

## **SW Engineering CSC648-848-05 Summer 2024**

# **TeamMate**

## **Personnel Management**

### **Team 05 - Concat**

<b>Name</b>	<b>E-Mail</b>	<b>Role</b>
Tharun Krishna	tkrishna1@sfsu.edu	Team Lead
Tyron Cheung	tcheung@sfsu.edu	Frontend Lead
Naw San	lsan@sfsu.edu	Frontend Dev
Randolf Gabrielle Uy	ruy1@sfsu.edu	Backend Lead
Yasemin Sucu	ysucu@sfsu.edu	Backend Dev
Carlos Hernandez	chernandez41@sfsu.edu	Database Lead
Xuefeng Guan	xguan6@mail.sfsu.edu	Database Dev
Jannelly Hernandez Umanzor	jhernandezumanzor@mail.sfsu.edu	Github Master

**Milestone 2**  
**July 9 2024**

#### **Version History**

<b>Revision</b>	<b>Date Submitted</b>
Version 1	9 July 2024

# Table of Contents

<u>Data Definitions</u>	3
<u>Prioritized Functional Requirements</u>	7
Priority 1	7
Priority 2	10
Priority 3	11
<u>UI Mockups and Storyboards</u>	12
Use Case #1	12
Use Case #2	21
Use Case #3	27
Use Case #4	31
Use Case #5	35
Use Case #6	38
Use Case #7	42
Use Case #8	45
Use Case #9	48
Use Case #10	51
Use Case #11	54
Use Case #12	57
<u>High Level Database Architecture</u>	60
DBMS Summary	60
Functional Requirements	60
ERD (Entity Relationship Diagram)	64
EER (Enhanced Entity-Relationship)	65
Media Storage:	66
<u>High Level APIs and Main Algorithms</u>	67
<u>System Design</u>	69
<u>High Level Application Network and Deployment Design</u>	72
<u>Actual Risks</u>	74
<u>Project Management</u>	75
<u>Team Contributions</u>	76

# Data Definitions

- Registered Users
  - Email
    - Correct format with @ and .com
  - Password
    - Modern standards with letters, numbers, special characters.
    - Stored in the database after being hashed.
  - Profile Picture
    - PNG or JPG/JPEG.
    - 400x400 as the max dimensions.
    - No greater than 2 megabytes.
  - Username
  - First Name
  - Last Name
- Organizations
  - Name
    - Plaintext title.
    - Must be unique between all organizations.
  - Owner
    - Connected to a registered user.
    - Transferable to another registered user.
  - Description
- Members
  - Permissions
    - Either 0, 1, or 2 for user, group leader, and admin respectively.
  - Organization
    - Connected to an organization.
  - Group
    - Connected to a group.
  - User
    - Connected to a registered user.
  - First Name
    - Different from the Registered User's name.
  - Last Name
    - Different from the Registered User's name.
  - Sort Order

- Integer that controls in which order members should appear on the frontend.
- Last Login
  - Datetime that represents the last time this member was logged-in..
- Active Member
  - Boolean to archive members that are no longer active for historical purposes.
- Groups
  - Organization
    - Connected to an organization.
  - Sort Order
    - Integer that controls in which order groups should appear on the frontend.
  - Group Image
    - PNG or JPG/JPEG.
    - 400x400 as the max dimensions.
    - No greater than 2 megabytes.
  - Group Name
- Events
  - Organization
    - Connected to an organization.
  - Title
  - Description
  - Date
  - Type
    - Simple string for users' organizational purposes.
  - Prerequisites
    - Serialized string that contains group and qualification prerequisites.
- Event Reminder
  - Event
    - Connected to an event.
  - Member
    - Connected to a member.
  - Attendance
    - Enumerated type that records if the member was present or absent.

- Qualification Type
  - Title
  - Description
  - Image
    - PNG or JPG/JPEG.
    - 400x400 as the max dimensions.
    - No greater than 2 megabytes.
- Qualification Instance
  - Issued By
    - Connect to a member.
  - Received By
    - Connect to a member.
  - Qualification Type
    - Connect to a qualification type.
  - Nomination
    - Boolean that determines whether this instance has been issued or is just a nomination.
  - Date Given
    - Datetime.
  - Citation
    - Plaintext citation for issuance.
- Award Type
  - Title
  - Description
  - Image
    - PNG or JPG/JPEG.
    - 400x400 as the max dimensions.
    - No greater than 2 megabytes.
- Award Instance
  - Issued By
    - Connect to a member.
  - Received By
    - Connect to a member.
  - Qualification Type
    - Connect to a qualification type.
  - Nomination
    - Boolean that determines whether this instance has been issued or is just a nomination.

- Date Given
  - Datetime.
- Citation
  - Plaintext citation for issuance.
- Comments
  - UserID
    - Connect to a member.
  - Page Type
    - Determines whether the comment is on an award instance, qualification instance, event, or another member.
  - Page Key
    - Determines the index/ID of where the comment is.
  - Comment
    - Plaintext comment body.
  - Comment Date
    - Datetime.
- Custom Section
  - Access Level
    - Determines whether the section is for all users, group leaders, or admins.
  - Plain Text
    - Plaintext body for the custom section.
  - Third Party
    - Open-ended field for code or other integrations.
  - Document Creator
    - Connected to a member.
  - Organization
    - Connected to an organization.
  - Title

# Prioritized Functional Requirements

## Priority 1

### Account

- Users shall register for a new account.
- Accounts shall have unique emails.
- An account shall have a username.
- An account shall have a password.
- An account shall have secure passwords as per modern standards.
- Users shall log into an existing account.
- An account shall be able to update profile information.
- An account shall be able to update the password.
- An account shall be able to update the associated email.
- An account shall create a new organization.
- Accounts shall join an existing organization.
- Accounts shall view all organizations they have joined.
- Accounts shall view all public organizations another account has joined.

### Member

- Member permissions shall be members, group leaders, or admins.
- Admins shall assign member and group leader permissions.
- Admins shall be able to issue awards to members.
- Admins reset member names if they are inappropriate.
- Admins reset member profile pictures if they are inappropriate.
- Users shall be able to filter for members that have specific attributes.
- A member's page shall show their information for that organization, along with all attributes, qualifications, awards, etc as a dashboard.

### Organization

- An organization shall have an owner.
- An owner can assign other members to be admins.
- An organization shall have permissions that determine what members, group leaders, and admins are able to create and edit.

- An organization shall have a roster page that shows all members in the organization separated by groups.
- Admins shall be able to change the order in which groups appear on the organization roster page.
- Users shall click on any member on the roster to access their member page.

## **Awards**

- Admins shall create award types.
- Admins shall update award types.
- Admins shall delete award types.
- Awards shall have a title.
- Awards shall have a description.
- Group leaders shall nominate members for awards.
- Admins shall view nominations.
- Admins shall approve or reject nominations.
- Admins shall directly assign awards.
- Members shall view individual award citations.
- Award instances shall have a citation describing the reason for issue.
- Award instances shall have a date received.
- Award instances shall state the issuing member.
- Users shall be able to leave comments on an award instance.

## **Qualifications**

- Admins shall create qualifications types.
- Admins shall update qualifications types.
- Admins shall delete qualifications types.
- Qualifications shall have a title.
- Qualifications shall have a description.
- Group leaders shall nominate members for qualifications.
- Admins shall view nominations.
- Admins shall approve or reject nominations.
- Admins shall directly assign qualifications.
- Members shall view individual qualifications.

- Qualification instances shall have a date received.
- Qualification instances shall state the issuing member.

## **Events**

- Admins shall create events.
- Group leaders shall create events.
- Events shall be public, by request, or invite only.
- Admins shall assign members to an existing event.
- Group leaders shall assign members to an existing event.
- Members shall join an event if it is public.
- Members shall request to join an event if it is by request.
- Events shall have a title.
- Events shall have a description.
- Events shall have a start datetime.

## Priority 2

### **Account**

- An account shall be able to upload a profile picture.

### **Member**

- Members shall set a separate first and last name in each organization.
- Members shall generate event attendance reports.
- Admins shall be able to upload an image for the organization profile.
- A member's page shall show the last time they were logged-in.

### **Organization**

- Organization ownership shall be able to be transferred to another member.
- Admins shall create custom sections for tailored purposes.
- Organizations shall include an inactive or past members group that archives past members for posterity.

### **Awards**

- Awards shall have an associated image.
- Users shall have the option to upload an associated pdf file for display with an award.

### **Qualifications**

- Qualifications shall have an associated image.
- Users shall have the option to upload an associated pdf file for display with a qualification.

### **Events**

- Events shall be recurring if desired, automatically creating new events.
- Events shall be able to have pre-requisite qualifications.
- Events shall be able to be restricted to chosen groups or roles.
- Users shall be able to see an attendance record for recurring events.

### **Custom Section**

- Admins shall create custom sections.
- Custom sections shall have an editable title.
- Admins shall restrict custom sections based on member attributes.
- Custom sections shall be deleted by admins if desired.

## **Priority 3**

### **Member**

- Users shall be able to leave comments and endorsements on a member's page.

### **Organization**

- Admins shall link external apps and embedded apps in custom pages.

### **Events**

- Users shall have the option to opt-in for event reminders.
- Users shall be able to leave comments on an event instance.

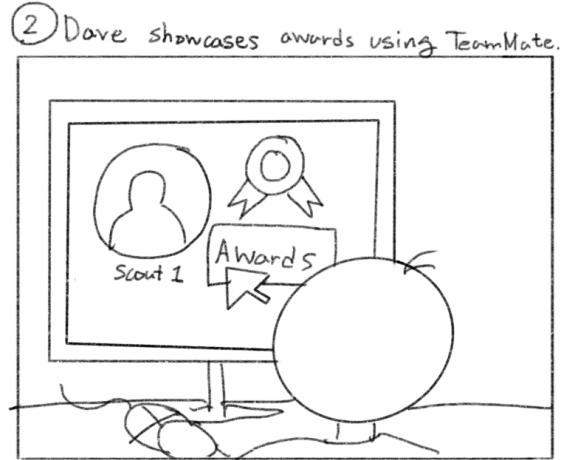
### **Custom Section**

- Custom sections shall support external application, APIs, and embedded application integrations.

# UI Mockups and Storyboards

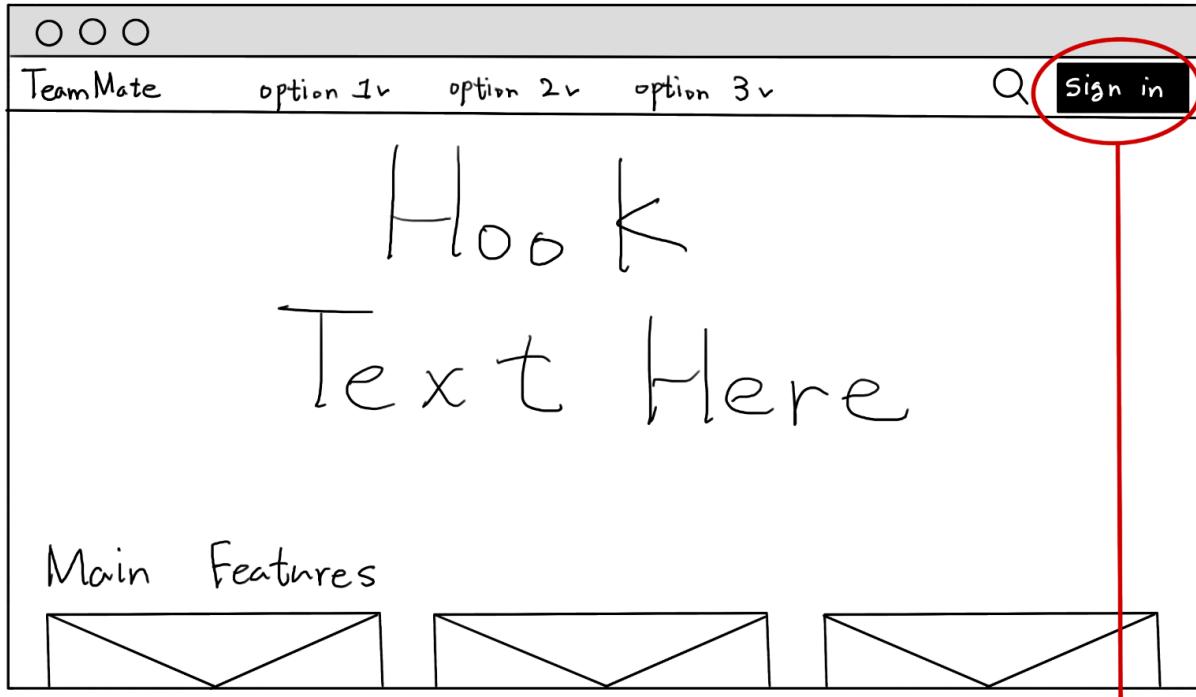
## Use Case #1

Dave, a leader for the Boy Scouts of America in Silicon Valley, faced challenges during COVID as scouts were unable to participate in outdoor activities. To maintain motivation, he created an organization on TeamMate to display scout progress and achievements online. With the help of his friend Chris, they managed the scouts' accounts, groups, and awards, making the process more efficient and ensuring scouts could still take pride in their accomplishments.

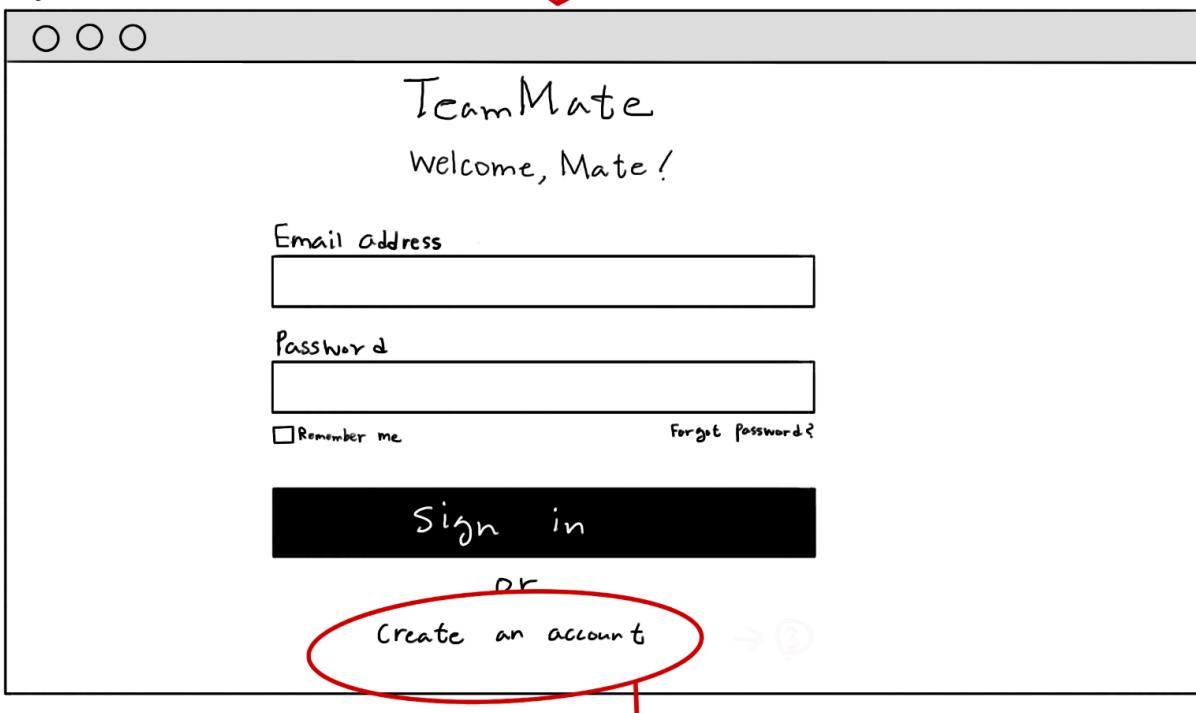


① Start - Leader's perspective  
Home Page

use case # 1



② Sign In Page



③

### Create Account Page

○○○

## TeamMate

Create an account

First name  Last name

Email

Password

Confirm Password

I agree to the Terms of Service and Privacy Policy

**Create account**

Already have an account? [Sign in](#)

④

### Dashboard Page

○○○

Organization

+ [Create](#)

 [Find](#)

Dash Board

 + 

Create or join an organization to get started



⑤

### Dashboard Page

The screenshot shows a dark-themed dashboard interface. On the left, there's a sidebar with 'Organization' buttons for '+ Create' and 'Find'. The main area has a 'Dash' title and a 'Create organization' form. The form fields are 'Organization name' (Boy Scouts) and 'Type of organization' (Other). A large red circle highlights the 'Create' button at the bottom.

⑥

### Organization Dashboard Page

The screenshot shows the 'Boy Scouts' organization dashboard. The sidebar includes '+ Create' and 'Boy Scouts' buttons, and a 'Find' button at the bottom. The main area displays organization details ('Boy Scouts') and several cards: 'Members', 'Groups' (which is circled in red), 'Events', 'Task/Project', 'Recognition', and 'Highlights'. A red arrow points from the 'Create' button in the first screenshot down to the 'Groups' card here.

⑦ Groups Page

Groups Page wireframe. The interface includes a top navigation bar with tabs: Members, Groups (underlined), Events, Tasks/Projects, Recognition, and Highlights. A sidebar on the left shows an organization structure with a 'Create' button and a 'Boy Scouts' group entry. A search bar at the top right has a magnifying glass icon and a plus sign icon, which is circled in red.

No groups created.

⑧ Group Create Page

Group Create Page wireframe. This page is overlaid on the Groups Page. It features a 'Create a group' header with a close button. The form fields include 'Group name' (containing 'Team 1'), 'Description (3/200)' (containing 'abc'), 'Access' (set to 'Invite only'), 'Member limit' (set to 'N/A'), and a large 'Create' button at the bottom, which is circled in red.

9

### Group Detail Page

Organization + Create B Boy Scouts Find

Members Groups Events Tasks/Projects Recognition Highlights

← Groups

Team 1 Edit Add member

Invite Only | 0 members | N/A limit

Description abc

Members No group members yet.

Share

10

### Recognition Page

Organization + Create B Boy Scouts Find

Members Groups Events Tasks/Projects Recognition Highlights

Recognition

No recognition created.

(11)

### Recognition Create Page

Organization  
+  
Create  
B  
Boy Scouts  
Find

Members Groups Events Tasks/Projects Recognition Highlights

Create a recognition X

Recognition name  
Award 1

Recognition type  
Award ✓

Create

(12)

### Recognition Detail Page

Organization  
+  
Create  
B  
Boy Scouts  
Find

Members Groups Events Tasks/Projects Recognition Highlights

← Recognition

Award 1 Edit Add member

Date : 6/27/2024

Description  
abc

Members  
No group members yet.

Share

(13)

Member's Page

Organization	Members	Groups	Events	Tasks/Projects	Recognition	Highlights
+ Create B Boy Scouts	Members	All groups	Roles	Etc.	✓	Q +
Name	Groups	Roles	Etc.			
Chris	N/A	Member	N/A	...		
Dave (You)	N/A	Leader	N/A	...		
Scout3	N/A	Member	N/A	...		
Scout4	N/A	Member	N/A	...		

(14)

Member's Profile Page

Organization	Members	Groups	Events	Tasks/Projects	Recognition	Highlights
+ Create B Boy Scouts	← Members					
Chris						
Group : N/A						
Role : Member						
Etc. : N/A						
Recognition						
No recognition.						

⑯

Member's Profile Edit Page

Organization + Create B Boy Scouts Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Chris

Group : N/A

Role : Member

Etc. : Admin

Recognition

Cancel Save

⑯ End

Member's Profile Page

Organization + Create B Boy Scouts Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Chris

Group : N/A

Role : Admin

Etc. : N/A

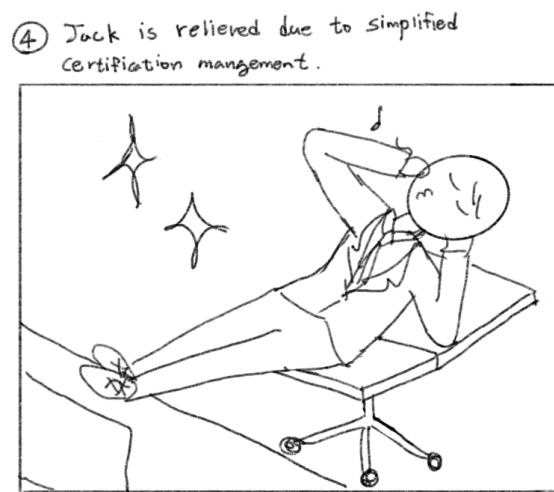
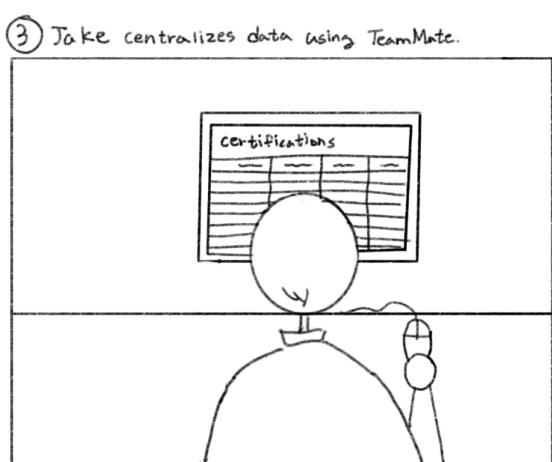
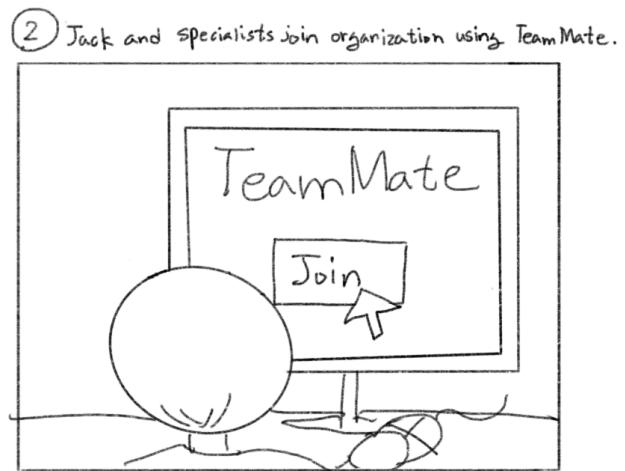
Recognition

No recognition.

Edit

## Use Case #2

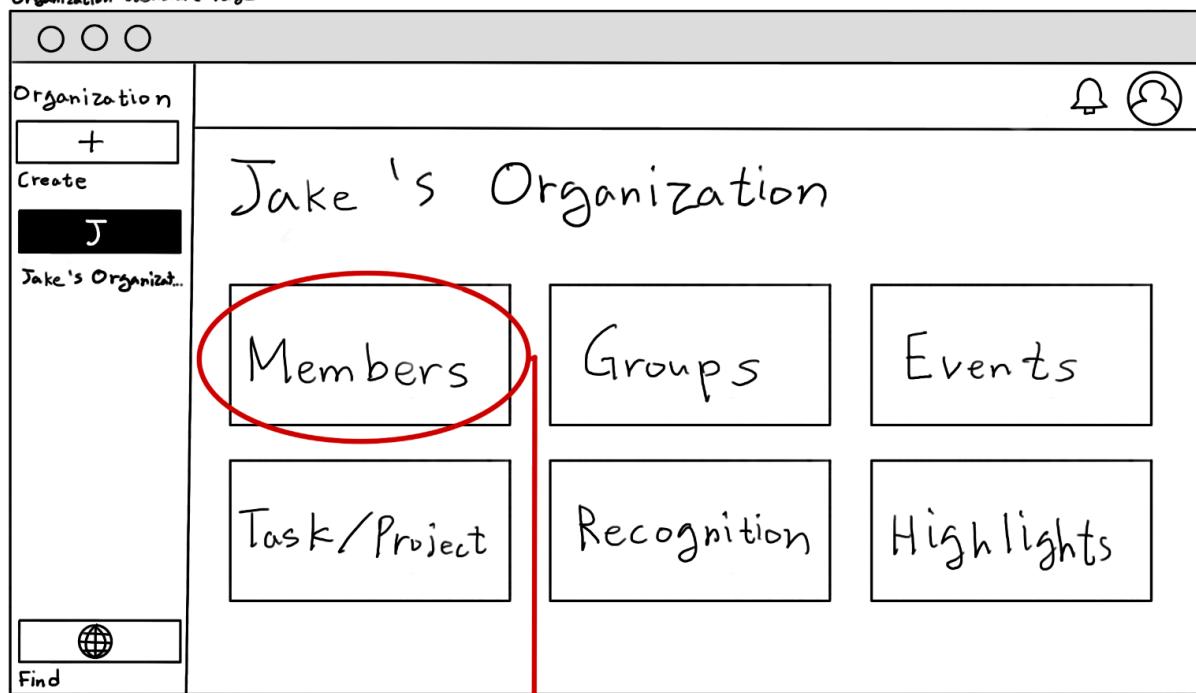
Jake, a leader of a professional group requiring numerous qualifications, found it tedious to track his subordinates' certifications across various platforms. To improve productivity, he created a TeamMate account and organization, where his specialists joined and had their qualifications imported into a centralized system. This allowed Jake to easily manage and update their qualifications, schedule training, and see attendance, simplifying the process and reducing the need to track multiple systems.



① Start - Leader Perspective

Use Case # 2

Organization Dashboard Page



②

Members' Page

The diagram shows the 'Members' Page for 'Jake's Organization'. The top navigation bar includes links for Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. The main content area shows a table of members:

Name	Groups	Roles	Etc.
Jake (You)	N/A	Leader	N/A
Specialist 1	N/A	Member	N/A
Specialist 2	N/A	Member	N/A
Specialist 3	N/A	Member	N/A

(3)

Member's Profile Page

○ ○ ○

Organization + Create J Jake's Organization Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Specialist 1 Edit

Group : N/A  
Role : Member  
Etc. : N/A

Recognition  
No recognition.

(4)

Member's Profile Edit Page

○ ○ ○

Organization + Create J Jake's Organization Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Specialist 1 Cancel Save

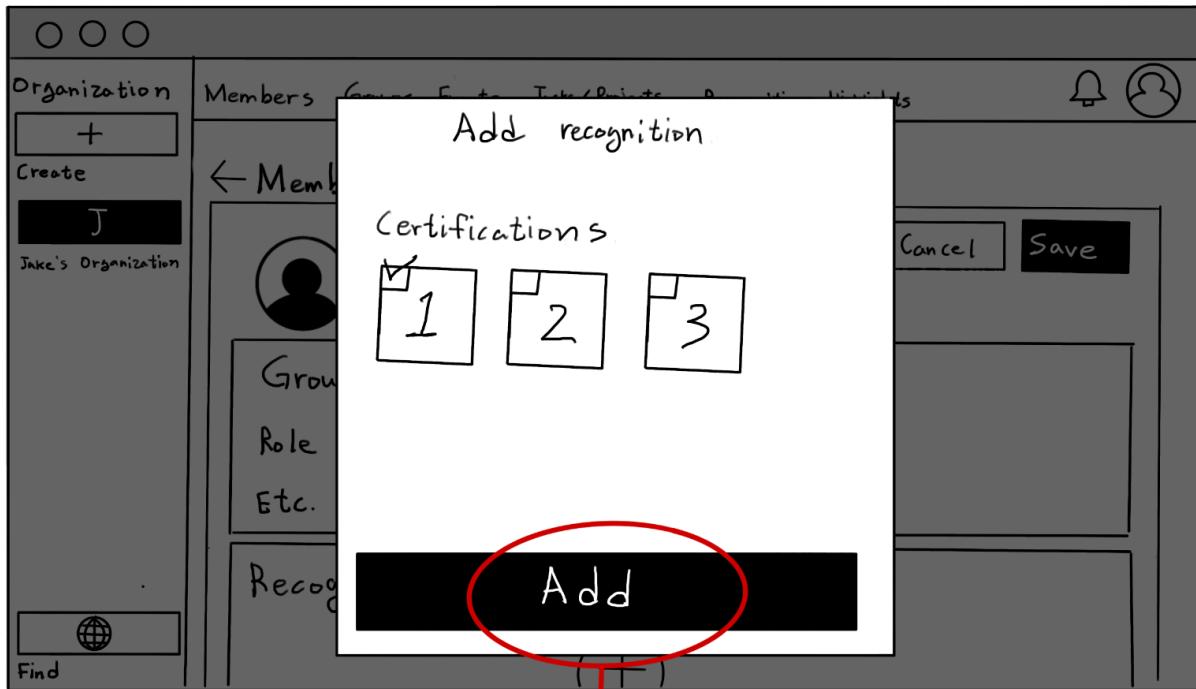
Group : N/A ▾  
Role : Member ▾  
Etc. : N/A ▾

Recognition

+

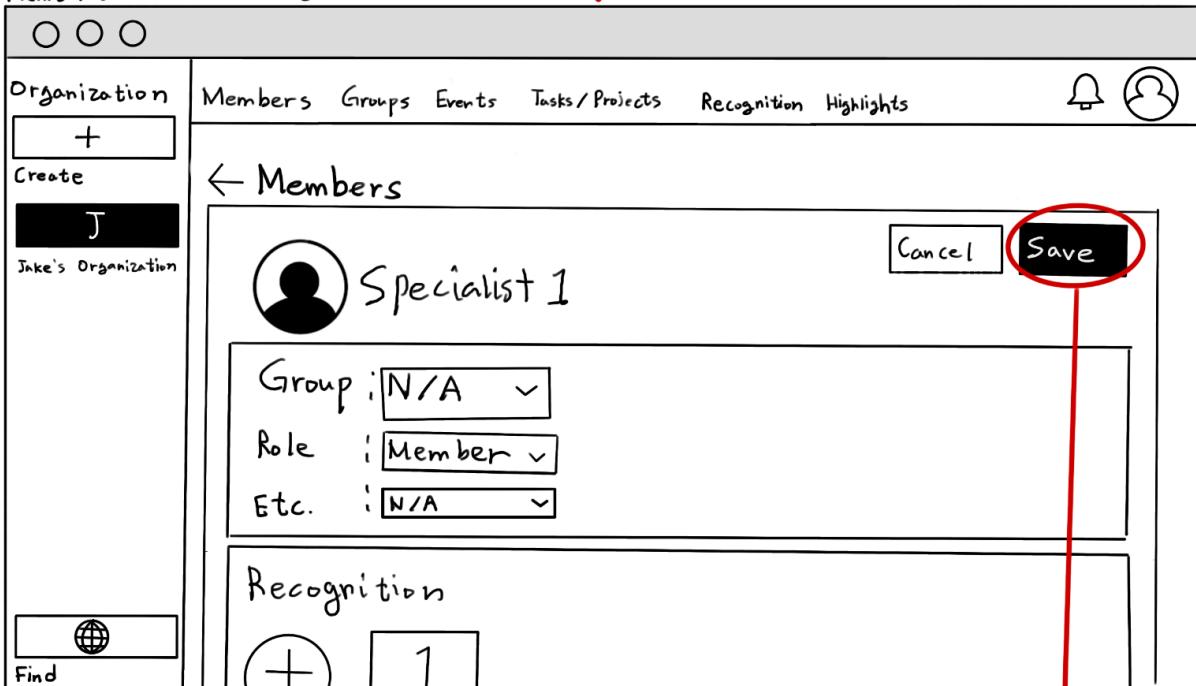
⑤

Member's Profile Edit Page



⑥

Member's Profile Edit Page



⑦

Member's Profile Page

Organization + Create J Jake's Organization Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Specialist 1 Edit

Group: N/A  
Role: Member  
Etc.: N/A

Recognition

Certification 1

⑧

Recognition Dashboard Page

Organization + Create J Jake's Organization Find

Members Groups Events Tasks/Projects Recognition Highlights

Recognition

Certifications 1 2 3

⑨End

Recognition Detail Page

The wireframe depicts a user interface for a recognition detail page. On the left, there's a sidebar with three circular icons at the top, followed by a section titled "Organization". Inside this section is a button with a plus sign and the word "Create", and below it is a dark rectangular button with a white letter "J". To the right of the "Create" button, the text "Jake's Organization" is written. At the bottom of this sidebar is a "Find" button with a magnifying glass icon. The main content area has a header with tabs: "Members", "Groups", "Events", "Tasks/Projects", "Recognition", and "Highlights". To the right of the tabs are a bell icon and a user profile icon. Below the header, a left arrow points to the word "Recognition". The main content box contains a title "Certification 1", an "Edit" button, and an "Add member" button. It also includes a "Description" field containing the text "abc", a "Members" field containing "Specialist 1", and a "Share" button.

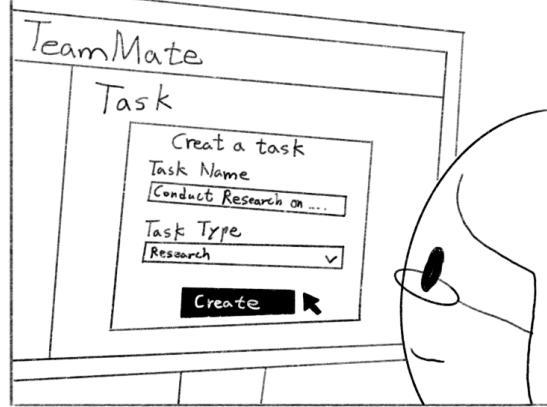
### Use Case #3

Dr. Sarah uses the TeamMate platform to manage a new research project by inputting its details, objectives, timelines, and assigned researchers. Researchers, lab assistants, and graduate students log their progress and update milestones regularly. Dr. Sarah reviews these updates, provides feedback, and adjusts timelines as needed, while the system generates a comprehensive progress report summarizing all ongoing research projects.

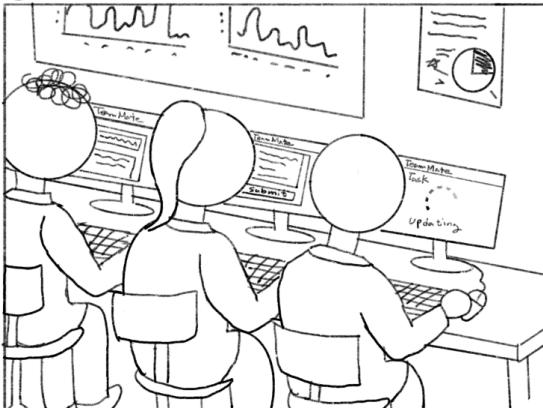
① Dr. Sarah ensures careful planning in the lab.



② She regularly uses TeamMate for project management.



③ Her researchers log progress on TeamMate.



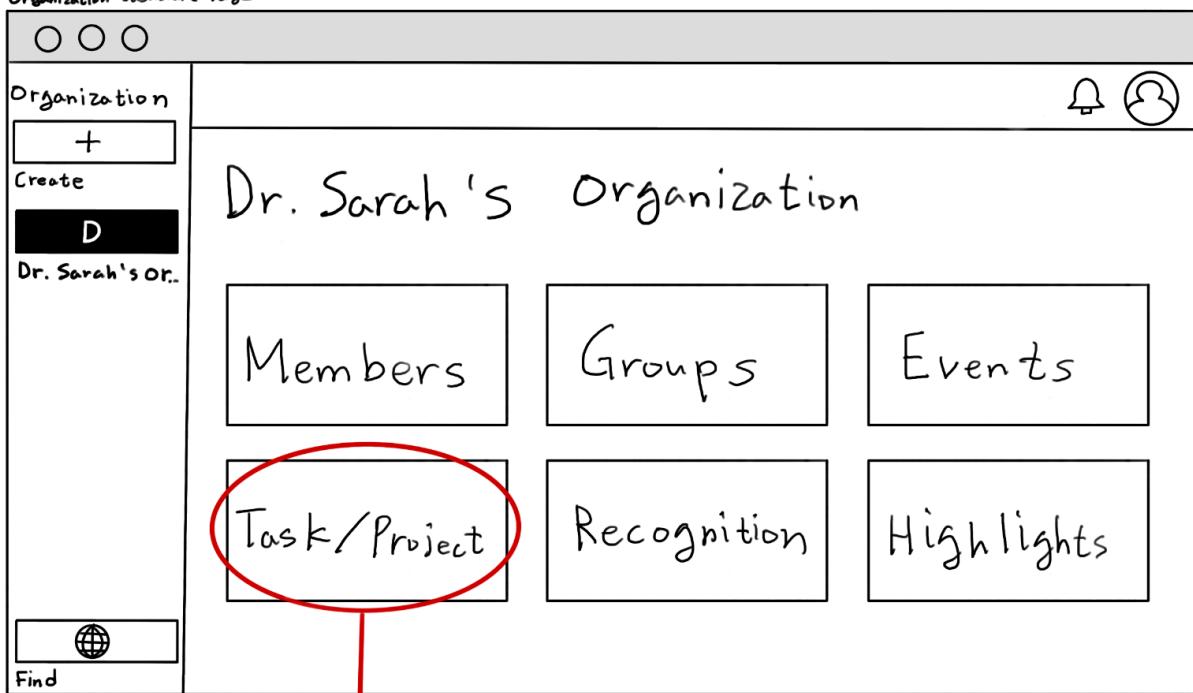
④ Dr. Sarah stays updated on research status.



① Start - Leader Perspective

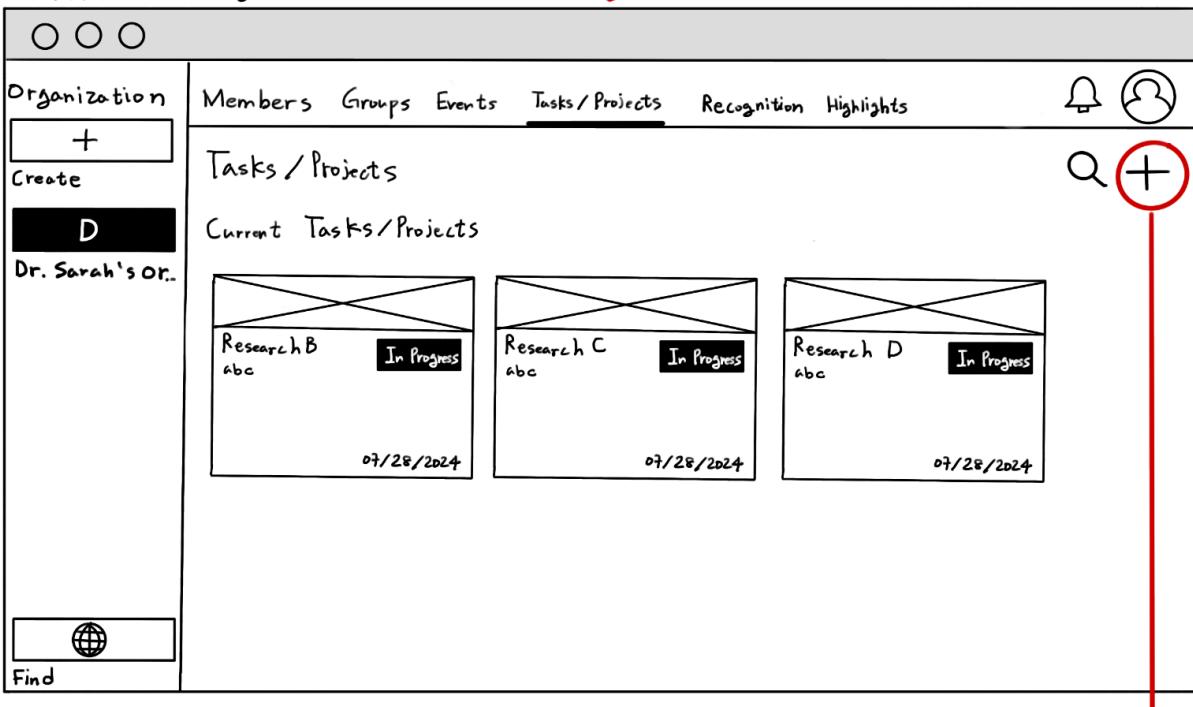
Use case #3

Organization Dashboard Page



②

Tasks / Projects Page



(3)

### Task / Project Create Page

○ ○ ○

Organization + Create D Dr. Sarah's Or.. Find

Members Groups Events Tasks / Projects Recognition Highlights

← Tasks / Projects

Create a task / project + Add recognition

Name : Research A

Description : abc

Upload Content : N/A ↑, Upload file

Assigned to :  Researcher 1  Researcher 2  Researcher 3

Deadline :

**Create**

A red circle highlights the "Create" button.

(4)

### Tasks / Projects Page

○ ○ ○

Organization + Create D Dr. Sarah's Or.. Find

Members Groups Events Tasks / Projects Recognition Highlights

Tasks / Projects

Current Tasks / Projects

Research A abc 07/28/2024	Research B abc In Progress 07/28/2024	Research C abc In Progress 07/28/2024
Research D abc 07/28/2024		

A red circle highlights the "Research B" card.



⑤

Task / Project View Page

○ ○ ○

Organization Members Groups Events Tasks / Projects Recognition Highlights

+  

Create ← Tasks / Projects

D Dr. Sarah's Or..

Research B

Deadline : 7/28/2024 | Submissions 2/10

Description : abc

Content : N/A

Submissions :

Name	Submitted date	Etc.
Researcher 1	07/26/2024	N/A
Researcher 2	07/27/2024	N/A

[View](#) 

[View](#)

Find

⑥ End

Task / Project Submission View Page

○ ○ ○

Organization Members Groups Events Tasks / Projects Recognition Highlights

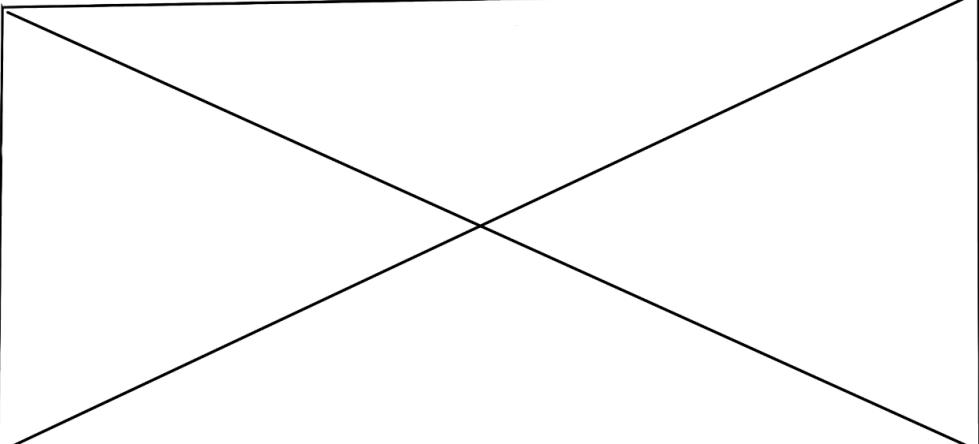
+  

Create ← Research B

D Dr. Sarah's Or..

<  Researcher 1 >

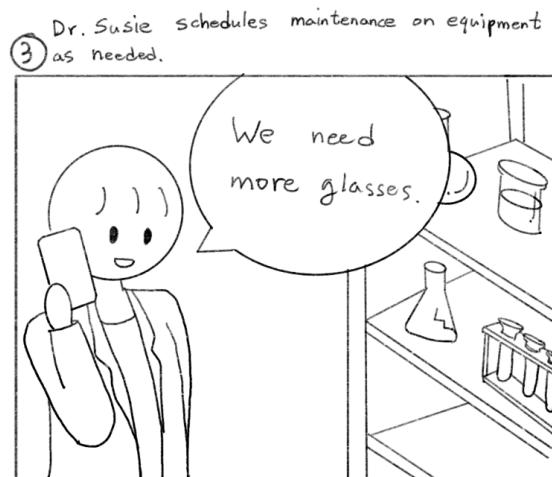
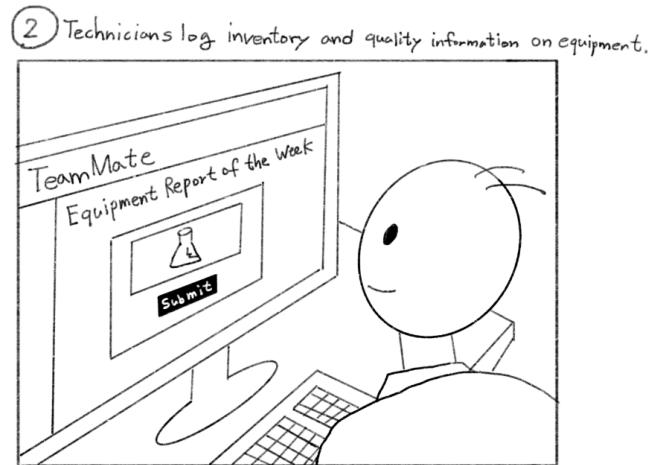
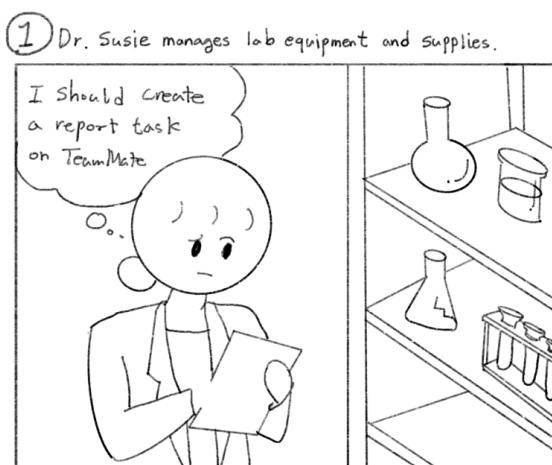
[Revise](#) [Complete](#)



Find

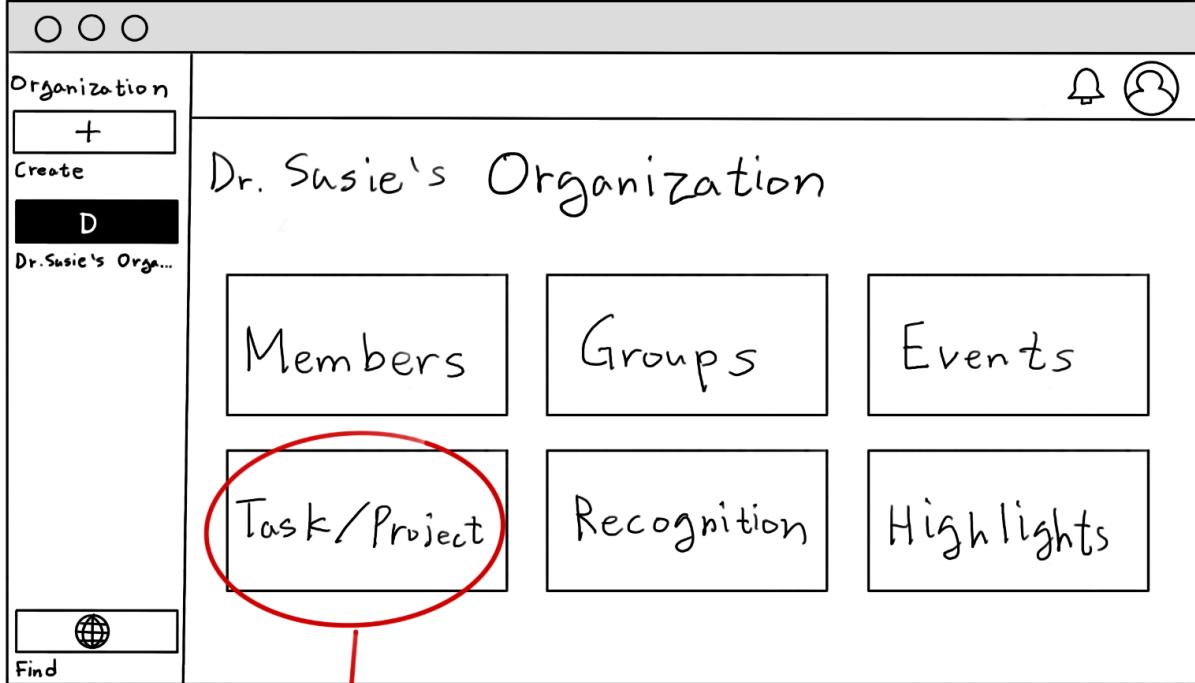
## Use Case #4

Dr. Susie uses the TeamMate platform's "Equipment and Supplies Management" section to input details about lab equipment and supplies, including quantities, locations, and maintenance schedules. Lab technicians log usage and report issues, while maintenance staff schedules and performs necessary maintenance. Dr. Susie reviews usage logs, schedules maintenance, and receives alerts when supplies are low, allowing her to place orders and update inventory, with the system regularly tracking and confirming these updates.

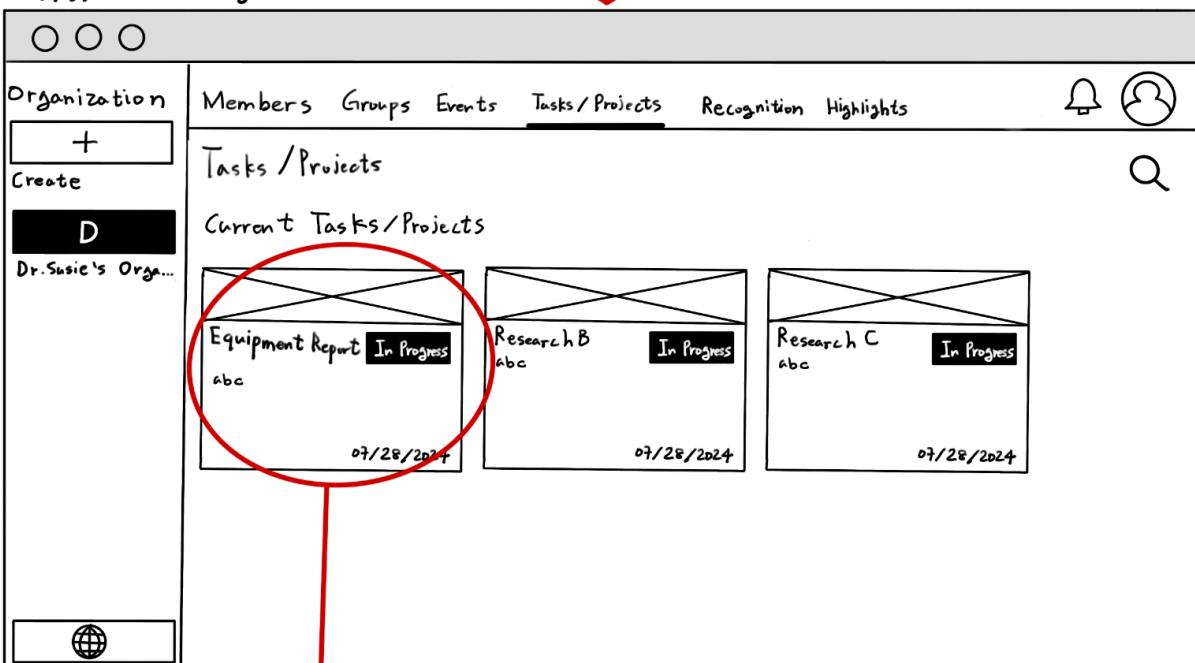


① Start - Member Perspective  
Organization Dashboard Page

Use Case #4



② Tasks / Projects Page



③

Tasks/Projects Detail Page

This screenshot shows the 'Tasks/Projects Detail Page'. On the left sidebar, there's a 'Create' button and a section for 'Dr. Susie's Org...'. The main content area displays a task titled 'Equipment Report' with a deadline of '7/31/2024'. It includes a 'Description' section with the text 'abc' and a 'Text entry' section with a rich text editor toolbar. A 'Choose upload type' dropdown menu is open, showing 'Text entry' (which is checked) and 'File'. At the bottom right of the content area, there is a large 'Upload Task/Project' button, which is circled in red.

④

Tasks/Projects Detail Page

This screenshot shows the same 'Tasks/Projects Detail Page' after a task has been uploaded. A modal dialog box is centered on the screen with the message 'Successfully uploaded your task/project.' Below this message is a 'Close' button, which is circled in red. The background of the page is darkened.

⑤

Tasks/Projects Detail Page

Organization + Create D Dr.Susie's Orga... Find

Members Groups Events Tasks/Projects Recognition Highlights

← Tasks/Projects

Equipment Report

Deadline: 7/31/2024

✓ Description abc

abc...

Text entry upload

Re-upload Task/Project

⑥ End

Tasks/Projects Page

Organization + Create D Dr.Susie's Orga... Find

Members Groups Events Tasks/Projects Recognition Highlights

Tasks / Projects

Current Tasks/Projects

Research B In Progress 07/28/2024

Research C In Progress 07/28/2024

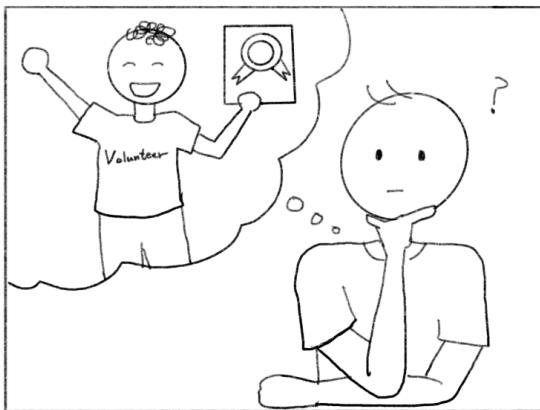
Past Tasks/Projects

Equipment Report Complete abc

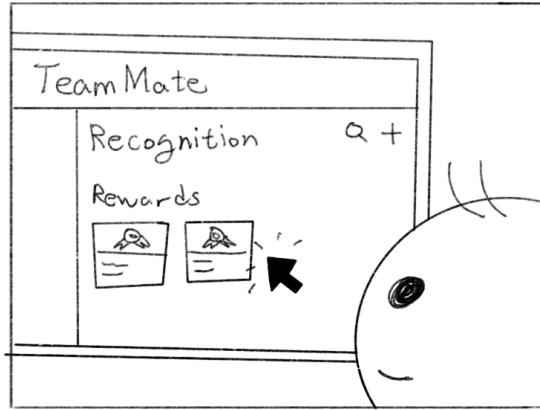
## Use Case #5

Steve, a manager at NGO XYZ, is eager to recognize and monitor volunteer activities but finds it challenging to ensure everyone is on the same pace. Discovering TeamMate, he uses the platform to track and provide a comprehensive overview of all members, ensuring no one is overlooked. He issues digital accolades visible on profiles, and TeamMate suggests awarding badges for completed tasks, which can be shared publicly on social media to recognize volunteer contributions.

① Steve struggles to recognize volunteers.



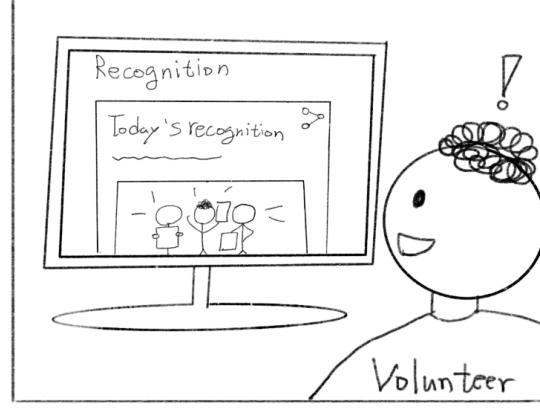
② Steve uses TeamMate for tracking and recognition.



③ Steve shares recognition on social media.



④ Volunteers feel recognized.



① Start - Leader Perspective

Use Case # 5

Member's Profile Page

This hand-drawn wireframe illustrates the 'Member's Profile Page'. On the left, there's a sidebar with icons for 'Organization' (with a '+ Create' button), 'Volunteers' Or...', and a 'Find' button. The main area has a navigation bar with tabs: Members, Groups, Events, Tasks/Projects, Recognition, and **Highlights**. A red circle highlights the 'Highlights' tab. Below it, a section titled '← Members' shows a profile for 'Volunteer 1' with details: Group: Team 1, Role: Member, Etc.: N/A. There's also a 'Recognition' section with a 'Certification' table containing numbers 1 through 6. An 'Edit' button is located in the top right corner.

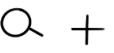
② Highlights Page

This hand-drawn wireframe illustrates the 'Highlights Page'. It features a similar layout to the profile page, with a sidebar for 'Organization', 'Volunteers' Or...', and 'Find'. The main area has a navigation bar with tabs: Members, Groups, Events, Tasks/Projects, Recognition, and **Highlights**. A red circle highlights the 'Highlights' tab. A large central box displays a post titled 'Best Volunteers of May' (published on 05/26/2024) with a large 'X' mark over it. Below the title, there's placeholder text 'abc ...'. Underneath, it says 'Tagged members' followed by three user icons and the text 'VolunteerX and others'. To the right of the post is a search icon and a red circle containing a '+' sign.

③  
Highlights Page

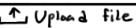
Organization +  
Create  
V  
Volunteers' Or...  
Find

Members Groups Events Tasks/Projects Recognition Highlights  

← Highlights  +

Create a highlight 

Header : Top Volunteer of June

Upload Content: Image 1 

Description: abc

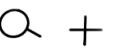
Tag members: Name or email address  
Volunteer 1 X

Post

④ End  
Highlights Page

Organization +  
Create  
V  
Volunteers' Or...  
Find

Members Groups Events Tasks/Projects Recognition Highlights  

Highlights  +

Top Volunteer of June   
Published: 06/26/2024

abc...

Tagged members  
 Volunteer 1

Best Volunteers of May

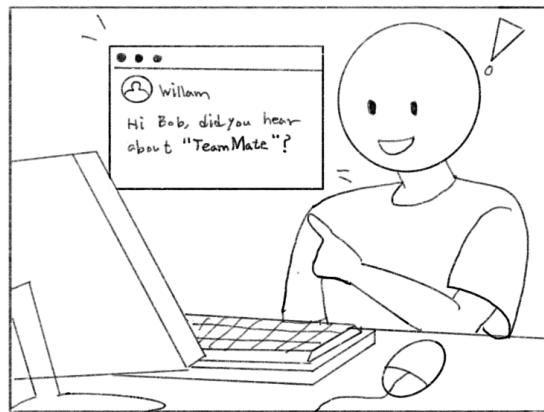
## Use Case #6

Bob, a CS graduate from SFSU, founded a gaming community called Inferno, which grew to 100 active members hosting events across the Bay Area. As the community expanded, managing events, teams, and individual achievements became challenging without raising membership fees. A member introduced him to TeamMate, an online management platform that organizes events, tracks achievements, and manages profiles. Using TeamMate, the community can efficiently handle events and tournaments, track attendance, and make announcements, ensuring smooth operations and detailed member profiles.

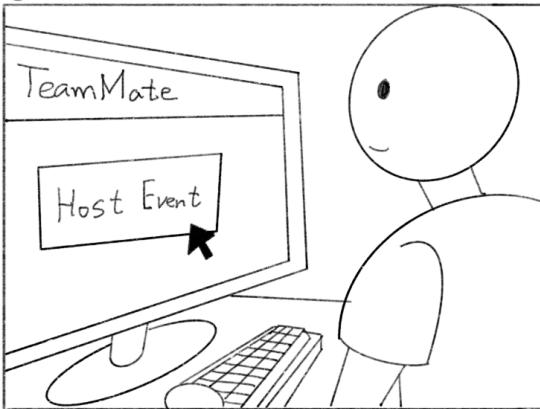
① Bob struggles managing growing gaming community.



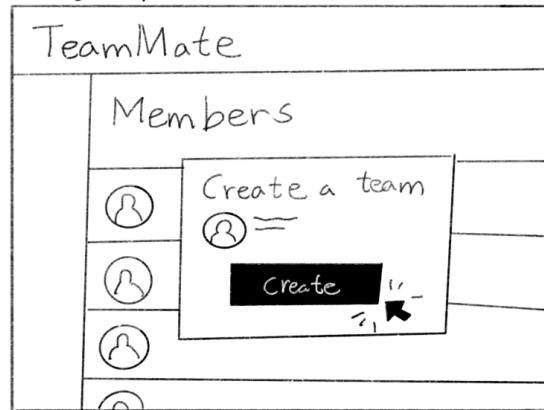
② Bob is introduced to TeamMate for management.



③ Bob hosts events for players to join.



④ Bob is creating teams based on who signed up to the event.



① Start - Leader Perspective  
Events Create Page

Use Case #6

Events Create Page

Create an event

Event name :	Team 1 vs. Team 2	Invited :	Name or everyone
Event type :	In-person	Virtual	Everyone X
Location :	Computer Lab	Content :	Image 1 Upload file
Date & time :	6/28/2024	to	7/28/2024
Hosted by :	Bob	Detail :	abc...

**Post**

② Events Page

Events Page

Events

Upcoming Events

Team 1 VS. Team 2  
7 pm - 7 pm | 07/28/2024

Past Events

Game Play 1

Game Play 2

(3)

Events Detail Page

Organization Members Groups Events Tasks / Projects Recognition Highlights

Gaming Organization

Team 1 vs. Team 2

Hosted by Bob

✓ Detail abc...

June 28 In 3 days  
5 pm - 7 pm Computer Lab

Attendees

Player 1
Player 2

**Edit** **Set up groups**

(4)

Events Group Create Page

Organization Members Groups Events Tasks / Projects Recognition Highlights

Gaming Organization

Team 1 vs.

Hosted by

✓ Detail abc...

Create groups

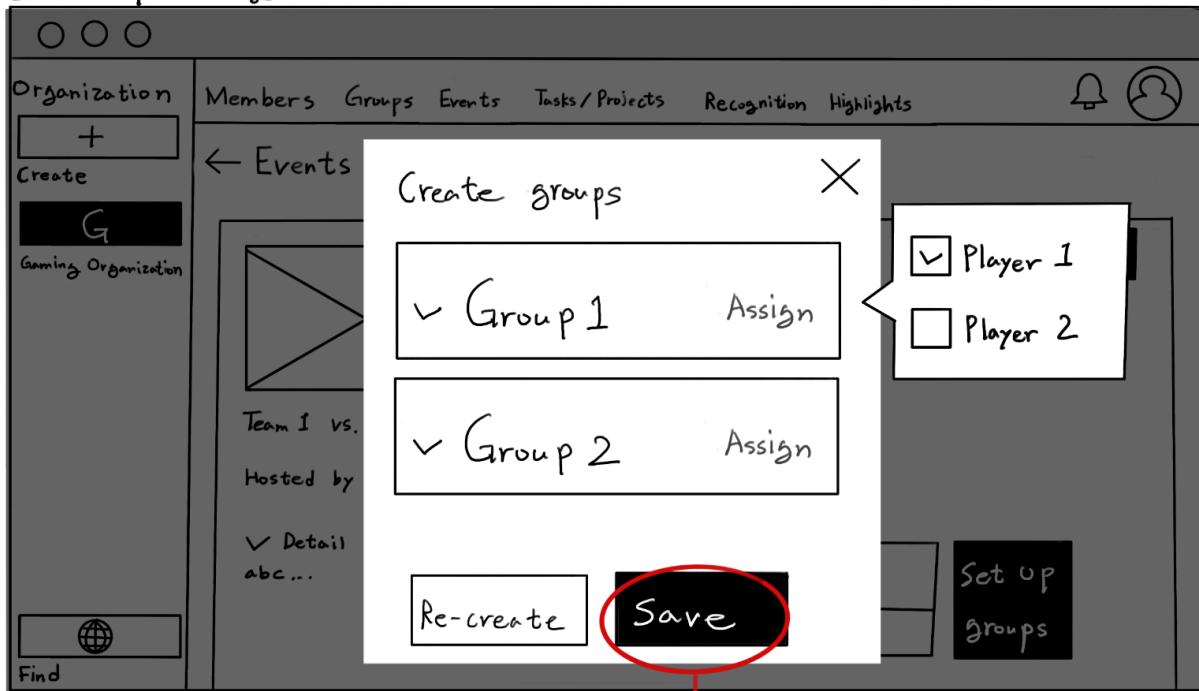
2 ^ groups

Assign automatically  
 Assign manually  
 Manually join

**Create** **Set up groups**

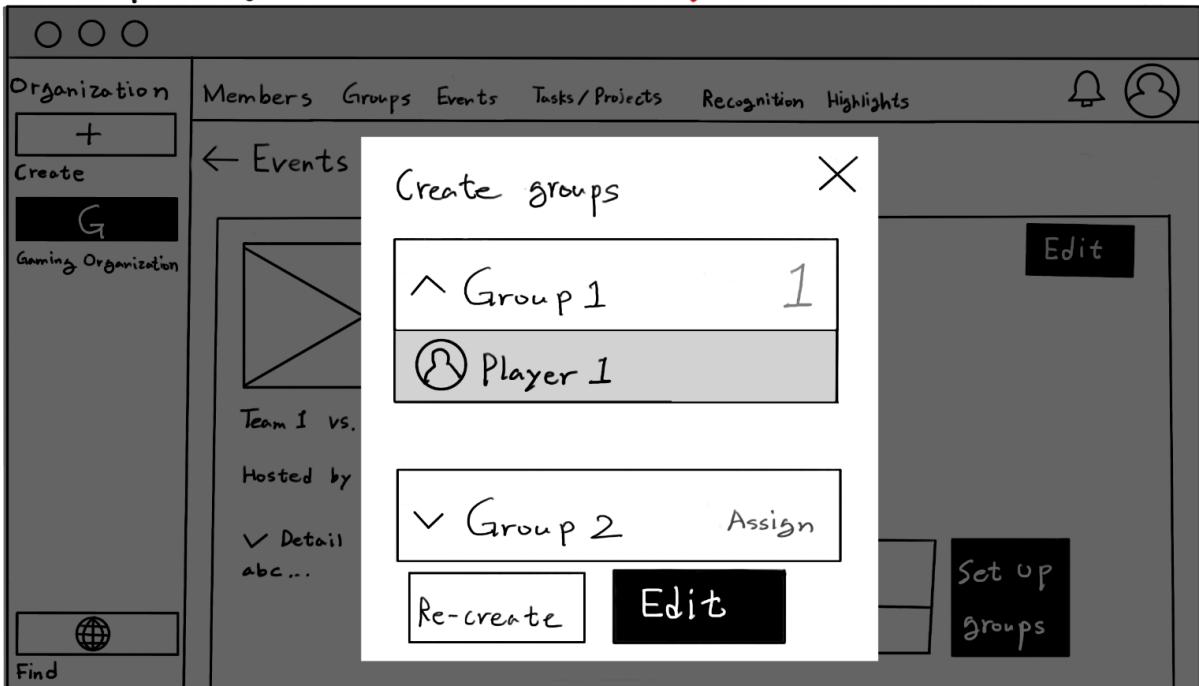
⑤

Events Group Create Page



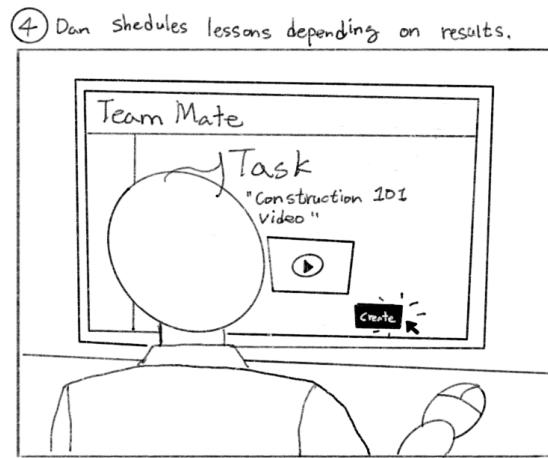
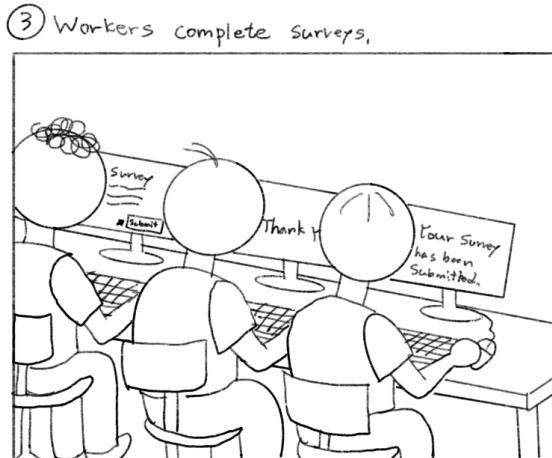
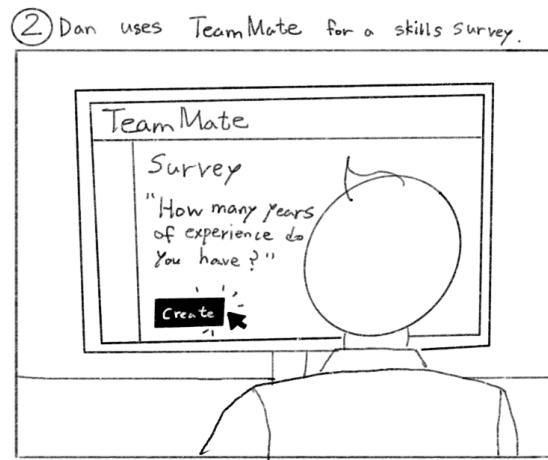
⑥ End

Events Group Create Page



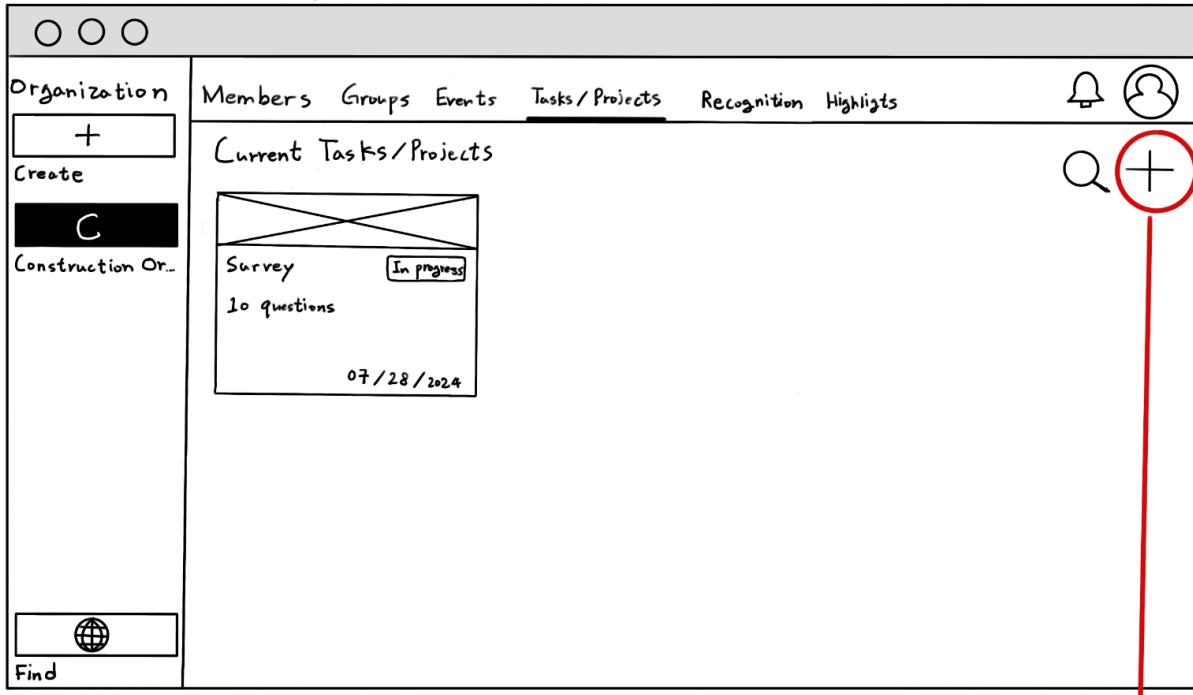
## Use Case #7

Dan, a manager at Danliland construction company, employs 20 workers and uses the TeamMate App to track their training needs and experience. Workers like Luigi, Angel, and Joshi complete a survey in the app to highlight their knowledge and past learning. The app allows Dan to manage worker accounts, monitor survey completion, and schedule meetings based on the survey results to address any learning gaps.



① Start - Leader Perspective  
Tasks/Projects Dashboard Page

Use Case # 7



② Task / Project Create Page

③ End

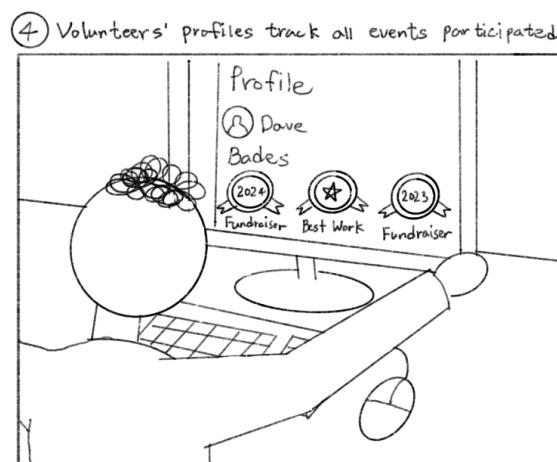
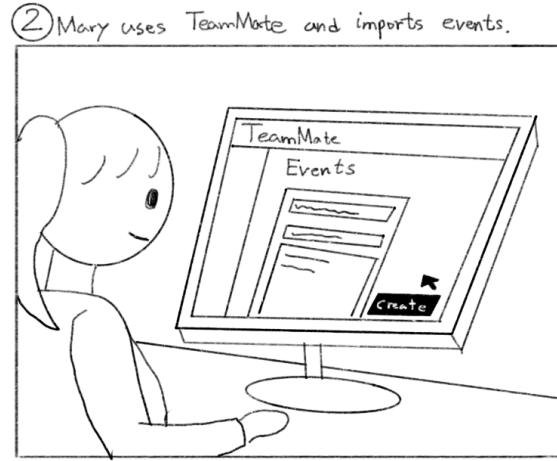
Task / Project Create Page

○ ○ ○

Organization + Create C Construction Or...   Find	Members Groups Events Tasks / Projects Recognition Highlights ◀ Tasks / Projects
<p>Create a task / project + Add recognition</p> <p>Name : Watch : Construction 101</p> <p>Description : abc</p> <p>Upload Content :  (X) Construction 101.mp4</p> <p>Assigned to : <input checked="" type="checkbox"/> Worker 1 ^</p>	

## Use Case #8

Mary, a coordinator at a local non-profit, struggled to track volunteer hours, activities, and accomplishments manually. Discovering TeamMate, she created an account, imported historical data, and set up her organization. The platform allows her to assign tasks, track hours in real-time, manage schedules, and generate reports to highlight volunteer efforts, making the process efficient and helping secure future funding.



① Start - Member Perspective  
Signin Page

Use Case # 8

TeamMate  
Welcome, Mate!

Email address  
Volunteer 1

Password  
•••••

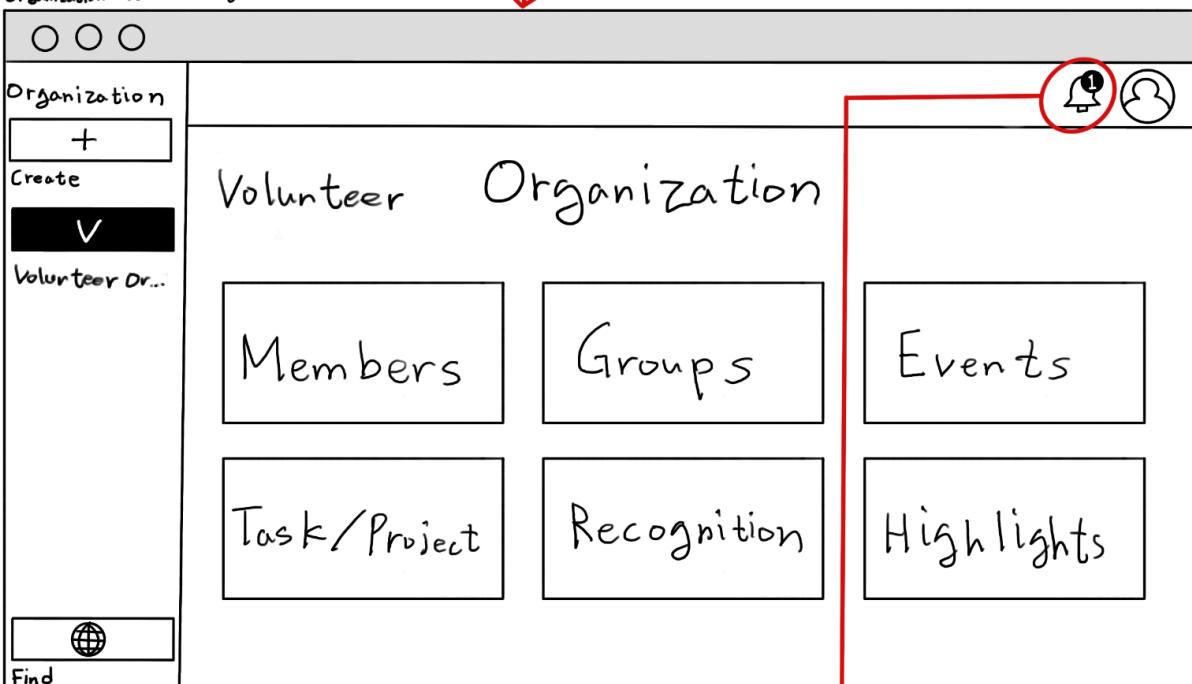
Remember me      [Forgot Password?](#)

Sign in

OR

Create an account

② Organization Dashboard Page



③

Organization Dashboard Page

Organization Dashboard Page

Volunteer Organization

Organization

Create

Volunteer Org...

Members

Groups

Task/Project

Recognition

Highlights

Notifications

All Unread

You earned a reward for attending event!

Event Coming up soon

Yesterday

Other notification

Find

④ End

Member's Profile Page

Member's Profile Page

Organization

Create

Volunteer Org...

Members Groups Events Tasks/Projects Recognition Highlights

← Volunteer Organization

Volunteer 1 (You)

Edit

Group: N/A

Role: Member

Etc.: N/A

Recognition

3 New

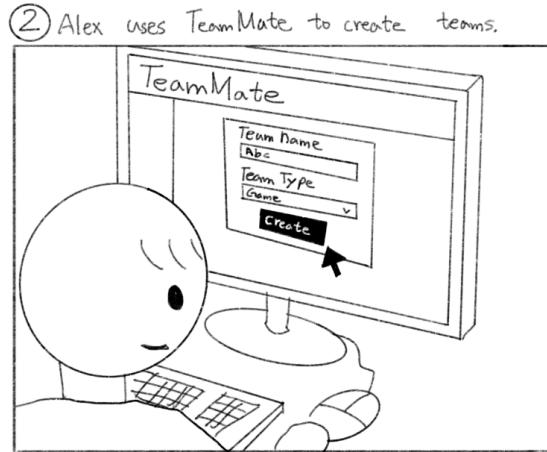
2

1

Find

## Use Case #9

Alex, a teacher running a high school gaming club, struggled to manage the growing number of participants, organize teams, track rankings, and find new opponents. Discovering TeamMate, he created an account and set up the club's organization, enabling students to join and participate through the platform. TeamMate allows Alex to efficiently create teams, update scores, track rankings, schedule matches, and organize tournaments, while students can view their rankings, team assignments, sign up for competitions, and communicate with teammates, keeping everyone informed and motivated.



① - Member Perspective  
Group Dashboard Page

Use Case #9

Organization Members Groups Events Tasks / Projects Recognition Highlights

**Groups**

My groups  
You're not in any groups.

All groups

Team 1 abc	Invite only 3/5
Team 2 abc	Full 5/5
Team 3 abc	Open 4/5

②  
Group Detail Page

Organization Members Groups Events Tasks / Projects Recognition Highlights

← Groups

**Team 3**

Anyone can join | 4 members | 5 members max

Join

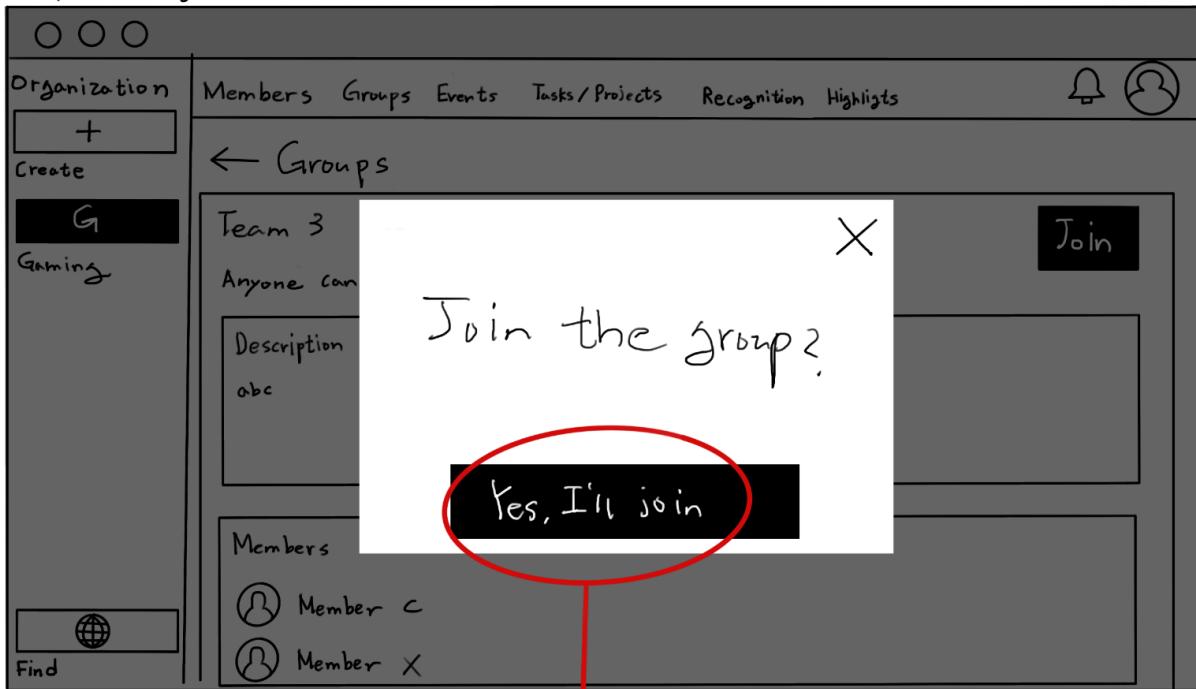
Description  
abc

Members

- Member C
- Member X

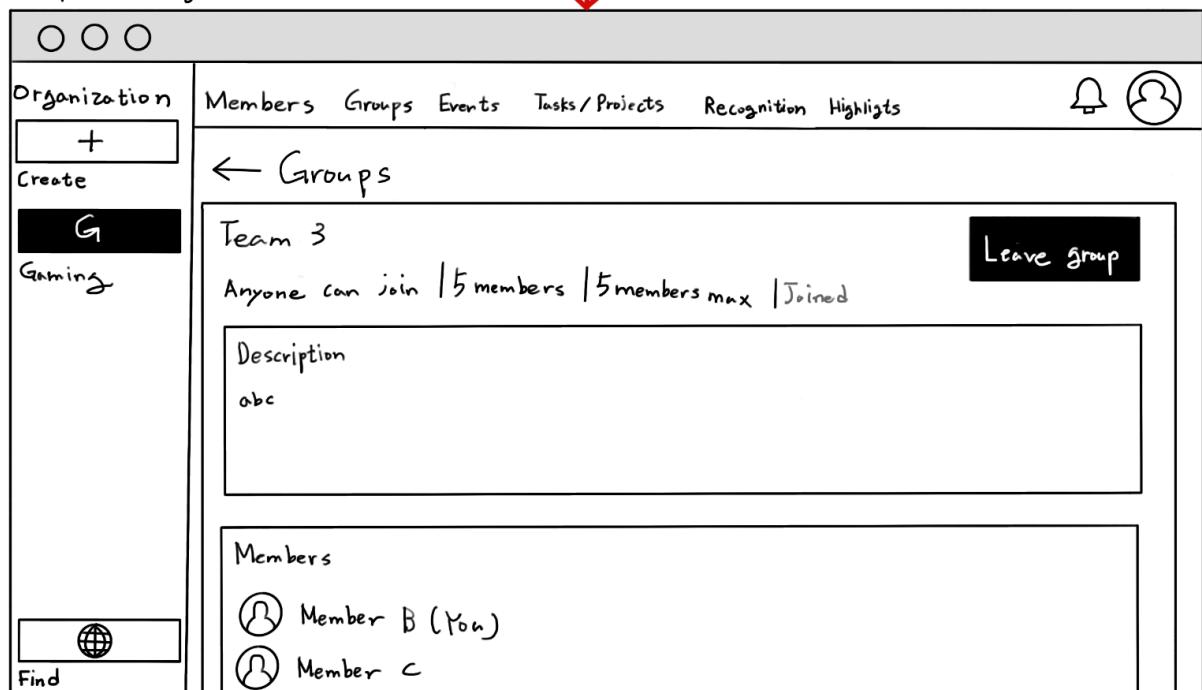


③ Group Detail Page



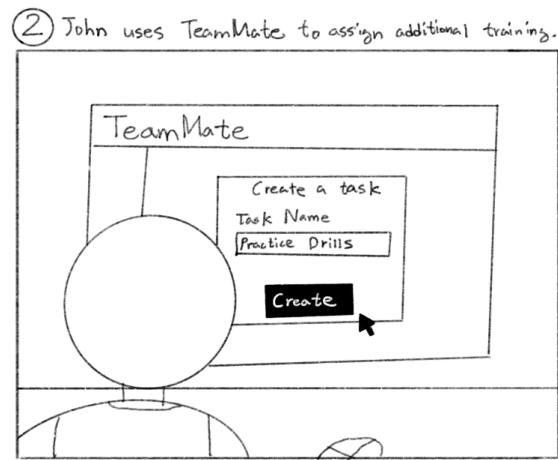
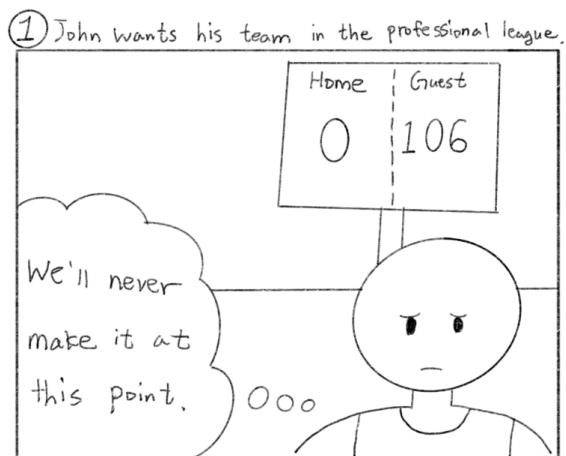
## ④ End

## Group Detail Page



## Use Case #10

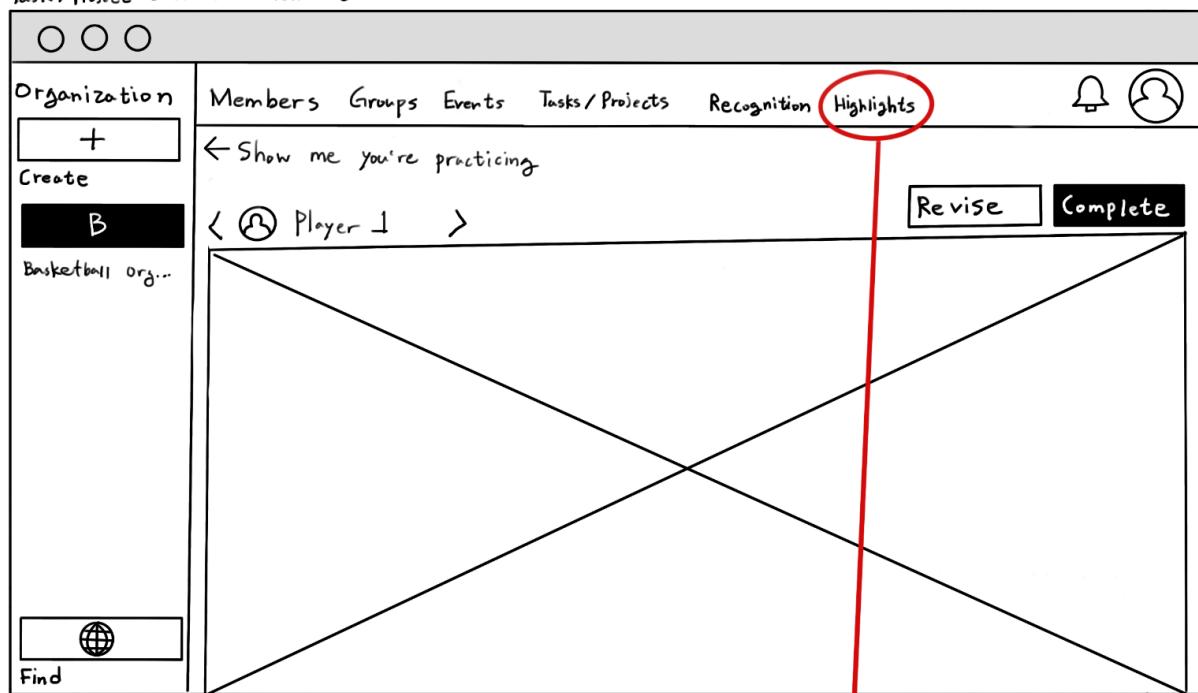
John, a full-time basketball team leader working multiple part-time jobs for funding, struggled to manage team activities, training schedules, physical conditions, and communication. After being introduced to TeamMate, he created an account and set up his organization, enabling teammates to join. The app allows John to assign tasks, track progress, make announcements, and reward active players, fostering a positive and engaged environment while simplifying team management.



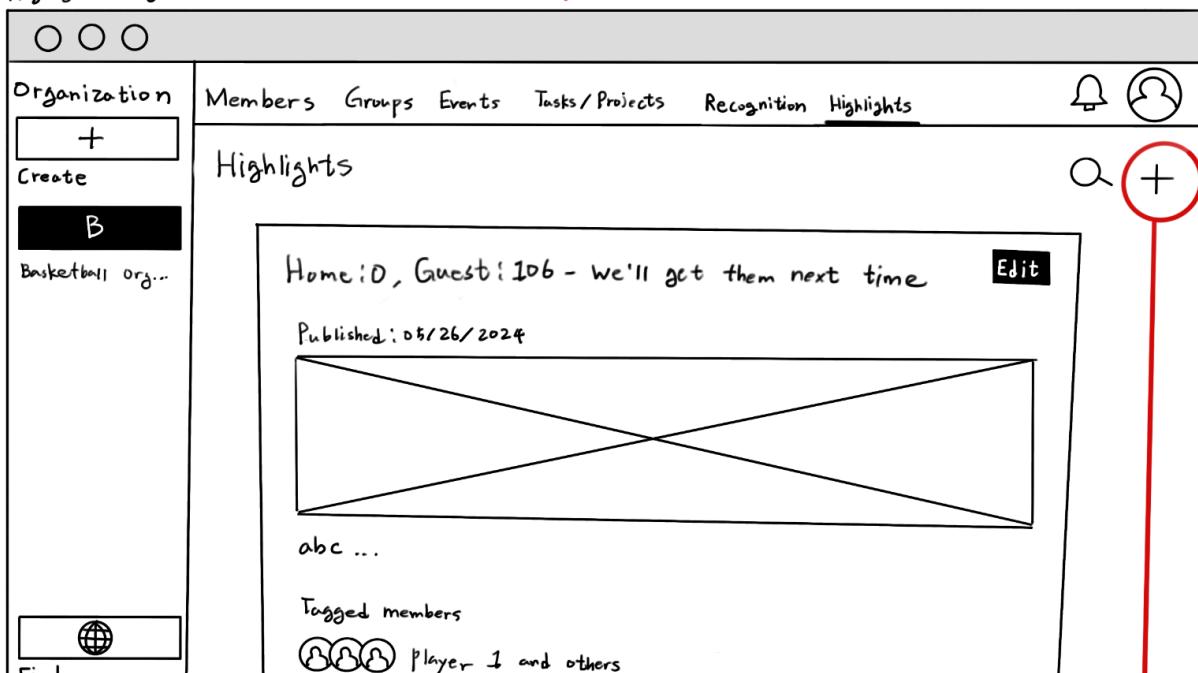
① Start - Leader Perspective

User Case #10

Task / Project Submission View Page



② Highlights Page



Use Case #10

③

Highlights Page

Organization + Create B Basketball org... Find

Members Groups Events Tasks/Projects Recognition Highlights

← Highlights Q +

Create a highlight Preview

Header : Home:110, Guests 86 - Our first time...

Upload Content: Image 1

Description: abc

Tag members: Name or email address  
Player 1 X Player 2 X

Post

④ End

Highlights Page

Organization + Create B Basketball org... Find

Members Groups Events Tasks/Projects Recognition Highlights

Highlights Q +

Home:110, Guests: 86 - Our first time win! Edit

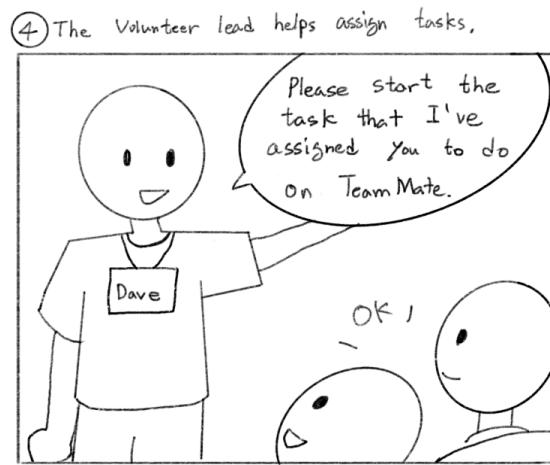
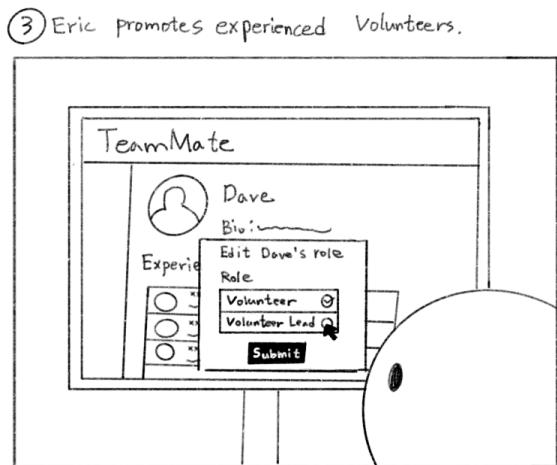
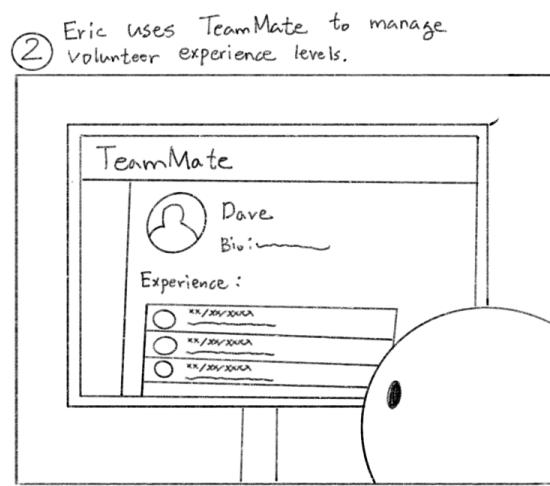
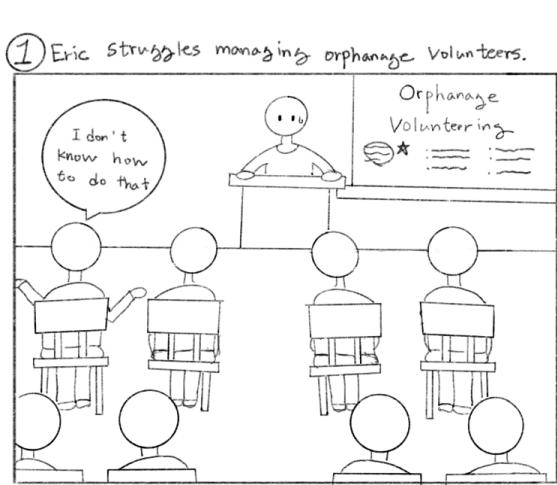
Published: 06/26/2024

abc ...

Tagged members Player 1 and others

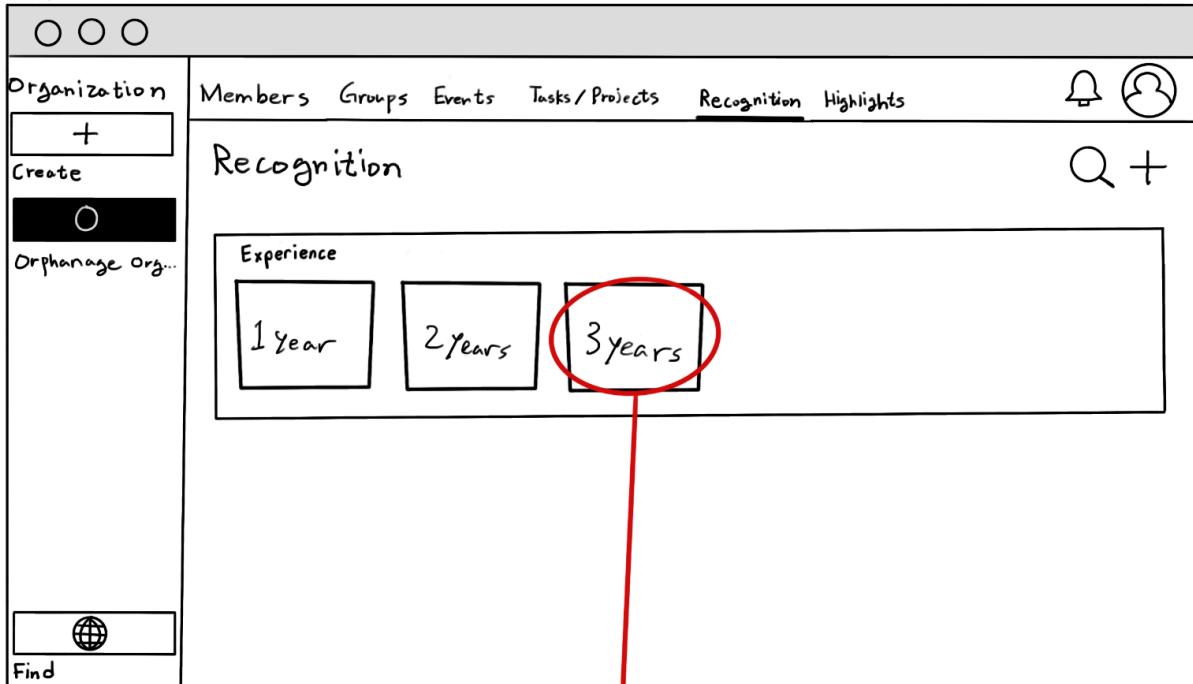
## Use Case #11

Eric, managing a large orphanage and struggling with limited staff, sought to streamline volunteer management. Introduced to the TeamMate app, he created an account and onboarded his faculty and volunteers to facilitate communication, task assignment, rewards, and a leaderboard. Despite initial convenience, Eric found managing still challenging due to his busy schedule, so he delegated administrative tasks to a trusted faculty member. This reduced Eric's workload significantly, allowing him to focus on overseeing operations and empowering active volunteers as temporary leaders.

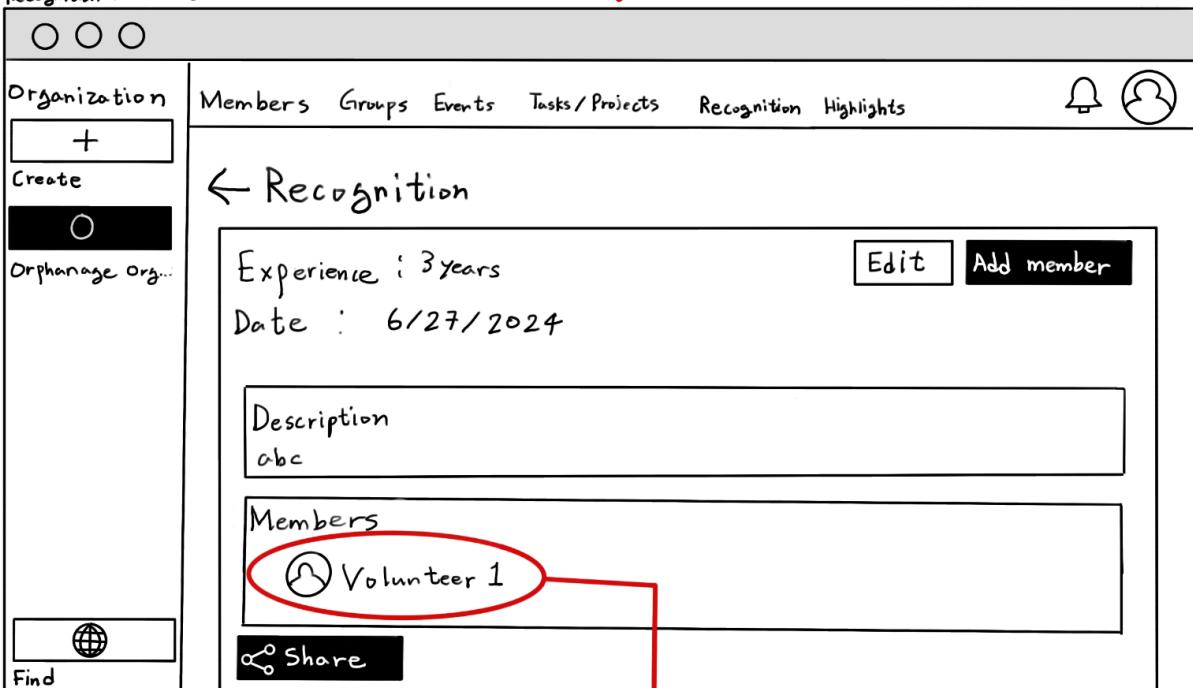


① Start - Leader Perspective  
Recognition Dashboard Page

User Case #11



② Recognition Detail Page



③

User Case #11

Member's Profile Page

Organization  
+  
Create  
Orphanage org...  
Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Volunteer 1

Edit

Group: N/A  
Role: Member  
Etc.: N/A

Recognition  
3 years

④ End

Member's Profile Edit Page

Organization  
+  
Create  
Orphanage org...  
Find

Members Groups Events Tasks/Projects Recognition Highlights

← Members

Volunteer 1

Cancel Save

Group: N/A ▾  
Role: Member ▾  
Etc.: Admin

Recognition  
3 years

## Use Case #12

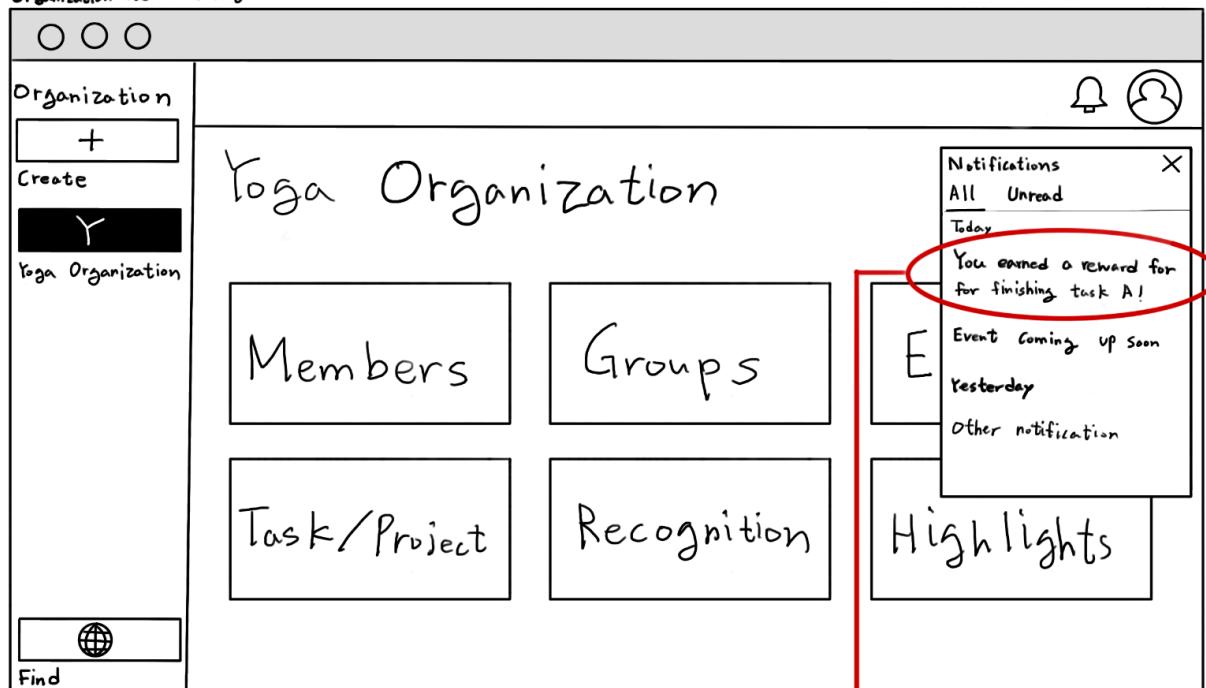
Cianci, director of a Yoga class in Novato, faces challenges with student motivation and goal achievement. To address this, she uses the TeamMate app to set monthly goals for her students and track their progress and methods used. The app allows her to monitor obstacles hindering progress, such as tardiness or distractions, and track designated days for achieving goals, ensuring accountability and continuous improvement among her students.



① Start - Member Perspective

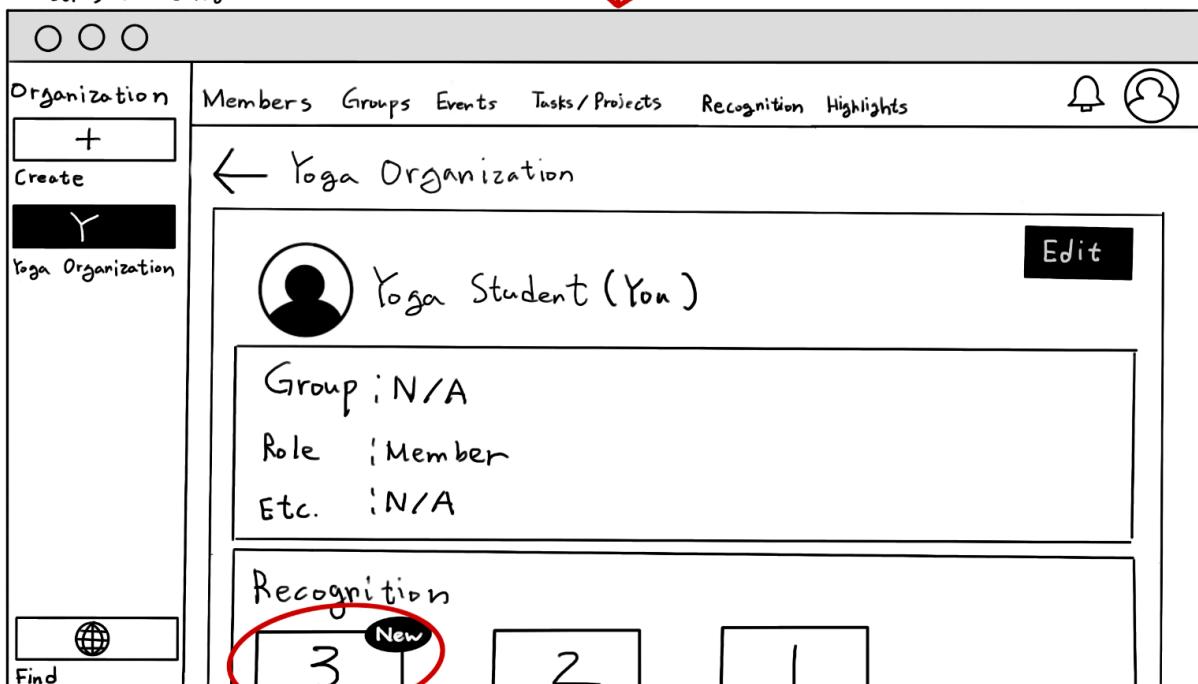
Use Case #12

Organization Dashboard Page



②

Member's Profile Page



③End

Recognition Detail Page

The wireframe depicts a user interface for a recognition system. On the left, a sidebar titled "Organization" contains a "Create" button with a plus sign and a "Yoga Organization" entry. A "Find" button with a magnifying glass icon is also present. The main content area has a header with tabs: Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. It includes a notification bell and profile icons. A back arrow labeled "Profile" is visible. The central area displays "Goals: Improve Balance" and "Date : 6/27/2024". Below this, a "Description" field contains the text "abc". A "Members" section shows "Yoga Student 1" with a user icon. A "Share" button with a network icon is located at the bottom.

# High Level Database Architecture

## DBMS Summary

For this project, we will use MySQL as the DBMS and connect it to an EC2 instance to manage and track the database. MySQL is chosen for our project because of its reliability, scalability, and wide support. It will be an excellent tool for efficiently handling database operations on a cloud-based environment like AWS EC2 we are using.

## Functional Requirements

### 1. Registered Users (pka: account)

- 1.1. Registered users shall at most make one new account.
- 1.2. A Registered user shall have one ID.
- 1.3. A Registered user shall have one unique email.
- 1.4. A Registered user shall have one username.
- 1.5. A Registered user shall have one password.
- 1.6. A Registered user shall have zero or one profile picture.
- 1.7. A Registered user shall have one first name.
- 1.8. A Registered user shall have one last name.
- 1.9. A Registered user can make zero or many comments.
- 1.10. A registered user can make zero or many organizations.
- 1.11. A registered user can be one or many members.

### 2. Member

- 2.1. A member shall have an ID.
- 2.2. A member is a registered user.
- 2.3. A member shall have one first name.
- 2.4. A member shall have one last name.
- 2.5. A member shall have one datetime stamp showing their last login.
- 2.6. A member can receive zero or many qualification instances.
- 2.7. A member can receive zero or many award instances.
- 2.8. A member can give zero or many qualification instances.
- 2.9. A member can give zero or many award instances.
- 2.10. A member shall have an organization.

- 2.11. A member shall have zero or many event reminders.
- 2.12. A member can have a group.
- 2.13. A member can create a custom section if able.

### **3. Organization**

- 3.1. An organization shall have an ID.
- 3.2. An organization shall have a name.
- 3.3. An organization shall have one owner (member).
- 3.4. An organization will have one description.
- 3.5. An organization shall have zero or many groups.
- 3.6. An organization shall have zero or many events.
- 3.7. An organization shall have one or many members.
- 3.8. An organization shall have zero or many custom sections.

### **4. Award Instance**

- 4.1. An award instance shall have an ID.
- 4.2. An award instance shall have one member who issued it.
- 4.3. An award instance shall have one member who received it.
- 4.4. An award instance shall have one award type.
- 4.5. An award instance shall have one nomination status.
- 4.6. An award instance shall have the date of when it was given.
- 4.7. An award instance shall have a citation.

### **5. Award Type**

- 5.1. An award type shall have an ID.
- 5.2. An award type shall have one description.
- 5.3. An award type shall have an associated image.
- 5.4. An award type shall have a title.
- 5.5. An award type shall have one or many award instances.

### **6. Qualification Instance**

- 6.1. An qualification instance shall have an ID.
- 6.2. A qualification instance shall have one member who issued it.
- 6.3. A qualification instance shall have one member who received it.
- 6.4. A qualification instance shall have one qualification type.

- 6.5. A qualification instance shall have one nomination status.
- 6.6. A qualification instance shall have the date of when it was given.
- 6.7. A qualification instance shall have a citation.

## **7. Qualification Type**

- 7.1. A qualification type shall have an ID.
- 7.2. A qualification type shall have one description.
- 7.3. A qualification type shall have an associated image.
- 7.4. A qualification type shall have a title.
- 7.5. A qualification type shall have one or many qualification instances.

## **8. Events**

- 8.1. An event shall have an ID.
- 8.2. An event shall have one participating organization.
- 8.3. An event shall have one title.
- 8.4. An event shall have one description.
- 8.5. An event shall have one start datetime.
- 8.6. An event shall have an event type.
- 8.7. An event shall have zero or many prerequisites.
- 8.8. An event shall have one or many event reminders.

## **9. Custom Section**

- 9.1. A custom section will have an ID.
- 9.2. A custom section will have one title.
- 9.3. A custom section will have access levels to edit.
- 9.4. A custom section will have a “plaintext” section.
- 9.5. A custom section will have a third party section.
- 9.6. A custom section will have a member be the creator.
- 9.7. A custom section will belong to one organization.

## **10. Comments**

- 10.1. A comment shall have an ID.
- 10.2. A comment shall have one user associated with it.
- 10.3. A comment shall have one page type.
- 10.4. A comment shall have one page key.

- 10.5. A comment shall have one comment description.
- 10.6. A comment shall have one date for when it was posted.

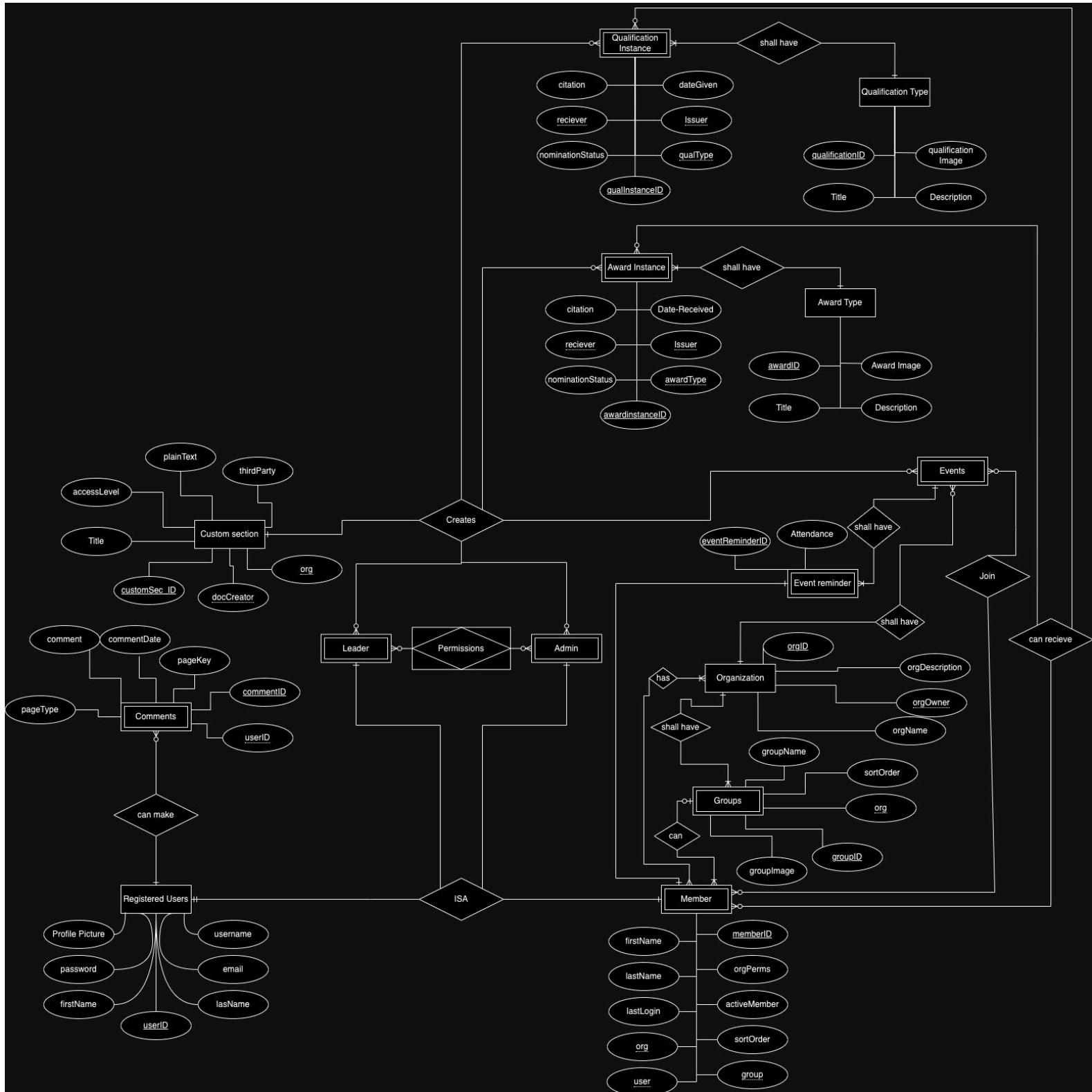
## **11. Event reminder**

- 11.1. An event reminder shall have an ID.
- 11.2. An event reminder shall have one event.
- 11.3. An event reminder shall have one member.
- 11.4. An event reminder shall track if a member went to the event or not (bool).

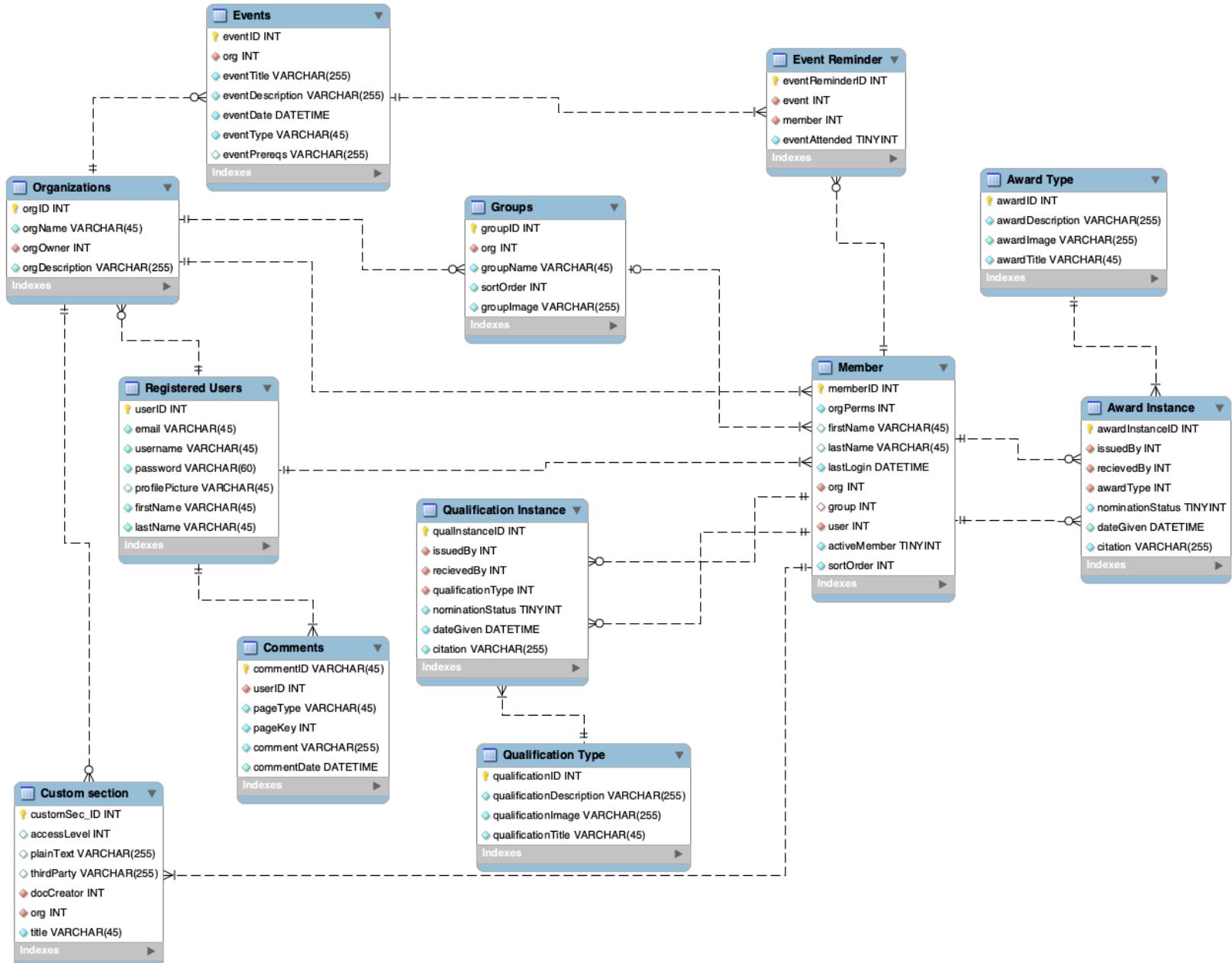
## **12. Groups**

- 12.1. A group shall have an ID.
- 12.2. A group shall have one organization.
- 12.3. A group shall have one group name.
- 12.4. A group shall have a sort order number to organize the groups.
- 12.5. A group shall have an associated image.
- 12.6. A group shall have one or many members.

## ERD (Entity Relationship Diagram)



## EER (Enhanced Entity-Relationship)



**Media Storage:**

For media storage we decided to use the file systems as opposed to DB BLOBS because we as a group decided it would be easier for us to do due to our familiarity with it. We will use strings to store the file names so that we can route to it.

# High Level APIs and Main Algorithms

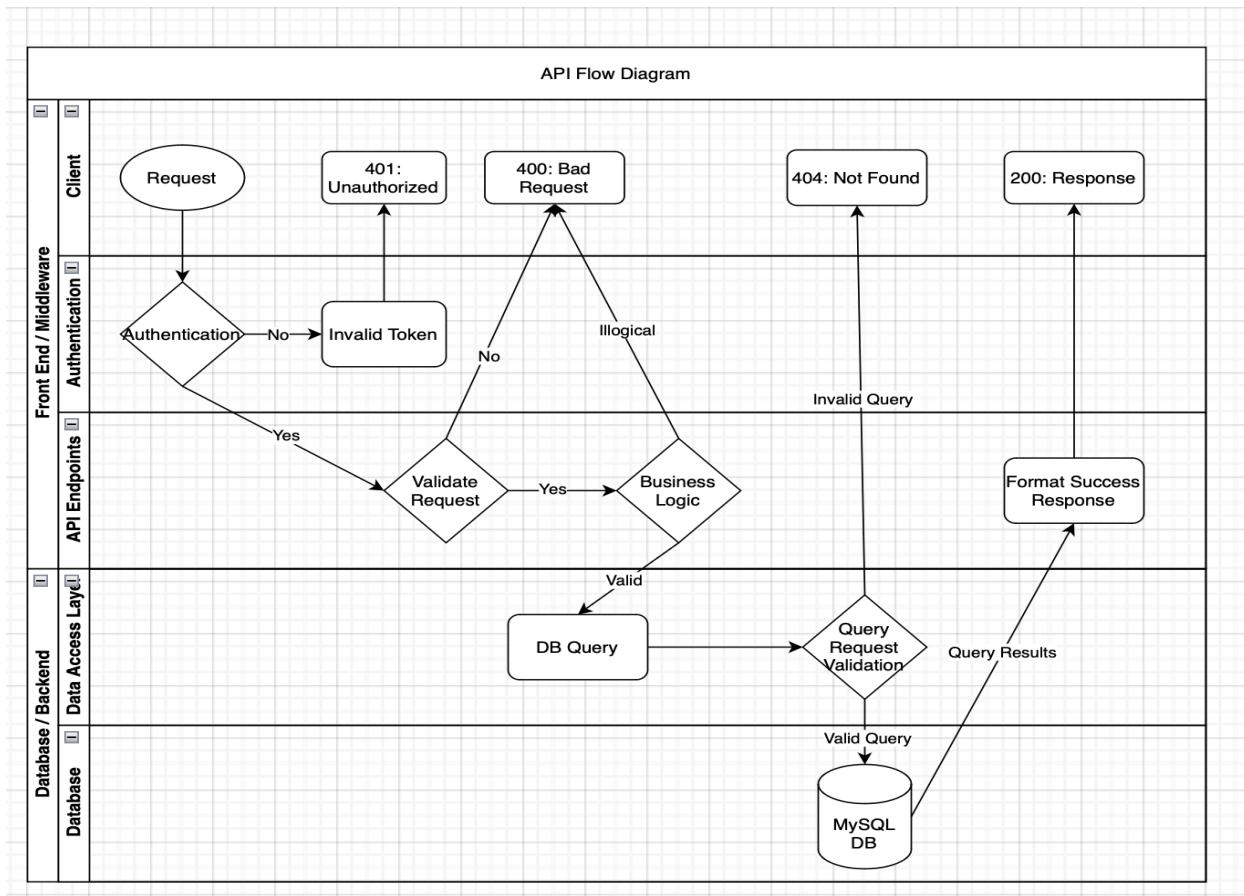
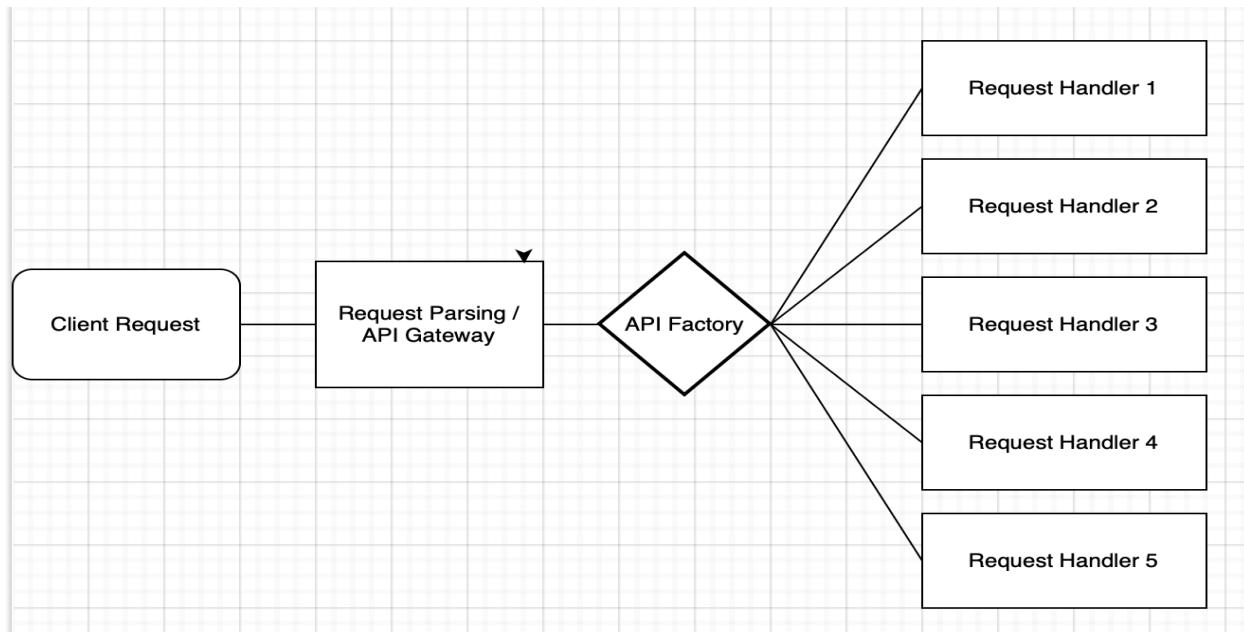
## API HIGH LEVEL ARCHITECTURE

Our app, TeamMate, requires various endpoints for its features. These endpoints handle different functions such as login, create organizations, give awards, admin permissions, etc. Because of this diversity, we decided to implement a control flow that allows us to determine the API response dynamically. We need to be able to add features without interfering with the overall structure of the APIs.

This will be achieved through a monolithic architecture with a single gateway but multiple endpoints. This architecture will direct traffic from different API requests based on the type of business logic they need. By utilizing a factory pattern (to determine where the requests need to go), we can scale the number of endpoints and handlers as needed. This way, adding new features will be straightforward for developers and will not affect performance of the API.

To add another business logic/user function, we would need to add the desired route to the API factory and then from the factory to the request handler that manages all the 'business logic' such as database queries.

Below is a control flow of the API illustrating how it handles requests. This method ensures that we validate the request at different stages and return the correct errors based on the context. This multi-stage validation enhances security and accuracy by ensuring that the API passes through several gates before responding to the user. Steps such as authentication, request validation, business logic processing, and query validation prioritize security while ensuring accuracy. By implementing this design into our API architecture, scaling APIs with the same structure becomes easier as these rules are consistently applied.



# System Design

Our approach prioritizes cost mitigation by maintaining a monolithic structure while leveraging various technologies to achieve scalability and maintainability. The design allows for the deployment of additional monolithic servers as needed, ensuring a balance between operational efficiency and scalability.

We have chosen Amazon Web Services (AWS) for its robust and stable platform, along with its ease of deployment on EC2 instances. AWS provides a comprehensive suite of services that support scalable application deployment. The application will be accessible via mobile and web applications, compatible across multiple platforms including Linux, Windows, Mac, Android, and iOS. This cross-platform compatibility will be achieved through the use of Docker containers, each configured with system-specific requirements.

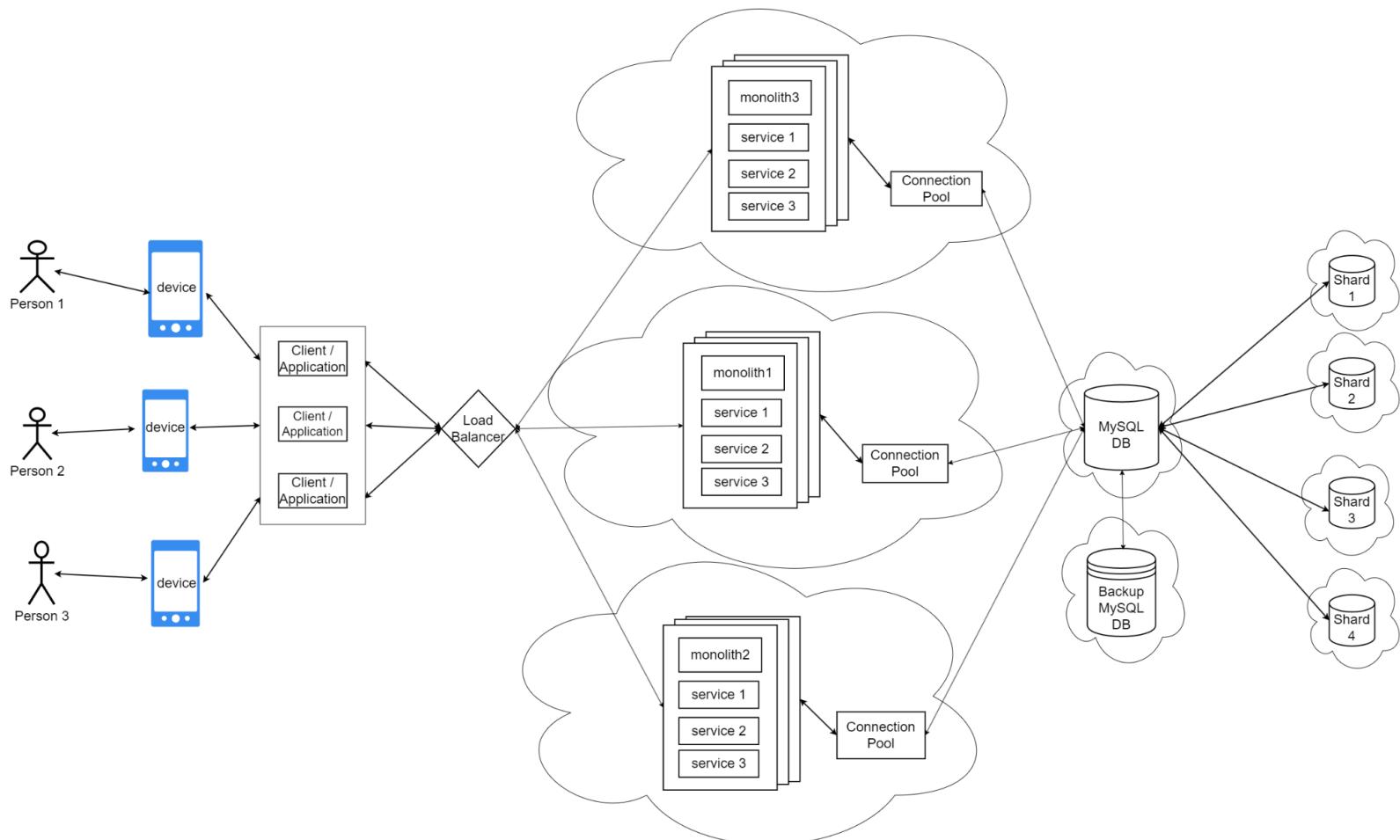
Users will interact with the platform through a React application deployed on EC2. This React front-end will communicate with a back-end server implemented using Express and TypeScript, also hosted on EC2. This setup ensures seamless interaction and maintains cross-language compatibility between the front-end and back-end. Communication between the user, application, and server will utilize TCP connections, with a message queue facilitating request and response management. This message queue decouples the application components, enabling asynchronous communication.

A message queue will precede a load balancer, distributing incoming requests among multiple servers. This ensures high availability, optimal resource utilization, reliability, and performance by preventing any single server from becoming a bottleneck. Horizontal scaling will be implemented by adding additional instances of the monolithic application on separate servers. The load balancer will manage traffic distribution across these instances, enhancing the system's ability to handle increased loads by simply adding more servers as necessary. Vertical scaling involves upgrading the resources (CPU, RAM, storage) of existing EC2 instances. AWS facilitates this by allowing easy upgrades to more powerful instance types, improving the performance of the application on individual servers when horizontal scaling is insufficient.

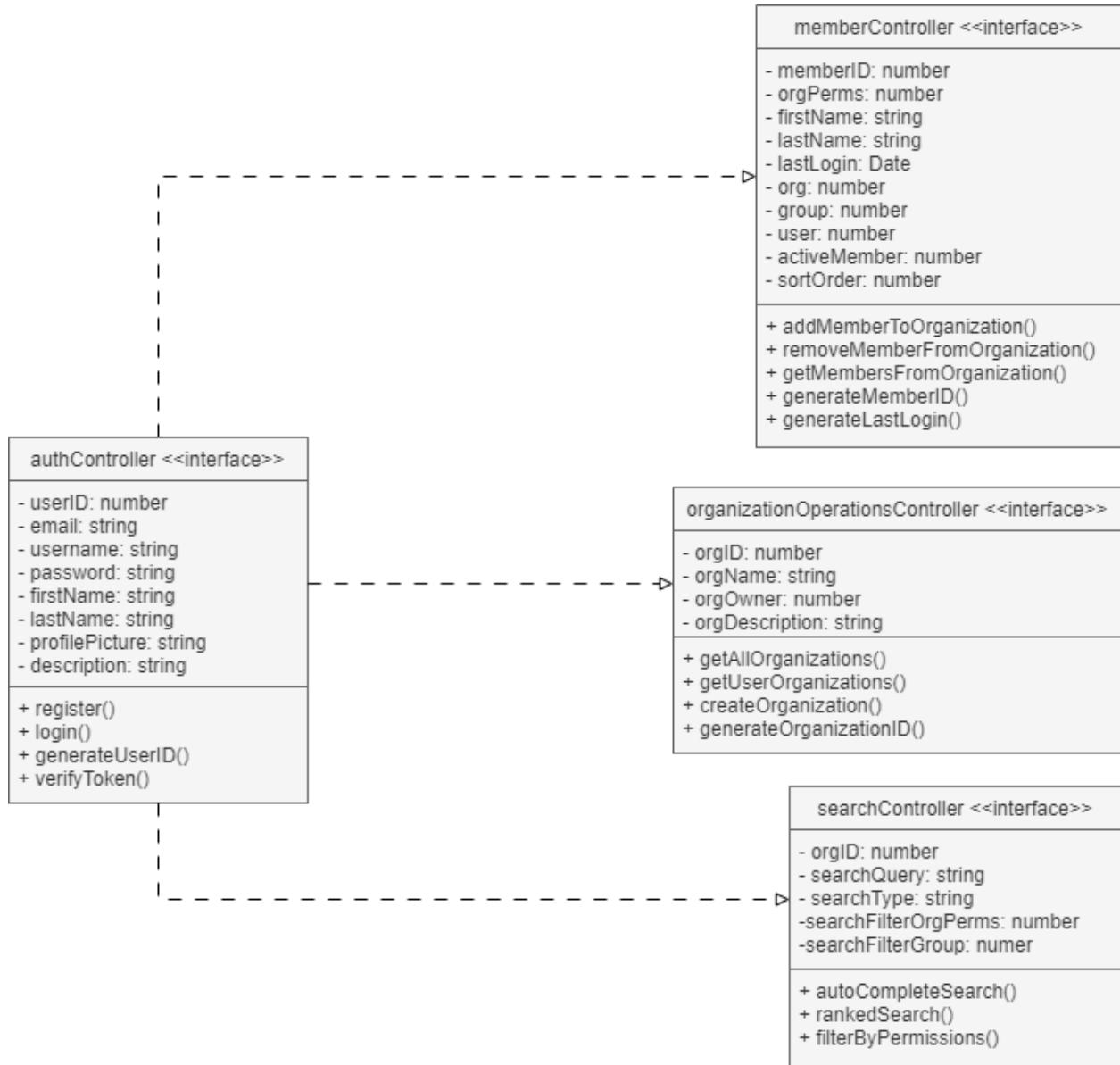
Database operations will be optimized using a connection pool, reusing connections for future database queries to reduce redundancy and manage resources efficiently. As the service grows, the database will be sharded, partitioning large datasets across multiple database nodes. This sharding strategy will increase throughput, expand storage capacity, and ensure high availability in case of a failure. A

backup database will maintain copies of critical data, updated on a regular schedule. This ensures data integrity and availability, preventing data loss in the event of a primary database failure.

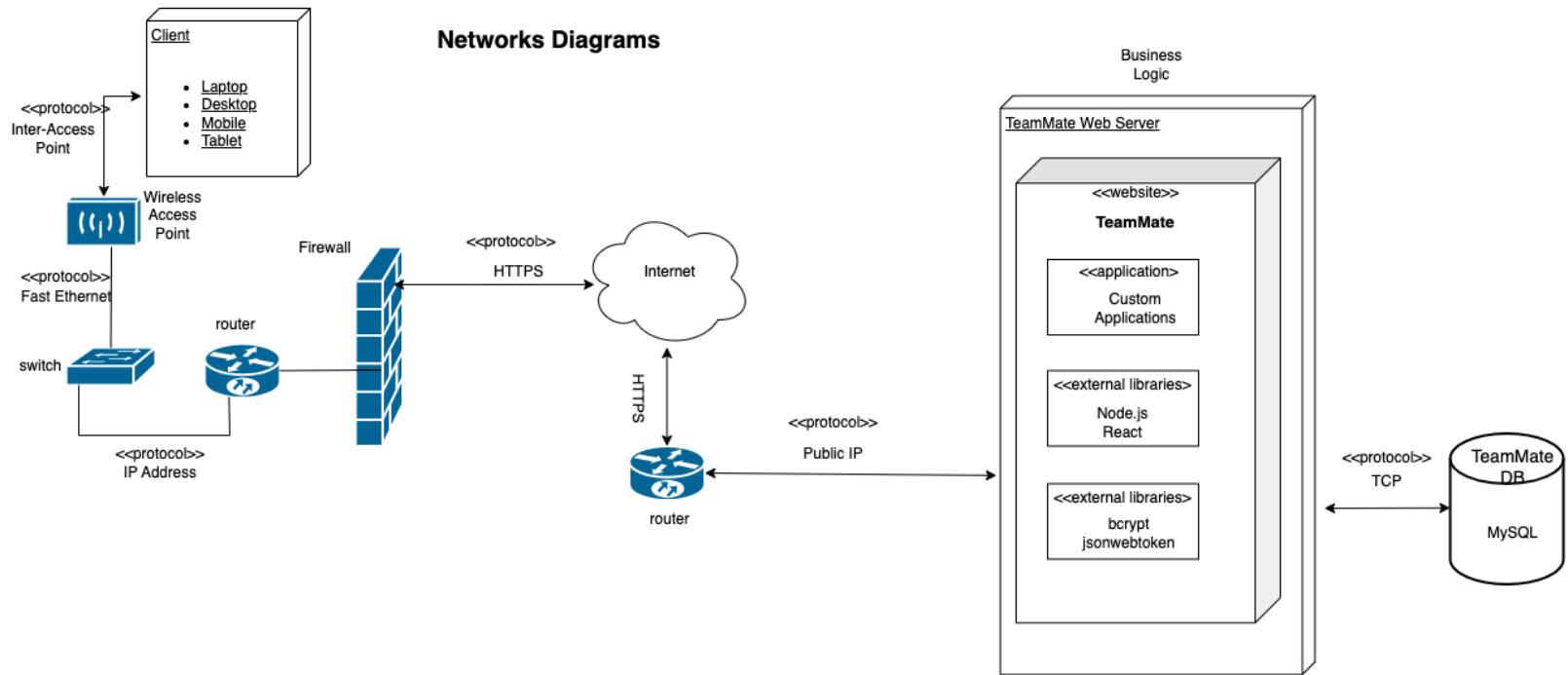
By integrating horizontal and vertical scaling methods, utilizing robust database management, and implementing message queuing for efficient communication, we can ensure that our monolithic architecture remains scalable, cost-effective, and maintainable. This design leverages AWS's capabilities to provide a reliable and scalable infrastructure that meets the needs of our users.



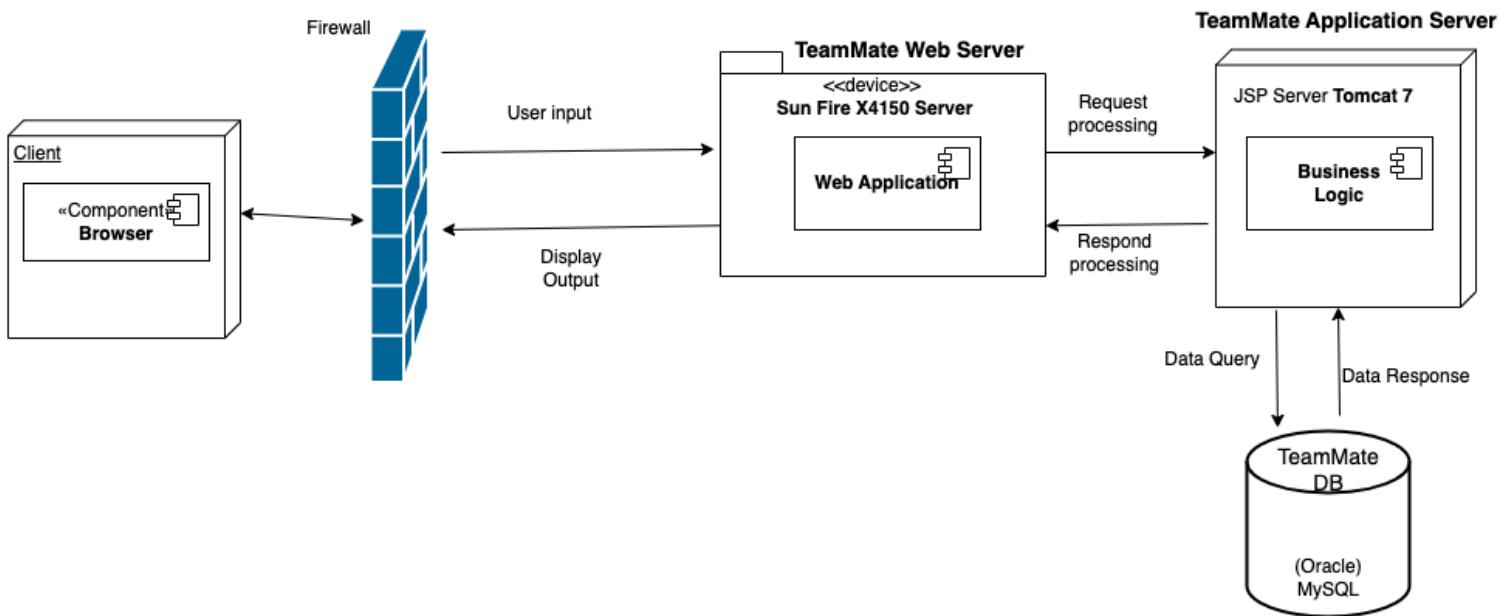
We are primarily using interfaces rather than classes. The interactions between the interfaces and the database tables are identical, and the database tables are well described in the EER (Enhanced Entity-Relationship) diagram. Below are the UML diagrams for our controllers and interfaces:



# High Level Application Network and Deployment Design



## Deployment Diagram



# Actual Risks

## Skills Risks:

A good portion of the team is lacking experience and skill in web development in-general, but also some tech specific areas like with TypeScript and React.

*We are resolving this through having our more experienced team members mentor those who need assistance, and having the inexperienced team members do some learning on their own time to get familiar with the technologies and languages.*

## Schedule Risks:

The tight deadlines due to the summer working cycle make it so we may not be able to achieve all of our goals.

*We are managing this by prioritizing our functional requirements and focusing on essential and good-to-have features, while shelving more opportunistic features and functions until later to ensure we have a viable product first.*

## Technical Risks:

There is some issue preventing us from accessing the RDS instance from our local machines, we currently can only access the RDS instance after SSH-ing into the EC2 instance.

*We are working on this issue and believe it is related to the AWS security groups between the EC2 and RDS instances. In the meantime however, we have resorted to learning MySQL through SSH and been able to make our prototype work through the terminal. It's not as convenient as MySQL Workbench, but it works for now.*

# Project Management

For milestone 2, and going forward, the team was broken up into three sub-teams; these sub-teams were the front-end team, back-end team, and the database team. These teams were assigned tasks in a Zoom meeting, and a complimenting Notion teamspace was set up to reflect the tasks ahead. From there, teams and individuals were able to claim/assign themselves to tasks and update their progress there. The teamspace included internal deadlines to keep the team on-track.

<https://www.notion.so/team/6382e2bc-5cea-4c8f-b3ca-ba076d604083/join>

Aa Task name	Status	Assignee	Due
▶ <b>Backend Coding Tasks</b>	In progress	Randolf Gabrielle	July 6, 2024
▶ <b>Data Definitions</b>	In progress	Tharun Krishna	June 30, 2024
▶ <b>Prioritized Functional Requirements</b>	Revising	Tharun Krishna	June 30, 2024
▶ <b>UI Mockups and Storyboards</b>	Done	Tyron Cheung	June 30, 2024
▶ <b>High Level Database Architecture</b>	Done	Tony Guan	June 30, 2024
▶ <b>High Level APIs and Main Algo</b>	Done	Randolf Gabrielle	June 30, 2024
▼ <b>System Design</b>	In progress		June 30, 2024
▶ <b>Scalability Diagrams</b>	Done		
▶ <b>Scalability Summary</b>	Done		
▶ <b>UML Class Diagram</b>	In progress		
▶ <b>Design Pattern Summaries</b>	In progress		
+ New sub-item			
▶ <b>High Level Application Network</b>	Done	Jazz Naw	June 30, 2024
▶ <b>Identify Actual Risks</b>	In progress	Tharun Krishna	June 30, 2024
▶ <b>Project Management</b>	In progress	Tharun Krishna	June 30, 2024
▶ <b>Minimum Viable Prototype Version</b>	In progress		July 9, 2024
▶ <b>Detailed List of Contributions</b>	In progress	Tharun Krishna	July 9, 2024
▶ <b>Feedback Email</b>	In progress	Tharun Krishna	July 9, 2024

# Team Contributions

- **Tharun Krishna**

- Transcribed Prioritized Functional Requirements from team meetings.
- Transcribed and updated Data Definitions through meetings.
- Identified Actual Risks for the project, and transcribed it to documentation.
- Documentation write up for the project management section.
- Notion Teamspace creation and setup.
- Migration and setup of project repository to EC2 and RDS instances.
- Transcribing UML Diagrams from Backend Team's concepts and ideas.
- Documentation formatting and consistency.

- **Tyron Cheung: Contributions Score 10**

- Created twelve Storyboards.
- Created twelve Mockups.
- Leading efforts on Frontend of the MVP.
- Created the frontend Login page.
- Created the frontend Registration page.
- Frontend debugging and revisions.
- Frontend search component.

- **Randolf Gabrielle Uy: Contributions Score 10**

- Reorganized GitHub repository.
- Created System Design Summaries.
- Created Scalability Diagrams.
- Created Scalability Summary.
- Created Design Pattern Summaries.
- Created Scalability Summary.
- Created High Level API Summary and API Flow Diagrams.
- Created Non-Trivial Algorithm Descriptions (Search).
- Leading efforts on Backend of the MVP.
- Backend Interfaces for Database Connection.
- Backend Controllers for Interfaces.
- Backend Search Component.
- Created Login Backend.
- Created Registration Backend.

- **Carlos Hernandez: Contributions Score 10**
  - Created Database Requirements.
  - Created writeup of DBMS definition.
  - Created Database Design Description.
  - Created Entity Relationship Diagram (ERD).
  - Created Entity Establishment Relationship (EER).
  - Forward Engineered EER.
  - Transcribed Media Storage decision from meeting.
  - Assisted Backend team.
  - Created SQL Queries for backend integration.
- **Xuefeng Guan (Tony): Contributions Score 7**
  - Contributed to ERD attributes and entities.
  - Contributed to frontend dashboard page.
  - Contributed to frontend members page.
  - Contributed to frontend styling.
  - Contributed to frontend sidebar.
  - Contributed to various miscellaneous frontend MVP features.
- **Naw San: Contributions Score 8**
  - Created Networks Diagram.
  - Created Deployments Diagram.
  - Created Frontend Header.
  - Created Frontend Footer.
  - Created additional Frontend Headers.
  - Contributed to frontend styling.
- **Yasmine Sucu: Contributions Score 7**
  - Contributed to backend interface controllers.
  - Contributed to backend page routing.
  - Contributed to backend APIs
  - API and System Design documentation polishing.
  - Documentation formatting pass.
  - Documentation spell and grammar check pass.
- **Jannelly Hernandez Umanzor: Contributions Score 7**
  - Contributed to various additional miscellaneous tasks.
  - Contributed to backend APIs.
  - Contributed to backend database queries.
  - Contributed to backend interface controllers.
  - Supervising GitHub commits and branches.