

## Final Project for SW Engineering Class CSC 648-848 Section 01 Summer 2021

### FitHub

**Team Info:** Team 07 | Error 404

**Project Application And Name:** Find a fitness partner - “FitHub”

**Team Members:**

1. Vidhi Vora (Team Lead and GitHub Master) - vvora@sfsu.edu
2. Roberto Simental (Front-End Lead) - rsimental@mail.sfsu.edu
3. Johnson Nguyen (Back-End Lead) - Jnguyen63@mail.sfsu.edu
4. Zhinan Zhao
5. Eduardo Hernandez
6. Ziming Wang
7. Michael Satumba

**Milestone:** Milestone 5

**URL:** <http://100.26.92.104:3000/>

**Date:** 03 August 2021

**History:**

Version	Date
M5V1	03 August 2021
M4V2	02 August 2021
M4V1	30 July 2021
M3V2	30 July 2021
M3V1	22 July 2021
M2V2	22 July 2021
M2V1	08 July 2021
M1V2	13 July 2021
M1V1	22 June 2021

## Table Of Contents

2. Product summary	3
3. Milestone documents – M1-M4	5
4. Screenshots of actual final product	141
5. Screenshots of key DB tables	144
6. Screen shot(s) of your task management system	146
7. Team member contributions	147
8. Post analysis – lessons learned	148

## **2. Product summary:**

- Name of the product: Fithub
- Final P1 (Mandatory):

### **Registered User:**

1. Registered users shall be able to access About us page
2. Registered users shall be able to access Contact us page
3. Registered users shall be able to access Support Page
4. Registered users shall be able to send workout invites to other users.
5. Registered users shall be able to select their interests
6. Registered users shall be able to edit their information
7. Registered users shall be able to update their profile picture
8. Registered users shall be able to delete their profile picture
9. Registered users shall be able to change their account passwords
10. Registered users shall be able to delete their account from Fithub
11. Registered users shall be able to access About us page
12. Registered users shall be able to access homepage

### **Searching People:**

13. Registered users shall be able search for buddies with similar interest
14. Registered users shall be able search for buddies nearby
15. Registered users shall be able to filter search options based on their interests.

### **Friends:**

16. Registered users shall be able to find friends to exercise with.
17. Registered users shall DM (direct message) other registered users only if they are friends
18. Registered users shall view friend's event postings

### **Events:**

19. Registered users shall be able to create events
20. Registered users shall be able to delete events they created
21. Registered users shall be able to edit the events they created
22. Registered users shall be able to join an event
23. Registered users shall be able to exit from an event
24. Registered users shall be able to rejoin an event
25. Registered users shall invite people to events they created

### **Chats:**

26. Registered users shall be able to create a private chat

**Web Application:**

27. Web Application shall have About us Page
28. Web Application shall have Contact us page
29. Web Application shall ask user to log in
30. Web Application shall display user's profile
31. Web Application shall show notifications to the user
32. Web Application shall allow user to check their messages
33. Web Application shall allow user to check event dates they are planning to go
34. Web Application shall allow user to change password of their account
35. Web Application shall allow user to deactivate their account
36. Web application shall show events occurring nearby
37. Web Application shall show people recommendation with similar interest
38. Web Application shall allow user to log out
39. Web Application shall allow user to delete the account
40. Web Application shall allow user to search for people using various filter options
41. Web Application shall show people recommendation with similar interest
42. Web application has admin account to approve events

- The unique points of Fithub:

1. Compared to many social platforms that already exist on the market, Fithub has clear user targeting: people who enjoy group exercise. So we combined group exercise events posting with social media. There will be no ads because we use paid membership.
2. Fithub has straightforward social features and User UI. White background with a clean navigation bar will give our users an experience of comfort and great efficiency.
3. For the best user experience, we only use ads specific to the interests of our users and related to working out. We also provide a premium subscription with unique benefits and perks for our users.

- Url: <http://100.26.92.104:3000/>

### 3. Milestones:

#### MILESTONE1

##### SW Engineering CSC648/848 Summer 2021

##### FitHub

**Project Application And Name:** Find a fitness partner - “FitHub”

**Team Info:** Team 07 | Error 404

##### Team Members:

1. Vidhi Vora (Team Lead and GitHub Master) - vvora@sfsu.edu
2. Roberto Simental (Front-End Lead) - rsimental@mail.sfsu.edu
3. Johnson Nguyen (Back-End Lead) - Jnguyen63@mail.sfsu.edu
4. Zhinan Zhao
5. Eduardo Hernandez
6. Ziming Wang
7. Michael Satumba

**Milestone:** Milestone 1

**Date:** 22 June, 2021

##### History:

Version	Date
M1V1	22 June, 2021
M1V2	13 July, 2021

## Table Of Contents

Executive Summary	3
Main Use Cases	5
List of main data items and entities	16
Initial list of functional requirements	18
List of non-functional requirements	23
Competitive analysis	28
High-level system architecture and technologies use	33
Checklist	34
Team Contribution	35

## **1. Executive Summary**

Ever since COVID-19 struck the world in March of 2020, people have stayed inside due to safety protocol. Staying inside offers multiple issues for human beings. It breeds loneliness, inactivity, and in some cases mental illness. There is a reason why solitary confinement is the worst punishment a prisoner can receive. With the progressing development of vaccines, we have come up with a solution to connect people, foster activity, and help people in search of a better life post-COVID-19. Our solution here is "Fithub".

"Fithub" will be the first web application with a simple user interface that successfully finds workout partners for people based on their interests. With customizable interest options and use of high-quality software engineering practices to match people, the focus on finding a partner with maximum compatibility and one who complement each other increases.

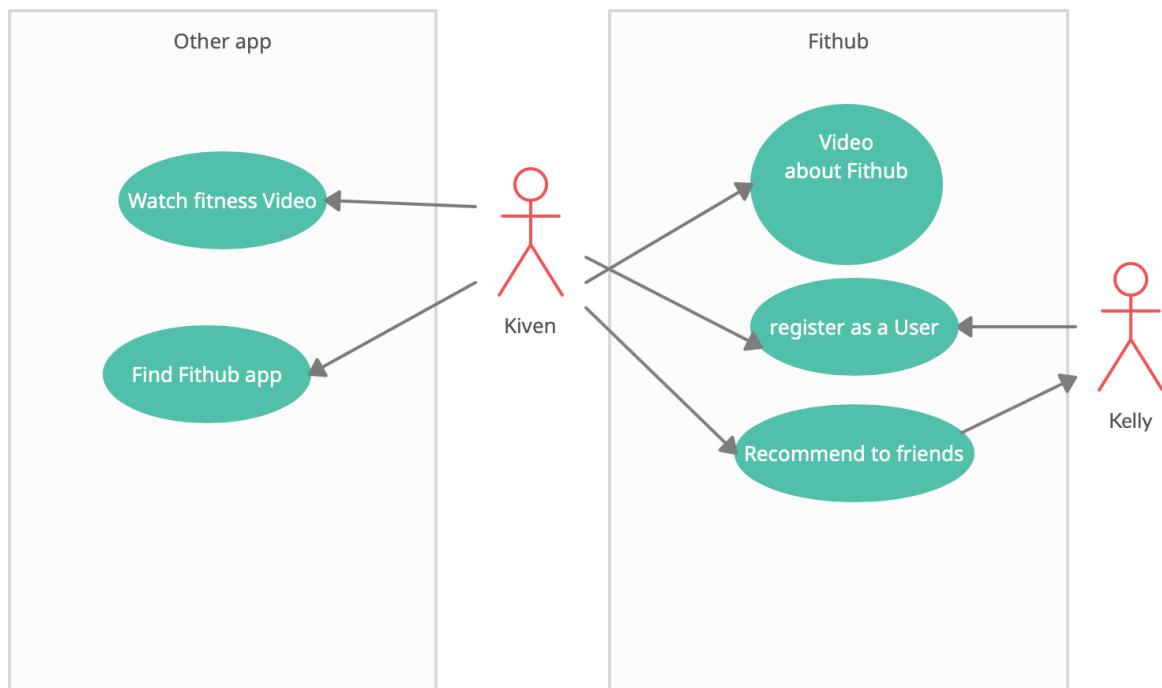
The goal of our application is to bring people with shared interests together so that they can help and motivate each other to get fit. With our app we can help tailor their exercise experience to their needs and on their schedule. Our app lets users choose their workout and personal goals to help them find nearby people that will join them in their journey. Workouts include various indoor activities such as gym, yoga, meditation, etc and outdoor activities like hiking or sports games with an effort to reach out to a wide span of audience and having something for everyone.

Fithub offers many free features like finding and posting events, adding people as friends, and filtering for specific exercises. We also offer a premium membership to enhance the user experience. Unlimited daily match requests, more specific filter options like same-gender only, or experienced lifters only, and access to our unique service the Buddy Pass are some of the features included in our membership plan. The Buddy Pass pairs up people who don't own a gym membership with someone who owns a full membership at a gym that allows them to bring a friend along free of charge. With the Buddy Pass users can experience many new gyms near their area without signing a contract or paying for overpriced 1 day passes. Fithub also allows users to pay for one-day access to the Buddy Pass or to refresh their daily match request limit, so that users have

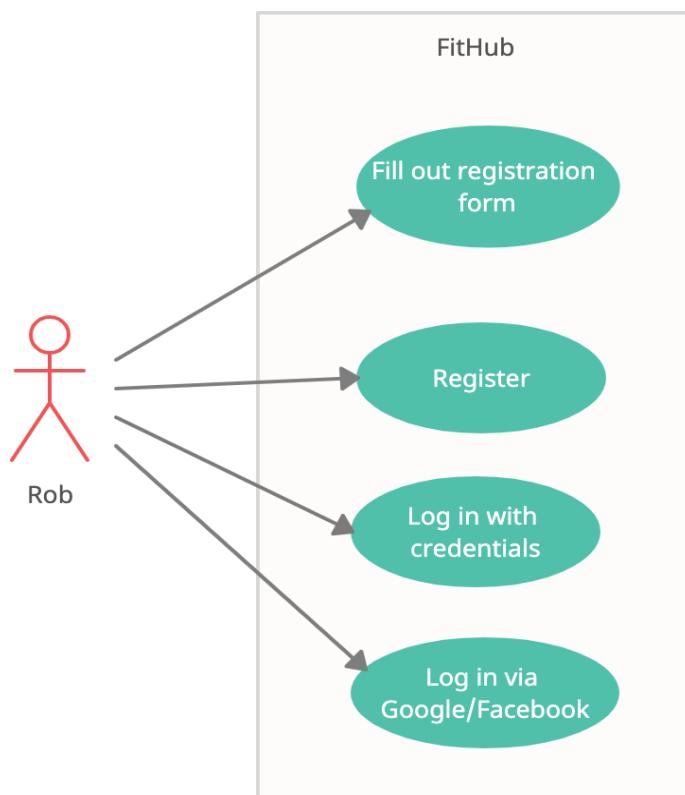
the opportunity to try out our premium features and the freedom to decide what works best for them.

## 2. Main Use Cases

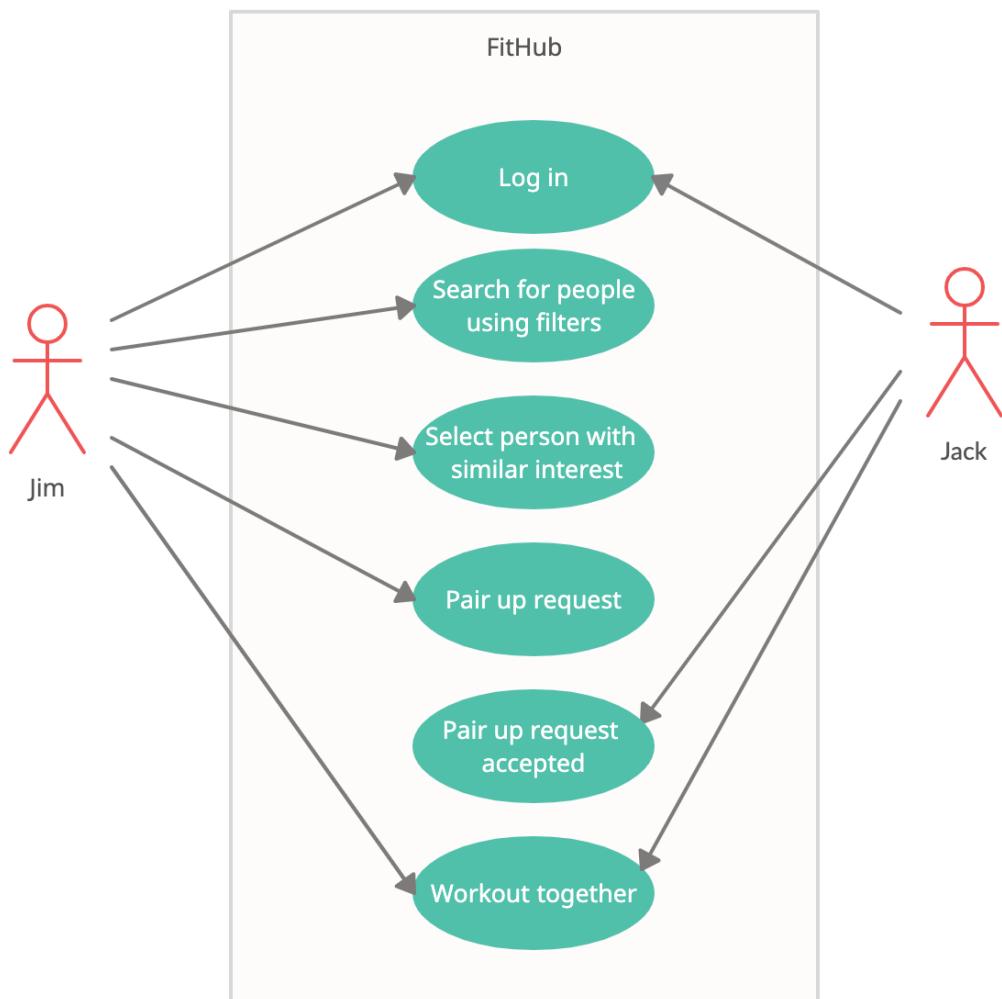
<b>Title</b>	Finding the Fithub App
<b>Actors</b>	Kiven, Kelly
<b>Description</b>	Kiven is a 19 year college student, he rarely goes to the gym. He realizes he is unfit and has become lazy during the pandemic lockdown. He started watching fitness videos at home and in one of the ads found out about Fithub - an app to find a workout buddy. He visits the app page, explores its features and realizes how cool it is to workout with a buddy. So he goes ahead and registers himself. He tells Kelly about this app - Fithub. Kelly too gets excited and registers herself.



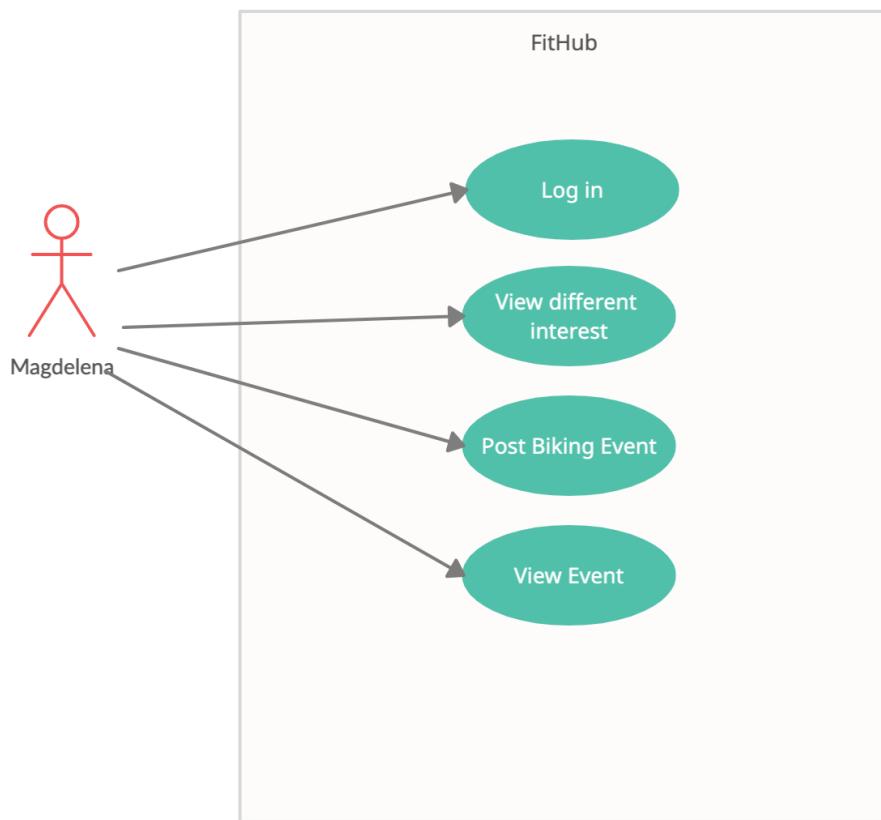
<b>Title</b>	Registration and Login
<b>Actors</b>	Rob
<b>Description</b>	Rob is a 19 year college student and is planning to begin working out post COVID-19. But, he is finding a lack of motivation to exercise. So he fills out a registration form and registers himself at FitHub. Once he logs in the system, he can find a partner and get motivated to workout together.



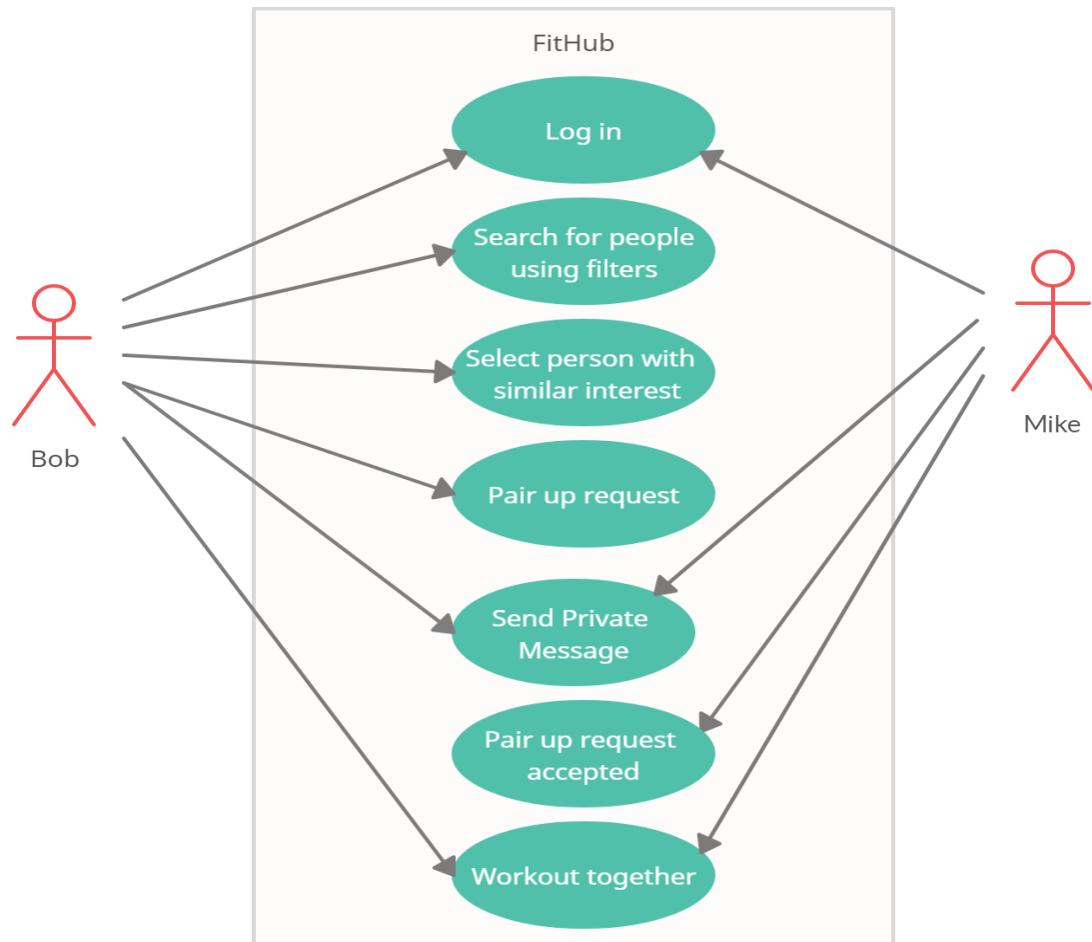
<b>Title</b>	Find A Partner
<b>Actors</b>	Jim and Jack
<b>Description</b>	Jim's buddy bailed on him last minute and he still wants to go to the gym with someone. He checks the app to see if anyone is going to his gym nearby. He finds Jack who is planning on lifting weights too. They pair up and meet each other at the gym.



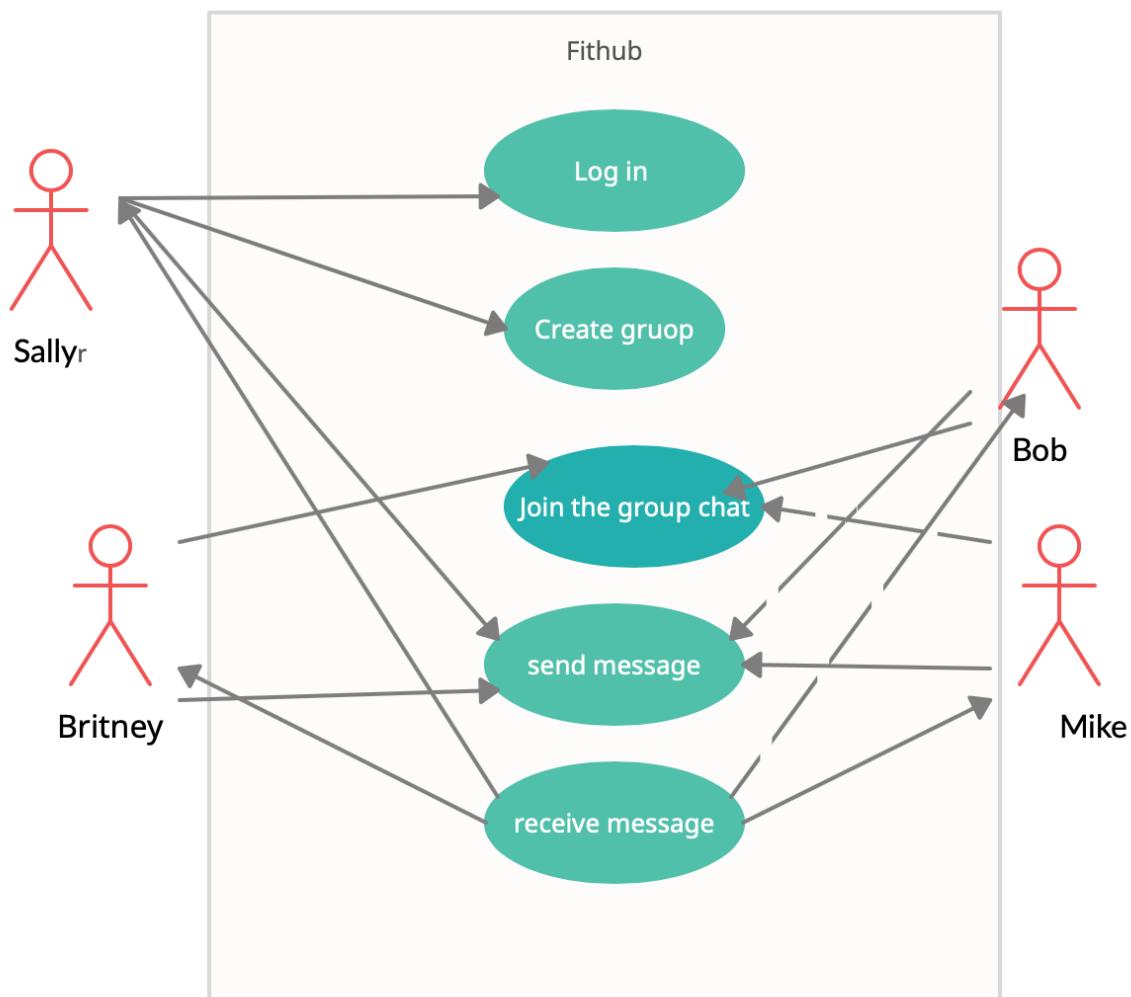
<b>Title</b>	Posting for events like - group biking
<b>Actors</b>	Magdelena
<b>Description</b>	Magdelena is planning on a bike trip this week. Magdelena used to be part of the bicycle team in her college. Magdelena misses the camaraderie with biking with others. Magdelena posts an event on the FitHub web application knowing that others will join her for the bike ride.



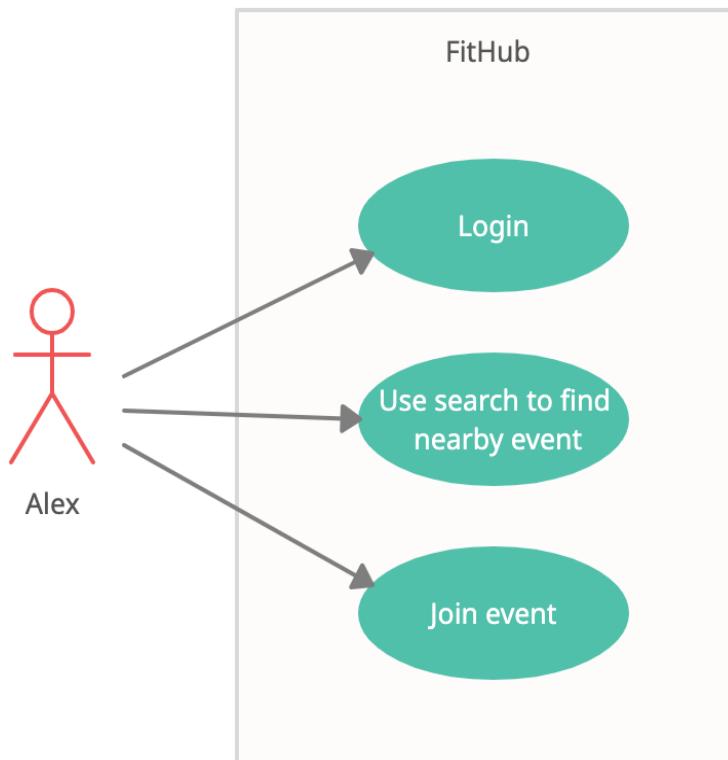
<b>Title</b>	Private Chat
<b>Actors</b>	Bob and Mike
<b>Description</b>	Bob and Mike had plans to meet up at the park at 7 pm to jog. Unfortunately, Bob is stuck in traffic and is going to be late. He messages Mike using the app to let him know that he will be 15 mins late. Mike then lets Bob know that he will wait and get started warming up.



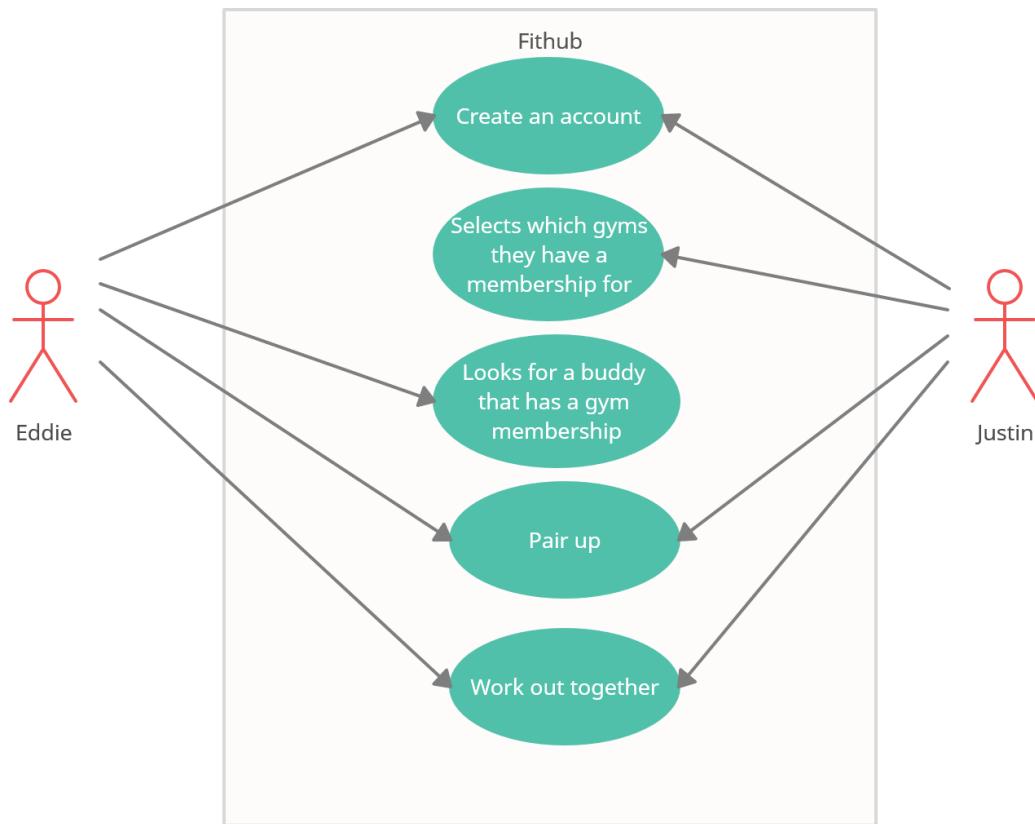
<b>Title</b>	Group Chat
<b>Actors</b>	Britney, Sally, Bob, Mike
<b>Description</b>	Britney, Sally, Bob and Mike are planning to workout together. However, Sally might not be able to come to the workout session. To inform her workout buddies, she informed them through our group chat.



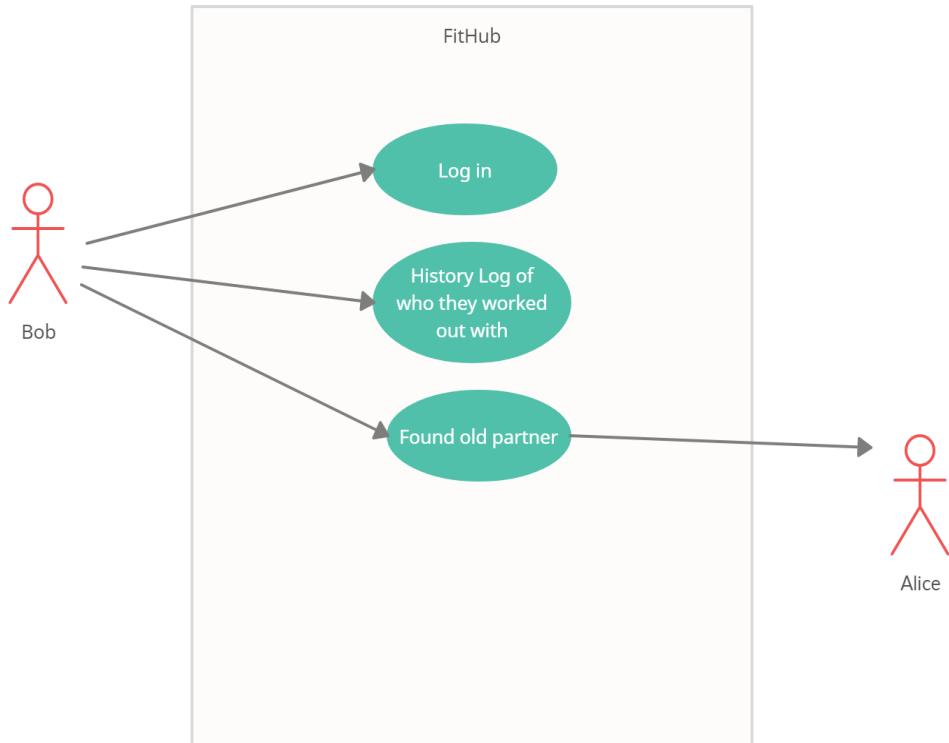
<b>Title</b>	Finding events nearby
<b>Actors</b>	Alex
<b>Description</b>	Alex has been working at home since the COVID-19. He wants to find some group biking events near his school to attend. The function allows him to find biking events by providing his location and the start time he expected.



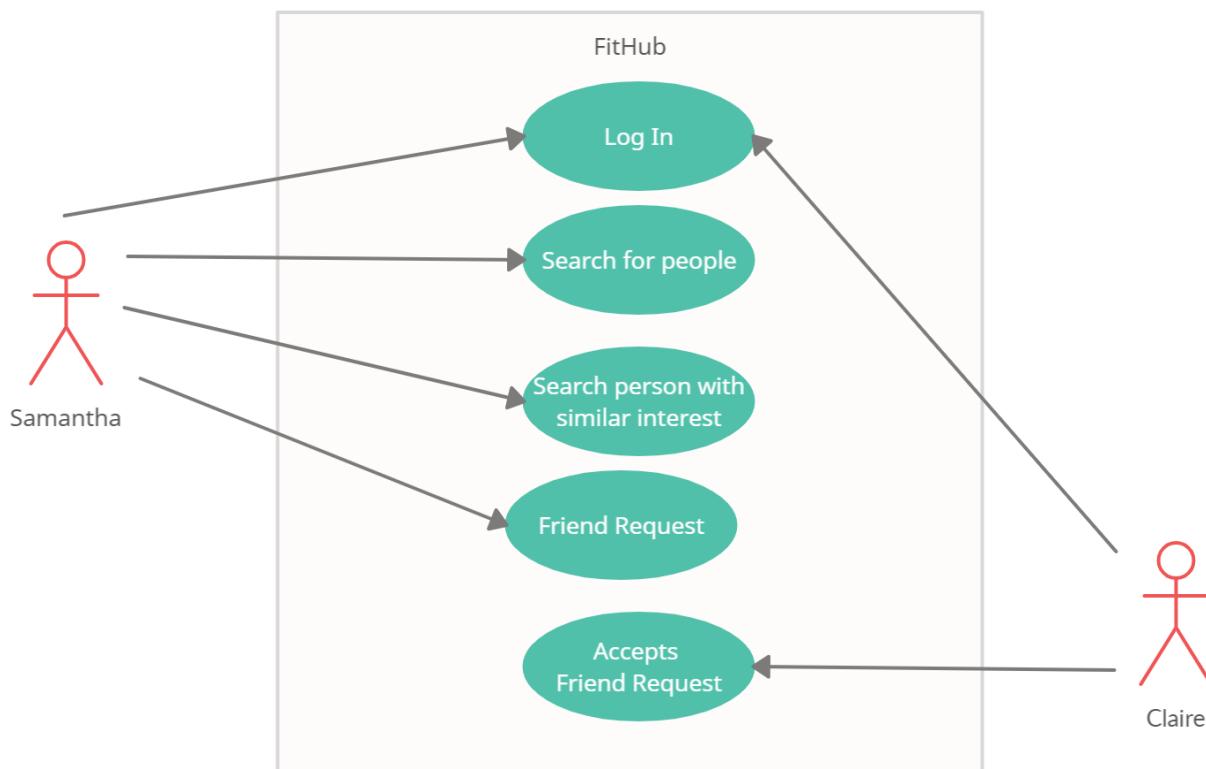
<b>Title</b>	Buddy Pass System
<b>Actors</b>	Eddie and Justin
<b>Description</b>	<p>Justin currently has a gym membership, and Eddie wants to start going to the gym. Eddie is nervous to go alone so he utilizes our “Buddy Pass System”. This system allows newcomers to have a workout partner and not be afraid to go to the gym by themselves.</p> <p>**NOTE** Our Buddy Pass System does not give gym passes, it only allows users to have a gym buddy. At the selected gym the users want to go to, they can request a gym pass. **END NOTE**</p>



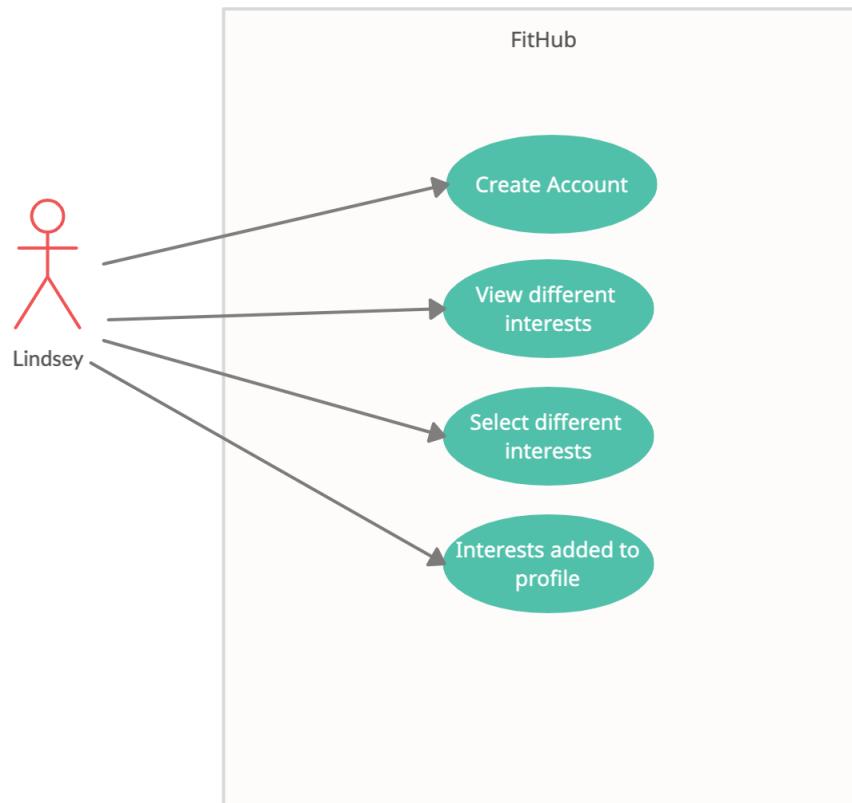
<b>Title</b>	History Log
<b>Actors</b>	Bob and Alice
<b>Description</b>	Bob could not remember the last workout partner who he enjoyed working with. So he looks into his previous logs and from there connects with Alice .



<b>Title</b>	Friends
<b>Actors</b>	Samantha and Claire
<b>Description</b>	<p>Samantha has a hard time finding people she feels comfortable meeting up with she met online. She has a great experience with Claire and they get along really well during a workout session. Samantha sends Claire a friendship invitation, so they can meet up with each other whenever they both have free time again. Claire accepts and now they can see whenever each other logs in and is looking to find a workout buddy.</p>



<b>Title</b>	Selecting interests
<b>Actors</b>	Lindsey
<b>Description</b>	Lindsey is a 35 year old woman who has been inactive during COVID-19. Ever since she got the vaccination, she is eager to exercise outdoors again. Lindsey in the past was interested in running and surfing. Lindsey selects these two activities in FitHub to see other like minded people around her. Knowing that it will be more enjoyable to run and surf with others.



### **3. List of main data items and entities**

- Guest user
  - A user who hasn't registered or provided their information to FitHub
- Registered user
  - A user who has successfully created their account with FitHub and can access all the app features
- User interests
  - This information will be used by FitHub to show recommendations of people with similar interest
- User Profile
  - All the user related information that can be used to create its profile
- User logs
  - Previous workout partners or events that the user has been to can be looked in the user logs
- Activities
  - All the indoor and outdoor activities that Fithub allows a user to search or select as interest.
- Gym membership
  - A user's gym membership information. This information will be used in order to sent buddy pass request by other users
- Buddy Pass Owner
  - A user having gym membership and willing to share his gym pass with a friend so that they can workout together
- Events
  - A user will be posting events occurring that can be viewed by other users to join if it matches their interest.
- Friends
  - Two people that match based on interests and location.can become friends by sending and accepting the friend request
- Friend List
  - List of people the users are friends with.
- Private chat

- This function can be accessed by people who are friends and can send private message on our web browser
- Group chat
  - This function can be accessed by people who are friends and belong in a group of people having a conversation.
- Clubs
  - Users can create physical activity clubs i.e. biking group, etc so as other users can join them
- Club owner
  - Owner of a distinct physical activity club.
- Club List
  - Particular set of clubs which users can view and/or join.

#### **4. Initial list of functional requirements**

##### **Guest User:**

1. Guest users shall be able to view events posted by other users.
2. Guest users shall be able to select their interests.
3. Guest users shall see other users' user names of people looking to work out nearby.
4. Guest users shall be able to register.
5. Guest users shall be able to Log in as registered users.
6. Guest users shall be able to access the homepage.
7. Guest users shall be able to access the About us page.
8. Guest users shall be able to access the Contact us page.
9. Guest users shall be able to access the Support page
10. Guest users shall be able to delete their account from FitHub

##### **Registered User:**

11. Registered users shall be able to access homepage
12. Registered users shall be able to access About us page
13. Registered users shall be able to access Contact us page
14. Registered users shall be able to access Support Page
15. Registered users shall be able to send workout invites to other users.
16. Registered users shall be able to select their interests
17. Registered users shall be able to edit their information
18. Registered users shall be able to update their profile picture
19. Registered users shall be able to delete their profile picture
20. Registered users shall be able to deactivate their account
21. Registered users shall be able to save their frequent searches
22. Registered users shall be able to change their account passwords
23. Registered users shall be able to limit how much of their profile is shown in other user's searches results.
24. Registered users shall be able to send an introductory message to a different user in their search result.
25. Registered users shall be able to share their real time location with FitHub

26. Registered users shall be able to delete their account from Fithub

## **Searching People**

27. Registered users shall be able to have a certain number of swipes per day.
28. Registered users shall be able search for buddies with similar interest
29. Registered users shall be able search for buddies nearby
30. Registered users shall be able to filter search options based on their interests.
31. Registered users shall be able to filter search options based on how far they are willing to travel.
32. Registered users shall be able to remove people from their search results

## **Friends**

33. Registered users shall be able to find friends to exercise with.
34. Registered users shall be able to add other registered users as friends.
35. Registered users shall be able to access their friend list
36. Registered users shall DM (direct message) other registered users only if they are friends.
37. Registered users shall have multiple friends if they choose to.
38. Registered users shall have 0 friends if they choose to.
39. Registered users shall have the ability to unfriend a former friend.
40. Registered users shall have the ability to block a former friend.
41. Registered users shall have the ability to report a former friend.
42. Registered users shall view friend's club postings.
43. Registered users shall view friend's event postings.
44. Registered users shall be able to decline friend requests.
45. Registered users shall be able to accept friend requests.

## **Events**

46. Registered users shall be able to create events
47. Registered users shall be able to delete events they created
48. Registered users shall be able to edit the events they created
49. Registered users shall be able to join an event
50. Registered users shall be able to exit from an event

51. Registered users shall be able to rejoin an event
52. Registered users shall invite people to events they created

## **Clubs**

53. Registered users shall be able to create the club
54. Registered users shall be able to update the picture of the club they created
55. Registered users shall be able to edit the club name they created
56. Registered users shall be able to delete the club they created
57. Registered users shall be able to delete the club picture they created
58. Registered users shall be able to send request to join the club
59. Registered users shall be able to leave a club
60. Registered users shall be able to send request rejoin a club
61. Registered users shall be able to add members to the club they created
62. Registered users shall be able to remove members from the club they created
63. Registered users shall be able to post in club
64. Registered users shall be member of multiple clubs
65. Registered users shall be admin of multiple clubs
66. Registered users shall be able to invite people to club
67. Registered users shall be able to accept the joining request for club they created
68. Registered users shall be able to reject the joining request for club they created

## **Chats**

69. Registered users shall be able to create a group chat
70. Registered users shall be able to create a private chat
71. Registered users shall be able to invite friends to the group chat
72. Registered users shall be able to remove people from group chat
73. Registered users shall be able to add unknown people from the chat
74. Registered users shall be able to post in the a group chat
75. Registered users shall be able to leave a group chat
76. Registered users shall be able to dissolution the group chat they cheated
77. Registered users shall be able to recall their posts
78. Registered users shall be able to reject to join in a private chat
79. Registered users shall be able to reject to join in a group chat

80. Registered users shall be able to accept to join in a private chat
81. Registered users shall be able to accept to join in a group chat

### **Buddy Pass**

82. Registered users shall be able to view people with gym membership
83. Registered users shall be able to put request to access buddy pass
84. Registered users shall be able to accept the request to access buddy pass
85. Registered users shall be able to reject the request to buddy pass

### **History logs**

86. Registered users shall access their event visits in logs
87. Registered users shall access their workout pairing information in logs
88. Registered users shall be able to access their workout partner's information from logs

### **Web Application**

89. Web Application shall have About us Page
90. Web Application shall have Contact us page
91. Web Application shall ask user to log in
92. Web Application shall display user's profile
93. Web Application shall show notifications to the user
94. Web Application shall show friend requests
95. Web Application shall allow user to check their messages
96. Web Application shall allow user to check event dates they are planning to go
97. Web Application shall allow user to check updates of the club
98. Web Application shall allow user to change password of their account
99. Web Application shall allow user to deactivate their account
100. Web Application shall show buddy pass requests
101. Web Application shall show the logs of user activities
102. Web application shall show events occurring nearby
103. Web Application shall show people recommendation with similar interest
104. Web Application shall allow user to log out
105. Web Application shall allow user to delete the account

106. Web Application shall allow user to see list of friends
107. Web Application shall allow user to see list of clubs
108. Web Application shall allow user to search for people using various filter options

## **5. List of non-functional requirements**

### **System Requirements**

1. The application shall be able to run on the android system(Version 10).
2. The application shall be able to run on the android system(Version 11).
3. The application shall be able to run on Windows10(Version 1803).
4. The application shall be able to run on Windows10(Version 1809).
5. The application shall be able to run on Windows10(Version 1903).
6. The application shall be able to run on Windows10(Version 2004).
7. The application shall be able to run on Windows10(Version 20H2).
8. The application shall be able to run on Windows10(Version 21H1).
9. The application shall be able to run on Ubuntu 21.04.
10. The application shall be able to run on iOS 12.
11. The application shall be able to run on iOS 13.

### **Marketing**

12. The application shall have the company logo visible on every page.
13. The application shall have the company name visible on every page.
14. The application shall have non-intrusive space dedicated to relevant ads.
15. The application shall have meta content for SEO purpose.

### **Legal requirements**

16. The application shall prompt for Terms and Conditions for each new user to accept.
17. The application shall have all users verify that they're 18 years of age or older.
18. The application shall have copyright information on every page.

### **Privacy**

19. The application shall prompt user to accept policies when creating account
20. The application shall ensure user's data will be shared between registered users
21. The password shall be needed for log-in action.
22. The user's passwords shall be hashed using Bcrypt.
23. The application's Network will be protected by the AWS Firewall
24. The application shall not allow more than 3 incorrect login attempts in 24 hours.

25. The application shall protect the privacy of users.
26. The application shall present all privacy policies appropriately to all users.
27. Users' private information should not be shared with 3rd platforms.
28. The basic security of the site shall all be tested.

## User Interface

29. The application shall be aesthetically pleasing
30. The application shall have complementary colors
31. The application shall be easy to navigate for everyone
32. The application shall have a search bar
33. The application shall have a simple user interface
34. The application shall have a Filter button
35. The application shall have a Register button
36. The application shall have a Login button
37. The application shall have a Home button
38. The application shall have a Search button
39. The application shall have a Private Chat room
40. The application shall have a Group Chat room
41. The application shall have a Terms and Condition page
42. The application shall have a Home page button
43. The application shall have a User page button
44. The application shall have a Logout Button
45. The application shall have a Profile Picture of the User
46. The application shall have an Upload button
47. The application shall have a Block button
48. The application shall have an Unblock button
49. The application shall have pictures regarding to the activity
50. The application shall have a Nearby Button
51. The application shall have aesthetically pleasing photography
52. The application shall have smooth scrolling
53. The application shall have footer on every page
54. The application shall have header on every page
55. The application shall provide clear instructions on usability of features

## **Usability**

56. The application shall be optimized for mobile browsers.
57. The application shall store data in the chosen database
58. The user personas shall be used.
59. The application shall have a Support page
60. Users that are older than the age of 18 shall be able to use the application to post information, search events, and register after spending 5 minutes reading the support page.

## **Performance and scalability**

61. The Response Time shall be 3 seconds
62. The Throughput shall be between 128 MiB/s and 250 MiB/s
63. The system availability shall be 99%(AWS)
64. A REST service call shall respond with 500 milliseconds,
65. A synchronous database request shall respond with 1000 milliseconds,
66. A batch process shall complete its processing within 60 minutes.
67. The application shall be able to handle large number of users
68. The application shall be developed using microservices architecture enabling easy maintenance and launching/editing features to FitHub
69. The application shall be capable to recover from failures
70. The application shall provide user the requested data
71. The application shall be able to update the chat data in real time
72. The application shall have live updates regarding events

## **Coding Standards**

73. The code shall be readable
74. The code shall be well aligned and properly formatted
75. The code shall be commented wherever needed
76. The code shall be organized with one coding style
77. The code shall be accessible to all the developers
78. The code versions shall be maintained using GITHUB
79. The code shall be pushed to different branches

80. The code shall be merge to the master once approved by leads and leader
81. The code shall be in the repository that the professor has created for us

### **Fault Tolerance**

82. The application shall be able to reload itself if loading request failed
83. The application shall try reconnecting database if the request fails

### **Storage**

84. The server shall have 16 concurrent Integrations
85. The server shall have 20 CPU Cores
86. The server shall have 12(RAID) Data Disk
87. The server shall have 8 Volume Size (GiB)

### **Expected Load**

88. The application shall support as many users AWS can support
89. The application shall support as many user requests as AWS can support
90. The application shall support as many database connections as AWS can support

### **Portability and compatibility**

91. The application shall be able to run on Chrome 91.0.4472.
92. The application shall be able to run on Chrome 92.0.4515.
93. The application shall be able to run on Firefox 88.0.1.
94. The application shall be able to run on Firefox 89.0.1.
95. The application shall be able to run on Microsoft Edge 91.0.864.37.
96. The application shall be able to run on Microsoft Edge 46.03.4.5155.
97. The application shall be able to run on Safari 5.
98. The application shall be able to run on Safari 14.
99. The application shall be able to run with 1.8GHz or faster processor. Quad-core or better recommended.
100. The application shall run with 2 or more GB of RAM.
101. The application shall need a minimum of 1 GB up to 10 GB available space, depending on features installed.
102. The application shall be tested and deployed utilizing Docker and Kubernetes.

103. The application shall be continuously running 24/7 except for reasons of maintenance.

## 6. Competitive analysis

	FitHub	Facebook Groups	Workout Buddies	Strava	Runsocial	Athlete Network	Meetup.com
<b>Link</b>		<a href="https://www.facebook.com/">https://www.facebook.com/</a>	<a href="https://workoutbuddies.com/">https://workoutbuddies.com/</a>	<a href="https://www.strava.com/">https://www.strava.com/</a>	<a href="https://www.runsocial.com/how">https://www.runsocial.com/how</a>	<a href="https://www.athletenetwork.com/">https://www.athletenetwork.com/</a>	<a href="https://www.meetup.com/">https://www.meetup.com/</a>
<b>Strengths</b>	- Strictly for physical activities - Allows to look for people with similar interest	-Mass users on Facebook, easily accessible	-Downloadable application on IOS/ Google Play -Simple UI	-Downloadable application on IOS/ Google Play	Focuses on running activity indoor	Dedicated for athletes looking to pursue their career in athletics	Multiple events and clubs posted daily
<b>Weaknesses</b>	-Less foot traffic -No mobile app	-Saturated market in Facebook -Not specially dedicated for fitness believers	-Does not have a web application -Not good reviews on play store -Does not have option to post events -Does not have option to create or join clubs	-Not a lot of clubs -Dedicated for only running and cycling	-Less choice of activities	-Not for everyone	-No functionality to look for people with similar interest
<b>Pricing</b>	Free	Free	Varies based on plan	\$5/month	Free	Free	\$30/month
<b>Social</b>	-A direct	-Facebook	-Facebook	-Blog	-Twitter,	-Facebook	-Blog

<b>Media</b>	and a group messaging feature -Allows posting of events		ok, YouTube, Instagram, Twitter	Posts, Medium, Instagram, Twitter, YouTube	Facebook, Instagram	, Twitter, Linkedin, Pinterest, Instagram	Posts, Instagram, Twitter
<b>Onboarding</b>	-Extremely easy to register with -clear instructions	Well-ordered instructions, but not much -support after the first step	No updates since 2018	Sleek instructions	-Very simple and quick -within few clicks	-Simple UI but not so pleasing -Lengthy process to register	Uniform directions
<b>Usability</b>	-Aesthetically pleasing and easy to use	-Simple, easy to use	-Old styling, but simple to use	-Premier UI and UX design	-Easy to use, but only for indoors	-Complicated to navigate the app	Straightforward and trouble-free

Features	FitHub	Facebook Groups	Workout Buddies	Strava	Runsocial	Athlete Network	Meetup.com
<b>Support</b>	++	-	-	++	++	+	++
<b>UI simplicity</b>	++	+	+	++	+	+	++
<b>Private Chat</b>	+	+	+	+	--	+	+
<b>Group Chat</b>	+	+	-	+	-	++	++
<b>Matching</b>	+	-	-	-	-	+	-
<b>Events</b>	++	++	-	+	--	++	++
<b>Clubs</b>	++	++	-	++	++	++	++

<b>Buddy Pass</b>	++	-	-	-	--	-	-
<b>History Log</b>	++	-	-	++	--	+	+

+: Feature exists    ++: Feature is superior    -: Feature doesn't exist

## SUMMARY

FitHub is a web application that helps the users to search buddies with similar fitness goals. The idea is to engage and motivate more people to carry out fitness activities together. FitHub provides this platform to regulate life post COVID-19 and helps them to keep going and carry out exercise routines similar to pre-pandemic life.

FitHub at a high level could be categorized into Healthcare (Fitness) sector and therefore we say there is something for everyone that we can offer starting from people gyming together to people undergoing chiropractic & physiotherapy sessions. FitHub thus targets the wide span of audience and serves as a one stop solution. Apart from finding out people with similar exercise goals, FitHub also provides the following set of great features to make app usage a unique experience.

1. Send Friends Requests,
2. 1:1 chats,
3. Create fitness groups/rooms,
3. Build customized clubs and organize special events.

Today, No other competitor app provides all such features together in one platform.

We understand that people now resuming their life back to normal routine are going to face more challenges to search for individuals with whom they can exercise or perform other sport activities such as running, hiking, biking, yoga, etc. and we want to leverage this opportunity to help them in such difficult times.

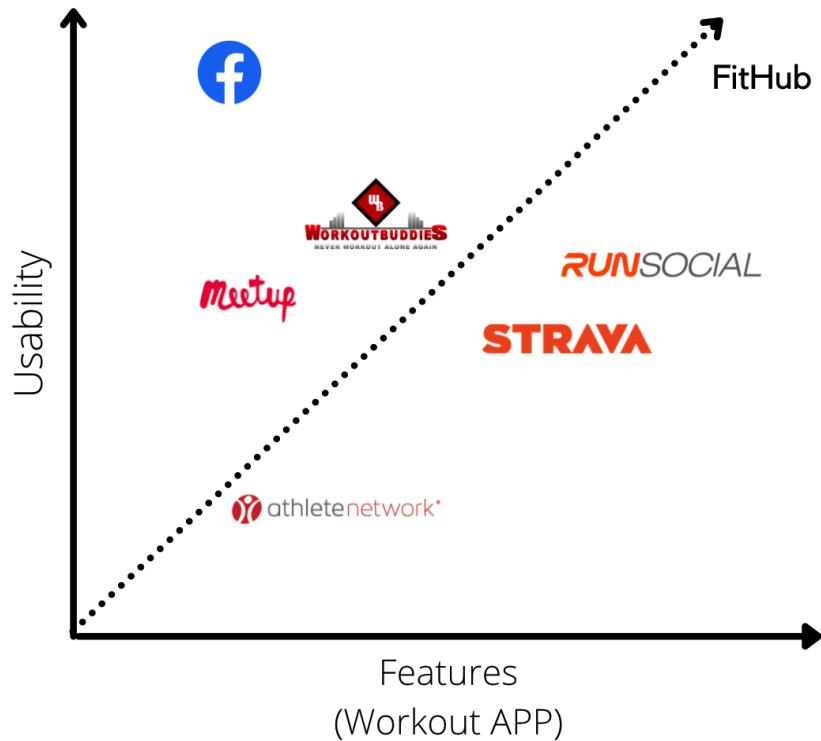
The competitive analysis:

1. Our competitors have discrete features where one allows finding buddies but doesn't allow creating groups and clubs to organize events and vice-versa. For instance, facebook

offers creating groups, adding friends, posting events but doesn't provide a feature to look for fitness partners whereas, WorkoutBuddy allows users to search for fitness partners with similar goals but does not provide any option to post events or create clubs. Fithub is the one stop solution for users to overcome this limitation.

2. Fithub offers all the same services that other websites charge for, for free. Instead of charging to use our service, we offer new premium services to improve the user's experience. Specific filters like "beginners only" allow the users more control over their workout plan and the Buddy Pass allows users to try out gyms for a much cheaper price.
3. Fithub uses targeted endemic ads that won't feel out of place or take away from the website's design or usability. Some examples are gyms, workout clothing brands, and workout supplements. The goal is to have ads that offer something our users might be interested in.
4. In this digital world with so many apps out there, Fithub is focusing on the ease of usability for our customers by dedicatedly wrapping everything about fitness into one program.

To summarize, currently there is no one stop app which offers everything that FitHub would provide to all their users.



Competitive Analysis (where we expect FitHub to stand)

## **7. High-level system architecture and technologies use**

The software components, frameworks, APIs, tools and systems that will be used in development :

- Server/ Platform Provider: Amazon Web Services (Deployment Platform)
- Operating System: Linux (Ubuntu 20.04.1)
- Tech Stack:
  - Front end stack: HTML, CSS, Javascript, Bootstrap 5.0
  - Server side language: Javascript, NodeJs v14.17.0
  - Frameworks: Express JS
- Databases: MySQL(8.0.25)
- Supported Browsers: Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Edge, Internet Explorer

## 8. Checklist

No.	Item	Status
1	Team found a time slot to meet outside of the class	DONE
2	Github master chosen	DONE
3	Team decided and agreed together on using the listed SW tools and deployment server	DONE
4	Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	ON TRACK
5	Team lead ensured that all team members read the final M1 and agree/ understand it before submission	DONE
6	Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	DONE

## 9. Team Contribution

No.	Team Member	Contribution
1	Vidhi Vora	<ul style="list-style-type: none"> <li>○ Being the Team Lead, introduced everyone to the task management tool (Asana) which helps to manage, assign and keep track of all tasks</li> <li>○ Setting internal deadline for M1, so as to have enough time for changes if advised by instructor</li> <li>○ Organizing team meeting every day except weekend with the agenda sections to discuss and work on, setting deadline for individual sections</li> <li>○ Worked in developing the content for following sections along with other teammates: <ul style="list-style-type: none"> <li>■ Executive Summary</li> <li>■ Main Use Cases</li> <li>■ Initial list of functional requirements</li> <li>■ Competitive analysis</li> </ul> </li> <li>○ Involved in discussing with team mates on <ul style="list-style-type: none"> <li>■ List of main data items and entities</li> <li>■ Non-functional requirements</li> </ul> </li> </ul>
2	Johnson Nguyen	<ul style="list-style-type: none"> <li>○ I contributed to the Use Cases, Use Cases Diagram, List of Main data items and entities, the list of Non-Functional Requirements, the Competitive analysis, and the High-Level System architecture and technologies used</li> </ul>
3	Roberto Simental	<ul style="list-style-type: none"> <li>○ Made a full draft for Executive Summary, later my draft and another teammate's draft were combined for final draft</li> <li>○ Designed the buddy pass system (unique premium feature that none of our competitors offer)</li> </ul>

		<ul style="list-style-type: none"> <li>○ Worked on Use Cases: buddy pass, private chat, friends</li> <li>○ Worked on Functional Requirements for: Guest users, Registered Users, and Searching People</li> <li>○ Worked on Non-Functional Requirements for: Marketing</li> <li>○ Worked on Competitive Analysis Charts for Competitors: Workout Buddies, Athlete Network</li> <li>○ Attended all team meetings, and have been actively participating in communication both vocal and through Discord</li> </ul>
4	Michael Satumba	<ul style="list-style-type: none"> <li>○ I contributed in writing the executive summary, the making of some of the use cases, the listing some of the items and entities, functional requirements, and competitive analysis</li> </ul>
5	Ziming Wang	<ul style="list-style-type: none"> <li>○ I am in charge of part 5, non-functional requirements. We had meetings everyday since last week and pretty much finished every part of the new version together.</li> <li>○ In the new version working, I was mainly writing the three parts of non-functional requirements: Portability and compatibility, System Requirements, and Storage. And I pretty much listed all other subclasses of non-functional requirements</li> <li>○ Like I said we pretty much write every part of the new version together. So I also finished one user cases (Alex) and a Competitive analysis(Athlete Network).</li> <li>○ I attended daily meetings on time and came up with a possible project(Improved Doordash) we can work on. Sadly our team decided not to choose it. But I would follow the choice of my team.</li> <li>○ I also actively answered others' questions on our</li> </ul>

		<p>discord. (Mostly communicating about the M1 writing)</p> <ul style="list-style-type: none"><li>○ And I set up React for the first time when my front-end leader told us about our tech stack.</li></ul>
6	Eduardo Hernandez	<ul style="list-style-type: none"><li>○ I contributed in the “Main Use Cases” creating the Friends graph only</li><li>○ List of main data items and entities, discussed which entities we were going to use</li><li>○ Competitive analysis, helped in filling out most of the social media section</li></ul>
7	Zhinan Zhao	<ul style="list-style-type: none"><li>○ I posted the first draft about functional requirements for my team to provide ideas. I also helped working on the main user case and competitive analysis Runsocial part.</li></ul>

**MILESTONE2****SW Engineering CSC648/848 Summer 2021****FitHub**

**Project Application And Name:** Find a fitness partner - “FitHub”

**Team Info:** Team 07 | Error 404

**Team Members:**

1. Vidhi Vora (Team Lead and GitHub Master) - vvora@sfsu.edu
2. Roberto Simental (Front-End Lead) - rsimental@mail.sfsu.edu
3. Johnson Nguyen (Back-End Lead) - Jnguyen63@mail.sfsu.edu
4. Zhinan Zhao
5. Eduardo Hernandez
6. Ziming Wang
7. Michael Satumba

**Milestone:** Milestone 2

**Date:** 08 July 2021

**History:**

<b>Version</b>	<b>Date</b>
M2V2	22 July 2021
M2V1	08 July 2021
M1V2	13 July 2021
M1V1	22 June 2021

## Table Of Contents

1. Data Definitions	3
2. Prioritized Functional Requirements	7
3. UI Mockups and Storyboards (high level only)	12
4. High level database architecture and organization	23
5. High Level APIs and Main Algorithms	32
6. High Level UML Diagrams	34
7. High Level Application Network and Deployment Diagrams	35
8. Identify actual key risks for your project at this time	37
9. Project management	39
10. Detailed list of contributions (this section must be done by the team lead)	40

## 1. Data Definitions

1. Guest user: A user who hasn't registered or provided their information to FitHub

1.1 Registration: Ability to register for an account

1.2 User name: Necessary to have username

1.3 Email: Necessary to have email

1.4 Password: Necessary to have password

1.5 Accept terms of use: Necessary to accept terms of use

Description: guest user will have the following attributes,

- Guest id
- Email id
- Is register

2. Registered user: A user who has successfully created their account with FitHub and can access all the app features

2.1 Login: Necessary to have username and Password

2.2 User interests: This information will be used by FitHub to show recommendations of people with similar interest

2.3 User Profile: All the user-related information that can be used to create its profile

2.4 User logs: Previous workout partners or events that the user has been to can be looked at in the user logs

2.5 Activities: All the indoor and outdoor activities that Fithub allows a user to search or select as interest.

Description: register user will have the following attributes,

- Reg id
- User id
- Phone
- Address
- Zip code
- Activity type

3. Gym membership: A user's gym membership information. This information will be used in order to send buddy pass request by other users

3.1 Buddy Pass Owner: A user having gym membership and willing to share his gym pass with a friend so that they can workout together

Description: gym membership will have the following attributes,

- Gym id
- Reg id
- Location
- Zip code
- Membership start date
- Membership expiry date

4. Events:

4.1 A user shall post events

4.2 Events shall be viewed by other users

4.3 Users can join the event

Description: event will have the following attributes<

- Event id
- Reg id
- Start time
- End time
- date

5. Friends:

5.1 Users that match based on interest

5.2 Users that match based on location

5.3 Users can send friend request

5.4 Users can accept friend request

Descriptions: friends will have the following attributes,

- User id
- Reg id
- Name
- Contact info

- Friend request
- send/accept request

## 6. Friend List:

### 6.1 User shall have a friends list

Description: A friends list will have the following attributes,

- reg\_id: all user's will have a register id
- Name
- Contact info

## 7. Private chat:

### 7.1 Users shall be friends

### 7.2 Users can send private messages from friends list

Description: private chat will have the following attributes,

- User id
- Reg id
- Name
- Date
- Time

## 8. Group chat:

### 8.1 Users shall be friends

### 8.2 Users can have group conversation

### 8.3 Users can invite other friends

Description: group chat will have the following attributes,

- User id
- Reg id
- Name
- Date
- Time

## 9. Clubs:

- 9.1 Users can create clubs
- 9.2 Users can join/send request
- 9.3 clubs can be public/private

Description: clubs will have the following attributes,

- Club id: will be used to identify the club
- Reg id: will have all the register users id
- Open to all: will be let all user know that the club will be open to every registered user

## 10. Club owner:

- 10.1 Clubs can only have one owner

Description: club owner will have the following attributes<

- Club owner id
- Reg is
- User id
- Date started

## 11. Club List:

- 11.1 List of clubs users can view

- 11.2 List of clubs users can join

Description: club list will have the following the following attributes,

- User id
- Reg id
- name

## **2. Prioritized Functional Requirements**

### **P1 (Mandatory)**

#### **Guest User:**

1. Guest users shall be able to view events posted by other users.
2. Guest users shall be able to select their interests.
3. Guest users shall see other users' user names of people looking to work out nearby.
4. Guest users shall be able to register.
5. Guest users shall be able to Log in as registered users.
6. Guest users shall be able to access the homepage.
7. Guest users shall be able to access the About us page.
8. Guest users shall be able to access the Contact us page.
9. Guest users shall be able to access the Support page
10. Guest users shall be able to delete their account from FitHub

#### **Registered User:**

11. Registered users shall be able to access homepage
12. Registered users shall be able to access About us page
13. Registered users shall be able to access Contact us page
14. Registered users shall be able to access Support Page
15. Registered users shall be able to send workout invites to other users.
16. Registered users shall be able to select their interests
17. Registered users shall be able to edit their information
18. Registered users shall be able to update their profile picture
19. Registered users shall be able to delete their profile picture
20. Registered users shall be able to deactivate their account
22. Registered users shall be able to change their account passwords
26. Registered users shall be able to delete their account from Fithub

#### **Searching People**

27. Registered users shall be able to have a certain number of swipes per day.
28. Registered users shall be able search for buddies with similar interest

29. Registered users shall be able search for buddies nearby
30. Registered users shall be able to filter search options based on their interests.
31. Registered users shall be able to filter search options based on how far they are willing to travel.

## **Friends**

33. Registered users shall be able to find friends to exercise with.
34. Registered users shall be able to add other registered users as friends.
35. Registered users shall be able to access their friend list
36. Registered users shall DM (direct message) other registered users only if they are friends.
37. Registered users shall have multiple friends if they choose to.
38. Registered users shall have 0 friends if they choose to.
39. Registered users shall have the ability to unfriend a former friend.
40. Registered users shall have the ability to block a former friend.
41. Registered users shall have the ability to report a former friend.
43. Registered users shall view friend's event postings.
44. Registered users shall be able to decline friend requests.
45. Registered users shall be able to accept friend requests.

## **Events**

46. Registered users shall be able to create events
47. Registered users shall be able to delete events they created
48. Registered users shall be able to edit the events they created
49. Registered users shall be able to join an event
50. Registered users shall be able to exit from an event
51. Registered users shall be able to rejoin an event
52. Registered users shall invite people to events they created

## **Chats**

70. Registered users shall be able to create a private chat
78. Registered users shall be able to reject to join in a private chat
80. Registered users shall be able to accept to join in a private chat

**Web Application**

89. Web Application shall have About us Page
90. Web Application shall have Contact us page
91. Web Application shall ask user to log in
92. Web Application shall display user's profile
93. Web Application shall show notifications to the user
94. Web Application shall show friend requests
95. Web Application shall allow user to check their messages
96. Web Application shall allow user to check event dates they are planning to go
98. Web Application shall allow user to change password of their account
99. Web Application shall allow user to deactivate their account
101. Web Application shall show the logs of user activities
102. Web application shall show events occurring nearby
103. Web Application shall show people recommendation with similar interest
104. Web Application shall allow user to log out
105. Web Application shall allow user to delete the account
106. Web Application shall allow user to see list of friends
108. Web Application shall allow user to search for people using various filter options
109. Web Application shall show people recommendation with similar interest

**P2 (Desired)****Registered User:**

24. Registered users shall be able to send an introductory message to a different user in their search result.
25. Registered users shall be able to share their real time location with FitHub

**Buddy Pass**

82. Registered users shall be able to view people with gym membership
83. Registered users shall be able to put request to access buddy pass
84. Registered users shall be able to accept the request to access buddy pass
85. Registered users shall be able to reject the request to buddy pass

Web Application shall show buddy pass requests

### **History logs**

86. Registered users shall access their event visits in logs
87. Registered users shall access their workout pairing information in logs
88. Registered users shall be able to access their workout partner's information from logs

### **Registered User:**

21. Registered users shall be able to save their frequent searches

### **Searching People**

32. Registered users shall be able to remove people from their search results

### **P3 (Opportunistic)**

#### **Friends**

42. Registered users shall view friend's club postings.

#### **Clubs**

53. Registered users shall be able to create the club
54. Registered users shall be able to update the picture of the club they created
55. Registered users shall be able to edit the club name they created
56. Registered users shall be able to delete the club they created
57. Registered users shall be able to delete the club picture they created
58. Registered users shall be able to send request to join the club
59. Registered users shall be able to leave a club
60. Registered users shall be able to send request rejoin a club
61. Registered users shall be able to add members to the club they created
62. Registered users shall be able to remove members from the club they created
63. Registered users shall be able to post in club
64. Registered users shall be member of multiple clubs
65. Registered users shall be admin of multiple clubs
66. Registered users shall be able to invite people to club
67. Registered users shall be able to accept the joining request for club they created

68. Registered users shall be able to reject the joining request for club they created

### **Chats**

69. Registered users shall be able to create a group chat

71. Registered users shall be able to invite friends to the group chat

72. Registered users shall be able to remove people from group chat

73. Registered users shall be able to add unknown people from the chat

74. Registered users shall be able to post in the group chat

75. Registered users shall be able to leave a group chat

76. Registered users shall be able to dissolution the group chat they cheated

77. Registered users shall be able to recall their posts

79. Registered users shall be able to reject to join in a group chat

81. Registered users shall be able to accept to join in a group chat

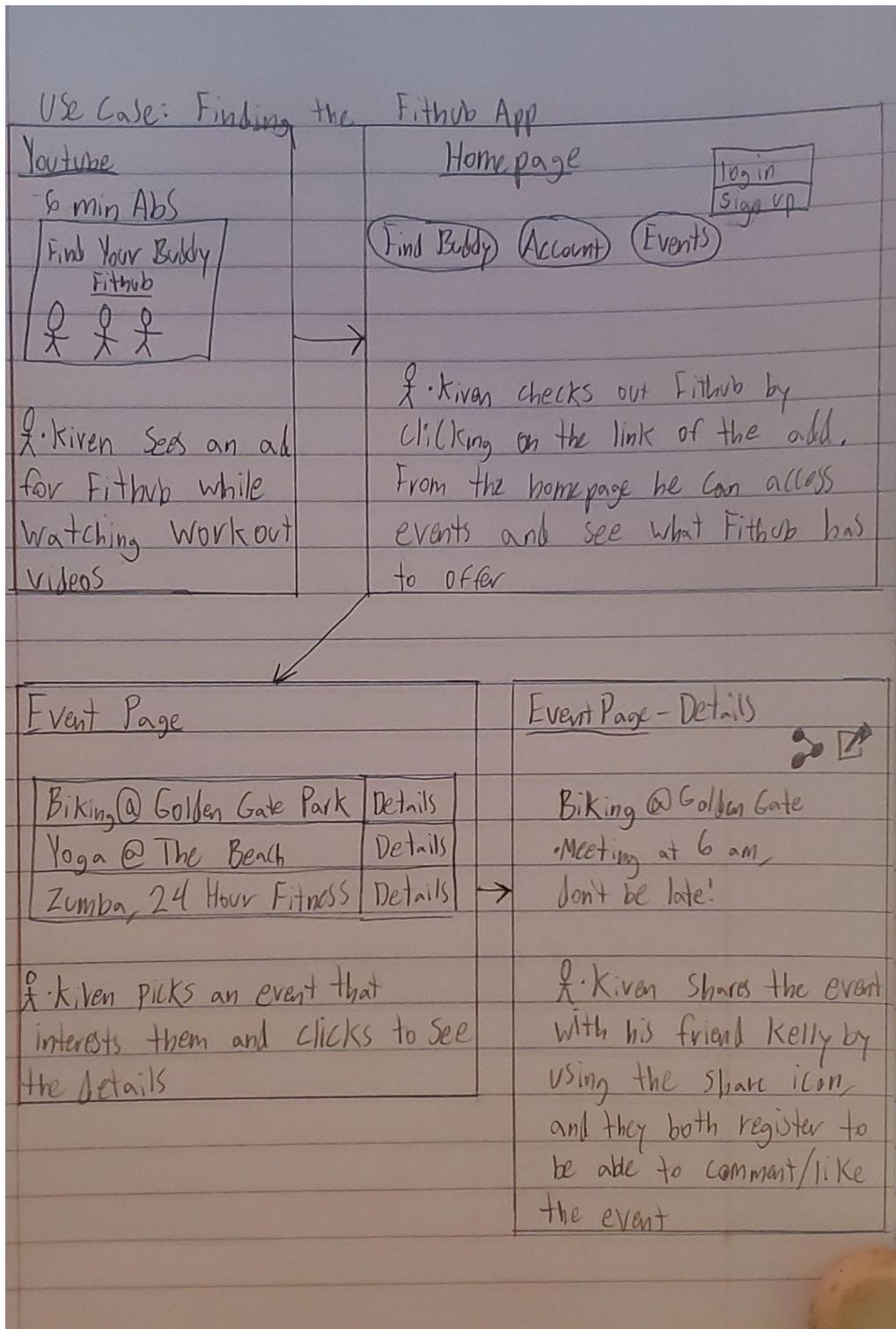
### **Web Application**

97. Web Application shall allow user to check updates of the club

107. Web Application shall allow user to see list of clubs

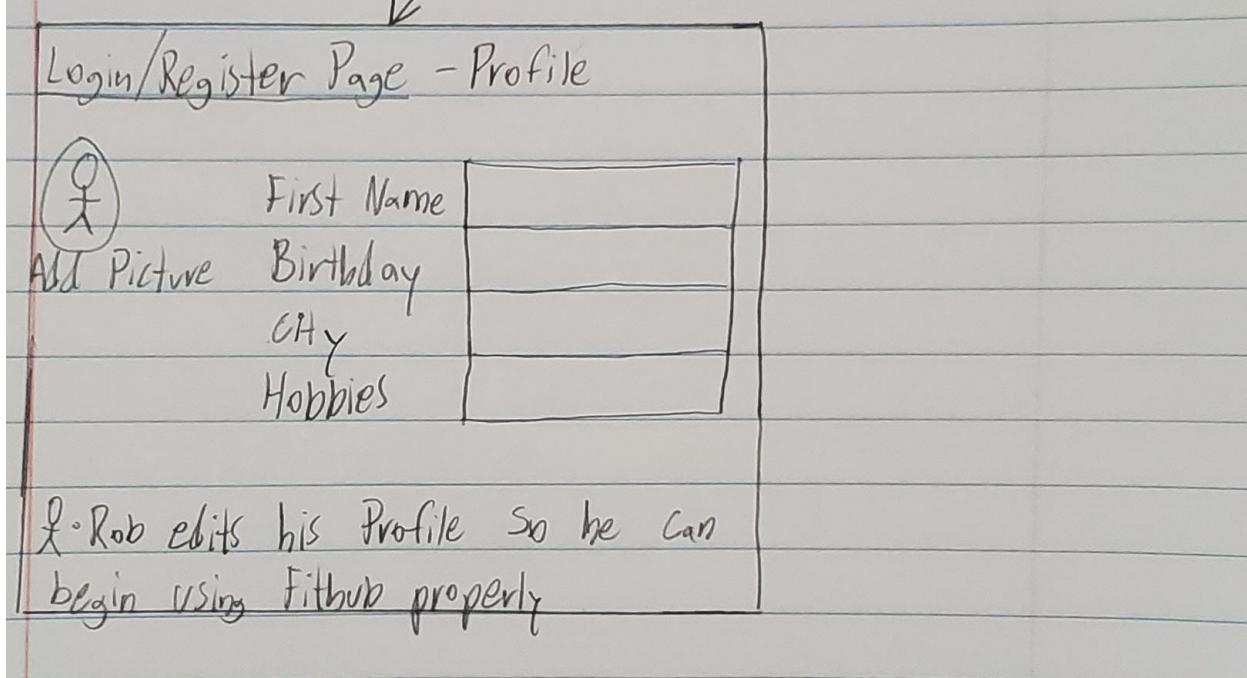
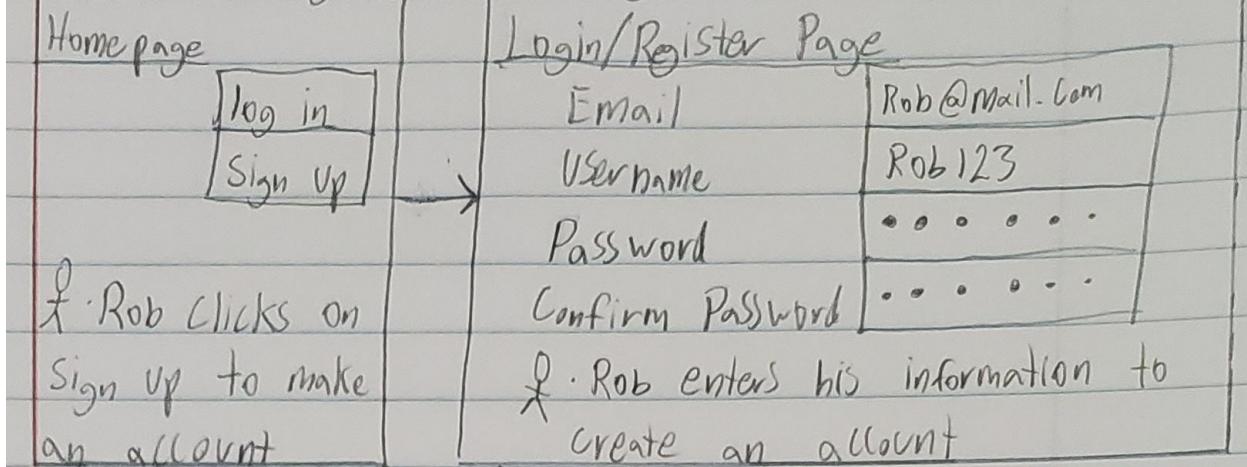
### 3. UI Mockups and Storyboards (high level only)

#### Use Case 1: Finding the Fithub Website

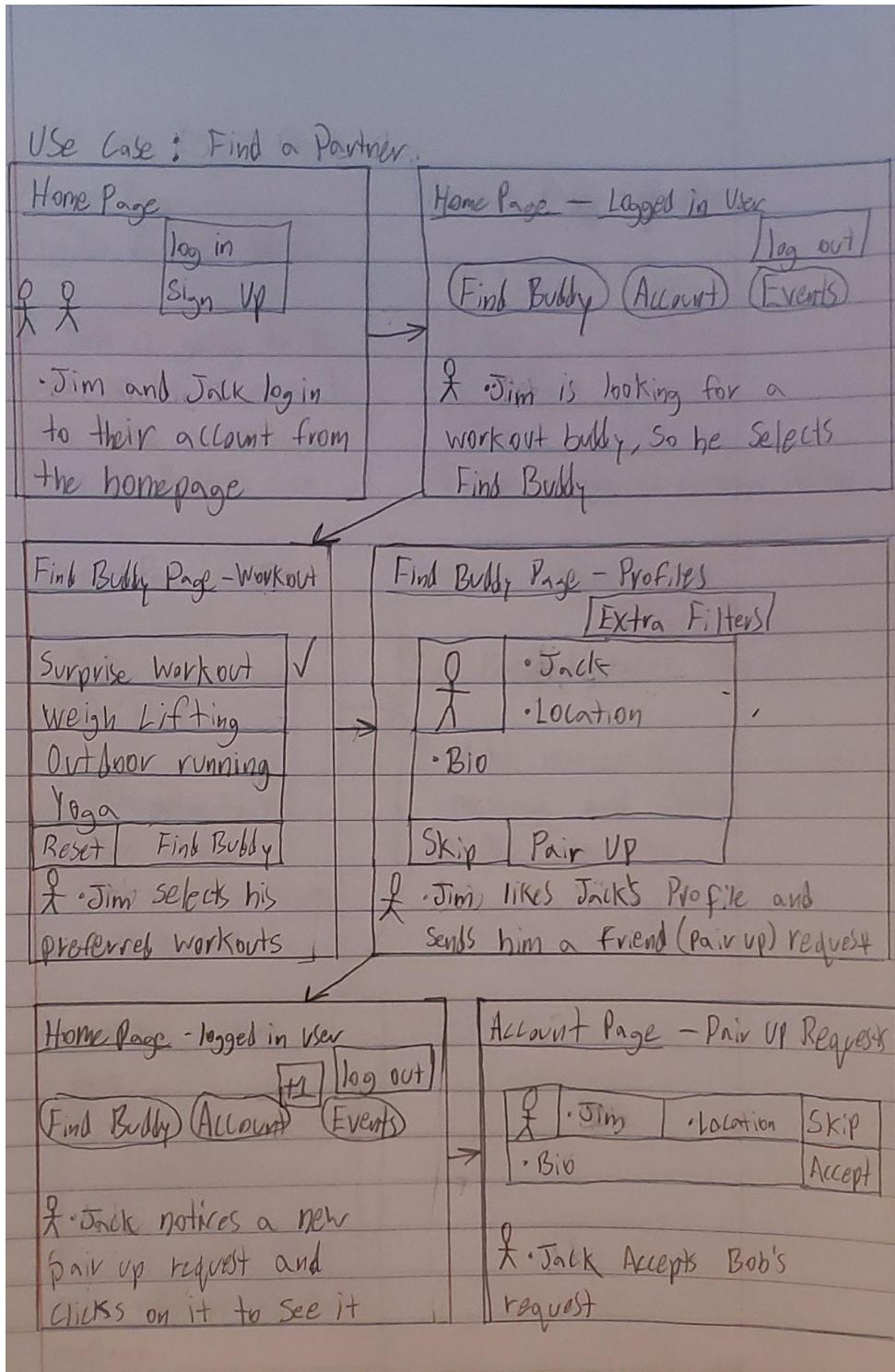


#### Use Case 2: Registration and Login

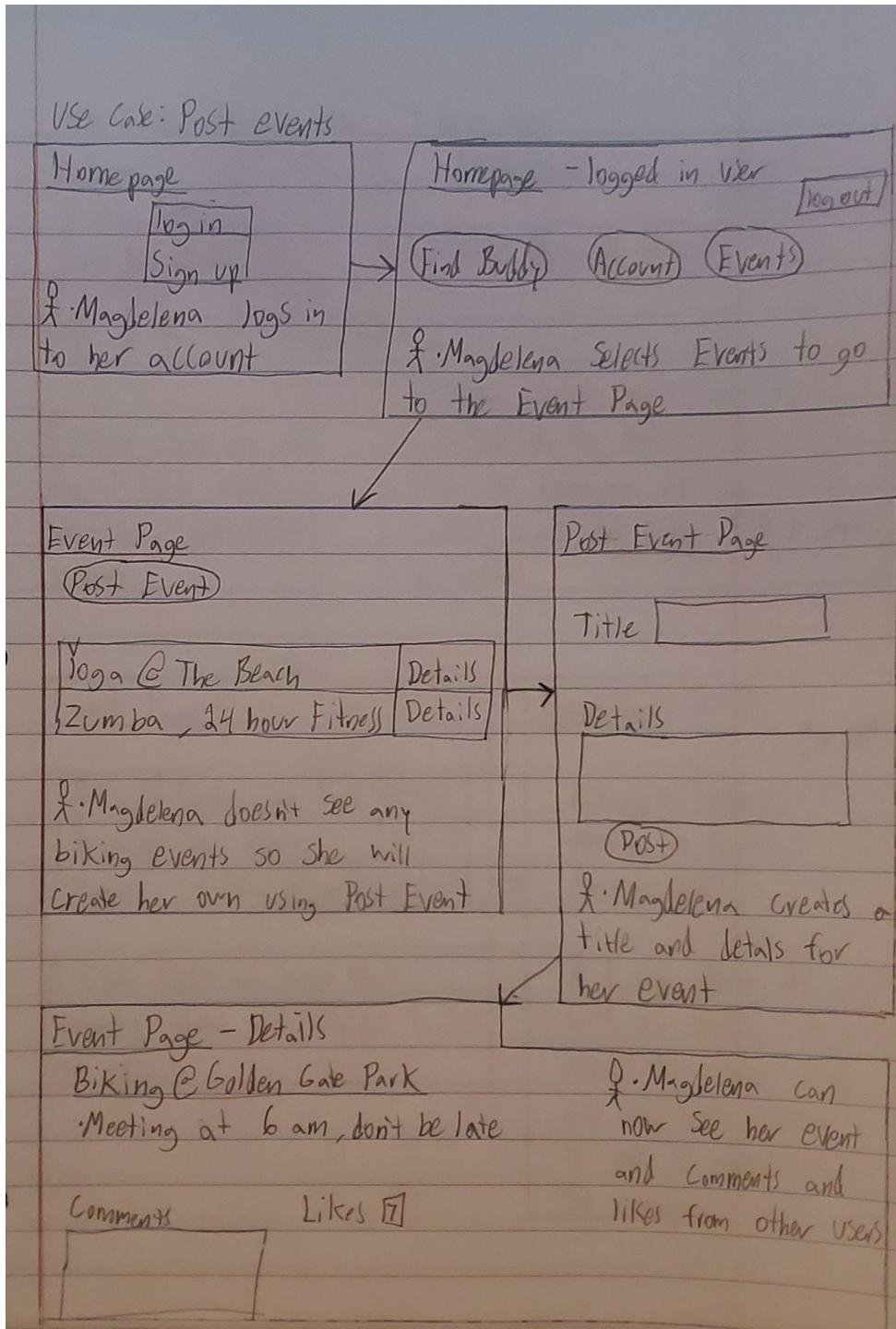
## Use Case: Registration and Login



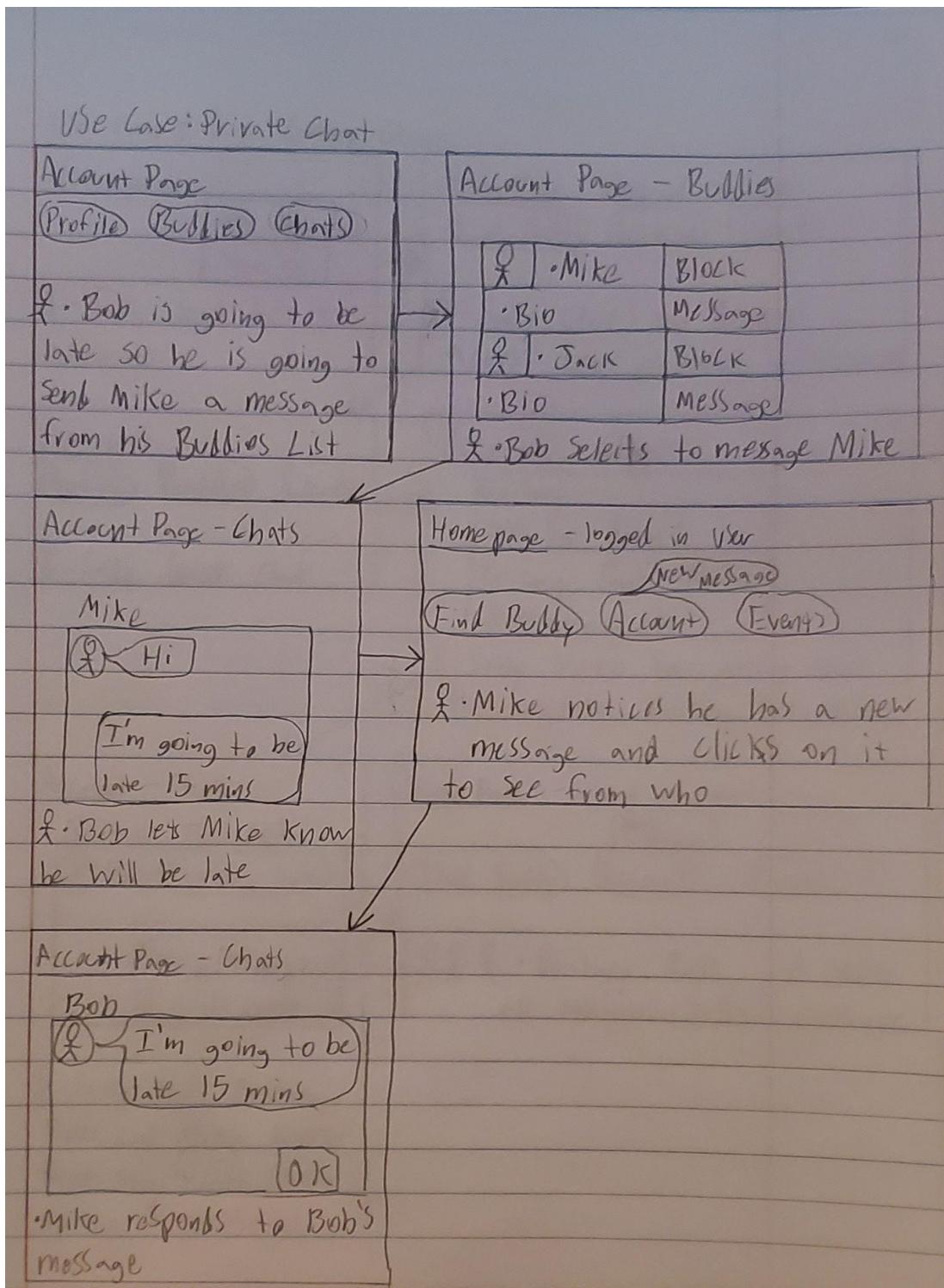
### Use Case 3: Finding a Partner



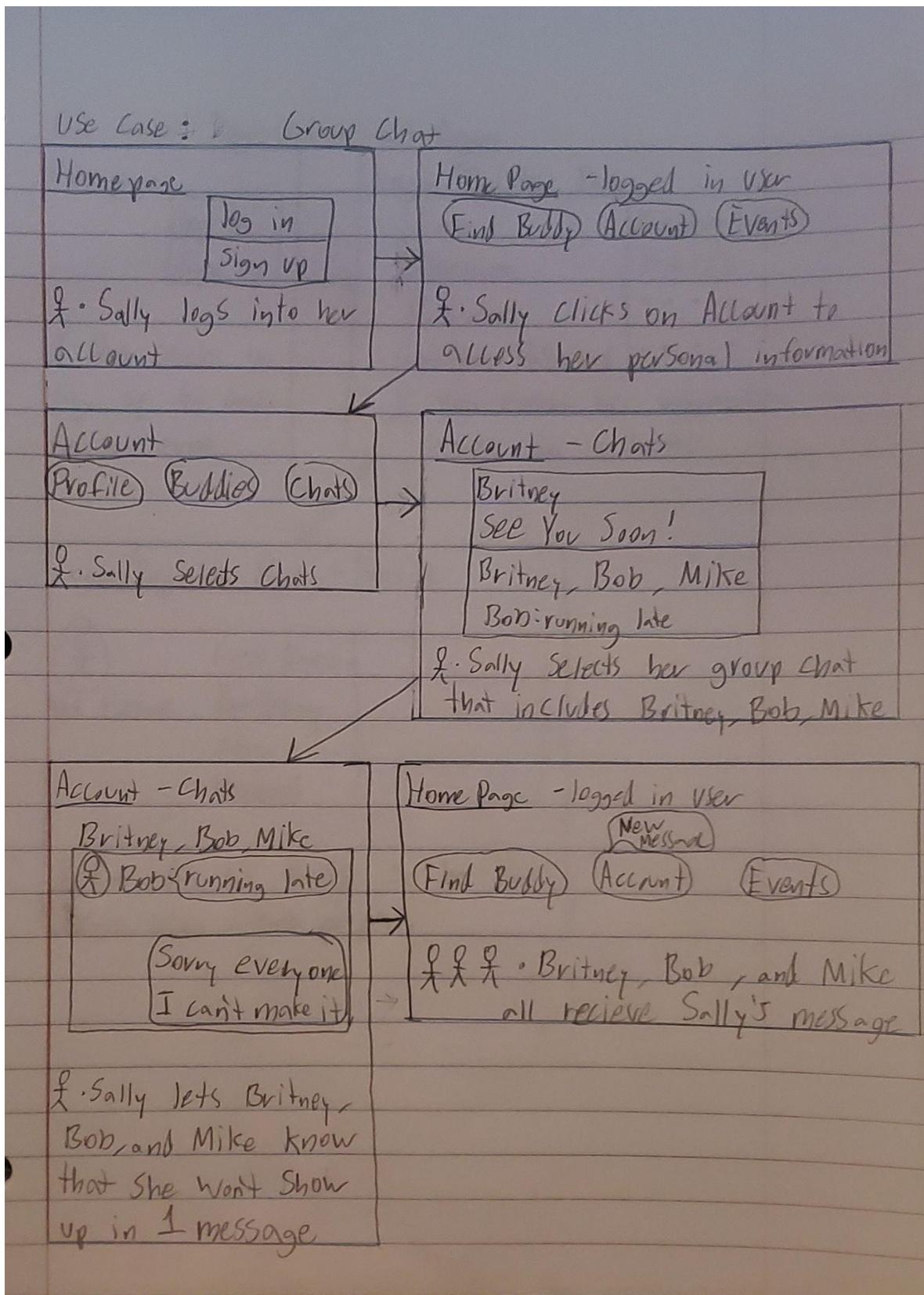
### Use Case 4: Posting Events



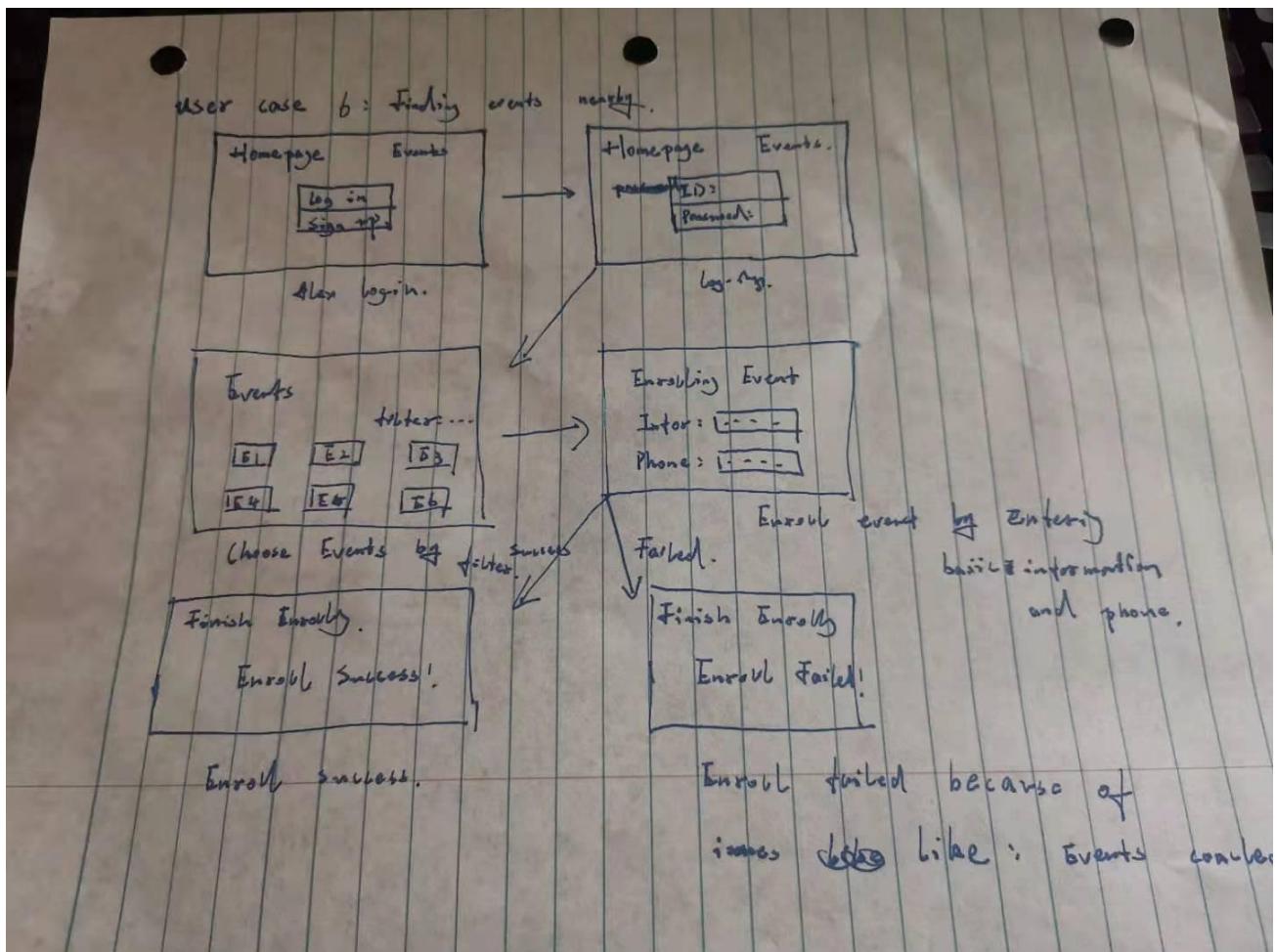
### Use Case 5: Private Chat



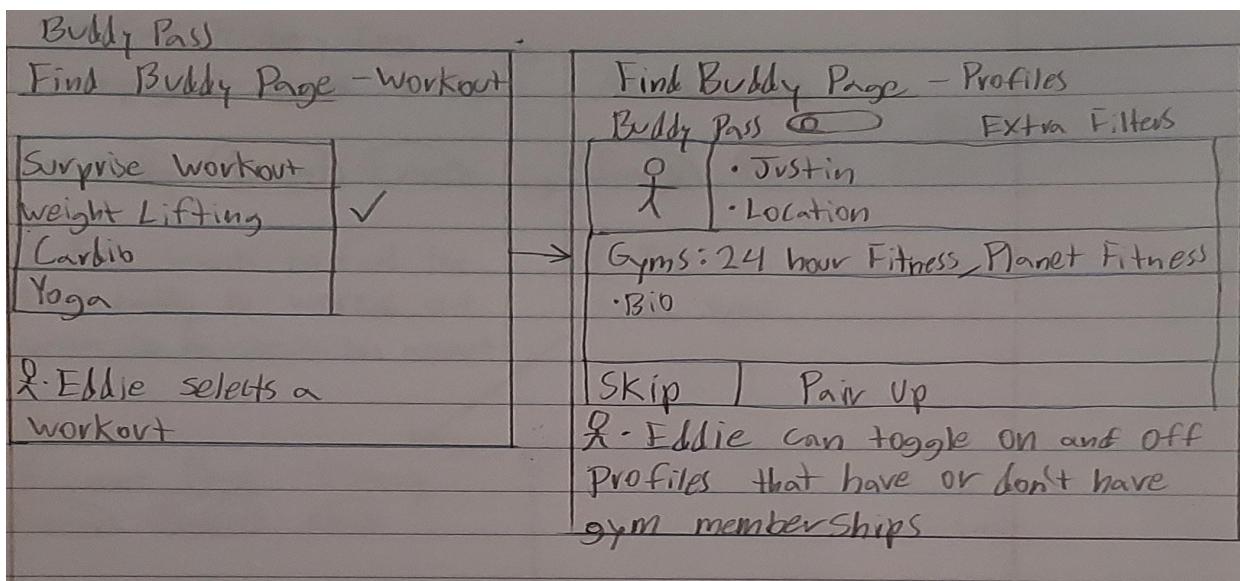
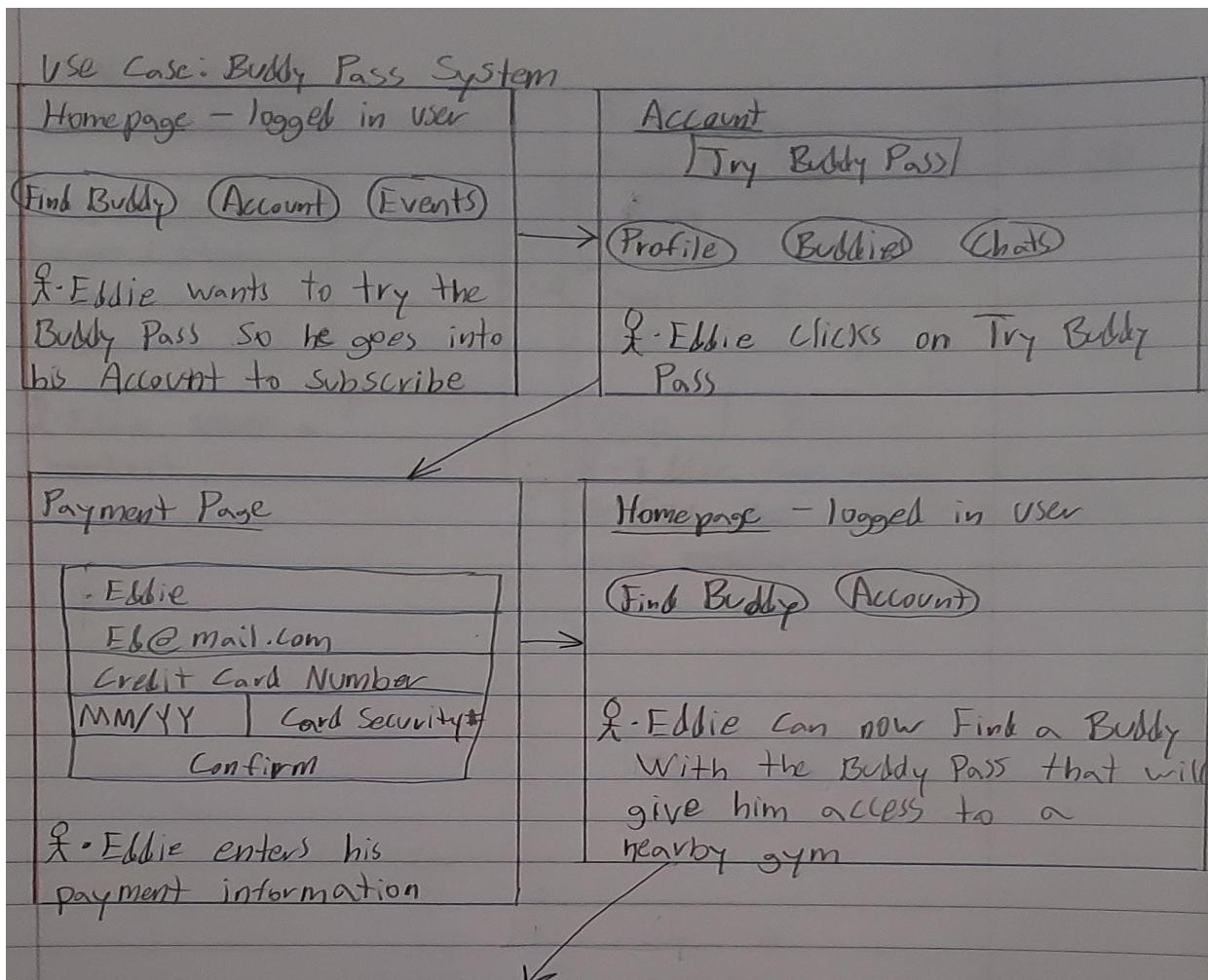
### Use Case 6: Group Chat



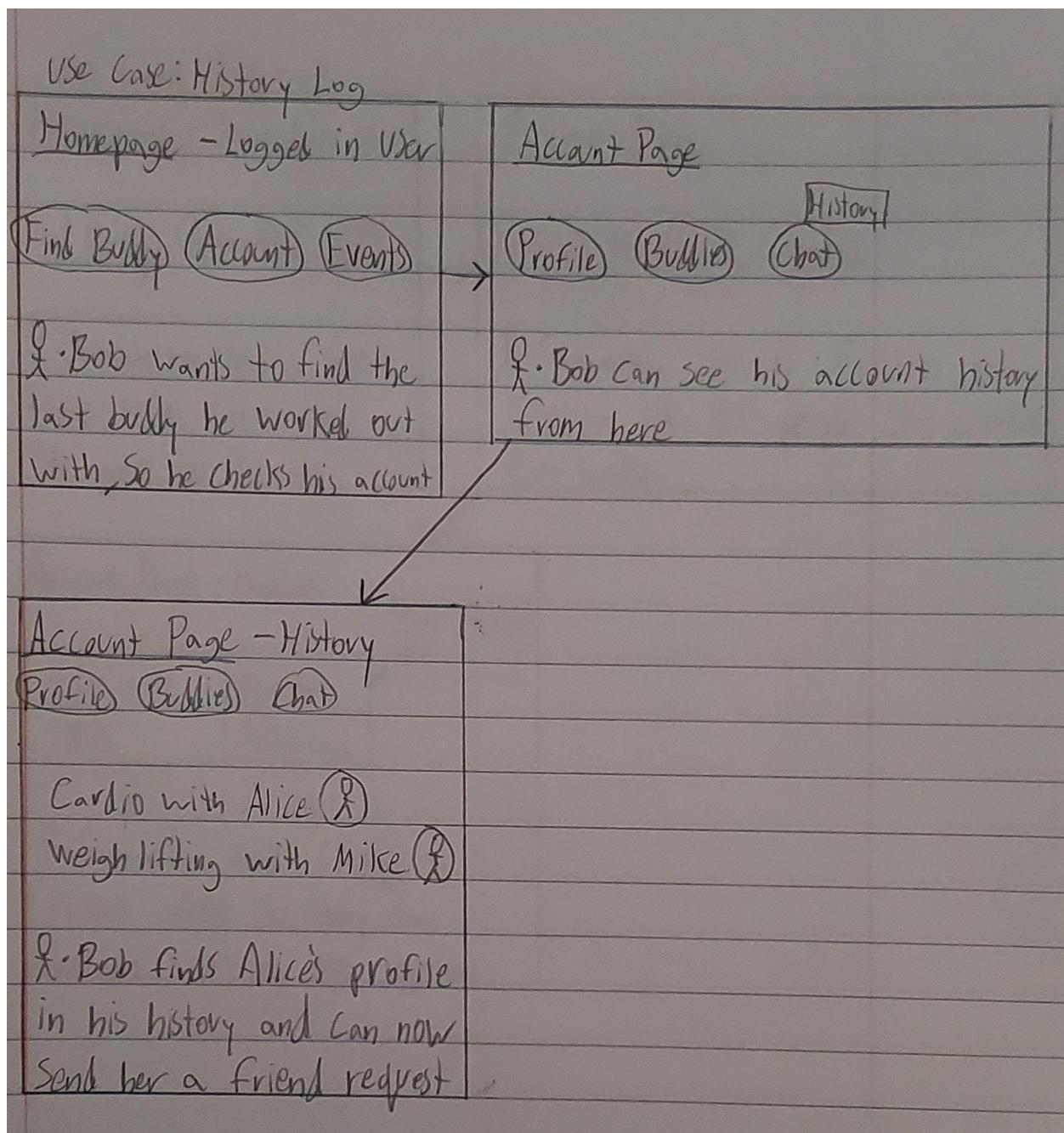
### Use Case 7: Finding events



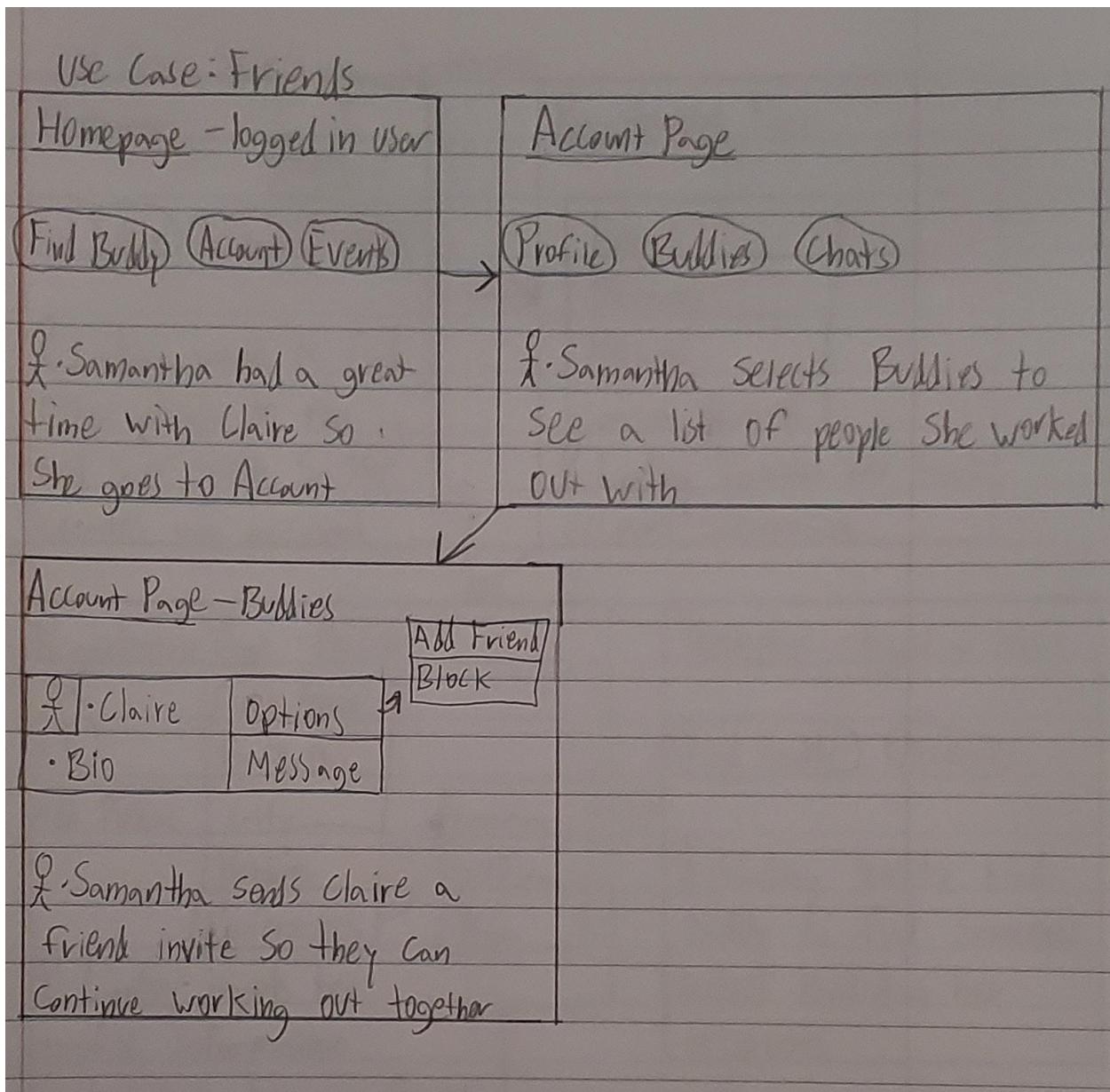
## Use Case 8: Buddy Pass System



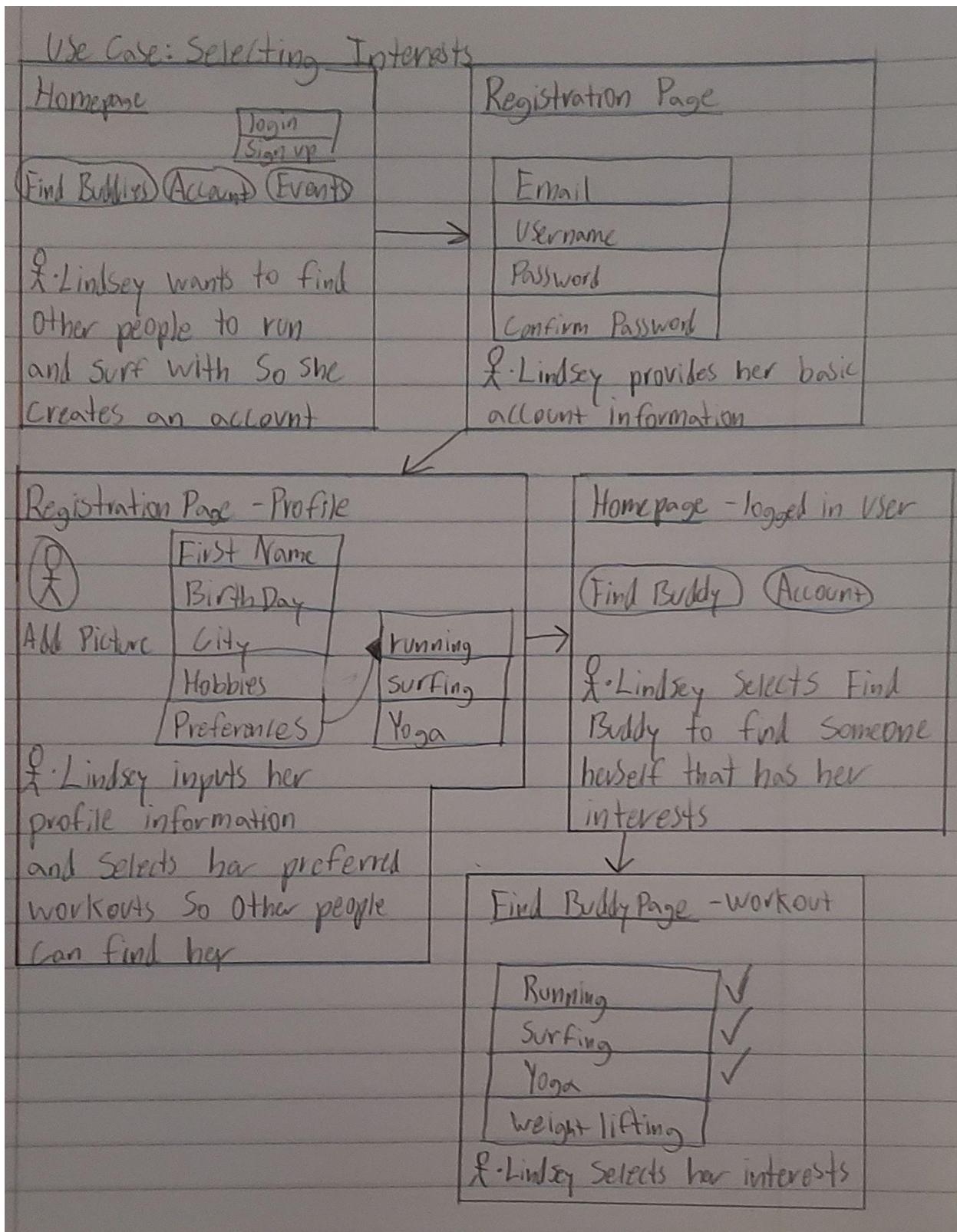
### Use Case 9: History Log



### Use Case 10: Friends



### Use Case 11: Selecting Interests



## **4. High level database architecture and organization**

- **DB organization:**

1. ***Business rules:***

- a. Guest User:

- A guest shall have one account
- A guest shall be at most one registered user
- A guest shall be able to search many Users
- A guest shall be able to view many events

- b. Registered User:

- A registered user shall have one account
- A registered user shall be able to search many Users
- A registered user shall be able to create and delete many events
- A registered user shall be able to send, cancel, accept and reject many workout requests
- A registered user shall have many gym memberships
- A registered user shall send, cancel, accept and reject many friend requests

- c. Account:

- An account shall be associated with a guest user or a registered user
- An account shall have all the information of a user
- An account shall be accessed by a single credentials combination only (username-password)

- d. Events:

- An event shall be created and deleted by a Registered User
- An event shall be viewed by many Users

- e. Clubs:

- An club shall be created and deleted by a Registered User
- An club shall be joined by many Registered Users

f. Gym Membership:

- › A gym membership shall belong to a registered user
- › A gym membership shall have option for atmost one buddy pass

g. History Logs:

- › A history log shall belong to a registered user
- › A history log shall have information related to user activities

h. Blocked Users:

- › A block list shall have many users blocked by other users
- › A block list shall be used by many registered users

i. Buddy pass:

- › A buddy pass belongs to a gym membership
- › A buddy pass shall be used by any registered user

j. Friend request

- › A friend request shall be sent by a registered user
- › A friend request shall be canceled by a registered user
- › A friend request shall be accepted by a registered user
- › A friend request shall be rejected by a registered user

k. Workout request

- › A workout request shall be sent by a registered user
- › A workout request shall be canceled by a registered user
- › A workout request shall be accepted by a registered user
- › A workout request shall be rejected by a registered user

**2. *Entities:***

a. Guest User (Strong)

- › user\_id: unique user id to identify the guest user
- › email\_id: unique email id associated with a user

- › is\_registered: if the user is a registered user
- b. Registered User (Weak)
- › reg\_id: unique user id to identify the registered user
  - › user\_id: id associated with a guest user
  - › phone: contact no of the registered user
  - › address: address of the registered user
  - › location: city name of the registered user
  - › zipcode: zipcode of the registered user
  - › activity\_type: like if the user is interested in indoor/outdoor activities
  - › workout\_type: kind of workout user prefers to carry out
- c. Account (Weak)
- › acc\_id: unique id for the account entity
  - › reg\_id: id associated with a registered user
  - › username: used to log in account
  - › password: password to log in account
- d. Events (Weak)
- › event\_id: unique id for the events entity
  - › reg\_id: id associated with registered user
  - › description: more info about the event
  - › start\_time: start time for the event
  - › end\_time: end time for the event
  - › from\_date: start date for the event
  - › to\_date: end date for the event
- e. Clubs (Weak)
- › club\_id: unique id for the clubs entity
  - › reg\_id: id associated with a registered user
  - › description: more information on the club
  - › open\_to\_all: if other registered users can join without invite

f. Gym Membership (Weak)

- › gym\_id: unique id for the gym membership info
- › reg\_id: id associated with a registered user
- › gym\_name: gym name the user has membership for
- › gym\_loc: location of the gym
- › gym\_zipcode: zipcode in which the gym is located
- › buddy\_pass: does it have the option for buddy pass
- › share\_pass: if the registered user is willing to share his buddy pass
- › membership\_start\_date: start date for gym membership
- › membership\_expiry\_date: end date for gym membership

g. History Logs (Weak)

- › log\_id: unique id to identify the log of a registered user
- › reg\_id: id associated with a registered user
- › activty\_info: information on the user activity
- › is\_workout\_request: if log is for workout request
- › is\_friend\_request: if log is for friend request
- › is\_buddy\_pass\_request: if log is for buddy pass request
- › is\_block\_user: if log is for blocking a user
- › is\_club\_request: if log is for a club request

h. Blocked Users (Weak)

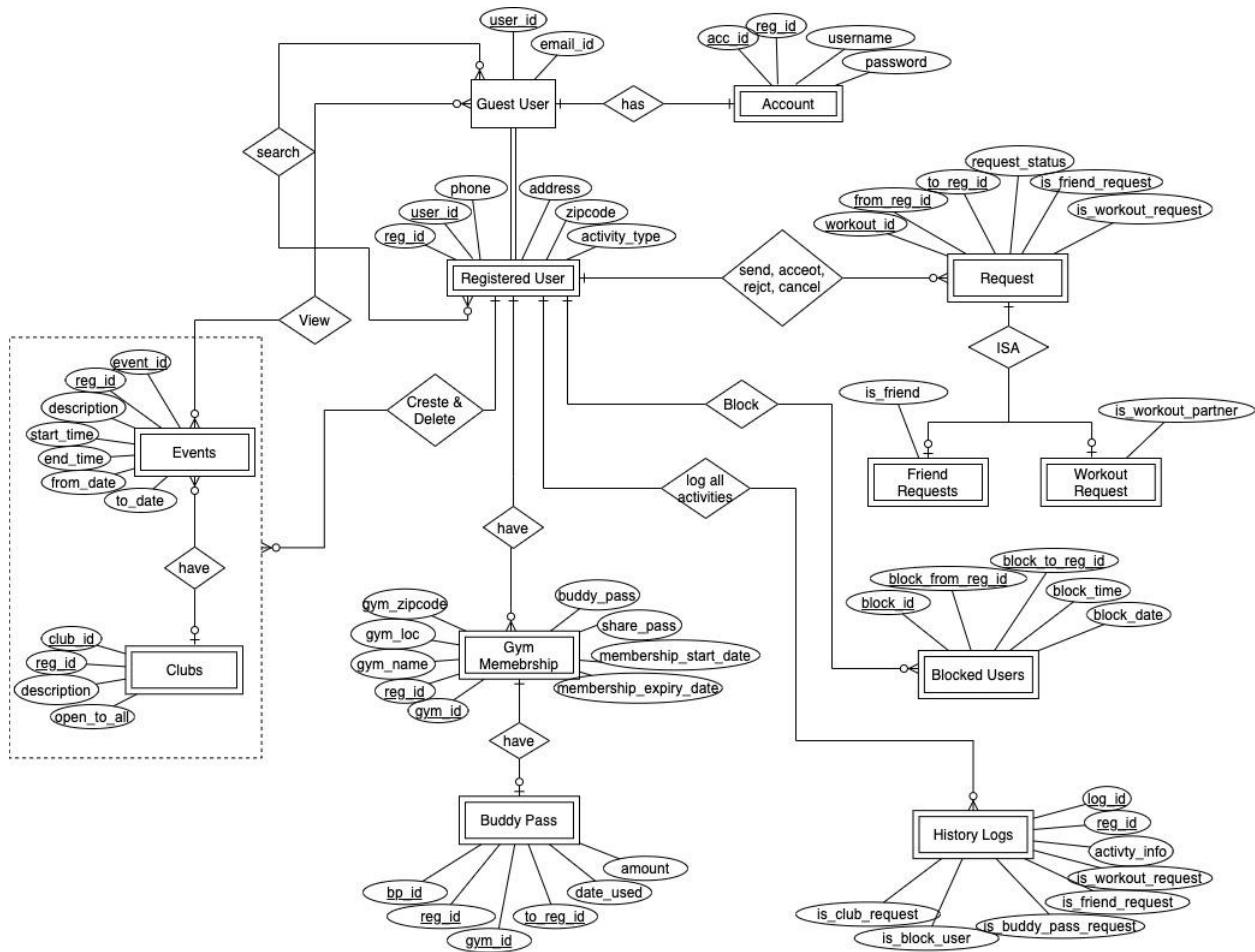
- › block\_id: unique id to for blocking entity
- › block\_from\_reg\_id: id associated with a registered user who is blocking
- › block\_to\_reg\_id: id of the registered user who is being blocked
- › block\_date: date the user was blocked
- › block\_time: time user was blocked

i. Buddy pass (Weak)

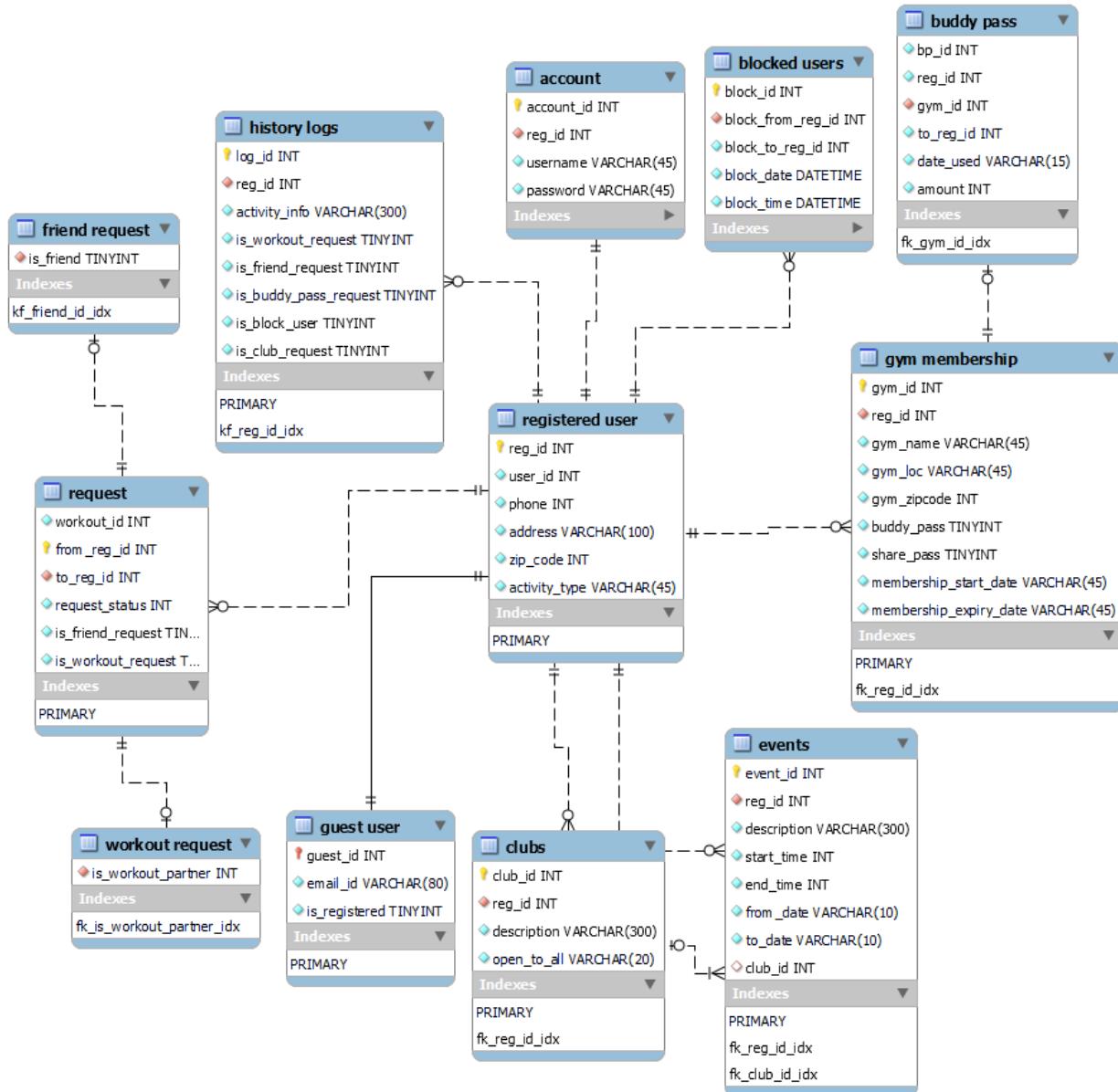
- › bp\_id: unique id for the buddy pass entity
- › reg\_id: id associated with a registered user
- › gym\_id: buddy pass belong to which gym
- › to\_reg\_id: registered user id with whom the buddy pass is being shared

- › date\_used: the date on which the user is sharing buddy pass
  - › amount: amount user collects for the buddy pass mechanism
- j. Friend request (Weak)
- › friend\_id: unique id for the friend entity
  - › from\_reg\_id: id of a registered user who is sending friend request
  - › to\_reg\_id: id of a registered user to whom the request is sent
  - › request\_status: status of the request as sent, cancel, accept, reject
- k. Workout request (Weak)
- › workout\_id: unique id for the workout entity
  - › from\_reg\_id: id of a registered user who is sending workout request
  - › to\_reg\_id: id of a registered user to whom the request is sent
  - › request\_status: status of the request as sent, cancel, accept, reject

### 3. ERD:



#### 4. Database Model:



## 5. DBMS:

The database chosen to develop the project is MySQL since it is well known RDBMS, easy to use and many GUI tools are available for the development and maintenance of the MySQL database.

- **Media storage:**

- The media used in the FltHub app will be images of the profile pictures and club pictures.
- They will be stored in the **file system** and the format of image will be JPEG/ JPG/ PNG.

- **Search/filter architecture and implementation:**

- Search Algorithm:

- The input to the search algorithm shall be provided from the search bar.
- The user will input only usernames to search from the search bar
- The input will be looked into the database
- The output for the search algorithm will be the list of usernames that fully / partially matches the user input in the ascending order

The DB fields that will be searched here is :

- Table: Account
  - username

- Filter Algorithm:

- The input to the filter algorithm shall be provided by clicking on search button from the filters section
- The user will be given the option to search based on the location, indoor/outdoor activities, workout type by selecting the appropriate filters.
- These filtered inputs shall be looked into the database. These fields shall also be indexed in the database for quick search results.

- The output for the search algorithm will be the list of items that match the user's filter input. The information in each item fetched from the database will have the username, location, zip code, activity\_type, workout\_type.

The db terms that will be searched here is :

- Table: Account
  - username
- Table: Registered User
  - location
  - zipcode
  - activity\_type
  - workout\_type

## 5. High Level APIs and Main Algorithms

In our project we will develop our own API as we do not plan to use an external API. Our API will be stored in a file called API.js and it will feature Post requests and Get requests such as registering the user, logging in/logging out of the user, Posting events, and search by filter.

**Post request** tends to be utilized for storing data. And **Get request** tends to retrieve data from the web application.

- a. API.js File
  - i. Register (INTERNAL API)
    - 1. Post request:
      - a. When a user registers to our web application, we would receive a Post request with the user's username, email, along with their password. This will send the data to our database which will store all of the data. However, if the data already exists, it will give the user an error.
  - ii. Login (INTERNAL API)
    - 1. Post request:
      - a. When a user logs into our web application, we would receive a post request with the user's username and password. This will send the data to our database which will store all of the data and check if what the user entered exists. If it does exist the user will be logged into our website. If it does not exist, the user will get an error message.
  - iii. Logout (INTERNAL API)
    - 1. Post request
      - a. When a user logs out of our web application, we would receive a post request. With this post request, the API will log them out of our website which will make them have no more features.
  - iv. Posting Events (INTERNAL API)

1. Post request:

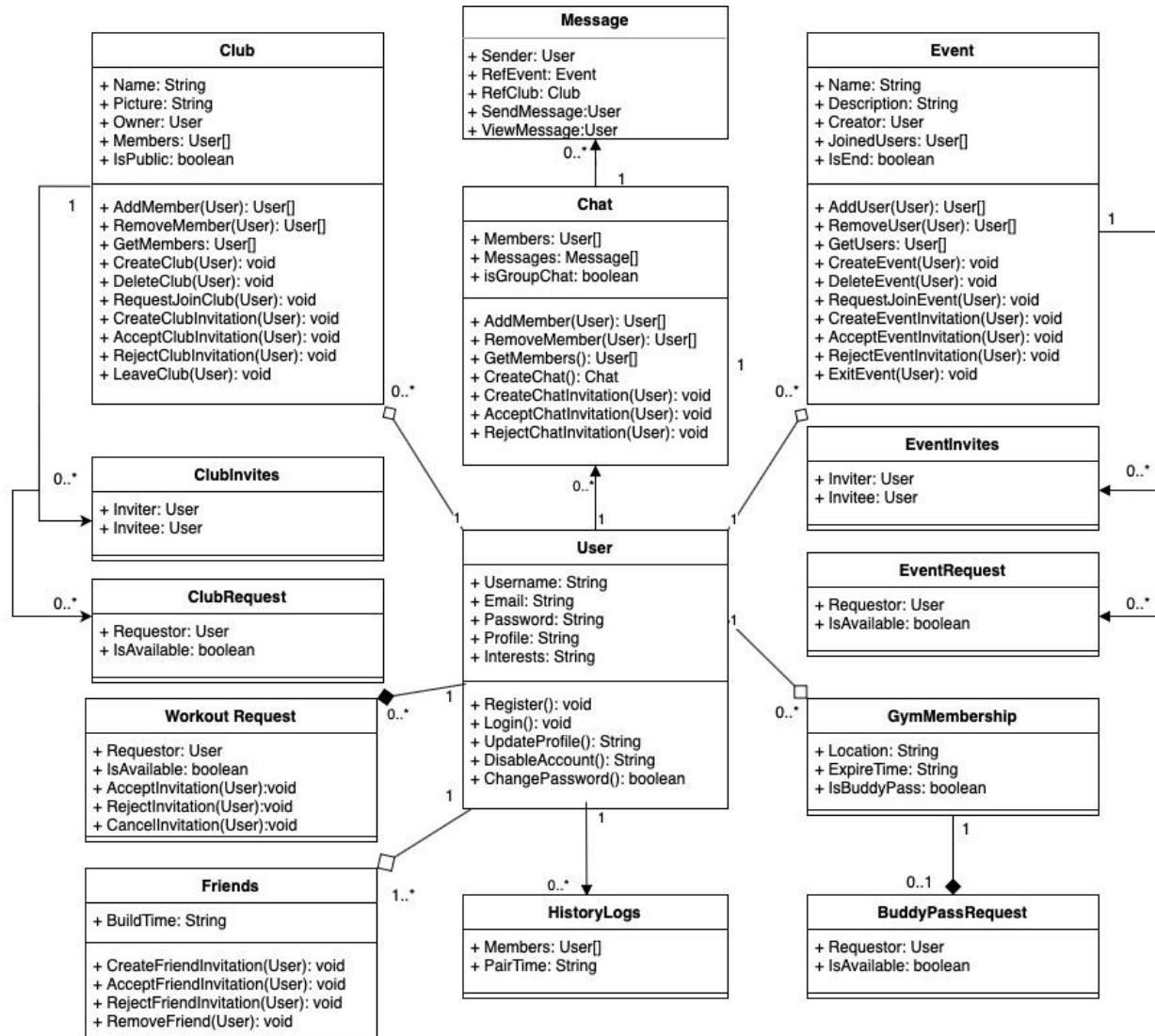
- a. When a user decides to post an event, the API will receive a post request. we would receive a post request with the user's information that they just inputted. This will send the data to our database which will store all of the data

v. Search by Filter (INTERNAL API)

1. Get request

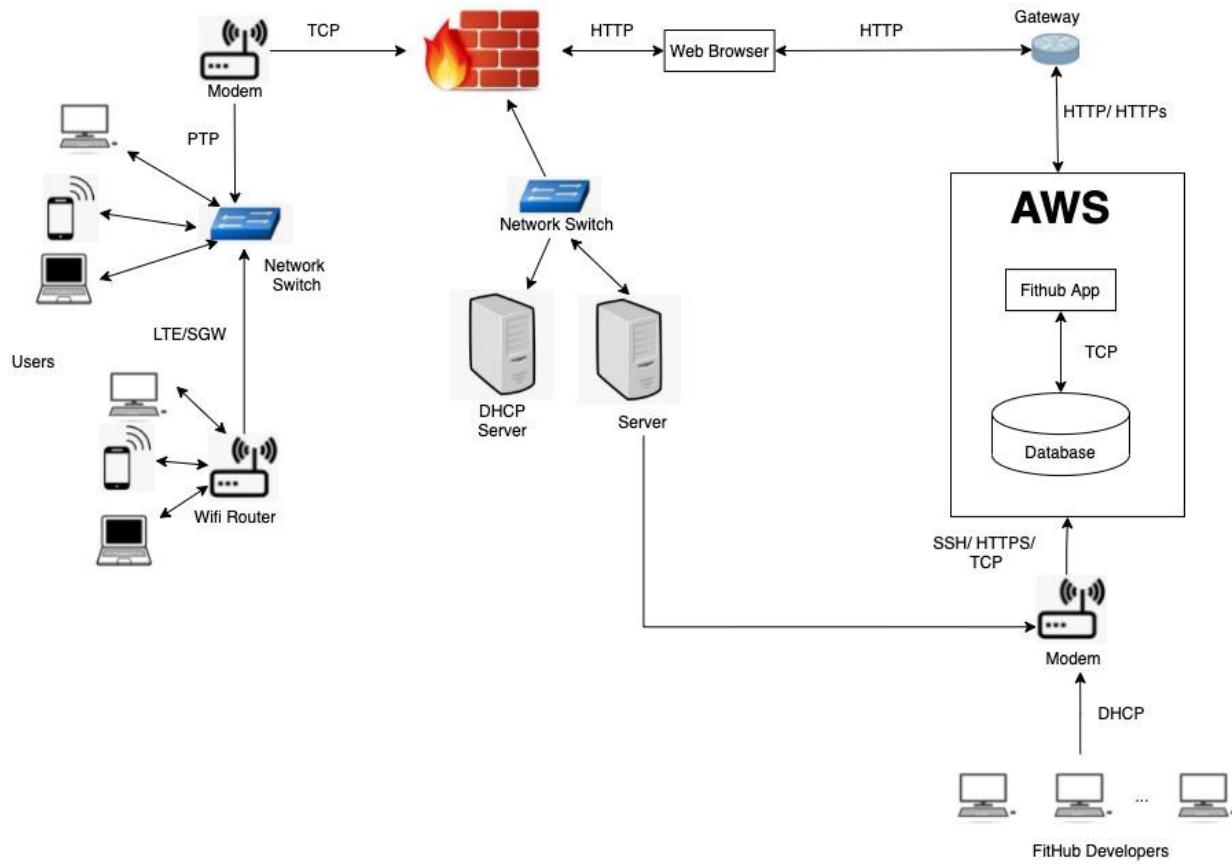
- a. When users search by filter, the backend will receive a Get request. This Get Request will then go into our database where every data is stored, and it will send back the necessary information. In example, a filter the user has selected.

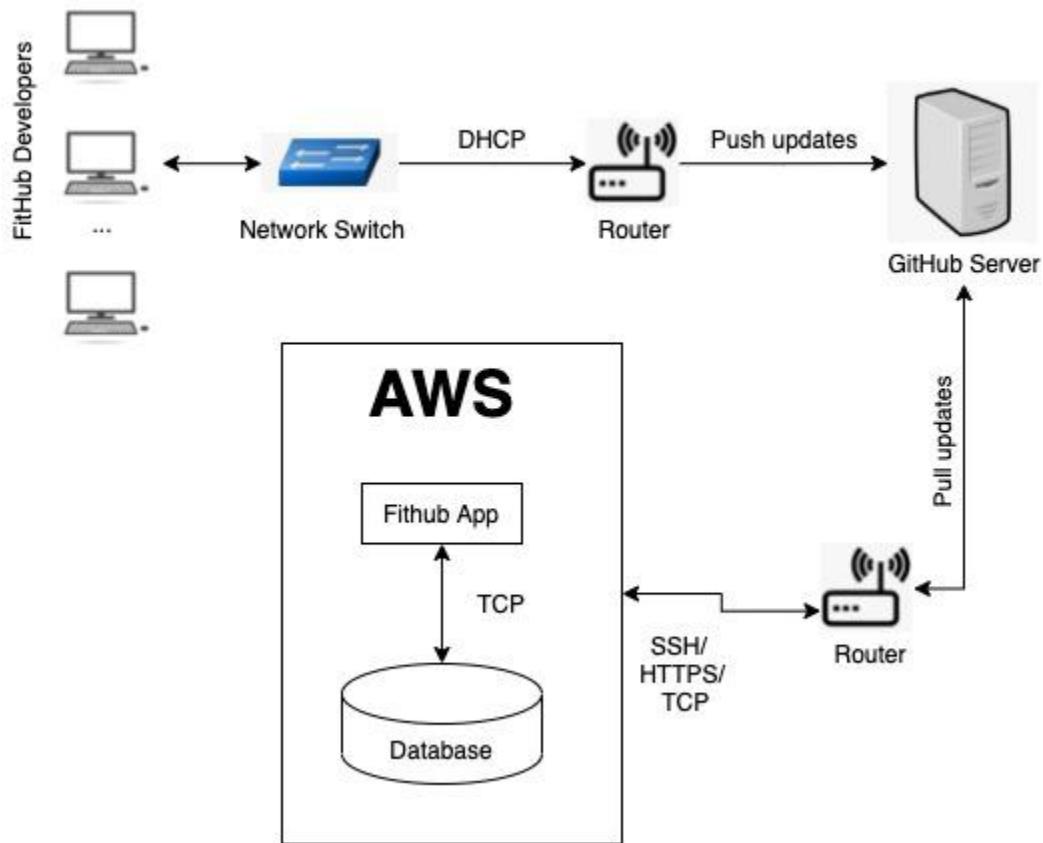
## 6. High Level UML Diagrams



## 7. High Level Application Network and Deployment Diagrams

### Application Network Diagram:



**Deployment Diagram:**

## 8. Identify actual key risks for your project at this time

### 1. No Legal/Content Risks

### 2. Skill risk: New Tech

We are using frameworks/technologies that are new to some members. Those members will have to learn the frameworks. If they fall behind they will have a harder time contributing and understanding the project.

→ Solution:

We have already decided what frameworks we will be using so that everyone has time to freshen up on them. We are also splitting work in a way so that more experienced members tackle the more technical tasks and we will have many meetings and discussions to make sure everyone is on track.

### 3. Schedule risk: Time constraints

We are working with tight deadlines and our goal as a team is to create an exceptional project. To achieve this goal we are meeting constantly throughout the week which has been difficult or stressful for some members.

→ Solution:

Our meetings now have a detailed schedule of topics that we need to cover so that we can get through them quickly. We will also be having less impromptu meetings and instead plan out meetings based on how we are doing, so that everyone has enough of a notice.

### 4. Teamwork risk: Team member contribution

Some members are difficult to contact through email and discord. We also have a problem with participation and attendance in team meetings.

→ Solution:

We will look for alternative methods of contacting team members like getting cell phone numbers in case they aren't responding back within an acceptable time frame. We also keep track of meeting attendance and work contributed so we can talk with any team members that are falling behind.

### 5. Technical risk: Map Integration

We are interested in having a map on the website that allows users to see the nearby gyms or the events that are happening. The simplest solution seems to be to use the Google maps API, but that will require learning it and possibly paying for access.

→ Solution:

We are learning more about how to integrate Google Maps and the various options and services they provide. We are also discussing alternatives such as linking to Google Maps search results or using a different service.

## **9. Project management**

- For every milestone received, the tasks are created and assigned to the team members
- The team meets every monday to plan and divide the weekly tasks
- There is an internal deadline set each of the tasks
- There are team meetings conducted on frequent basis in the week in order to track the ongoing progress, resolve any queries, discuss the concepts taught in class and how to implement them in our project
- Also, the team members are available on discord for any instant meeting, updates, announcements, queries and to ensure everyone is up to date
- The team uses Github inorder to manage the development of code for the project. There are two branches created 1. Master and 2. Develop. The users take the pull of the develop branch, write down their code and then push the changes to develop branch. These changes are merged with the master branch by the github master and then the server instance is restarted.
- For task management purposes, we use Asana. It helps us to keep track of who is doing a task, and what is the progress of it.
- For team meetings, we use zoom and discord.

## **10. Detailed list of contributions (this section must be done by the team lead)**

<b>Team Member</b>	<b>Contribution</b>
<p>Vidhi Vora <i>(Team Lead)</i></p>	<p><i>As a Team Lead:</i></p> <ul style="list-style-type: none"> <li>› Ensuring all the requirements are met and adhering to the M2 guidelines</li> <li>› Assigning and supervising the task progress, organizing regular team meetings and the agenda</li> <li>› Assigning internal deadlines, resolving queries, discussing the concepts for the M2</li> </ul> <p><i>As a Team Member (Milestone 2):</i></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Project management</li> <li>› Detailed list of contributions</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Prioritized Functional Requirements</li> <li>› High level database architecture and organization</li> <li>› High Level Application Network and Deployment Diagrams</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› Setting up the node express (api.js)</li> <li>› Connecting to the database</li> </ul>
<p>Johnson Nguyen <i>(Backend Lead)</i></p>	<p><i>Milestone 2:</i></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› High Level APIs and Main Algorithms</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› Developing and inserting data into the database for the registration page</li> </ul>
<p>Roberto Simental <i>(Frontend Lead)</i></p>	<p><i>Milestone 2:</i></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› UI Mockups and Storyboards</li> </ul>

	<ul style="list-style-type: none"> <li>› Identify actual key risks for your project at this</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› -</li> </ul>
Michael Satumba <i>(Frontend Member)</i>	<p><b>Milestone 2:</b></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Data definitions</li> <li>› UI Mockups and Storyboards</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› Developing the home page and the registration page</li> </ul>
Eduardo Hernandez <i>(Backend Member)</i>	<p><b>Milestone 2:</b></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Data definitions</li> <li>› High level database architecture and organization</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› Developing and inserting data into the database for the registration page</li> </ul>
Zhinan Zhao <i>(Backend Member)</i>	<p><b>Milestone 2:</b></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› High Level UML Diagrams</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› High Level Application Network and Deployment Diagrams</li> </ul> <p><i>Vertical Prototype:</i></p> <ul style="list-style-type: none"> <li>› -</li> </ul>
Ziming Wang <i>(Frontend Member)</i>	<p><b>Milestone 2:</b></p> <p>Partial Contribution to following sections:</p>

- |  |   |
|--|---|
|  | <ul style="list-style-type: none"><li>▸ Prioritized Functional Requirements</li></ul> |
|--|---|

*Vertical Prototype:*

- -

**MILESTONE3****SW Engineering CSC648/848 Summer 2021****FitHub**

**Project Application And Name:** Find a fitness partner - “FitHub”

**Team Info:** Team 07 | Error 404

**Team Members:**

1. Vidhi Vora (Team Lead and GitHub Master) - vvora@sfsu.edu
2. Roberto Simental (Front-End Lead) - rsimental@mail.sfsu.edu
3. Johnson Nguyen (Back-End Lead) - Jnguyen63@mail.sfsu.edu
4. Zhinan Zhao
5. Eduardo Hernandez
6. Ziming Wang
7. Michael Satumba

**Milestone:** Milestone 3

**Date:** 22 July 2021

**History:**

<b>Version</b>	<b>Date</b>
M3V2	30 July 2021
M3V1	22 July 2021
M2V2	22 July 2021
M2V1	08 July 2021
M1V2	13 July 2021
M1V1	22 June 2021

## Table Of Contents

1. Data Definitions V3	3
2. Functional Requirements V3	7
3. Wireframes Based on your Mockups/Storyboards V2	12
4. High level database architecture and organization V2	16
5. High Level Diagrams V2	25
6. List of Contributions in this milestone	28

## **1. Data Definitions V3**

1. Guest user: A user who hasn't registered or provided their information to FitHub

1.1 Registration: Ability to register for an account

1.2 User name: Necessary to have username

1.3 Email: Necessary to have email

1.4 Password: Necessary to have password

1.5 Accept terms of use: Necessary to accept terms of use

Description: guest user will have the following attributes,

- Guest id, INT
- Email id, VARCHAR(80)
- Is register, TINYINT

2. Registered user: A user who has successfully created their account with FitHub and can access all the app features

2.1 Login: Necessary to have username and Password

2.2 User interests: This information will be used by FitHub to show recommendations of people with similar interest

2.3 User Profile: All the user-related information that can be used to create its profile

2.4 User logs: Previous workout partners or events that the user has been to can be looked at in the user logs

2.5 Activities: All the indoor and outdoor activities that FitHub allows a user to search or select as interest.

Description: register user will have the following attributes,

- Reg id, INT
- User id, INT
- Phone, VARCHAR(12)
- Address, TEXT
- Zip code, VARCHAR(5)
- Activity type, TEXT
- Birthdate, DATE

3. Gym membership: A user's gym membership information. This information will be used in order to send buddy pass request by other users

3.1 Buddy Pass Owner: A user having gym membership and willing to share his gym pass with a friend so that they can workout together

Description: gym membership will have the following attributes,

- Gym id, INT
- Reg id, INT
- Gym Name, VARCHAR(45)
- Gym Location, VARCHAR(45)
- Zip code, INT
- Membership start date, VARCHAR(45)
- Membership expiry date, VARCHAR(45)

4. Events:

4.1 A user shall post events

4.2 Events shall be viewed by other users

4.3 Users can join the event

Description: event will have the following attributes<

- Event id, INT
- Reg id, INT
- Start time, INT
- End time, INT
- Date start, DATE
- Date end, Date

5. Friends:

5.1 Users that match based on interest

5.2 Users that match based on location

5.3 Users can send friend request

5.4 Users can accept friend request

Descriptions: friends will have the following attributes,

- User id, INT
- Reg id, INT

- Name, VARCHAR(40)
- Contact info, VARCHAR((50))
- send/accept request, TINYINT

## 6. Friend List:

### 6.1 User shall have a friends list

Description: A friends list will have the following attributes,

- Reg\_id, INT
- Name, VARCHAR(40)
- Contact info, VARCHAR(50)

## 7. Private chat:

### 7.1 Users shall be friends

### 7.2 Users can send private messages from friends list

Description: private chat will have the following attributes,

- User id, INT
- Reg id, INT
- Name, VARCHAR(40)
- Date, DATE
- Time, TIME

## 8. Group chat:

### 8.1 Users shall be friends

### 8.2 Users can have group conversation

### 8.3 Users can invite other friends

Description: group chat will have the following attributes,

- User id, INT
- Reg id, INT
- Name, VARCHAR(40)
- Date, DATE
- Time, TIME

9. Clubs:

- 9.1 Users can create clubs
- 9.2 Users can join/send request
- 9.3 clubs can be public/private

Description: clubs will have the following attributes,

- Club id, INT
- Reg id, INT
- Open to all, TINYINT

10. Club owner:

- 10.1 Clubs can only have one owner

Description: club owner will have the following attributes<

- Club owner id, INT
- Reg id, INT
- User id, INT
- Date started, DATE

11. Club List:

- 11.1 List of clubs users can view

- 11.2 List of clubs users can join

Description: club list will have the following the following attributes,

- User id, INT
- Reg id, INT
- Name, VARCHAR(40)

## **2. Functional Requirements V3**

### **P1 (Mandatory)**

#### **Guest User:**

1. Guest users shall be able to view events posted by other users.
2. Guest users shall be able to select their interests.
3. Guest users shall be able to register.
4. Guest users shall be able to Log in as registered users.
5. Guest users shall be able to access the homepage.
6. Guest users shall be able to access the About us page.
7. Guest users shall be able to access the Contact us page.
8. Guest users shall be able to access the Support page
9. Guest users shall be able to delete their account from FitHub

#### **Registered User:**

11. Registered users shall be able to access homepage
12. Registered users shall be able to access About us page
13. Registered users shall be able to access Contact us page
14. Registered users shall be able to access Support Page
15. Registered users shall be able to send workout invites to other users.
16. Registered users shall be able to select their interests
17. Registered users shall be able to edit their information
18. Registered users shall be able to update their profile picture
19. Registered users shall be able to delete their profile picture
20. Registered users shall be able to deactivate their account
22. Registered users shall be able to change their account passwords
26. Registered users shall be able to delete their account from FitHub

#### **Searching People**

27. Registered users shall be able to have a certain number of swipes per day.
28. Registered users shall be able search for buddies with similar interest
29. Registered users shall be able search for buddies nearby

30. Registered users shall be able to filter search options based on their interests.
31. Registered users shall be able to filter search options based on how far they are willing to travel.

## **Friends**

33. Registered users shall be able to find friends to exercise with.
34. Registered users shall be able to add other registered users as friends.
35. Registered users shall be able to access their friend list
36. Registered users shall DM (direct message) other registered users only if they are friends.
37. Registered users shall have multiple friends if they choose to.
38. Registered users shall have 0 friends if they choose to.
39. Registered users shall have the ability to unfriend a former friend.
40. Register users shall have the ability to block a former friend.
41. Register users shall have the ability to report a former friend.
43. Registered users shall view friend's event postings.
44. Registered users shall be able to decline friend requests.
45. Registered users shall be able to accept friend requests.

## **Events**

46. Registered users shall be able to create events
47. Registered users shall be able to delete events they created
48. Registered users shall be able to edit the events they created
49. Registered users shall be able to join an event
50. Registered users shall be able to exit from an event
51. Registered users shall be able to rejoin an event
52. Registered users shall invite people to events they created

## **Chats**

70. Registered users shall be able to create a private chat
78. Registered users shall be able to reject to join in a private chat
80. Registered users shall be able to accept to join in a private chat

## **Web Application**

89. Web Application shall have About us Page
90. Web Application shall have Contact us page
91. Web Application shall ask user to log in
92. Web Application shall display user's profile
93. Web Application shall show notifications to the user
94. Web Application shall show friend requests
95. Web Application shall allow user to check their messages
96. Web Application shall allow user to check event dates they are planning to go
98. Web Application shall allow user to change password of their account
99. Web Application shall allow user to deactivate their account
101. Web Application shall show the logs of user activities
102. Web application shall show events occurring nearby
103. Web Application shall show people recommendation with similar interest
104. Web Application shall allow user to log out
105. Web Application shall allow user to delete the account
106. Web Application shall allow user to see list of friends
108. Web Application shall allow user to search for people using various filter options
109. Web Application shall show people recommendation with similar interest
- 110. Web application has admin account to approve events**
- 111. Web Application shall have Support/FAQ page**

## **P2 (Desired)**

### **Registered User:**

24. Registered users shall be able to send an introductory message to a different user in their search result.
25. Registered users shall be able to share their real time location with FitHub

### **Buddy Pass**

82. Registered users shall be able to view people with gym membership
83. Registered users shall be able to put request to access buddy pass
84. Registered users shall be able to accept the request to access buddy pass

85. Registered users shall be able to reject the request to buddy pass  
Web Application shall show buddy pass requests

### **History logs**

86. Registered users shall access their event visits in logs  
87. Registered users shall access their workout pairing information in logs  
88. Registered users shall be able to access their workout partner's information from logs

### **Registered User:**

21. Registered users shall be able to save their frequent searches

### **Searching People**

32. Registered users shall be able to remove people from their search results

### **P3 (Opportunistic)**

#### **Friends**

42. Registered users shall view friend's club postings.

#### **Clubs**

53. Registered users shall be able to create the club  
54. Registered users shall be able to update the picture of the club they created  
55. Registered users shall be able to edit the club name they created  
56. Registered users shall be able to delete the club they created  
57. Registered users shall be able to delete the club picture they created  
58. Registered users shall be able to send request to join the club  
59. Registered users shall be able to leave a club  
60. Registered users shall be able to send request rejoin a club  
61. Registered users shall be able to add members to the club they created  
62. Registered users shall be able to remove members from the club they created  
63. Registered users shall be able to post in club  
64. Registered users shall be member of multiple clubs  
65. Registered users shall be admin of multiple clubs  
66. Registered users shall be able to invite people to club

67. Registered users shall be able to accept the joining request for club they created
68. Registered users shall be able to reject the joining request for club they created

### **Chats**

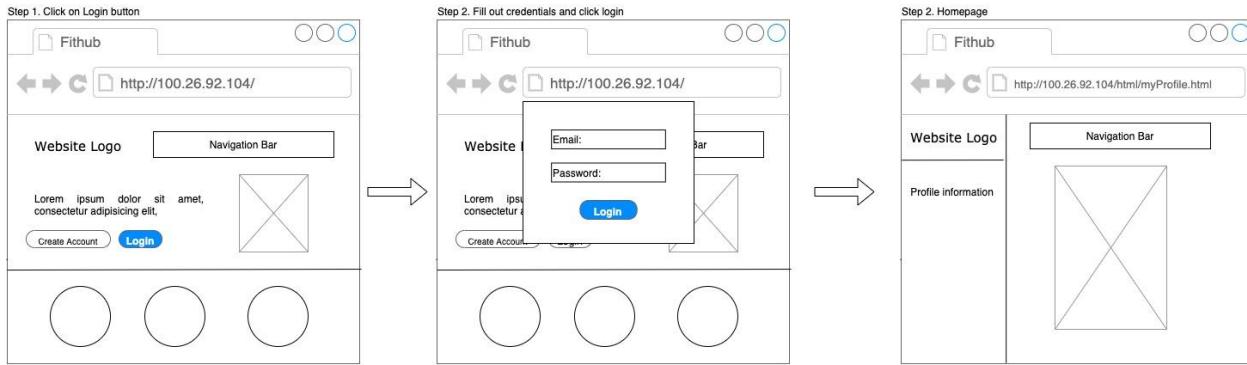
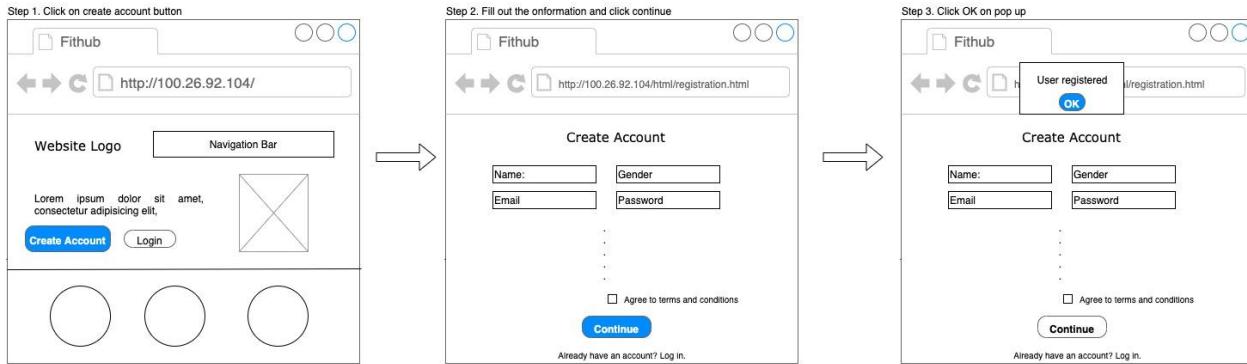
69. Registered users shall be able to create a group chat
71. Registered users shall be able to invite friends to the group chat
72. Registered users shall be able to remove people from group chat
73. Registered users shall be able to add unknown people from the chat
74. Registered users shall be able to post in the group chat
75. Registered users shall be able to leave a group chat
76. Registered users shall be able to dissolution the group chat they cheated
77. Registered users shall be able to recall their posts
79. Registered users shall be able to reject to join in a group chat
81. Registered users shall be able to accept to join in a group chat

### **Web Application**

97. Web Application shall allow user to check updates of the club
107. Web Application shall allow user to see list of clubs

### 3. Wireframes Based on your Mockups/Storyboards V2

#### 1. Registration and log in



## 2. Finding a Partner

Step 1: Log in successfully and click on "Find a buddy"

My Profile | Fithub

http://100.26.92.104/html/myProfile.html

FITHUB

Find A Buddy

My Profile

Last work out partner

Matches Messages Friends

User 1 User 2 User 3

Step 2: Select filters and click search

Find a buddy | Fithub

http://100.26.92.104/html/matching.html

FITHUB

Find A buddy

Select the filters

Search

Matches Messages Friends

User 1 User 2 User 3

Step 4: Click ok on the pop up and your request has been sent

Find a buddy | Fithub

http://100.26.92.104/html/matching.html

FITHUB

Find A buddy

Select the filters

Search

Matches Messages Friends

User 1 User 2 User 3

Workout request sent

OK

Search Results:

✓ ✗ ✎

Step 3: Click on tick to send workout request

Find a buddy | Fithub

http://100.26.92.104/html/matching.html

FITHUB

Find A buddy

Select the filters

Search

Matches Messages Friends

User 1 User 2 User 3

Search Results:

✓ ✗ ✎

### 3. Posting Events

Step 1: Log in successfully and click on "Events"

The screenshot shows the FITHUB application interface. At the top, there is a header with a file icon, 'My Profile | Fithub', and three circular icons. Below the header is a navigation bar with back, forward, and search/clear buttons, and the URL 'http://100.26.92.104/html/myProfile.html'. The main area is titled 'FITHUB' with a 'My Profile' link. A large button labeled 'Events' is highlighted in blue. To its right are 'Account' and '...' buttons. The central part of the screen features a large square with a diagonal cross through it.

Step 2: Click on "Create Event" button

This screenshot shows the same FITHUB interface as Step 1, but with a different focus. The 'Events' tab is still selected, but the 'Create Event' button within the 'Events' section has been highlighted in blue. Below the tabs, there is a 'Search filters' section with a 'Search' button. Three search boxes are visible, each containing 'Event name @ location', 'Time Day', and 'Created by - user' with a checked checkbox.



Step 4: Click OK on the pop up, The event will be created

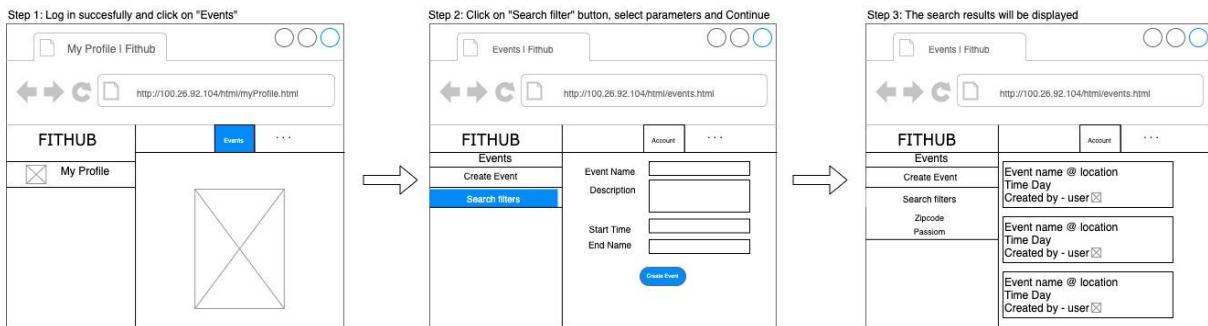
This screenshot shows the FITHUB interface again. A small pop-up window in the center says 'Event Created' with an 'OK' button. The main area contains a 'Create Event' form with fields for 'Event Name' (empty), 'Description' (empty), 'Start Time' (empty), and 'End Name' (empty). Below the form is another 'Create Event' button. The 'Events' tab is still highlighted.



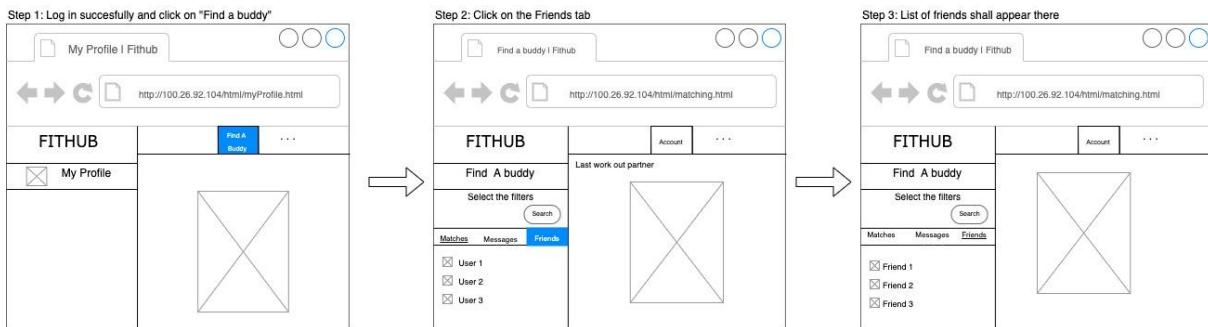
Step 3: Fill out the event information and click on Save event

This screenshot shows the FITHUB interface with the 'Events' tab selected. The 'Create Event' button is highlighted in blue. The event creation form below has all fields filled: 'Event Name' (empty), 'Description' (empty), 'Start Time' (empty), and 'End Name' (empty). A 'Create Event' button is at the bottom right of the form.

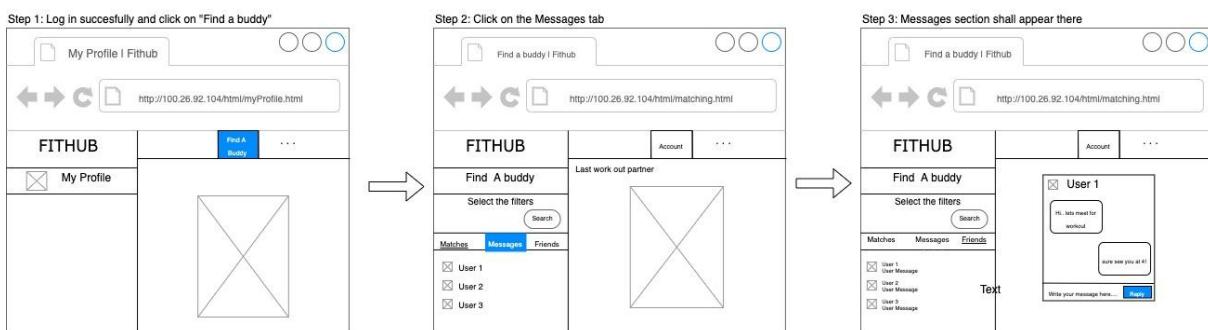
## 4. Finding events



## 5. Friends



## 6. Private Chat



## **4. High level database architecture and organization V2**

- **DB organization:**

1. ***Business rules:***

- a. Guest User:

- A guest shall have one account
- A guest shall be at most one registered user
- A guest shall be able to search many Users
- A guest shall be able to view many events

- b. Registered User:

- A registered user shall have one account
- A registered user shall be able to search many Users
- A registered user shall be able to create and delete many events
- A registered user shall be able to send, cancel, accept and reject many workout requests
- A registered user shall have many gym memberships
- A registered user shall send, cancel, accept and reject many friend requests

- c. Account:

- An account shall be associated with a guest user or a registered user
- An account shall have all the information of a user
- An account shall be accessed by a single credentials combination only (username-password)

- d. Events:

- An event shall be created and deleted by a Registered User
- An event shall be viewed by many Users

- e. Clubs:

- An club shall be created and deleted by a Registered User
- An club shall be joined by many Registered Users

f. Gym Membership:

- › A gym membership shall belong to a registered user
- › A gym membership shall have option for at most one buddy pass

g. History Logs:

- › A history log shall belong to a registered user
- › A history log shall have information related to user activities

h. Blocked Users:

- › A block list shall have many users blocked by other users
- › A block list shall be used by many registered users

i. Buddy pass:

- › A buddy pass belongs to a gym membership
- › A buddy pass shall be used by any registered user

j. Friend request

- › A friend request shall be sent by a registered user
- › A friend request shall be canceled by a registered user
- › A friend request shall be accepted by a registered user
- › A friend request shall be rejected by a registered user

k. Workout request

- › A workout request shall be sent by a registered user
- › A workout request shall be canceled by a registered user
- › A workout request shall be accepted by a registered user
- › A workout request shall be rejected by a registered user

**2. Entities:**

a. Guest User (Strong)

- › user\_id: unique user id to identify the guest user
- › email\_id: unique email id associated with a user

- › is\_registered: if the user is a registered user
- b. Registered User (Weak)
- › reg\_id: unique user id to identify the registered user
  - › user\_id: id associated with a guest user
  - › phone: contact no of the registered user
  - › address: address of the registered user
  - › location: city name of the registered user
  - › zipcode: zipcode of the registered user
  - › activity\_type: like if the user is interested in indoor/outdoor activities
  - › workout\_type: kind of workout user prefers to carry out
- c. Account (Weak)
- › acc\_id: unique id for the account entity
  - › reg\_id: id associated with a registered user
  - › username: used to log in account
  - › password: password to log in account
- d. Events (Weak)
- › event\_id: unique id for the events entity
  - › reg\_id: id associated with registered user
  - › description: more info about the event
  - › start\_time: start time for the event
  - › end\_time: end time for the event
  - › from\_date: start date for the event
  - › to\_date: end date for the event
- e. Clubs (Weak)
- › club\_id: unique id for the clubs entity
  - › reg\_id: id associated with a registered user
  - › description: more information on the club
  - › open\_to\_all: if other registered users can join without invite

f. Gym Membership (Weak)

- › gym\_id: unique id for the gym membership info
- › reg\_id: id associated with a registered user
- › gym\_name: gym name the user has membership for
- › gym\_loc: location of the gym
- › gym\_zipcode: zipcode in which the gym is located
- › buddy\_pass: does it have the option for buddy pass
- › share\_pass: if the registered user is willing to share his buddy pass
- › membership\_start\_date: start date for gym membership
- › membership\_expiry\_date: end date for gym membership

g. History Logs (Weak)

- › log\_id: unique id to identify the log of a registered user
- › reg\_id: id associated with a registered user
- › activty\_info: information on the user activity
- › is\_workout\_request: if log is for workout request
- › is\_friend\_request: if log is for friend request
- › is\_buddy\_pass\_request: if log is for buddy pass request
- › is\_block\_user: if log is for blocking a user
- › is\_club\_request: if log is for a club request

h. Blocked Users (Weak)

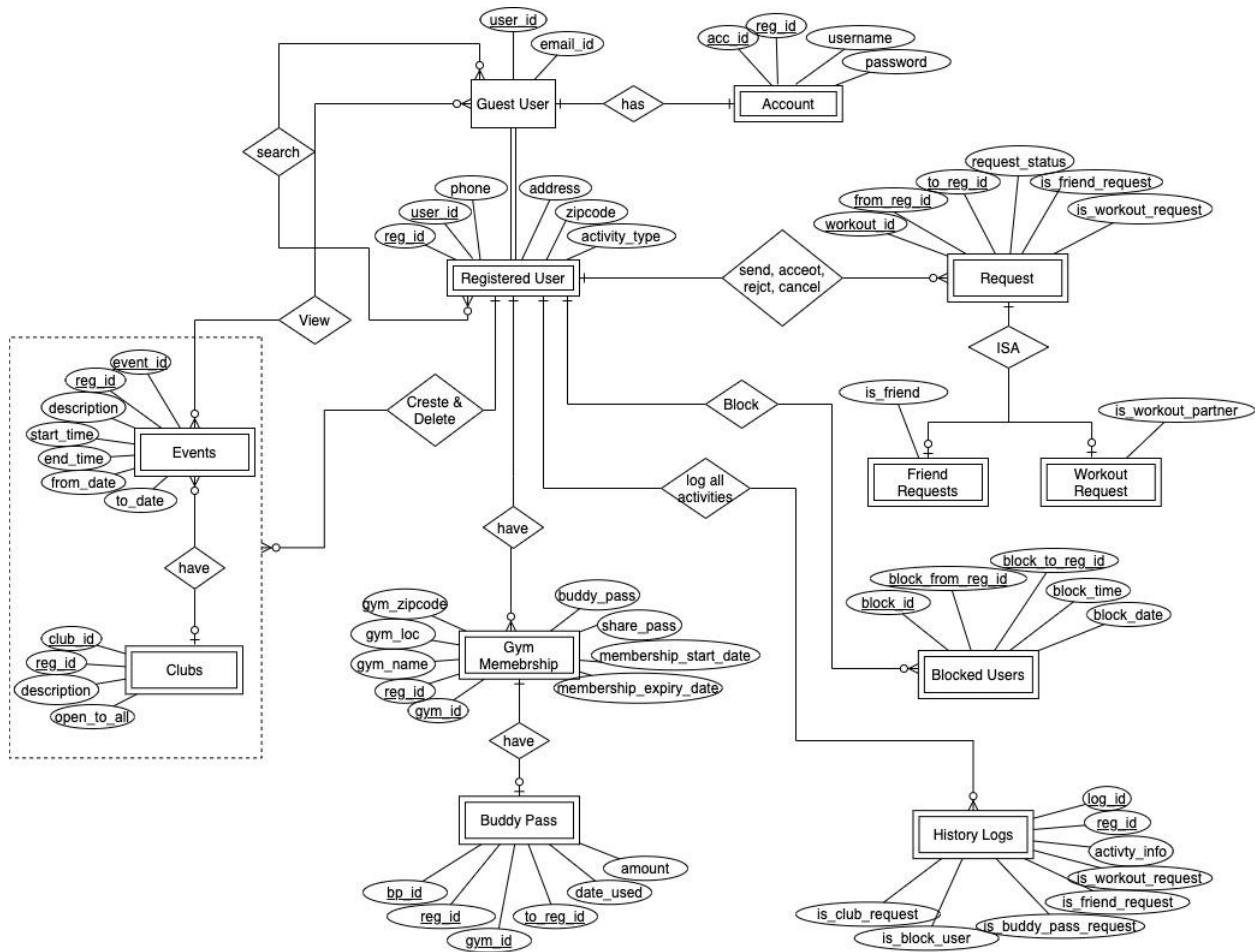
- › block\_id: unique id to for blocking entity
- › block\_from\_reg\_id: id associated with a registered user who is blocking
- › block\_to\_reg\_id: id of the registered user who is being blocked
- › block\_date: date the user was blocked
- › block\_time: time user was blocked

i. Buddy pass (Weak)

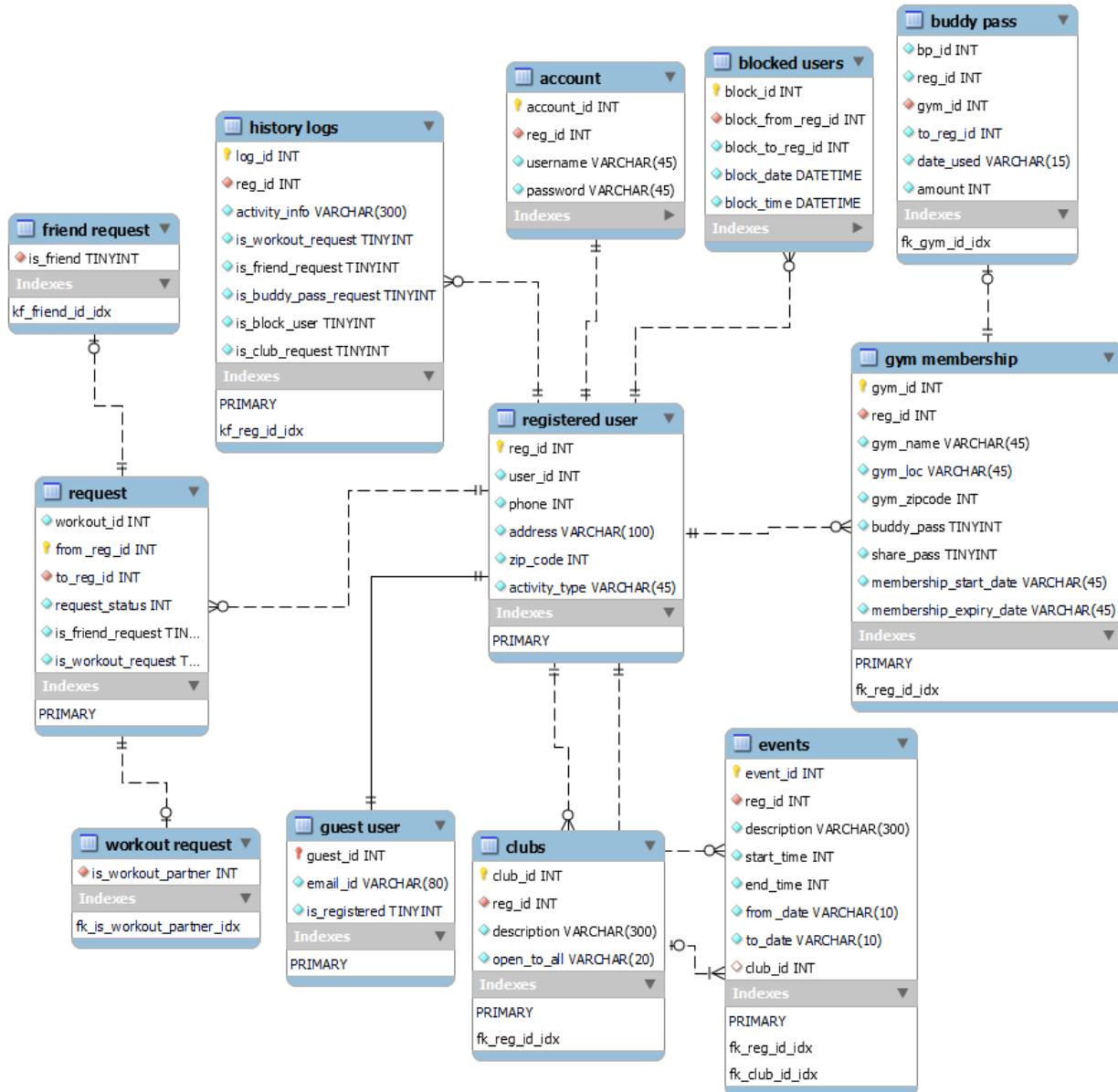
- › bp\_id: unique id for the buddy pass entity
- › reg\_id: id associated with a registered user
- › gym\_id: buddy pass belong to which gym
- › to\_reg\_id: registered user id with whom the buddy pass is being shared

- › date\_used: the date on which the user is sharing buddy pass
  - › amount: amount user collects for the buddy pass mechanism
- j. Friend request (Weak)
- › friend\_id: unique id for the friend entity
  - › from\_reg\_id: id of a registered user who is sending friend request
  - › to\_reg\_id: id of a registered user to whom the request is sent
  - › request\_status: status of the request as sent, cancel, accept, reject
- k. Workout request (Weak)
- › workout\_id: unique id for the workout entity
  - › from\_reg\_id: id of a registered user who is sending workout request
  - › to\_reg\_id: id of a registered user to whom the request is sent
  - › request\_status: status of the request as sent, cancel, accept, reject

### 3. ERD:



#### 4. Database Model:



## 5. DBMS:

The database chosen to develop the project is MySQL since it is well known RDBMS, easy to use and many GUI tools are available for the development and maintenance of the MySQL database.

- **Media storage:**

- The media used in the FltHub app will be images of the profile pictures and club pictures.
- They will be stored in the **file system** and the format of image will be JPEG/ JPG/ PNG.

- **Search/filter architecture and implementation:**

- Search Algorithm:

- The input to the search algorithm shall be provided from the search bar.
- The user will input only usernames to search from the search bar
- The input will be looked into the database
- The output for the search algorithm will be the list of usernames that fully / partially matches the user input in the ascending order

The DB fields that will be searched here is :

- Table: Account
  - username

- Filter Algorithm:

- The input to the filter algorithm shall be provided by clicking on search button from the filters section
- The user will be given the option to search based on the location, indoor/outdoor activities, workout type by selecting the appropriate filters.
- These filtered inputs shall be looked into the database. These fields shall also be indexed in the database for quick search results.

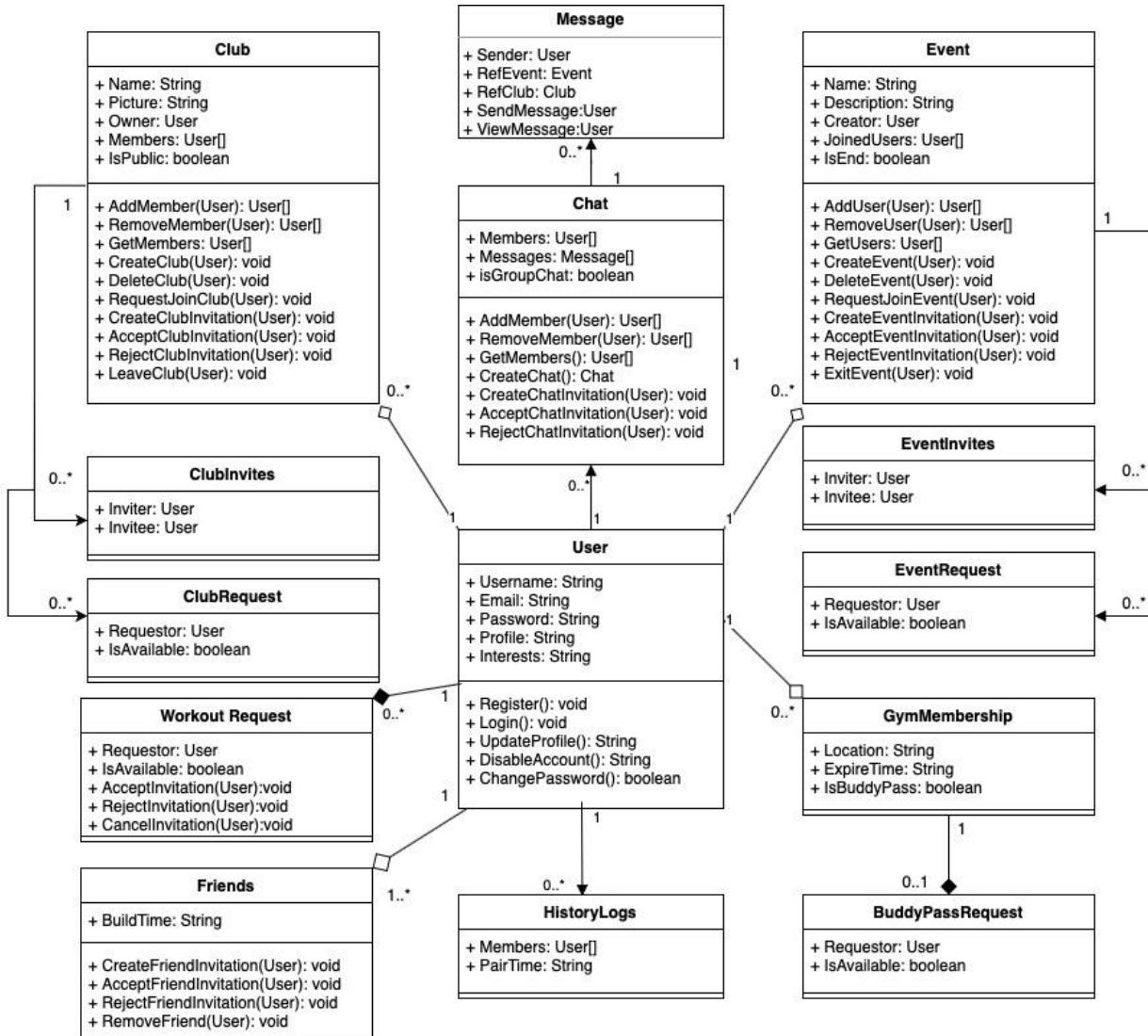
- The output for the search algorithm will be the list of items that match the user's filter input. The information in each item fetched from the database will have the username, location, zip code, activity\_type, workout\_type.

The db terms that will be searched here is :

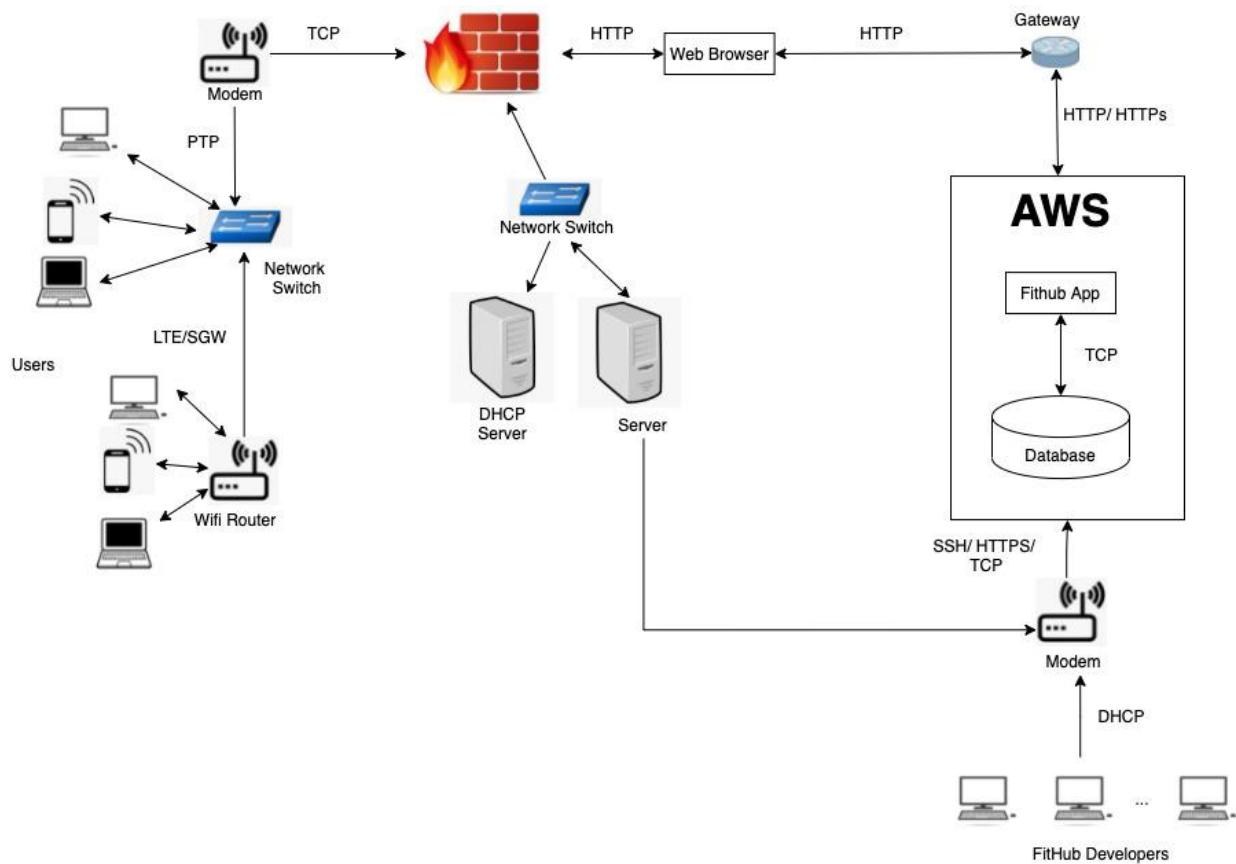
- Table: Account
  - username
- Table: Registered User
  - location
  - zipcode
  - activity\_type
  - workout\_type

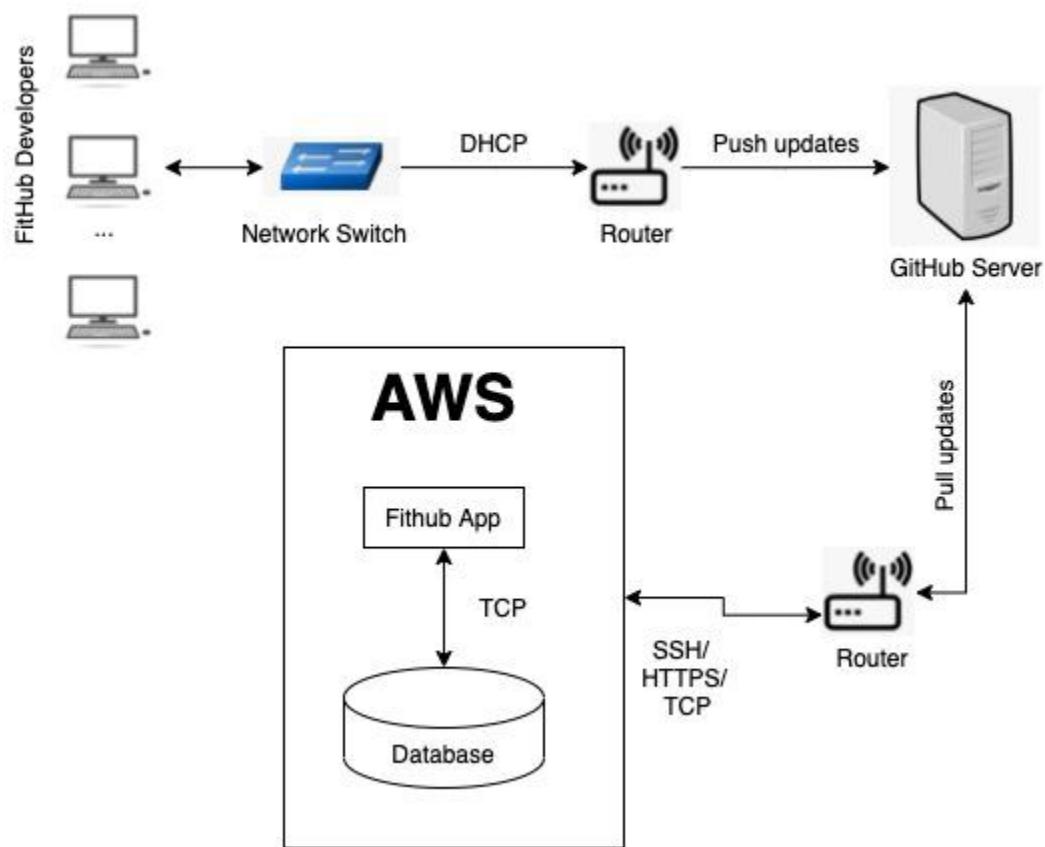
## 5. High Level Diagrams V2

**UML:**



### Application Network Diagram:



**Deployment Diagram:**

## **6. List of Contributions in this milestone (detailed including contributions to the horizontal prototype)**

<b>Team Member</b>	<b>Contribution</b>
Vidhi Vora <i>(Team Lead)</i>	<p><i>As a Team Lead:</i></p> <ul style="list-style-type: none"> <li>› Ensuring all the requirements are met and adhering to the M3 guidelines</li> <li>› Assigning and supervising the task progress, organizing regular team meetings and the agenda</li> <li>› Resolving queries, discussing the concepts for the M3</li> </ul> <p><i>As a Team Member (Milestone 2):</i></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Wireframes Based on your Mockups/Storyboards</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Create event API</li> <li>› Login API</li> <li>› My Profile page</li> </ul>
Johnson Nguyen <i>(Backend Lead)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Find a buddy page</li> <li>› Creating sessions for log in</li> </ul>
Roberto Simental <i>(Frontend Lead)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Events Page</li> <li>› Event profile page</li> <li>› Design fixes on pages</li> </ul>

Michael Satumba <i>(Frontend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Find a buddy page</li> <li>› My Profile page</li> </ul>
Eduardo Hernandez <i>(Backend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Data definitions</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Creating sessions for log in</li> <li>› Create event page</li> <li>› Testing for bugs</li> </ul>
Zhinan Zhao <i>(Backend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Creating sessions for log in</li> <li>› Search events API</li> <li>› Search bar in home page header</li> </ul>
Ziming Wang <i>(Frontend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>›</li> </ul> <p><i>Horizontal Prototype:</i></p> <ul style="list-style-type: none"> <li>› Event profile page</li> </ul>

(implementation of session is still in progress)

**MILESTONE4****SW Engineering CSC648/848 Summer 2021****FitHub**

**Project Application And Name:** Find a fitness partner - “FitHub”

**Team Info:** Team 07 | Error 404

**Team Members:**

1. Vidhi Vora (Team Lead and GitHub Master) - vvora@sfsu.edu
2. Roberto Simental (Front-End Lead) - rsimental@mail.sfsu.edu
3. Johnson Nguyen (Back-End Lead) - Jnguyen63@mail.sfsu.edu
4. Zhinan Zhao
5. Eduardo Hernandez
6. Ziming Wang
7. Michael Satumba

**Milestone:** Milestone 4

**Date:** 30 July 2021

**History:**

<b>Version</b>	<b>Date</b>
M4V2	02 August 2021
M4V1	30 July 2021
M3V2	30 July 2021
M3V1	22 July 2021
M2V2	22 July 2021
M2V1	08 July 2021
M1V2	13 July 2021
M1V1	22 June 2021

## Table Of Contents

1. Product summary	3
2. Usability test plan	5
3. QA test plan	15
4. Code Review	18
5. Self-check on best practices for security	20
6. Self-check: Adherence to original Non-functional specs	22
7. List of contributions to the document	27

## **1. Product summary**

- Name of the product: Fithub
- Final P1 (Mandatory):

### **Registered User:**

1. Registered users shall be able to access About us page
2. Registered users shall be able to access Contact us page
3. Registered users shall be able to access Support Page
4. Registered users shall be able to send workout invites to other users.
5. Registered users shall be able to select their interests
6. Registered users shall be able to edit their information
7. Registered users shall be able to update their profile picture
8. Registered users shall be able to delete their profile picture
9. Registered users shall be able to change their account passwords
10. Registered users shall be able to delete their account from Fithub
11. Registered users shall be able to access About us page
12. Registered users shall be able to access homepage

### **Searching People:**

13. Registered users shall be able search for buddies with similar interest
14. Registered users shall be able search for buddies nearby
15. Registered users shall be able to filter search options based on their interests.

### **Friends:**

16. Registered users shall be able to find friends to exercise with.
17. Registered users shall DM (direct message) other registered users only if they are friends
18. Registered users shall view friend's event postings

### **Events:**

19. Registered users shall be able to create events
20. Registered users shall be able to delete events they created
21. Registered users shall be able to edit the events they created
22. Registered users shall be able to join an event
23. Registered users shall be able to exit from an event
24. Registered users shall be able to rejoin an event
25. Registered users shall invite people to events they created

### **Chats:**

26. Registered users shall be able to create a private chat

**Web Application:**

27. Web Application shall have About us Page
28. Web Application shall have Contact us page
29. Web Application shall ask user to log in
30. Web Application shall display user's profile
31. Web Application shall show notifications to the user
32. Web Application shall allow user to check their messages
33. Web Application shall allow user to check event dates they are planning to go
34. Web Application shall allow user to change password of their account
35. Web Application shall allow user to deactivate their account
36. Web application shall show events occurring nearby
37. Web Application shall show people recommendation with similar interest
38. Web Application shall allow user to log out
39. Web Application shall allow user to delete the account
40. Web Application shall allow user to search for people using various filter options
41. Web Application shall show people recommendation with similar interest
42. Web application has admin account to approve events

- The unique points of Fithub:

1. Compared to many social platforms that already exist on the market, Fithub has clear user targeting: people who enjoy group exercise. So we combined group exercise events posting with social media. There will be no ads because we use paid membership.
2. Fithub has straightforward social features and User UI. White background with a clean navigation bar will give our users an experience of comfort and great efficiency.
3. For the best user experience, we only use ads specific to the interests of our users and related to working out. We also provide a premium subscription with unique benefits and perks for our users.

- Url: <http://100.26.92.104:3000/>

## **2. Usability test plan**

### **Send Workout Request**

**Test objectives:** Users should be able to view profiles of other people and send workout requests to them so they can pair up. The workout request has to be accepted by the other user so they can begin messaging each other. The user can send requests to multiple different users. It is important that the requests are received quickly so that users can respond and pair up. Users can also deny work out requests if they choose to.

#### **Test description:**

**System Setup:** Tester creates an account with their information and preferences to become a registered user. There are 5 users testing the system. Testers are not students and have only an average understanding of computers and software.

**Starting Point:** Tester begins the test from the logged in Homepage. From there they navigate to the appropriate page to begin testing.

**Intended user:** The target audience for Fithub are young people looking to make friends or simply work out with others. Young people are more likely to be active or trying to become more fit. They are also more likely to be using Social Media for the purpose of meeting new people.

**Test Measurements:** The test will measure how intuitive the Find a Buddy page is for new users. It will also test if the concept of our product is appealing to users.

**URL:** <http://100.26.92.104:3000/html/matching.html>

## Usability Test Table

Send Workout Request

Skip a profile if it doesn't match interests

Description	Test	% Completed	Errors	Comments	Time in seconds
Find a buddy shows a workout buddy	Clicked on “Find a Buddy Tab”	100%	none	Default buddy	0
Skip a profile if it doesn't match interest	Clicked on the X and it doesn't pass the profile	0%	Skip/reject doesn't work	none	5
Workout request sent	Clicked on the Checkmark	100%	none	Change confirmation prompt	3
User receives the work out request	Does not know if user receives it	0%	Work requests received out not	none	0

## User Questionnaire

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Matching with a buddy is simple.	0	0	3	2	0
The UI for matching buddy profiles is easy to understand	0	4	1	0	0
After matching you want to continue looking at other profiles to match with.	0	0	2	3	0

## Posting Events

**Test Objective:** Users should be able to create events. The event will then appear under the Events page. Users should be able to click on the event to show the event details. From there they can join or leave the event. Events are important for users that are looking for a group experience instead of a one on one workout. There are also many activities that can only be done in groups like playing sports like soccer or basketball. Events also allows for organizations to promote their own workout events. The join and leave feature is important to know how many people will be attending the event, so organizers can plan accordingly.

### Test description:

**System Setup:** Tester creates an account with their information and preferences to become a registered user. There are 5 users testing the system. Testers are not students and have only an average understanding of computers and software.

**Starting Point:** Tester begins the test from the logged in Homepage. From there they navigate to the appropriate page to begin testing.

**Intended user:** Any user can post an event but most events will be most likely made by organizers of some sort. Usually clubs or gyms that want to promote their events.

**Test Measurements:** The test will measure how easy it is to create an event and if our events page does a good job of promoting the events that are posted. Events should be appealing and so we must offer enough freedom for event creators to properly show off their event.

**URL:** <http://100.26.92.104:3000/html/events.html>

## Usability Task Table

### Posting/Viewing Events

Description	Test	% Completed	Errors	Comments	Time in seconds
Event is Created	Created an Event	100%	none	Change confirmation	20
Event is viewable under Events page	Tried to view my event under event's page	0%	Event does not show up	Only other events appear	5
Event details appear when you click on Event	Prior events appeared on the page but not mine	100%	none	Visuals can be improved	3
Event joinable	Tried to join an event	0%	Cannot join	Buttons are there but no confirmation	4
Event leavable	Tried to leave an event	0%	Cannot leave	Buttons are there but no confirmation	3

### Questionnaire

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Creating an event with all the necessary details is simple.	0	0	1	4	0
Events are enticing to click on.	0	2	1	2	0
Event profiles are interesting and make you want to join.	1	3	1	0	0

## Sending Messages

**Test Objective:** Users should be able to message each other after pairing up. This allows them to plan their meeting and work out. Messages have to be received immediately so that users can reply back in a timely manner. We are testing if messages are properly going from one user to another reliably. Also it is important that our chat system can compete with other websites so that users don't feel compelled to switch to a different website to continue their conversations there.

### Test description:

**System Setup:** Tester creates an account with their information and preferences to become a registered user. There are 5 users testing the system. Testers are not students and have only an average understanding of computers and software.

**Starting Point:** Tester begins the test from the logged in Homepage. From there they navigate to the appropriate page to begin testing.

**Intended user:** All users will make use of the message feature to be able to coordinate with any buddies they pair up. Using our message system allows people to get to know each other before giving away private information like their phone number or social media account.

**Test Measurements:** This test measures the user satisfaction of sending messages on Fithub versus other social media sites. It will compare our messaging features to sites that users are already familiar with.

**URL:** <http://100.26.92.104:3000/html/matching.html>

## Usability Task Table

### Sending User Messages

Description	Test	% Completed	Errors	Comments	Time in seconds
Send a message	Tried to send a message	0%	Does not send message	Not ready	0
Other user receives the message	Tried to view message from other perspective	0%	Does not send message	Not ready	0
Other user replies	Tried to type in a reply	0%	Does not send or receive messages	Not ready	0
Message is received	Tried to look for a message sent by a user	0%	Does not send message	Not ready	0

### Questionnaire

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
It is easy to send a message to another user.	5	0	0	0	0
The chat system has all the features you want.	5	0	0	0	0
The chat system is appealing to look at.	5	0	0	0	0

## Edit Profile

**Test Objective:** Users should be able to change their profile information. A user's profile should be up-to-date as much as possible so they can find events and buddies that match their location and preferences. We are testing that changes are easy to make and update the profile of the user.

### **Test description:**

**System Setup:** Tester creates an account with their information and preferences to become a registered user. There are 5 users testing the system. Testers are not students and have only an average understanding of computers and software.

**Starting Point:** Tester begins the test from the logged in Homepage. From there they navigate to the appropriate page to begin testing.

**Intended user:** Most users will want to have a detailed profile page. That way it is easier for them to find like minded people to workout with. Users should frequently update their profile page to improve their experience and that of other users.

**Test Measurements:** The test measures how long it takes for users to be able to make the desired changes to their profile. Also if we provide enough customization for the user profiles so that they are a good reflection of the individual user.

**URL:** <http://100.26.92.104:3000/html/myProfile.html>

## Usability Task Table

### Edit Profile

Description	% Completed	Errors	Comments	Time in seconds
Account settings are selected	90%	none	Hard to find	13
Profile Changes are made	90%	none	Can't change username	20
Profile changes are saved	50%	Profile doesn't update	No way to verify	8
My Profile shows updated changes	0%	Data doesn't show	none	5

### Questionnaire

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
You can quickly find the profile settings you want to edit.	0	0	0	4	1
Your profile has enough preference options to match your personality.	0	0	0	2	3
The edit profile module is easy to understand.	0	0	1	3	1

## Finding a Buddy with Filters

**Test Objective:** Users should be able to use the different search settings to find a buddy that matches their preferences. Search filters allow you to find like minded individuals for a better workout experience. The profile shown for potential matches should be more generic at the start and become more detailed as the user applies more specific search settings. The user should be happy with potential matches because they should match the user's interests.

### Test description:

**System Setup:** Tester creates an account with their information and preferences to become a registered user. There are 5 users testing the system. Testers are not students and have only an average understanding of computers and software.

**Starting Point:** Tester begins the test from the logged in Homepage. From there they navigate to the appropriate page to begin testing.

**Intended user:** Most users will want to make use of the filter settings to get better matches. Location is the most important setting because if you plan on meeting up you need to be reasonably close or else both people will feel like they wasted their time. Preferences will also play a big part on what the actual workout will be, so it is important that users use filter settings properly to make sure they have a good experience.

**Test Measurements:** This test measures how intuitive the search settings are and if the user is happy with the amount of options they have. It also measures how well our system is able to provide a profile that appeals to the user based on their settings.

**URL:** <http://100.26.92.104:3000/html/matching.html>

## Usability Task Table

Finding a workout buddy

Description	Test	% Completed	Errors	Comments	Time in seconds
Change search settings by location	Inputted a new located	100%	none	none	5
Change search settings by passion	Tried search to by adding "passions"	80%	none	Pop up is cluttered	13
A profile matching the settings is shown	Could not find any matches	0%	Profile doesn't change	Shows default profile still	2

## Questionnaire

Question	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
You are able to edit your search settings easily.	0	0	0	2	3
The search settings have enough options for you to customize your ideal workout buddy.	0	0	0	4	1
The profiles shown match your expectations based on the search settings.	5	0	0	0	0

### **3. QA test plan**

#### **Test Objectives:**

The objective of testing these functions is to make sure all of the information has passed correctly from users to our server. For all information that the users typed in when they registered, we will test if it is saved in our database in correct data form. We will mainly test if the password and email of the registrar user will be encrypted correctly. And for another key feature: event management, we will test if the searching bar and event pages are working. To make sure the registered users can see their profile, we will test the user profile page as well.

#### **HW and SW setup**

For running Fithub, the user will need a PC(tower personal computer or laptop) with stable power and Internet connection. The operating system needs to be Windows(Windows 10), Ubuntu, or IOS. The user will also need Firefox or Chrome at their latest version on the user's computer. And the user will need to paste <http://100.26.92.104:3000/> to their browser to enter Fithub. The recommended HW setup is below.

#### **Features to be tested:**

- Encrypted Password:

Test if the user's password has been hashed by MD5 and stored in the database.

- Testing of the search bar

Test if a registered user can use the search bar to find the desired event by tags.

- Testing the filter option on find a buddy page

Test if a registered user can enter the buddy's profile page by clicking specific tags.

- Testing the app on Firefox, Chrome, and IE

Test if all features of Fithub can correctly function on Firefox, Chrome, and IE.

- The application shall have the company logo visible on every pages

Test if the company logo is shown on all of the pages of Fithub.

## Test Results

No	Description	Test Input	Expected Output	Pass/Fail
1	Test user's passwords shall be hashed using Md5	testingtest@t est.test	cc03e747a6afbbcb f8be7668acfebee5	pass
1	Test user's passwords shall be stored on server	testingtest@t est.test	cc03e747a6afbbcb f8be7668acfebee5	pass
1	Test user's passwords shall be able to be checked in database	testingtest@t est.test	cc03e747a6afbbcb f8be7668acfebee5	pass
2	Test search bar by entering event tags	Running	Running as a filter for events	pass
2	Search event tags	Biking	Jump to user profile with Biking tag	pass
2	Search nonexistent event profile tags	Swimming	Please type again	fail
3	Test the filter option on find a buddy page	Biking	Biking is applied as a filter	fail
3	Test the filter option on find a buddy page	Walking	Walking is applied as a filter	fail
4	Test the app on firefox	Open app in Firefox	App opens	pass
4	Test the app on chrome	Open app in chrome	App opens	pass
4	Test the app on IE	Open app in IE	App opens	pass
5	Test application shall have the company logo visible on events page	Open event page of the app	Logo appears in event page	pass

5	Test application shall have the company logo visible on user profile page	Open user profile page of the app	Logo appears in user profile page	pass
5	Test application shall have the company logo visible on find a buddy page	Open find a buddy page	Logo appears in find a buddy page	pass
5	Test application shall have the company logo visible on event profile page	Open event profile page	Logo appears in event profile page	pass

## 4. Code Review

a. The coding style chosen here is:

- Proper code commenting. Mentioning the creator and functionality of the code
- Proper code indentation
- Meaningful variable names to ease user readability
- Dividing a major functionality into sub functions to enable reuse of code and simplicity in implementation

b. Peer code review:

The functionality that has been sent for code review is:

- A user, searching for users based on filters and sending workout requests to users whose profiles interest the most.
- The user sending a message to the user to whom the request has been sent.
- A user rejecting, cancelling a users request
- Review from another team member (Johnson Nguyen)
  - Documentation
    - There is no proper code commenting on the block of the code whoever wrote that block. Along with a description of what that block of code is specifically doing.
  - Coding conventions
    - The tabs indentations are accurate with proper closing brackets.
    - The Variable names are easily recognizable, accurate, and descriptive to what the variable will be used for.
    - Throughout the project, some of the functions that were written were modularized therefore it can be re-used and won't cause any redundancies.
  - With that being said, the code was readable and for the most part it had some good practices. I do agree with Darshan Shah that async call backs would have made it even greater than it already is.
- Review from external member (Darshan Shah - was a CS student and currently a Google employee)



Darshan Shah [REDACTED]

Fri 7/30/2021 6:09 PM

To: Vidhi Vora



1. Proper code commenting. Mentioning the creator and functionality of the code:

The comments present are descriptive and accurate. I would recommend using code comment annotations like @Author Vidhi instead of just listing the name. The name should also be on a separate line from the description.

2. Proper code indentation:

Tabs are accurate but at points excessive. Guard clauses can be used to reduce nesting in the code. There are a few nested async callbacks in the code, these can be pulled out into separate functions to reduce confusion in the code.

3. Meaningful variable names to ease user readability:

Variable names are following js conventions and are accurate and descriptive.

4. Dividing a major functionality into sub-functions to enable reuse of code and simplicity in implementation:

Overall, the functions have been descriptive and accurate. These is definitely code which is being repeated in open and close tab functions which can extracted to reduce code duplications. As mentioned above, the async callbacks could also be extracted to improve code readability.

...

## 5. Self-check on best practices for security

The major assets that are being protected are:

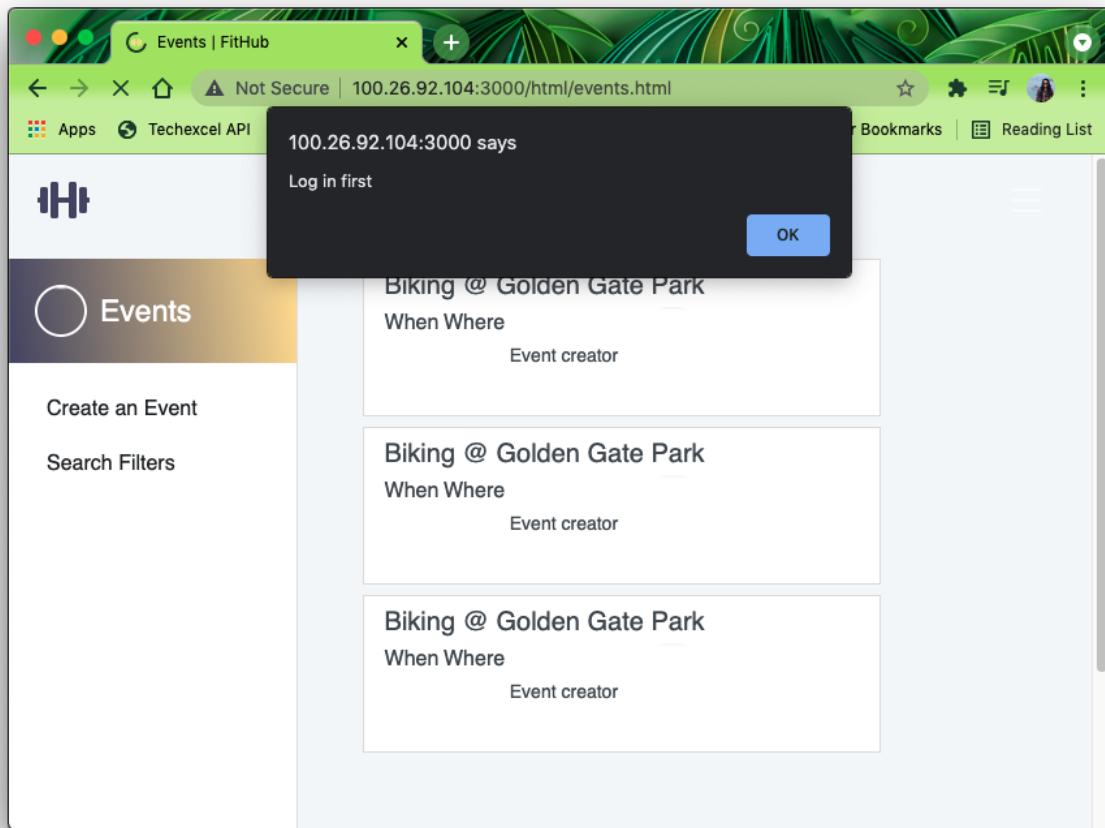
- User information: Only registered user shall be able to view information of other registered users
- User Account Password: The password is stored in the database in encrypted format. It is encrypted using MD5 that is a one way hash function and cannot be decrypted. Below is the screenshot for the same:

account_id	reg_id	username	password
1	1	vidhi1@sfsu.com	39f3fe61989747afed312c7b7465e004
2	2	users2@sfsu.com	86425c35a33507d479f71ade53a669aa

- Validated Data: All the input data entered by a user is adhered to basic validation for example email shall be of form [xxx@xx.xx](mailto:xxx@xx.xx). Below is the code snippet in JS for email validation:

```
function validateEmail(email) {
  const re =
    `/^(([^<>()\\[\\],;:\\s@"]+(\\.[^<>()\\[\\],;:\\s@"]+)*|(.+))@(([0-9]{1,3}\\.){3}[0-9]{1,3}\\.){3}[0-9]{1,3}\\]|(([a-zA-Z]-[0-9]+\\.)+[a-zA-Z]{2,}))$/;
  return re.test(String(email).toLowerCase());
}
```

- Access to Unique features: Only logged in users shall be able to access the application features. When a user requests a feature of the fithub app, the app first checks if the user is registered. Below is the screenshot for the same:



## **6. Self-check: Adherence to original Non-functional specs**

### **System Requirements**

1. The application shall be able to run on the android system(Version 10).- On track
2. The application shall be able to run on the android system(Version 11).- On track
3. The application shall be able to run on Windows10(Version 1803).- Done
4. The application shall be able to run on Windows10(Version 1809).- Done
5. The application shall be able to run on Windows10(Version 1903).-Done
6. The application shall be able to run on Windows10(Version 2004).- Done
7. The application shall be able to run on Windows10(Version 20H2).- Done
8. The application shall be able to run on Windows10(Version 21H1).- Done
9. The application shall be able to run on Ubuntu 21.04. - Done
10. The application shall be able to run on iOS 12. - Done
11. The application shall be able to run on iOS 13. - Done

### **Marketing**

12. The application shall have the company logo visible on every page. - Done
13. The application shall have the company name visible on every page. - Done
14. The application shall have non-intrusive space dedicated to relevant ads. - Done
15. The application shall have meta content for SEO purpose.- On track

### **Legal requirements**

16. The application shall prompt for Terms and Conditions for each new user to accept. - Done
17. The application shall have all users verify that they're 18 years of age or older. - Done
18. The application shall have copyright information on every page. - Done

### **Privacy**

19. The application shall prompt user to accept policies when creating account - Done
20. The application shall ensure user's data will be shared between registered users - Done
21. The password shall be needed for log-in action. - Done

22. The user's passwords shall be hashed using Md5. - Done
23. The application's Network will be protected by the AWS Firewall - Done
24. The application shall protect the privacy of users. - Done
25. The application shall present all privacy policies appropriately to all users. - Done
26. Users' private information should not be shared with 3rd platforms. - Done
27. The basic security of the site shall all be tested. - Done

## User Interface

28. The application shall be aesthetically pleasing - Done
29. The application shall have complementary colors - Done
30. The application shall be easy to navigate for everyone - Done
31. The application shall have a search bar - Done
32. The application shall have a simple user interface - Done
33. The application shall have a Filter button - Done
34. The application shall have a Register button - Done
35. The application shall have a Login button - Done
36. The application shall have a Home button - Done
37. The application shall have a Search button - Done
38. The application shall have a Private Chat room - Done
39. The application shall have a Terms and Condition - Done
40. The application shall have a Home page button - Done
41. The application shall have a User page button - Done
42. The application shall have a Logout Button - Done
43. The application shall have a Profile Picture of the User - Done
44. The application shall have an Upload button - Done
45. The application shall have a Block button - Issue (feature removed from P1)
46. The application shall have an Unblock button - Issue (feature removed from P1)
47. The application shall have pictures regarding to the activity - Done
48. The application shall have a Nearby Button - Issue (feature removed from P1)
49. The application shall have aesthetically pleasing photography - Done
50. The application shall have smooth scrolling - Done
51. The application shall have footer on every page - Done
52. The application shall have header on every page - Done

53. The application shall provide clear instructions on usability of features - Done

## **Usability**

54. The application shall be optimized for mobile browsers. - Done

55. The application shall store data in the chosen database - Done

56. The user personas shall be used. - Done

57. The application shall have a Support page - Done

58. Users that are older than the age of 18 shall be able to use the application to post information, search events, and register after spending 5 minutes reading the support page. - Done

## **Performance and scalability**

59. The Response Time shall be 3 seconds - Done

60. The Throughput shall be between 128 MiB/s and 250 MiB/s - Done

61. The system availability shall be 99%(AWS) - Done

62. A REST service call shall respond with 500 milliseconds, - Done

63. A synchronous database request shall respond with 1000 milliseconds, - Done

64. A batch process shall complete its processing within 60 minutes.- Done

65. The application shall be able to handle large number of users - Done

66. The application shall be developed using microservices architecture enabling easy maintenance and launching/editing features to FitHub - Done

67. The application shall be capable to recover from failures - On track

68. The application shall provide user the requested data - Done

69. The application shall be able to update the chat data in real time - Done

70. The application shall have live updates regarding events - On Track

## **Coding Standards**

71. The code shall be readable - Done

72. The code shall be well aligned and properly formatted - Done

73. The code shall be commented wherever needed - Done

74. The code shall be organized with one coding style - Done

75. The code shall be accessible to all the developers - Done

76. The code versions shall be maintained using GitHub - Done

- 77. The code shall be pushed to different branches - Done
- 78. The code shall be merge to the master once approved by leads and leader - Done
- 79. The code shall be in the repository that the professor has created for us - Done

### **Fault Tolerance**

- 80. The application shall be able to reload itself if loading request failed - Done
- 81. The application shall try reconnecting database if the request fails - Done

### **Storage**

- 82. The server shall have 16 concurrent Integrations - Done
- 83. The server shall have 20 CPU Cores - Done
- 84. The server shall have 12(RAID) Data Disk - Done
- 85. The server shall have 8 Volume Size (GiB) - Done

### **Expected Load**

- 86. The application shall support as many users AWS can support - Done
- 87. The application shall support as many user requests as AWS can support - Done
- 88. The application shall support as many database connections as AWS can support -  
Done

### **Portability and compatibility**

- 89. The application shall be able to run on Chrome 91.0.4472. - Done
- 90. The application shall be able to run on Chrome 92.0.4515. - Done
- 91. The application shall be able to run on Firefox 88.0.1.- Done
- 92. The application shall be able to run on Firefox 89.0.1.- Done
- 93. The application shall be able to run on Microsoft Edge 91.0.864.37 - Done
- 94. The application shall be able to run on Microsoft Edge 46.03.4.5155.- Done
- 95. The application shall be able to run on Safari 5. - Done
- 96. The application shall be able to run on Safari 14. - Done
- 97. The application shall be able to run with 1.8GHz or faster processor. Quad-core or  
better recommended. - Done
- 98. The application shall run with 2 or more GB of RAM. - Done

99. The application shall need a minimum of 1 GB up to 10 GB available space, depending on features installed. - Done
100. The application shall be tested and deployed utilizing AWS - Done
101. The application shall be continuously running 24/7 except for reasons of maintenance. - Done

## 7. List of contributions to the document

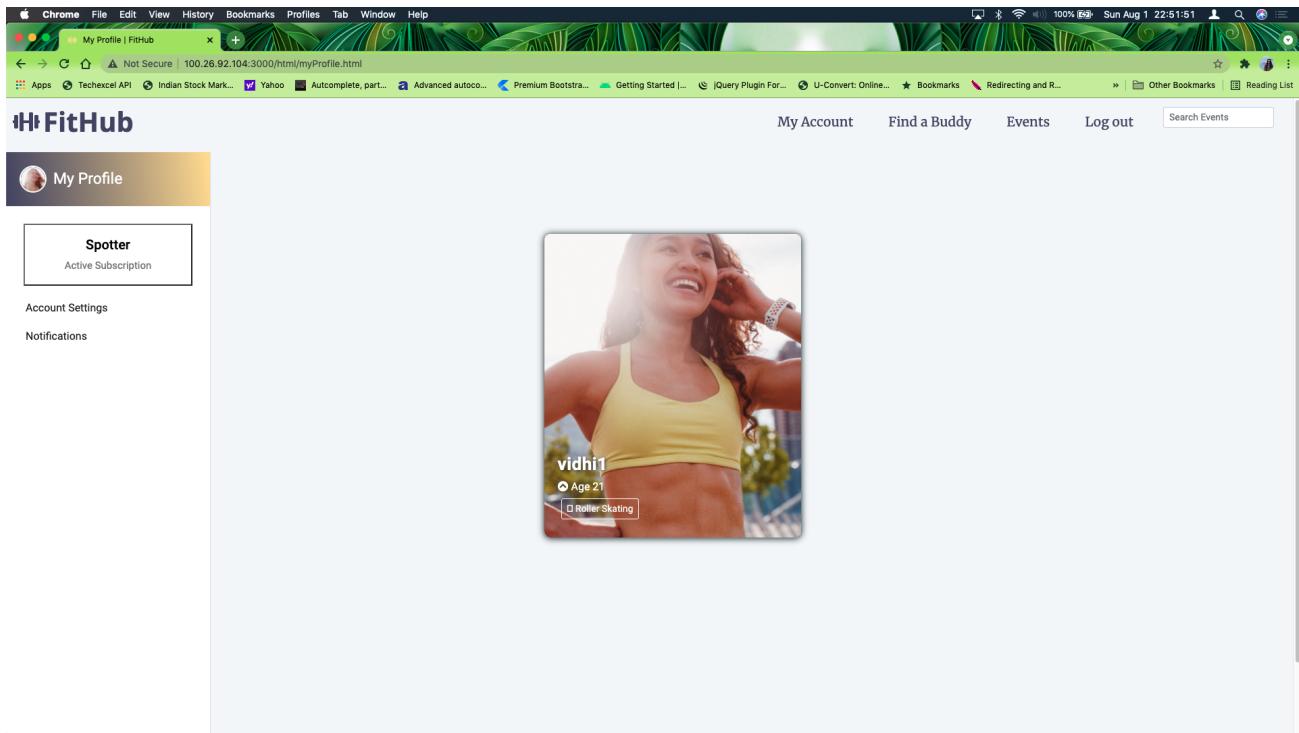
<b>Team Member</b>	<b>Contribution</b>
Vidhi Vora <i>(Team Lead)</i>	<p><i>As a Team Lead:</i></p> <ul style="list-style-type: none"> <li>› Ensuring all the requirements are met and adhering to the M4 guidelines</li> <li>› Assigning and supervising the task progress, organizing regular team meetings and the agenda</li> <li>› Resolving queries, discussing the concepts for the M4</li> </ul> <p><i>As a Team Member (Milestone 2):</i></p> <p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Implementation of find buddy page features</li> <li>› Self-check on best practices for security</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Implementation of sessions</li> <li>› Code Review</li> </ul>
Johnson Nguyen <i>(Backend Lead)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Implementation of sessions</li> <li>› Usability test plan</li> <li>› Code Review</li> </ul>
Roberto Simental <i>(Frontend Lead)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Product summary</li> <li>› Usability test plan</li> <li>› Changes to feedback from M3 (prototype)</li> </ul>
Michael Satumba <i>(Frontend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› -</li> </ul> <p>Partial Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› QA test plan</li> <li>› Changes to feedback from M3 (prototype)</li> </ul>
Eduardo Hernandez <i>(Backend Member)</i>	<p>Full Contribution to following sections:</p> <ul style="list-style-type: none"> <li>› Implementation of the events page</li> </ul> <p>Partial Contribution to following sections:</p>

	‣ -
Zhinan Zhao <i>(Backend Member)</i>	Full Contribution to following sections: ‣ Implementation of the user profile page  Partial Contribution to following sections: ‣ -
Ziming Wang <i>(Frontend Member)</i>	Full Contribution to following sections: ‣ -  Partial Contribution to following sections: ‣ Usability test plan ‣ QA test plan

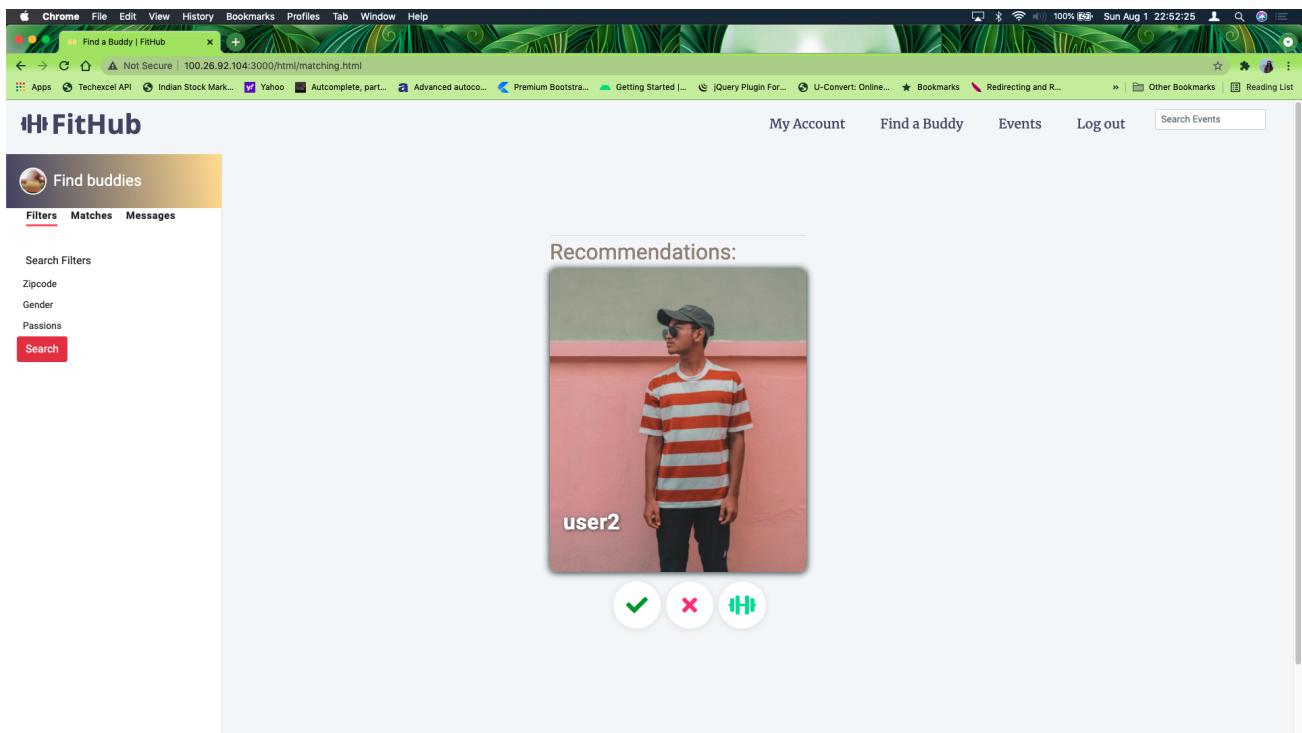
## 4. Screenshots of actual final product

Product Screenshots:

- User Profile Page



- Find a buddy Page



- User messages

My Account Find a Buddy Events Log out Search Events

Find a Buddy | FitHub Not Secure | 100.26.92.104:3000/html/matching.html

Apps TechExcel API Indian Stock Mark... Yahoo Autocomplete, part... Advanced autoc... Premium Bootstrap... Getting Started ... jQuery Plugin For... U-Convert: Online... Bookmarks Redirecting and R...

Sun Aug 1 22:53:28

Hi FitHub

Find buddies

Filters Matches **Messages**

Recent Messages

 user2 Last sent on: 2021-07-31 21:12:01

 Vidhi v Last sent on: 2021-07-31 21:12:08

User2

see you  
2021-07-31 08:50:23

hey it was a great workout  
2021-07-31 17:20:46

yea it was fun see you tomorrow then  
2021-07-31 17:21:45

hi user2  
2021-07-31 21:12:01

Write message... **Send**

## 5. Screen shots of key DB tables (1-2 pages)

- workout request

workout_id	from_user_id	to_user_id	request_status	date_sent	date_updates
1	1	2	3	2021-07-29 13:48:29	2021-07-29 22:21:38
2	1	3	3	2021-07-29 13:48:30	2021-07-30 01:23:51
3	2	1	0	2021-07-29 13:48:30	2021-07-30 01:10:37
4	1	4	3	2021-07-30 01:53:58	2021-07-31 10:19:05
6	1	2	3	2021-07-30 01:03:24	2021-07-30 23:10:49
7	1	13	3	2021-07-30 01:05:23	2021-07-31 00:41:49
8	2	1	1	2021-07-30 01:09:22	2021-07-31 00:42:03
9	1	3	0	2021-07-31 10:18:16	NULL
10	1	2	0	2021-07-31 14:12:01	NULL
11	1	13	0	2021-07-31 14:12:08	NULL
12	17	15	0	2021-08-01 14:05:57	NULL
NULL	NULL	NULL	NULL	NULL	NULL

- user\_messages

id	to_user_id	from_user_id	message	is_read	date_updated
1	2	1	hi..	1	2021-07-30 01:03:24
3	13	1	hi...vidhi...	0	2021-07-30 01:05:23
4	1	2	hi vidhi1	1	2021-07-30 01:09:23
5	2	1	hello....	1	2021-07-31 00:58:21
6	2	1	how are you ?	1	2021-07-31 01:17:54
7	1	2	m good	1	2021-07-31 01:22:17
8	1	2	what about you	1	2021-07-31 01:22:49
9	2	1	nice	1	2021-07-31 01:23:32
10	1	2	lets workout together?	1	2021-07-31 01:35:57
11	2	1	ok sure ..when?	1	2021-07-31 01:36:33
12	1	2	today at 6pm?	1	2021-07-31 01:39:05
13	2	1	ok..location?	1	2021-07-31 01:39:36
14	1	2	gym near the park?	1	2021-07-31 01:42:53
15	2	1	let me check the dista...	1	2021-07-31 01:49:37
16	1	2	sure...np	1	2021-07-31 01:50:05
17	2	1	sounds good to me	1	2021-07-31 01:50:19
18	2	1	see you	1	2021-07-31 01:50:23
19	3	1	hello...	0	2021-07-31 10:18:16
20	2	1	hey it was a great wor...	1	2021-07-31 10:20:46
21	1	2	yea it was fun see you...	1	2021-07-31 10:21:45
22	2	1	hi user2	0	2021-07-31 14:12:01
23	13	1	hi vidhi1	0	2021-07-31 14:12:08
24	15	17	12345123	0	2021-08-01 14:05:57
NULL	NULL	NULL	NULL	NULL	NULL

- user\_activities

<b>id</b>	<b>user_id</b>	<b>user_activities</b>	<b>date_updated</b>
1	1	roller_skating	2021-07-30 00:05:53
2	2	roller_skating	2021-07-30 00:05:53
3	2	skating	2021-07-30 00:05:54
4	2	mma	2021-07-30 00:05:54
5	2	fishing	2021-07-30 00:05:54
6	3	weights	2021-07-30 00:05:54
7	3	cycling	2021-07-30 00:05:54
8	3	basketball	2021-07-30 00:05:54
9	3	calisthenics	2021-07-30 00:05:55
10	3	mma	2021-07-30 00:05:55
11	3	cardio	2021-07-30 00:05:55
12	3	muscle	2021-07-30 00:05:55
13	13	basketball	2021-07-29 23:24:10
14	13	calisthenics	2021-07-29 23:24:10
15	13	wrestling	2021-07-29 23:24:10
16	13	parasailing	2021-07-29 23:24:10
17	13	skiing	2021-07-29 23:24:10
18	13	curling	2021-07-29 23:24:10
19	13	sleding	2021-07-29 23:24:10
20	13	roller_skating	2021-07-29 23:24:10
21	14	archery	2021-07-30 04:39:34
22	15	calisthenics	2021-07-31 12:35:26
23	15	wrestling	2021-07-31 12:35:26
24	15	snowboarding	2021-07-31 12:35:26
25	15	skiing	2021-07-31 12:35:26
26	15	sleding	2021-07-31 12:35:26

- events

## 6. Screen shot(s) of your task management system

The screenshot shows the Asana web interface with the following details:

- Project:** SWE
- View:** List
- Tasks (To do):**
  - M4 - doc write the 1st section (Assignee: Roberto, Due date: 10/08/18)
  - M4 - M3 feedback changes (Assignee: Roberto, Due date: 5/8)
  - M4 - Ads section on the home page (Assignee: Roberto, Due date: 10/08/18)
  - M4 - Meta data on each page (Assignee: Roberto, Due date: 10/08/18)
  - Modify User Information (Assignee: ZhenanZhao, Due date: 10/08/18)
  - Joining, Exiting, Rejoin Events (Assignee: Eduardo Hernandez, Due date: 10/08/18)
  - Sessions (Assignee: Johnson Nguyen, Due date: 10/08/18)
  - Messaging (Assignee: Vidhi Vora, Due date: 10/08/18)
  - Usability testing (Assignee: Roberto, Due date: 10/08/18)
  - QA test plan (Assignee: Michael Satumba, Due date: 10/08/18)
  - Implement Checking and Declining "Find a Buddy" (Assignee: Vidhi Vora, Due date: 10/08/18)
  - QA test plan - 2 (Assignee: Ziming Wang, Due date: 10/08/18)
- Tasks (Doing):**
  - M3 - horizontal prototype (Assignee: Eduardo Hernandez, Due date: 10/08/18)
  - M3 - data definitions (Assignee: Eduardo Hernandez, Due date: 10/08/18)
  - M2 - Data definitions (Assignee: Eduardo Hernandez, Due date: 10/08/18)
  - Milestone 2 (Assignee: Eduardo Hernandez, Due date: Jul 1)
- Buttons and Links:**
  - + Add task
  - All tasks, Filter, Sort, Customize
  - Share, Upgrade
  - Invite teammates, Help & getting started

## 7. Team member contributions

<b>Team Member</b>	<b>Contribution</b>
Vidhi Vora <i>(Team Lead)</i>	<ul style="list-style-type: none"> <li>› Score: 9</li> <li>› Have given my best as a team lead as well as a team member.</li> <li>› Guiding the team members to achieve assigned goals</li> <li>› Task: Github Master, Implementing login, find a buddy section, events (Partial) &amp; Documentation</li> </ul>
Johnson Nguyen <i>(Backend Lead)</i>	<ul style="list-style-type: none"> <li>› Score: 8</li> <li>› He has performed the role of a good backend lead very well with his abilities and qualities</li> <li>› Excellent work on setting up the server and hosting our app</li> <li>› Written scripts to ease the process of pushing code from git to live server</li> <li>› Task: Implementing admin panel, server setup &amp; Documentation</li> </ul>
Roberto Simental <i>(Frontend Lead)</i>	<ul style="list-style-type: none"> <li>› Score: 8</li> <li>› A very good frontend lead as well a team player.</li> <li>› Has always managed to perform his tasks very well.</li> <li>› He has very good writing skills and has always worked on his sections of document as well as helped other to finish theirs</li> <li>› Task: Designing frontend page &amp; Documentation</li> </ul>
Michael Satumba <i>(Frontend Member)</i>	<ul style="list-style-type: none"> <li>› Score: 7</li> <li>› Excellent job in developing the html pages for the web app</li> <li>› Always has a clear idea on what needs to be presented in the application and keep design clear</li> <li>› Task: Designing frontend page &amp; Documentation</li> </ul>
Eduardo Hernandez <i>(Backend Member)</i>	<ul style="list-style-type: none"> <li>› Score: 7</li> <li>› Has improved his communication skills over the semester</li> <li>› He shows the eagerness to learn when is assigned a task</li> <li>› Task: Implementing events (Partial) section &amp; Documentation</li> </ul>
Zhinan Zhao <i>(Backend Member)</i>	<ul style="list-style-type: none"> <li>› Score: 7</li> <li>› I am impressed with the way he performs his task despite minimal to no communication with members</li> <li>› Task: Implementing profile section &amp; Documentation</li> </ul>
Ziming Wang <i>(Frontend Member)</i>	<ul style="list-style-type: none"> <li>› Score: 6</li> <li>› Has good communication skills but needs to work on completing tasks in the assigned deadline.</li> <li>› Task: Designing frontend page (event profile) &amp; Documentation</li> </ul>

## **8. Post analysis – lessons learned**

The few of the major challenges that were being faced while developing the project:

1. Frequent missing of internal deadlines by all team members due to which we were running behind the schedule
2. The team was facing minimum to no communication issue from a few team members
3. For some of the features, it was hard to keep an eye on the bigger picture.
4. Team members were facing a lot of merge and conflict issue over Github while committing their code since we had only two branches to work on (develop & master)

To overcome these challenges, we can take following measures:

1. Imposing strict restrictions to achieve deadlines
2. Making the team meetings more of an opinionated form so everyone needs to interact. This will help the team members to improve their communication and ask more questions to all
3. While selecting a feature to implement, try to think more in detail as to how the flow of the process will be helping us to identify if there are chances of major conflicts at that point
4. For Github, the team lead needs to study more in detail about the requirements of the project and the approach that suits all team members. For eg. creating more branches, either user wise or project module wise