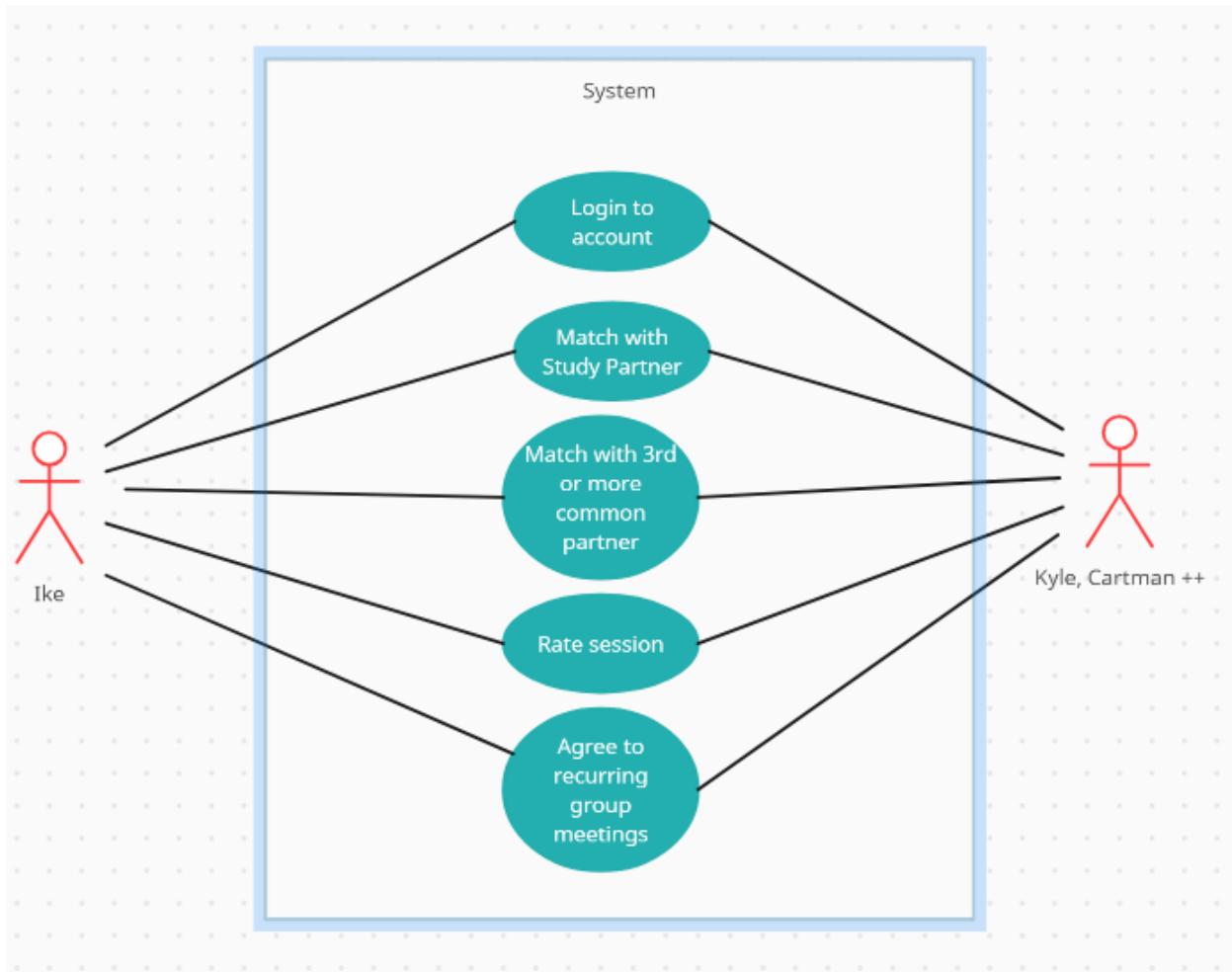
- All students have an active StudySwamp account

Use Case: Kyle is a very outgoing individual. He hates to be alone, and thrives in a social setting. He usually skips studying to go hang with his buddies. Like many other users he tries StudySwamp so that if he has to study, he at least won't have to study alone. He finds the first couple of sessions interesting but not enough to keep him from going out and neglecting his studies. One day looking at the app's settings, he finds that he can opt in to "study groups". Curious by this, he enables it. Later he finds himself matched with 2 people he had previously swiped yes with. He learns that all 3 of them had swiped yes on each other and the system had created a 3-person group with himself, Ike, and Cartman. Before their meeting, a fourth person joins their group. Kyle is excited to attend this meeting. He is able to socialize like he wanted to and learn his class material at the same time.

Benefits Kyle, Ike, and Cartman received:

Social: all group members got to experience a social environment that was still geared toward education

Increased Engagement: this style of studying goes well with these users personalities and allows for more positive engagement.



Use case 7: Introverts

Actors: Taylor (Student), Travis (Student), StudySwamp (Company)

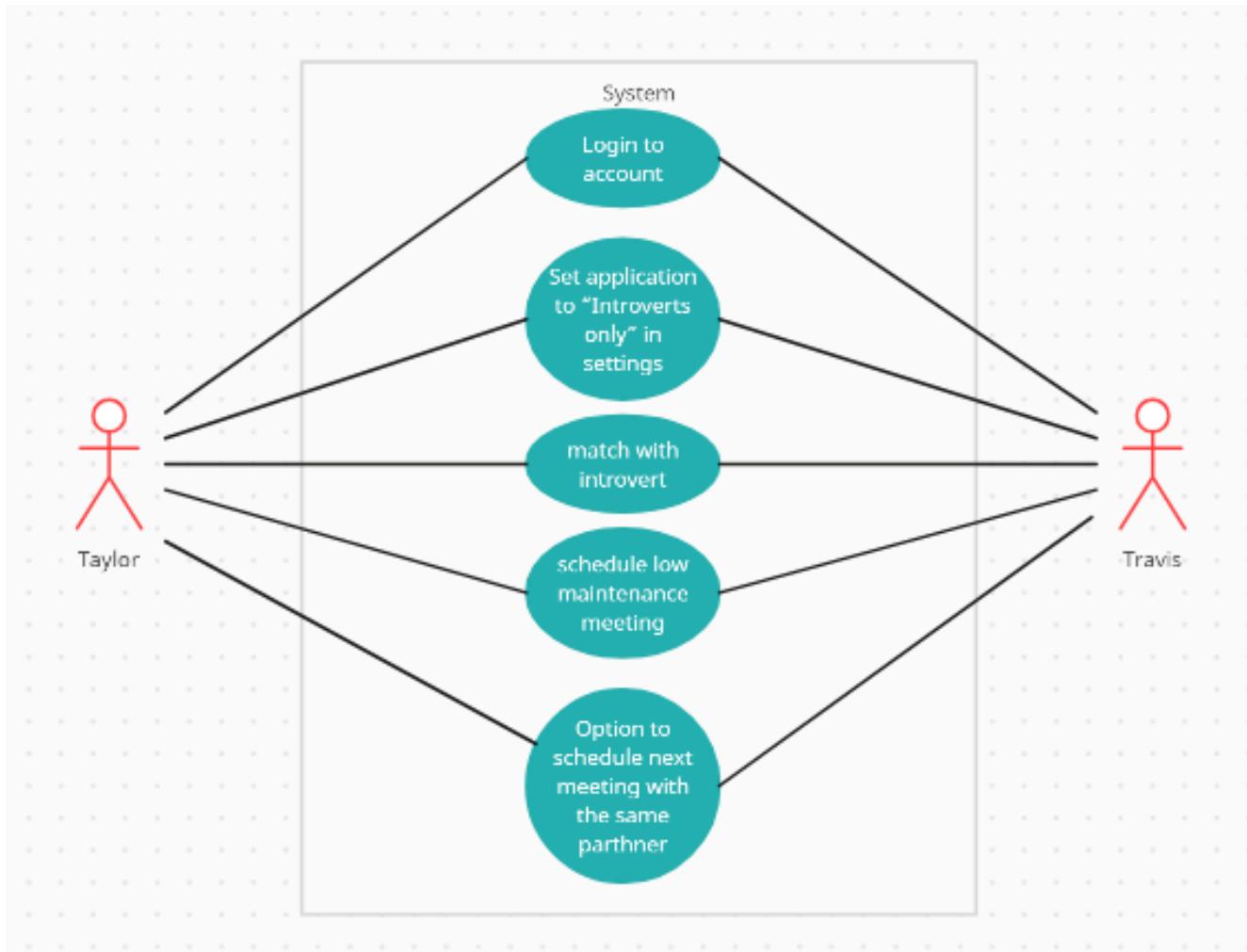
Assumptions:

- All students have an active StudySwamp account

Use Case: Taylor and Travis are both shy individuals. They both want to succeed in college. They are doing ok on their own, but acknowledge that with outside help they could be even better. The problem is they are too shy, they hate having to answer personal questions when meeting new people, so they tend to keep to themselves. With some resistance, they both finally start swiping on the StudySwamp app they had installed. They were getting no matches until they realized they had placed their account settings to “introvert”, they looked it up and the system will only match introverts with other introverts. They were about to switch to the general population when they matched with each other. Meeting up they had a funny story to tell each other and were at ease since they knew already that the other just wanted to primarily focus on studying.

Benefits Taylor and Travis received:

- **Safe space:** Neither Taylor nor Travis felt any pressure to share personal information or other topics that did not correlate to their studies.
- **Common ground:** both users went in knowing each others boundaries and allowed for an easier more relaxed experience.



Use case 8: Rating

Actors: Michelle(Student), Karen(Student), StudySwamp (Company)

Assumptions:

- All students have an active StudySwamp account

Use Case: Michelle is the nicest person at school, she tries to befriend everyone and tries not to take anything personally. She is matched with Karen for their upcoming nursing midterm. Michelle finds Karen to be a complete slob, rude and condescending. Michelle tries to work with her but Karen keeps going into political and racist tangents. Michelle's patience is running very low but is saved by the conclusion of the meeting. Like the Uber app, a rating system pops up asking Michelle how the meeting went. Against her good nature she not only rates Karen 1 star but also sends a report to StudySwamp on everything that happened. StudySwamp thanks her for her feedback and Michelle leaves hoping no one else has to deal with Karen anymore.

Benefits Michelle received:

- **Peace of mind:** By rating Karen low, she will be confident that other students likely will not match with the awful Karen.
- **No Retaliation:** Reports are anonymous and have the option to block users



Use case 9: Monetization/ Premium

Actor: Erika (Student), StudySwamp (Company)

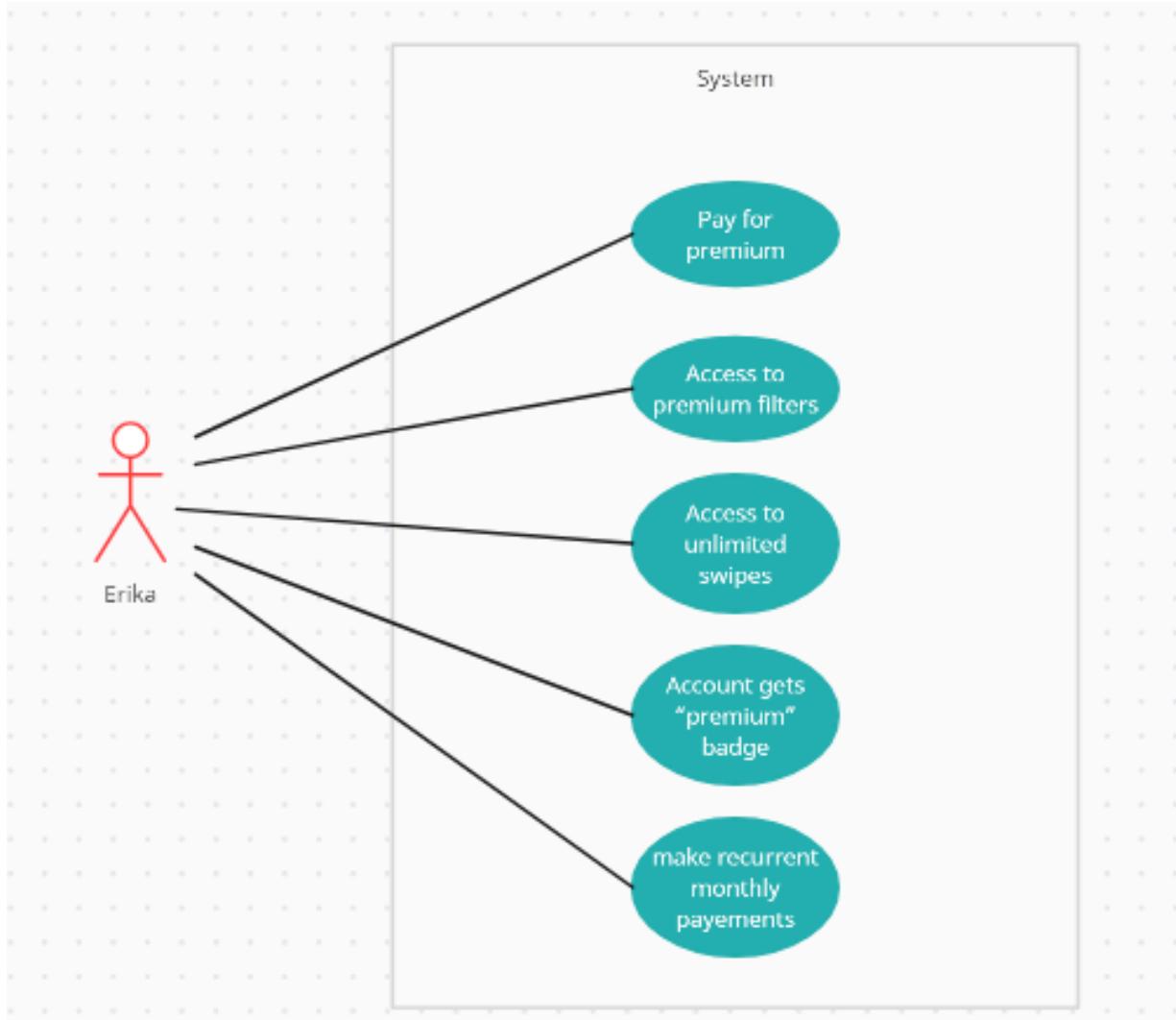
Assumptions:

- Students have StudySwamp installed and working internet access.

Use Case: Erika is deterred from joining a large study group because in the past she ended up having to do most of the work in group settings so she is hesitant to help strangers study if she won't get anything positive back. She downloads the app and is pleasantly surprised to see a GPA filter and a 5-star rating system implemented. These are premium features, she rationalizes this knowing that people with a premium account take studying seriously. The rating system also allowed her to choose someone who wouldn't waste her limited time.

Benefits Erika received:

- **Confidence:** She is confident that the partner she picks will be equally as devoted as her and she won't be left with the bulk of the work to herself.
- **Visibility:** Easy to find premium users with a sticker/badge on their profile.
- **Pay to win:** With the premium badge, Erika has unlimited swipes and is able to filter out his matches further than the basic package. She is able to match with the smartest people in the app who know what they are learning.



Use case 10: Reward program

Actor: Brook (Student), Steve (Student), StudySwamp (Company)

Assumptions:

- Students have StudySwamp installed and working internet access.

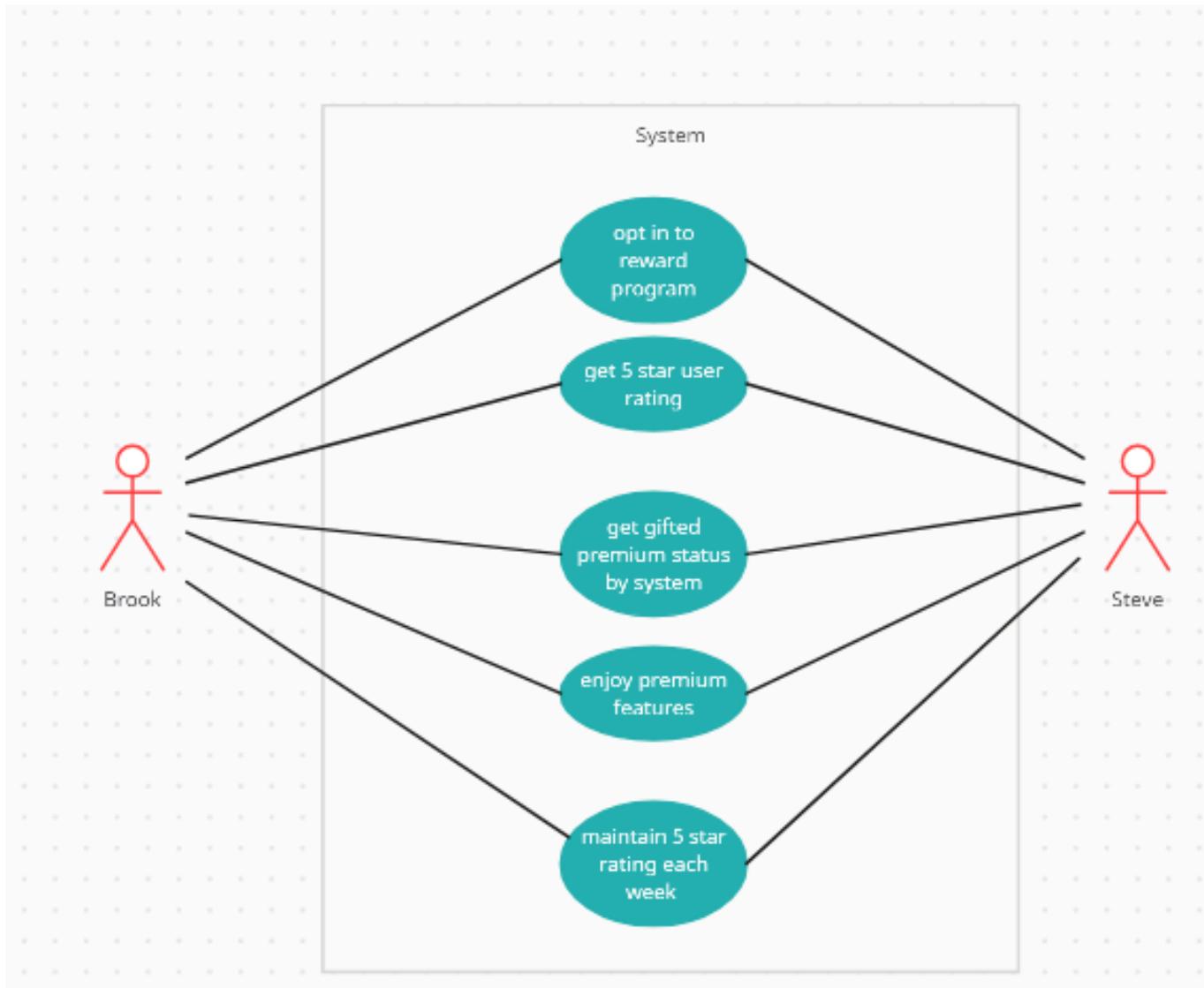
Use Case: Brook is an existing user, she downloaded the app because she needed serious help in math. Every user she has matched with so far has let her down by not showing up or goofing off. She has heard great things about the premium filters, and how those users tend to be more serious and take the appointments seriously, but Brook doesn't have any available income to spend. Although she can't afford the premium subscription, she finds that StudySwamp offers a workaround. She finds that with a high enough rating, the system will reward users with a free membership (contingent on keeping the rating up). She is proficient at science and so devotes herself to making her rating go up through that subject. Once her rating is high enough, she uses her premium filter and gets a match in math which is potentially better for her needs.

Benefits Brook and Steve received:

- **Premium for free:** They do not have to spend real money, but still enjoy the app's full potential.

Benefits StudySwamp received:

- **Quality Users:** Brook becomes a top-tier user in science. Steve becomes a top-tier user in math. What they bring in science and math is worth more for our user base than the premium fee.



Use case 11: Educator

Actors: Vihn (Educator), users (Students), StudySwamp (Company)

Assumptions:

- Vihn is a graduate student
- Vihn has experience beneficial to current student body
- Vihn has a StudySwamp account from when he went to school

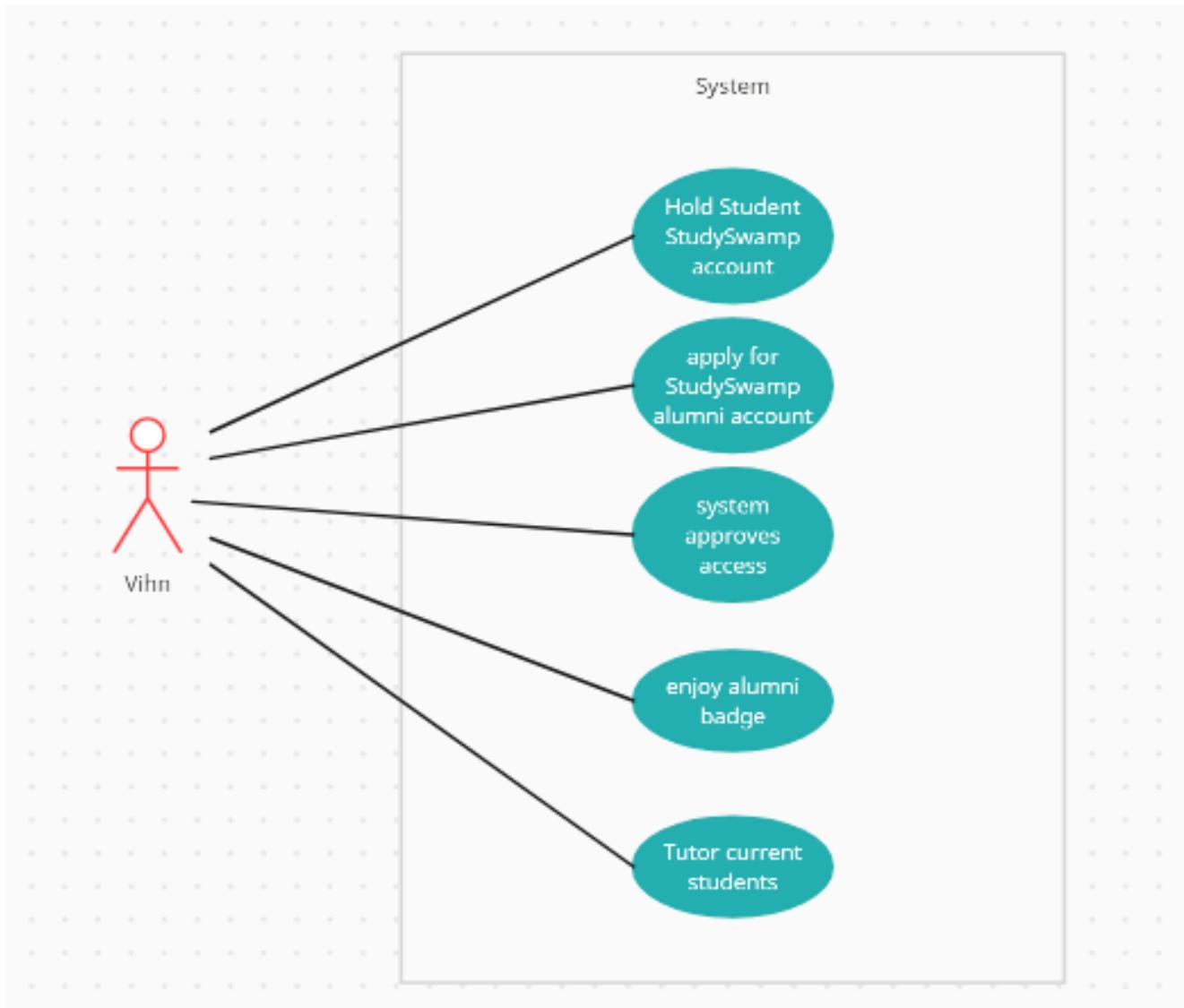
Use Case: Vihn is a very talented professional who graduated and decided to return to San Francisco State University for graduate school. Vihn now has work experience along with his degree and would like to help other undergraduates who are majoring in the same field he did. Vihn signs up for StudySwamp to meet potential students he can help. He finds Trip, Will, and Mary all looking for study partners for courses he did well in. StudySwamp is able to connect him with those students whom Vihn is able to help with homework and tutor. The tutoring helps Vihn reinforce the material he learned as an undergraduate.

Benefits Vihn received:

- **Reinforces his knowledge:** Being a tutor will also help Vihn

Benefits Students received:

- **Real-world insight:** Current students will be able to take in Vihn's real-world experience in their field
- **Building a community:** as students graduate, they may want to also become tutors like Vihn, and keep other generations coming back.
- **Networking:** making friends with people in the business is never a bad thing, it can lead to opportunities.



Use case 12: Forum

Actor: Geronimo (Student), StudySwamp (Company)

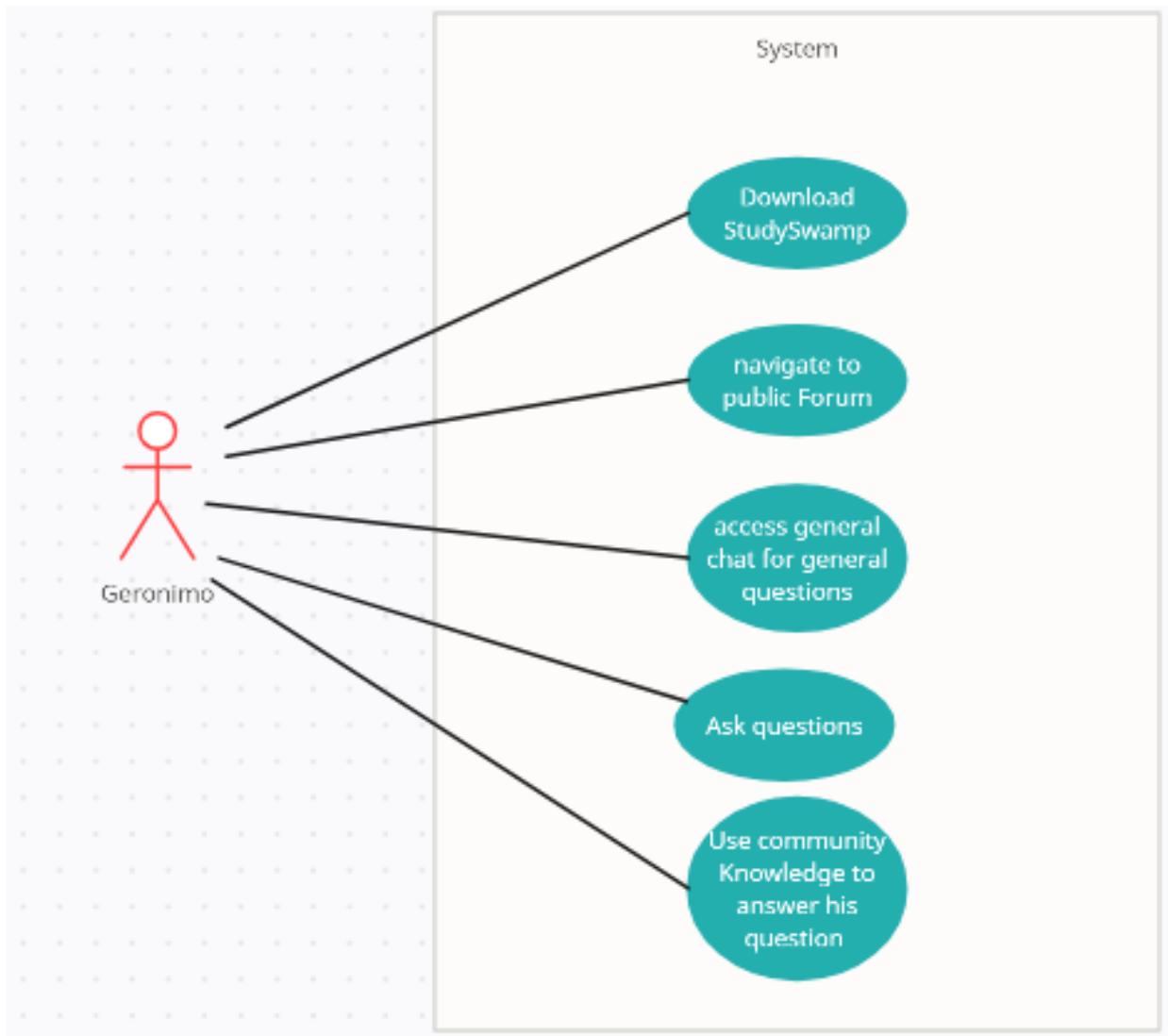
Assumptions:

- Students have StudySwamp installed and working internet access.

Use Case: Geronimo needs to know where to get the previous year's class textbook for one of his math classes, it is so obscure that Google is no help, he tried asking around but no one seems to know. He doesn't have any friends who took that course previously. He asked around other forums but had no luck until someone mentioned that StudySwamp had a general forum as well. He had heard of the app, but he only thought it was for matching with a study partner. He checked it out and found that since subjects were also categorized by class, not just subject he was able to narrow down where to ask.

Benefits Geronimo received:

- **Ease of use:** he was able to access the forum without needing to go through the matching portion that did not apply to him
- **Collective Knowledge:** the forum gives users access to various points of view, and different information.



Use case 13: Quality Control

Actor: Stephanie (Student), StudySwamp (Company)

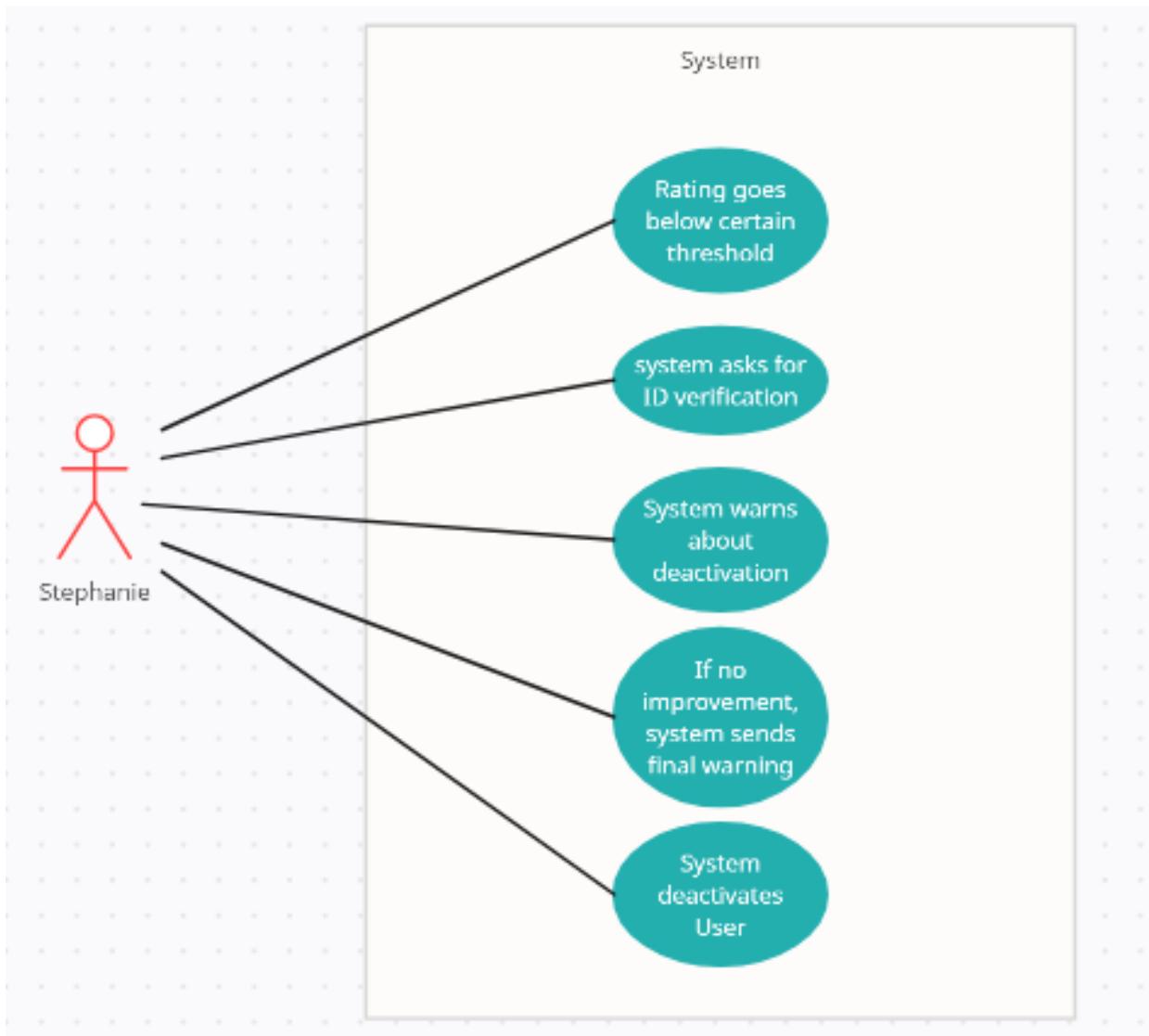
Assumptions:

- Students have StudySwamp installed and working internet access.

Use case: Stephanie is a jokester, and she likes to mess with people. She has been using StudySwamp successfully for a year now, getting her English grades up. One day she was bored and decided to start matching with people with Physics majors, she had no intention of showing up or if she did she would waste their time. Naturally, her rating began to go down. Once she was done goofing off she tried to schedule a legitimate English session but got a pop-up asking for her school ID for verification purposes. She thought it was weird but once she verified herself she got a message saying that her account was close to being deactivated due to low ratings. Realizing that she needed help in English she could not afford to lose her account, and so from then on she never played her pranks on the app again.

Benefits Stephanie received:

- **Verification:** Verify the user is a current student or educator
- **Return her focus:** The threat of deactivation changed the way Stephanie used the app



Data Items and Entities

User

A register user is a user that has created and verified an account

Premium user

A premium user is a user that has purchased the premium package.

Matched Pair

A matched pair is a set of 2 or users that have both swipe “yes” to each other.

Introvert

An introvert is a user that has been selected to only match with other introvert users.

Blocked user

A blocked user is a user that has been reported and blocked by another user.

Educator

An educator is a user that can prove they attended SFSU and graduated. May also be a teacher or tutor. Allowed to moderate individual or group sessions. Is allowed in Forum.

Tutor

A tutor is a user that has a high user rating, and opted in to become a tutor, allowed to moderate forum.

Group Cluster

A group match cluster is a set of three or more users that opted in to group matches. All matches swiped “yes” to each other.

Profile

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

Rating

A rating is a number that correlates to how many thumbs up/ thumbs down a user gets from their peers upon completing a study session.

Availability

Availability is the times available a user can attend a study session.

Proficient Subject

A proficient subject is the user’s best subject they know, used for tutoring or study sessions.

Challenged Subject

A challenged subject is the user's worst subject, used for tutoring or study sessions.

Forum

A forum is a public general chat platform all users can use.

Reward

A reward is given to exceptional users with high ratings, and will allow free premium status.

Warned User

A warned user is a user that has been reported by others, the system will keep track and remove the user if necessary.

Account

An account will hold a user's personal information and preferences.

Application

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

Call center

A call center is a system that allows users to communicate with their voice in real time.

Video call center

A video call center is a system that allows users to communicate virtually face to face.

Appointment center

An appointment center will create appointments for matched users to meet using their availability.

Functional Requirements

User Account Management:

1. A user shall be able to create an account.
2. A user shall have only one account.
3. An Account shall have 10 likes per day.
4. An Account shall have one email associated with it.
5. An account is allowed to log in from only one device at a time.

User Profile Management:

7. A user shall be able to view other users' profiles in the system.
8. A user shall be able to set their availability.
9. A user shall be able to establish their profile.
10. A user shall be able to set a filter for their study partner preference.
11. A user shall be able to accept or decline study partner invitations.
12. A user shall be able to video call with their study partner.
13. A Profile shall be owned and edited by one and only one user.
14. A Profile shall be allowed one picture of the owner.
15. A profile shall allow one picture of the user's proof of their GPA.
16. A profile shall be allowed a short biography.
17. A profile shall be allowed to be invisible when not wanting to study.
18. A profile shall have many subject attributes.

User Interface:

19. The interface shall allow users to swipe right or left on potential study partners.
20. The interface shall allow a user to navigate through a potential partner's profile.
21. The interface shall allow a user to access/edit their account.
22. The interface shall allow a user to access/edit their profile.
23. The interface shall allow a user to access/edit their preferences.

Matching System:

24. The system shall highlight common attributes users have with each other.
25. A set of users shall become matched if they both like each other.
26. Two matched users shall be able to connect with each other.
27. Two matched partners shall tell the system the date of their first meeting.

28. A user shall be able to rate their study partner after their meeting.

29.

Forum:

35. The Forum shall allow one main question per thread.

36. The Forum shall allow as many answers until the question is marked done.

37. The Forum shall allow each class its section.

38. The Forum shall allow users to post in their class forum.

39. The Forum shall allow many questions.

40. Only users who are in or have passed the class can ask or answer questions.

41. An Educator profile shall only have access to the forum.

42. An Educator shall be allowed to answer questions in the forum.

43. An Educator shall be allowed to mark questions for removal in the forum.

44. A user shall be able to ask many questions in the forum.

45. A user shall be able to answer many questions in the forum.

46. A user shall be able to receive stars for their answers in the forum.

47. A user shall be able to give 2 stars per day.

Additional Features:

48. A matched pair shall be able to unmatch.

49. A matched pair shall be able to choose whether they want to meet in person or online.

50. A matched pair shall have an appointment created for them automatically

51. A matched pair shall have a zoom meeting link automatically created for them.

52. A profile shall be marked student or professor.

53. A user shall be able to block a user from seeing them.

54. A user shall be able to flag another user for review.

Non-Functional Requirements

1. Usability

- 1.1. The U.I. will be designed to be as intuitive as possible allowing use from all technical backgrounds.
- 1.2. The Application shall be compatible with web browsers, and mobile devices
- 1.3. The Application shall conduct surveys on features that could improve usability
- 1.4. The Application U.I. shall be streamlined and responsive
- 1.5. The system shall send notifications to the user's device.
- 1.6. A user shall be allowed to match with many partners.
- 1.7. A high rating shall make the user more visible.
- 1.8. The system shall show a user their ideal match.
- 1.9. The system shall notify users when they have a match.
- 1.10. A high rating shall gift users one free super like.

2. Reliability

- 2.1. The application shall achieve 90% uptime
- 2.2. The Application shall minimize the occurrence of bugs and crashes

3. Scalability

- 3.1. The System shall be containerized allowing for simple scaling
- 3.2. The System shall automatically provision resources to handle fluctuating demand

4. Maintainability

- 4.1. The Application shall be formed in a manner that supports hot fixes to minimize downtime
- 4.2. The Restoration period shall be less than 24 hours
- 4.3. The Application shall be able to update and upgrade without downtime

5. Utility

- 5.1. The site shall support concurrent usage by many users with negligible performance reduction
- 5.2. The System shall scale to accommodate a growing user base
- 5.3. The System shall support Firefox version 123.0
- 5.4. The System shall support Google Chrome Version 122.0.6261.70

6. Performance

- 6.1. The site shall load in less than 7 seconds on a wireless connection with a speed of 350Mbps.

7. Privacy

- 7.1. Users shall have access to privacy settings

- 7.2. Users shall be able to hide their profile
- 7.3. Users shall be able to view the privacy agreement
- 7.4. The Platform shall treat customer data as sensitive

8. Data Integrity

- 8.1. The Application shall ensure that all billable data is accurate
- 8.2. The Application shall

9. Security

- 9.1. The Application shall encrypt sensitive user data
- 9.2. The Application shall comply with regional data safety regulations
- 9.3. Payment Information shall be routes through a trusted third party service provider using state of the art security measures

10. Storage

- 10.1. The Application shall assign 10MB of Memory per table
- 10.2. The Application shall make automatic backups to prevent data loss

11. Deployment

- 11.1. The Application shall use github integrations to build the Docker container
- 11.2. The Cloud instance shall automatically transition to the newest version of the application when available

12. Coding Standards

- 12.1. The Team shall use the same styling for all written code
- 12.2. The Team shall use spaces for indentation
- 12.3. The Team shall write thorough documentation and comments
- 12.4. The Team shall own the codebase and act responsibly
- 12.5. The Team shall write tests that ensure a good working product

13. Environmental Sustainability

- 13.1. The Application shall use as few resources as needed
- 13.2. The Application shall work with responsible service providers
- 13.3. The Application shall minimize its footprint and make continuous efforts to improve further on sustainability

Competitive Analysis

Competitive Analysis Table

Company/app	Chegg Study	Studypal.co	MoocLab	Study Together
Strengths	<p>24hr study support</p> <p>Expert or Student answers</p> <p>UI is straightforward.</p> <p>Very large user base.</p> <p>Other amenities are offered</p> <p>Very large amounts of questions were answered.</p> <p>Tutoring sessions</p>	<p>Searches for people or professionals on the criteria near the user's area</p> <p>Worldwide access</p> <p>Helpful for major tests like BAR exam GMAT, LSAT etc.</p>	<p>It is a free website</p> <p>Worldwide: anyone around the world can access</p> <p>Many groups to join</p> <p>Many forums for many different topics</p> <p>There are a lot of forums for different topics</p>	<p>Free</p> <p>World wide access</p> <p>24-hour access to study rooms</p> <p>Can make your own or have friends join in.</p> <p>Tools to help studying such as timers, goals, and customizable backgrounds</p> <p>Discord community to ask questions and such</p> <p>Direct messaging to people in the study group</p>
Weakness	<p>Incorrect Answers</p> <p>Terrible Customer Service</p> <p>Difficult to cancel a subscription</p>	<p>Reports of crashing.</p> <p>Very little to no interaction on social media.</p>	<p>The entire website is a forum board.</p> <p>You either look for users or advertise yourself to make a study group.</p> <p>Mostly online</p> <p>Making an in-person meeting will be difficult</p>	<p>It is online only.</p> <p>You have to form your own study groups.</p> <p>Promotes procrastination.</p> <p>The main use is the discord voice channels to study so if someone enters there will be a discord notification noise</p> <p>Reports of people looking at other people who have cams on</p>
Pricing	<p>Chegg study is \$12.96 per month only access to homework solutions.</p> <p>Chegg study pack is \$19.95 per month and comes with</p>	<p>Free to make an account, but it costs \$5 per month for direct messaging features \$10 for 3 months Premium is \$10</p>	<p>free</p>	<p>Free</p>

	homework solutions, quiz and exam help, expert proofreading, plagiarism detection, and math help	per month		
Social media	Facebook, X, Instagram	Facebook, Instagram, LinkedIn	X, LinkedIn, Facebook	Facebook, LinkedIn, Instagram, TikTok Discord
Onboarding	Account Creation is super simple and only takes 3 clicks. Automatic creation with Apple, Facebook, or Google accounts. Without using them it took just an email and password. There is no free access to Chegg study	3 clicks for account creation. Can automatically create an account with Google or LinkedIn. Upon creating an account, email verification is required	The UX is slightly worse than Chegg or Studypal. The UX To access a forums board to find study partners is worse because it takes 5-6 clicks and I went to the wrong page labeled "find a study buddy" when the correct page to find a new study partner was "Study Buddy Members"	Making an account is easy but it can only be joined with a preexisting social media account like Discord, Facebook, X, Unidays and a Google account

Competitive analysis Cont.

Company	Discord	Canvas
Strengths	<ul style="list-style-type: none"> Fast and responsive Relatively easy to make a discord channel Voice channels are high quality for free Easy to connect with others Direct messaging Easy to use text channels Sign in scanning QR code with the phone app if it's already signed in. 	<ul style="list-style-type: none"> Already established with colleges Makes accounts for students Flexible to work with all departments. Available list of all the users in your course. Tons of features.
Weaknesses	<ul style="list-style-type: none"> Difficult for people who are new or not tech savvy There's a learning curve for fully utilizing Settings can be overwhelming 	<ul style="list-style-type: none"> Not many students use the tools provided by Canvas Discussion forums are rarely used and sometimes looked down on in certain majors.
Pricing	<ul style="list-style-type: none"> Free for general usage. \$2.99 per month for the basic package for larger uploads and special emoji usage. \$9.99 per month for larger uploads higher quality video streaming and customizable discord profiles 	<ul style="list-style-type: none"> Free for students
Social media	X, Instagram	Instagram X
Onboard experience	<p>Discord website is simple to create an account it takes just 2 total clicks from the home screen.</p>	<p>Creating a Canvas account is pre-made by the school.</p>

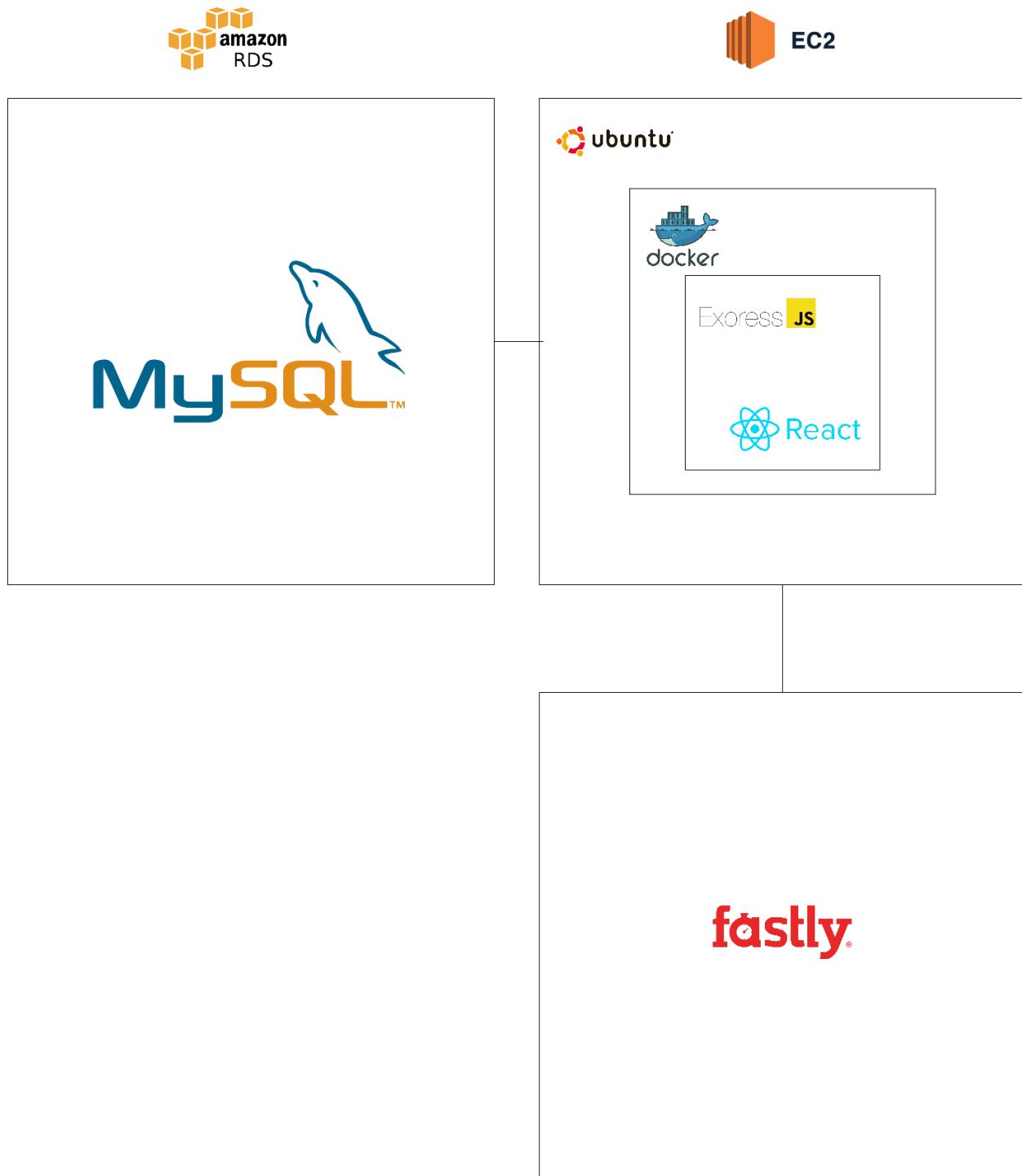
Competitive Features Table:

Features/ Company	Chegg Study	StudyPal	MoocLab	Study Together	Discord	Canvas	StudySwamp
Forums	-	-	++	-	+	+	++
Peer-to-peer questions and answers	++	+	+	+	+	+	++
Expert answer questions	++	+	-	+	-	+	++
Automatically schedule a study group.	-	-	-	-	-	-	++
Unique matching system	-	++	-	-	-	-	++
Enhanced profile feature	-	+	+	-	+	+	+
Academic integrity integration	-	-	-	-	-	+	+
Unique reward system for questions answers and study sessions	+	-	-	++	-	-	++

Summary

StudySwamp distinguishes itself with a suite of features designed to enhance the learning experience: our forums offer a more focused and secure environment than those on MoocLab, Discord, or Canvas by exclusively allowing access to students and professors from the same course, effectively preventing misinformation from external sources. Our unique matching system surpasses StudyPal by facilitating more personalized connections, matching students by courses, majors, and availability for more relevant and effective study partnerships. Furthermore, we enhance user comfort and safety beyond what is available on Discord, StudyTogether, or StudyPal with a swipe feature that enables students to easily manage their interactions, ensuring a secure and comfortable environment for all. By surpassing Chegg Study, StudySwamp provides a reliable platform for forming study groups, engaging in accurate forums, and receiving expert answers directly from professors, thereby minimizing the risk of incorrect information. To add to our already groundbreaking application, we are developing an application feature that will automatically schedule meetings on Zoom or in-person based on the users' preference as soon as two users approve of each other, streamlining group study. Here at StudySwamp, your present is your future success.

Technical Architecture



1. Server Host:
 - 1.1. AWS EC2 t2.micro-Instance 1vCPU 1GB RAM
2. Operating System
 - 2.1. Ubuntu 22.04
3. Database
 - 3.1. AWS RDS db.t3.micro 2vCPU 1GB RAM
 - 3.2. MySQL 8.0.35
4. Web Server
 - 4.1. Docker Container
5. CDN
 - 5.1. Fastly CDN
 - 5.2. SSL Cert: GlobalSign Atlas R3 DV TLS CA 2023 Q4 (Fastly Provided)
6. Server Side Language
 - 6.1. JavaScript
7. Additional Technologies
 - 7.1. Web Framework: React
 - 7.2. Backend Framework: Express.js
 - 7.3. IDE: Visual Studio Code
 - 7.4. Browsers: Mozilla Firefox 123.0
 - 7.5. Browsers: Google Chrome 122.0.6261.70
8. Note on Technical Architecture Change
 - 8.1. In our initial draft of the project, we intended to use Let's Encrypt Certbot to obtain an SSL certificate. After careful deliberation, we decided to change that and use a CDN service to provide faster delivery of our application and to provide an SSL certificate.

Checklist

1. Meeting Outside of Class DONE
2. Github Master chosen DONE
3. Decided on SW tools and Server Provider DONE
4. Ready to use chosen Front/Back end frameworks and ready to learn DONE
5. Team Lead Made sure everyone read M1 and agrees DONE
6. Github Organized as outlined in class DONE

Team Evaluation

Lennart (Team Lead) 10

Checkpoint 1

1. Created the Documents
2. Wrote the Executive Summary
3. Wrote 2 Use Cases and Diagrams
4. Reviewed and Reworked the non-functional requirements
5. Reviewed the whole document
6. Participated on Discord and in team meetings
7. Submitted all work on time

Checkpoint 2

1. Created the AWS EC2 Instance
2. Created the AWS RDS Instance
3. Wrote and deployed the throw-away
4. Wrote About Me Page
5. Wrote Documentation

Rafael (Backend Lead) 10

Checkpoint 1

1. Wrote 2 Use Cases
2. Reviewed the Executive Summary
3. Participated on Discord and team meetings
4. Active voice in deciding the architecture
5. Participating actively in Discord and in team meetings
6. Submitted all work on time

Checkpoint 2

1. Wrote About Me Page
2. Refactored the React app to use better architecture
3. Refactored the About Me Page to be simpler
4. Implemented Styling on the About Me Page
5. Set the precedent for good coding practices by researching the best methods

Edmund (Frontend Lead) 10

Checkpoint 1

1. Wrote 2 Use Cases
2. Conducted a Survey to determine requirements and use cases
3. Evaluated survey results and shared summary with the group
4. Functional Requirements
5. Reviewing Competitive Analysis
6. Participating actively in Discord and in team meetings

7. Revised the docs to correct grammar
8. Submitted all work on time

Checkpoint 2

1. Wrote About Me Page

Julio (Docs-Editor) 10

Checkpoint 1

1. Wrote 2 Use Cases
2. Reviewed Use Cases and added more
3. Standardized the diagrams for the Use Cases
4. Wrote Non-Functional Requirements
5. Iteratively Edited the Document
6. Proofed the final document
7. Participated actively on Discord and in team meetings
8. Wrote and sent the meeting notes in the Discord
9. Submitted all work on time

Checkpoint 2

1. Wrote About Me Page

Conrad (Github-Admin) 10

Checkpoint 1

1. Wrote 2 Use Cases
2. Wrote the Competitive Analysis
3. Conducted a survey to determine requirements
4. Summarized and presented survey results
5. Reviewed Functional Requirements
6. Participated actively on Discord and in team meetings
7. Submitted all work on time

Checkpoint 2

1. Wrote About Me Page

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
Milestone 2 V2	Apr 24, 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

Schedule

A Schedule is the mapping of which classes a User is taking. Each entry contains the user and the class

Functional Requirements

Priority 1: Must Have

1. User

- 1.1. A User shall be able to create an account
- 1.2. A User shall only have one account
- 1.3. A User shall be able to view other users' accounts in the system
- 1.4. A User shall be able to set their availability
- 1.5. A User shall be able to establish their profile
- 1.6. A User shall be able to ask many questions in the Forum
- 1.7. A User shall be able to answer many questions in the Forum
- 1.8. A User shall be able to like other Users
- 1.9. A User shall be able to match with other Users

2. Profile

- 2.1. A Profile shall be owned and edited by one and only one user
- 2.2. A Profile shall be allowed one picture of the owner
- 2.3. A Profile shall be allowed a short biography
- 2.4. A Profile shall have many subject attributes
- 2.5. A Profile shall be able to set a filter for their study partner preferences
- 2.6. A Profile shall be marked Student or Educator
- 2.7. A Profile marked Educator shall be able to access the Forum
- 2.8. A Profile marked Educator shall be allowed to answer questions in the Forum

3. Account

- 3.1. An Account shall have one email associated with it

4. User Interface

- 4.1. The Interface shall allow a User to navigate through a potential Match's profile
- 4.2. The Interface shall allow a User to access/edit their Profile
- 4.3. The Interface shall allow a User to access/edit their Account

5. Match

- 5.1. A Match shall be provided a meeting link
- 5.2. A Match shall be provided a meeting time
- 5.3. A Match shall be made based on different criteria

6. Forum

- 6.1. The Forum shall allow one main question per Thread
- 6.2. The Forum shall allow many answers until marked done
- 6.3. The Forum shall allow each class its section
- 6.4. The Forum shall allow users to post in their class Forum

6.5. The Forum shall allow many questions

Priority 2: Desired

1. Profile

- 1.1. A Profile shall be allowed to be invisible when not wanting to study
- 1.2. A Profile marked Educator shall be allowed to remove questions from the Forum

2. User

- 2.1. A User shall be able to flag another user for review
- 2.2. A User shall have a match rating which determines visibility
- 2.3. A User shall review a match after their first meeting
- 2.4. A User shall be allowed to block another User

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 Matching System shall highlight common attributes users have with each other
- 4.2. The Matching System shall notify users when they have a Match
- 4.3. The Matching System shall send a notification to the user's device
- 4.4. The Matching System shall show a user their ideal match
- 4.5. The Matching System shall let users decide whether they want to meet in person or online

5. Forum

- 5.1. The Forum shall limit access to students taking that class
- 5.2. The Forum shall allow educators to access all the Threads

Priority 3: Opportunistic

1. Account

- 1.1. An Account shall be allowed to log in from one device at a time

2. User

- 2.1. A User shall be able to Video Call with their partner

Use Cases

Priority 1: Must Have

Use case 1: Students using the site need a partner for a group project.

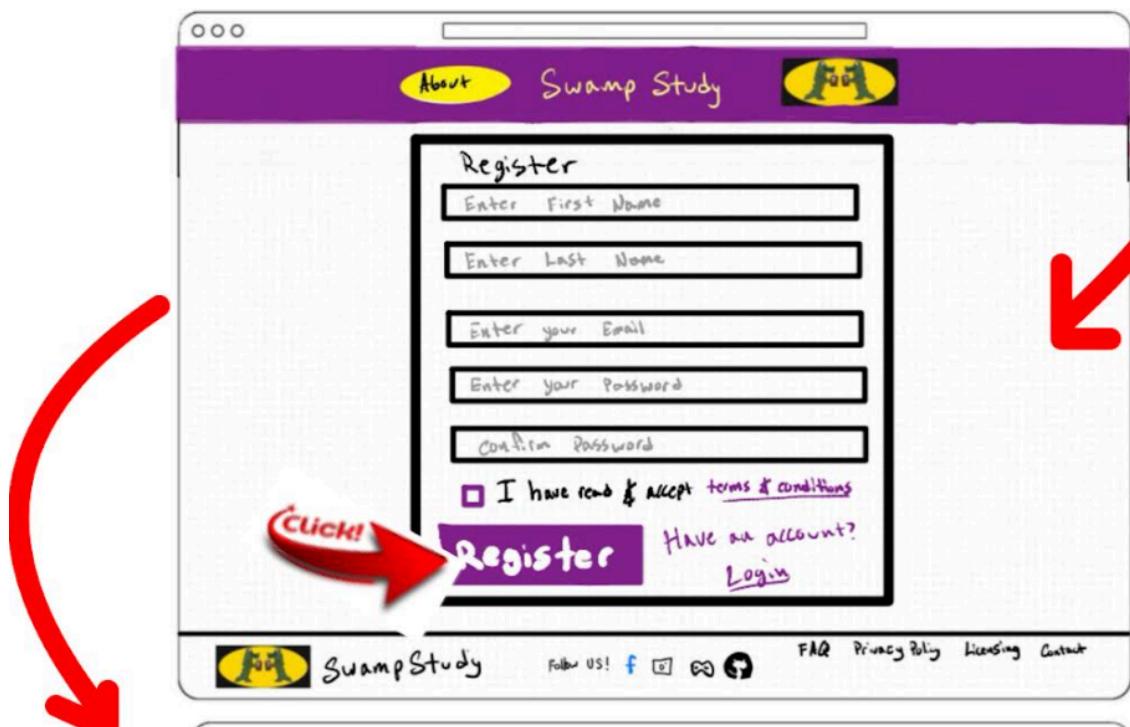
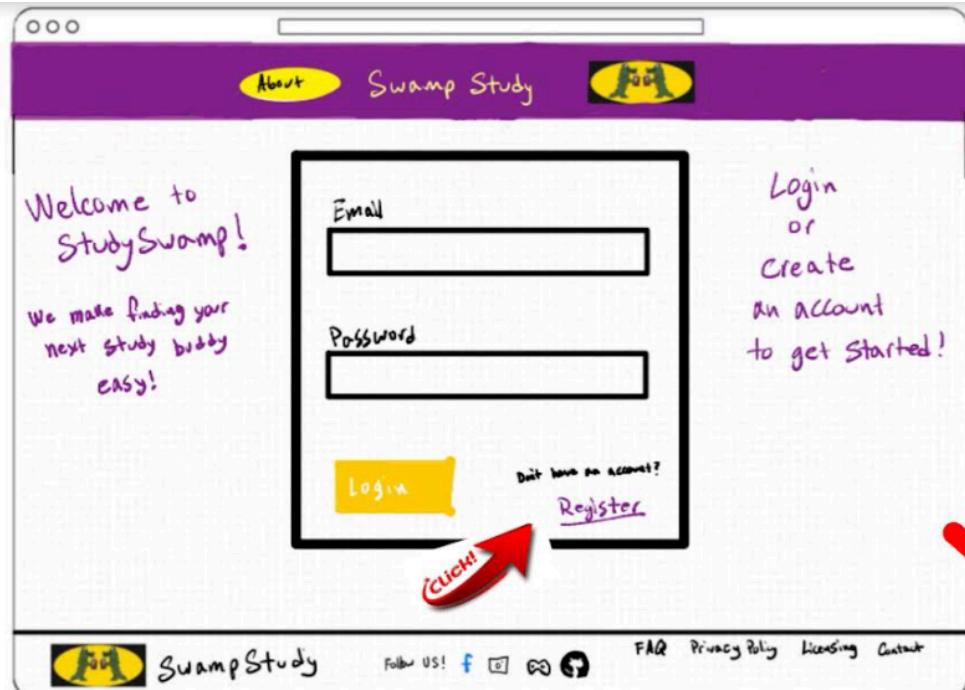
Use case 1 Storyboard:

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.





The image displays two screenshots of the Swamp Study mobile application interface, showing the process of finding study partners.

Screenshot 1: Matching Screen

- Top Navigation:** Includes "About", "Swamp Study" logo, "Settings", and "Logout".
- Section Headers:** "Matching" and "IN Forums!".
- Filters:** "Filters" section with dropdown menus for "Department" and "Class".
- Search:** A search bar with placeholder "Search..." and a yellow "Search" button.
- Buttons:** "Make a Post" and "Match Me".
- Bottom Navigation:** Includes "FAQ", "Privacy Policy", "Licensing", and "Contact".

Annotations: A large red arrow points from the bottom of the first screenshot down to the second screenshot. A red arrow labeled "CLICK!" points to the "Match Me" button on the first screen.

Screenshot 2: Matching Result Screen

- Top Navigation:** Includes "About", "Swamp Study" logo, "Settings", and "Logout".
- Section Headers:** "In Matching!" and "Forums".
- Text:** "Find Someone to Study With Below!"
- User Profile:** Shows a placeholder profile picture and five yellow stars. The user's name is obscured by a black marker.
- Availability:** A table showing weekly availability:

Monday	Yes
Tuesday	Yes
Wednesday	No
Thursday	No
Friday	No
Saturday	No
Sunday	No
- Courses:** A table showing courses taken:

Course 1
Course 2
- Right Column:** "Your matches!" section with two entries: "Matched with ..." and "Matched with ...".
- Bottom Navigation:** Includes "FAQ", "Privacy Policy", "Licensing", and "Contact".

Annotations: A large red X is placed over the user profile picture. A green checkmark is placed next to the "Matched with ..." entry. Three red arrows labeled "CLICK!" point to the "Matched with ..." entries on the right side of the second screen.

ooo

About Swamp Study Settings Logout

In Matching!

Match Details

Name: [Redacted]
Bio: [Redacted]
Rating: [Redacted] Email: [Redacted]

Meeting Details

Date: [Redacted] Time: [Redacted]
Topic: [Redacted] Location: [Redacted]

Unmatch / Report Rate Back

Your matches!

Matched With ---
Matched With ---

Swamp Study Follow US! f FAQ Privacy Policy Licensing Contact

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

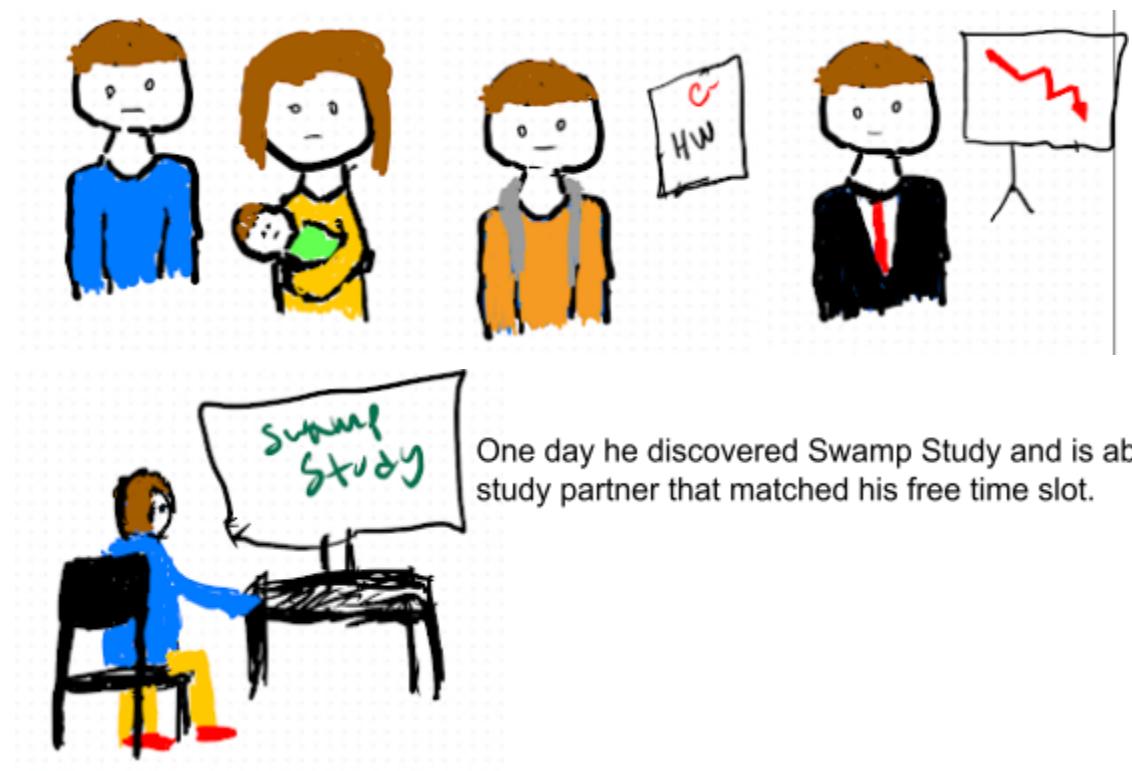


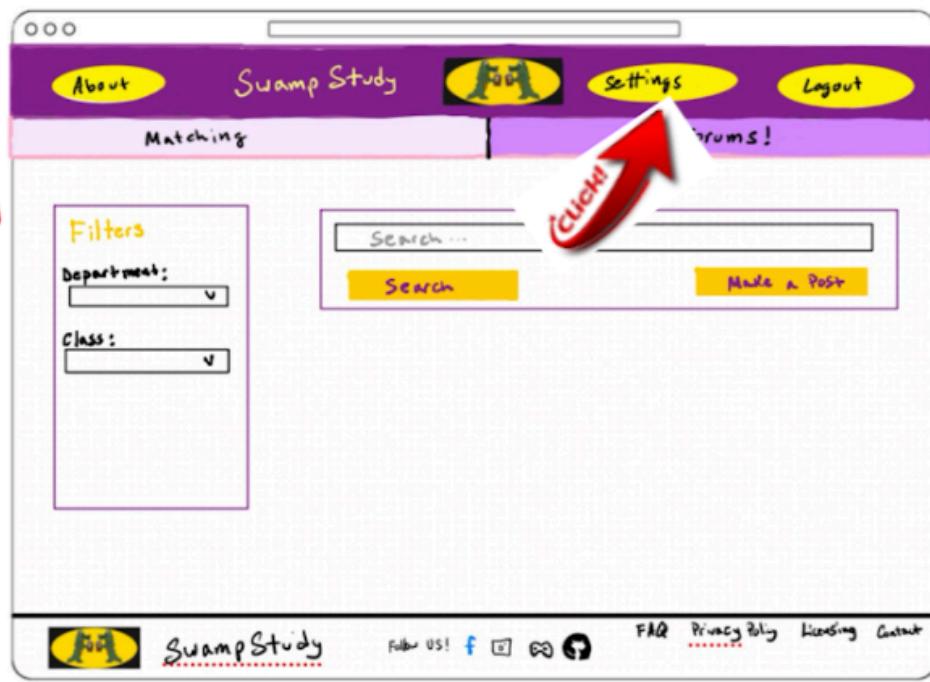
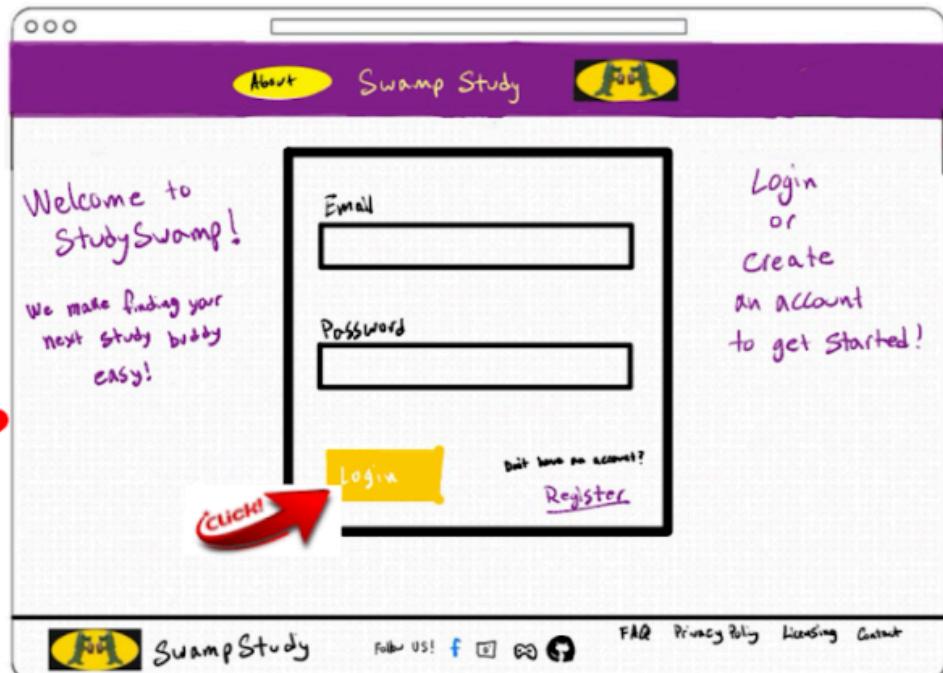
Use case 3: Non-Traditional Student looking for study partner/ Availability

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





User Settings

Name: [Placeholder]

Bio: [Placeholder] [Edit Profile](#)

Availability:

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

[Edit Availability](#)

Class Schedule:

department: [Placeholder] [View](#)

Class: [Placeholder] [View](#)

Add Class [Save Changes](#)

Selected Classes:

[Save Changes](#)

Update Account

Email: [Placeholder]

New Password: [Placeholder]

Confirm Password: [Placeholder]

Introvert

Is Hidden

[Hide Account](#)

Update Account **Delete Account**

Swamp Study Follow US! [Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

In Matching!

Find Someone to Study With Below!

Name: [Placeholder] ★★★★
[Placeholder]

Availability:

Monday	Yes
Tuesday	Yes
Wednesday	No
Thursday	No
Friday	No
Saturday	No
Sunday	No

Courses:

- Course 1
- Course 2

Your matches!

Matched with ...

Matched with ...

CLICK HERE

Swamp Study Follow US! [Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

ooo

About Swamp Study Settings Logout

In Matching!

Match Details

Name: 

Bio:

Rating: Email:

Meeting Details

Date: 3/20 Time: 2PM

Topic: Location:

Unmatch / Report Rate

forums

Your matches!

Matched with ---

Matched with ---

Back

Swamp Study

Follow US!   

FAQ Privacy Policy Licensing Contact



Now he can stop worrying so much about school and focus on his job and family. Everyone is happier.

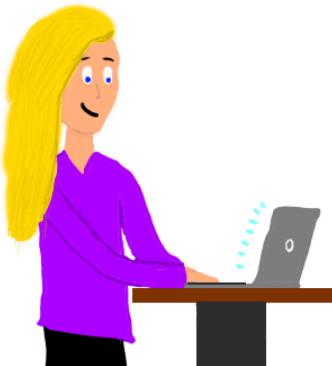
Use case 4: User looking to find a safe person to study with

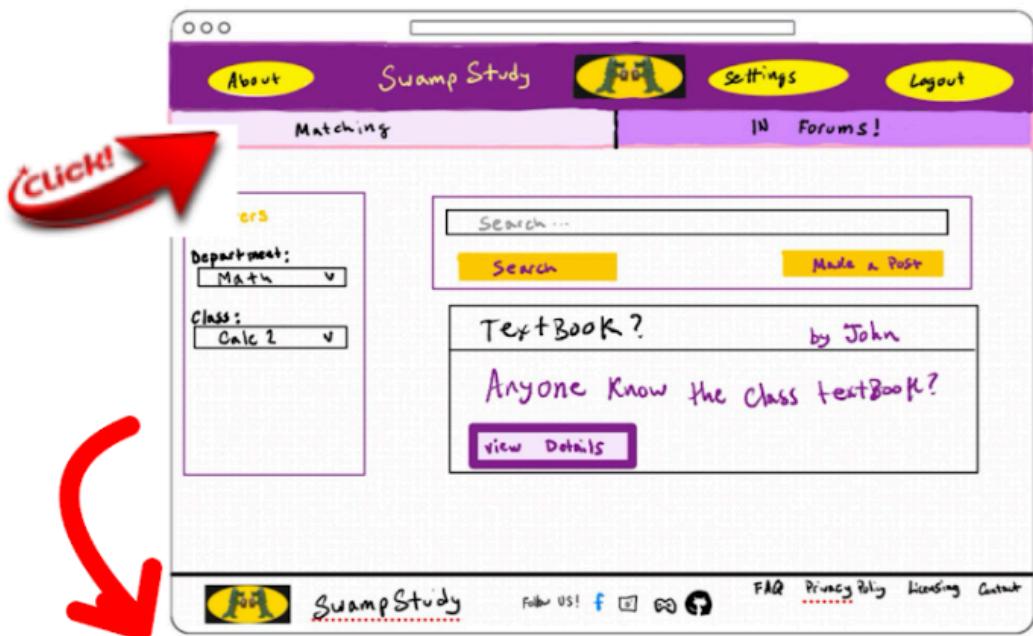
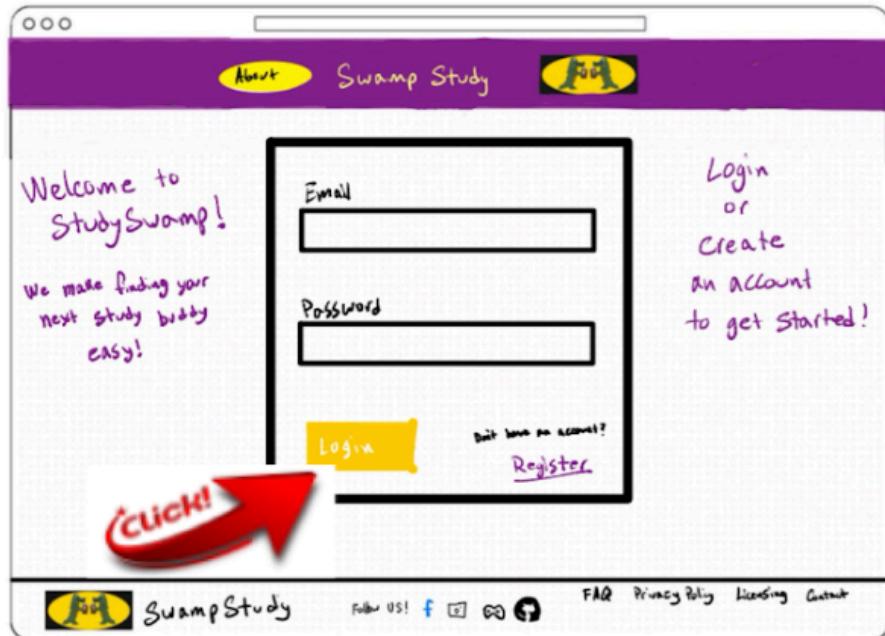
Use case 4 Storyboard: Draft in progress

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.





CLICK HERE

This screenshot shows the user's profile page. On the left, there is a large red 'X' over the user's name and a black silhouette icon. Below the name is a 5-star rating. To the right of the user profile is a green checkmark. On the far right, under 'Your matches!', it says 'Matched with ---'. At the bottom, there is a navigation bar with links for 'FAQ', 'Privacy Policy', 'Licensing', and 'Contact'.

This screenshot shows the 'Match Details' screen. It displays the user's name, bio, and rating. Below this is a section for 'Meeting Details' with fields for 'Date', 'Time', 'Topic', and 'Location'. At the bottom, there are three buttons: 'Unmatch / Report' (red), 'Rate' (yellow), and 'Back'.

ooo

About Swamp Study Settings Logout

In Matching!

Match Details

Name: 

Bio:

Rating:

Email:

Meeting Details

Date: Time:

Topic: Location:

Unmatch / Report Rate Back

Your matches!

Matched with ---

Matched with ---

Swamp Study

Follow US! f o g

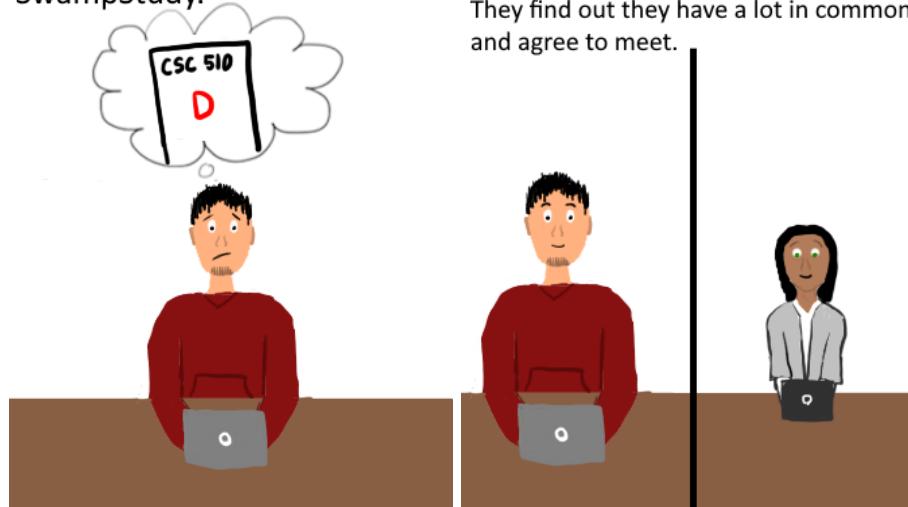
FAQ Privacy Policy Licensing Contact



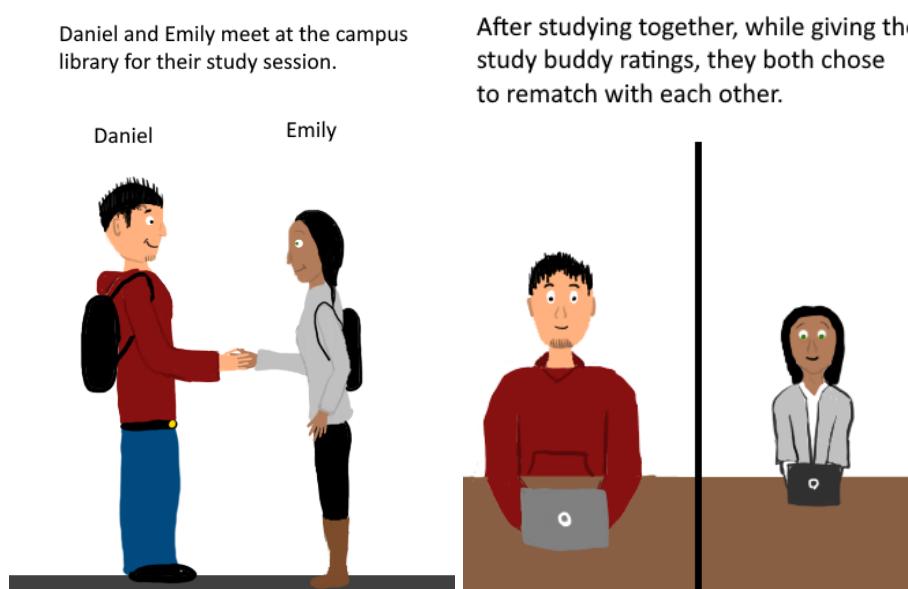
Use case 5: Finding a long term study partner

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.

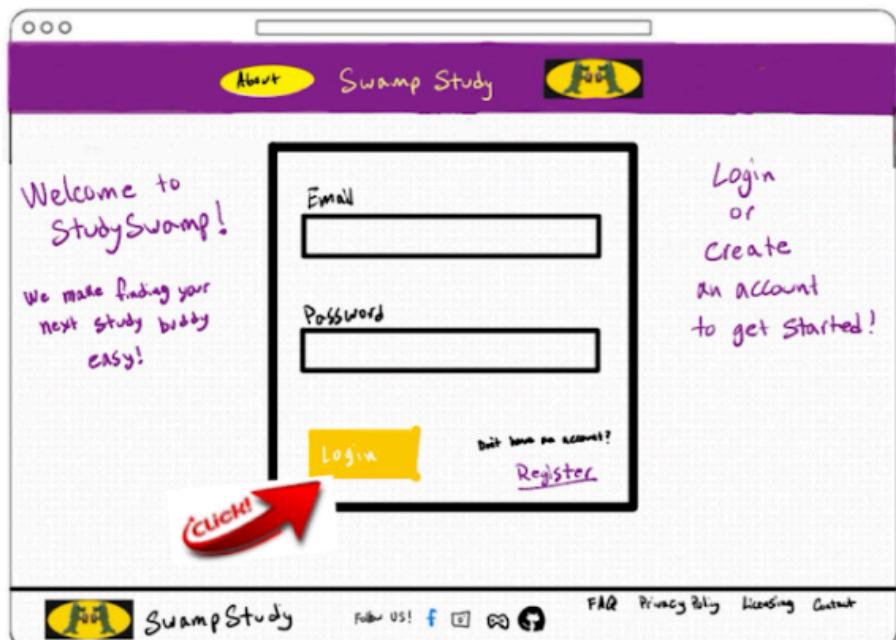


Daniel matches with Emily and they discuss their study habits and etc. They find out they have a lot in common and agree to meet.



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.



Matching

Filters

Department:

Class:

Search ...

Search

Made a Post

TextBook? by John

Anyone Know the class textbook?

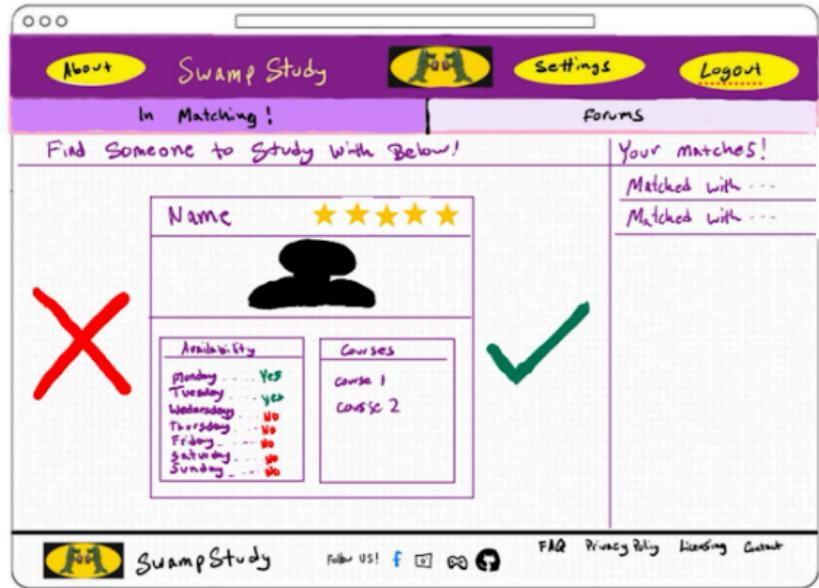
[View Details](#)

Settings

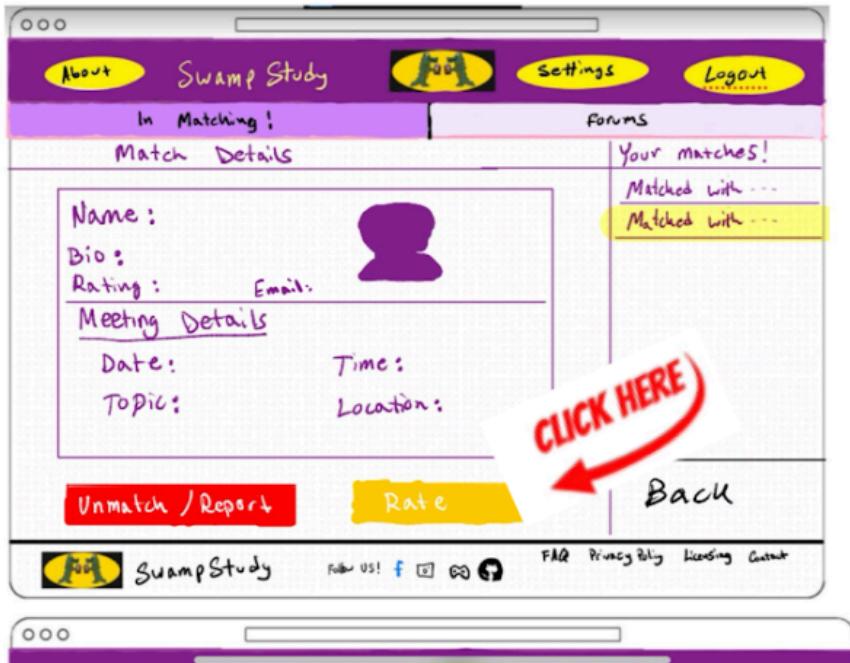
Logout

Follow US! [Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

FAQ Privacy Policy Licensing Contact



CLICK HERE



CLICK HERE



ooo

About Swamp Study Settings Logout

In Matching!

Rate Study Partner

Partner Name

Match with this person Again??

Submit

Forums Your matches!

Matched With ---

Matched With ---

Back

Swamp Study Follow us! FAQ Privacy Policy Licensing Contact

Use case 6: Hiding Profile

Use case 6 Storyboard



Kyle is a social butterfly, he craves company. He is also very studious, he is a regular SwampStudy user. But sometimes he is overwhelmed.

He needs the app but he also needs a break.

He is happy to see StudySwamp has a "Hide your Profile" feature.

Welcome to StudySwamp!

We make finding your next study buddy easy!

About Swamp Study

Email

Password

Login

Register

But have no account?

Login or Create an Account to get Started!

Follow US! [f](#) [t](#) [g](#) [o](#)

FAQ Privacy Policy Licensing Contact

About Swamp Study

Matching

Settings

Logout

Filters

Department:

Class:

Search ...

Search

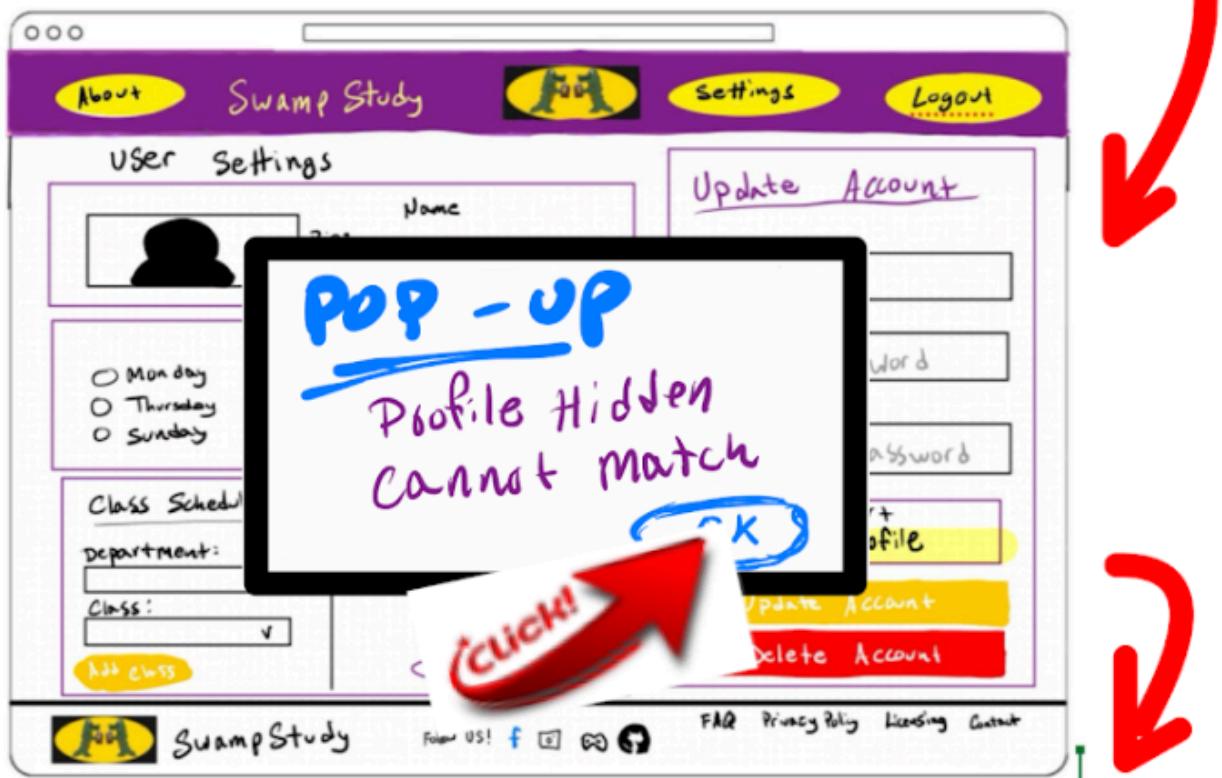
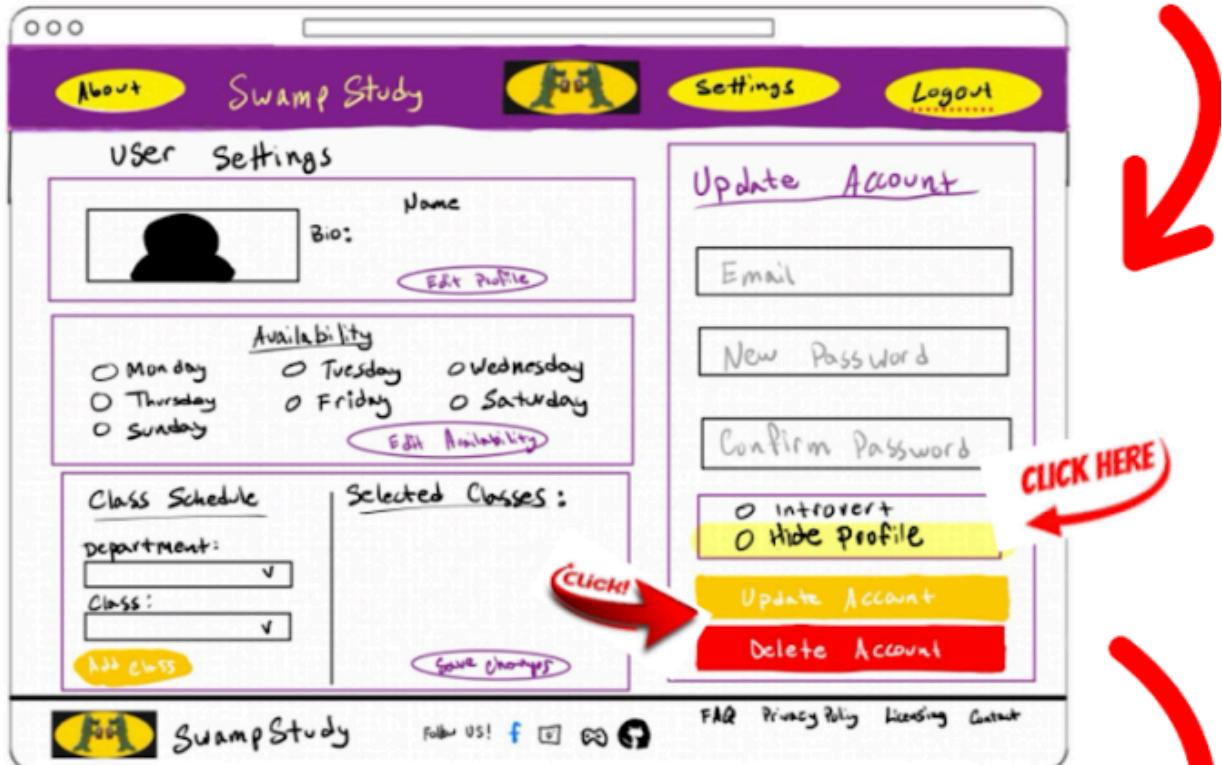
Make a Post

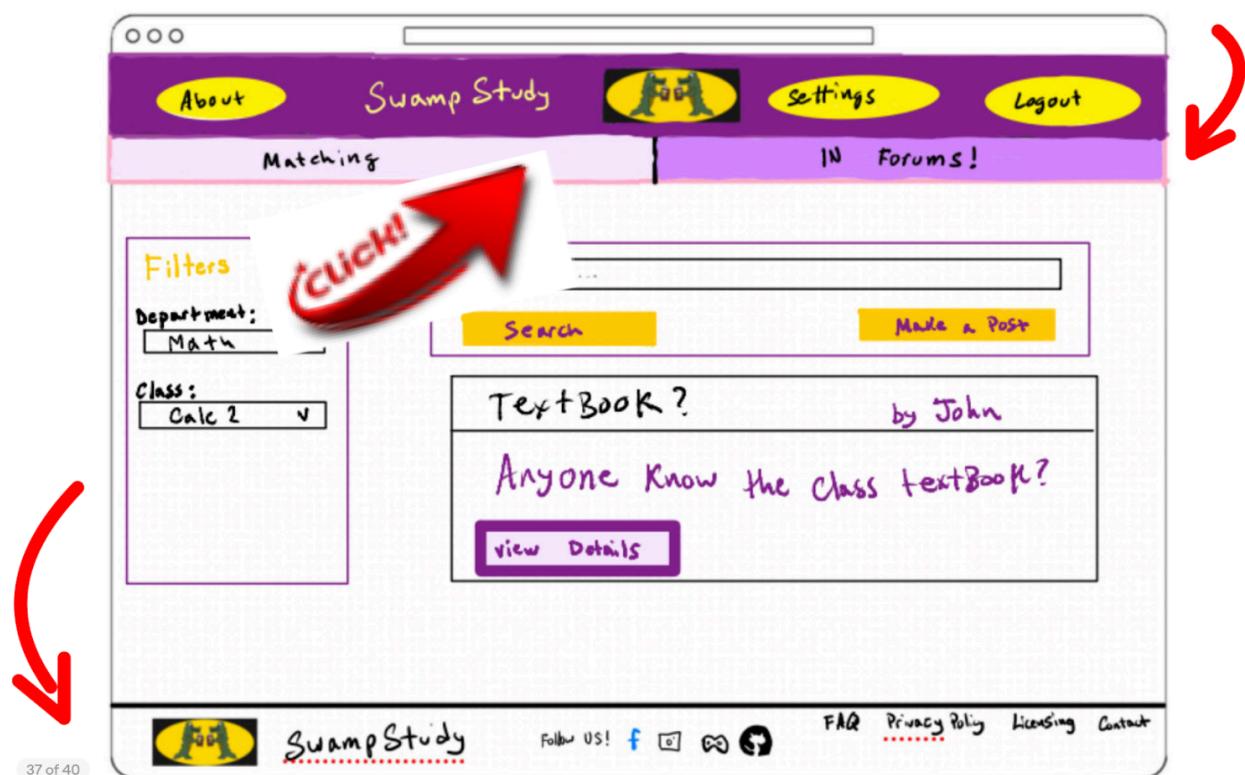
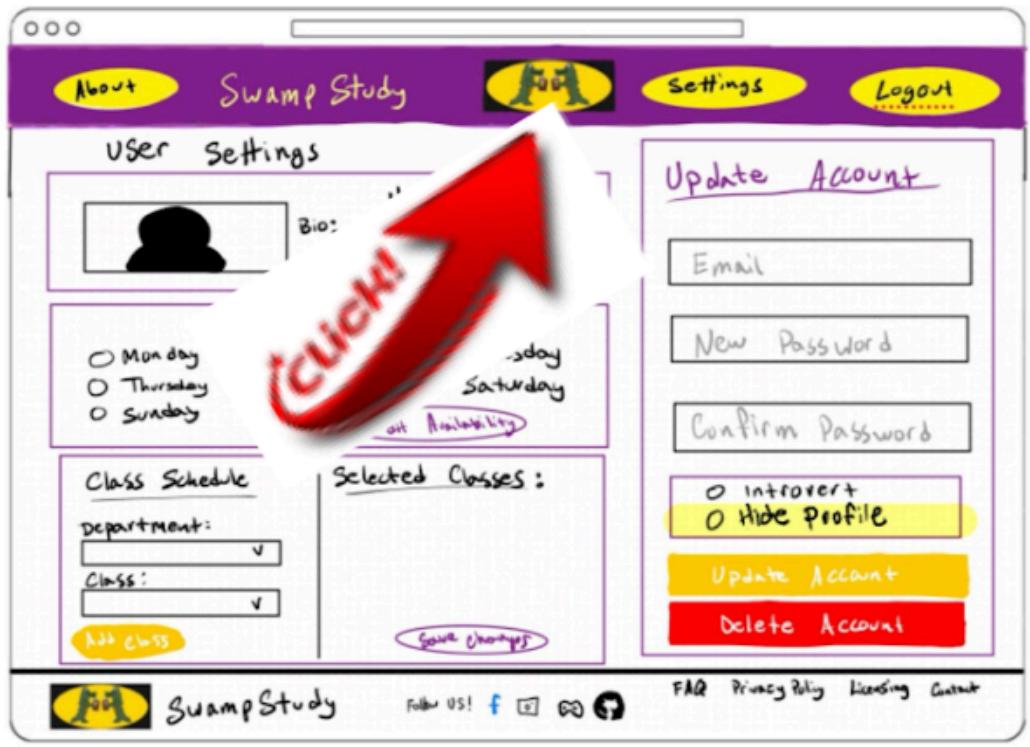
Forums!

About Swamp Study

Follow US! [f](#) [t](#) [g](#) [o](#)

FAQ Privacy Policy Licensing Contact





In Matching!

Forums

Name ★★★★☆

Availability

Monday	Yes
Tuesday	Yes
Wednesday	No
Thursday	No
Friday	No
Saturday	No
Sunday	No

Courses

- Course 1
- Course 2

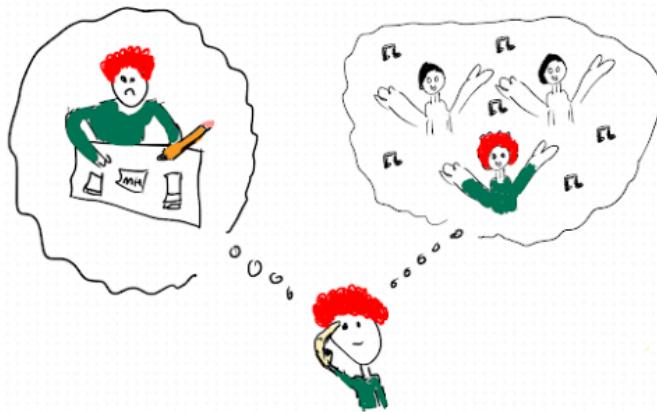
Your matches!

Matched with ---

Matched with ---

Follow US! [f](#) [y](#) [i](#) [t](#)

FAQ Privacy Policy Licensing Contact



Hiding his profile allows him to focus on himself, not on studying or social pressure.

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.

Welcome to StudySwamp!

We make finding your next study buddy easy!

Email

Password

Login

Don't have an account? Register

Click!

FAQ Privacy Policy Licensing Contact

About Swamp Study Settings Layout

Matching IN Forums!

Click!

Filters

Department: Math

Class: Calc 2

Search ...

Search

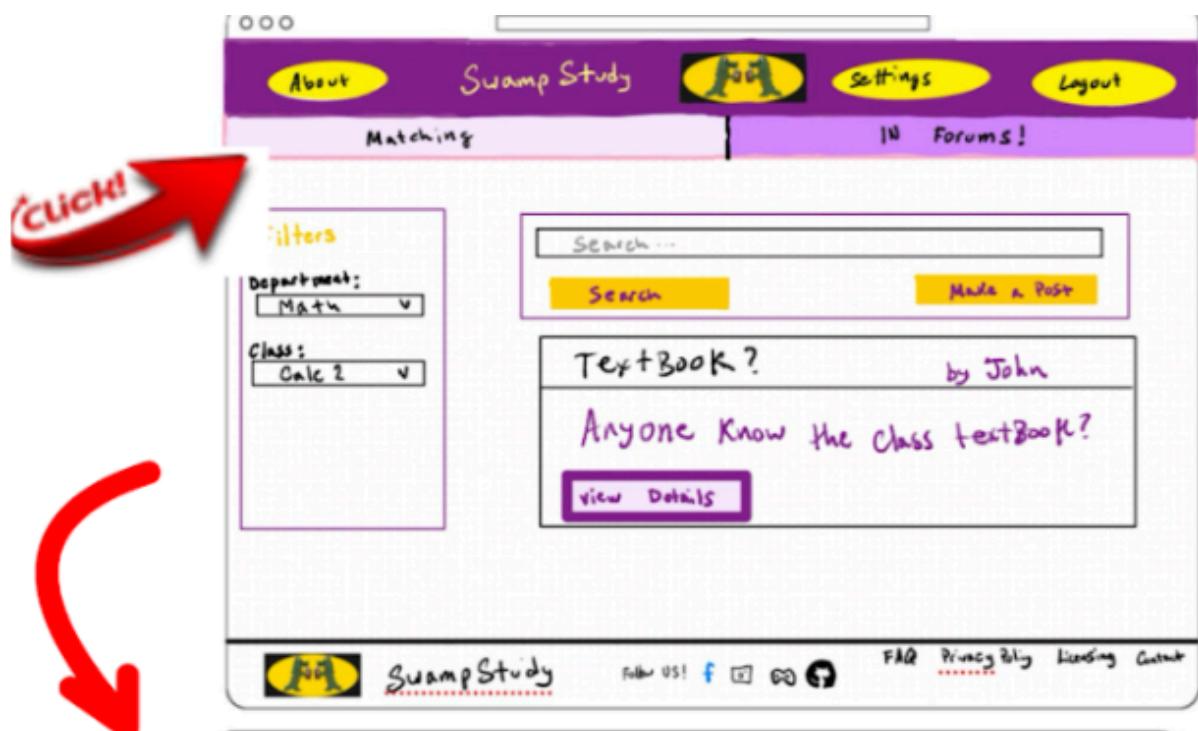
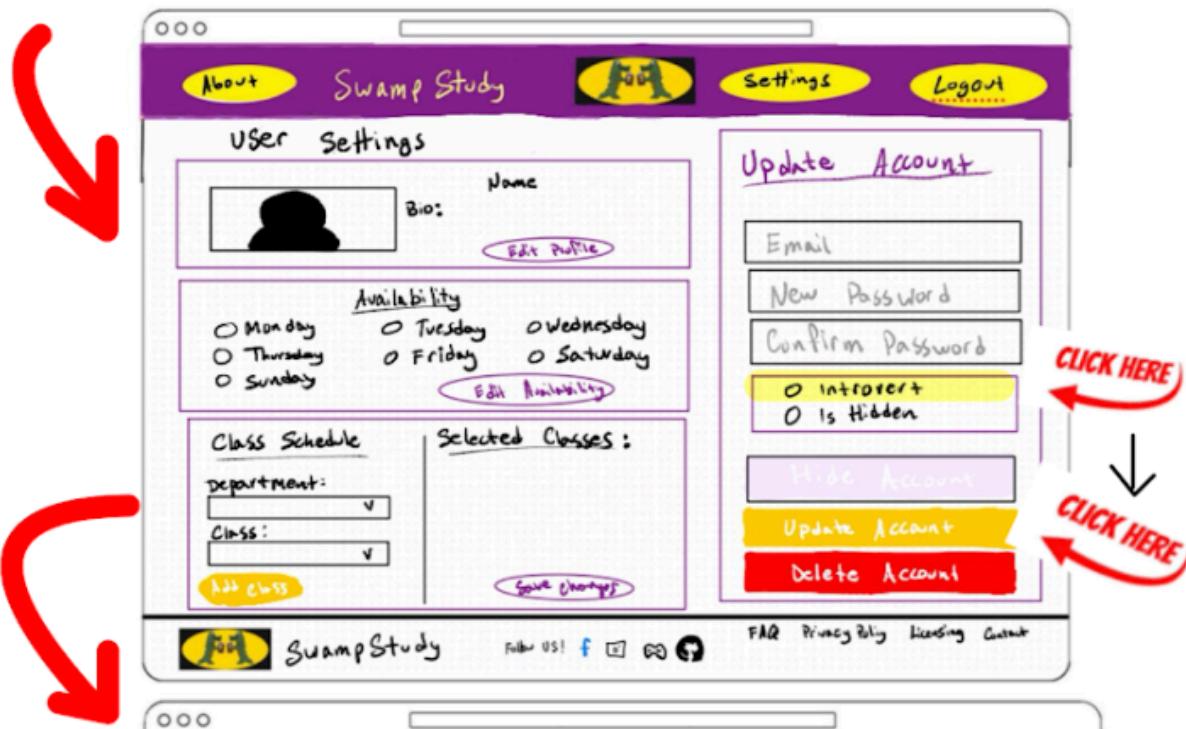
Made a Post

TextBook? by John

Anyone Know the class TextBook?

view Details

FAQ Privacy Policy Licensing Contact



Top Prototype: User Profile

Bottom Prototype: Match Details

Annotations:

- A large red curved arrow on the left side points to the user profile section.
- A green checkmark is placed next to the user's availability table.
- A red curved arrow on the right side points to the "Matched With" section.
- A red curved arrow at the bottom right points to the "Match Details" section.
- A red arrow labeled "CLICK HERE" points to the "Matched With" section.

ooo

About Swamp Study Settings Logout

In Matching!

Match Details

Name: 

Bio: **Introvert**

Rating: Email:

Meeting Details

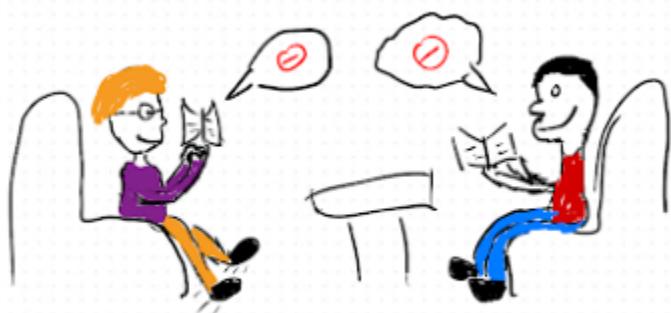
Date: Time:

Topic: Location:

Unmatch / Report Rate

Back

FAQ Privacy Policy Licensing Contact

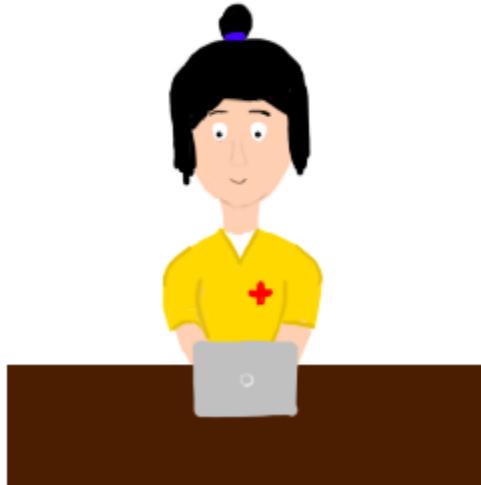


She is now free to study with others just like her, and avoid the uncomfortable small talk extroverts tend to do.

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.

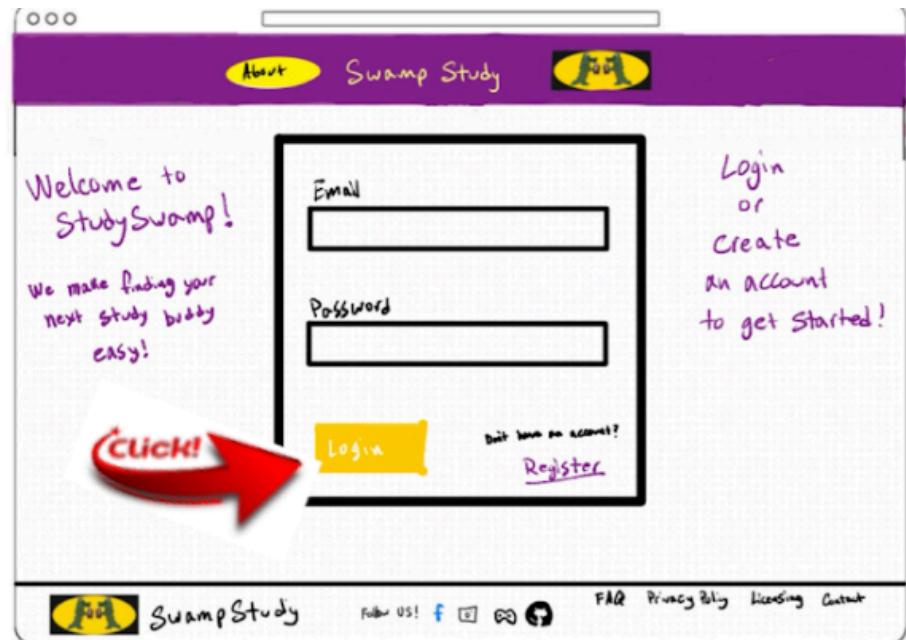


Michelle is finding Karen to be intollerable and is losing patience.



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





Click!

About Swamp Study Settings Logout

Matching | IN Forums!

Filters

Department:

Class:

Search ... Make a Post

TextBook? by John

Anyone Know the class textbook?

Follow US! [Facebook](#) [Twitter](#) [Instagram](#) [YouTube](#)

FAQ Privacy Policy Licensing Contact

CLICK HERE

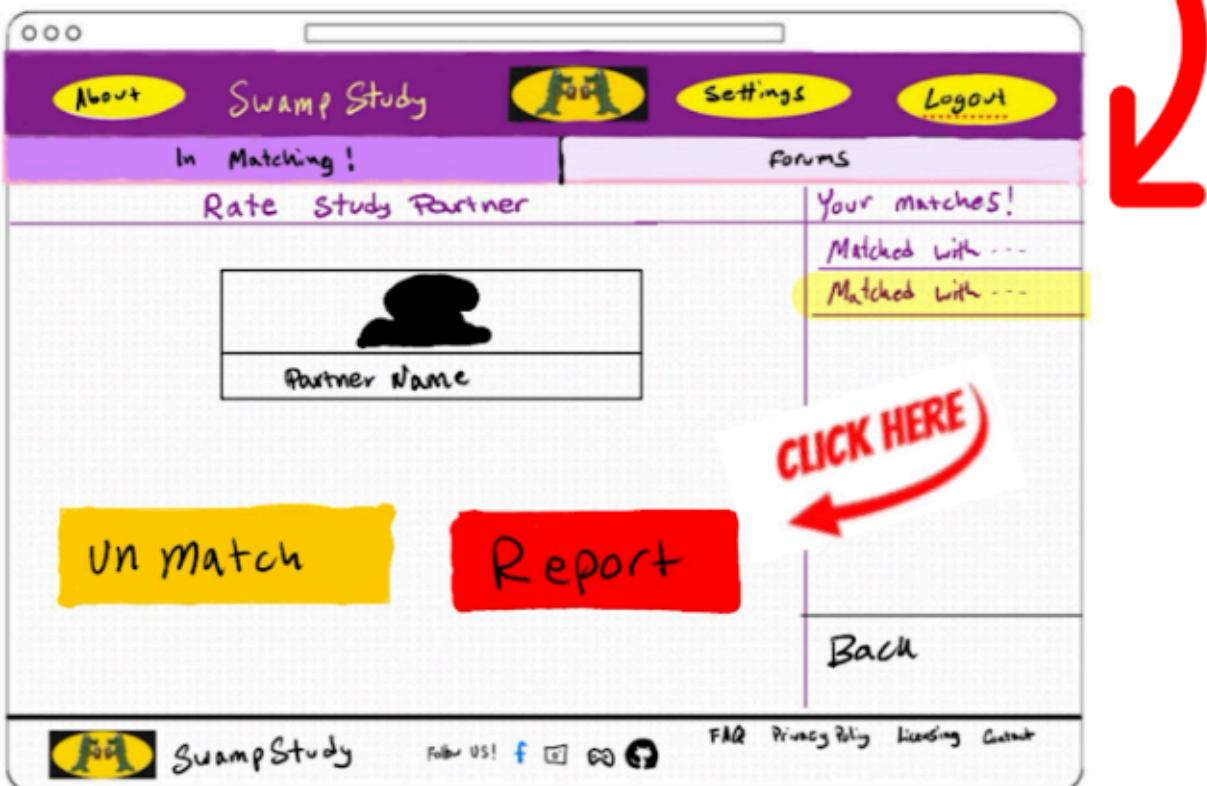
This screenshot shows the user's profile information. The profile picture is blacked out. The availability table shows the following days and availability:

Monday	Yes
Tuesday	Yes
Wednesday	No
Thursday	No
Friday	No
Saturday	No
Sunday	No

The courses listed are "course 1" and "course 2". To the right, under "Your matches!", there is a section titled "Matched with ..." which is currently empty.

Click!

This screenshot shows the "Match Details" screen. It includes fields for Name, Bio, Rating, Email, and Meeting Details (Date, Time, Topic, Location). At the bottom, there are two buttons: "Unmatch / Report" and "Rate". The "Rate" button is highlighted with a yellow background. On the right side, there is a "Back" button. The "Your matches!" section on the right is also visible.



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

Use Case 12 Storyboard



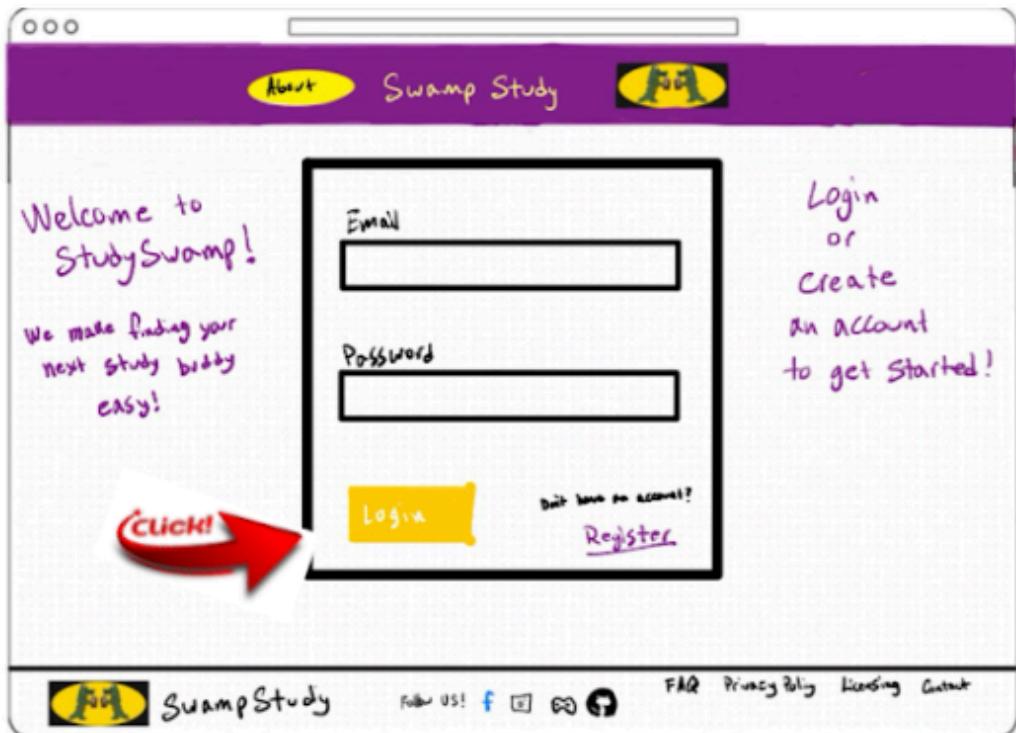
Geronimo missed the first week of an online class, he finds out there is a textbook that is required, but can't find which one is right.

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.



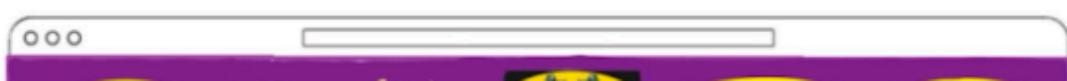


About Swamp Study  Settings Layout
Matching IN Forums!

Filters
Department:
Class:

Search ...
Search  Make a Post

Follow US!  FAQ Privacy Policy Licensing Contact



ooo

About Swamp Study Settings Logout

Matching IN Forums!

MAKE A POST

Department:

Class:

Title: Enter Title

Question: Type your Question here...

CANCEL **Submit** **CLICK HERE**

Swamp Study Follow US! FAQ Privacy Policy Licensing Contact



ooo

About Swamp Study Settings Logout

Matching IN Forums!

Filters

Department: Math

Class: Calc 2

Search ...

Search **MAKE A POST**

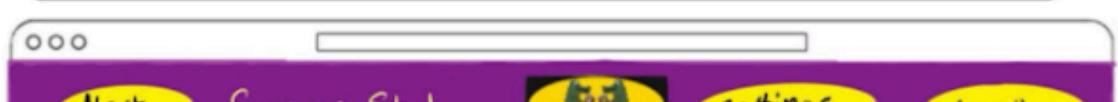
TextBOOK? by John

Anyone Know the class textbook?

view Details

Click!

Swamp Study Follow US! FAQ Privacy Policy Licensing Contact



ooo

About Swamp Study settings Logout

Return to Forum

Calc 2

Anyone Know the class textbook?

Jerry: yeah here is the link. <https://...>

Add your Answer... //

Submit

Swamp Study

Follow us! [f](#) [i](#) [t](#) [g](#) [p](#)

FAQ Privacy Policy Licensing Contact

Database Architecture

1. Database Requirements

1.1. Registered User (Weak)

- 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 zero or one account

1.2. Account (Strong)

- 1.2.1. An Account shall belong to one or zero 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 or zero profiles
- 1.2.5. An Account shall have many Ratings
- 1.2.6. An Account shall take many Classes

1.3. Match (Strong)

- 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 one meeting link
- 1.3.5. A Match shall have one meeting time

1.4. Question (Weak)

- 1.4.1. A Questions shall be posed by one Account
- 1.4.2. A Question shall have one or 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 one 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 one or 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
- 1.8.4. A Class shall be taken by many Users

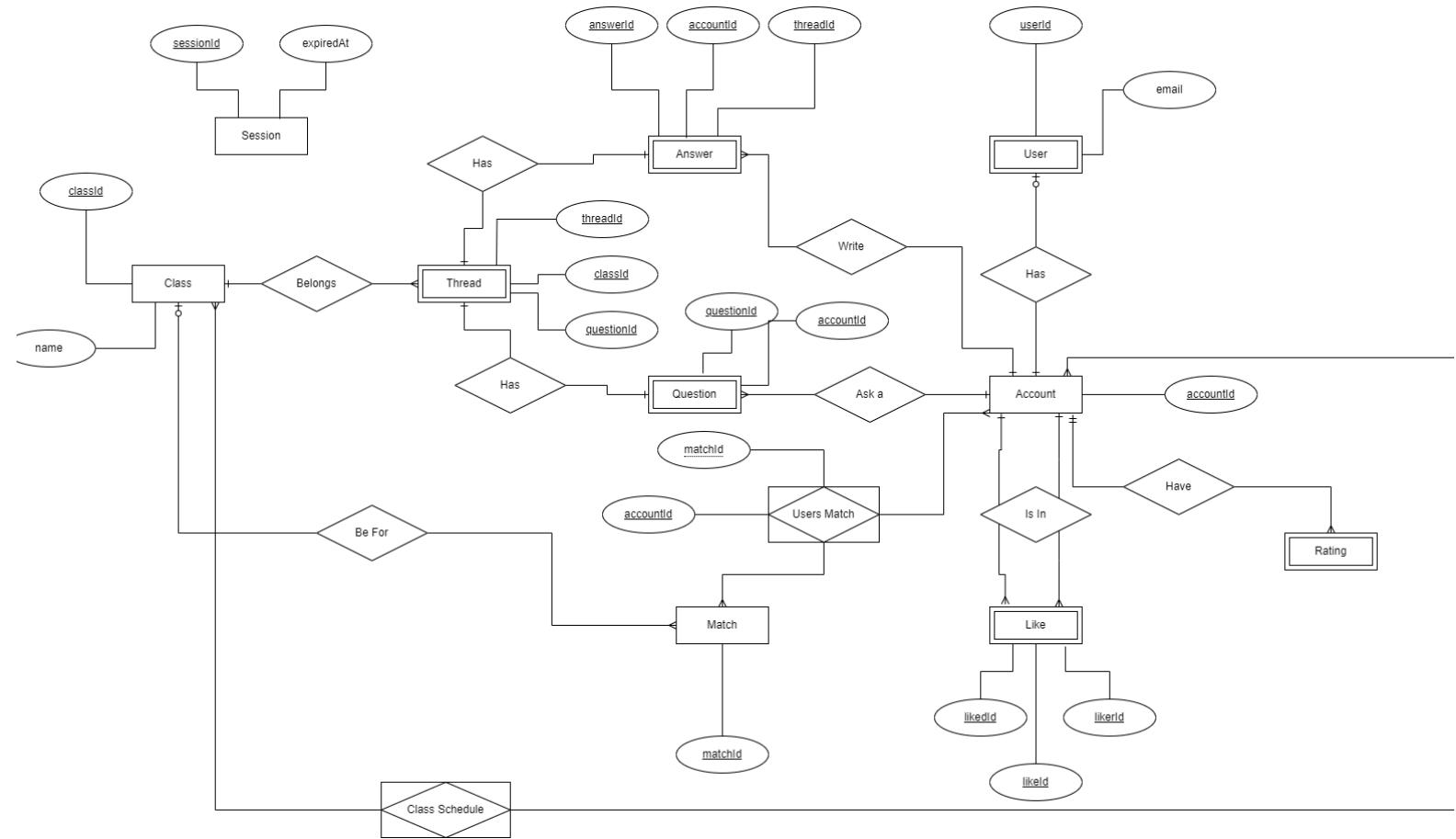
1.9. Profile (Weak)

- 1.9.1. A Profile shall belong to one account

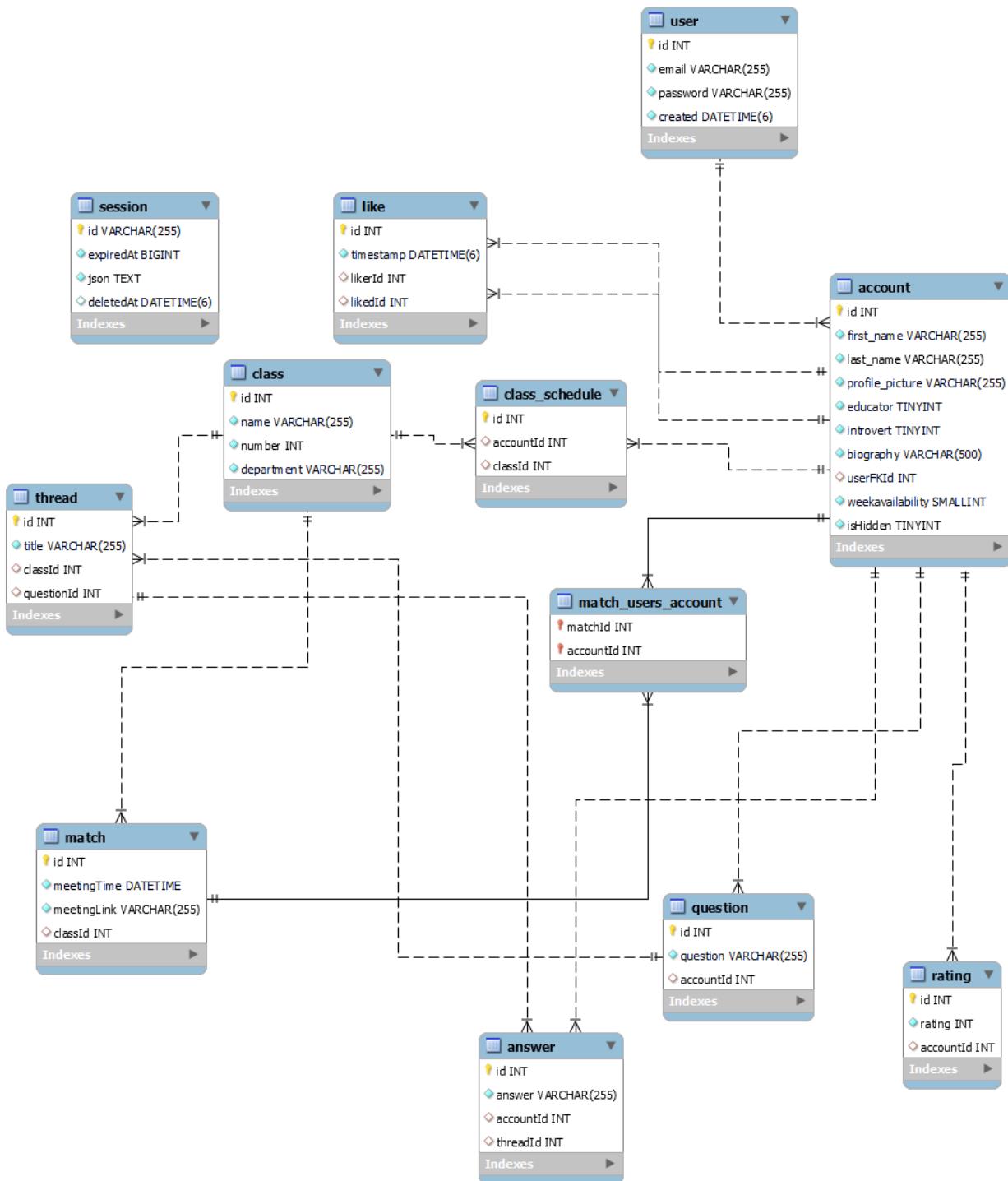
- 1.9.2. A Profile shall have one or many attributes
- 1.10. Like (Weak)
 - 1.10.1. A Like shall have one or more accounts
2. Entity Attributes, Relationships, and Domains
 - 2.1. user (Weak)
 - 2.1.1. id: PK, Numeric, Generated
 - 2.1.2. email: unique, alphanumeric
 - 2.1.3. password: alphanumeric, hashed
 - 2.1.4. created: datetime, generated
 - 2.1.5. account: Numeric, ForeignKey
 - 2.2. account (Strong)
 - 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. educator: tinyint (Boolean)
 - 2.2.6. introvert: tinyint (Boolean)
 - 2.2.7. biography: varchar(500)
 - 2.2.8. userFkId: Number, Foreign Key
 - 2.2.9. weekavailability: smallInt
 - 2.2.10. isHidden: tinyInt
 - 2.3. match (Strong)
 - 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. rating: Numeric, int
 - 2.11.3. accountId, int
- 2.12. classSchedule (weak)
 - 2.12.1. Id: PK, Numeric, Generated
 - 2.12.2. account: FK, Numeric Account ID
 - 2.12.3. class: FK, Numeric, Class ID

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

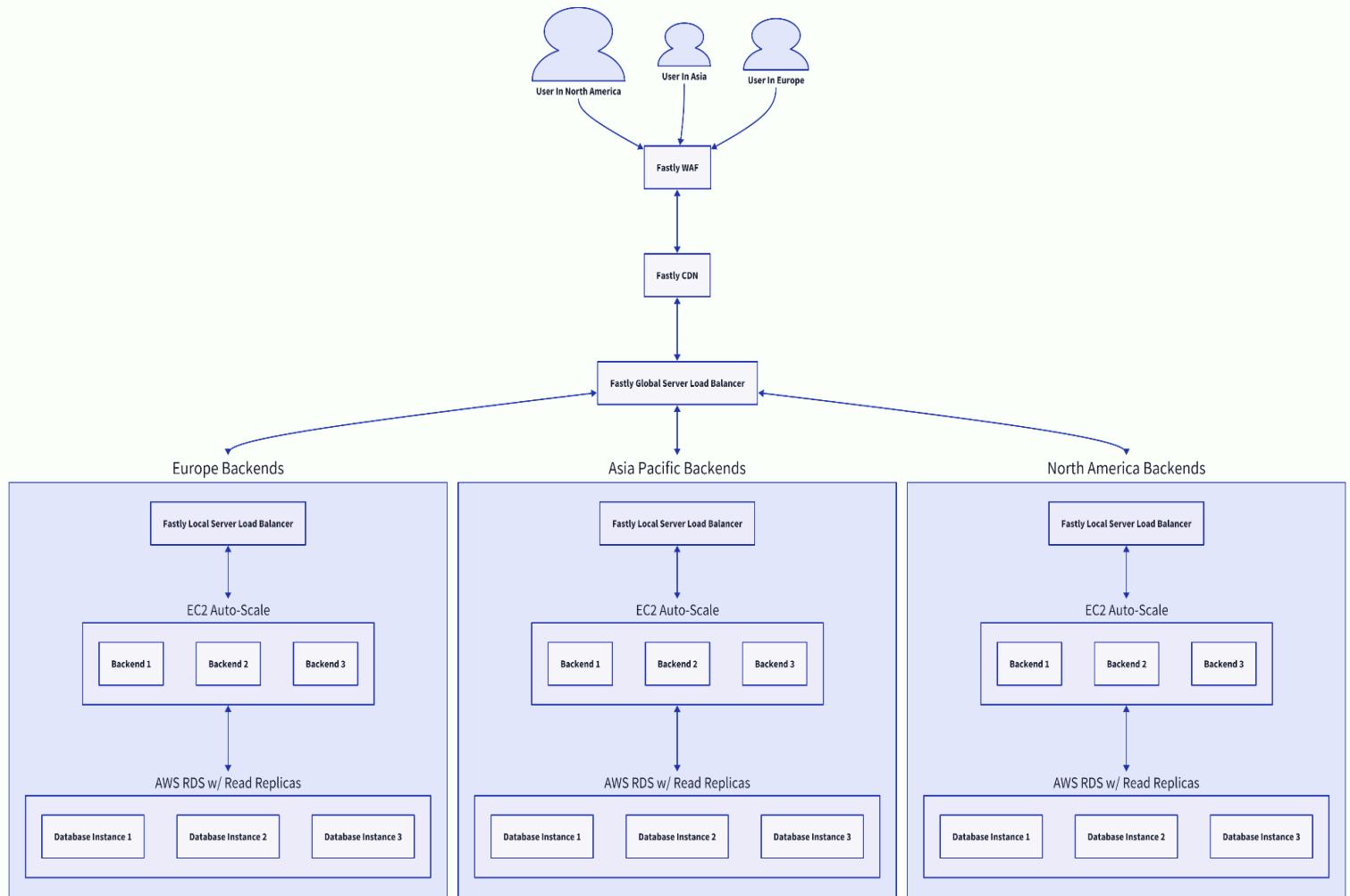
search box and then use a query with the ‘where’ keyword and regex wildcards to determine which posts fit the given criteria and then filter and display the results based on that keyword. We also scan the answers of each question as well in order to give a more comprehensive search. What gets returned 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:

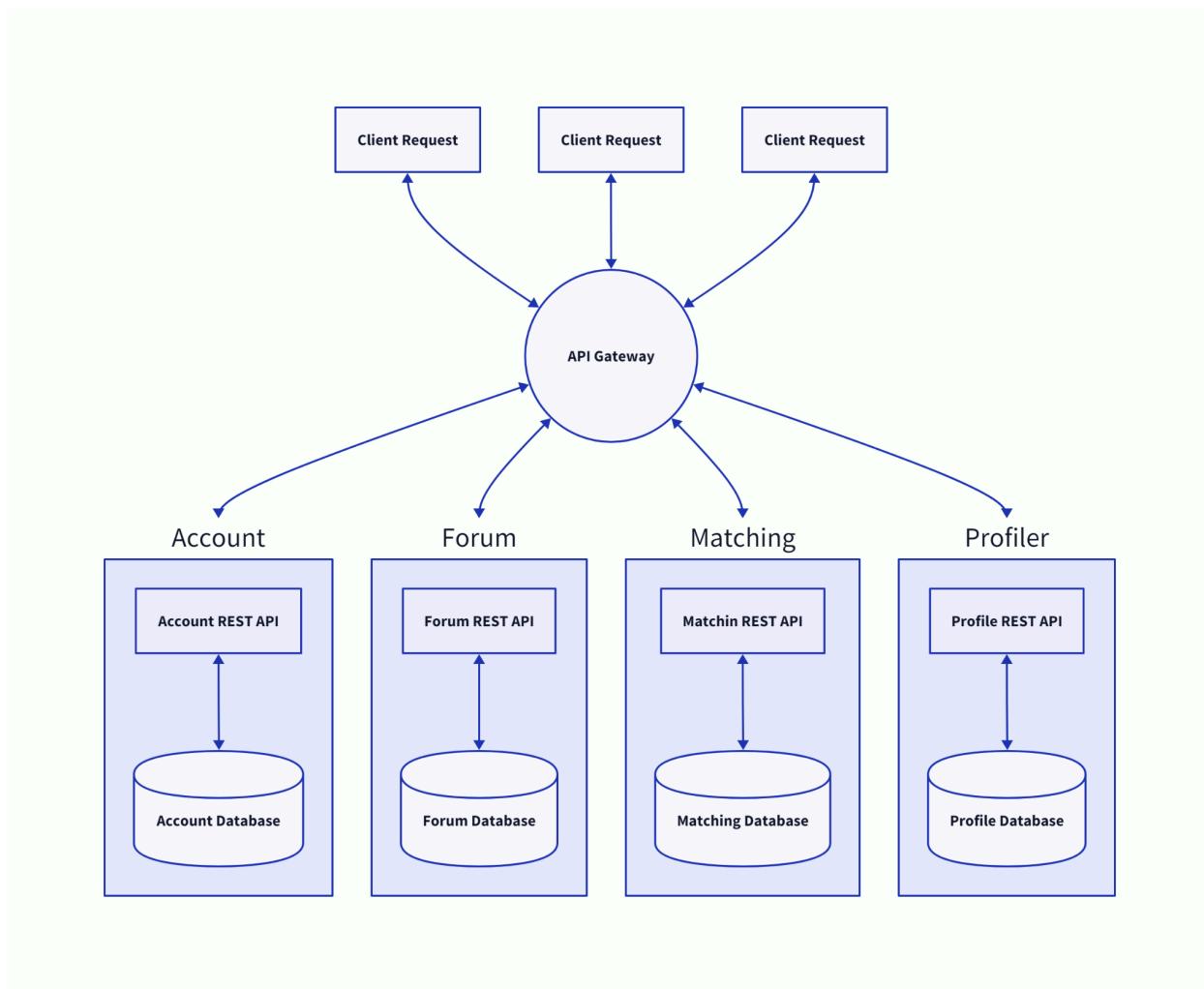
See the end of the Document for full size diagram

Scalability Diagram



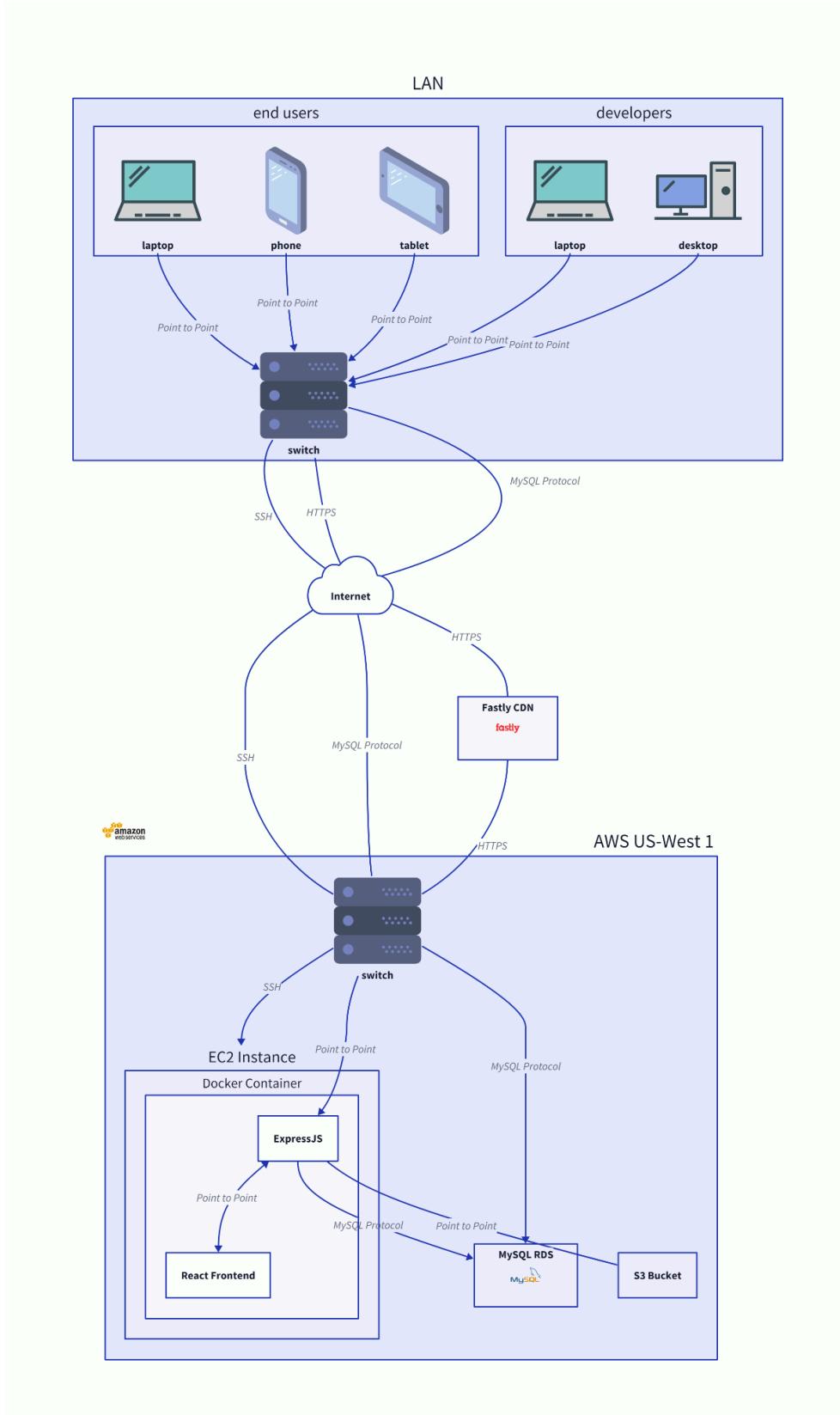
To let our service scale with ease, we first introduce a WAF at the top of the chain. This will allow us to filter malicious requests and bots trying to gain access to the system. We then run all the requests through our CDN which allows for request collapsing and to deliver cached content far quicker 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

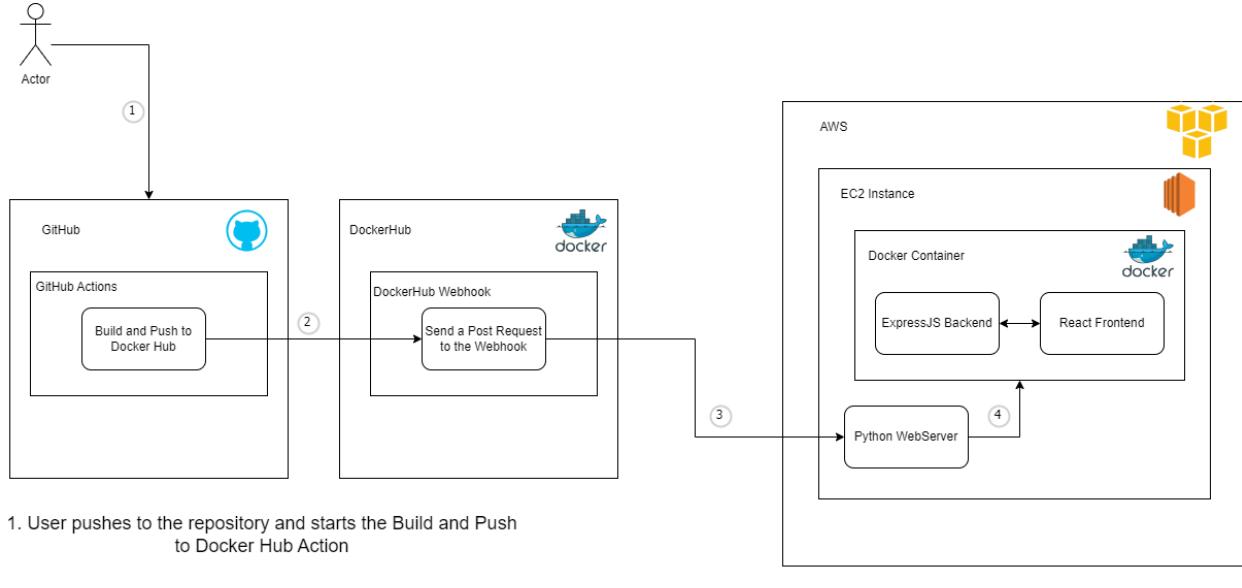


High Level Application Network and Deployment Design

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

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
Milestone 2 V2	Apr 24, 2024
Milestone 3	Apr 24, 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
Schedule.....	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: Students using the site need a partner for a group project.....	8
Use case 1 Storyboard:.....	8
Use case 3: Non-Traditional Student looking for Study Partner/ Availability.....	13
Use case 3 Storyboard:.....	13
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.....	21
Use case 5 Storyboard (FROM DANIEL'S PERSPECTIVE):.....	21
Use case 6: Hiding Profile.....	25
Use case 6 Storyboard.....	25
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 reporting.....	34
Use Case 8 Storyboard.....	34
Use case 12: User getting his questions answered on the forum.....	38
Use Case 12 Storyboard.....	38
Database Architecture.....	44
Entity Relationship Diagram.....	47
Entity Establishment Relation Diagram (EER).....	48
DBMS System.....	49
Media Storage.....	49
Searching the Database.....	49
High Level API's and Main Algorithms.....	51
Login API:.....	51
Logout API:.....	51
Register API:.....	51
Match API.....	51
Like API.....	52
Forum API:.....	52
Like and Match Algorithm.....	52
Session Algorithm.....	53
Search Algorithm.....	54
System Design.....	55
UML Class Diagram:.....	55
Scalability Diagram.....	56
Microservices Diagram.....	57
High Level Application Network and Deployment Design.....	58
Deployment Diagram.....	59
Team Evaluation.....	60
Lennart (Team Lead) 10.....	60
Rafael (Backend Lead) 10.....	60
Edmund (Frontend Lead) 10.....	60
Julio (Docs-Editor) 10.....	61
Conrad (Github-Admin) 10.....	61

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

Schedule

A Schedule is the mapping of which classes a User is taking. Each entry contains the user and the class

Functional Requirements

Priority 1: Must Have

1. User

- 1.1. A User shall be able to create an account
- 1.2. A User shall only have one account
- 1.3. A User shall be able to view other users' accounts in the system
- 1.4. A User shall be able to set their availability
- 1.5. A User shall be able to establish their profile
- 1.6. A User shall be able to ask many questions in the Forum
- 1.7. A User shall be able to answer many questions in the Forum
- 1.8. A User shall be able to like other Users
- 1.9. A User shall be able to match with other Users

2. Profile

- 2.1. A Profile shall be owned and edited by one and only one user
- 2.2. A Profile shall be allowed one picture of the owner
- 2.3. A Profile shall be allowed a short biography
- 2.4. A Profile shall have many subject attributes
- 2.5. A Profile shall be able to set a filter for their study partner preferences
- 2.6. A Profile shall be marked Student or Educator
- 2.7. A Profile marked Educator shall be able to access the Forum
- 2.8. A Profile marked Educator shall be allowed to answer questions in the Forum

3. Account

- 3.1. An Account shall have one email associated with it

4. User Interface

- 4.1. The Interface shall allow a User to navigate through a potential Match's profile
- 4.2. The Interface shall allow a User to access/edit their Profile
- 4.3. The Interface shall allow a User to access/edit their Account

5. Match

- 5.1. A Match shall be provided a meeting link
- 5.2. A Match shall be provided a meeting time
- 5.3. A Match shall be made based on different criteria

6. Forum

- 6.1. The Forum shall allow one main question per Thread
- 6.2. The Forum shall allow many answers until marked done
- 6.3. The Forum shall allow each class its section
- 6.4. The Forum shall allow users to post in their class Forum

6.5. The Forum shall allow many questions

Priority 2: Desired

1. Profile

- 1.1. A Profile shall be allowed to be invisible when not wanting to study
- 1.2. A Profile marked Educator shall be allowed to remove questions from the Forum
- 1.3. Send a verification email to users that attempt to set their account as educator
- 1.4. Add a forgot password page to allow users to reset forgotten passwords
- 1.5. let the user pick a unique username to display on the site

2. User

- 2.1. A User shall be able to flag another user for review
- 2.2. A User shall have a match rating which determines visibility
- 2.3. A User shall review a match after their first meeting
- 2.4. A User shall be allowed to block another User

3. User Interface

- 3.1. The Interface shall allow a user to swipe right or left on a potential study partner
- 3.2. Add a faQs page that has site information on it
- 3.3. Add a footer with links to terms of service, privacy policy, licensing
- 3.4. Replace JS alert with toastify notifications
- 3.5. Have submitted posts include a hyperlink to the users profile page
- 3.6. Log a user in after the submit password and email in step one of registration and redirect them to the settings page to set more profile settings
- 3.7. View other users who are currently online
- 3.8. Add a submit help ticket page to allow users to submit help tickets
- 3.9. Allow users to report users they have not matched with

4. Matching System

- 4.1. The Matching System shall highlight common attributes users have with each other
- 4.2. The Matching System shall notify users when they have a Match
- 4.3. The Matching System shall send a notification to the user's device
- 4.4. The Matching System shall show a user their ideal match
- 4.5. The Matching System shall let users decide whether they want to meet in person or online
- 4.6. Allow users to unmatched / delete matches

5. Forum

- 5.1. The Forum shall limit access to students taking that class
- 5.2. The Forum shall allow educators to access all the Threads
- 5.3. Add a created time stamp to posts
- 5.4. Add a created time stamp to submitted answers
- 5.5. Let users delete their posts
- 5.6. Let users delete their submitted answers
- 5.7. Let users delete answers made to their posts
- 5.8. Add a general section where users can free post about events, extracurriculars and general topics
- 5.9. Track the views made to a post
- 5.10. Suggest the top ten most viewed posts when a user's search does not have results

Priority 3: Opportunistic

1. Account

- 1.1. An Account shall be allowed to log in from one device at a time
- 1.2. An account shall be able to link their external social media accounts on their profile

2. User

- 2.1. A User shall be able to Video Call with their partner
- 2.2. Add user to user messaging

3. Profile

- 3.1. Prescale images for faster loading
- 3.2. Verify emails by sending the user a verification email

Use Cases

Priority 1: Must Have

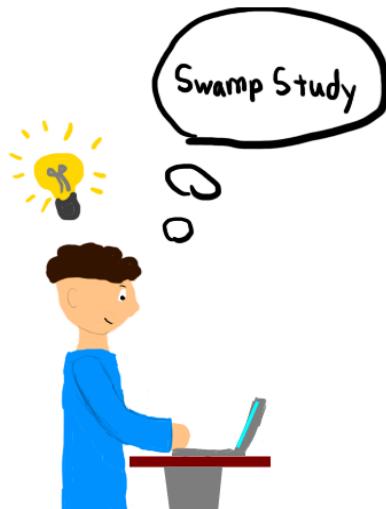
Use case 1: Students using the site need a partner for a group project.

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.



The image displays two screenshots of the SwampStudy website, illustrating the user account creation process.

Screenshot 1: Login Page

Welcome! Finding study buddies made easy!

Email:
Password:

Login Don't have an account? Register

CLICK HERE

Screenshot 2: Registration Page

First Name: Last Name:
Email:
Password: Confirm Password:

I have read and accept the [Terms and Conditions](#)

Click!

Register Already have an account? [Login](#)

About **SwampStudy** About Privacy Policy Licensing Contact

The image displays two screenshots of the SwampStudy website, connected by a large red curved arrow pointing from the top screenshot down to the bottom one.

Top Screenshot (Matching Page):

- Header:** SwampStudy, About, Settings, Logout.
- Search Bar:** Search for threads... (with a yellow "Search" button), Make a Post (with a yellow "Post" button).
- Form Fields:** Department: General Chat, Class: College Help.
- Post Preview:** Graduation tickets, Asked By: Sarah, Does anyone have any extra graduation tickets???, View Details.

Bottom Screenshot (Profile Settings Page):

- Header:** SwampStudy, About, Settings, Logout.
- Left Column (Profile Settings):**
 - Profile Picture: Click!
 - Name: [Input field]
 - Bio: [Input field]
 - Rating: ★★★★★
 - Availability: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday (each with a switch)
 - Current Courses: [Input field]
- Right Column (Account Settings):**
 - Update Account:
 - First Name: [Input field]
 - Last Name: [Input field]
 - Email: [Input field]
 - New Password (optional): [Input field]
 - Confirm New Password: [Input field]
 - Hide Account (switch)
 - Introverts only (switch)
 - Update Account (yellow button)
 - Delete Account (button)

Common Elements: SwampStudy logo, follow us! (Facebook, Instagram, Twitter, YouTube), About, Privacy Policy, Licensing, Contact.

The screenshot shows the SwampStudy matching interface. At the top, there are navigation links: 'About' (yellow), 'SwampStudy' (large yellow text), 'Settings' (yellow), and 'Logout' (yellow). Below this, a purple bar indicates 'Currently in Matching'. The main area has a heading 'Find your next study partner'. On the left, there is a red 'X' icon. In the center, a profile card displays a 'Profile picture' placeholder, 'Name' and 'Rating' columns, 'Is Available:' status, and 'Courses Studying:'. To the right, a 'Your Matches' section is shown with a green checkmark and a white box containing a red arrow pointing to the text 'CLICK HERE'.

SwampStudy About Privacy Policy Licensing Contact

follow us! [f](#) [g](#) [t](#) [q](#)

This screenshot shows the same matching interface after a match has been made. The 'Your Matches' section now displays a yellow box with the text 'You matched with [Name]'. The profile card on the left includes 'Availability' details for the user and their partner. The right side shows 'Meeting Details' fields: Date, Time, Location, Study Course, and Email, along with a 'Rate Partner' button. A 'Back to Matches' link is at the bottom.

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

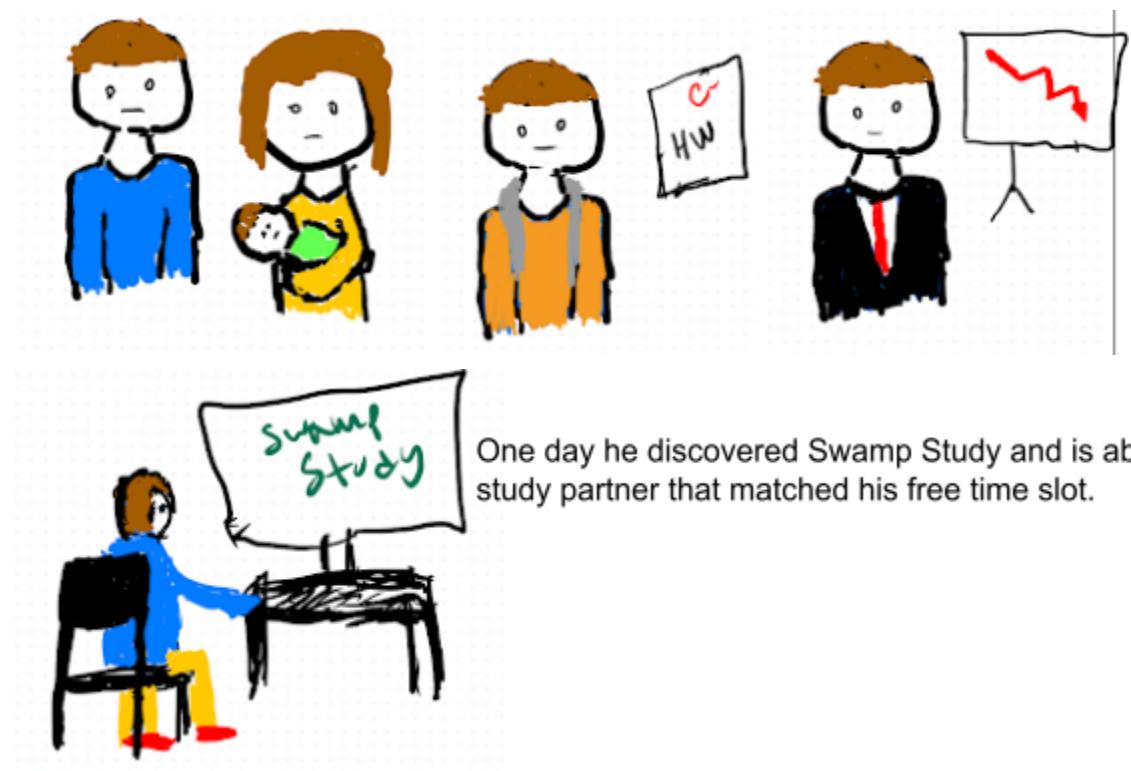


Use case 3: Non-Traditional Student looking for Study Partner/ Availability

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



Welcome!
Finding
study buddies
made easy!

Email:
Password:

Login

Don't have an account?
Register

SwampStudy

click!

About Privacy Policy Licensing Contact

follow us!

About

Matching

Currently in

Graduation tickets

Does anyone have any extra graduation tickets???

Asked By: Sarah

View Details

Search for threads...
Search

Department:
General Chat: v

Class:
College Help: v

click!

Logout

Settings

About Privacy Policy Licensing Contact

follow us!

The image displays two screenshots of the SwampStudy website, showing different sections of the platform.

Top Screenshot: Profile Settings and Account Settings

This screenshot shows the "Profile Settings" and "Account Settings" sections. The "Profile Settings" section includes fields for Name, Bio, Profile Picture (with an "upload image" button), Availability (checkboxes for Monday through Sunday), and Current Courses. The "Account Settings" section includes fields for First Name, Last Name, Email, New Password (optional), Confirm New Password, Hide Account (checkbox), Introverts only (checkbox), Update Account (button), and Delete Account (button). Red arrows labeled "Click!" point to the "upload image" button in the profile picture section, the "Edit Bio" button, the "Update Account" button, and the "Delete Account" button.

Bottom Screenshot: Matching and Forums

This screenshot shows the "Matching" and "Currently in Forums" sections. The "Matching" section includes a dropdown for "General Chat" and "Class" (set to "College Help"). The "Currently in Forums" section features a search bar ("Search for threads...") with a "Search" button and a "Make a Post" button. A thread titled "Graduation tickets" is shown, asking "Does anyone have any extra graduation tickets???", posted by "Sarah". A red arrow labeled "Click!" points to the "View Details" link in the forum thread.

Common UI Elements:

- Header:** SwampStudy logo, About, Settings, Logout.
- Footer:** SwampStudy logo, follow us! (Facebook, Instagram, YouTube, Twitter icons), About, Privacy Policy, Licensing, Contact.

SwampStudy

Currently in Matching

Forums

Find your next study partner

Name Rating

Profile picture

Is Available: Courses Studying:

X ✓

Your Matches

Rate your previous study partner

View Matches

CLICK HERE

SwampStudy

follow us!

SwampStudy

Currently in Matching

Forums

Meeting Details with [Name]

Profile picture

Date:
Time:
Location:
Study Course:
Study Partner:

Availability:
Mon
Tue
Wed
Thu
Fri
Sat
Sun

Courses Studying:

Unmatch

Your Matches

Rate your previous study partner

You matched with [Name]

Back to Matches

SwampStudy

follow us!



Now he can stop worrying so much about school and focus on his job and family. Everyone is happier.

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.



Welcome!
Finding
study buddies
made easy!

Email: _____

Password: _____

Login

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

click!

SwampStudy About Privacy Policy Licensing Contact

follow us!

click!

Matching **Currently in Forums**

Department: General Chat:

Class: College Help

Search for threads...

Graduation tickets

Does anyone have any extra graduation tickets???

View Details

SwampStudy About Privacy Policy Licensing Contact

follow us!

The screenshot shows a user profile card on the left and a 'Your Matches' section on the right. A large red arrow on the left points to the user profile card, which has a red 'X' over it. A green checkmark is placed above the 'Your Matches' section. A red box with the text 'CLICK HERE' and a curved arrow points to the 'View Matches' button.

SwampStudy

Currently in Matching

Forums

About

Settings

Logout

Find your next study partner

Name Rating

Profile picture

Is Available:

Courses Studying:

View Matches

Rate your previous study partner

Your Matches

CLICK HERE

The screenshot shows a 'Meeting Details with [Name]' section on the left and a 'Your Matches' section on the right. A large red arrow on the right points to the 'Your Matches' section, which displays a purple box indicating a match. A yellow box highlights the 'Email:' field in the meeting details section.

SwampStudy

Currently in Matching

Forums

About

Settings

Logout

Meeting Details with [Name]

Date:

Time:

Location:

Study Course:

Study Partner:

Email:

Unmatch

Your Matches

You matched with [Name]

Back to Matches

Rate your previous study partner

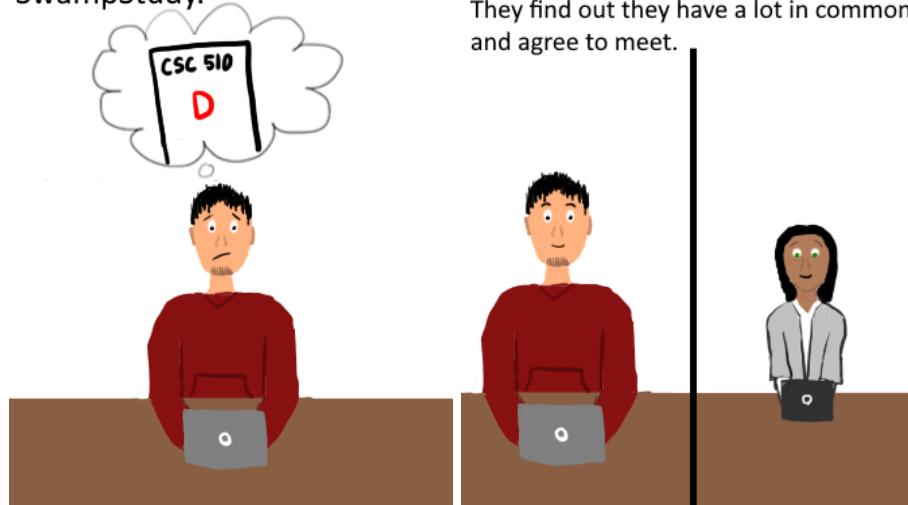
About Privacy Policy Licensing Contact

follow us:

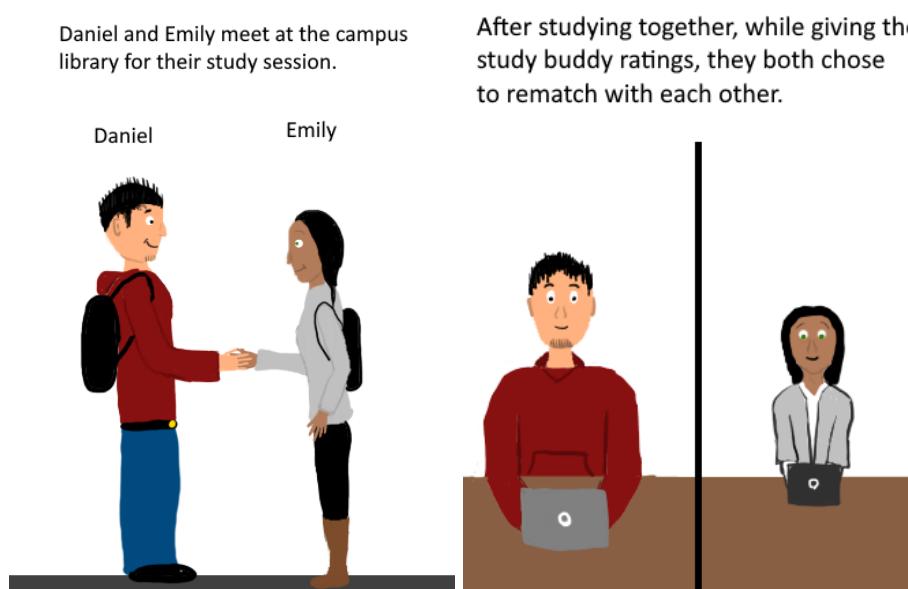
Use case 5: Finding a long-term study partner

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.

The image displays three screenshots of the SwampStudy website, each with red arrows indicating user interactions:

- Screenshot 1: Login Page**
 - Welcome!** Finding study buddies made easy!
 - Email:** [Input Field]
 - Password:** [Input Field]
 - Login** (Yellow button)
 - Don't have an account?**
 - Register**
- Screenshot 2: Matching Page**
 - Matching**
 - Currently in Forums**
 - Department:** General Chat
 - Class:** College Help
 - Search for threads...**
 - Search**
 - Make a Post**
 - Graduation tickets**
 - Asked By: Sarah**
 - Does anyone have any extra graduation tickets???**
 - View Details**
- Screenshot 3: Forum Thread Page**
 - SwampStudy**
 - About** **Privacy Policy** **Licensing** **Contact**
 - follow us!**

SwampStudy

About  Settings Logout

Currently in Matching Forums

Find your next study partner

Name Rating

Profile picture

Is Available: Courses Studying:

✓ **CLICK HERE**

Your Matches

Rate your previous study partner

View Matches

SwampStudy

About Privacy Policy Licensing Contact

follow us!    



SwampStudy

About  Settings Logout

Currently in Matching Forums

You matched with [Name]

Name Rating

Profile picture

Availability: Courses Studying:

Mon : Thu : Fri : Sat : Sun :

Meeting Details

Date: _____

Time: _____

Location: _____

Study Course: _____

Email: _____

Rate Partner

Your Matches

You matched with [Name]

Back to Matches

SwampStudy

About Privacy Policy Licensing Contact

follow us!    





Currently in Matching

Forums

About SwampStudy Settings Logout

Rate your previous study partner [Name]

How would you rate your study partner?

[Name]

★ ★ ★ ★ ★

would you study with this person again??

Submit

Your Matches

You matched with [Name]

Back to Matches

About Privacy Policy Licensing Contact

SwampStudy

follow us!

Use case 6: Hiding Profile

Use case 6 Storyboard



Kyle is a social butterfly, he craves company. He is also very studious, he is a regular SwampStudy user. But sometimes he is overwhelmed.

He needs the app but he also needs a break.

He is happy to see StudySwamp has a "Hide your Profile" feature.

About

SwampStudy



Welcome!
Finding
study buddies
made easy!

Email:
Password:

Login

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

SwampStudy

click!

Follow us!

About Privacy Policy Licensing Contact

About

SwampStudy



Matching

Currently in...

click!

Search for threads... Search

Graduation tickets

Does anyone have any extra graduation tickets???

Asked By: Sarah

View Details

Department: General Chat: v

Class: College Help: v

SwampStudy

click!

Follow us!

About Privacy Policy Licensing Contact

SwampStudy

About Settings Logout

Profile Settings

Name:

Profile Picture

upload image

Bio:

Edit Bio

Rating: ★★★★★

Availability

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Current Courses

SwampStudy

follow us! [f](#) [i](#) [t](#) [g](#)

Account Settings

Update Account

First Name:

Last Name:

Email:

New Password (optional):

Confirm New Password:

Hide Account

CLICK HERE 

Update Account

Delete Account

About Privacy Policy Licensing Contact



SwampStudy

About Settings Logout

Profile Settings

Name:

Profile Picture

upload image

Bio:

Edit Bio

Rating: ★★★★★

Availability

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Current Courses

SwampStudy

follow us! [f](#) [i](#) [t](#) [g](#)

Account Settings

POP UP !!!

ACCOUNT IS HIDDEN
WE WILL NOT BE ABLE TO MATCH

OK

New Password (optional):

Confirm New Password:

Hide Account Introverts only

CLICK HERE 

Update Account

Delete Account

About Privacy Policy Licensing Contact



SwampStudy

About  Settings Logout

Profile Settings

Name:

Profile Picture: 

Bio:

Rating: 

Availability:

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
<input checked="" type="radio"/>						

Current Courses:

Account Settings

POP UP !!!
ACCOUNT IS HIDDEN
WE WILL NOT BE ABLE TO MATCH

New Password (optional):

Confirm New Password:

Hide Account Introverts only

 SwampStudy About Privacy Policy Licensing Contact

Follow us!    

SwampStudy

About  Settings Logout

Currently in Matching Forums

Find your next study partner

	Name	Rating	
Profile picture			
Is Available:		Courses Studying:	

<

Your Matches

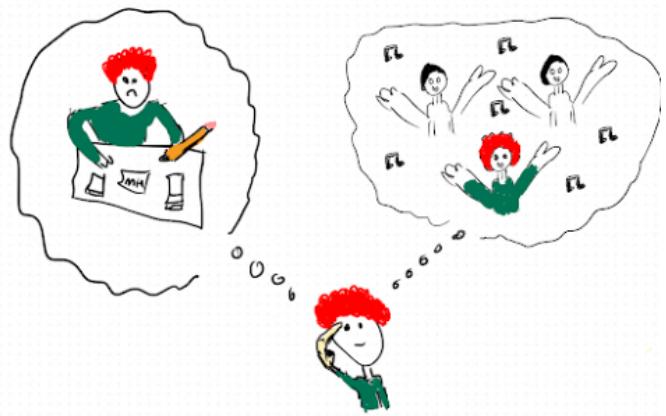
Rate your previous study partner

>

 SwampStudy About Privacy Policy Licensing Contact

Follow us!    





Hiding his profile allows him to focus on himself, not on studying or social pressure.

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.

About

SwampStudy



Welcome!
Finding
study buddies
made easy!

Email:
Password:

Login

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

SwampStudy

click!

About Privacy Policy Licensing Contact

follow us!

About

SwampStudy



Matching

Currently in...

Settings Logout

Department: General Chat: v

Class: College Help: v

Search for threads... Search

Graduation tickets

Does anyone have any extra graduation tickets???

Asked By: Sarah

View Details

click!

About Privacy Policy Licensing Contact

follow us!

The screenshot shows the SwampStudy website interface. At the top, there are navigation links: 'About' (highlighted), 'Settings', and 'Logout'. The main title 'SwampStudy' is displayed prominently. On the left, the 'Profile Settings' section includes fields for 'Name' (with a placeholder 'Profile Picture' and a 'upload image' button), 'Bio' (with a multi-line text area and an 'Edit Bio' button), and 'Availability' (with a grid of days from Monday to Sunday, each with a red toggle switch). Below these is a 'Rating' section with five yellow stars. A large red arrow points from the bottom center towards the 'Edit Bio' button. On the right, the 'Account Settings' section contains fields for 'First Name', 'Last Name', 'Email', and 'Phone' (labeled as optional). There is also a 'Confir.' field, a 'Hide Ac.' button, and a toggle switch for 'Introverts only'. Below these are 'Update Account' and 'Delete Account' buttons. A second red arrow points from the bottom center towards the 'Delete Account' button. At the bottom, there is a logo for 'SwampStudy' and social media links for Facebook, Instagram, and YouTube. Navigation links at the very bottom include 'About', 'Privacy Policy', 'Licensing', and 'Contact'.



A screenshot of the SwampStudy website. At the top, there's a purple navigation bar with yellow circular icons for 'About', 'Settings', and 'Logout'. The main title 'SwampStudy' is in large yellow letters. Below the title, a grey bar says 'Matching' and 'Currently in Forums'. The main content area has a white header with a purple search bar containing 'Search for threads...'. Below the search bar is a yellow button labeled 'Search' and a blue button labeled 'Make a Post'. A purple box contains a post from 'Sarah' asking if anyone has extra graduation tickets. A large red arrow points from the left towards the search bar. The bottom of the page features a yellow footer with the 'SwampStudy' logo, social media links for Facebook, Instagram, and YouTube, and links for 'About', 'Privacy Policy', 'Licensing', and 'Contact'.

[About](#)  **SwampStudy**  [Settings](#) [Logout](#)

Currently in Matching [Forums](#)

Find your next study partner

	Name	Rating	
	Profile picture		Introvert
Is Available:	Courses Studying:		

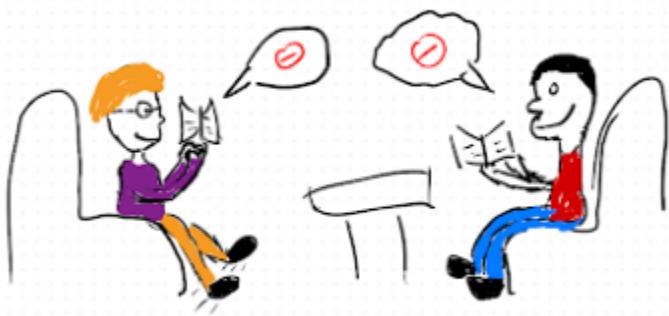


Your Matches

 **SwampStudy**

About Privacy Policy Licensing Contact

follow us!    



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

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

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.



[About](#) SwampStudy 

Welcome! Finding study buddies made easy!

Email:
Password:

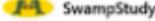
Login Don't have an account?
[Register](#)

About Privacy Policy Licensing Contact
follow us!    

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

Matching Currently in Forums

Search for threads... **Search** [Create a Post](#)

Class: College Help

Graduation tickets Asked By: Sarah
Does anyone have any extra graduation tickets???

[View Details](#)

About Privacy Policy Licensing Contact
follow us!    

The screenshot shows the SwampStudy website interface. At the top, there is a purple header bar with the following elements from left to right: a yellow 'About' button, the 'SwampStudy' logo (a green frog icon), a yellow 'Forums' button, a yellow 'Settings' button, and a yellow 'Logout' button. Below the header, the page title 'Currently in Matching' is displayed. To the right of the title is a green checkmark icon followed by the text 'Your Matches' and 'Rate your previous study partner'. A large rectangular box contains a user profile with fields for 'Name', 'Rating', 'Profile picture', 'Is Available:', and 'Courses Studying:'. To the left of this box is a red 'X' icon. To the right is a green checkmark icon followed by a red arrow pointing to a 'CLICK HERE' button. At the bottom right of the profile box is a 'View Matches' button. Along the bottom of the page are several links: 'About', 'Privacy Policy', 'Licensing', and 'Contact'. On the far left, there is a large red curved arrow pointing towards the profile box. At the very bottom, there is a footer bar with the 'SwampStudy' logo, social media icons for Facebook, Instagram, and YouTube, and the text 'follow us!'. There is also a small 'f' icon near the bottom center.

The image displays two screenshots of the SwampStudy website, illustrating the reporting process for a matched user.

Screenshot 1: Initial Match View

This screenshot shows the user interface after a match has been made. The main content area displays the matched user's profile picture, name, rating, availability, and study details. A red arrow points to the "Report" button, which is highlighted with a red box and labeled "Click".

Name	Rating
[Profile Picture]	[Rating]

Screenshot 2: Reporting Confirmation

This screenshot shows the result of the reporting action. A purple modal dialog box is displayed, stating "USER has been reported. You have been unmatched." An "OK" button is visible at the bottom right of the dialog. A large red curved arrow on the left side of the image points from the first screenshot down to the second, indicating the progression of the action.

User Profile Details (Visible in Both Screenshots)

Availability:	Profile picture	Rating:	Meeting Details
Mon : Tue : Wed : Thu : Fri : Sat : Sun :	[Profile Picture]	[Rating]	Date: Time: Location: Study Course: Email:

Buttons (Visible in Both Screenshots)

- Report (Red button)
- Rate Partner (Yellow button)
- Back to Matches (Grey button)

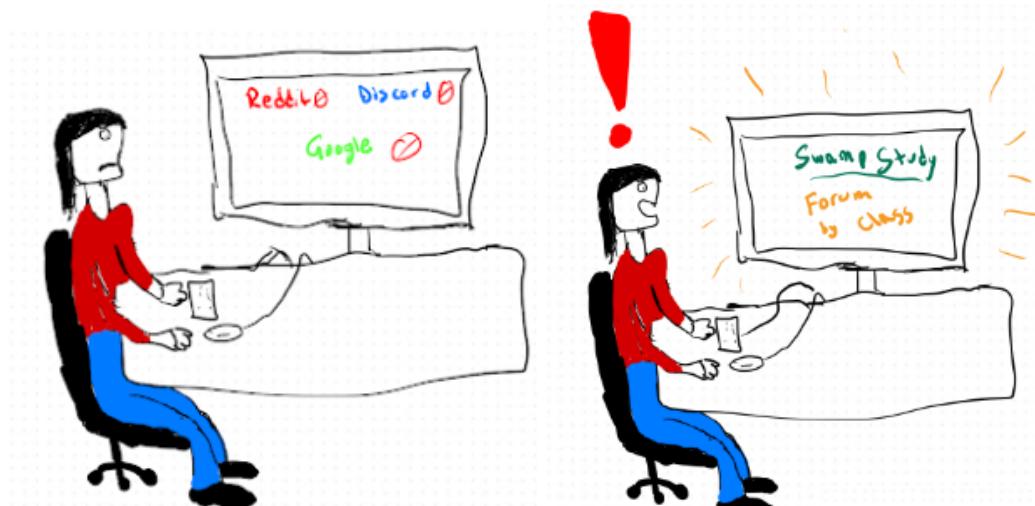
Use case 12: User getting his questions answered on the forum

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.



Welcome!
Finding
study buddies
made easy!

Email: _____

Password: _____

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

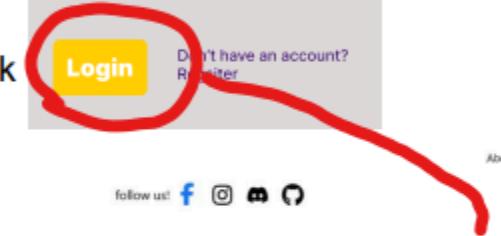
Login

Click **Login**

SwampStudy

follow us!

About Privacy Policy Licensing Contact



Matching

Currently in Forums

Department: General Chat: v

Class: College Help v

Search for threads... **click** [Search](#)

Graduation tickets

Does anyone have any extra graduation tickets???

[View Details](#)

Asking by: Sarah

Make a Post

SwampStudy

follow us!

About Privacy Policy Licensing Contact



The screenshot shows the SwampStudy website's header. It features a purple navigation bar with several buttons: "About", "Matching", "Currently in Forums" (which is highlighted in yellow), "Settings", and "Logout". A red arrow points from the "Logout" button towards the "Currently in Forums" button.

A modal window titled "Make a Post" is displayed. It contains fields for "Department" (set to "General Chat") and "Class" (set to "College Help"). Below these are fields for "Title" (containing "enter title") and "Question" (containing "type your question here..."). At the bottom are "Cancel" and "Submit" buttons, with a red arrow pointing to the "Submit" button.

The footer of the website includes the "SwampStudy" logo, followed by links to "About", "Privacy Policy", "Licensing", and "Contact". Below this, there is a "follow us!" section with icons for Facebook, Instagram, YouTube, and Twitter, each accompanied by a small red arrow pointing to it.

The screenshot shows the "Currently in Forums" page. On the left, there is a sidebar with "Department" and "Class" filters set to "General Chat" and "College Help". The main area displays search results for threads. The first thread is titled "Graduation Tickets" and was asked by "Sarah". The second thread is titled "Finding my textbook" and was asked by "John". Red arrows point to the "View Details" buttons for both threads.

The footer of the website includes the "SwampStudy" logo, followed by links to "About", "Privacy Policy", "Licensing", and "Contact". Below this, there is a "follow us!" section with icons for Facebook, Instagram, YouTube, and Twitter, each accompanied by a small red arrow pointing to it.

The screenshot shows the SwampStudy platform. At the top, there's a purple header bar with the logo 'SwampStudy' in yellow. Below the header, there are two tabs: 'Currently in Matching' (grey background) and 'Currently in Forums' (purple background). A red arrow points down from the 'Matching' tab towards a central message box. Another red arrow points up from the 'Forums' tab towards the same message box. The message box contains a conversation about finding a textbook. It includes a text input field with placeholder 'add your answer.' and two buttons: 'Cancel' (red) and 'Submit' (purple).

Finding my textbook

Is there a place to ask about old textbooks around here?

Asked by: Joe

John: yeah what class is this for?

Joe: really, its for calc 1

John: ok just go to the calc 1 forum they'll help you out!

add your answer.

Cancel Submit

The screenshot shows the SwampStudy platform. At the top, there's a purple header bar with the logo 'SwampStudy' in yellow. Below the header, there are two tabs: 'Matching' (grey background) and 'Currently in Forums' (purple background). A red arrow points down from the 'Matching' tab towards a sidebar on the left. Another red arrow points up from the 'Forums' tab towards the main content area. The main content area displays a search bar with 'Search for threads...' and a 'Search' button. Below the search bar, there are two threads listed: 'Graduation Tickets' (asked by Sarah) and 'Finding my textbook' (asked by John). Each thread has a 'View Details' button.

click

1. Department: General Chat v

2. Class: College Help v

Search for threads...

Search

Make a Post

Graduation Tickets

Asked By: Sarah

Anyone have any extra graduation tickets??

View Details

Finding my textbook

Asked By: John

View Details

About Privacy Policy Licensing Contact

follow us!

The screenshot shows the SwampStudy homepage with a purple header. On the left, there's a sidebar with dropdown menus for 'Department' (Math) and 'Class' (Calculus 1). The main content area has a search bar with the placeholder 'Search for threads...'. Below it, a thread titled 'Integrals' is listed, asking 'what is an Integral?' and noting it was 'Asked By: John'. A red arrow points from the top-left towards the sidebar, and another red arrow points from the top-right towards the search bar.

This screenshot shows the 'Make a Post' dialog box overlaid on the SwampStudy page. It contains fields for 'Department' (Math), 'Class' (Calc 1), 'Title' (enter title), and 'Question' (type your question here...). At the bottom are 'Cancel' and 'Submit' buttons. A red arrow points from the top-right towards the 'Submit' button.

The screenshot shows the SwampStudy interface. At the top, there's a purple navigation bar with yellow circular buttons for 'About', 'Settings', and 'Logout'. The title 'SwampStudy' is in the center, flanked by two green frog icons. Below the bar, there are two tabs: 'Matching' (gray) and 'Currently in Forums' (purple). A red arrow points down from the 'Matching' tab towards the main content area.

Matching

Currently in Forums

Search for threads...

Search **Make a Post**

Department: MATH **Class:** Calculus I

Integrals Asked By: John

what is an integral? **View Details**

Finding my textbook Asked By: Joe

Anyone take calc 1 with mr Jones, need help finding the text book. **View Details**

click

The screenshot shows a detailed view of a forum post titled 'Finding my textbook'. The post was made by 'Joe' and has a link: 'AVgAQVigUYQ6iCBBAGBSYAvrClBgGGSerUljAuMaAHmww&solient=ges-wiz-serp'. There is a text input field labeled 'add your answer.' and two buttons at the bottom: 'Cancel' and 'Submit'.

Finding my textbook

Anyone take calc 1 with mr Jones, need help finding the text book. Asked by: Joe

Steve: dude he used this old pdf its impossible to find, heres the link :
AVgAQVigUYQ6iCBBAGBSYAvrClBgGGSerUljAuMaAHmww&solient=ges-wiz-serp

add your answer.

Cancel **Submit**

About **Privacy Policy** **Licensing** **Contact**

follow us!

Database Architecture

1. Database Requirements

1.1. Registered User (Weak)

- 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 zero or one account

1.2. Account (Strong)

- 1.2.1. An Account shall belong to one or zero 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 or zero profiles
- 1.2.5. An Account shall have many Ratings
- 1.2.6. An Account shall take many Classes

1.3. Match (Strong)

- 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 one meeting link
- 1.3.5. A Match shall have one meeting time

1.4. Question (Weak)

- 1.4.1. A Questions shall be posed by one Account
- 1.4.2. A Question shall have one or 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 one 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 one or 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
- 1.8.4. A Class shall be taken by many Users

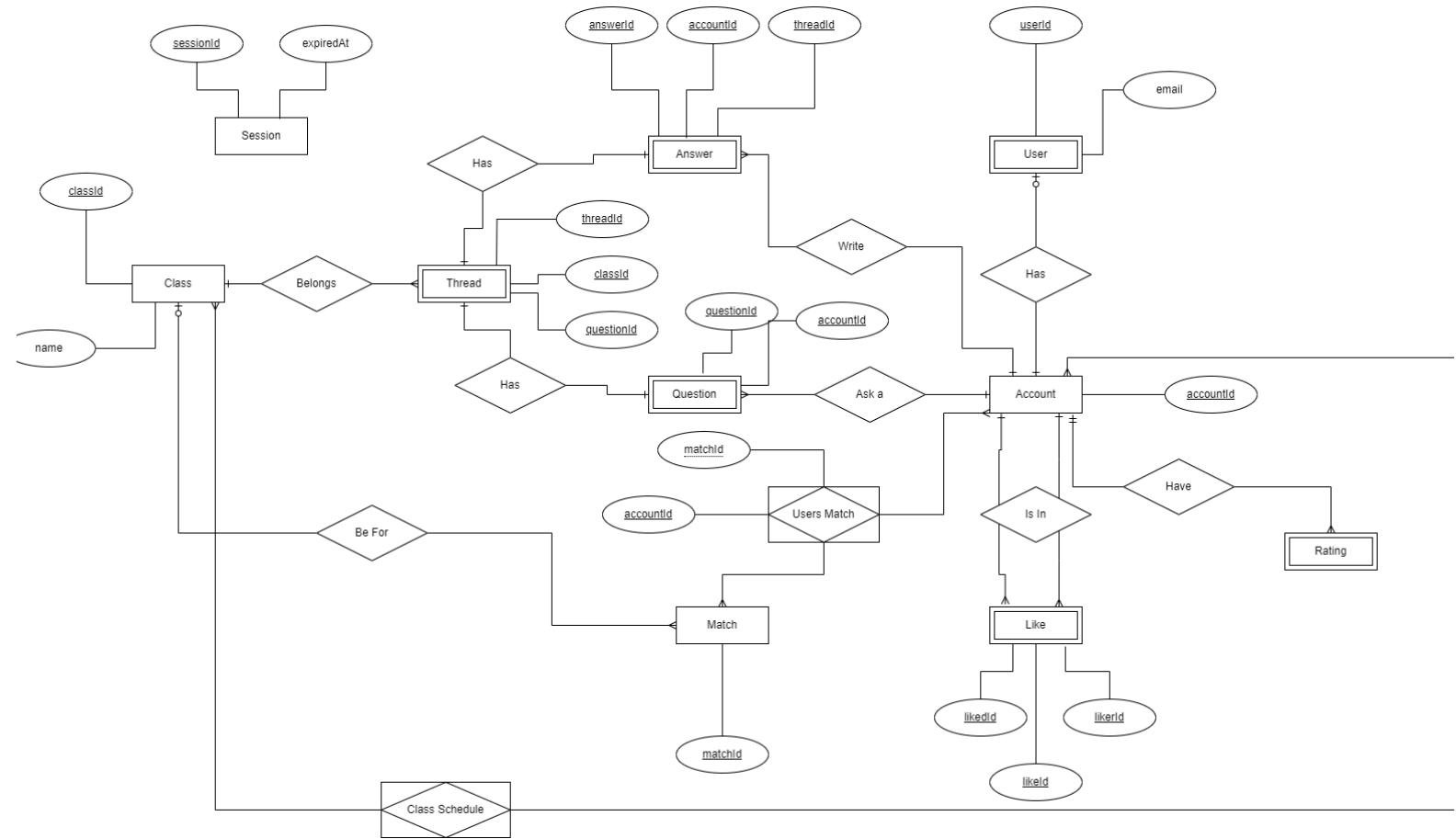
1.9. Profile (Weak)

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

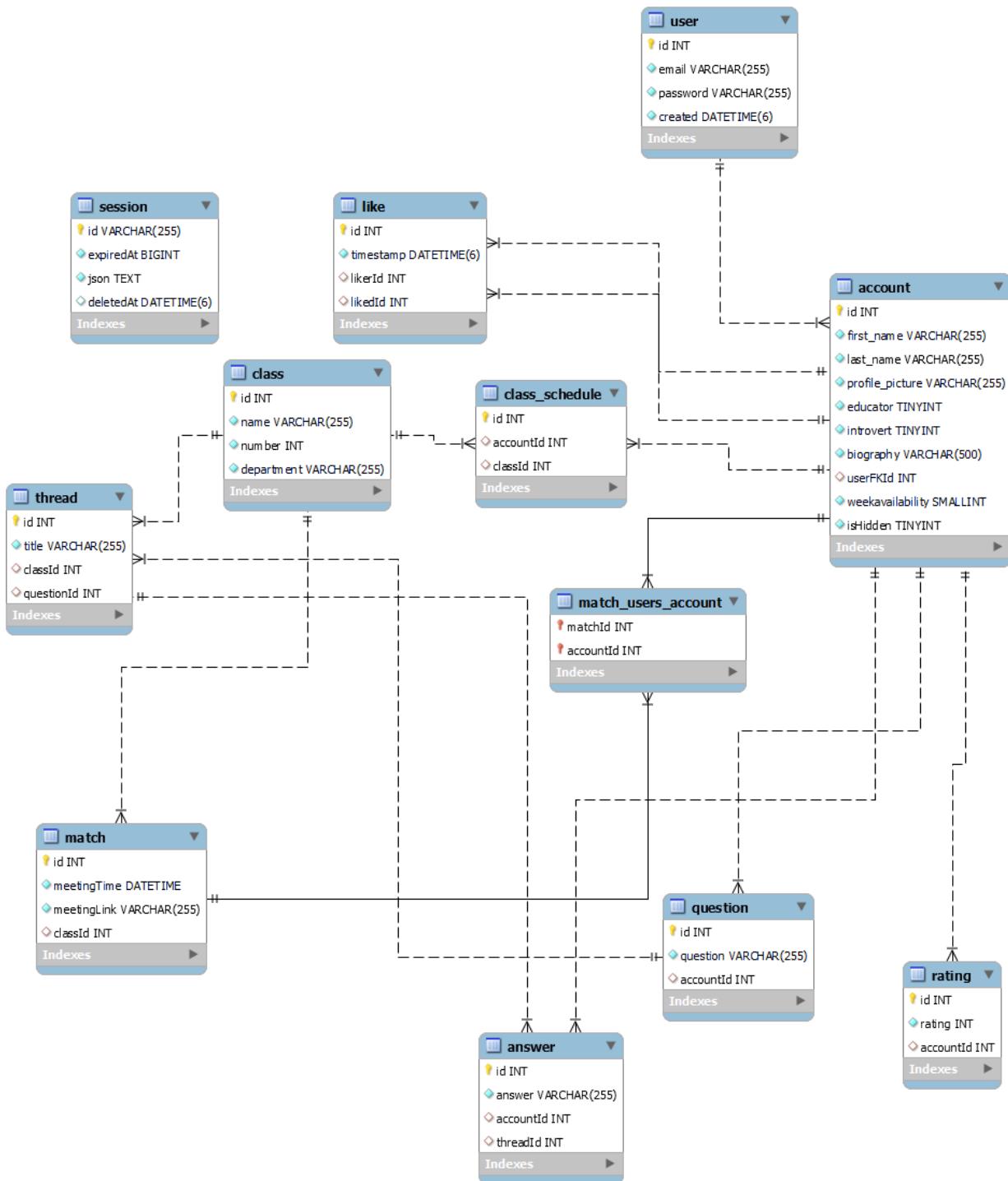
- 1.10. Like (Weak)
 - 1.10.1. A Like shall have one or more accounts
2. Entity Attributes, Relationships, and Domains
 - 2.1. user (Weak)
 - 2.1.1. id: PK, Numeric, Generated
 - 2.1.2. email: unique, alphanumeric
 - 2.1.3. password: alphanumeric, hashed
 - 2.1.4. created: datetime, generated
 - 2.1.5. account: Numeric, ForeignKey
 - 2.2. account (Strong)
 - 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. educator: tinyint (Boolean)
 - 2.2.6. introvert: tinyint (Boolean)
 - 2.2.7. biography: varchar(500)
 - 2.2.8. userFKId: Numberi, Foreign Key
 - 2.2.9. weekavailability: smallInt
 - 2.2.10. isHidden: tinyInt
 - 2.3. match (Strong)
 - 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. rating: Numeric, int
 - 2.11.3. accountId, int
- 2.12. classSchedule (weak)
 - 2.12.1. Id: PK, Numeric, Generated
 - 2.12.2. account: FK, Numeric Account ID
 - 2.12.3. class: FK, Numeric, Class ID

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

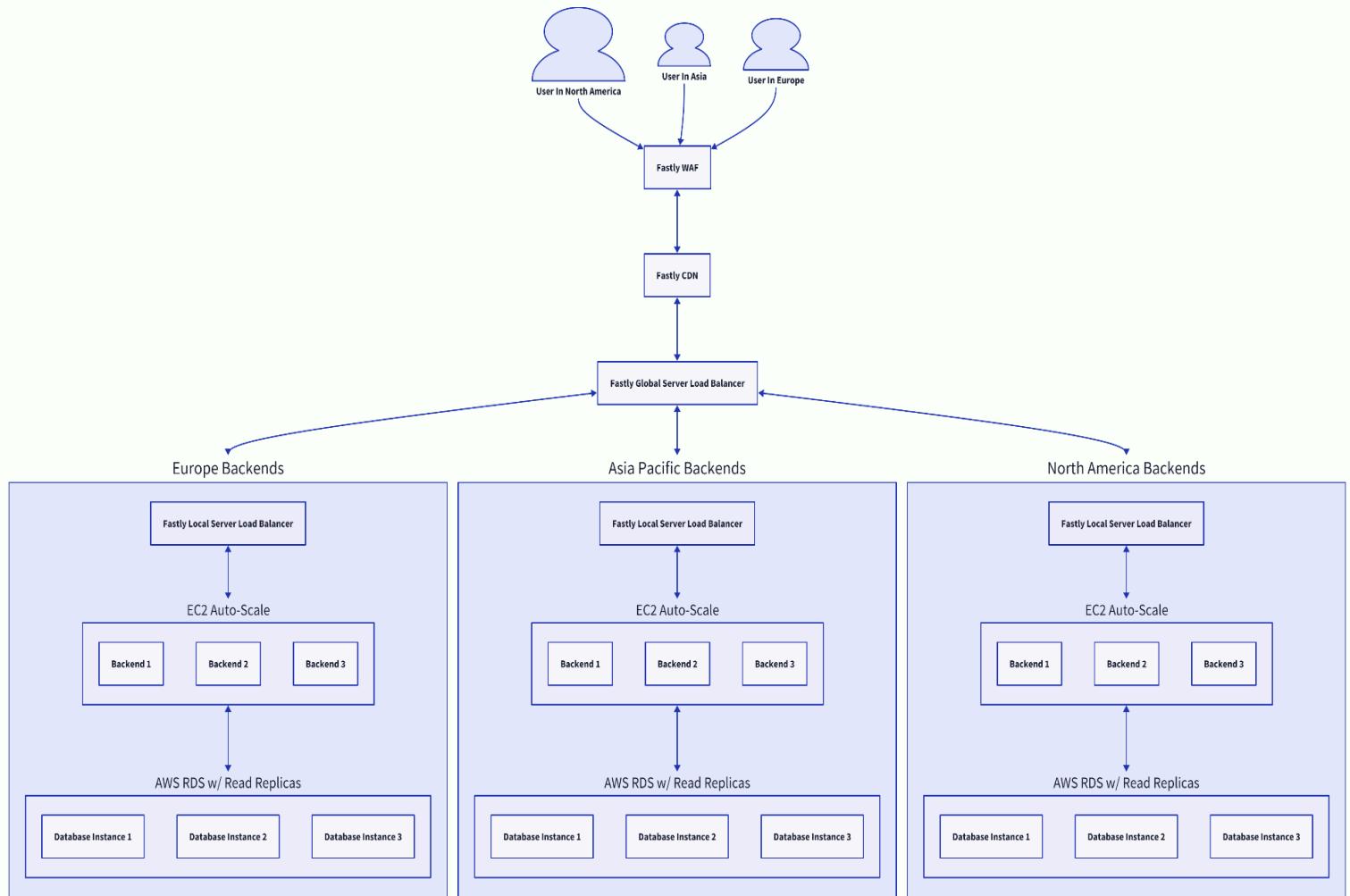
search box and then use a query with the ‘where’ keyword and regex wildcards to determine which posts fit the given criteria and then filter and display the results based on that keyword. We also scan the answers of each question as well in order to give a more comprehensive search. What gets returned 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:

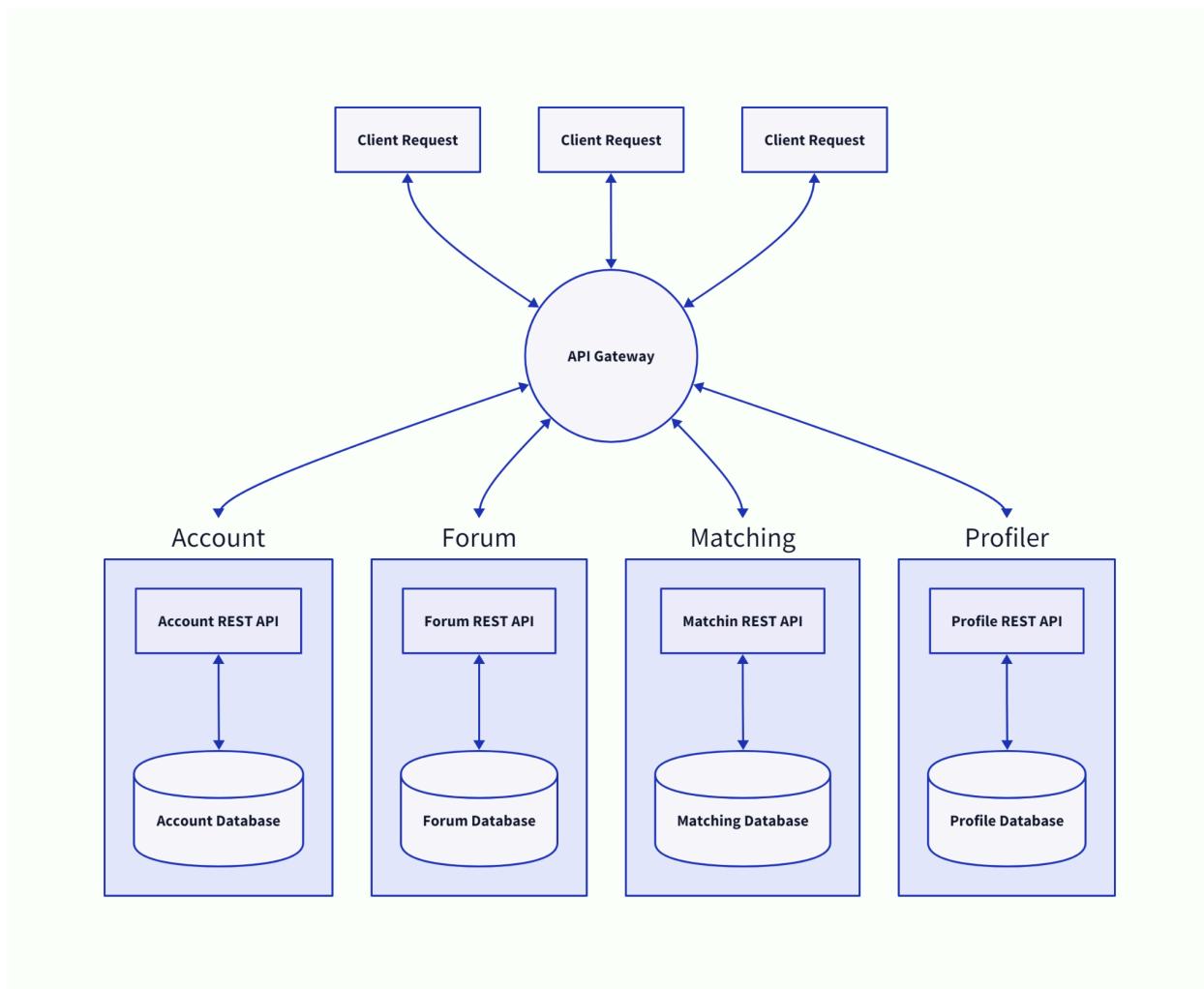
See the end of the Document for full size diagram

Scalability Diagram



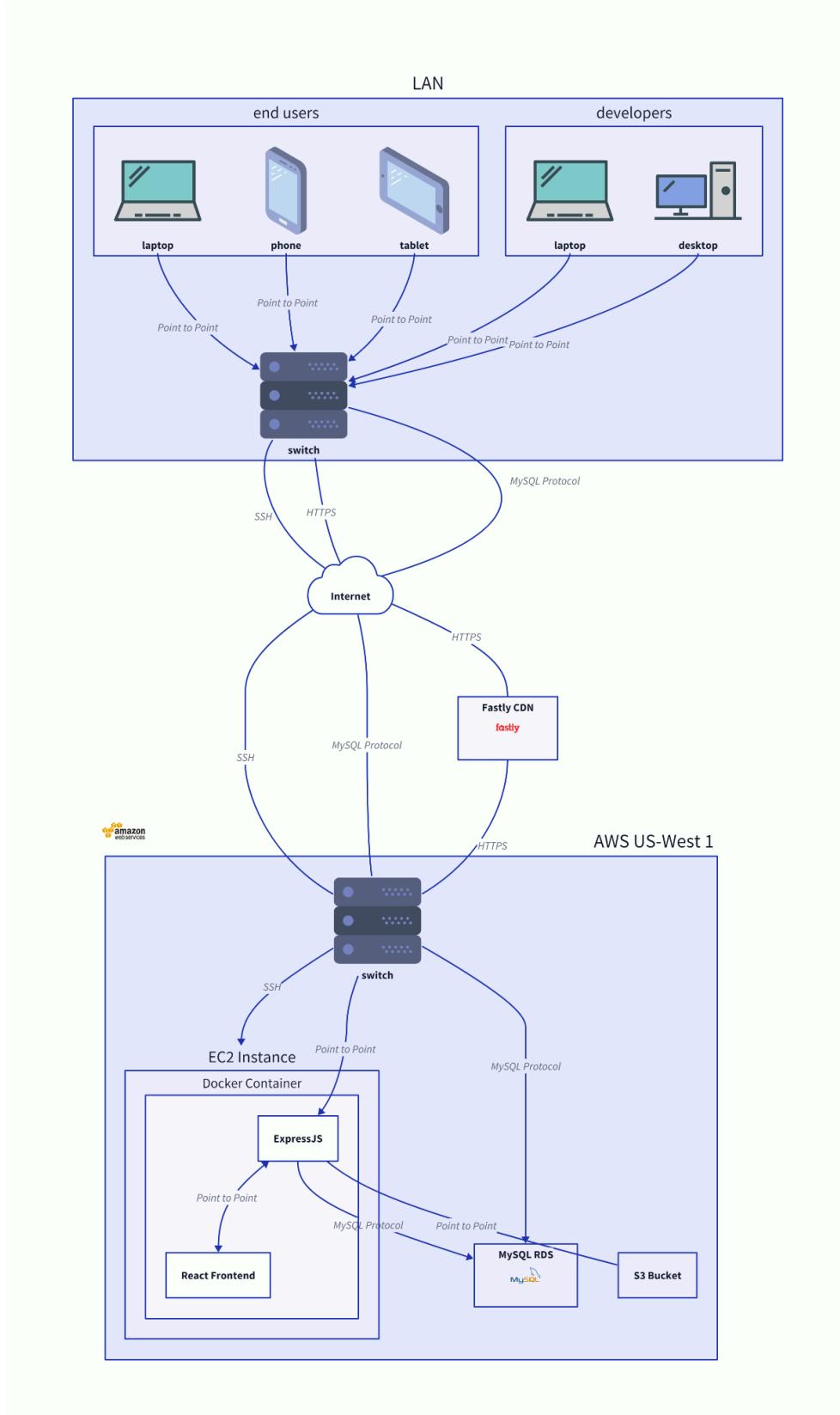
To let our service scale with ease, we first introduce a WAF at the top of the chain. This will allow us to filter malicious requests and bots trying to gain access to the system. We then run all the requests through our CDN which allows for request collapsing and to deliver cached content far quicker 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

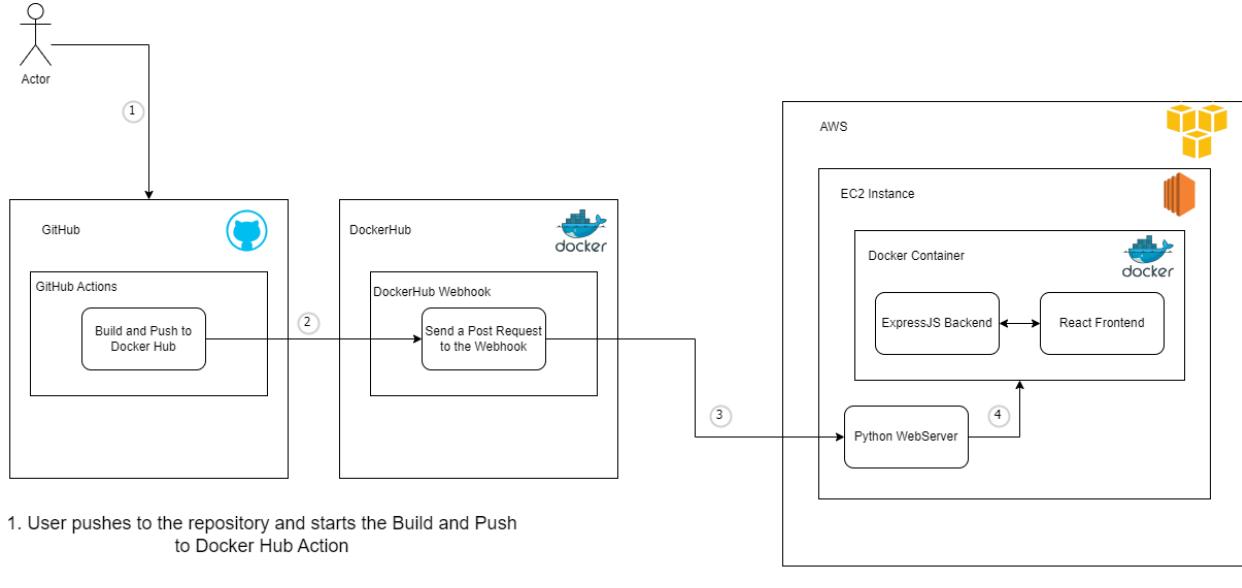


High Level Application Network and Deployment Design

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

Team Evaluation

Lennart (Team Lead) 10

- Project Planning
- Developed Backend for Image Upload
- Implemented S3 Bucket on AWS
- Outline for File Upload on the Frontend
- Revision for M2V2
- Revised the DB Diagram
- Organized the Functional Requirements
- Attended All Team Meetings
- Responsive On Discord

Rafael (Backend Lead) 10

- Settings Backend Implementation
- Setting Page Frontend Connection
- Matching Prototype
- Input Data into Database
- Style Inputs
- Organized Backend Architecture
- Lots of bug fixes
- Tech Support for the Team
- Attended All Team Meetings
- Responsive on Discord

Edmund (Frontend Lead) 10

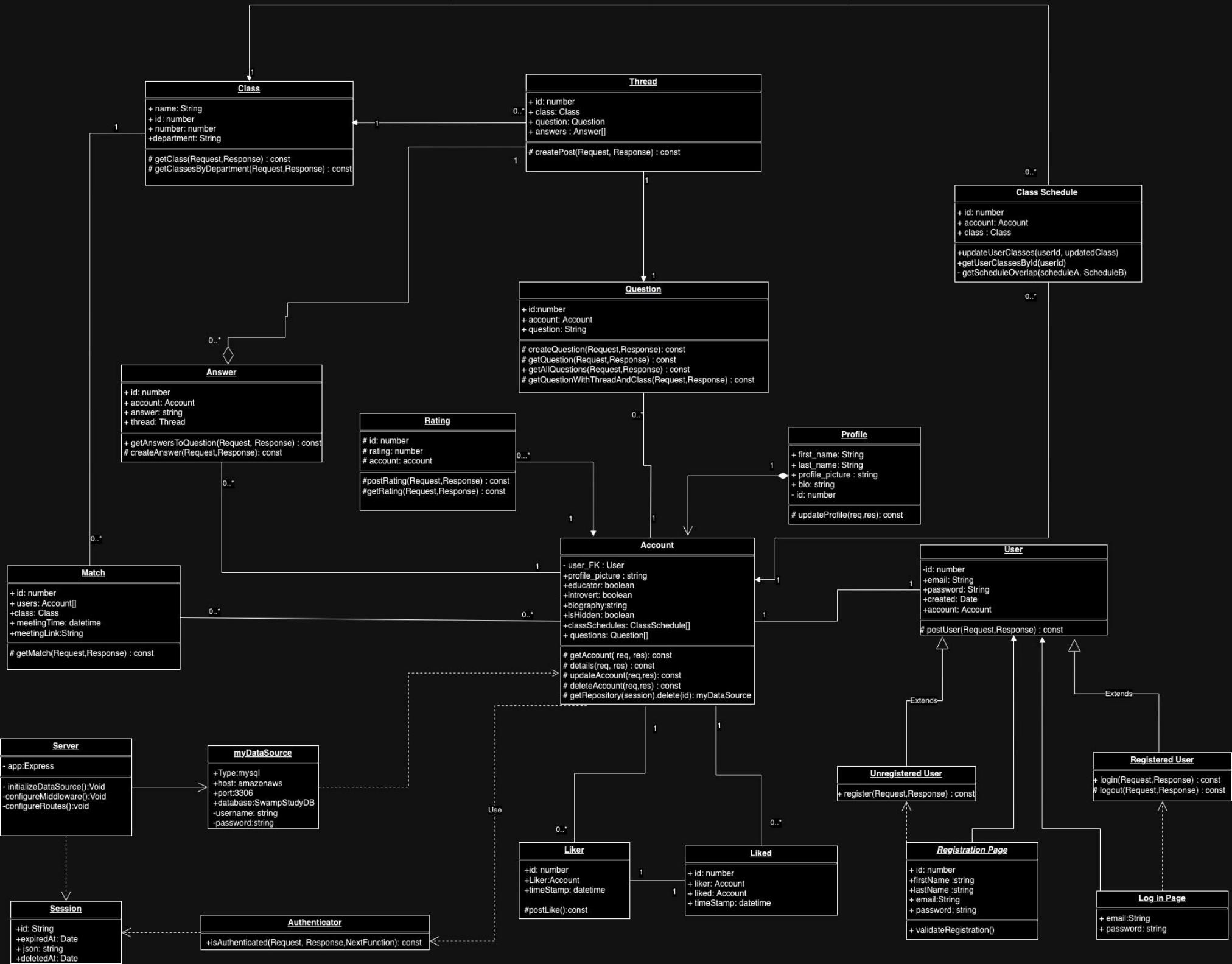
- Multiple Figma Use Cases
- Revised Mockups for M2V2
- Worked on the M3 Documentation
- Frontend Matching Implementation
- Backend Matching Support
- Attended All Team Meetings
- Responsive on Discord

Julio (Docs-Editor) 10

- Made Multiple Figma Use Cases
- Revised Mockups for M2V2
- Submitted all the Documents
- Organized and Wrote M3
- Frontend for Settings Page
- Backend support for Settings Page
- Attended All Team Meetings
- Responsive on Discord

Conrad (Github-Admin) 10

- Managed GitHub Conflicts
- UML Diagram for Document
- Implemented the Footer
- Added the Footer to all the required pages
- Implemented all the Links from the footer
- Created mock accounts for the social links
- Multiple Bug Fixes
- Attended All Team Meetings
- Responsive on Discord



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
Milestone 2 V2	Apr 24, 2024
Milestone 3	Apr 24, 2024
Milestone 3 V2	May 15, 2024
Milestone 4	May 15, 2024

Table of Contents

SW Engineering CSC648-848-05 Spring 2024.....	1
Table of Contents.....	2
Product Summary.....	4
Committed Application Functions.....	4
Summary/Unique Brag.....	5
Usability Test Plan.....	6
Objective: Our objective for our usability test aims to evaluate and refine the StudySwamp app's matching feature to deliver an optimal user experience that is both intuitive and satisfying. Our usability test checks whether the integral functions of the matching feature are intuitive, efficient, and user-friendly.....	6
1. Function 1 - Filling User Profile	
Test Objective.....	7
2. Function 2 - Matching.....	8
3. Function 3 - View Matches.....	9
4. Function 4 - Rating.....	10
5. Function 5 - Report Users.....	11
Test Objective.....	11
QA Test Plan.....	15
1. Performance: The site shall load in less than 7 seconds on a wireless connection with a speed of 350 Mbps.....	15
2. Scalability: The system shall be containerized allowing for simple scaling.....	16
3. Data Integrity: The Application shall allow only registered users to access their data.	17
4. Privacy: Only Registered Users shall be able to access the application.....	18
5. Utility: The site shall support concurrent usage by many users without negligible performance reduction.....	19
Security Self Check.....	20
Code Review.....	21
Coding Style.....	21
Code Review.....	21
External Code Review.....	24
Adherence to Non-Functional Requirements.....	25
1. Usability.....	25
2. Reliability.....	25
3. Scalability.....	25
4. Maintainability.....	25
5. Utility.....	26
6. Performance.....	26
7. Privacy.....	26
8. Data Integrity.....	26

9. Security.....	26
10. Storage.....	26
11. Deployment.....	26
12. Coding Standards.....	26
13. Environmental Sustainability.....	27
Team Evaluation.....	27
Lennart (Team Lead) 10.....	27
Rafael (Backend Lead) 10.....	27
Edmund (Frontend Lead) 10.....	28
Julio (Docs-Editor) 10.....	28
Conrad (Github-Admin) 10.....	28

Product Summary

Name of the product: Swamp Study

URL: <https://swamp-study.global.ssl.fastly.net/>

Committed Application Functions

1. User

- 1.1. A User shall be able to create an account
- 1.2. A User shall only have one account
- 1.3. A User shall be able to view other users' accounts in the system
- 1.4. A User shall be able to set their availability
- 1.5. A User shall be able to establish their profile
- 1.6. A User shall be able to ask many questions in the Forum
- 1.7. A User shall be able to answer many questions in the Forum
- 1.8. A User shall be able to like other Users
- 1.9. A User shall be able to match with other Users

2. Profile

- 2.1. A Profile shall be owned and edited by one and only one user
- 2.2. A Profile shall be allowed one picture of the owner
- 2.3. A Profile shall be allowed a short biography
- 2.4. A Profile shall have many subject attributes
- 2.5. A Profile shall be able to set a filter for their study partner preferences
- 2.6. A Profile shall be marked Student or Educator
- 2.7. A Profile marked Educator shall be able to access the Forum
- 2.8. A Profile marked Educator shall be allowed to answer questions in the Forum

3. Account

- 3.1. An Account shall have one email associated with it

4. User Interface

- 4.1. The Interface shall allow a User to navigate through a potential Match's profile
- 4.2. The Interface shall allow a User to access/edit their Profile
- 4.3. The Interface shall allow a User to access/edit their Account

5. Match

- 5.1. A Match shall be provided a meeting link
- 5.2. A Match shall be provided a meeting time
- 5.3. A Match shall be made based on different criteria

6. Forum

- 6.1. The Forum shall allow one main question per Thread

- 6.2. The Forum shall allow each class its section
- 6.3. The Forum shall allow users to post in their class Forum
- 6.4. The Forum shall allow many questions

Summary/Unique Brag

The crowded field of educational applications is one of little breadth. With the COVID pandemic and teachers' reliance on online tools, the educational process is more intertwined with technology than ever before. The "Legacy" applications in the study space focus on providing a platform in which to ask and answer homework questions. This has led to a rise of unethical behavior to the educational process. These platforms want to sell a subscription that lets you get homework answers quickly for a price. This takes out the most important aspect of education, the process of learning and collaboration.

Swamp Study aims to be the leader in bringing collaborative learning to the online space. A common quandary faced by college students and those of all ages is the process of finding a study partner, someone with whom to share and enhance the learning experience. This is where Swamp Study comes in, we provide a state of the art matching system to give students the access to study partners at their learning institution. We provide users with their most likely matches first ensuring that you get quality partners which will lead to better grades. By letting the students put in their class schedule and availability we can make sure that you'll find people with commonalities that will surely make the study experience an enjoyable one.

Usability Test Plan

Objective:

Our objective for our usability test aims to evaluate and refine the StudySwamp app's matching feature to deliver an optimal user experience that is both intuitive and satisfying. Our usability test checks whether the integral functions of the matching feature are intuitive, efficient, and user-friendly.

5 Functions To Be Tested:

- Filling user profile
- Matching
- View Matches
- Rating
- Report

Test Descriptions Per Function:

1. Function 1 - Filling User Profile

Test Objective

- We are evaluating the usability and intuitiveness of the profile setup process in the SwampStudyweb service, focusing on user satisfaction, and the time taken to complete the profile. This function is important because users that have setup their profile will get better matches.

Test Description

- Users will fill out their profile in the StudySwamp web service. This includes adding the courses they are currently taking, submitting their available weekdays, and adding a personal bio (not required).

Usability Test Description

- Participants will log in to a pre-established account.
- Participants will then navigate to the profile page.
- Participants will start to edit their profiles.
- Participants will enter a bio.
- Participants will add courses and weekly availability.
- The test will measure the time it takes to complete the profile.

Task	Description
Task	Complete personal profile page
Machine State	Logged In
Successful Completion State	The tester has entered his availability and courses.
Benchmark	5 minutes

2. Function 2 - Matching

Test Objective

- The Goal of this usability test is to determine whether the interface and process for liking and disliking users is an intuitive and user friendly one
- We want to assess the efficiency and accuracy of the process of making matches within the system for a user

Test Description

- The Matching process consists of reading profiles and deciding whether to like their profile as a potential match or to dislike
- Users will be put on the matching page and asked to start interacting with the matching system

Usability Task Description

- For this task a user will go through the process of liking and disliking potential study partners. The ideal user for this is someone of college age who is familiar with the tinder style of interface for making matches
- The user will start their test on the matching page, they will be given a testing account which already has the profile filled out. They should be able to interact with the matching interface right away

Task	Description
Task	Likes/Dislike Users Profiles
Machine State	Logged in on the Matching Page
Successful Completion State	The Tester has liked and disliked at least 10 different profiles
Benchmark	5 Minutes

3. Function 3 - View Matches

Test Objective

- We are testing the navigation and the user interaction with the View Matches function within the application.
- The objective is to evaluate the intuitiveness and the efficiency of navigating from the first page a user sees to the matching page with prior matches. The test is to ensure that students can easily form study groups by navigating and interacting with their matched profiles.

Test Description

- System setup will be conducted on an internet browser.
- Participants will start on a pre-established account on the Forums page.
- The intended audience for this test is students seeking to form study groups.

Usability Test Description

- The starting point will be on the Forums page the users will manually navigate to the Matching page. This is to test the application's navigational design. Also found on (<https://swamp-study.global.ssl.fastly.net/>)
- Reaching the Matching tab the participants will interact with any element in the matches revealing a detailed profile description of the selected profile.
- The test will measure the time it takes to navigate between tabs and ease of accessing the matched profiles.
- Feedback will be collected based on the clarity, usefulness, and satisfaction of using the View Matches function.

Task	Description
Task	Interact with any profile you matched with.
Machine State	The homepage logged in a pre-established profile.
Successful Completion Criteria	Participants place the mouse cursor on the Your Matches tab.
Benchmark	8.45 Seconds

4. Function 4 - Rating

Test Objective

- We test the rating component with the intent to gauge the efficiency and intuitiveness of the functionality itself as the user interacts with it.
- Moreover the tests aims to assess how accurately and effectively a user can submit a rating for a user.
- The test will also be used to asses and identify potential pitfalls, bugs or problems with the current implementation.

Test Description

- System setup will be conducted on an internet browser.
- Each participant will perform the task of rating other users by selecting a number of stars so as to reflect their perception of said users profile.
- The intended audience for this test is users who are familiar with interactive rating systems similar to those seen in e-commerce or other social media platforms.
- Participants will use a pre configured testing account with the intent that they can start rating immediately upon accessing the match details page.

Usability Test Description

- Participants will begin the test from the Meeting Details page within the application. Here, they will click the "Rate User" button, which opens a view displaying the user's current rating and the interface for submitting a new rating.
- Feedback will be collected based on the timeliness, clarity, and effectiveness of using the View Matches function.

Task	Description
Task	Submit a 0 to 5 star rating for a user
Machine State	The users current rating
Successful Completion Criteria	Participants choose a rating and submit it
Benchmark	7 Seconds

5. Function 5 - Report Users

Test Objective

- Main objective is to assess the ease of use, and effectiveness of users being able to Report other users.

Test Description

- Participants will be asked to successfully match with a user and then assess if they can Report that user to us.

Usability Test Description

- The ideal user for this usability test is an individual that will do the community a favor by reporting inappropriate behavior and content. Overall enhancing the user experience of our website.

Task Description

Task	Description
Task	Successfully use the Report function
Machine State	Users own “previous matches” page
Successful Completion Criteria	Pop Up stating Report was successful
Benchmark	1 minutes

Usability Test Table

Tests/Use Cases	Total Completed (in Percentage)	Errors	Comments	Efficiency
Filling User Profile	100%	Added the wrong class but was able to fix it	Hard to find specific classes without reading all the text.	4 minutes
Matching	100%	N/A	If you aren't familiar with the "Swiping" it's difficult to understand what exactly is happening. I didn't find any matches as well so it's hard to test	45 Seconds
View Matches	100%	Didn't notice the Matching tab to move to the matching page.	"It can be easy to miss."	8.93 Seconds
Rating	100%	N/A	"Its quick, i like that the button is big / yellow its not hard to find"	~7 seconds
Report Users	100%	none	<p>"I think it is a little confusing finding your matches. I had to click around to figure out I had to do previous matches. It would be nice to have a separate section that can show all matches. When I think of previous matches I think that those are the most recent and not all"</p> <p>"Also you have to have a match in order to report them, is that intentional? Like what if I see someone's profile and I think they are sus, do i need to match with them to report."</p>	2-3 min

Questionnaire

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Is filling out the profile straight forward?				X	
Is the clear separation of fields beneficial?					X
Did you learn what you could edit quickly?					X
Potential Match Profile info is displayed in an easily digestible manner				X	
The Like Mechanism is intuitive				X	
The Dislike Mechanism is intuitive			X		
Easy to Navigate to Matching Tab					X

Information displayed in Your Matches was clear and easy to locate				X	
I understood the functionality of Your Matches at first glance.					X
Rating a user is simple and straight forward					X
Selecting the rating i want was easy					X
I understood that the stars in the rating page represent the rating i want to submit					X
Easily found a Previous Match		X			
Easily found the Report button		X			
UI was not confusing, user did not get lost trying to Report a user			X		

QA Test Plan

1. Performance: The site shall load in less than 7 seconds on a wireless connection with a speed of 350 Mbps

- Test Objective:
 - The Test will ensure that the website loads in a reasonable amount of time
- HW and SW Setup:
 - The Website needs to be successfully deployed to the EC2 instance for this test
- Test Environment:
 - The Test will utilize 3 popular and reputable tools for website testing
 - <https://pagespeed.web.dev/>
 - Google Chrome built in Lighthouse analyzer
 - <https://www.webpagetest.org/>

Number	Description	Test Input	Expected Output	Actual Output	Pass/Fail
1	Running the test on pagespeed.web.dev	The home page link, ran on a throttled connection of 10.2 Mbps	Speed Score of <= 7s	Speed Score of 2.2s	Pass
2	Running the test on lighthouse-metrics	The home page link, ran on a throttled connection of 10.2 Mbps	Speed score of <= 7s	Speed Score of 0.7s	Pass
3	Running the test on webpagetest.org	The home page link, ran on a throttled connection of 5Mbps from an East Coast location	Speed score of <= 7s	Speed Score of 1.159s	Pass

2. Scalability: The system shall be containerized allowing for simple scaling

- Test Objective:
 - To test the containerization of our application
- HW and SW Setup:
 - Testing can be done when the container is running on the EC2 instance.
Docker Hub needs to be available, as well as GitHub Workflows
- Test Environment:
 - 1 Person with access to all the required resources, 1 hour

Number	Description	Test Input	Expected Output	Actual Output	Pass/Fail
1	Running the GitHub workflow for automated pushing to DockerHub	Manually running the GitHub Workflow for Container creation and push to DockerHub	A new version of the container in DockerHub	A new version of the container in DockerHub	Pass
2	Running the DockerHub Webhook for automatic deployment on the EC2 instance	Sending a new version of the container to DockerHub using the above workflow, and having the webhook send a signal to the EC2 instance	Checking the uptime and version of the Docker Container running on the EC2 to make sure the process worked successfully	A new version of the container was pulled and started running on the instance	Pass
3	Rolling back to another version of the Docker container	Running the workflow for creating a new version of the container with issues and then rolling back to a previous	We expect the broken/buggy version to stop running and the “new” reverted version to take over	The Docker container successfully stopped and was rolled back to a previous version	Pass

		version of the container on the instance			
--	--	--	--	--	--

3. Data Integrity: The Application shall allow only registered users to access their data

- Test Objective:
 - To determine whether data integrity is being upheld by allowing users to only access data pertinent to them
- HW and SW Setup:
 - The EC2 instance needs to be running, the S3 bucket active, and the Fastly CDN active
- Test Environment:
 - 1 person required, and 30 minutes

Number	Description	Test Input	Expected Output	Actual Output	Pass/Fail
1	Test whether the settings page can be accessed without being logged in to change a password	Trying to access the https://swamp-study.global.ssl.fastly.net/settings url without being logged in. And curling the endpoint without authorization	A Redirect in the case of the URL and a forbidden on the Curl to the endpoint	When trying to access the URL we get a redirect. When curling the endpoint we get a json response saying "User not authenticated"	Pass
2	Test whether we can delete a User by hitting the API Endpoint	Curling the Endpoint responsible for deleting user accounts	A message stating it's forbidden to access that feature without being authenticated	Json response message saying "User is not authenticated"	Pass
3	Checking whether authorized	Logging in and changing the password	We expect the change to go through	Message saying the account was	We receive a popup stating that the

	changes work	on the settings page for the logged in user	and change the user password	updated	account was successfully updated
--	--------------	---	------------------------------	---------	----------------------------------

4. Privacy: Only Registered Users shall be able to access the application

- Test Objective:
 - To determine whether Privacy is being upheld by only allowing registered users to access any parts of the site besides the about and login
- HW and SW Setup:
 - The Application has to be running on the EC2 instance
- Test Environment:
 - 1 Person, 30 Minutes

Number	Description	Test Input	Expected Output	Actual Output	Pass/Fail
1	Accesing the Forum Page	Trying to access https://swamp-study.global.ssl.fastly.net/forum	To not be on the Forum page	An Empty page	Pass
2	Accessing the Matching Page	Trying to access https://swamp-study.global.ssl.fastly.net/matching	To not be on the Matching page	An Empty Page	Pass
3	Accessing the Settings page	Trying to access https://swamp-study.global.ssl.fastly.net/login	To not be on the settings page	The Login Page	Pass

5. Utility: The site shall support concurrent usage by many users without negligible performance reduction

- Test Objective:
 - Stress testing the website to make sure concurrent usage does not negatively impact other users of the system. The tests will be compared to a benchmark of 100 requests
- HW and SW setup
 - The application needs to be on the EC2 instance and the Fastly CDN needs to be configured
- Test Environment:
 - 1 Person, 30 minutes. Using Apache Benchmark on WSL to perform the requests
 - Benchmark: Sending 1000 requests, 50 concurrently
 - Requests per second: 219.24
 - 99% of Requests served within 0.508s

Number	Description	Test Input	Expected Output	Actual Output	Pass/Fail
1	Light Load	Sending 1000 requests 100 concurrently		227.83 Requests Per Second, and 99% served within 0.553s	Pass
2	Medium Load	Sending 1000 Requests, 250 concurrently		195.72 Requests Per Second, and 99% served within 2.064s	Pass
3	Heavy Load	Sending 10000 requests, 1000 concurrently		374 Requests Per Second, and 99% served within 3.701s	Pass

Security Self Check

1. Using the express-session middleware we have implemented token based sessions. Any requests that try accessing the API or Data that is in a password protected area will need a valid session to access.
2. Passwords are all encrypted and once entered by the user will not be able to get accesses in clear text again. We are using the bcryptjs library within our program to encrypt passwords as they are entered into our application. When logging into the application the encrypted password which is stored in the DB gets retrieved and compared to the password the user answered and if the two match a user can login. This allows us to effectively store user passwords.
3. All data and text fields which are entered into the application have front-end and back-end validation. This double sided approach allows us to catch any type of invalid or dangerous data that users may try to input into the application. In our Service Layer we also validate all the data and only pass requests to the Database once we can be sure the request will have no adverse effect on the Application, and other users data.

Code Review

Coding Style

- All Frontend Folders in lowercase
- All React Components Completely Capitalized
- We use spaces instead of tabs
- All Controllers are camelCase
- All entities are lowercase
- All Repositories are Completely Capitalized
- All Routers are in camelCase
- All Services are Completely Capitalized
- Use Consistent Coding Style Throughout the Application
- Explain any Difficult Sections
- Provide Consistent Comments

Code Review

We used GitHub to make pull requests and review each others code. Below are some excerpts from those reviews.

implement requested changes and include a screen shot.



application/frontend/src/App.jsx

rf922 last week

added faqq route correctly



Reply...

Resolve conversation

application/frontend/src/components/faq/FAQ.jsx

rf922 last week

there are multiple components within `FAQ.jsx` namely `FaqItem` and `FAQ` itself, the convention we use is one component per jsx, each jsx file should live then in a folder whose name is the name of the jsx file in all lower case., seperate the components to their own jsx files and import or retool to only use one component. this is done with the intent that each component be modular and easier to maintain., also consider moving faqData to a seperate .js file `faqData.js` and importing it



Reply...

 rf922 reviewed last week [View reviewed changes](#)

rf922 left a comment

...
nice restyle for the registration component using consistent styling and addresses what is bein fixed, in the future make the pull request to Development branch


application/frontend/src/components/register/Register.jsx

 rf922 last week
good job addressing the concern of new users being able to upload files


 Reply...

[Resolve conversation](#)

 rf922 closed this last week

len898 left a comment

...
Looks Great. The rewrite was done with our code style and looks to be bug free.


 rf922 commented 2 weeks ago
good consistent code style with well chosen color scheme to align with site colors while adding contrast for visibility,


 rf922 approved these changes 2 weeks ago [View reviewed changes](#)

rf922 left a comment

...
code looks clean and fixes the issue as intended,


application/frontend/src/components/footer/footer.jsx

 rf922 2 weeks ago
solid use of tailwind directly addresses the concern of the footer being too "thick", consistent styling changes to the margins `mr-4 md:mr-6` to `mr-2 md:mr-3`. good use of break points


 Reply...

External Code Review

We performed a code review with Team 5. The code review was for one of our Controller files from the backend. This is the feedback we received. We implemented all the changes that made sense within the scope of our application

I checked out the LikeController code you sent over. Here are a few pointers:

Constructor: You don't need to assign the services inside the constructor since TypeScript does this automatically when you list them as private in the constructor itself.

Error Handling: The way you check if userId2 is a number is good. For errors, the message you send back could be clearer or more specific based on what went wrong.

Status Codes: Make sure you use CREATED only when something new is actually made. For other errors, 500 Internal Server Error might be more suitable than 422 Unprocessable Entity.

Business Logic: The part where you check if both users liked each other could perhaps go into the LikeService or MatchService. This would keep your controller simpler and just focused on handling the request and response.

Clean Up: If there's any code you're not using anymore, like // const userId1 = 1; it's a good idea to remove it to keep things neat.

I'm also attaching a file with some code I need help reviewing. Could you take a look when you have a chance?

Let me know if you have any questions or need more help with this!

Cheers,

The Feedback we sent to Team 5 is pictured below

We've reviewed your code and had a few pieces of feedback.

- there's a typo "const preferencess = await userService.getUserPreferences(req.params.userId);;" there's an extra S in "preferences"
- i don't see validation but I'm assuming there is one somewhere else in their code.
- not a big thing but changing the order of parameters in the "try{ const user" to match the order of the "const createUser" or vice versa
- Everything else looks really good

Adherence to Non-Functional Requirements

1. Usability

- 1.1. The U.I. will be designed to be as intuitive as possible allowing use from all technical backgrounds. - Done
- 1.2. The Application shall be compatible with web browsers, and mobile devices - Done
- 1.3. The Application shall conduct surveys on features that could improve usability - Issues
- 1.4. The Application U.I. shall be streamlined and responsive - Done
- 1.5. The system shall send notifications to the user's device - Moved to P2
- 1.6. A user shall be allowed to match with many partners - Done
- 1.7. A high rating shall make the user more visible - Done
- 1.8. The system shall show a user their ideal match - Issues
- 1.9. The system shall notify users when they have a match - Moved to P2
- 1.10. A high rating shall gift users one free super like - Moved to P2

2. Reliability

- 2.1. The application shall achieve 90% uptime - Done
- 2.2. The Application shall minimize the occurrence of bugs and crashes - Done

3. Scalability

- 3.1. The System shall be containerized allowing for simple scaling - Done
- 3.2. The System shall automatically provision resources to handle fluctuating demand - Done

4. Maintainability

- 4.1. The Application shall be formed in a manner that supports hot fixes to minimize downtime - Done
- 4.2. The Restoration period shall be less than 24 hours - Done
- 4.3. The Application shall be able to update and upgrade without downtime - Done

- 5. Utility**
 - 5.1. The site shall support concurrent usage by many users with negligible performance reduction - Done
 - 5.2. The System shall scale to accommodate a growing user base - Done
 - 5.3. The System shall support Firefox version 123.0 - Done
 - 5.4. The System shall support Google Chrome Version 122.0.6261.70 - Done
- 6. Performance**
 - 6.1. The site shall load in less than 7 seconds on a wireless connection with a speed of 350Mbps. - Done
- 7. Privacy**
 - 7.1. Users shall have access to privacy settings - Done
 - 7.2. Users shall be able to hide their profile - Done
 - 7.3. Users shall be able to view the privacy agreement - Done
 - 7.4. The Platform shall treat customer data as sensitive - Done
 - 7.5. Only Registered Users shall be able to access the application - Done
- 8. Data Integrity**
 - 8.1. The Application shall ensure that all billable data is accurate - Moved to P2
 - 8.2. The Application shall only allow registered users to access data and the sites - Done
- 9. Security**
 - 9.1. The Application shall encrypt sensitive user data - Done
 - 9.2. The Application shall comply with regional data safety regulations - Done
 - 9.3. Payment Information shall be routed through a trusted third party service provider using state of the art security measures - Moved to P2
- 10. Storage**
 - 10.1. The Application shall assign 10MB of Memory per table - Done
 - 10.2. The Application shall make automatic backups to prevent data loss - Done
- 11. Deployment**
 - 11.1. The Application shall use github integrations to build the Docker container - Done
 - 11.2. The Cloud instance shall automatically transition to the newest version of the application when available - Done
- 12. Coding Standards**
 - 12.1. The Team shall use the same styling for all written code - Done
 - 12.2. The Team shall use spaces for indentation - Done
 - 12.3. The Team shall write thorough documentation and comments - Done
 - 12.4. The Team shall own the codebase and act responsibly - Done
 - 12.5. The Team shall write tests that ensure a good working product - Done

13. Environmental Sustainability

- 13.1. The Application shall use as few resources as needed - Done
- 13.2. The Application shall work with responsible service providers - Done
- 13.3. The Application shall minimize its footprint and make continuous efforts to improve further on sustainability - Done

Team Evaluation

Lennart (Team Lead) 10

- Project Planning
- Finished the M3V2 Revision
- Organize All The Team Meetings
- Wrote a Test Plan
- Wrote the QA Test Plans
- Wrote the Security Check
- Participated in the Code Review
- Attended All Team Meetings
- Responsive On Discord

Rafael (Backend Lead) 10

- Wrote the Matching Algorithm
- Implemented QOL Improvements for the Platform
- Participated in the Code Review
- Reviewed all internal Pull Requests
- Wrote a Test Plan
- Lots of bug fixes
- Tech Support for the Team
- Attended All Team Meetings
- Responsive on Discord

Edmund (Frontend Lead) 10

- Helped revise the Footer
- Multiple frontend improvements
- Participated in the Code Review
- Revised the Register Layout
- Spaced out the Navbar
- Attended All Team Meetings
- Responsive on Discord

Julio (Docs-Editor) 10

Change the paths on the navbar to be more accurate
Worked on the FAQ
Implemented other routing changes
Recorded all the meetings
Uploaded Feedback video from Review session
Made a Test Plan
Attended All Team Meetings
Responsive on Discord

Conrad (Github-Admin) 10

Wrote the Overarching Test Plan
Wrote the Tables in which we organized the Testing
Implemented small fixes on the footer
Various bug fixes
Wrote a Testing Plan
Attended All Team Meetings
Responsive on Discord

Team Evaluation

Lennart (Team Lead) 8

- Project Planning
- Organized meetings and assigned tasks
- Initial Setup of the Project
- AWS Admin
- Setup CI/CD Pipeline
- Coded parts of the backend
- Various Bug fixes
- Setup the S3 for Images
- Database Management/Schema

Signature: *Lennart Richter*

Rafael (Backend Lead) 9

- Wrote lots of the Backend Code
- Wrote pieces of the Frontend
- Developed the Matching Algorithm
- Developed the Settings Page
- Reviewed code changes
- Implemented MVC architecture in the backend
- Led the coding style for the team
- Led the research for technologies within the team
- Attended all meetings and classes
- Signature: *Rafael Fabiani*

Edmund (Frontend Lead) 8

- designed & created the logo for SwampStudy
- created the login page FRONTEND
- created terms and conditions page + logic to require users to agree
- created the mockups/design the frontend pages.
- created well drawn storyboards.
- created the skeleton matching page based on the mockups
- created the Figma pages
- made some quality UI improvements to the navbar & registration page
- created usability test for filling out user profile

- Attended every meeting and class

Signature: *Edmund Huang*

Julio (Docs-Editor) 7.5

- Took notes during all 14 group meetings, ensuring that crucial information and decisions were accurately documented.
- Notified the professor of said meetings, notified him of updates him on what we discussed during meetings.
- Played a role in refining early milestone documents, contributing edits to enhance clarity and coherence.
- Made half of the first use cases, analyzing requirements and implementing functional solutions to meet project objectives.
- Made half of the wireframes, translating conceptual ideas into visual representations
- Coded my "About Me" page
- Coded initial draft of the forum, trying to match the groups use cases
- Coded initial registration page, again trying to match use cases
- Scheduled private library rooms for the team almost every week
- Had perfect attendance to all meetings and responded to every discord notification involving me

Julio Reyes

Signature:

Conrad (Github-Admin) 8

- Helped maintain the organization of the github branches
- Helped develop forums api
- Worked on the footer which includes
- Social media buttons
- And different pages.
- Worked on regex
- Debugging
- Did UML class diagram
- Used connections to bring in a UI/UX researcher from Cisco
- Competitive analysis
- Set up and design for the Questionnaire and usability test table
- Assist in Use Cases

- Conducted interviews with participants on Usability
- Attended every meeting and class

Signature:

Conrad Choi

Post Analysis

Being in charge of a team has led me and everyone involved to confront some interesting challenges and gave us the opportunity to grow as a unit. First of all I am very proud of what my team was able to accomplish and how we became a close knit group throughout the process that is CSC648. Fortunately we didn't have any major issues that seriously threw us off our path or led to long term issues, everything henceforth are minor things and improvements I would make were the semester to start again tomorrow. The first thing that I would change is being more insistent on equal participation within certain segments of the code. I think we had a mix of very eager teammates and some who were less comfortable asserting themselves in a coding environment especially when confronted with a foreign language or processes that weren't comfortable. I should have pushed people more outside the comfort zones and made sure we checked in more to give everyone equal exposure to the process of building this app in all contexts. This led to some isolated knowledge as well where some people were super well versed in certain aspects of the application while others were better versed in other areas completely leading to an incomplete picture for most people. I think having a better overview and discourse of the application as a whole and making sure everyone has a holistic overview would have been greatly beneficial for learning. Besides this minor complaint there's really not much to be disgruntled with. Everyone came to every meeting and class, we had everyone finishing their assigned

tasks in a timely manner and respectful communication between team members. I consider myself quite fortunate to have led a team of this caliber and with the amount of buy-in that we had. By the end of the semester you could feel the passion from everyone for the product and it made the whole experience even more rewarding, getting to share that with other people.