# 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:

Version	Date
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: clubes 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. 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

#### 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. <u>UI Mockups and Storyboards (high level only)</u>

Use Case 1: Finding the Fithub Website

116 ( 1.5 5: 1.	T'IL 1	٨	
Use Case: Finding the	Fithub	A	PP
Youtube 1	1701	ne	Account (Events)
6 min Abs	Find Buld	0	(Account) (Events)
Find Your Buddy	The same	Y	(ECONII)
1888	*		
	1 f. kiva	^	checks out Fithub by
2. Kiven Sed an ad	clicking	la la	on the link of the add,
for Fithub while	From +	he	homepage he can alloss
Watching Workout	events	0	nd see what Fithup has
Videos	to Off		
Event Page			Event Page - Details
	7-1		20
Biking @ Golden Gate Park	Details		Biking @ Gollan Gate
Yoga @ The Beach	Details		·Meeting at 6 am
Zumba, 24 Hour Fitness	Details	>	Jon't be lote!
10.			0 16 61 11
A. k. Ven Picks an event +	nat 1 6 a		X Kiven Shores the event
interests them and click	(s to see		with his friend Kelly by
the details			using the space icon
			and they both register to be able to commant/like
			the event
			THE CYMI)
			10

**Use Case 2: Registration and Login** 

	Use Case: Registration and Login
	Homepage Login/Roister Page
	Too in Email Rob@Mail-Com
	Sign up Username Rob 123
	Password
	F. Rob Clicks on Confirm Password
	Sign up to make & Rob enters his information to
	an account create an account
	Login/Register Page - Profile
	First Name
	Ald Picture Birthday
	CHY
	Hobbies
-	
	f. Rob edits his Profile So he can
	begin using Fithub properly
-	

Use Case 3: Finding a Partner

Use Case: Find a Part Home Page log in	Harry In al 2 Illes
Jim and Jack log in to their account from the homepage	Find Buddy (Account) (Events)  **Tim is looking for a work out bully, so he selects Find Buddy
Find Bully Pag-Workout  Surprise Workout  Weigh Lifting  Out Anov running  Yoga  Reset   Find Bubby  A Jim selects his  preferred workouts	Find Buldy Page - Profiles  [Extra Filters]  O "Jack  T "Location"  Bio  Skip Pair UP  L Jim likes Jack's Profile and  Sends him a Friend (pair up) request
Home Page - logged in vser  (Find Bully) (Account (Eve  2. Jack notices a new pair up request and clicks on it to see it	Account Page - Pair UP Requests  and River Accept  A Jack Accept Bob's  request

**Use Case 4: Posting Events** 

Sign up Find Buddy	-Ingged in vier Ironout)  (Account) (Events)  a Selects Events to go ent Page
1	
Event Page	Post Event Page
Post Event	
	Title 1
loga @ The Beach Details	
12cm ba, 24 hour Fitness Details	Details
0 M	
X. Magdelena doesn't see any	Past
biking events so she will	
Create her own using Post Event	A. Maglelena creates a
	title and details for
Front Para - Tetil	her event
Event Page - Details	O. Ma dala
Biking & Golden Gate Park Meeting at 6 am, don't be late	2. Magdelena can
Treeling at 2 and , bunt to take	now see her event
Comments Likes 1	and Comments and likes from other users
	train other users

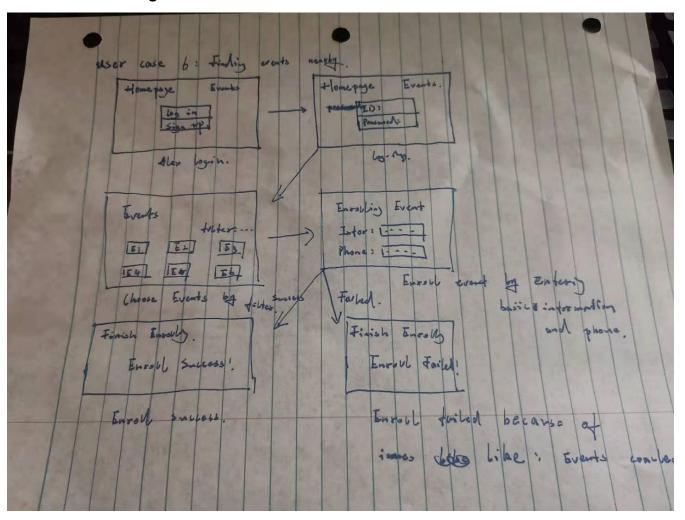
**Use Case 5: Private Chat** 

	Use Case: Private Chat			
	Account Page	FA	Mount Page	- Bullies
	Profile Bullies Chats			
			& Mike	BIOCK
	f. Bob is going to be	X	1. Bio	Message
	late so he is going to		& JACK	BIOCK
	Sent Mike a message		·Bio	Message
	from his Buddies List	]	3 Bob Selects	to mesage Mike
	K			
	Account Page - Chats	Hon	epage - logged	in vour
				NEW MESSAGO
	Mike	Fin	1 Bully Acco	wut (Eveny)
	(A) Hi	K		
	F1			s he has a new
	I'm going to be		message and	1 clicks on it
	late 15 mins	Lto	See from	who
	7. Bob let Mike Know	/		
	be will be late /			
	N 11 5 (1 1)			
	Account Page - Chats			
	Bob I'm going to be			
	1 m going to be			
	Vate 15 mins			
-	(010)			
1	·Mike responds to Bob's			
1	message			
1	1,000,00			

# **Use Case 6: Group Chat**

Use Case: 6 Group	Chat
	Home Page - logged in vier
Homepage los in	(Eins Buddy (Account) (Evants)
Sign up	<b>**</b>
J. Sally logs into her	2. Sally Clicks on Account to
allount	access her personal information
V	
Account	Account - Charts
Profile Buddies (Chots)	Britney
0	See You Soon!
2. Sally Selects Chots	Britney, Bob, Mike
	Bon: running late
	2. Sally selects her group that
	I that includes Britney Bob, Mike
Acces to a Clark	$\prod_{i=1}^{n}$
Account - Chats	Home Page - logged in user
Britney, Bob, Mike (A) Bob (running late)	(Find Buddy (Account) (Events)
1) 1500 Tollying Into	Frank OUDDY) (ICCIONI) (EVENTS)
(Sovry even and	1989 Britue Ruh 1 Miles
Sovry everyone I can't make it.	A & Britney Bob, and Mike all recieve Sally's message
	The sage
f. Sally lets Britney	
Bob, and Mike know	
that she won't show	
up in 1 message	

**Use Case 7: Finding events** 



# Use Case 8: Buddy Pass System

USE Case: Bully Pass Sust	em
Use Case: Buddy Pass Syste Homepage - logged in user	Account
	Try Beddy Pass
(Find Buddy (Account) (Events)	3
9 5 11: 12 1	Profile Bublies Chats
X-Eddie wants to try the	2. Ellie Clicks on Try Beddy
Bully Pass so he goes into his Account to subscribe	Pass
The substitute	
V	
Payment Page	Homepage - lugged in user
T sub	
- Eddie	Find Buddy Account
Eb@ mail.com  Credit Card Number	
MM/YY Cord Security#	2. Eddie can now Find a Buddy
Confirm	With the Buldy Pass that will
0 -11	give him access to a
X. Eddie enters his	hearby gym
payment information	
	4

Bully Pass	
Find Buddy Page - Workout	Find Buddy Page - Profiles
	Bully Pars @ Extra Filters
Surprise Workout	9 . Justin
weight Lifting V	1 Location
Cardib	Gyms: 24 hour Fitness Manet Fitness
Yoga	1 ·Bio
2. Ellie selets a	Skip   Pair Up
workort	2. Eldie can toggle on and off
	Profiles that have or don't have
	gum memberships

**Use Case 9: History Log** 

Use Case: History Log
Use Case: History Log Homepage - Logged in Usar Account Page
History
Find Buddy (Account) Events , Profile Buddies Chot
2. Bob wants to find the 2. Bob can see his account history
last bully he worked out from here
with so he checks his account
4:
Account Page - History
Profile Bellies Chat
Cardio with Alice ()
Weigh lifting with Mike (8)
R. Bob finds Alices profile
in his history and can now
Send her a friend request

Use Case 10: Friends

Use Case: Friends
Homepage - logged in user Account Page
Find Buddy (Account (Events) , Profile Buddies (Charles)
2. Samantha had a great 2. Samantha selects Buddies to
time with Claire so: See a list of people she worked
She goes to Account Out with
Account Page - Bublies Add Friend
2. Claire Options 19 Block
· Bio Message
Z. Samantha souls Claire a
Friend invite So they can
Continue working out together

**Use Case 11: Selecting Interests** 

Homefore    First Bullier Account (Events)   Email     Vernance   Password     Attiniser wants to find   Vernance     Attiniser wants to find   Vernance     Attiniser wants to find   Vernance     Attiniser people to run   Confirm Rassword     And Surf with So she   Attiniser provides her basic     Creates an account   account information     Registration Page - Profile   Homepage - lagged in User     All Picture City   running   Attiniser selects Find     Preferences   Surfing   Attiniser selects Find     Preferences   Yoga   Buddy to find Someone     Attiniser information   Interests     Workouts So Other people   Find Buddy Page - Workout     Can find her   Running   Very     Yoga   Very     Very   Very     Attiniser   Very     Yoga   Very     Very   Very     Attiniser   Very     Very   Very     Attiniser   Very     Yoga   Very     Very   Very     Attiniser   Very     Yoga   Very     Very   Very     Attiniser   Very     Very   Very     Attiniser   Very     Yoga   Very     Very   Very     Attiniser   Very     Very   Very     Attiniser   Very	Use Case: Selecting Interest	5
Email  Vernance  Registration Page - Profile  First Name  Registration Page - Profile  Find Buddy Account  Residuted Find Someone  Residuted Find Page - Workout  Can find her  Running  Surfing  Voga  Veright lifting	Homespal	Registration Page
Registration Page - Profile  Registration Page - Profile  All Picture City Hobbies  References  Preferences  Profile  Resolution Formation  Registration Page - Profile  Registration Page - Profile  Registration Page - Profile  First Name  Registration Page - Profile  First Name  First Name  Registration Page - Profile  First Name  First Name  First Suddy Account  Resolute City Hobbies  Preferences  Profile  Prefere	Sign vp	Engl
At Picture City  Preferences  Preferences  Preferences  Profile  P	TIME DIONIED A-(COUNT) (TACALTA	
Other people to run  and surf with So she  Creates an account  Registration Page - Profile  First Name  Birthpay  Add Picture City  Preferances  Yoga  Funding  Full Buddy  Account  Reself that has her  profile information  and selects har preferred  workouts So Other people  Can find her  Running  Surfing  Running  Surfing  Running  Surfing  Running  Surfing  Running  Surfing  Surfing  Running  Surfing  V  Yoga  V  Y  Yoga  V  Yoga  V  Yoga  V  Yoga  V  Y  Yoga  V  Y  Yoga  V  Y  Yoga  Y  Y  Y  Y  Y  Y  Y  Y  Y  Y  Y  Y  Y	2. Lindsey wants to find	
and surf with So she  Creates an account account information  Registration Page - Profile  First Name  Right Day  All Picture City  Hobbies  Surfing  Running  Full Buddy Account  All Picture City  Preferences  Yoga  Find Buddy fo find Someone  Running  Interests  Lindsey selects Find  Preferences  Profile information  Interests  Loop Find her  Running  Surfing  Voga		Confirm Password
Registration Page - Profile  Registration Page - Profile  Registration Page - Profile  First Name  Birth Day  Hobbies  Freferances  Yoga  Ruddy fo find someone  Lindsey inputs her  profile information  workouts So Other people  Can find her  Running  Voga  Running  Voga  Running  Voga  V		
Registration Page - Profile  (First Name Birth Day Funning > Find Buddy) (Account)  All Picture City running > All Picture City selects Find Preferences > Yoga Buddy to find someone hewelf that has her profile information interests  and Selects has preferred workouts so other people Find Buddy Page - workout Can find her Running > Voga Voga Voga Voga		account information
First Name  Birth Day  All Picture City  Hobbies  Surring  Preferences  Yoga  Buddy fo find someone  Lindsey inputs her  profile information  and selects has preferred  workouts so other people  Can find her  Running  Surring  Yoga  Weight lifting	V	
First Name Birth Day  All Picture City running  Hobbies SurFing Lindsey Selects Find  Preferences Yoga Buddy to find someone Lindsey inputs her profile information and Selects har preferred workouts So Other people  Can find her  Find Buddy Page - workout  Running  SurFing  Yoga  Weight lifting	Registration Page - Profile	Homepage - logged in User
All Picture City trunning > 2. Lindsey Selects Find  Preferences Yoga Buddy to find someone  Lindsey inputs her heuself that has her profile information interests  and selects has preferred workouts so other people Find Buddy Page - workout  Can find her Running Yoga  Weight lifting		
Hobbies Surfing X-Lindsey Selects Find Preferences Yoga Buddy to find Someone  R. Lindsey inputs her howself that has her profile information interests  and selects har preferred workouts So other people Find Buddy Page - workout  Can find her Running V  Surfing V  Yoga Weight lifting	(A) Birth Day	(Find Buddy) (Account)
Preferences Yoga Buddy to find Someone  2. Lindsey inputs her hewself that has her profile information interests  and selects has preferred workouts so other people Find Buddy Page - workout  Can find her Running V  Surfing V  Yoga V  Weight lifting		
Find Selects has preferred workouts so Other people Find Buddy Page - workout  Can find her Running V  Surfing V  Yoga V  Weight lifting		
profile information  and Selects has preferred  workouts So Other people  Can find her  Running  Yoga  Weight lifting		
and Selects has preferred  workouts so other people  Can find her  Running  Surfing  Yoga  Weight lifting	I. Lindsey inputs her	
workouts so other people Find Buddy Page -workout  Can find her  Running  Surfing  Yoga  Weight lifting		interests
Can find her  Running  Surfing  Yoga  Weight lifting		T 1 2 11 2 11 4 1
Running  Surfing  Yoga  Weight lifting		Find Buddy Page - WOVROUT
Surfing V Yoga V Weight lifting	Can find her	IP: 1/
[ Weight lifting ]		Nonthing
[ Weight lifting ]		You
IN PROPERTY OF THE PROPERTY OF		A ringel server has interests

# 4. High level database architecture and organization

#### • <u>DB organization:</u>

#### 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

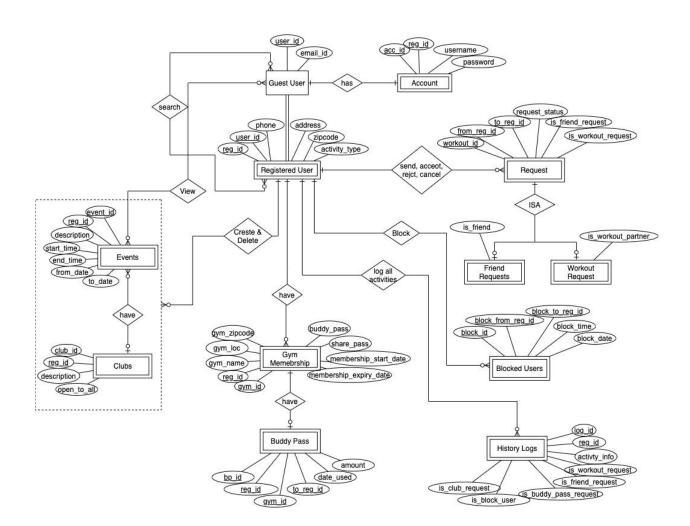
#### j. Friend request (Weak)

- 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

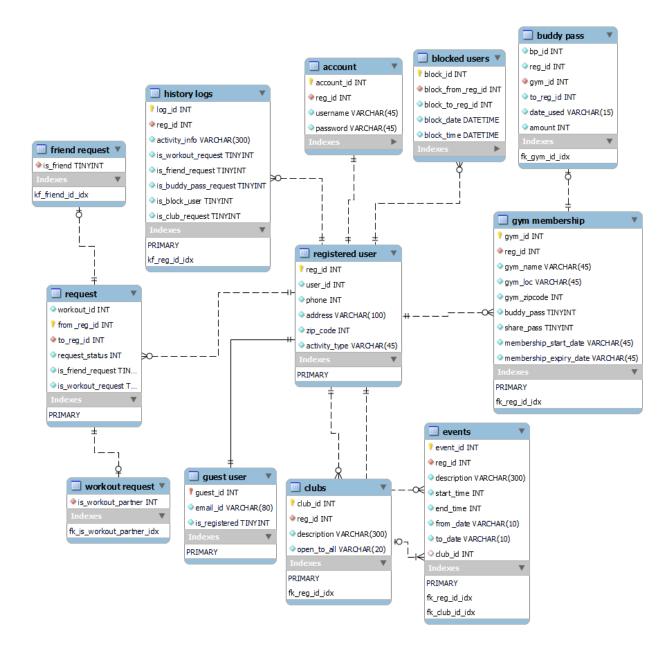
#### k. Workout request (Weak)

- 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 FItHub 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.

## • <u>Search/filter architecture and implementation</u>:

#### 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.is File
- 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:

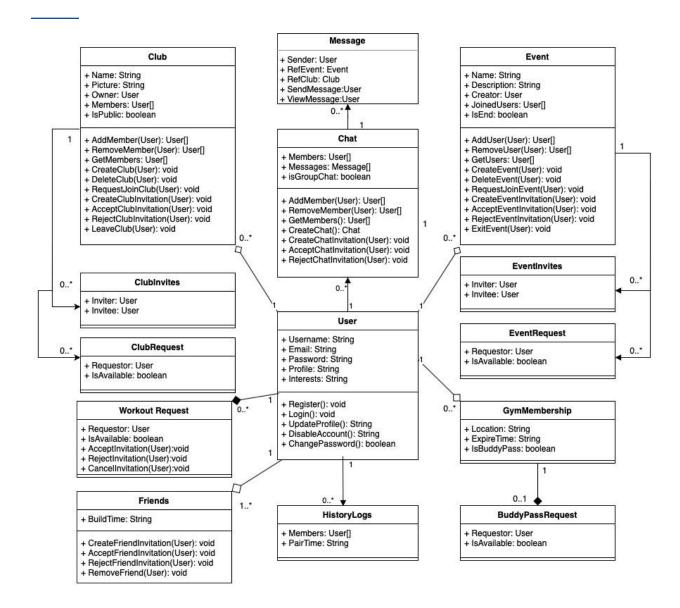
a. When a user decides to post an event, the API will receive a post request. we would receive a post request with the user's information that they just inputted. This will send the data to our database which will store all of the data

# v. Search by Filter (INTERNAL API)

#### 1. Get request

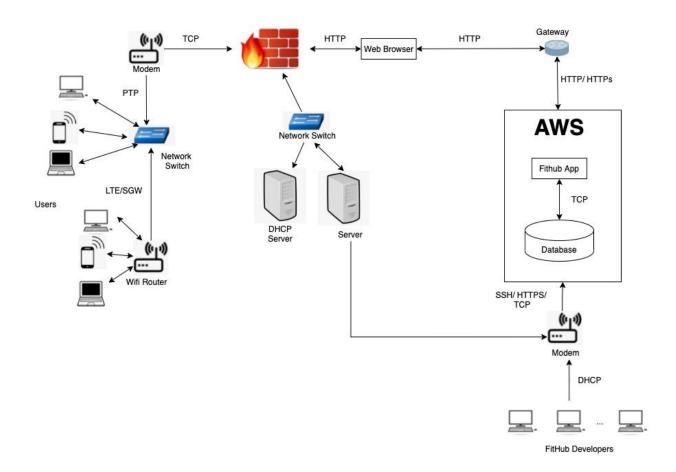
a. When users search by filter, the backend will receive a Get request. This Get Request will then go into our database where every data is stored, and it will send back the necessary information. In example, a filter the user has selected.

# 6. <u>High Level UML Diagrams</u>

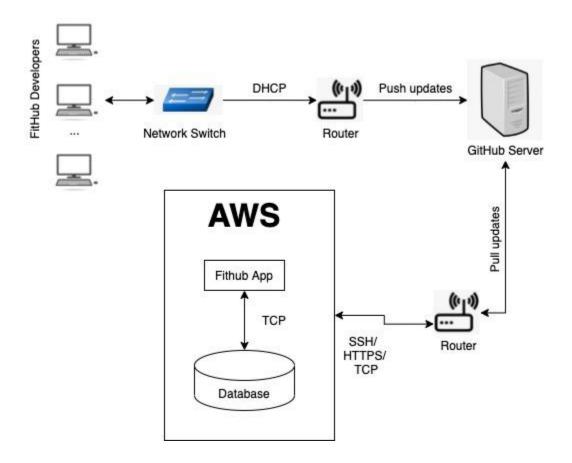


# 7. <u>High Level Application Network and Deployment Diagrams</u>

# **Application Network Diagram:**



# **Deployment Diagram:**



# 8. <u>Identify actual key risks for your project at this time</u>

# 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. <u>Technical risk: Map Integration</u>

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. <u>Detailed list of contributions (this section must be done by the team lead)</u>

Team Member	Contribution
Vidhi Vora (Team Lead)	<ul> <li>As a Team Lead:         <ul> <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 querries, discussing the concepts for the M2</li> </ul> </li> </ul>
	As a Team Member (Milestone 2):  Full Contribution to following sections:  Project management  Detailed list of contributions
	Partial Contribution to following sections:  Prioritized Functional Requirements  High level database architecture and organization  High Level Application Network and Deployment Diagrams
	<ul><li>Vertical Prototype:</li><li>Setting up the node express (api.js)</li><li>Connecting to the database</li></ul>
Johnson Nguyen (Backend Lead)	<ul> <li>Milestone 2:</li> <li>Full Contribution to following sections: <ul> <li>High Level APIs and Main Algorithms</li> </ul> </li> <li>Partial Contribution to following sections: <ul> <li>-</li> </ul> </li> <li>Vertical Prototype: <ul> <li>Developing and inserting data into the database for the registration page</li> </ul> </li> </ul>
Roberto Simental (Frontend Lead)	Milestone 2:

	,
	Full Contribution to following sections:  UI Mockups and Storyboards Identify actual key risks for your project at this  Partial Contribution to following sections:
	Vertical Prototype:
Michael Satumba	Milestone 2:
(Frontend Member)	Full Contribution to following sections:
	Partial Contribution to following sections:
	<ul><li>Data definitions</li><li>UI Mockups and Storyboards</li></ul>
	Vertical Prototype:  → Developing the home page and the registration page
Eduardo Hernandez (Backend Member)	Milestone 2:
	Full Contribution to following sections:
	Partial Contribution to following sections:  Data definitions  High level database architecture and organization
	<ul> <li>Vertical Prototype:         <ul> <li>Developing and inserting data into the database for the registration page</li> </ul> </li> </ul>
Zhinan Zhao (Backend Member)	Milestone 2:
	Full Contribution to following sections:  High Level UML Diagrams
	Partial Contribution to following sections:  High Level Application Network and Deployment Diagrams
	Vertical Prototype:

	<b>&gt;</b> -
Ziming Wang (Frontend Member)	Milestone 2:  Partial Contribution to following sections:  Prioritized Functional Requirements  Vertical Prototype: