

Final Project for SW Engineering Class

CSC648-848 Section 01 Summer 2024

Team 05 - Concat

TeamMate - Personnel Management

Name	E-Mail	Role
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

Demo Link

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/>

Milestone 5

August 01 2024

Revision	Date Submitted
M1	13 June 2024
M1V1	20 June 2024
M1V2	09 July 2024
M2V1	09 July 2024
M2V2	23 July 2024
M3V1	23 July 2024
M3V2	30 July 2024
M4V1	30 July 2024
M4V2	01 August 2024
M5	01 August 2024

Table of Contents

<u>Product Summary</u>	3
<u>Committed Functions</u>	4
<u>Application</u>	5
<u>Milestone Documents</u>	6
<u>Post Analysis</u>	7
<u>Team Contributions</u>	8

Product Summary

TeamMate is a comprehensive personnel management solution designed for small to mid-sized groups, such as non-profits and community organizations. It consolidates essential management functions into a single, user-friendly platform, enabling easy registration, member tracking, event management, and award issuance.

TeamMate's standout and superior feature is its Automated Recognition and Progress Tracking System. This system automatically tracks member participation, awards achievements, and updates progress, providing timely recognition and broad overviews of individual and group accomplishments. This holistic view is valuable for performance reviews, planning, setting future goals along with motivating the members of the organization.

Discord focuses on communication and lacks recognition and structured progress tracking, **Airtable** and **Google sheets** require extensive manual setup to achieve similar functionality, whereas **TeamMate** offers an out-of-the-box solution that simplifies administrative tasks. **PERSCOM** provides good tracking but demands more manual effort for recognition, while **Rippling HCM** caters to larger organizations with complex and costly systems, making it less suitable for smaller groups.

Committed Functions

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 create a new organization.
- Accounts shall join an existing organization.
- Accounts shall view all organizations they have 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.
- 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 a roster page that shows all members in the organization separated by groups.
- 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.
- Admins shall directly assign awards.
- Award instances shall have a date received.
- Award instances shall state the issuing member.

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.
- 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.
- Admins shall assign members to an existing event.
- Group leaders shall assign members to an existing event.
- Members shall be able to join an event.
- Events shall have a title.
- Events shall have a description.
- Events shall have a start datetime.

Application

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/>

Milestone Documents

Attached below are the version 2 pdfs of all the milestones up to milestone 4.

SW Engineering CSC648-848-05 Summer 2024

TeamMate Personnel Management Team 05 - Concat

Team Members

Name	E-Mail	Role
Tharun Krishna	tkrishna1@sfsu.edu	Team Lead
Tyron Cheung	tcheung@sfsu.edu	Frontend Lead
Xuefeng Guan	xguan6@mail.sfsu.edu	Database Admin
Naw San	lsan@sfsu.edu	Docs-Editor
Jannelly Hernandez Umanzor	jhernandezumanzor@mail.sfsu.edu	Github Master
Yasemin Sucu	ysucu@sfsu.edu	Backend Lead

**Milestone 1
July 09 2024**

Version History

Revision	Date Submitted
Initial Draft	13 June 2024
Revision 1	20 June 2024
Revision 2	09 July 2024

Table of Contents

<u>Executive Summary</u>	3
<u>Main Use Cases</u>	4
<u>Use Case #1</u>	4
<u>Use Case #2</u>	6
<u>Use Case #3</u>	8
<u>Use Case #4</u>	10
<u>Use Case #5</u>	12
<u>Use Case #6</u>	14
<u>Use Case #7</u>	16
<u>Use Case #8</u>	18
<u>Use Case #9</u>	20
<u>Use Case #10</u>	22
<u>Use Case #11</u>	24
<u>Use Case #12</u>	26
<u>Main Data Entities</u>	28
<u>Functional Requirements</u>	29
<u>Account</u>	29
<u>Member</u>	29
<u>Organization</u>	30
<u>Awards</u>	30
<u>Qualifications</u>	31
<u>Events</u>	31
<u>Custom Section</u>	32
<u>Non-Functional Requirements</u>	33
<u>Performance</u>	33
<u>Security</u>	33
<u>Coding Standard</u>	33
<u>Media Storage</u>	33
<u>Privacy</u>	33
<u>Scalability</u>	34
<u>Reliability</u>	34
<u>Usability</u>	34
<u>Compliance</u>	34
<u>Expected Load</u>	34
<u>Fault Tolerance</u>	34
<u>Database High Level Specs</u>	35
<u>Browser and OS Support</u>	35
<u>Competitive Analysis</u>	36
<u>Checklist</u>	40
<u>Technology Stack</u>	41
<u>Team Contributions</u>	42

Executive Summary

TeamMate is a comprehensive personnel management solution specifically tailored for small to mid-sized groups. Recognizing that smaller non-profit organizations and community groups often lack the resources or need for full-scale professional HR services, TeamMate fills this gap by providing an accessible and user-friendly platform. Designed for less formal environments, TeamMate offers a range of customizable features that also make it adaptable for professional settings. Instead of juggling multiple services to handle various aspects of personnel management, TeamMate consolidates these functions into a single, streamlined platform.

TeamMate is designed to provide leadership with a comprehensive overview of their organization's members, enabling them to quickly assess activities and track progress throughout each member's tenure. The platform showcases individual accolades and accomplishments, making them visible to other users within the organization, while allowing administrators to issue said accolades. TeamMate includes robust event management capabilities, allowing administrators to easily set up and manage various events such as meetings, training sessions, workshops, and other organizational activities. The system tracks attendance, ensuring that all participation is documented and easily accessible for future reference. Additionally, TeamMate provides tools for monitoring and managing qualifications and certifications. Administrators can keep track of who holds specific certifications and when they were obtained.

TeamMate empowers a diverse range of smaller communities. Whether it's tracking a student's progress, a gamer's achievements, a volunteer's service hours, or a non-profit member's contributions. TeamMate not only simplifies management but also enhances transparency and accountability within the group. The system fosters a culture of recognition and appreciation by highlighting individual achievements and contributions. This recognition encourages members to take pride in their efforts and motivates them to strive for excellence. By using TeamMate, organizations can create a motivating environment that drives continuous improvement and engagement.

Main Use Cases

Actors: Dave (Customer), Chris (Dave's Friend), Scouts (Users), TeamMate (Company)

Assumptions:

Dave has information about his scouts in separate offline platforms.

Dave, Chris, and Scouts all have reliable internet access.

Use Case #1

Dave is part of the leadership for Boy Scouts of America in the silicon valley.

During COVID, the Boy Scout troops were hardly able to go outside and do their activities, Dave is worried that such events will negatively affect troop motivation and participation. A platform to display scout progress and achievements online appeals to him, as it would still allow the scouts to take pride in their achievements and showcase them without needing in-person meetings. Dave creates an account and an organization on TeamMate for all troops in the Silicon Valley. He uses the groups category to ensure each Troop has their own section, and imports all awards. As the scouts create their own account and join the organization one by one, Dave starts assigning them their awards and groups, but realizes that this is tedious work for one person. Dave gets his friend Chris to also create and account and join the organization, where he gives Chris administrator permissions. Now, Dave and Chris both assign groups and awards to the incoming scouts, and Dave is relieved that the workload is much more manageable. On the dashboard screen, Dave can see all the scouts in his organization, separated by Troops; he can click on a scout to open a page that displays information about them and their awards.

Benefits:

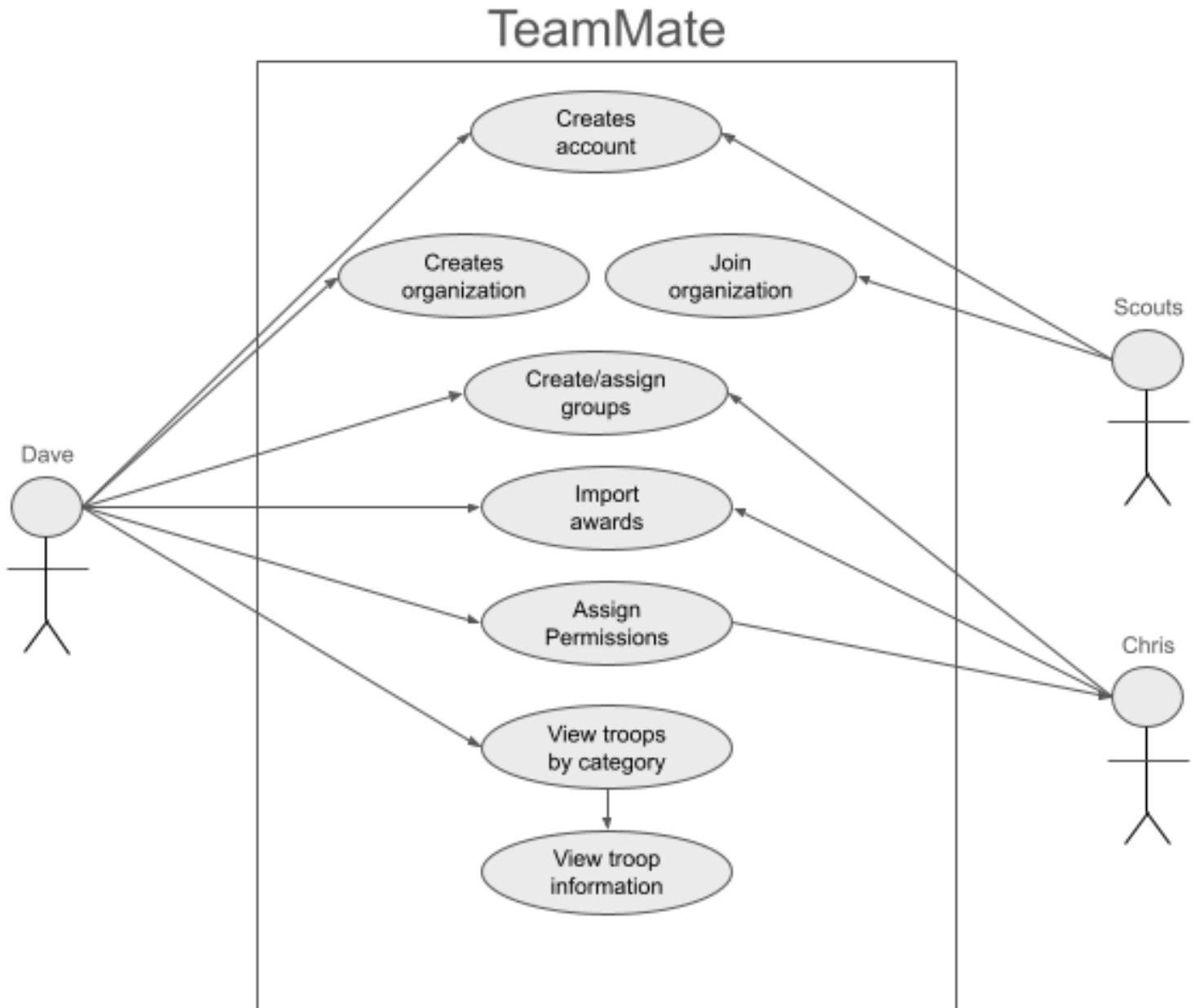
Dave and Chris can track all their scouts on one central platform.

Scouts, and their friends and family, can quickly check their progress.

Scouts feel recognized and motivated to participate more in their community.

Diagram

Use Case #1



Actors: Jake (Customer), Specialists (Users), TeamMate (Company)

Assumptions:

Jake is keeping track of his specialists certifications separately.

Jake and his Specialists have access to the internet.

Use Case #2

Jake is the leader of a professional group in a field with many qualifications. His specialists need these qualifications to be considered for work by potential customers. However, each of these certifications use their own system and platform. Jake finds it tedious and time consuming to track all of his subordinates' qualifications through the various platforms they acquired them from. In order to improve productivity and keep everything centralized, Jake creates a TeamMate account and organization. Jake's Specialists also create TeamMate accounts and join his organization. Jake then imports his Specialists' qualifications, certifications, and training into the system in order to see all of them at a glance. Jake can also schedule training for certifications, see who attends them, and update their qualifications accordingly. This centralized dashboard relieves Jake, since he no longer has to track many different systems and applications, but can instead have all required information on just TeamMate.

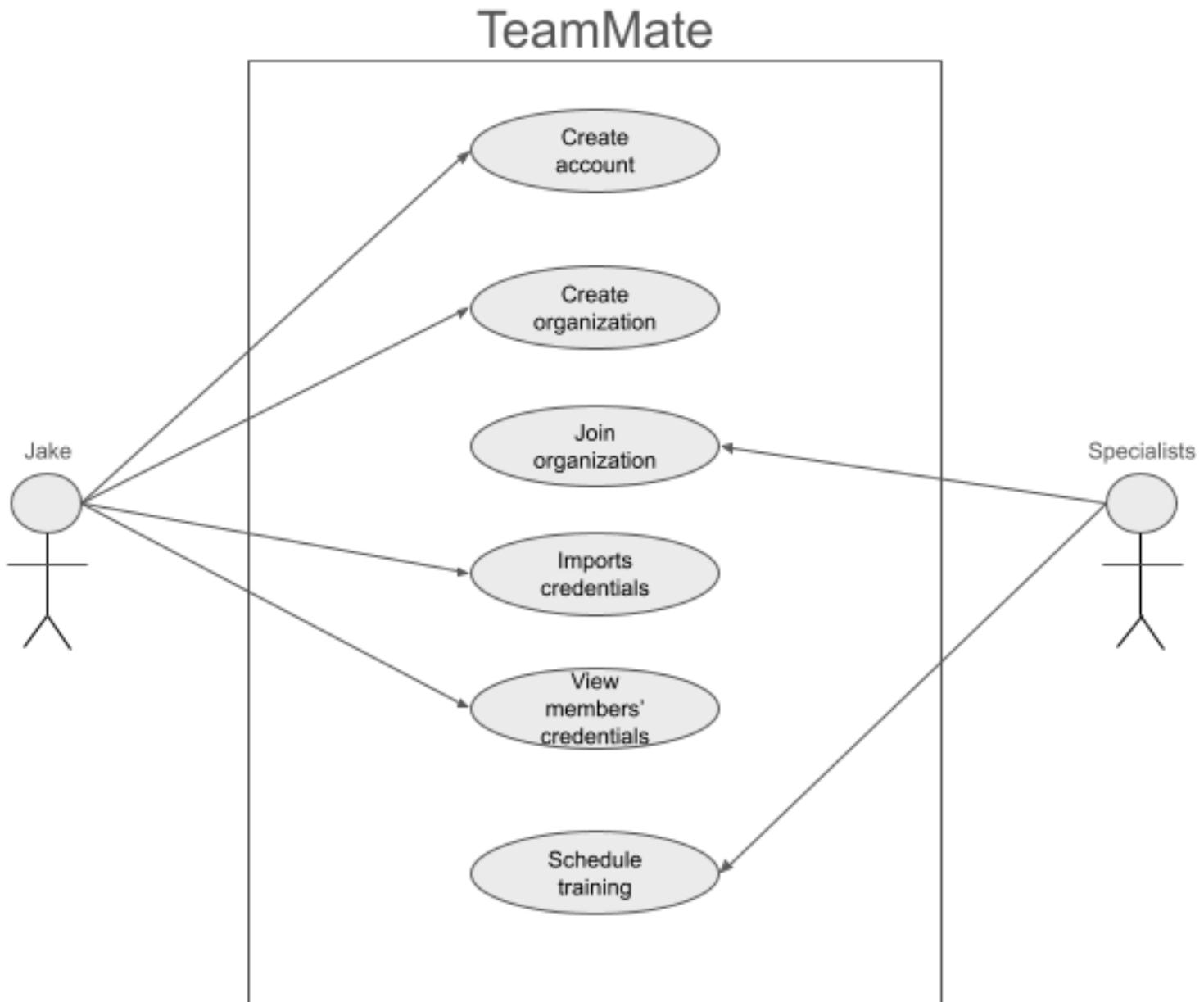
Benefits:

Jake can quickly check if his team has the qualification a customer asks for.

Jake can check which of his team members are attending the training events.

Diagram

Use Case #2



Actors: Dr. Sarah (Supervisor), Researchers (Alice, Bob), Lab Assistants (Eve, Frank), Graduate Students (Heidi, Jose)

Assumptions:

Dr. Sarah has access to the TeamMate platform

Researchers, Lab assistants, and graduate students have created their accounts on TeamMate and joined the lab organization.

Use Case #3

Dr. Sarah logs into the TeamMate platform and navigates to the custom “Project Management” section. She inputs the details of the new project, including objectives, timelines, and assigned researchers. Researchers log their progress and update project milestones regularly. Lab assistants support researchers by updating their progress on the tasks. Graduate students input their progress and updates on their specific research tasks. Dr. Sarah reviews the progress updates and provides feedback or adjusts timelines as needed. The system generates a progress report summarizing the status of all ongoing research projects and confirms the progress updates are saved and the report is generated successfully.

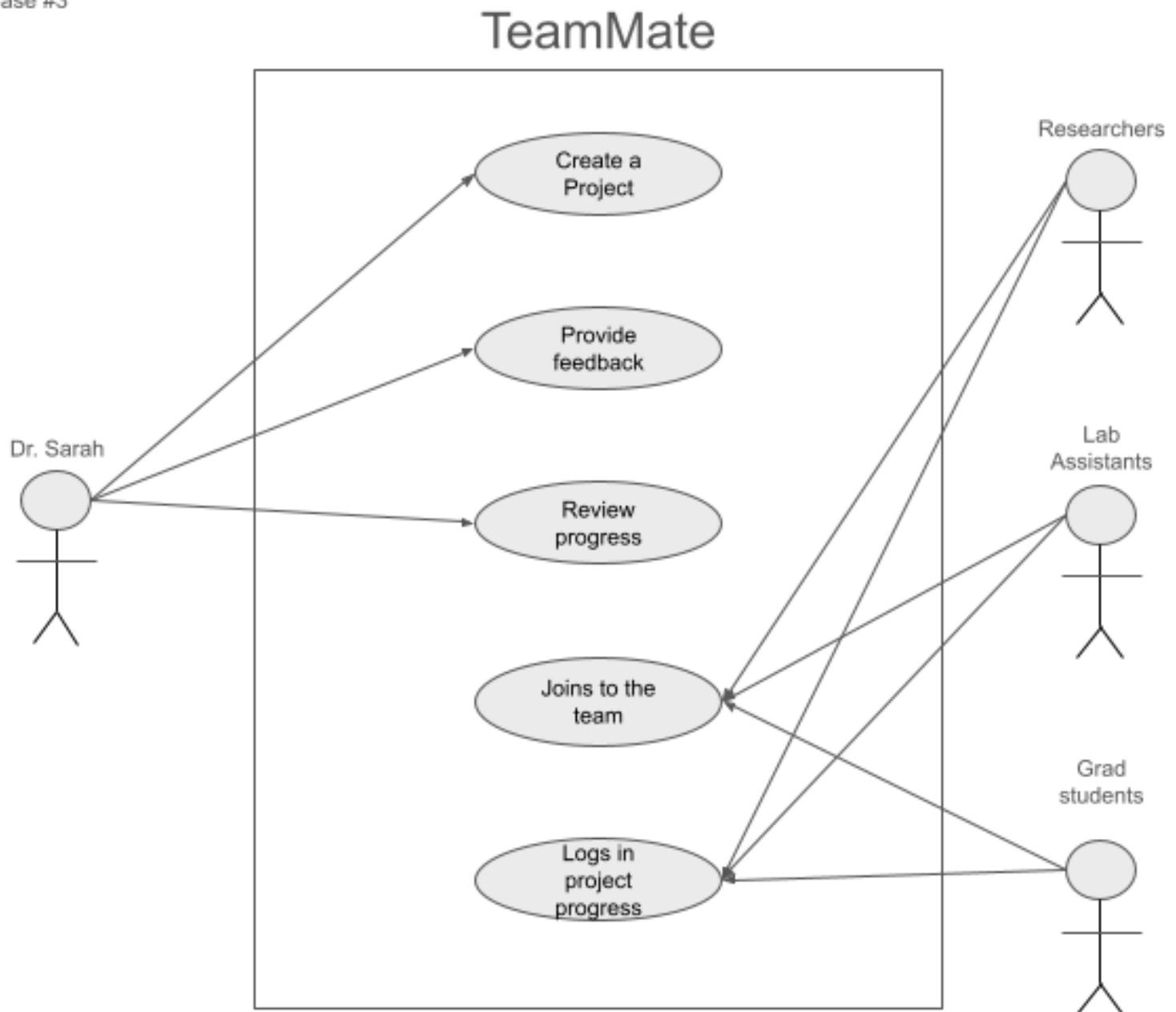
Benefits:

Dr. Sarah can efficiently monitor the progress of multiple research projects.

Researchers, Lab assistants, and graduate students receive timely feedback and support from the lab supervisor.

Diagram

Use Case #3



Actors: Dr. Susie (Supervisor/Lab Manager), Technicians(Ingrid, Jack), Maintenance Staff (Nancy, Billie)

Assumptions:

Dr. Susie has access to the TeamMate platform

The team members have created their accounts individually on TeamMate and joined the lab organization.

Use Case #4

Dr. Susie logs into the TeamMate platform and navigates to the custom “Equipment and Supplies Management” section. She inputs the details of the lab equipment and supplies, including quantities of them, locations, and maintenance schedules. Lab technicians log equipment usage and report any issues or maintenance needs. Maintenance staff schedules and performs maintenance based on the logs and reports from the lab technicians. Dr. Susie reviews the equipment usage logs and schedules maintenance as needed. The system tracks inventory levels and alerts Dr. Susie when supplies are low. Dr. Susie places orders for new supplies and updates the inventory accordingly. The system confirms the equipment and supplies logs are updated and inventory levels are tracked regularly.

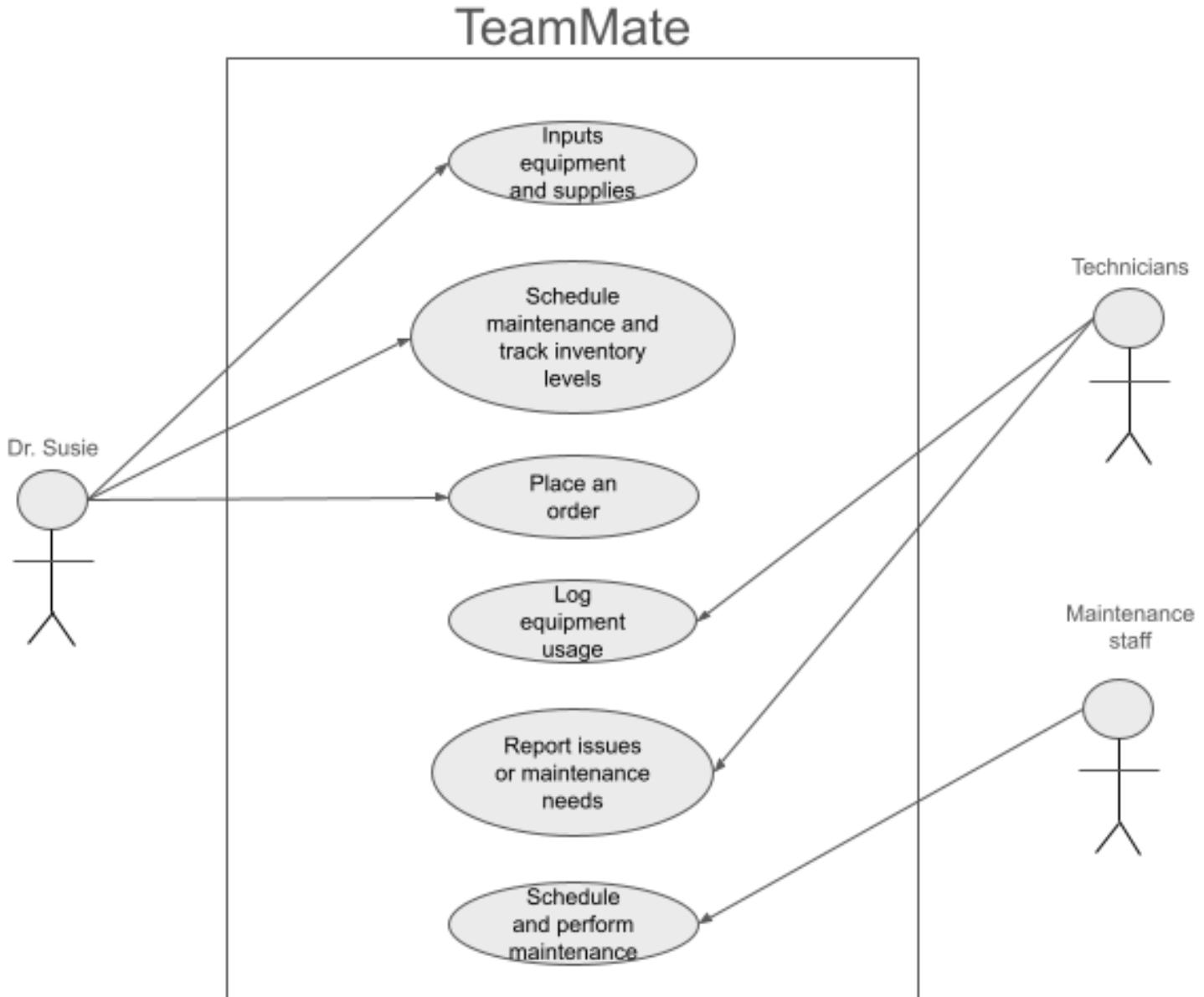
Benefits:

Dr. Susie can efficiently manage lab equipment and supplies in one centralized platform.

Equipment maintenance is timely and well documented.

Diagram

Use Case #4



Actors: Steve (Manager), XYZ staff (User), TeamMate(Company)

Assumptions:

Steve is knowledgeable in the usage of applications on websites and smartphones.

Steve and his staff have access to the internet

Use Case #5

Steve is a manager at a non-government organization, XYZ, and he's enthusiastic about recognizing activities and workflow in the organization because mostly, the body of the workforce is volunteers. At the same time, he is concerned that everyone in the company is at the same pace. In reality, it was difficult to monitor, encourage, and recognize everyone at the same time.

Meanwhile, he found out that an online platform TeamMate can track and provide a comprehensive overview of all the members, making sure no one is left behind by recognizing their activities. He can issue digital accolades to the XYZ staff to recognize their contributions, which are visible on their profiles. The TeamMate would suggest Steve or HR manager to issue digital badges when volunteers complete a certain amount of tasks or projects they were assigned. He can also share the recognition to the public via social media.

Benefits for Steve

The app saves Steve time and energy by tracking workflow and issuing digital accolades to recognize volunteers' contributions.

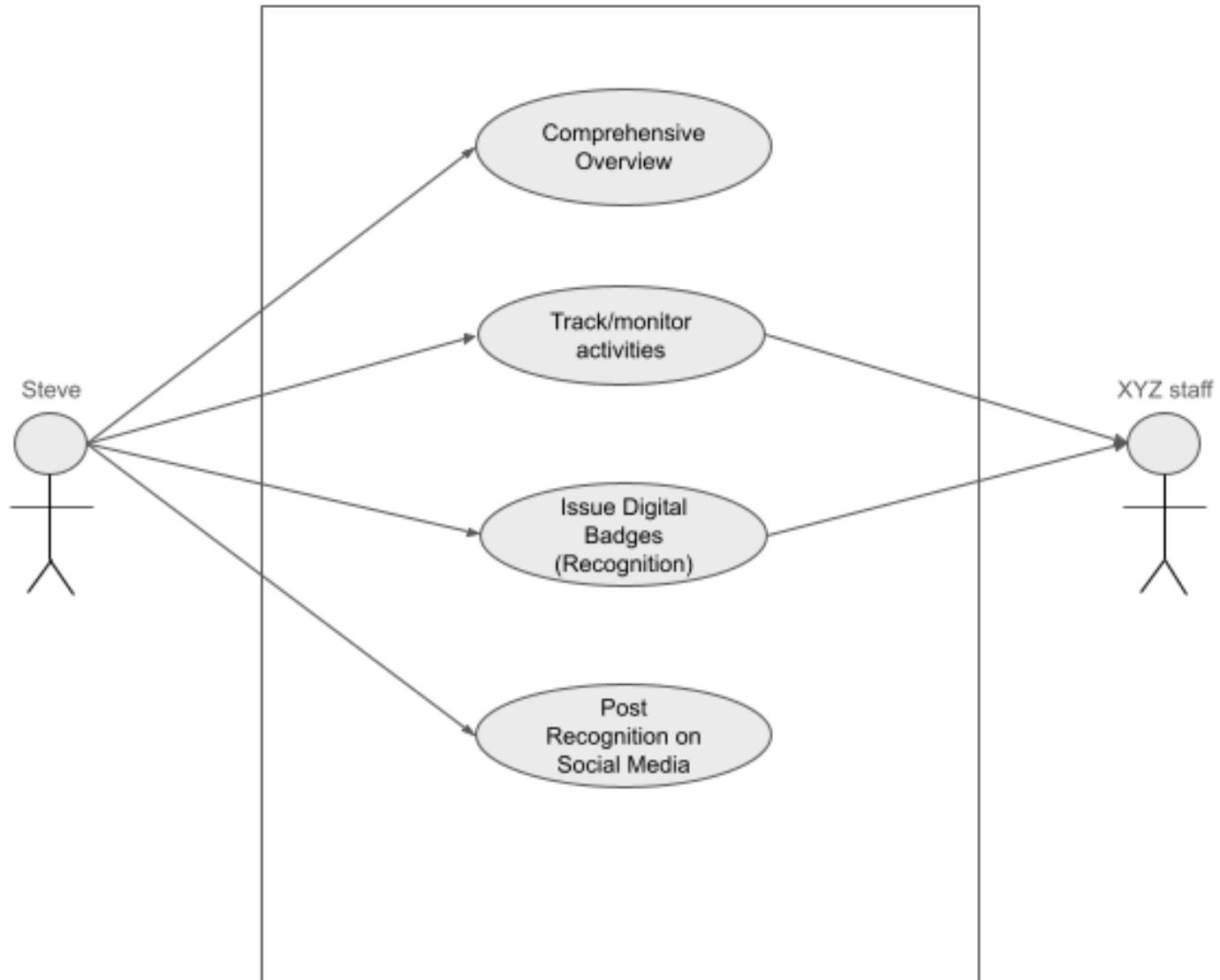
The volunteers get motivated by the culture of public recognition to continue their efforts.

Steve is happy because the way the platform transforms the volunteers more organized, motivated, and dedicated to their tasks, missions.

Diagram

Use Case #5

TeamMate



Actors: Bob (Founder of Inferno), TeamMate (Company), Teams/ players (Inferno community)

Assumptions:

Bob is willing to try the platform(TeamMate).

All the teams create accounts on TeamMate, and reliable internet is accessible.

Use Case #6

Bob is a CS graduate from SFSU and he founded a gaming community called Inferno. After one year, the community can host gaming events across the Bay Areas with active 100 members. As members grow, it is challenging to host events, organize teams, track individual achievements, and post announcements. He could have hired more people, but he doesn't want to raise membership fees. He is worried that he could not manage teams and events perfectly.

One of the members introduces him to an online team/event/organization management platform (TeamMate) that can set up events, organize teams, check qualifications, track streamlined activities, recognize achievements, and a few to name. Now the gaming community creates events and tournaments with features of signing up more teams without schedule conflicts. The platform records attendance, ensuring not just teams, but individuals have detailed profiles that include biography, participating history, level, rewards, and achievements. The administrators can also make announcements on the platform.

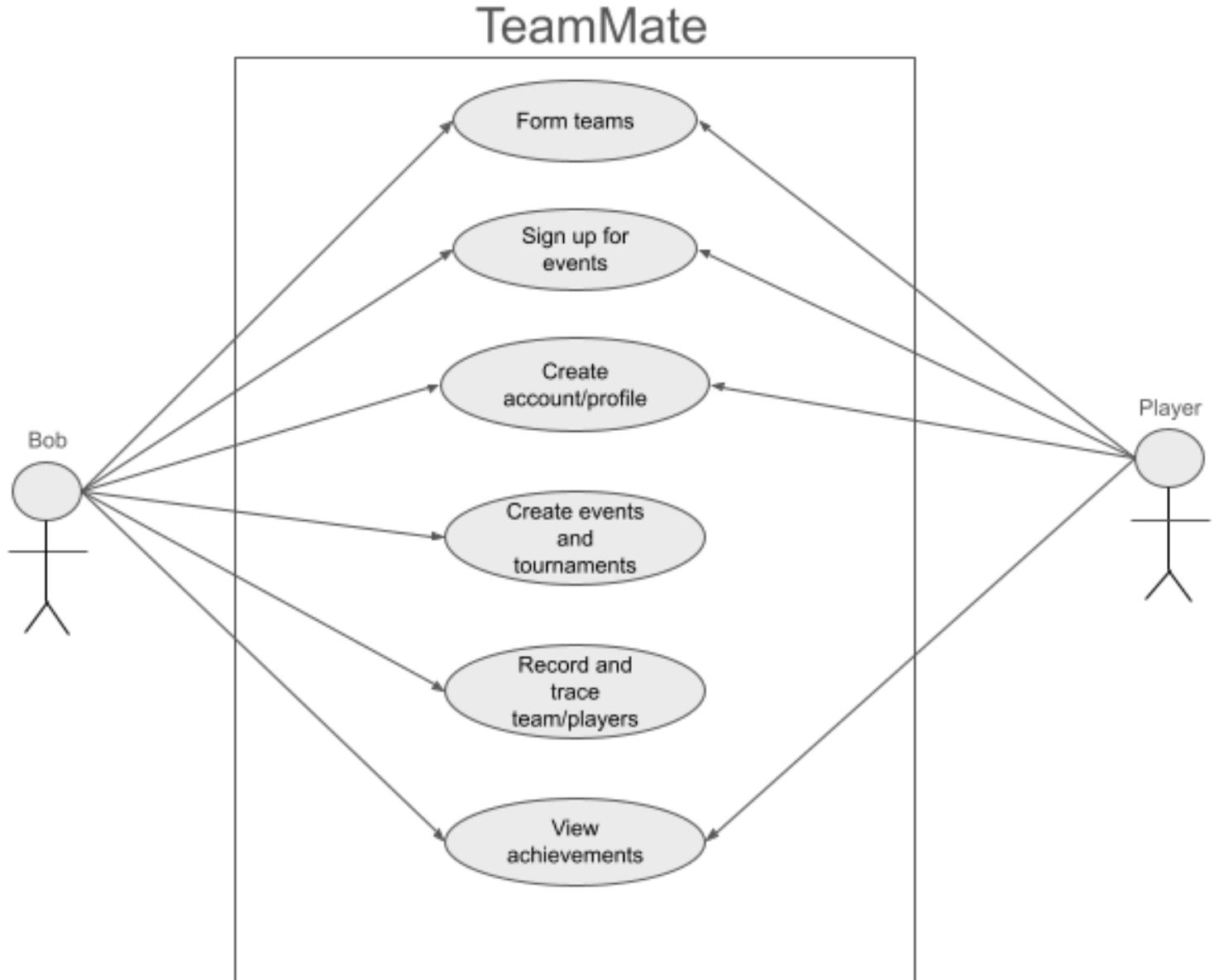
Benefits for Bob:

The online platform saves Bob resources and energy to handle teams, events and organization effectively.

Bob can access the platform anywhere and anytime.

Diagram

Use Case #6



Actors: Dan (Construction Company manager) Luigi, Angel, Joshi (Construction workers, users) Danliland (Construction Company), and TeamMate (App Company)

Assumptions:

Dan is currently having a difficult time interviewing new prospects for his construction company and knowing why other are failing to do the requested work

Dan is aware of the TeamMate App knows to use it and has introduced it to his workers

Use Case #7

Dan serves as a construction company manager at the Danliland construction company. As manager he employs at least 20 workers in his company. Some of his workers are Luigi, Angel and Joshi, as construction workers Dan asks them to complete a training survey in order to track what the employers need to learn about construction, what they already know, but also emphasize what they have learned in the past about construction working with other companies. Dan uses the TeamMate App to create an account for each worker so they can login and complete the survey within it. Having the name of each worker, the email, and experience in construction keep track of who has completed the survey and who has not, also track what needs to be learned while the workers work. With the app the construction manager is also able to manage when the survey was opened or completed. Depending on survey results some workers will have to meet with Dan

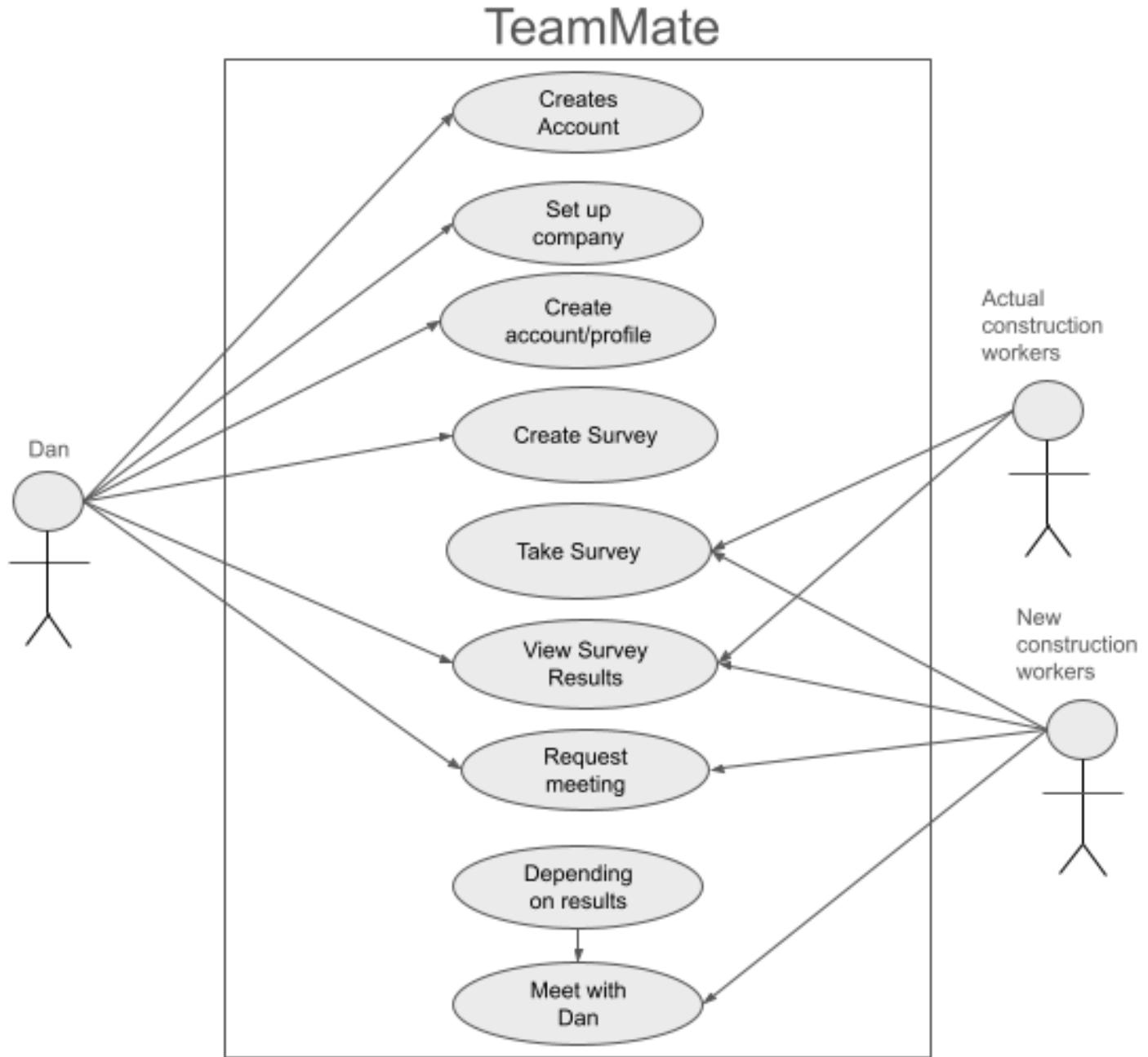
Benefits:

The TeamMate App is going to help Dan have a more organized life when he asks new and current employers to complete the work survey because he will keep track of what each worker needs to learn about construction, what they already know about construction, and what knowledge they plan to take out while working in a construction company

The TeamMate App can help Dan learn more about each of his workers (background wise) in relation to the experience they currently have in construction

Diagram

Use Case #7



Actors: Mary (Customer), Volunteers (Users), TeamMate (Company)

Assumptions:

Mary is currently tracking her volunteer activities and hours manually or using multiple tools

Mary and her volunteers have reliable internet access

Use Case #8

Mary works as a coordinator at a local non-profit organization that relies on volunteers to carry out and support its mission. She finds it overwhelming and challenging to keep track of the volunteers' hours, activities, and accomplishments using tools such as spreadsheets and manually keep track of these processes. In desperate need for a solution that can help streamline this process and provide a clear overview of each volunteer's contribution, her search begins, and she comes across the website, TeamMate. Mary creates an account on TeamMate and sets up her organization. Mary then is able to import historical volunteer data which includes their hours worked, tasks completed, and any awards they achieved. TeamMate allows her to easily assign new tasks to volunteers, track their hours in real time, and manage schedules for upcoming events. Volunteers can create an account, log their hours directly into the system, review their progress, and receive recognition for their contributions. Mary can also generate reports to showcase or highlight the volunteers' efforts which help recognize their hard work and secure future funding for the organization.

Benefits:

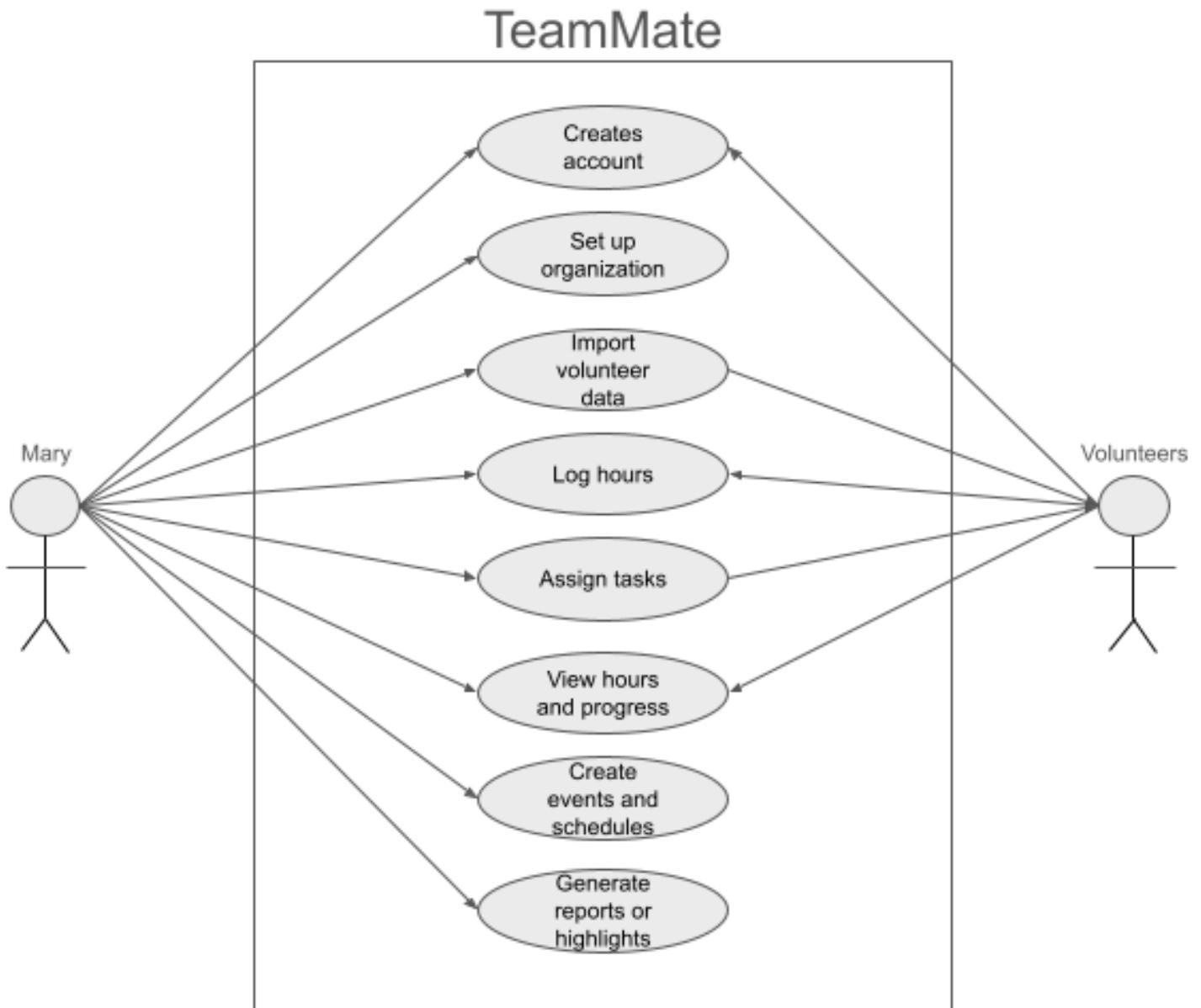
Mary is able to efficiently manage all volunteer activities and hours in one centralized platform.

Volunteers can easily log their own hours and view their contributions which can promote transparency and recognition.

The organization within TeamMate can provide detailed reports or highlights on volunteer activities that can further aid funding or grant opportunities.

Diagram

Use Case #8



Actors: Alex (Customer), Students (Users), TeamMate (Company)

Assumptions:

Alex is responsible for managing the school's gaming club

Alex and the Students have reliable internet access

Use Case #9

Alex is a teacher who is in charge of running a gaming club at a local high school. The club has recently grown in popularity, and many students participate in various games and competitions. However, it is becoming increasingly complex to manage the club with the increase of participants in the club. Alex needs to organize teams, track the players' rankings, and continuously find new opponents for the club to compete against.

Alex comes across TeamMate in his search for a tool to streamline the management of the gaming club. He creates an account on TeamMate and sets up the club's organization. Students create their own TeamMate accounts and join the gaming club. Alex can then create teams, assign players to these teams, update scores, and track individuals and team rankings based on performance in competitions. This platform also allows Alex to schedule matches, find new teams to compete against, and organize internal tournaments. The students can view their rankings, see their team assignments, and sign up for upcoming competitions. They are also able to communicate with their teammates and receive updates about matches and events. This centralized system allows Alex to efficiently manage the club while the students can stay informed and motivated to participate.

Benefits:

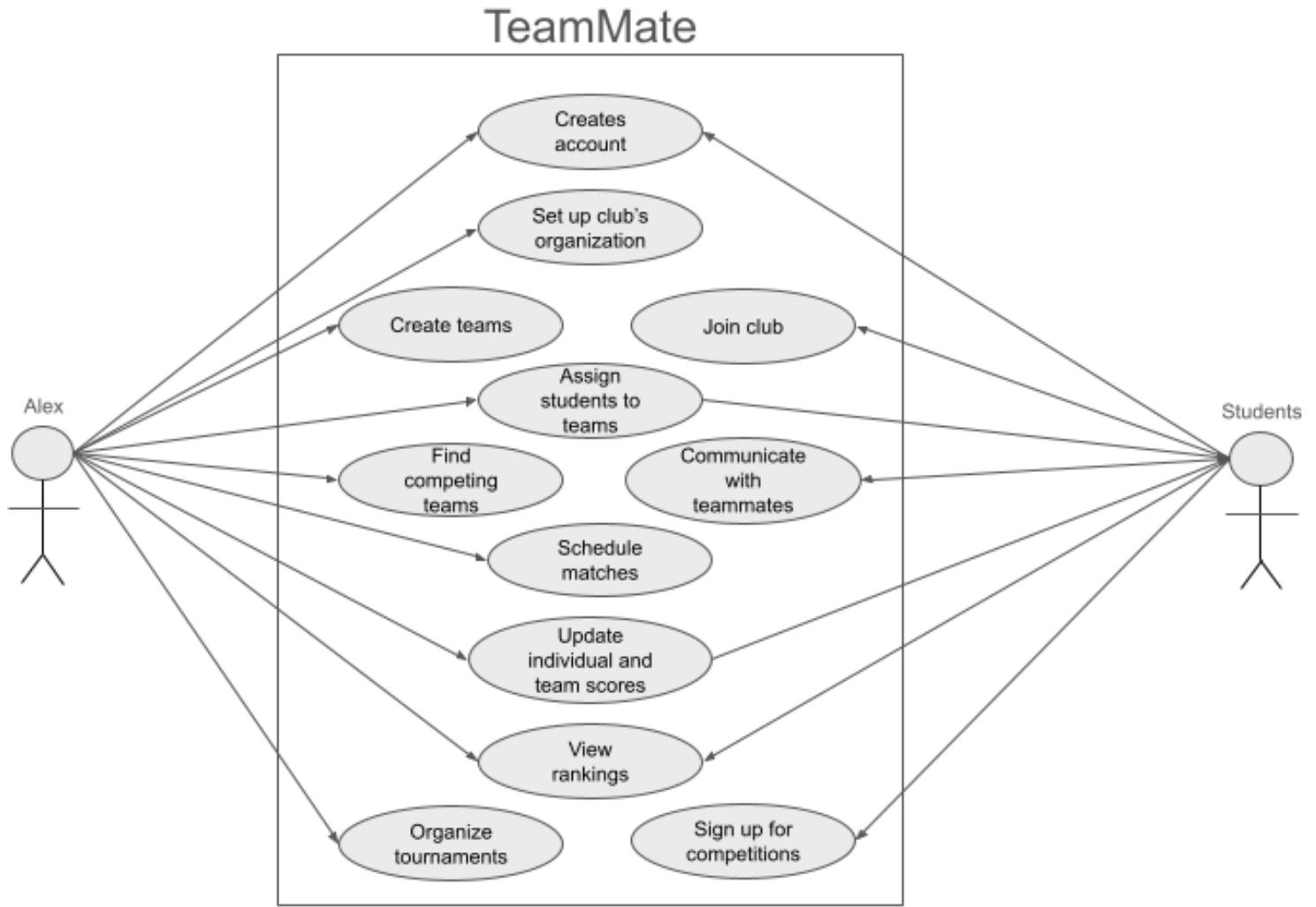
Alex can efficiently manage team assignments, track rankings, and organize competitions in one platform.

Students can easily view their ranking, team assignments, and view upcoming competitions.

The club can find new teams to compete against, enhancing the overall gaming experience.

Diagram

Use Case #9



Actors: John (Customer, a sport team leader), John's Teammates (Customers, sport team teammates), TeamMate (Company)

Assumptions:

John is a full-time basketball team leader, but he has a lot of part time jobs so it makes it hard to use the little time he has to manage the team he has.

John and his teammates have access to the internet and the TeamMate platform.

Use Case #10

John is a full-time basketball team leader, who is trying to get his team in the professional league. But right now he's running multiple part times to make sure the team has enough fundings. While the team is benefited by the extra money they have, John is having an extra hard time managing the team. Such as, to manage the team activities, setting up a training time and locations, keeping track of everyone's physical condition or even having group communications.

One member of the basketball team introduced John TeamMate, it was an app to make managing more convenience that promises to be a life changer. John created an account on this app, and setted up his organization. His teammates always create an account and join the organization. Then he can conveniently assign tasks for his team to complete, keeping track of the team progress, making announcements, or even rewarding the players who actively complete tasks. With the help of this app, John and his team feel more connected to each other. Players within the teammate will compete with a positive environment, also encouraging players to be more engaged..

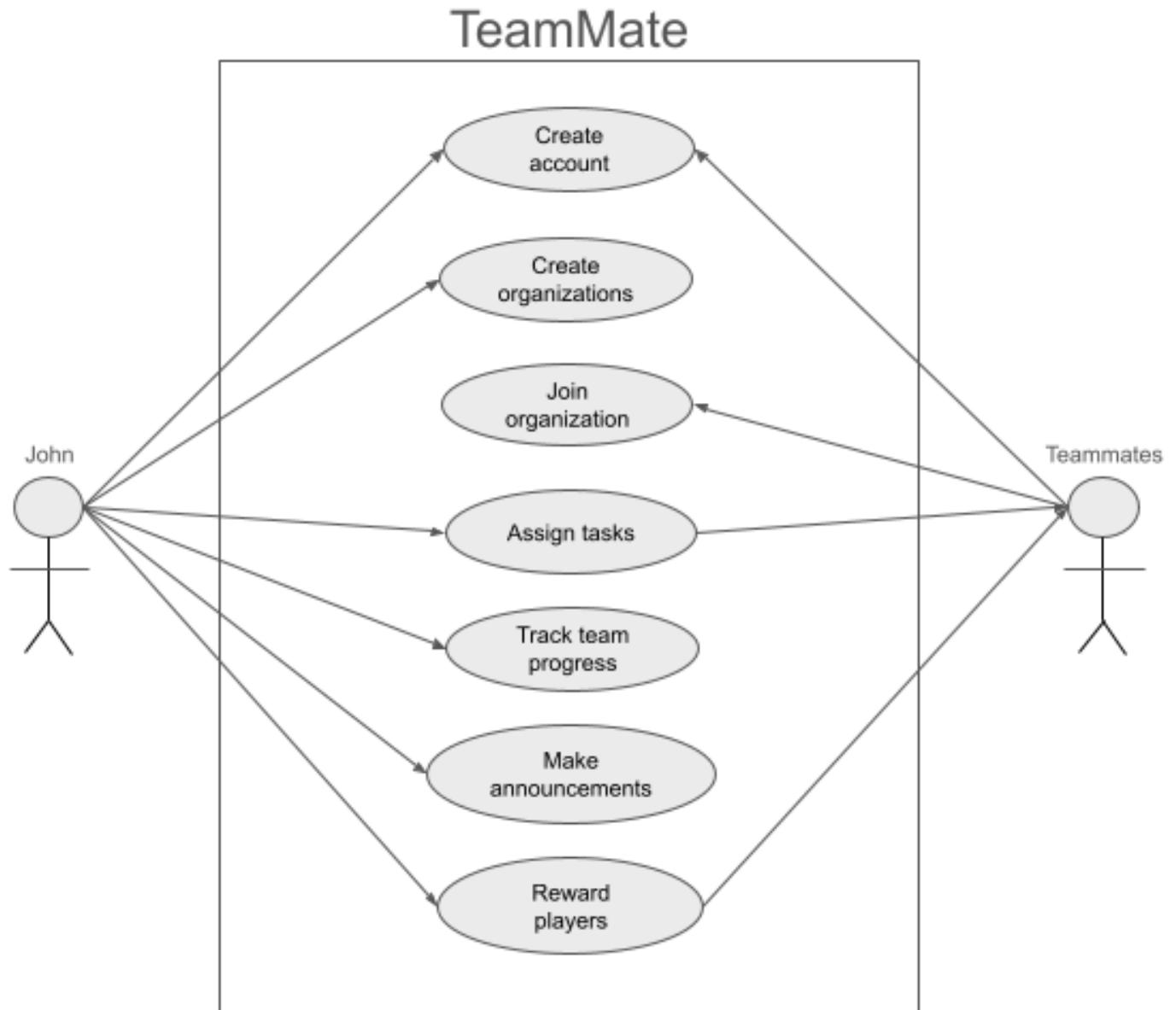
Benefits:

John can use TeamMate to manage the whole group, which it was hard to do with one app before. Increasing efficiency.

The features in the app keeps the team more connected to each other, which can potentially increase morale.

Diagram

Use Case #10



Actors: Eric (An orphanage Director, Customer), Faculty (Orphanage Faculty, Customers), Volunteers (Customers)

Assumptions:

The orphanage Eric runs lacks hands, he wants to hire some volunteers to help.

Everyone in the orphanage has access to a stable internet, and has basic knowledge about phone apps.

Use Case #11

Eric has a huge orphanage to direct, and it's lacking faculties to run the place. He's trying to hire some volunteers to help out, but he is concerned if he can manage all the volunteers who don't have much experience. Then one of Eric's colleagues introduced Eric the TeamMate app, which is an app to help management. This app comes just in time, Eric was feeling overwhelmed with work, and he can't manage the volunteers without help. He decides to create an account along with his faculty and volunteers. With the help of this app, which integrates communication, assigning tasks, rewarding system, and leaderboard all together, Eric should be able to manage the volunteers easier.

The use of TeamMate did create so much convenience, but Eric is still having a hard time managing due to how busy he is. Eric then lets one of the faculties create an account on TeamMate, who is assigned to be an administrator which allows him to also provide awards or assign tasks. Now with the help of his colleagues, Eric's workload is drastically decreased. Eric can even check the active volunteers, and make them the temporary leaders.

Benefits:

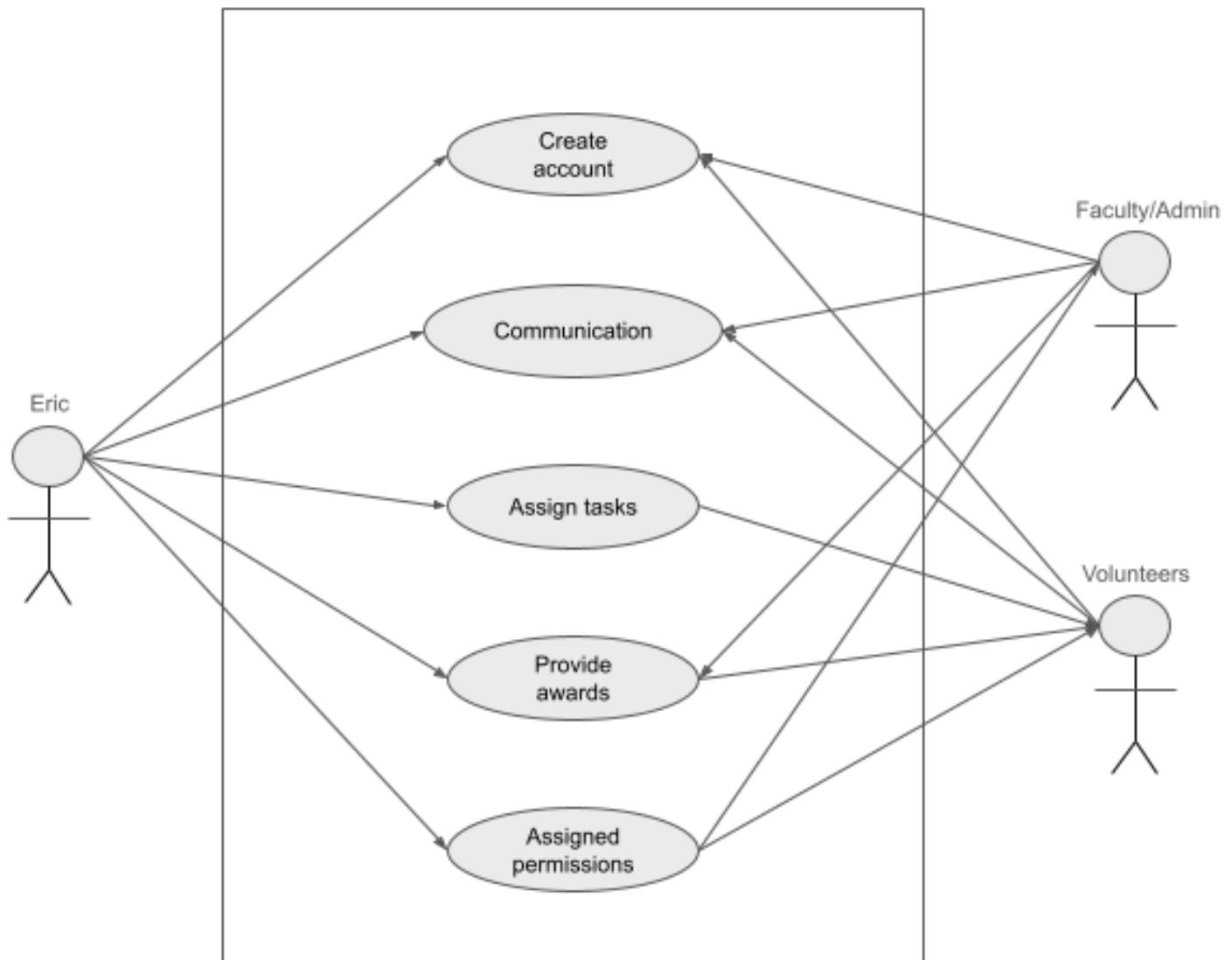
Increase the efficiency when managing a large group.

Can quickly see who's active over a group of people, and see who's not active.

Diagram

Use Case #11

TeamMate



Actors: Cianci (Yoga Center Director), Yoga students (users) TeamMate (Company)

Assumptions:

Cianci struggles to keep track of students' goals and what they need to do in order to achieve them.

Cianci has used other apps in the past to keep track of her goals but no apps seem to help her.

Students can oftenly keep track of what goal they are planning to achieve.

Use Case #12

Cianci is the director of a Yoga class in Novato, she teaches Yoga classes every other day of the week, Monday, Wednesday, and Friday. Cianci wants each of her students to have new exercises/goals for her class 1 time each month. Each goal has to be achieved by the end of the month because new goals need to be created at the beginning of each month. Cianci has students that tend to have difficulty in constantly staying motivated to achieve those goals and keep working to reach where they want to reach. She decides to use the TeamMate app and create an account for her Yoga class students to be able keep track of each student's goals and achievements during each month, also to keep track of what methods her students are using to achieve those goals. She is also able to keep track of what obstacle is preventing a student from achieving that Yoga exercise/goal. For example an obstacle can be being tardy everyday, not attending class regularly, procrastination, distractions (doing other things, other then yoga in class). She will also keep track of the days a student specifically designated to work on these achievements (one by one)

Benefits:

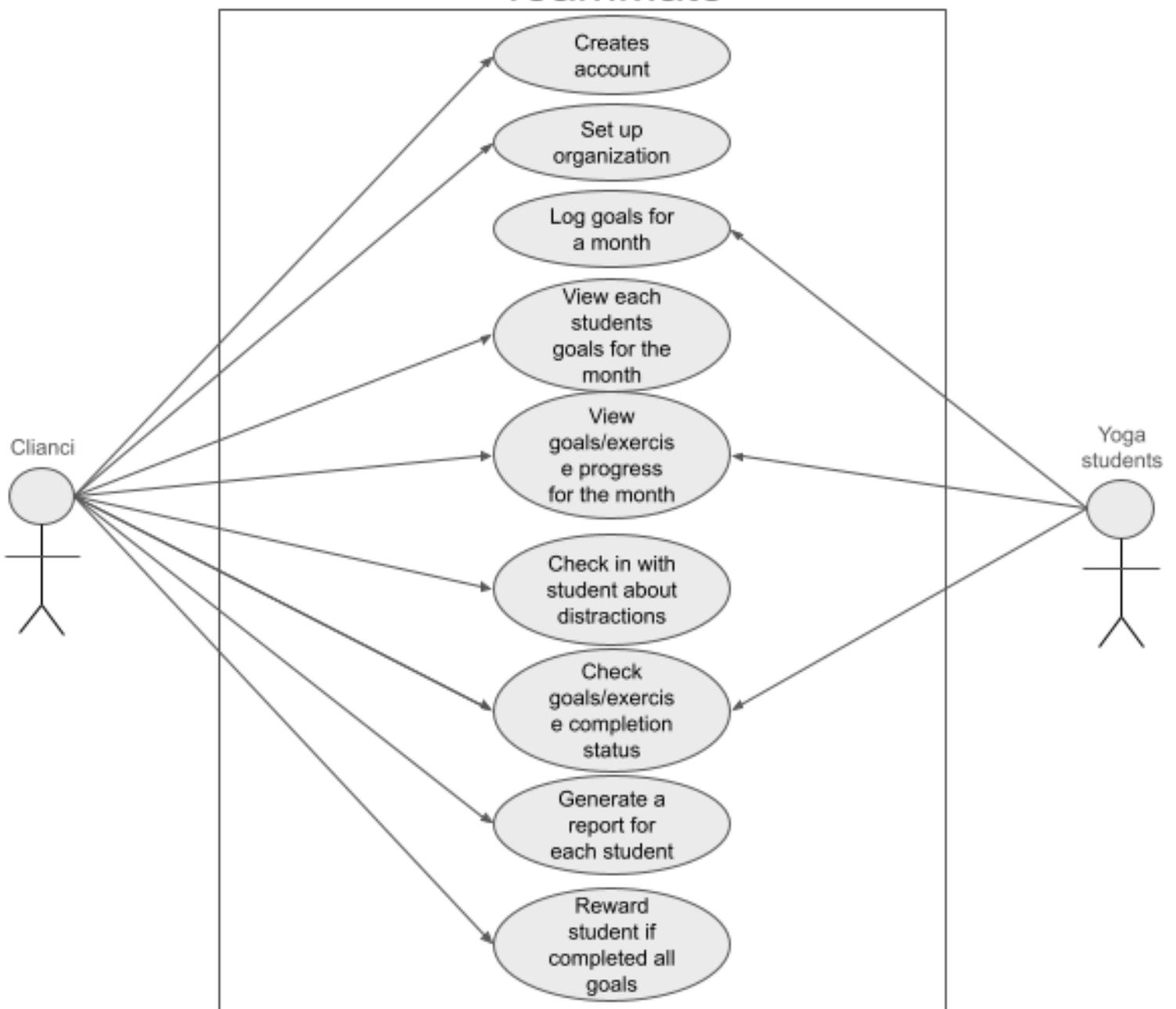
If she uses the app to keep track of all the recent and future achievements each student of hers has, she can clearly see how much they have done during the month, it gives a sense of satisfaction to see how much her students are able to complete during a month or guides to implement changes in her class that would help the student be more engaged and complete their exercises/ achievements .

She will learn how to be more organized. All students' goals/exercises have to be completed one by one depending on what motivates them and how much time students designate each day to work towards that certain goal. As soon as they achieve the accomplishment of the goal she will reward each student with something (if the goal is reached before the month ends)

Diagram

Use Case #12

TeamMate



Main Data Entities

- Account - Store user data.
 - Used to store data like username, email, password, etc to allowed users to register and login.
- Organization - Store organization data.
 - Used to store organization information, and also customization data, allowed people to rename.
- Member - Store organization member data.
 - Used to associate an account with an organization, holds organization specific information. Allowed to have one per-account per-organization.
- Events - Stores instances of events.
 - Used to store event information data, dates, times, and attendance.
- QualificationTypes - Stores types of qualifications and certifications.
 - Used to store different types of qualifications. Separated from an instance to allow easy mass-editing and issue.
- Qualifications - Stores instances of qualifications and certifications.
 - Used to store instances of qualifications, including date and time received along with any optional information about issuing party and notes.
- AwardTypes - Stores types of awards.
 - Used to store different types of awards. Separated from an instance to allow easy mass-editing and issue.
- Awards - Stores instances of issued awards.
 - Used to store instances of awards, including date and time received along with issuing party, citation, and any additional notes.
- Sections - Stores custom sections and pages for any purpose.
 - Used to store custom sections, which will be displayed on the dashboard, that invited members can access external and embedded apps on.
- Comments - Stores comments and their corresponding information.
 - Used to store comments and what page they are for, which will be displayed on their respective pages.

Functional Requirements

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 upload a profile picture.
- 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.
- Members shall overwrite their account profile picture with a new picture.
- 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.
- Admins shall assign member and group leader permissions.
- Admins shall be able to issue awards to members.
- Admins reset member profile pictures if they are inappropriate.
- Admins reset member names if they are inappropriate.
- Users shall be able to leave comments and endorsements on a member's page.
- Users shall be able to filter for members that have specific attributes.
- A member's page shall show the last time they were logged-in.

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

Awards

- Admins shall create award types.
- Admins shall update award types.
- Admins shall delete award types.
- Awards shall have an associated image.
- 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.
- Users shall have the option to upload an associated pdf file for display with a qualification.

Qualifications

- Admins shall create qualifications types.
- Admins shall update qualifications types.
- Admins shall delete qualifications types.
- Qualifications shall have an associated image.
- 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.
- Users shall have the option to upload an associated pdf file for display with a qualification.

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.

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

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 support external application, APIs, and embedded application integrations.
- Custom sections shall be deleted by admins if desired.

Non-Functional Requirements

Performance

- The application shall respond quickly to user prompts with minimal latency.
- The application shall be able to host/handle up to 1,000 accounts without any performance problems.
- The application shall support many active-concurrent users without significant degradation in performance.

Security

- The application shall only store hashed passwords.
- The application database shall be backed up regularly.

Coding Standard

- The application shall follow consistent coding and naming conventions throughout.
- The application shall gracefully handle errors without detrimentally affecting the rest of the user experience.
- The application shall be well documented for easy understandability for other developers and admins alike.

Media Storage

- The application shall restrict profile pictures to 500x500 pixels and 5 megabytes or less.

Privacy

- The application shall have the option for organizations to be set as private, preventing users not in the organization from viewing it.
- Data shall be anonymized and pseudonymized wherever possible to minimize damage in the event of a leak.
- Only the minimum amount of data required shall be stored in order to minimize the risk of security breaches.

Scalability

- The application shall be designed to scale efficiently to handle an increasing number of users and data without performance degradation.
- The database shall scale down if the workload is low to remain cost effective and optimize resource utilization.

Reliability

- The application shall ensure high availability, with a 99.9% uptime, and incorporate robust error-handling mechanism to maintain stability.

Usability

- The application shall provide an user friendly interface, ensuring that users can easily navigate and use all features without extensive training.

Compliance

- The application shall comply with relevant industry standards and regulations, such as for data protection and privacy.

Expected Load

- The application shall respond to user queries within 500 milliseconds for requests.
- The application shall handle a minimum of 1000 transactions per second during peak hours.
- The application shall horizontally scale to support up to 10,000 concurrent users without degradation in response time.

Fault Tolerance

- The application shall recover from a server failure within 15 minutes without loss of data or interruption of service.
- Critical components of the application shall have a backup system that can take over within 5 minutes of a primary system failure.
- The application shall log all errors and provide automatic recovery for non-critical errors within 60 seconds.

Database High Level Specs

- The database shall use appropriate indexes to optimize query performance, especially for frequently accessed data.
- The database should be capable of storing up to 1 terabyte of data efficiently, with automatic data compression for storage optimization.
- All sensitive data stored in the database should be encrypted.

Browser and OS Support

- The application shall support Chrome Stable 126 (126.0.6478.153).
- The application shall support Windows 10 22H2 (19045.4651).
- The application shall support Windows 11 23H2 (22631.3737).
- The application shall support macOS Sonoma (14.5).
- The application shall support Amazon Linux (2023.4.20240611.0).

Competitive Analysis

Feature/ Company	Discord	Airtable	Google Sheets	PERSCOM	Rippling HCM
Strengths	<ul style="list-style-type: none"> - Learning curve is not steep. User interface is simple and clear. - Provides customizability like creating bots - Provides easy communication among the group, faster response time than emails - Provides easy management because of the features like assigning moderators or admins - Easy and fast sign up - Can create different channels for different purposes. 	<ul style="list-style-type: none"> -A hybrid that combines spreadsheets and databases. -Able to import many different file types and other platform links. -UX and UI are ease of use, flexibility, design and visual, view system. -A wide range of built-in ready-to-use templates. -Real time Collaboration and audit trail. -Automate tasks -Use cases 	<ul style="list-style-type: none"> - Real-time collaboration with other users - Accessible from any devices with internet -Automatically saves and stores progress to cloud -Can integrate with other Google apps such as Google Docs, Drive, etc. 	<ul style="list-style-type: none"> - Personal profile to be able to manage various records and qualifications. - It includes tracking, calendar and event management, custom forms and data collection, various suites of communication tools, as well as the ability to integrate your personnel data into any third-party system. - Includes a roster to track information from different departments. 	<ul style="list-style-type: none"> -Comprehensive HCM solutions -Integrates HR, payroll, and IT management -Automation of administrative tasks -Strong integration capabilities -Detailed reporting and analytics -Scalable for companies of all sizes -Strong data security
Weaknesses	<ul style="list-style-type: none"> - Don't have much built in features, like payroll or leaderboard - Hard to track chat history or files. - Discord mostly need to be used with other 	<ul style="list-style-type: none"> -There is a learning curve for some advanced features -Some platform features require fees. -Time consuming to set up a database system 	<ul style="list-style-type: none"> - Complex for most users to use advances features to automate tasks. They must be familiar with formula or have experience with scripting - Does not support a more 	<ul style="list-style-type: none"> - The organization of each category is displayed in tabs, Not quick to find information - Complex interface, difficult to understand - The website doesn't suit 	<ul style="list-style-type: none"> -Can be complex to set up initially -Pricing might not be suitable for smaller size teams or for individuals - Feature set focuses entirely on professional

	softwares like google docs or google calendars to make it more efficient	when you start from scratch.	formal role or team management. You can only really give others view or edit permissions.	everyone or other purposes - Does not keep a reference of the data imputed overtime	use, not friendly for volunteer, non-profit, non-professional organizations.
Pricing	Mostly Free, only some not so essential features are charged. Features such as large file sharing, HD video calls, etc.	Mostly Free, fees required for larger applications or databases or more collaborators.	- Free for basic use, but subscription based if you want Google Workspace for additional features like video conferencing with a lot of people.	- Has three different tiers with a free one week trial. Pricing increases as the user base increases. Access to powerful API, widgets and website integration, custom subdomain, and ticket and email support are locked behind higher tiers.	- Personalized and tiered pricing based on features and company size.
Social Media	- Very active on Twitter, YouTube, and other famous platforms. Lots of additional user generated content and awareness.	-They are on a number of famous sites, like Linkedin, Facebook, Instagram and so on. And they post or update their blogs regularly.	-Active on Twitter, Facebook, LinkedIn, and Youtube	Community Forums, Submitting a support ticket, giving feedback, submitting feature requests	- Active on social media, Linkedin, facebook, and Twitter - Community forums and help centers
Onboarding	- Simple and fast. As easy as creating an account and inviting people to the group	- Simple and easy. A user can sign up with username and password. And the user can sign in with google or other platform accounts.	- Simple since most people already have Google accounts. There are also a bunch of tutorials and support documentation online.	- Self-guided onboarding. Documentation included, but little support from providers.	- Comprehensive and guided onboarding with direct support and tutorials.

Feature	Discord	Airtable	Google Sheets	PERSCOM	Rippling HCM	TeamMate
Real-time Collaboration	+	+	++	+	+	+
File Sharing	++	-	-	+	+	+
Text Chats	++	-	-	-	+	-
Voice/Video Calls	++	-	-	-	-	-
User Interface	+	+	+	+	+	++
Mobile Support	++	+	+	-	+	+
Customization	+	+	++	+	-	+
Role/Group Management	+	+	+	+	+	++
Reports/Analytics	-	-	-	+	+	++
Data Recovery	-	+	+	-	+	+
Event Management	+	-	-	+	+	++
Payroll and Health Insurance	-	-	-	-	++	-
Training Scheduling and Qualification Tracking	-	-	-	+	+	++
Filtering Users based on Attributes	-	+	+	-	+	++
Attendance Tracking	-	+	+	+	+	++
External Application Integration	+	-	-	++	+	+

Summary

Each of the competitors mentioned above serves and focuses on a different niche than we do. Consequently, there are some features that TeamMate either lacks entirely or offers at a level that may seem inferior compared to other competitors. This is not a disadvantage for TeamMate; in fact, it is a strength. TeamMate is designed for a specific, yet sizable, audience that currently lacks purpose-built solutions tailored to their needs. TeamMate is designed to be used alongside other applications, including some of the listed competitors like Discord, to enhance an organization's capabilities. It remains streamlined and user-friendly, avoiding unnecessary bulk that might deter potential users from adopting TeamMate as their solution.

Checklist

DONE - Team found a time slot to meet outside of the class.

DONE - Github master chosen.

DONE - Team decided and agreed together on using the listed SW tools and deployment server.

DONE - Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing.

DONE - Team lead ensured that all team members read the final M1 and agree/understand it before submission.

DONE - Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.).

Technology Stack

Server: AWS EC2.

Operation System: Amazon Linux AMI.

Database: MySQL 8.0.37 on an AWS RDS instance.

Web Server: Express 4.19.2.

Server-Side Language: TypeScript.

Additional Tech:

Frontend Framework: React.

Backend Framework: Node.js 22.3.0.

IDE: Visual Studio Code, MySQL Workbench.

SSL Cert: Lets Encrypt (Cert Bot).

Docker: Docker Container on the EC2 instance.

Team Contributions

- **Tharun Krishna**

- Created Title Page and transcribed Team Discussion into Executive Summary.
- Created two Main Use Cases.
- Created Main Data Items and Entities.
- Created initial Functional and Non-Functional Requirements.
- Transcribed Competitive Analysis Summary from Team Discussion.
- Created index and about page, setup repo as per tech stack specifications.
- Created and set up AWS EC2 and RDS instances.
- Installed Tech Stack on Cloud Instance and cloned Repository.
- Writing credentials and cloud connection documentation and instructions.

- **Tyron Cheung: Contributions Score 10**

- Created two Main Use Cases and corresponding Diagrams.
- Created additional Main Use Case Diagrams.
- Researched Google Sheets as a Competitor.
- Filled out overview table in Competitive Summary.
- Filled out features table in Competitive Summary.
- Created about page.
- Appended index page.

- **Xuefeng Guan (Tony): Contributions Score 8**

- Created two Main Use Cases and corresponding Diagrams.
- Researched Discord as a Competitor.
- Filled out overview table in Competitive Summary.
- Filled out features table in Competitive Summary.
- Created about page.
- Appended index page.

- **Naw San: Contributions Score 9**

- Created two Main Use Cases and corresponding Diagrams.
- Added additional Functional Requirements.
- Added additional Non-Functional Requirements.
- Researched Airtable as a Competitor.
- Filled out overview table in Competitive Summary.
- Filled out features table in Competitive Summary.
- Created about page.
- Appended index page.

- **Yasmine Sucu: Contributions Score 8**
 - Created two Main Use Cases.
 - Researched Rippling HCM as a Competitor.
 - Filled out overview table in Competitive Summary.
 - Filled out features table in Competitive Summary.
 - Created about page.
 - Appended index page.
- **Jannelly Hernandez Umanzor: Contributions Score 7**
 - Created two Main Use Cases.
 - Researched PERSCOM as a Competitor.
 - Filled out overview table in Competitive Summary.
 - Filled out features table in Competitive Summary.
 - Created about page.
 - Appended index page.

SW Engineering CSC648-848-05 Summer 2024

TeamMate Personnel Management Team 05 - Concat

Name	E-Mail	Role
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 23 2024**

Version History

Revision	Date Submitted
M1V1	20 June 2024
M1V2	09 July 2024
M2V1	09 July 2024
M2V2	23 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 (Anyone with a registered account)
 - 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 (Organizations created by a registered user)
 - Name
 - Plaintext title.
 - Must be unique between all organizations.
 - Owner
 - Connected to a registered user.
 - Transferable to another registered user.
 - Description
- Members (Association between a user and an organization)
 - 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 (Member created category for dividing people)
 - 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 (Member created event that other members can “join”)
 - 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 (Associates a Member to an Event to track attendance)
 - Event
 - Connected to an event.
 - Member
 - Connected to a member.
 - Attendance
 - Enumerated type that records if the member was present or absent.

- Qualification Type (Member-defined custom qualifications)
 - Title
 - Description
 - Image
 - PNG or JPG/JPEG.
 - 400x400 as the max dimensions.
 - No greater than 2 megabytes.
- Qualification Instance (An instance of a qualification assigned to a member)
 - 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 (Member-defined custom awards)
 - Title
 - Description
 - Image
 - PNG or JPG/JPEG.
 - 400x400 as the max dimensions.
 - No greater than 2 megabytes.
- Award Instance (An instance of an award assigned to a member)
 - 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 (Members can leave comments on other member's pages)
 - 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 (Customizable section usable for whatever the client needs)
 - 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.

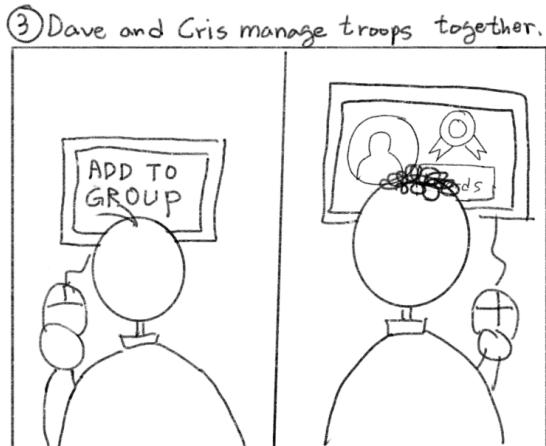
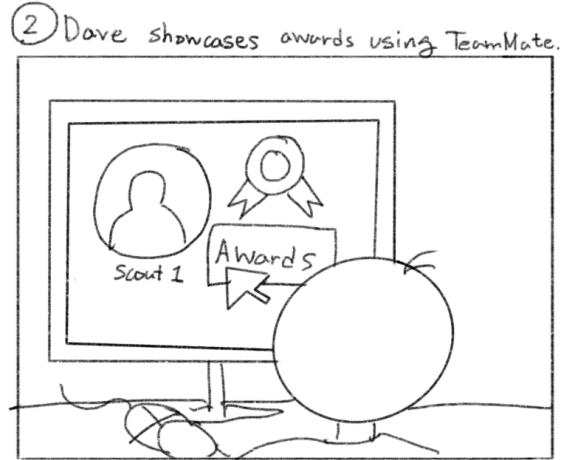
Custom Section

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

UI Mockups and Storyboards

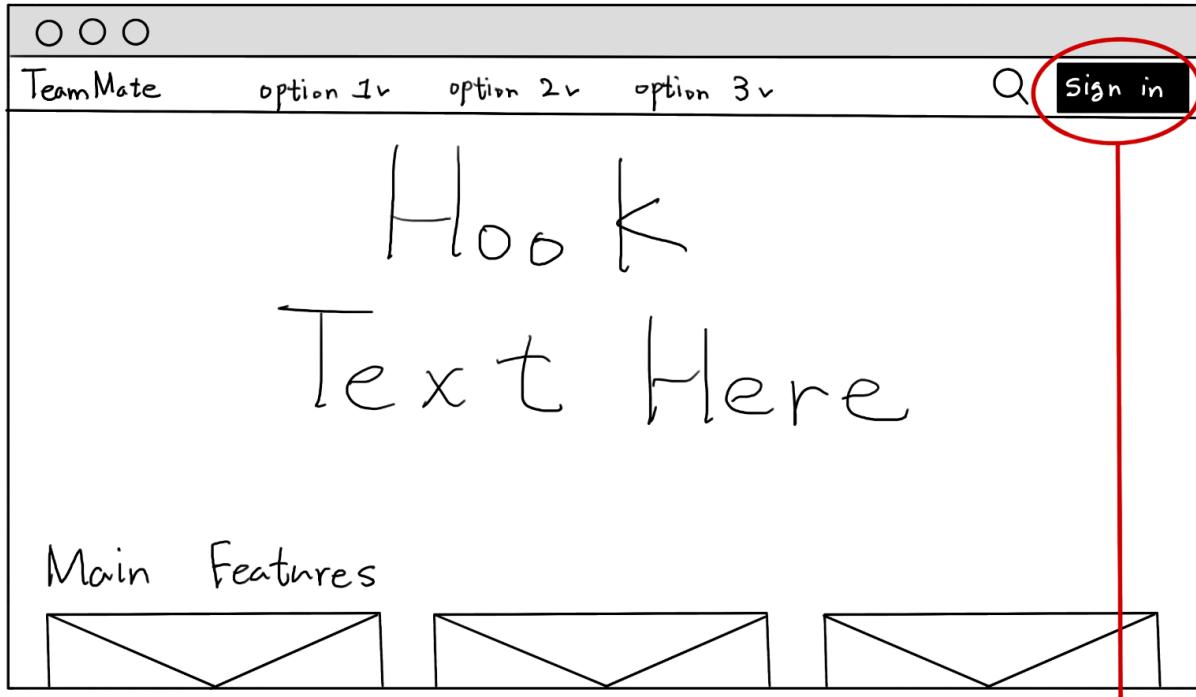
Use Case #1

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.

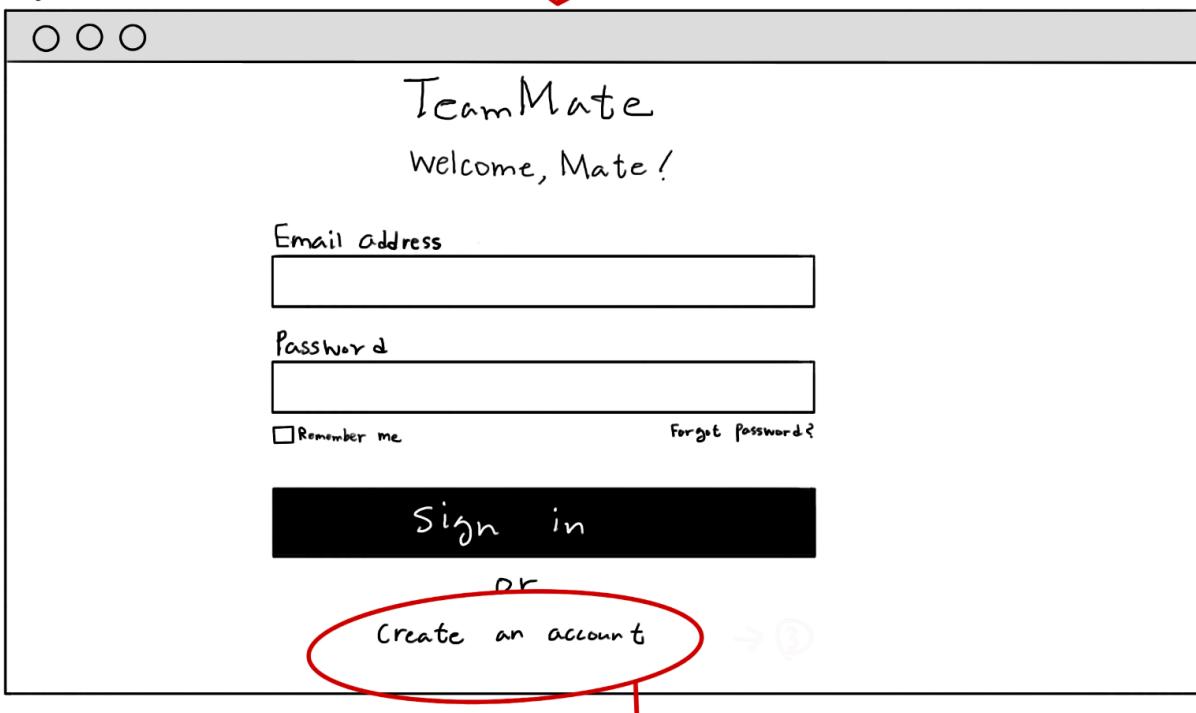


① Start - Leader's perspective
Home Page

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

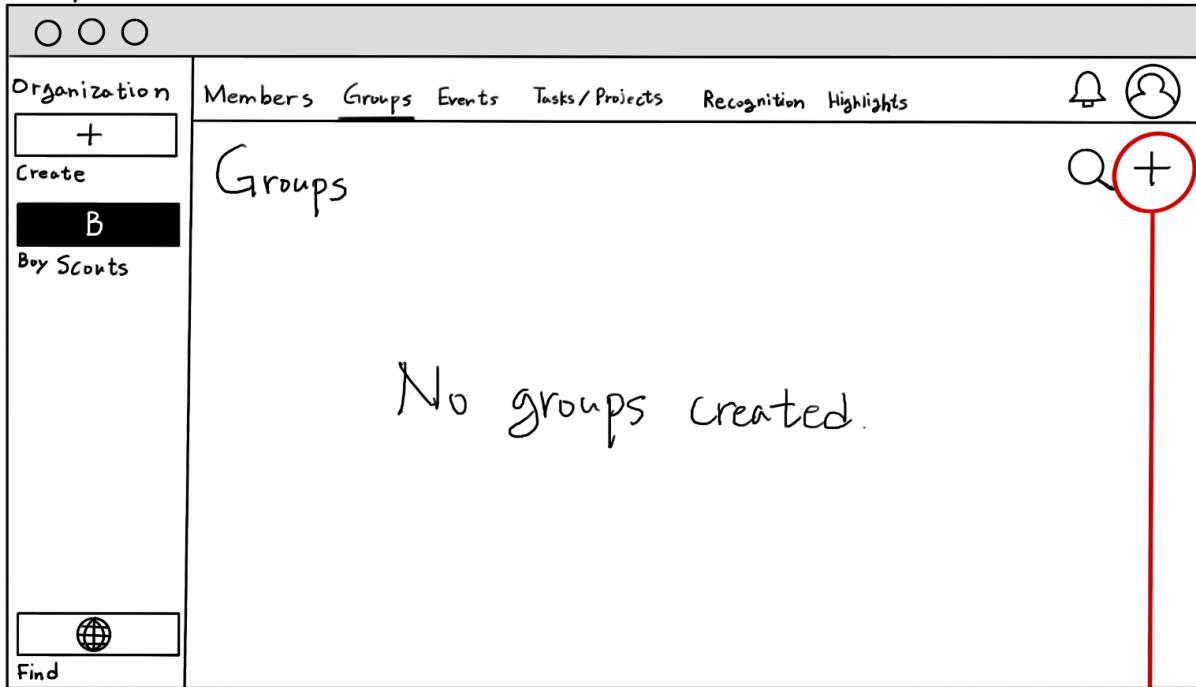
The screenshot shows a dashboard interface. On the left, there's a sidebar with 'Organization' buttons: '+ Create' and 'Find'. The main area has a 'Dash' title and a 'Create organization' form. The form fields are 'Organization name' (Boy Scouts) and 'Type of organization' (Other). A large red circle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

⑥

Organization Dashboard Page

The screenshot shows the 'Boy Scouts' organization dashboard. The sidebar includes '+ Create' and 'Find' buttons. 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



Wireframe of the Groups Page:

- Header: Three circular icons.
- Left sidebar:
 - Organization section with a '+' button and 'Create' link.
 - A dark box labeled 'B' containing 'Boy Scouts'.
 - 'Find' button with a globe icon.
- Main content area:
 - Header navigation: Members, Groups (underlined), Events, Tasks / Projects, Recognition, Highlights.
 - Search bar with magnifying glass icon.
 - Large text: 'Groups'.
 - Text: 'No groups created.'

⑧ Group Create Page



Wireframe of the Group Create Page:

- Header: Three circular icons.
- Left sidebar:
 - Organization section with a '+' button and 'Create' link.
 - A dark box labeled 'B' containing 'Boy Scouts'.
 - 'Find' button with a globe icon.
- Main content area:
 - Header navigation: Members, Groups (underlined), Events, Tasks / Projects, Recognition, Highlights.
 - Search bar with magnifying glass icon.
 - Form:
 - Title: 'Create a group' with a close 'X' button.
 - Group name input field: 'Team 1'.
 - Description input field: 'abc'.
 - Access dropdown: 'Invite only'.
 - Member limit input field: 'N/A'.
 - Large 'Create' button at the bottom.

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

A red arrow points from the 'Edit' button in the Member's Profile Page below to the 'Save' button on this page.

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

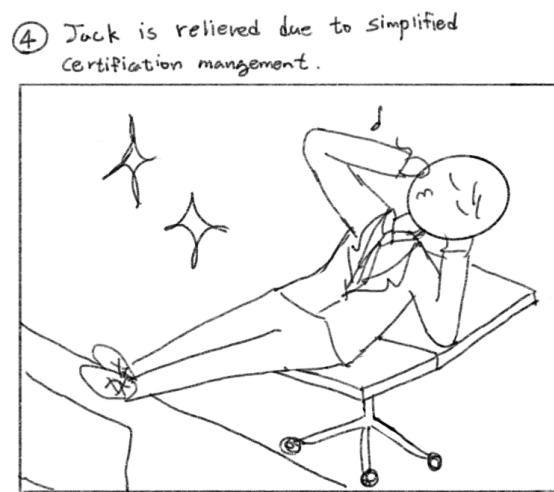
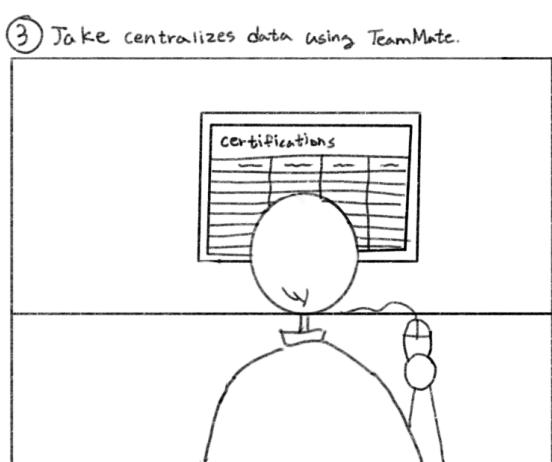
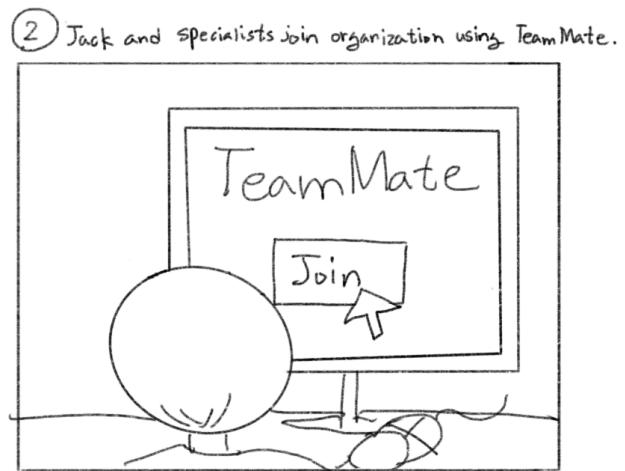
Recognition

No recognition.

Edit

Use Case #2

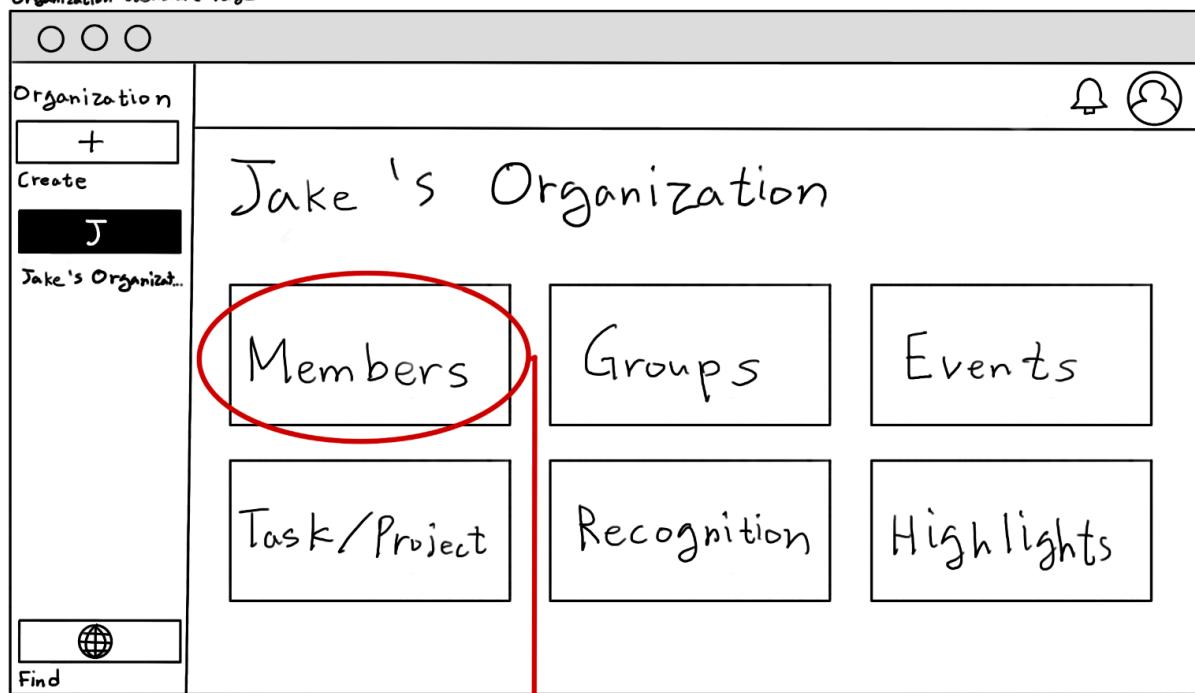
Jake, a leader of a professional group requiring numerous qualifications, found it tedious to track his subordinates' certifications across various platforms. To improve productivity, he created a TeamMate account and organization, where his specialists joined and had their qualifications imported into 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

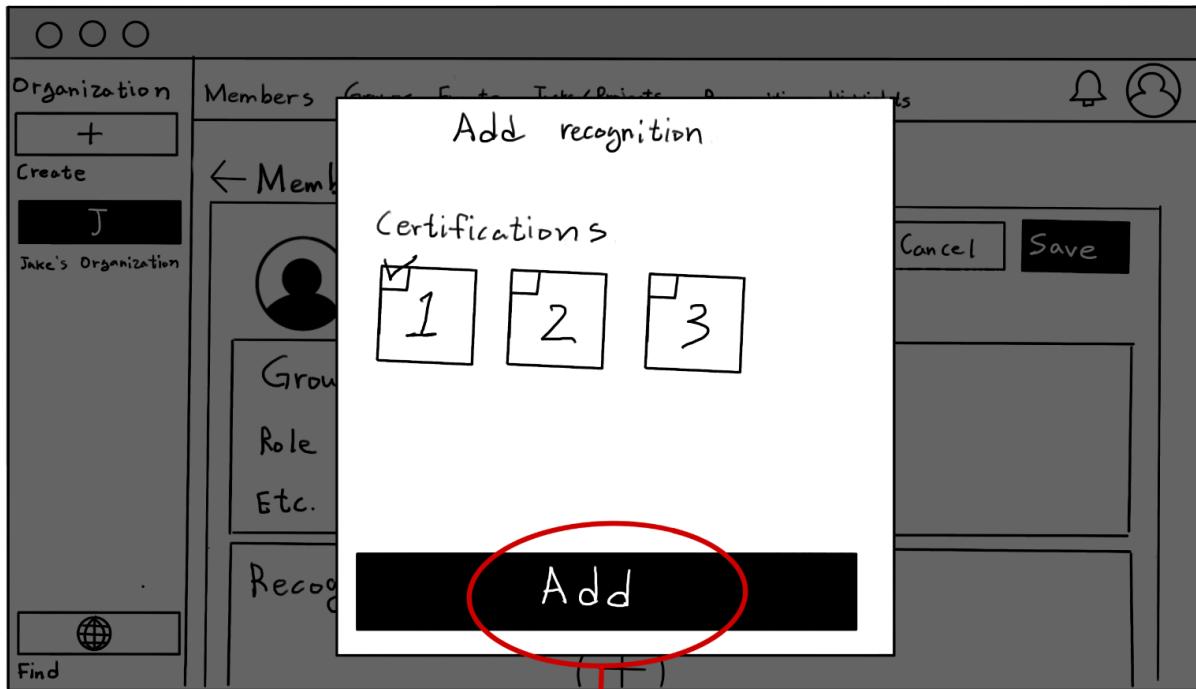
Group : N/A ▾
Role : Member ▾
Etc. : N/A ▾

Recognition

+

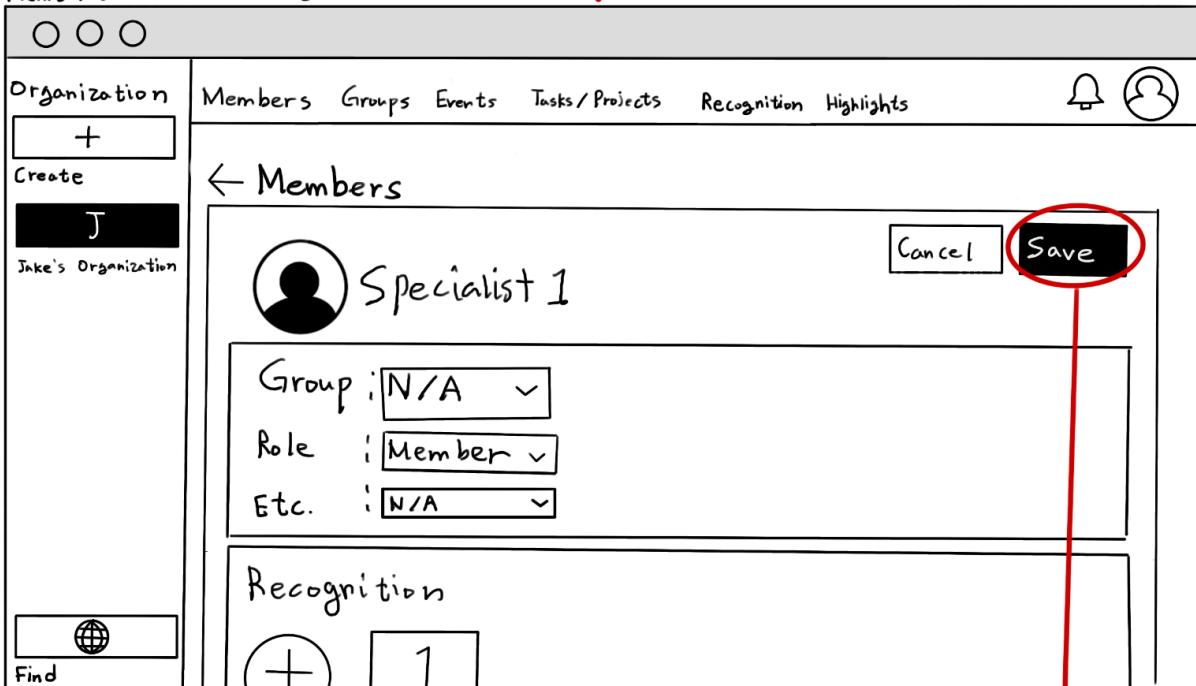
⑤

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.

Organization

+ Create J Jake's Organization

Find

Members Groups Events Tasks/Projects Recognition Highlights

← Recognition

Certification 1

Date : 6/27/2024

Description
abc

Members
Specialist 1

Share

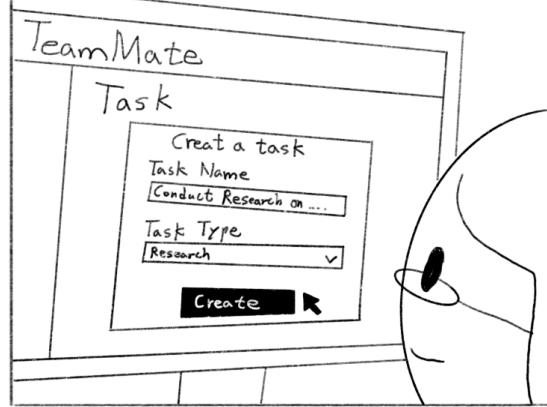
Use Case #3

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.

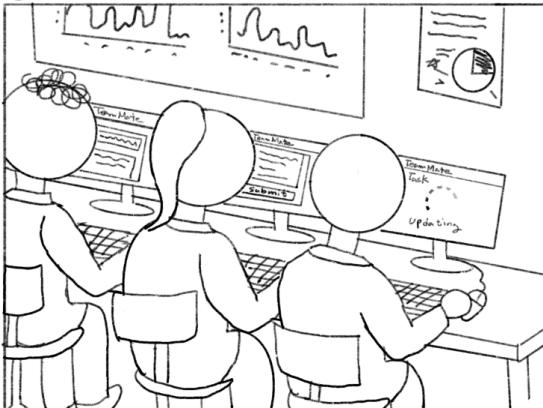
① Dr. Sarah ensures careful planning in the lab.



② She regularly uses TeamMate for project management.



③ Her researchers log progress on TeamMate.



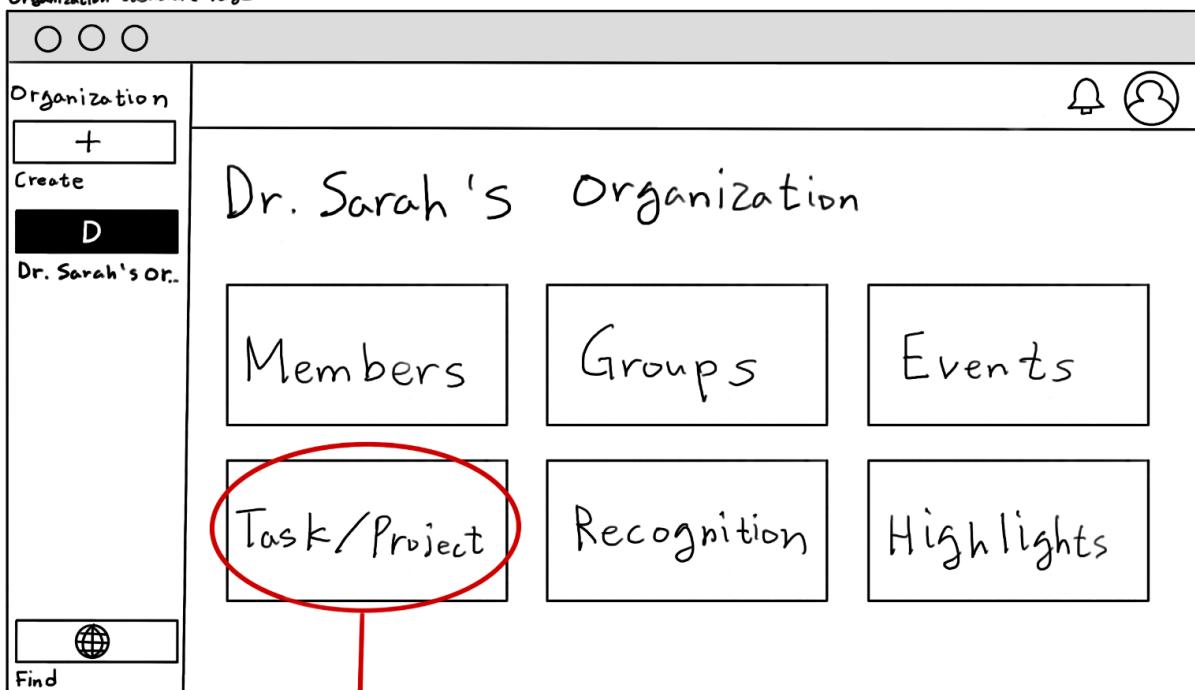
④ Dr. Sarah stays updated on research status.



① Start - Leader Perspective

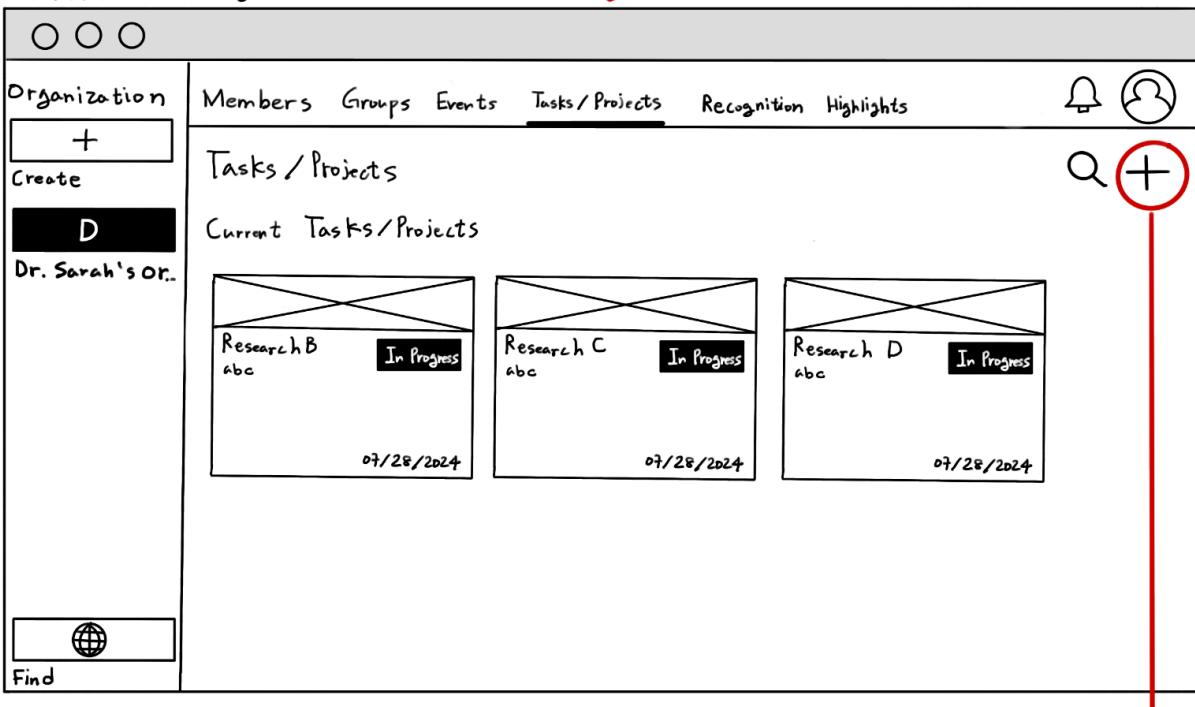
Use case #3

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



(4)

Tasks / Projects Page

○ ○ ○

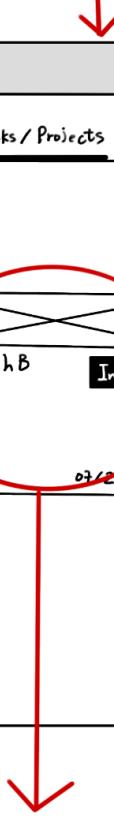
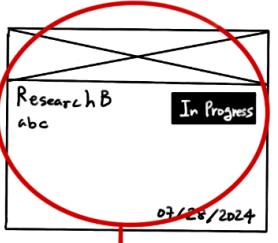
Organization + Create D Dr. Sarah's Or.. Find

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		



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

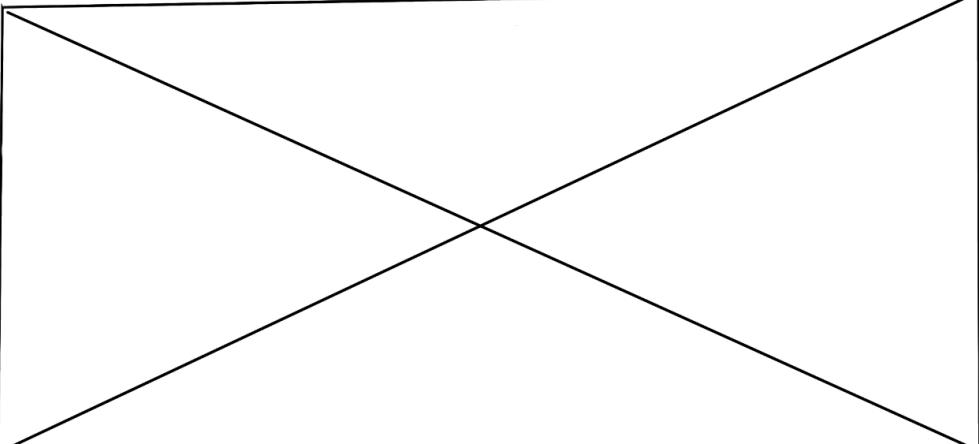
+  

Create ← Research B

D Dr. Sarah's Or..

<  Researcher 1 >

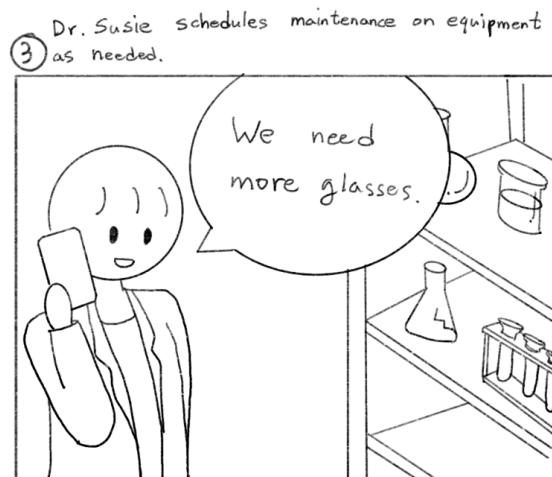
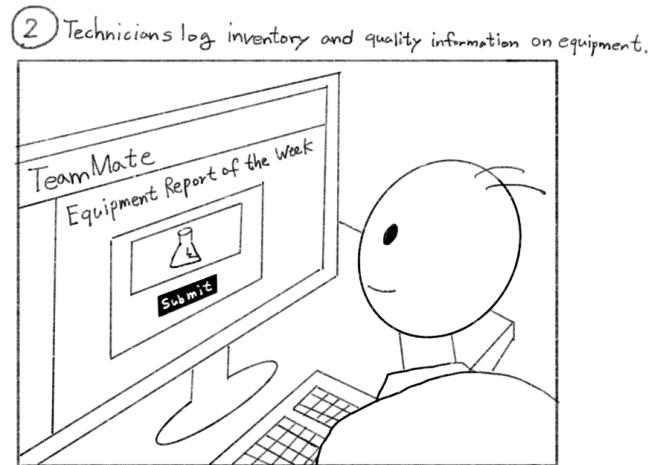
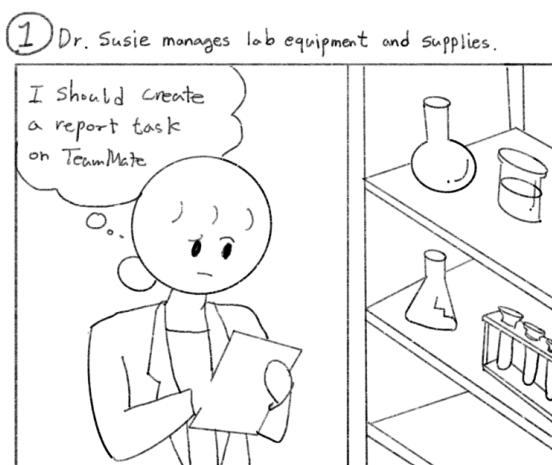
Revise Complete



Find

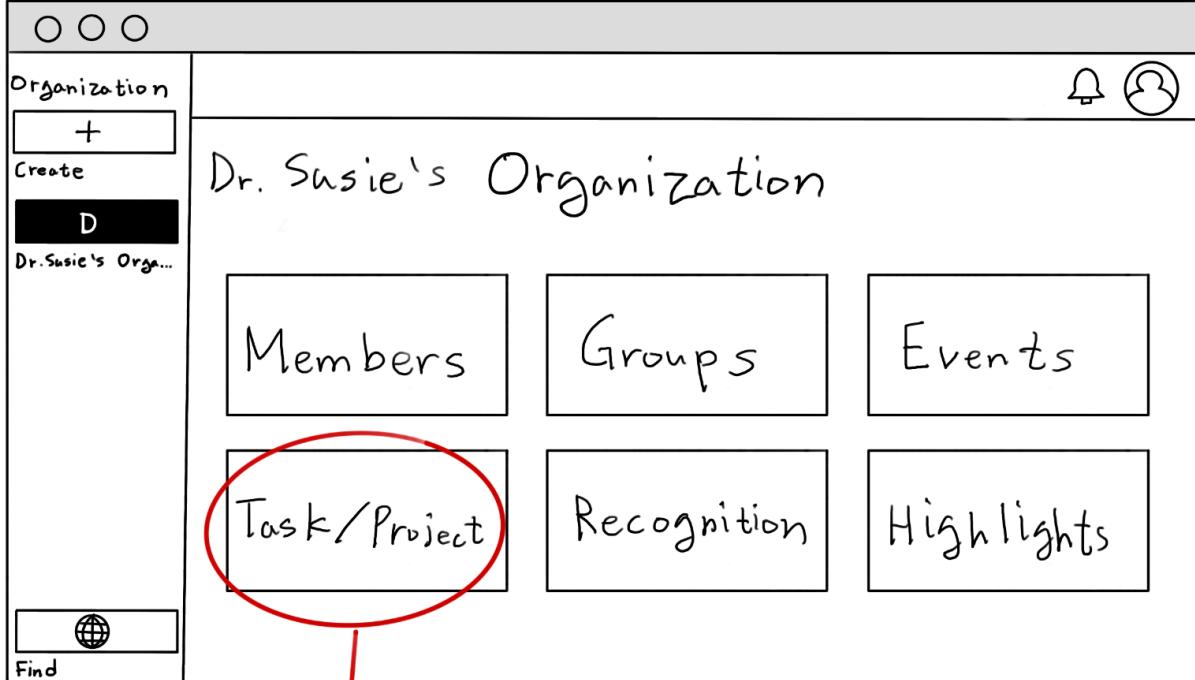
Use Case #4

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.

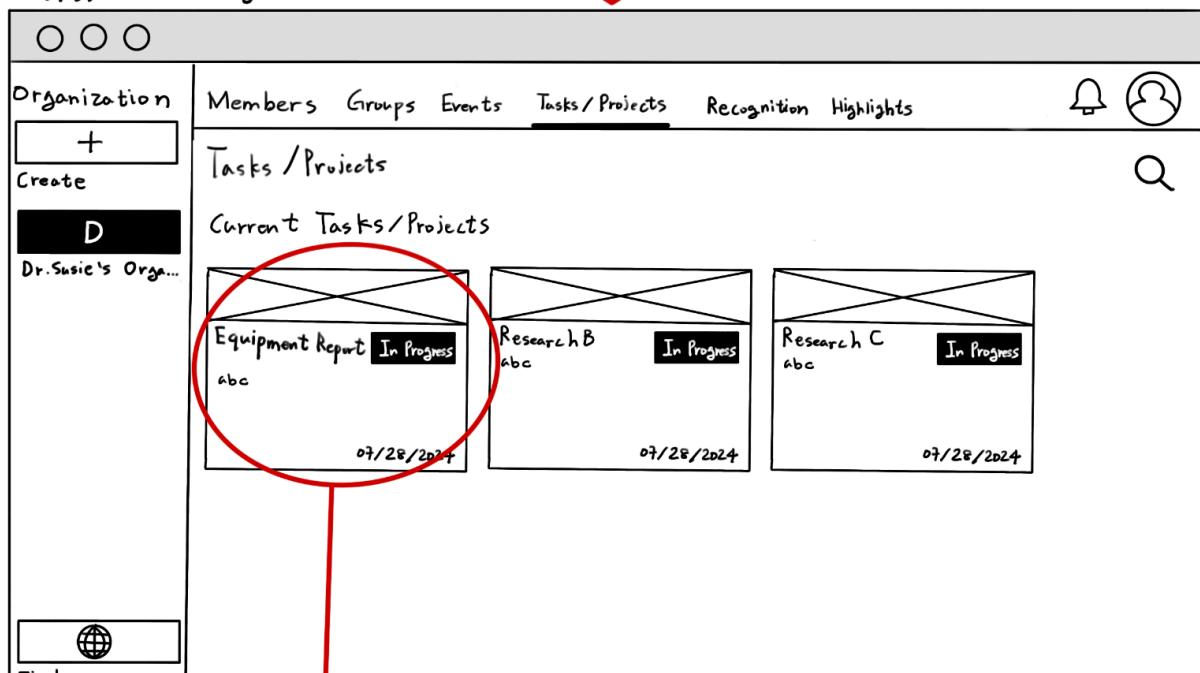


① Start - Member Perspective
Organization Dashboard Page

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 is a large, bold, black button labeled 'Upload Task/Project', which is circled in red.

④

Tasks/Projects Detail Page

This screenshot shows the same 'Tasks/Projects Detail Page' after an action has been taken. A white modal window is centered on the page 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 dark gray, indicating it is no longer active.

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
abc 07/28/2024

Research C In Progress
abc 07/28/2024

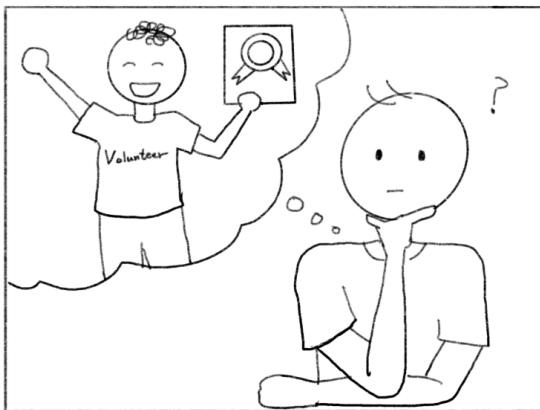
Past Tasks/Projects

Equipment Report Complete
abc

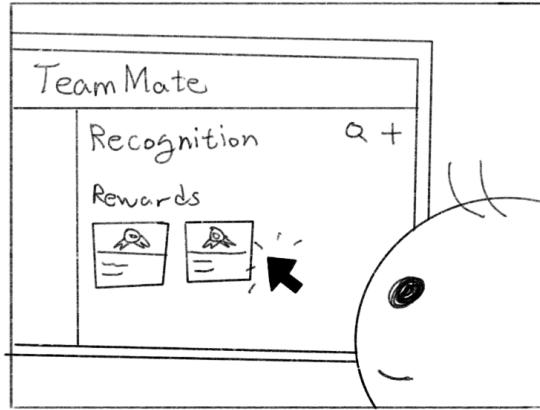
Use Case #5

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.



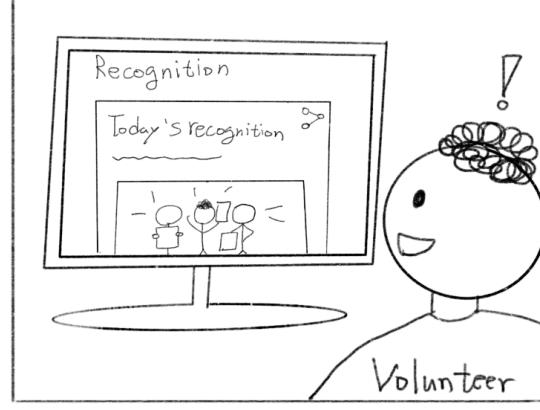
② Steve uses TeamMate for tracking and recognition.



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.

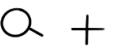
② Highlights Page

This hand-drawn wireframe illustrates the 'Highlights Page'. It features a similar sidebar and navigation bar as the previous page. The main content area displays a post titled 'Best Volunteers of May' with the date 'Published: 05/26/2024'. The post contains a large rectangular placeholder for an image or video, followed by the text 'abc ...'. Below this, there's a section for 'Tagged members' showing three user icons and the text 'VolunteerX and others'. A red circle highlights the '+' icon in the top right corner of the main content area.

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
↑ Upload file

Description: abc

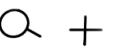
Tag members: Name or email address
Volunteer 1 X

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

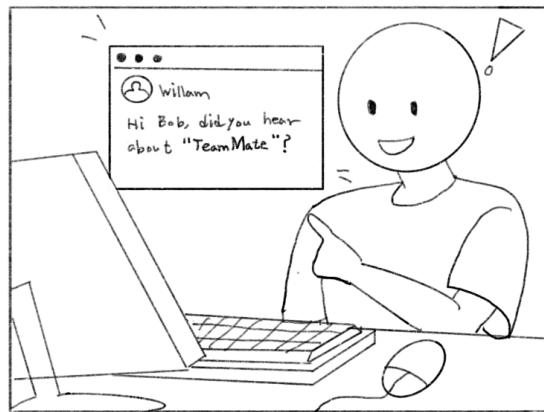
Use Case #6

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.

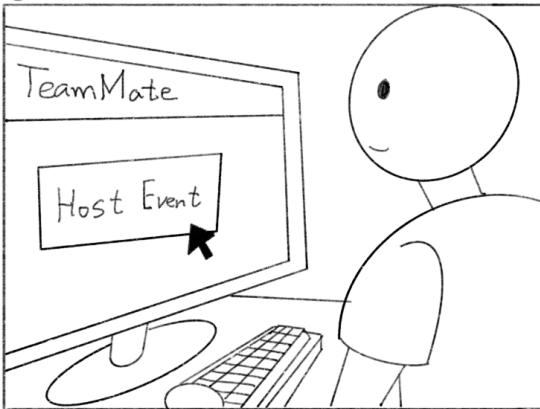
① Bob struggles managing growing gaming community.



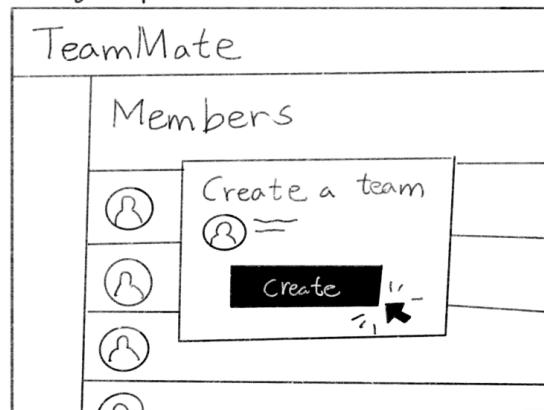
② Bob is introduced to TeamMate for management.



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

Location : Computer Lab Content : Image 1
Upload file

Date & time : 6/28/2024 to 7/28/2024 Detail : abc...

Hosted by : Bob

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...

Find

In 3 days

June 28 5 pm - 7 pm

Computer Lab

Attendees

Player 1	Set up groups
Player 2	

(4)

Events Group Create Page

Organization Members Groups Events Tasks / Projects Recognition Highlights

Gaming Organization

Team 1 vs.

Hosted by

Detail abc...

Find

Create groups

2 ^ groups

Assign automatically

Assign manually

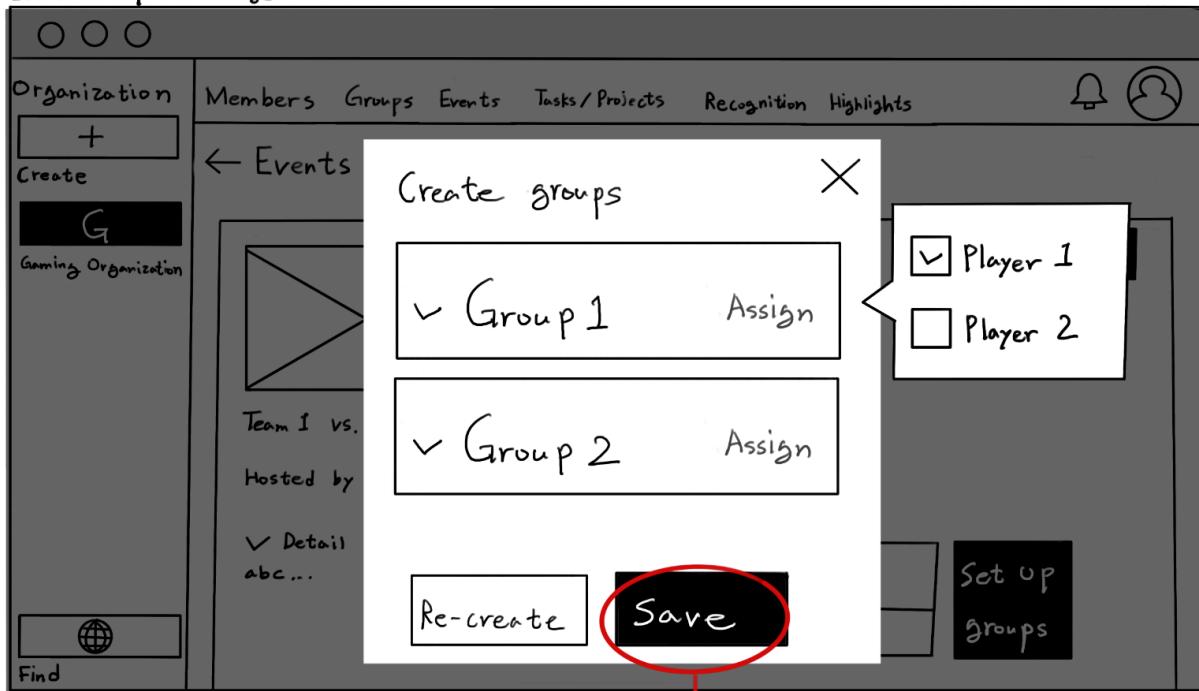
Manually join

Create

Set up groups

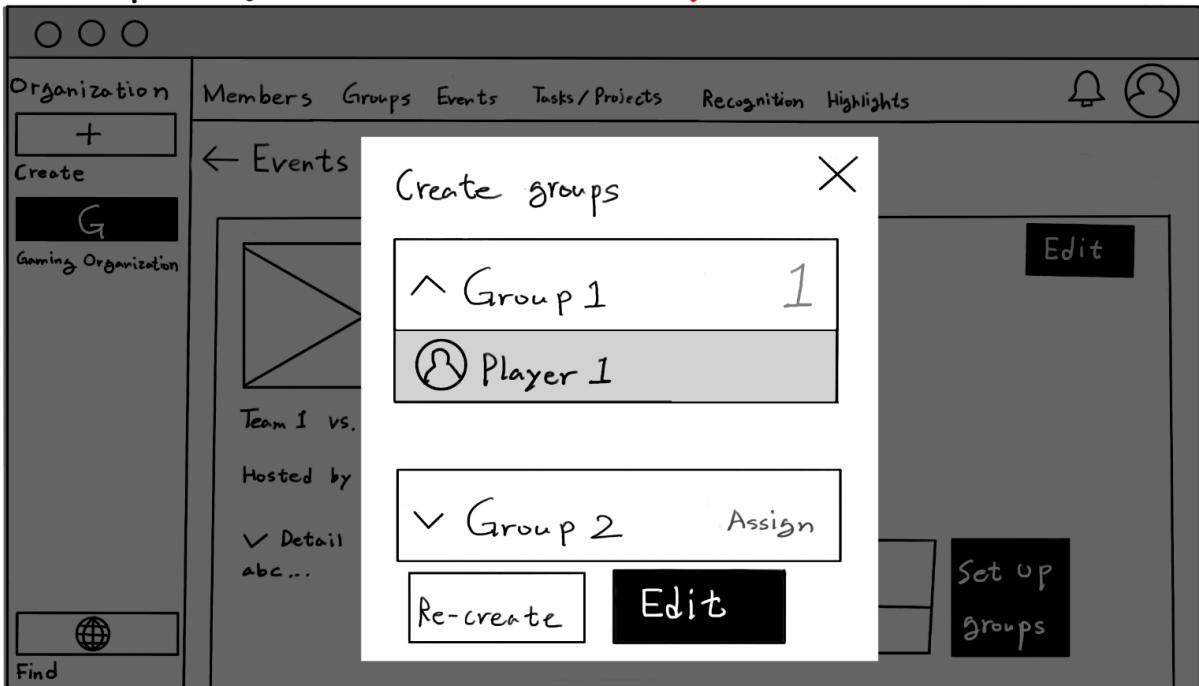
⑤

Events Group Create Page



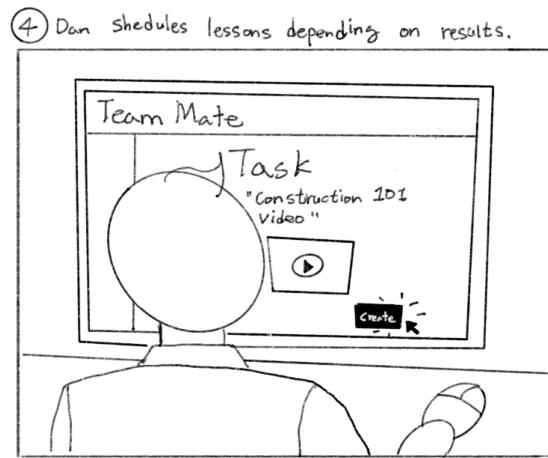
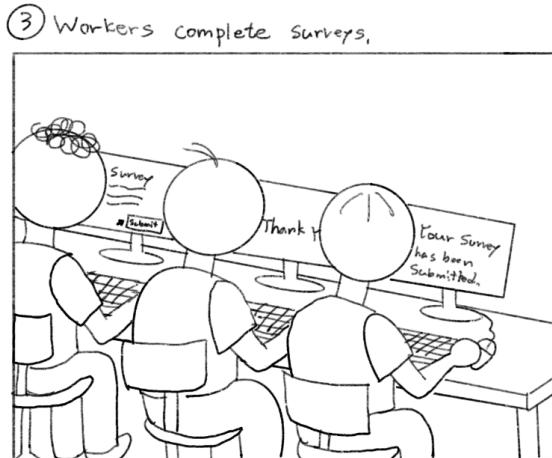
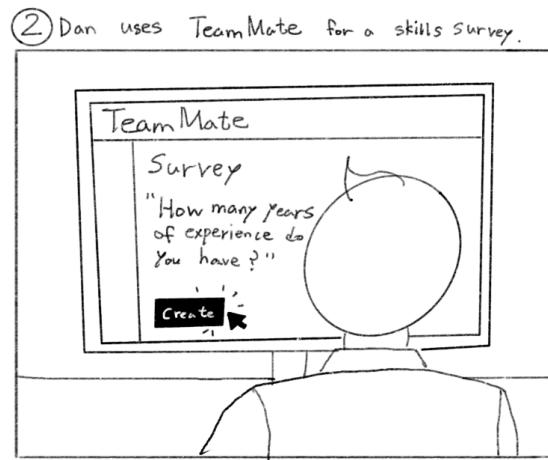
⑥ End

Events Group Create Page



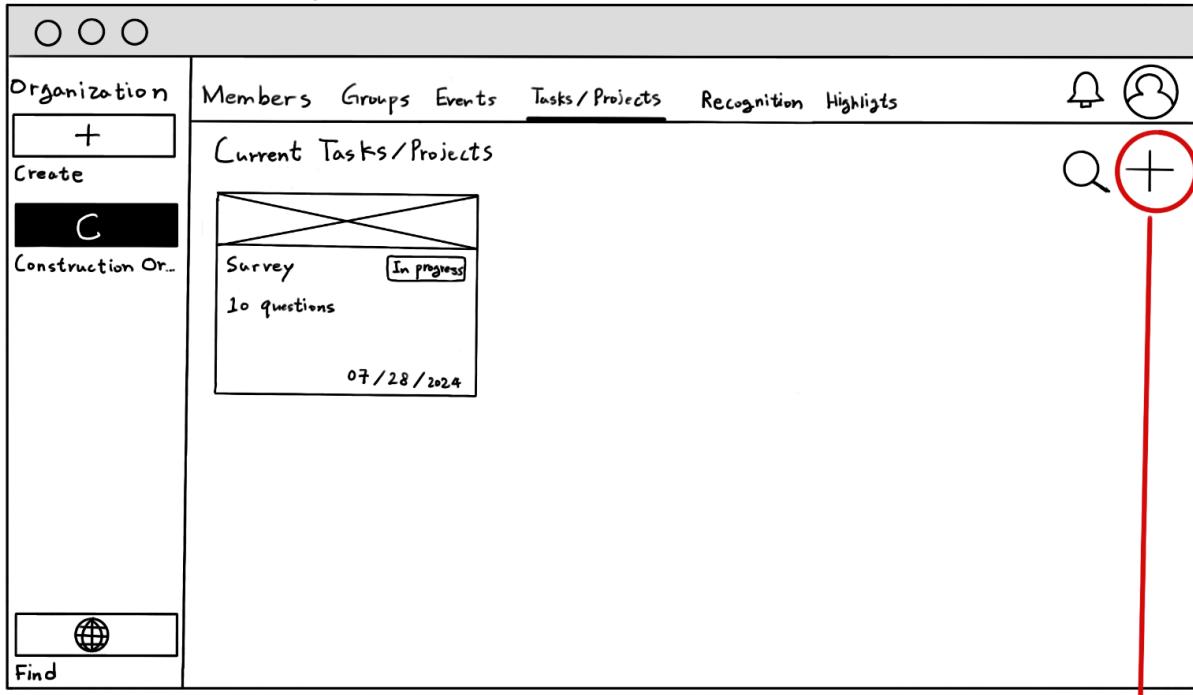
Use Case #7

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

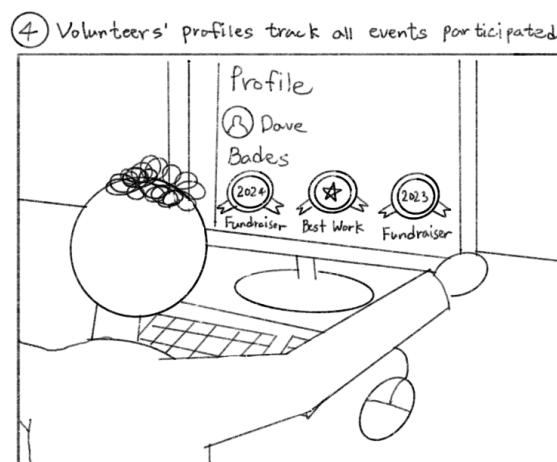
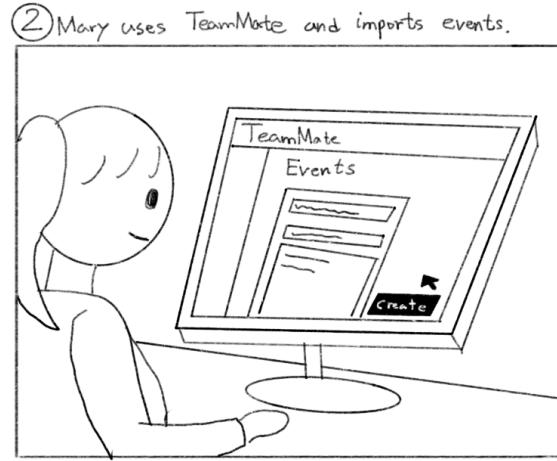
Task / Project Create Page

○ ○ ○

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 :  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
• • • • •

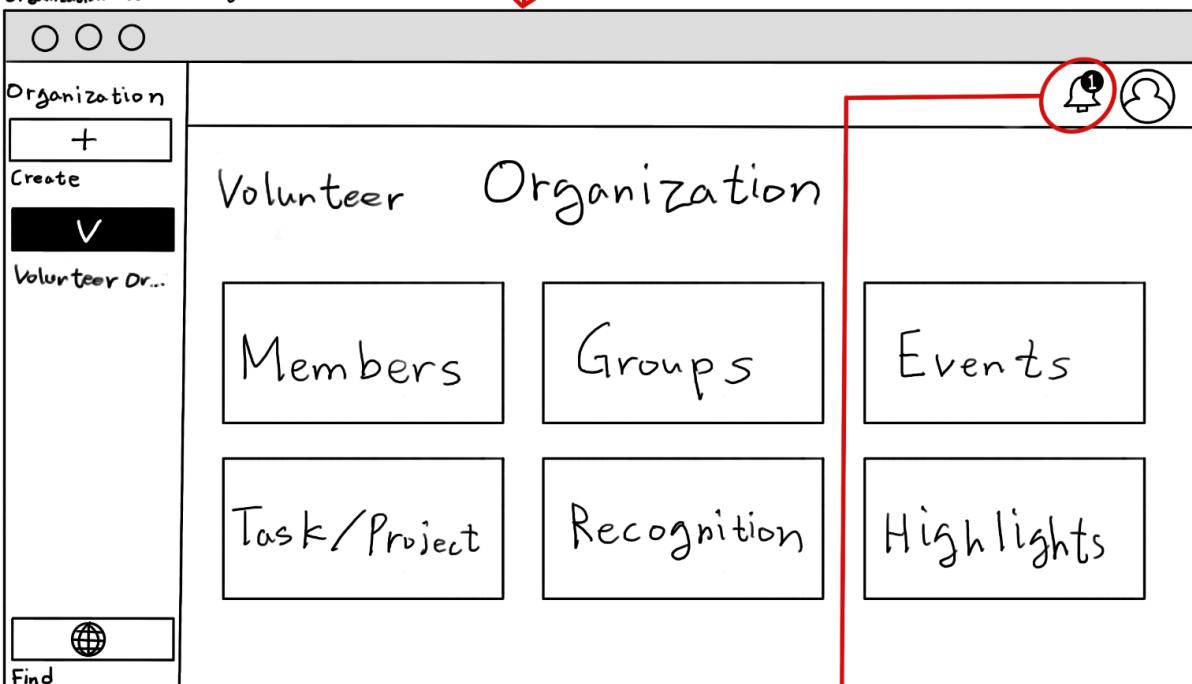
Remember me [Forgot Password?](#)

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

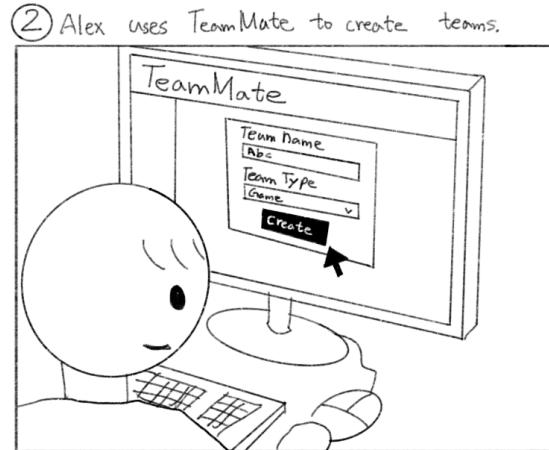
2

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

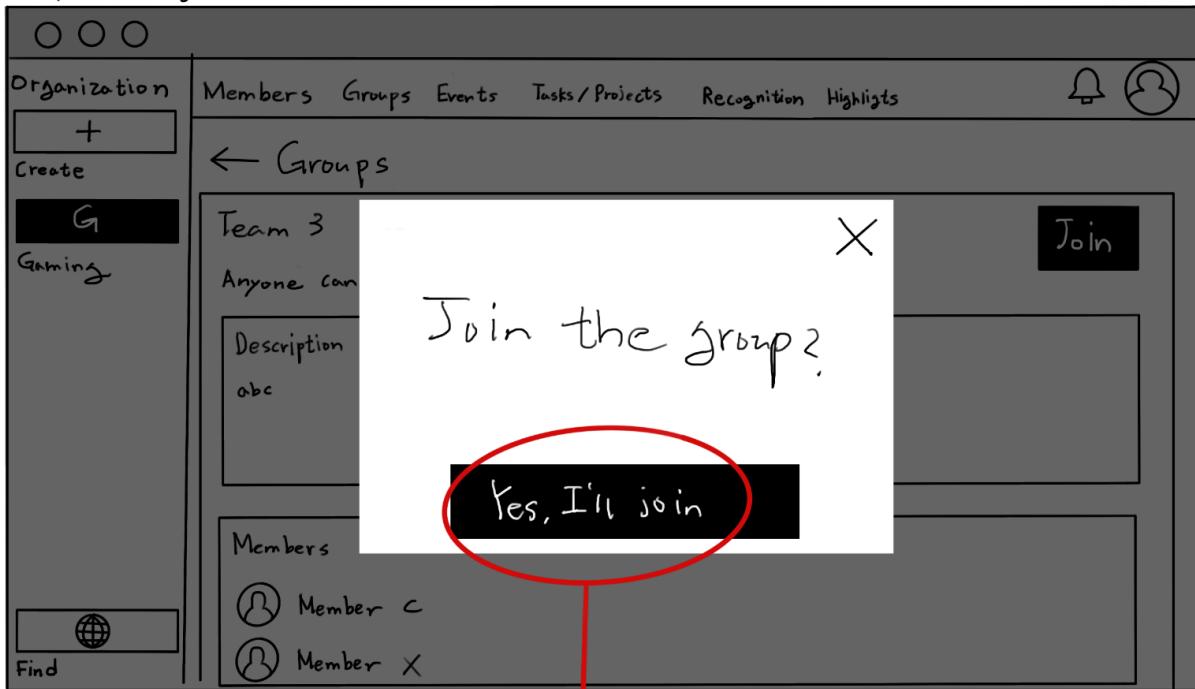
Description
abc

Members

- Member C
- Member X

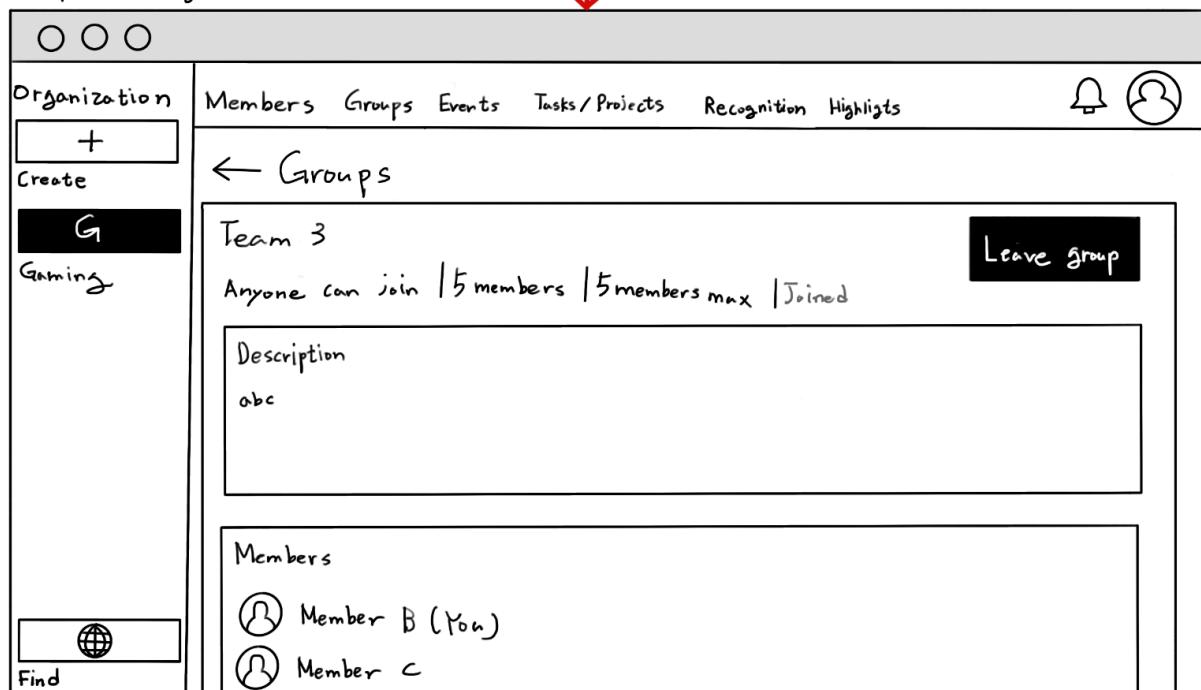


③ Group Detail Page



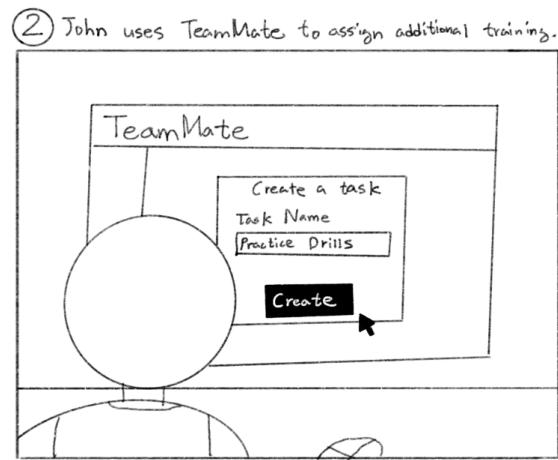
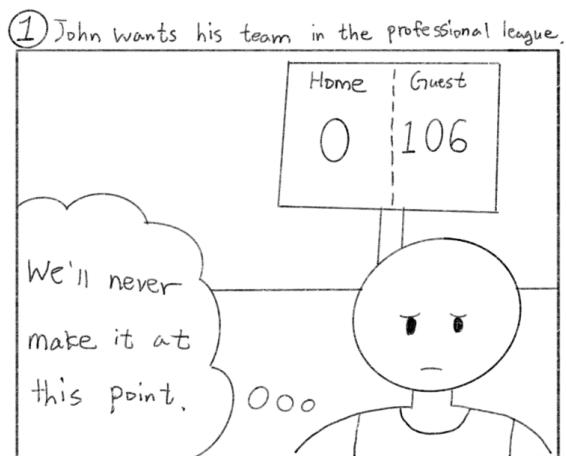
④ End

Group Detail Page



Use Case #10

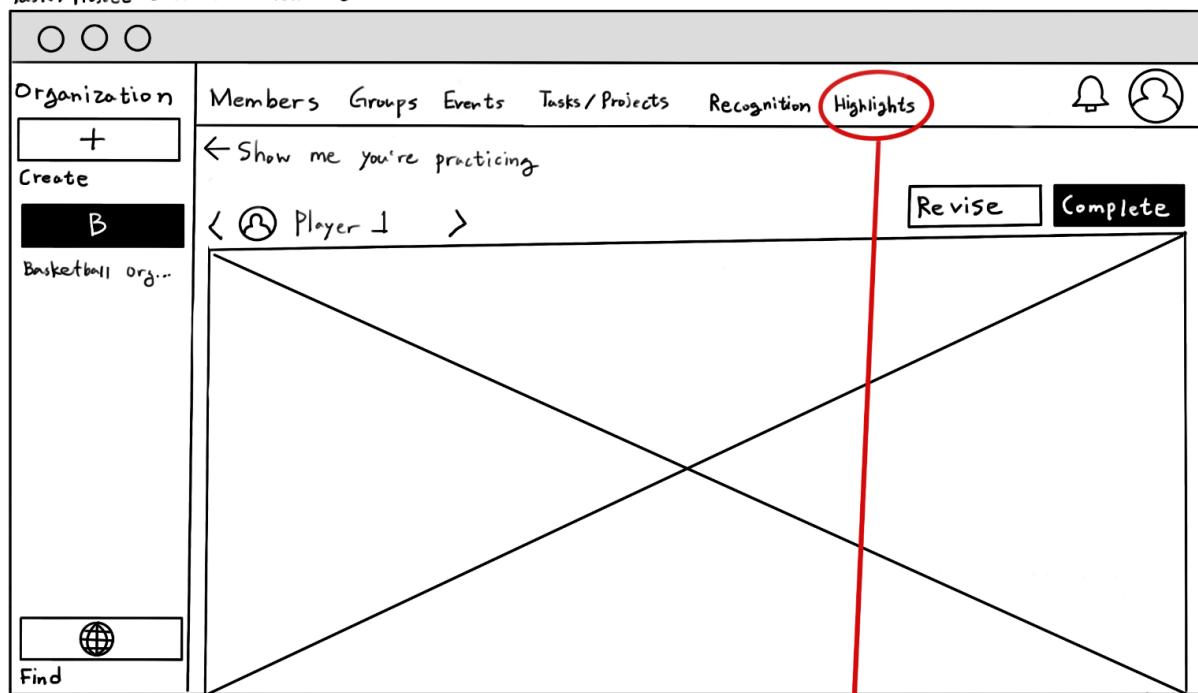
John, a full-time basketball team leader working multiple part-time jobs for funding, struggled to manage team activities, training schedules, physical conditions, and communication. After being introduced to TeamMate, he created an account and set up his organization, enabling teammates to join. The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



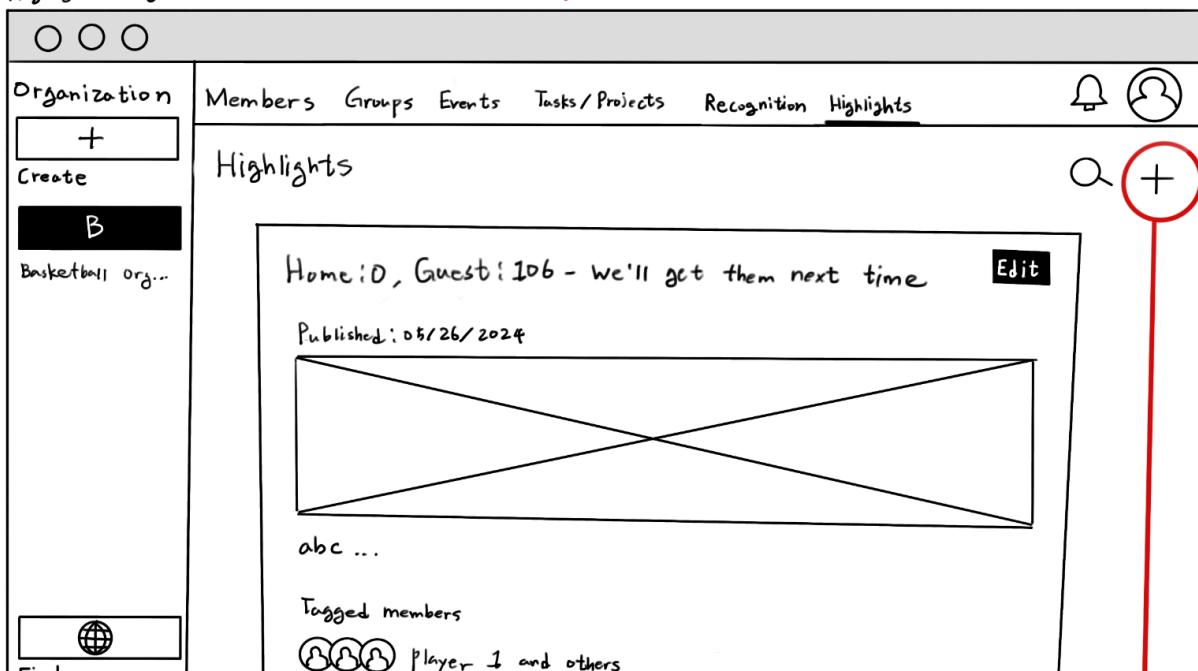
① Start - Leader Perspective

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

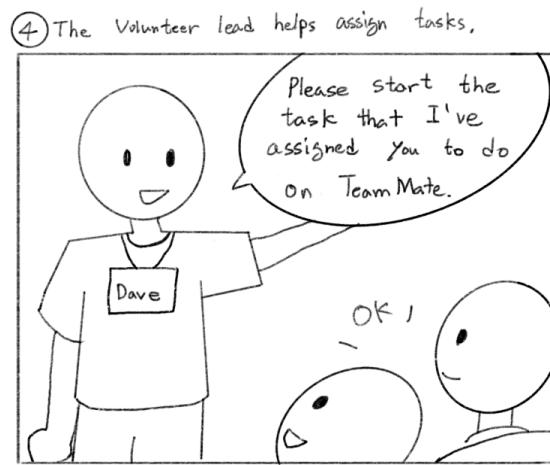
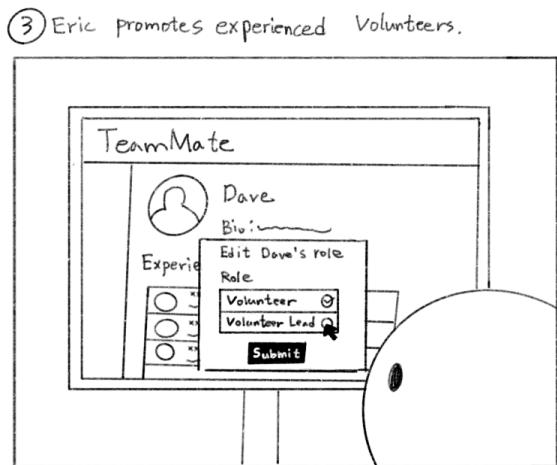
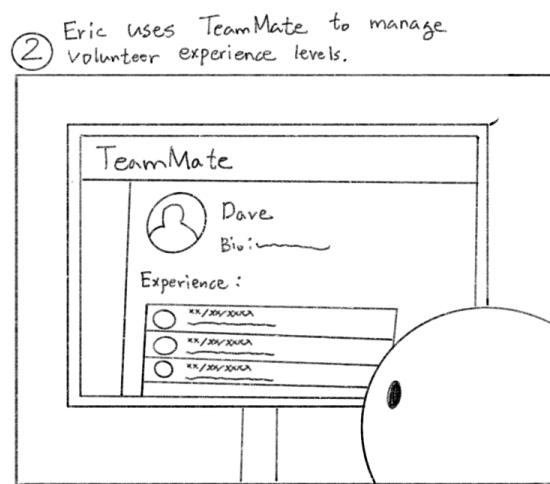
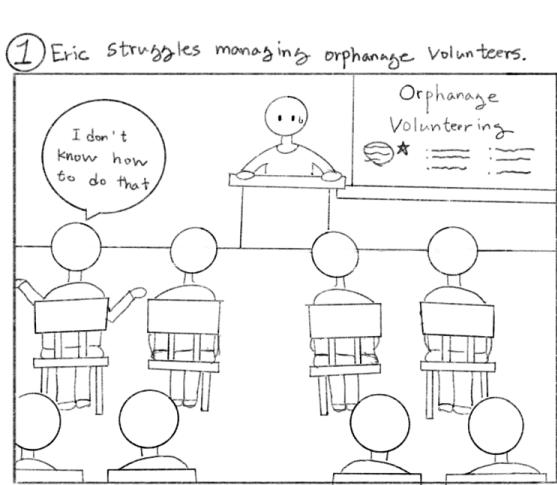
Published: 06/26/2024

abc ...

Tagged members  Player 1 and others

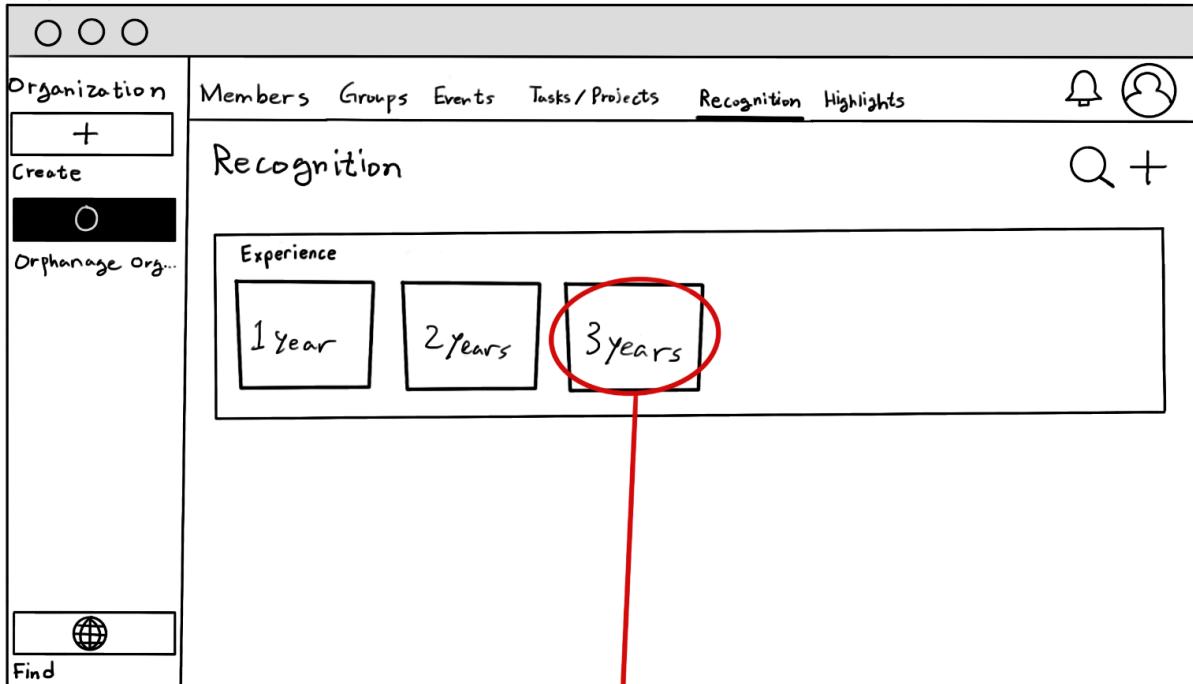
Use Case #11

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.

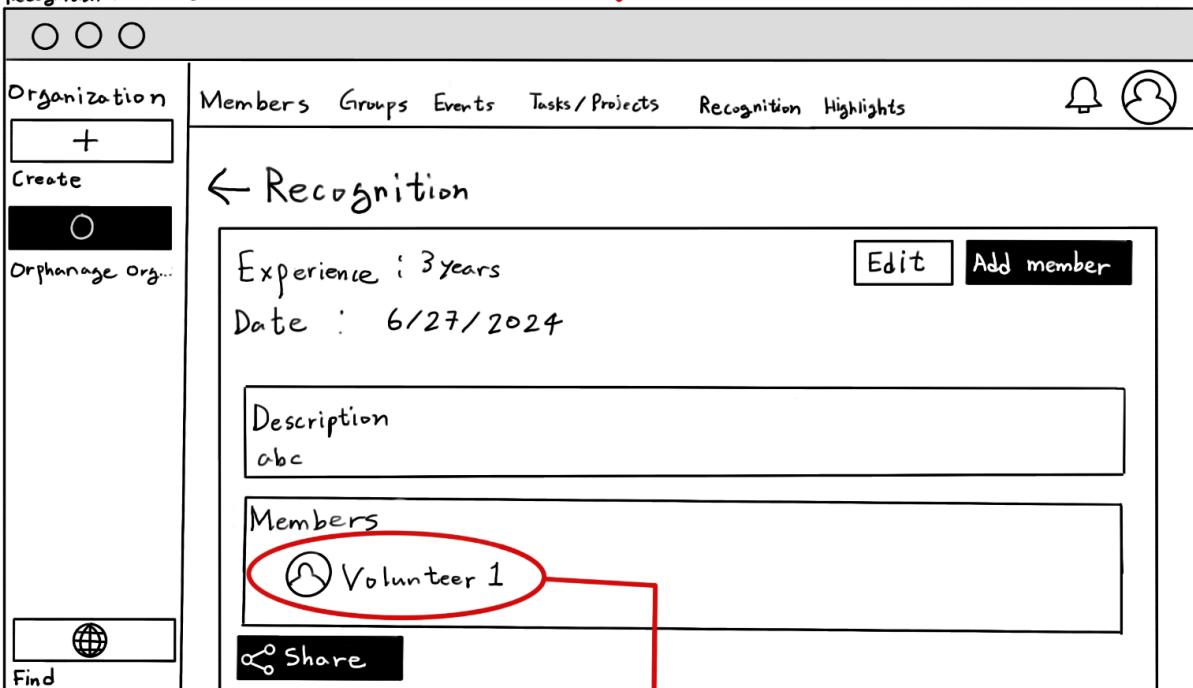


① Start - Leader Perspective
Recognition Dashboard Page

User Case #11



② Recognition Detail Page



③

User Case #11

Member's Profile Page

This wireframe shows a member's profile page. On the left is a sidebar with three circles at the top, followed by sections for 'Organization' (with a '+ Create' button and 'Orphanage org...' text), 'Find' (with a magnifying glass icon), and 'Members' (with a back arrow). The main area has tabs for 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', and 'Highlights'. A bell and user icons are in the top right. Below the tabs, a section labeled '← Members' shows a profile for 'Volunteer 1' with a person icon. To the right of the profile is an 'Edit' button, which is circled in red. Below the profile, there are fields for 'Group : N/A', 'Role : Member', and 'Etc. : N/A'. A 'Recognition' section below shows '3 years'. A red arrow points from the 'Edit' button on the first page down to the 'Edit' button on the second page.

④ End

Member's Profile Edit Page

This wireframe shows a member's profile edit page. It has the same layout as the previous page, with a sidebar and a main 'Members' section. The 'Volunteer 1' profile is shown again, but the 'Edit' button is now highlighted with a red circle. Below the profile, the 'Group' field is set to 'N/A', the 'Role' field is set to 'Member', and the 'Etc.' field is set to 'Admin'. The 'Recognition' section shows '3 years'. In the top right of the main area, there are 'Cancel' and 'Save' buttons, with 'Save' also being circled in red. A red arrow points from the 'Edit' button on the first page down to the 'Save' button on the second page.

Use Case #12

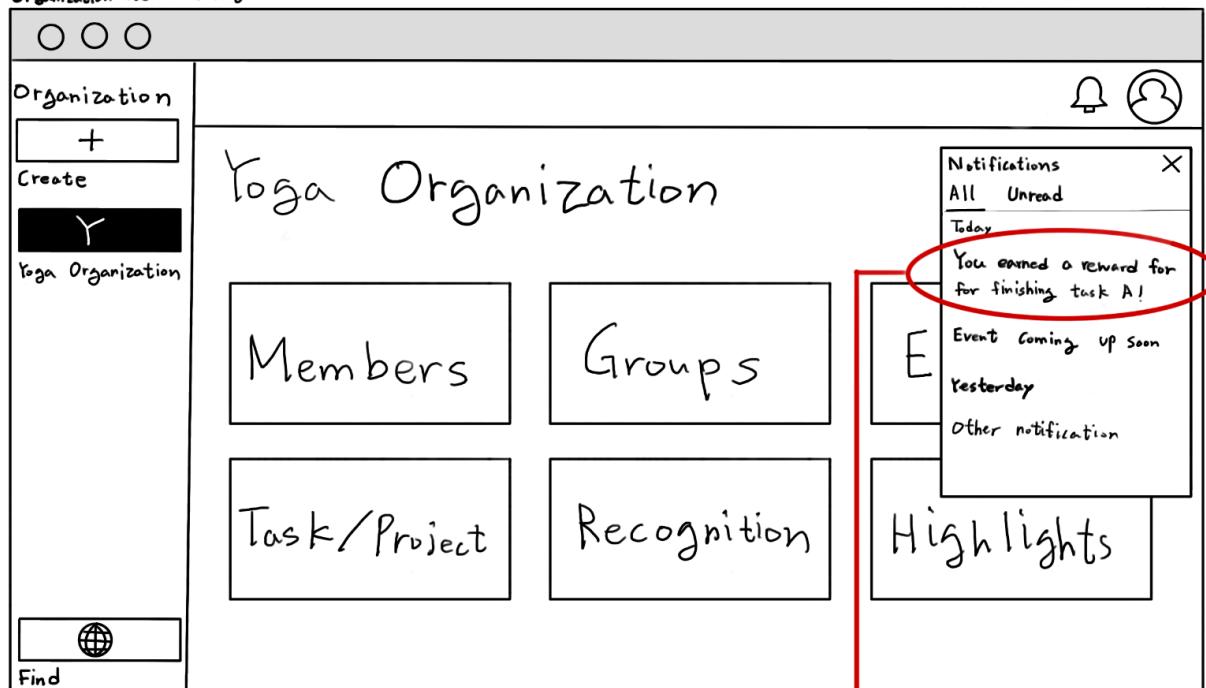
Cianci, director of a Yoga class in Novato, faces challenges with student motivation and goal achievement. To address this, she uses the TeamMate app to set monthly goals for her students and track their progress and methods used. The app allows her to monitor obstacles hindering progress, such as tardiness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



① Start - Member Perspective

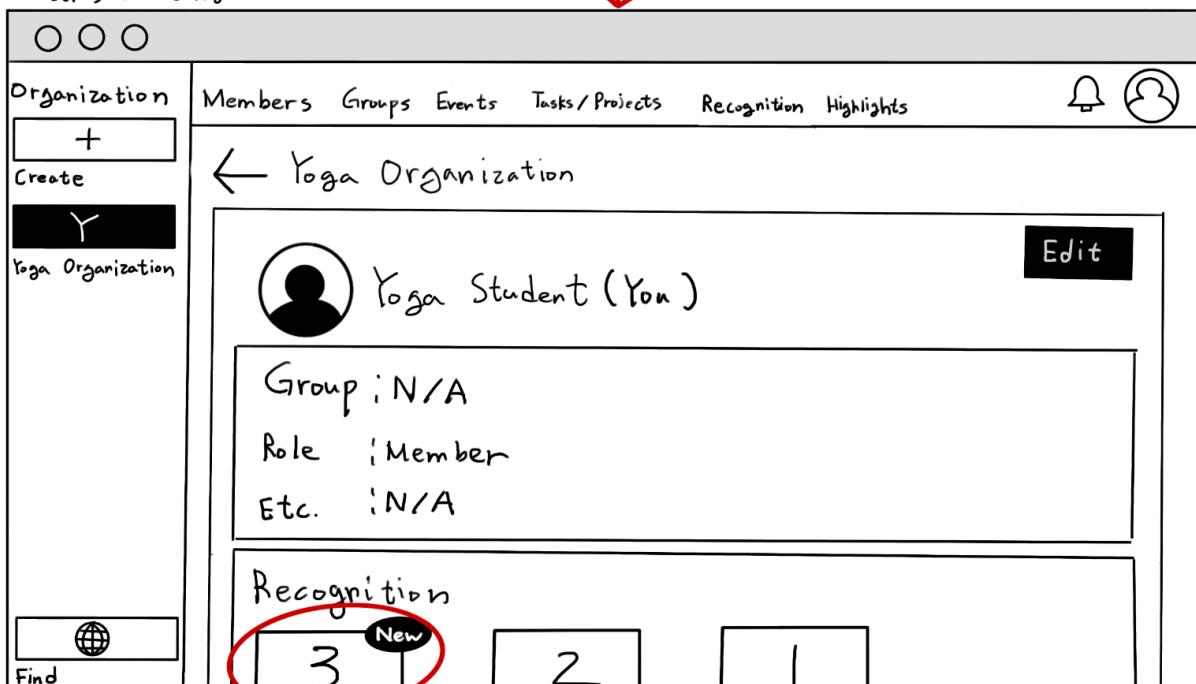
Use Case #12

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 at most 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 many comments.
- 1.10. A registered user can make 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 many qualification instances.
- 2.7. A member can receive many award instances.
- 2.8. A member can give many qualification instances.
- 2.9. A member can give many award instances.
- 2.10. A member shall have an organization.

- 2.11. A member shall have 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 many groups.
- 3.6. An organization shall have many events.
- 3.7. An organization shall have one or many members.
- 3.8. An organization shall have many custom sections.
- 3.9. An organization shall have many award types
- 3.10. An organization shall have many qualification types

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.
- 5.6. An award type will have an associated org

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.
- 7.6. A qualification type will have an associated org

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 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.

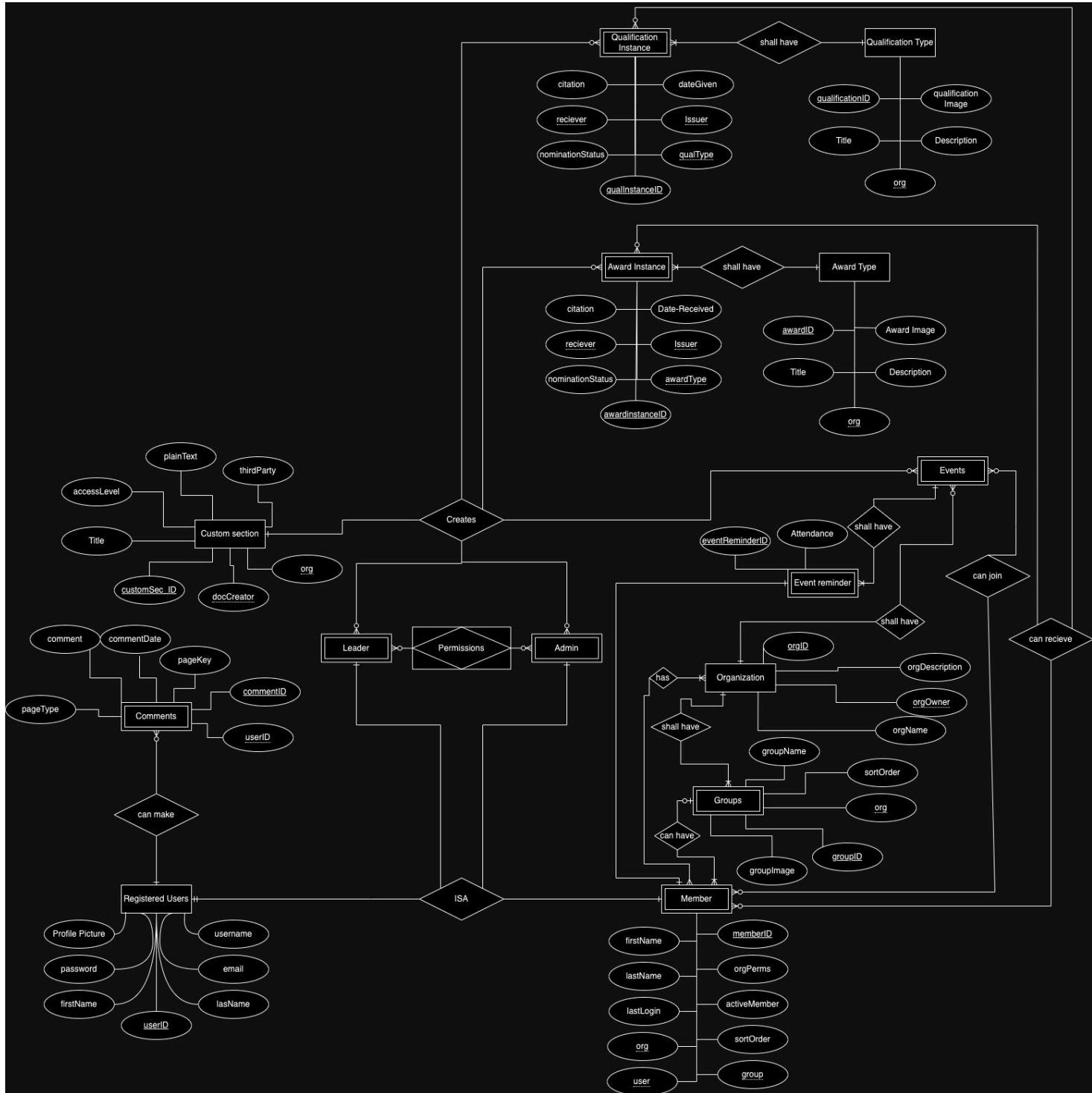
11. Event reminder

- 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).

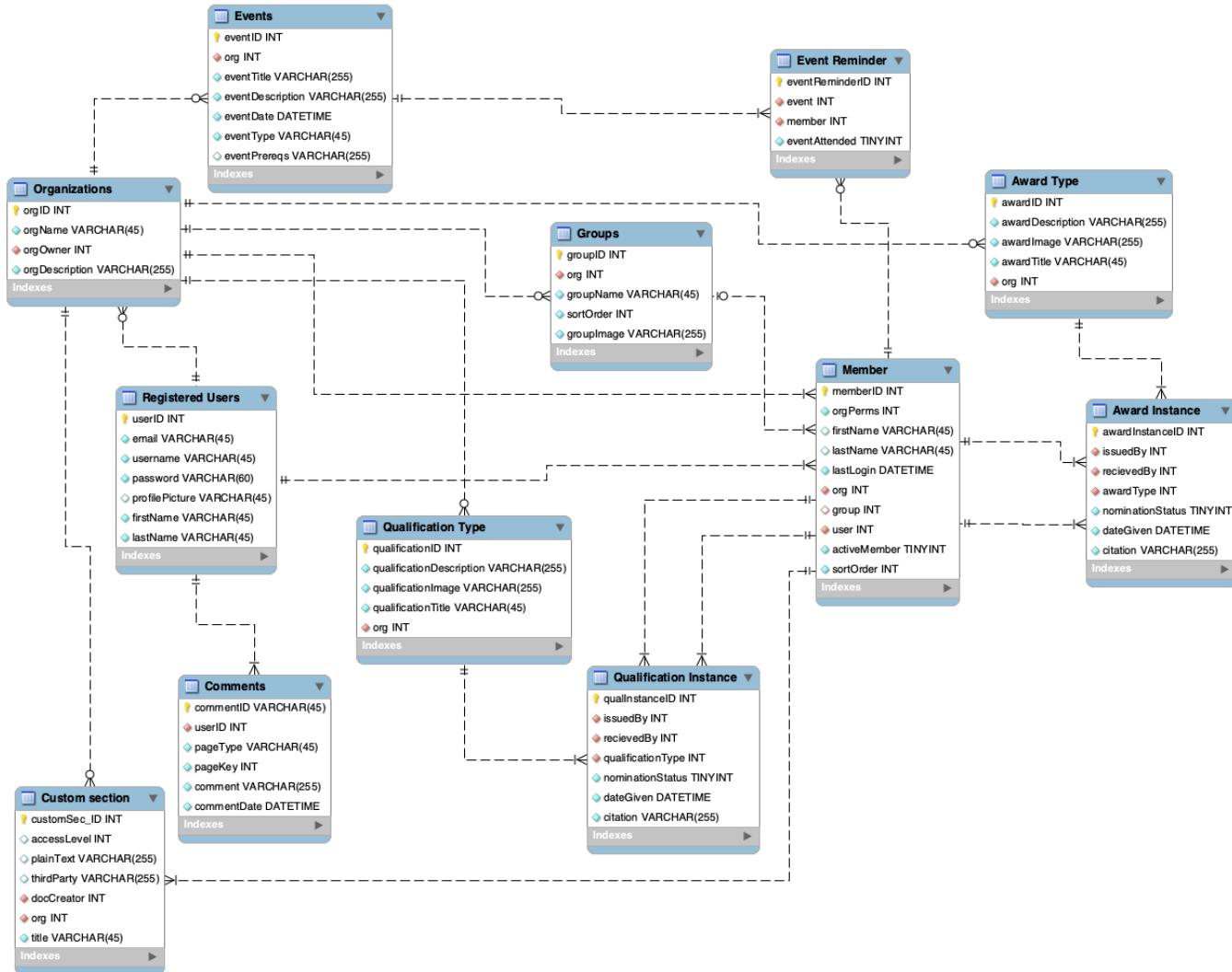
12. Groups

- 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

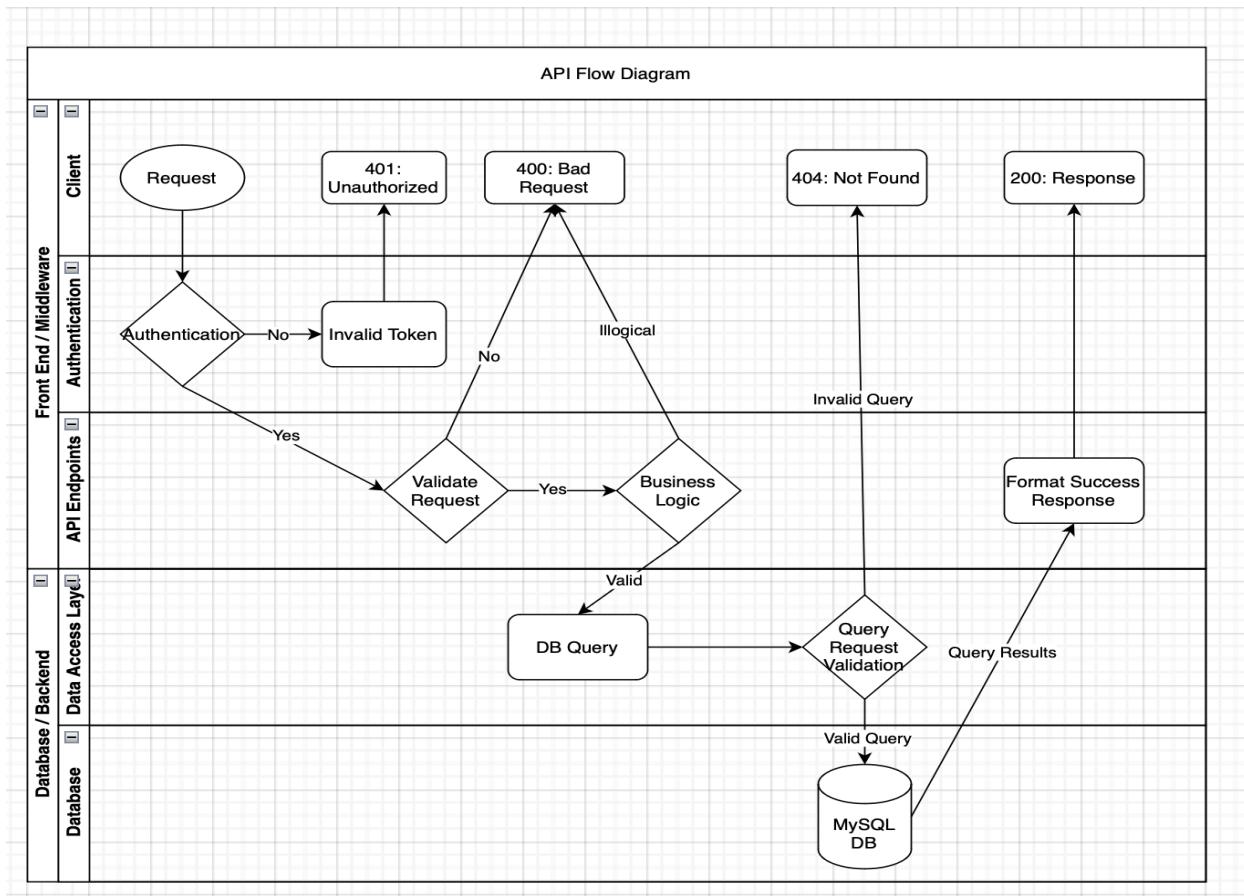
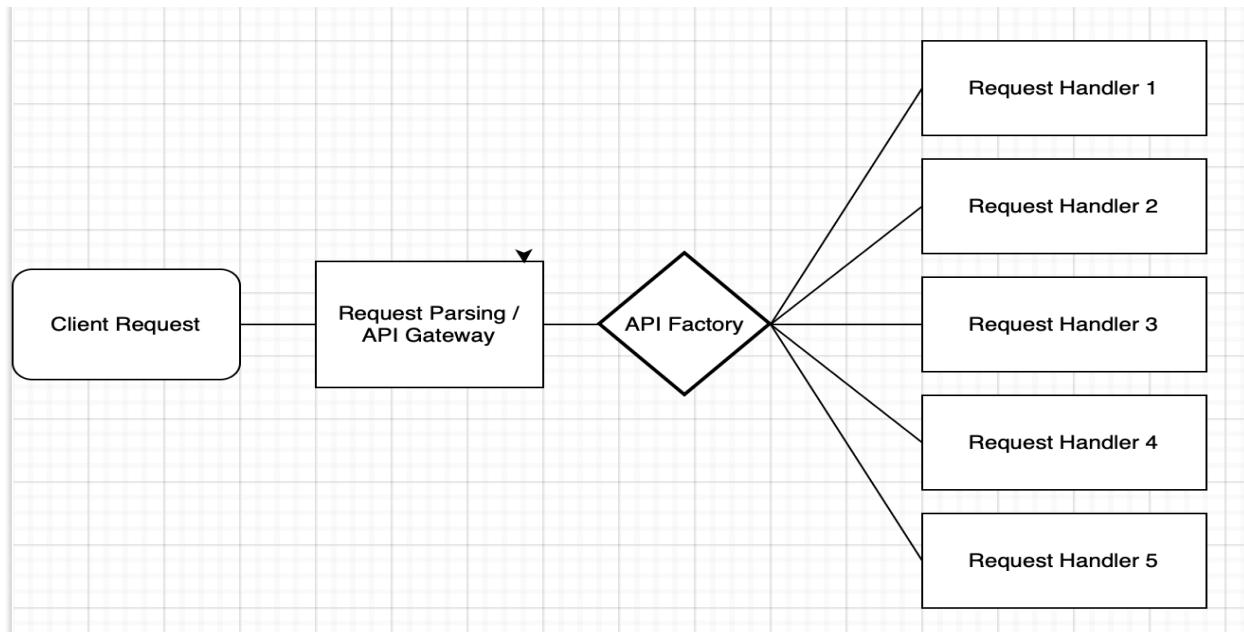
API HIGH LEVEL ARCHITECTURE

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.

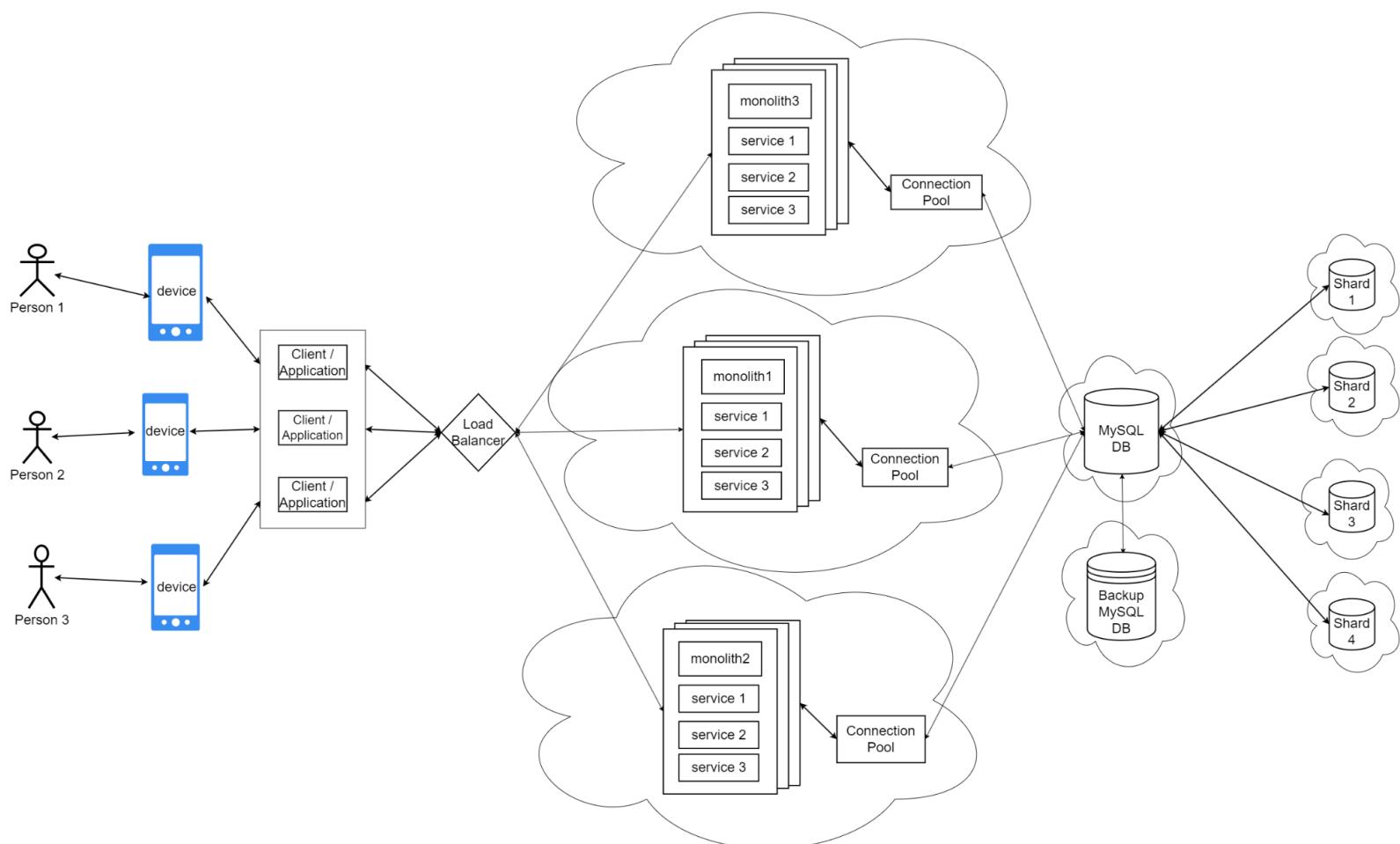
Users will interact with the platform through a React application deployed on EC2. This React front-end will communicate with a back-end server implemented using Express and TypeScript, also hosted on EC2. This setup ensures seamless interaction and maintains cross-language compatibility between the 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.

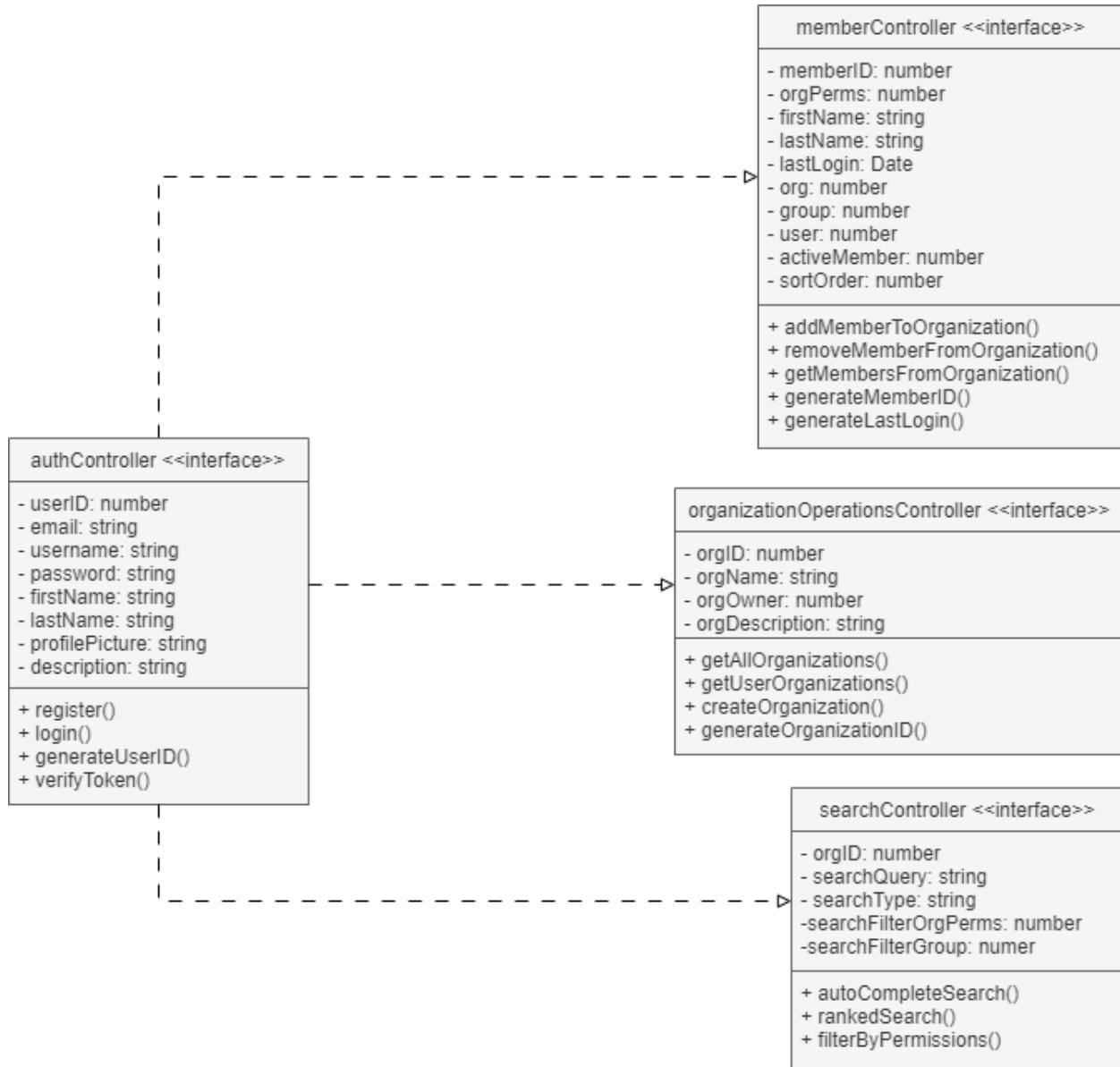
Database operations will be optimized using a connection pool, reusing connections for future database queries to reduce redundancy and manage resources efficiently. As the service grows, the database will be sharded, partitioning large datasets across multiple database nodes. This sharding strategy 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.

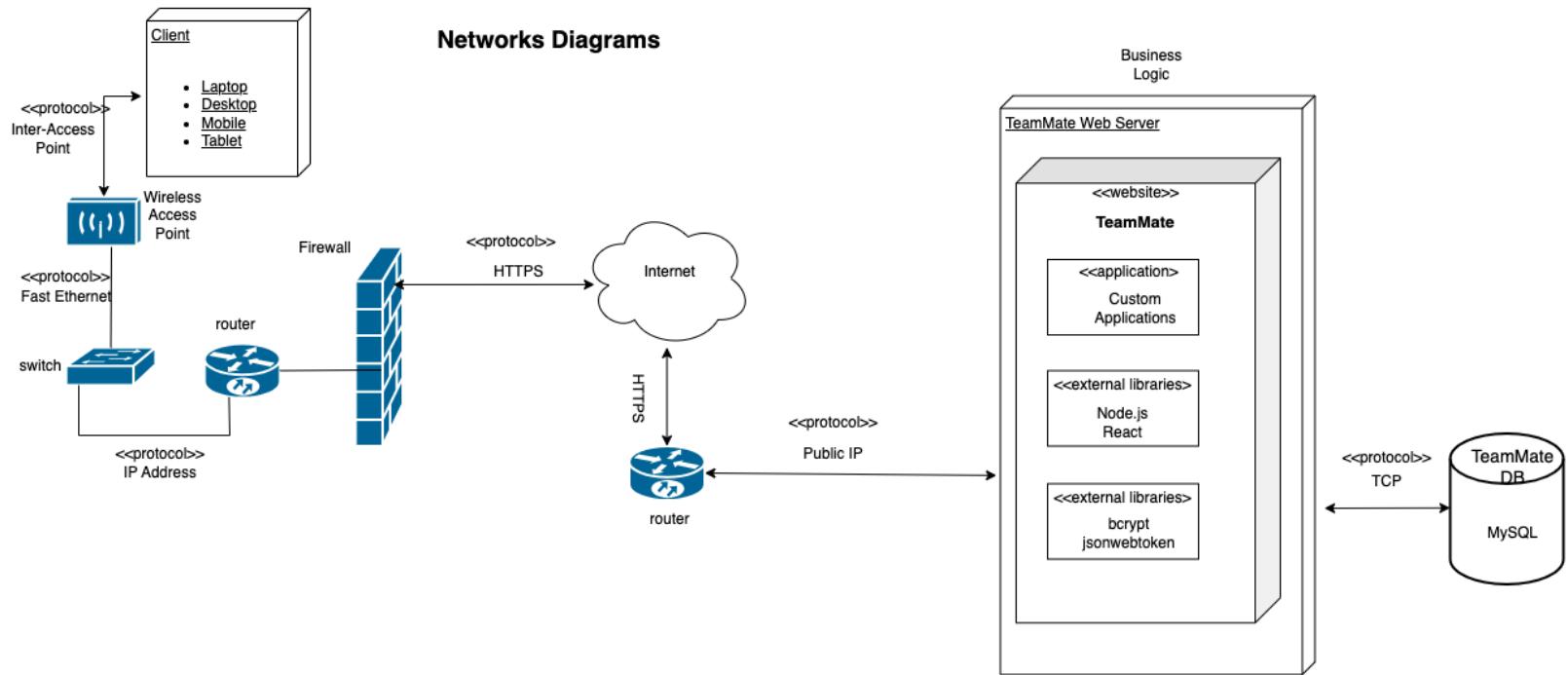
By integrating horizontal and vertical scaling methods, utilizing robust database management, and implementing message queuing for efficient communication, we can ensure that our monolithic architecture remains scalable, cost-effective, and maintainable. This design leverages AWS's capabilities to provide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



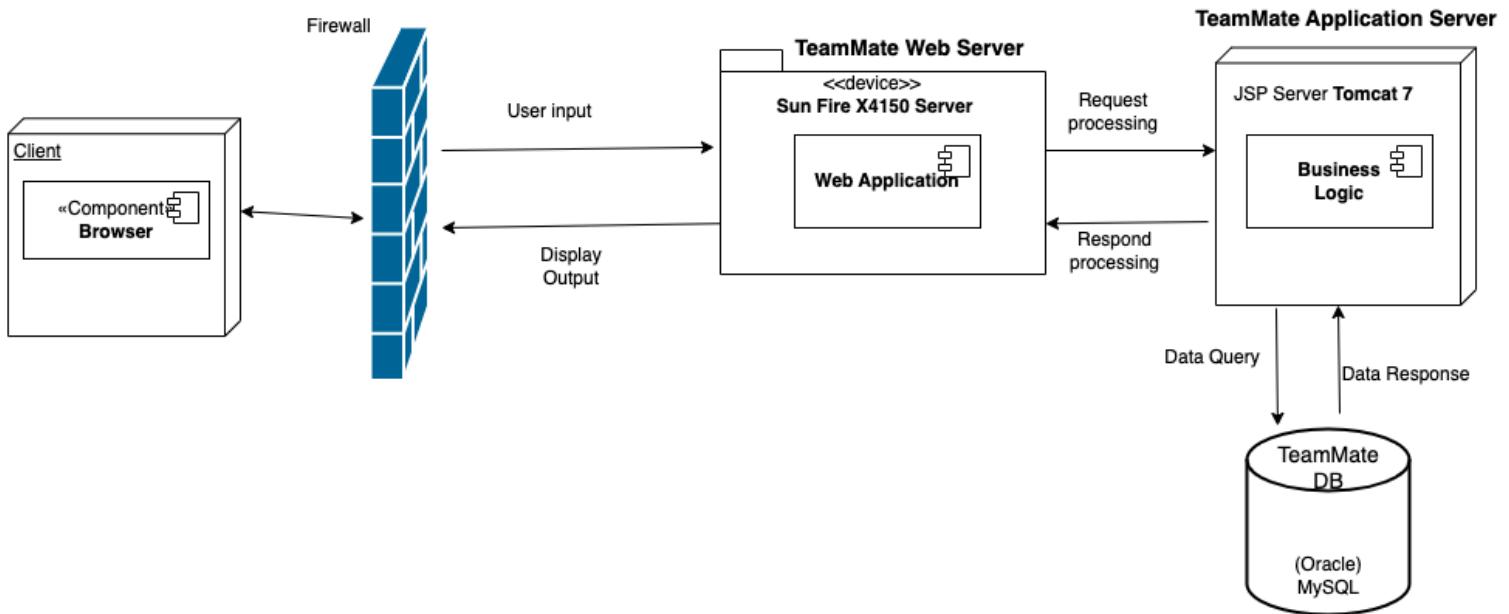
We are primarily using interfaces rather than classes. The interactions between the interfaces and the database tables are identical, and the database tables are well described in the EER (Enhanced Entity-Relationship) diagram. Below are the UML diagrams for our controllers and interfaces:



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

SW Engineering CSC648-848-05 Summer 2024

TeamMate

Personnel Management

Team 05 - Concat

Name	E-Mail	Role
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 3
July 23 2024

Version History

Revision	Date Submitted
M1	13 June 2024
M1V1	20 June 2024
M1V2	09 July 2024
M2V1	09 July 2024
M2V2	23 July 2024
M3V1	23 July 2024
M3V2	30 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>Wireframes Based on your Mockups/Storyboards (detailed)</u>	12
Use Case #1	12
Use Case #2	22
Use Case #3	28
Use Case #4	32
Use Case #5	37
Use Case #6	40
Use Case #7	44
Use Case #8	47
Use Case #9	50
Use Case #10	54
Use Case #11	57
Use Case #12	61
<u>High Level Database Architecture</u>	64
DBMS Summary	64
Functional Requirements	64
ERD (Entity Relationship Diagram)	68
EER (Enhanced Entity-Relationship)	69
Media Storage:	69
<u>High Level APIs and Main Algorithms</u>	70
<u>System Design</u>	72
<u>High Level Application Network and Deployment Design</u>	75
<u>Team Contributions</u>	77

Data Definitions

- Registered Users (Anyone with a registered account)
 - 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 (Organizations created by a registered user)
 - Name
 - Plaintext title.
 - Must be unique between all organizations.
 - Owner
 - Connected to a registered user.
 - Transferable to another registered user.
 - Description
- Members (Association between a user and an organization)
 - 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 (Member created category for dividing people)
 - 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 (Member created event that other members can “join”)
 - 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 (Associates a Member to an Event to track attendance)
 - Event
 - Connected to an event.
 - Member
 - Connected to a member.
 - Attendance
 - Enumerated type that records if the member was present or absent.

- Qualification Type (Member-defined custom qualifications)
 - Title
 - Description
 - Image
 - PNG or JPG/JPEG.
 - 400x400 as the max dimensions.
 - No greater than 2 megabytes.
- Qualification Instance (An instance of a qualification assigned to a member)
 - 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 (Member-defined custom awards)
 - Title
 - Description
 - Image
 - PNG or JPG/JPEG.
 - 400x400 as the max dimensions.
 - No greater than 2 megabytes.
- Award Instance (An instance of an award assigned to a member)
 - 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 (Members can leave comments on other member's pages)
 - 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 (Customizable section usable for whatever the client needs)
 - 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.

Wireframes Based on your Mockups/Storyboards (detailed)

[Link to Figma prototype](#)

[Link to wireframes](#)

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.

Use Case #1: Leader's Perspective

1. Start - Homepage

The screenshot shows the homepage of the TeamMate platform. At the top right, there are two buttons: "Sign In" and "Sign Up". The "Sign Up" button is circled in red and has a number "1" above it. Below the buttons, the main heading reads "Recognize, Reward, and Reimagine Team Collaboration". A descriptive paragraph follows: "Transform your organization with TeamMate. Our platform highlights individual achievements, tracks certifications, and manages events, fostering a culture of recognition and continuous improvement. Join TeamMate and see your team thrive." Below the text are three cards: "Track Members", "Create and Assign Tasks", and "Provide Recognitions". Each card has an icon and a brief description.

Sign Up 1

Recognize, Reward, and Reimagine Team Collaboration

Transform your organization with TeamMate. Our platform highlights individual achievements, tracks certifications, and manages events, fostering a culture of recognition and continuous improvement. Join TeamMate and see your team thrive.

Track Members
Keep track of your organization's members, their roles, and their progress with ease

Create and Assign Tasks
Effortlessly create and assign tasks and projects to your team members, ensuring everyone stays on track

Provide Recognitions
Motivate your team by providing recognitions and awards for their achievements and contributions

2. Sign Up Page

The screenshot shows the "Create an account" page. It features a form with fields for First Name, Last Name, Email Address, Password, and Confirm Password. There is also a checkbox for agreeing to the Terms of Service and Privacy Policy. At the bottom, there is a large "Create Account" button. A red box highlights the "First Name" and "Last Name" fields, with a red circle labeled "2" above them. Another red box highlights the "Create Account" button, with a red circle labeled "3" above it. Red arrows point from the "2" and "3" labels down to their respective highlighted areas.

Create an account

First Name Last Name
User A

Email Address
User@gmail.com

Password

Confirm Password

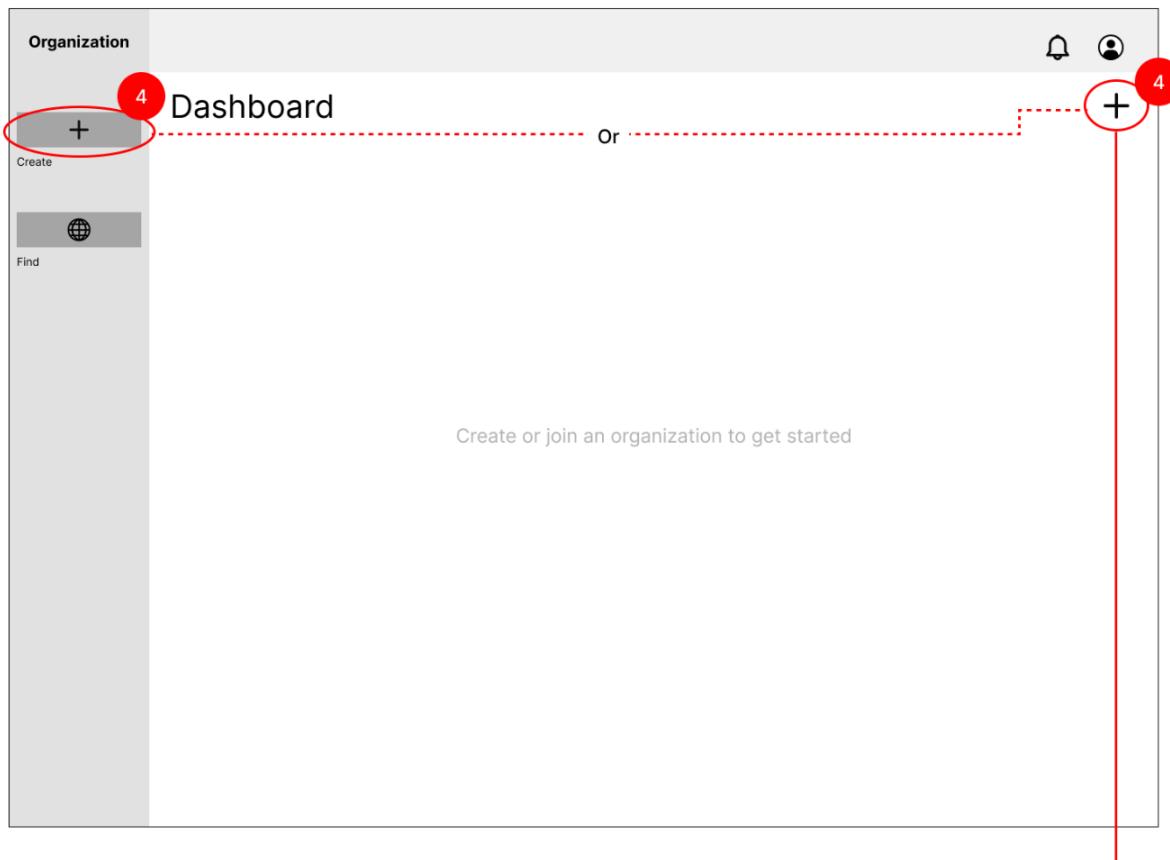
I agree to the [Terms of Service](#) and [Privacy Policy](#)

Create Account 3

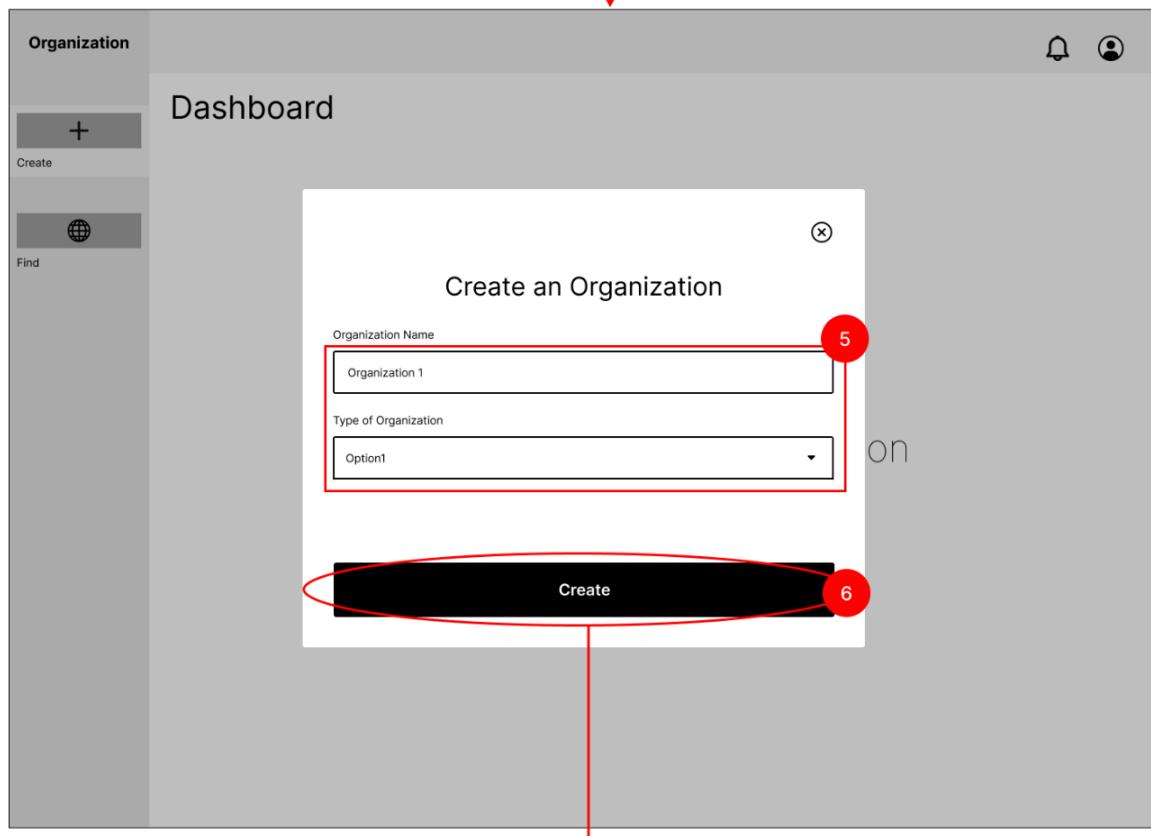
Already have an account? [Sign in](#)

Use Case #1: Leader's Perspective

3. Empty Organization Dashboard Page



4. Empty Organization Dashboard Page / Create Organization



5. Organization Dashboard Page

Use Case #1: Leader's Perspective

The screenshot shows the Organization Dashboard Page for "Organization 1". The left sidebar includes a "Create" button, a "1 Organization 1" indicator, and a "Find" button. The main content area displays several cards: "Members", "Groups" (which is circled in red with a red number "7" indicating an alert), "Events", "Tasks/Projects", "Recognition", and "Highlights".

6. Empty Group List Page

The screenshot shows the Empty Group List Page under the "Groups" tab. The left sidebar is identical to the dashboard. The main area features a search bar with a magnifying glass icon and the word "Search", and a prominent red "+ Add" button with a red number "8" indicating new items. A message "No Groups Created" is displayed below the search bar.

Use Case #1: Leader's Perspective

7. Empty Group List Page / Create Group

Create a Group

Group Name: Group 1 (Red Box)

Description (10/100 words): Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Access: Invite Only (Red Box)

Member Limit: 5

Create (Red Oval)

8. Group List Page

All Groups

Group 1 (Red Box)

Invite Only

Description

3/5

11

Use Case #1: Leader's Perspective

9. Group Detail Page

The screenshot shows a user interface for a group detail page. On the left is a sidebar with 'Organization' (Create, +, Organization 1, Find), 'Page Title' (1), and 'Description' (Invite Only, 0 Members, N/A Members Limit). The main content area shows 'Group 1' with 'Edit' and 'Add Member' buttons. Below is a 'Members' section with 'No group members yet'. The top navigation bar includes tabs for Members, Groups, Events, Tasks/Projects, Recognition (circled in red with a value of 12), and Highlights, along with a bell icon and user profile.

10. Empty Recognition List Page

The screenshot shows a user interface for a recognition list page. On the left is a sidebar with 'Organization' (Create, +, Organization 1, Find). The main content area shows 'Recognition' with a search bar and a '+ Create' button (circled in red with a value of 13). Below is a message 'No Recognition Created'. The top navigation bar includes tabs for Members, Groups, Events, Tasks/Projects, Recognition (highlighted), and Highlights, along with a bell icon and user profile.

Use Case #1: Leader's Perspective

11. Recognition List Page / Create Recognition

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Search +

Recognition

Create

1 Organization 1

Find

Create a Recognition

Recognition Name

Description (5/50 words)

Type of Recognition

Recognition 1

>Lorem ipsum dolor sit amet.

Option1

Create

12. Recognition List Page

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Search +

Recognition

Create

1 Organization 1

Find

Option 1

Recognition 1

Recognition 1
Lorem ipsum dolor sit amet.

Use Case #1: Leader's Perspective

11. Recognition Detail Page

The screenshot shows the 'Recognition Details' page. On the left is a sidebar with 'Organization' (1), 'Create', and 'Find' buttons. The main area shows 'Recognition 1' with a 'Description' placeholder and a 'Members' section indicating 'No members yet'. A 'Share' button is present. At the top, a navigation bar includes 'Members' (17), 'Groups', 'Events', 'Tasks/Projects', 'Recognition', 'Highlights', and user icons.

12. Members' List Page

The screenshot shows the 'Members' list page. It features a sidebar with 'Organization' (1), 'Create', and 'Find' buttons. The main content area is titled 'Members' with a search bar and a large '+' button circled in red with the number '18'. Below is a table with columns 'Name', 'Groups', and 'Roles'. A single row is shown with a user icon, 'Name (You)', 'Groups', 'Roles', and an ellipsis. The top navigation bar includes 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', 'Highlights', and user icons.

Use Case #1: Leader's Perspective

13. Members' List Page / Add Member

The screenshot shows the 'Members' list page with a modal window titled 'Add a Member'. The modal contains two input fields: 'Member's ID' with the value 'Member 1' and 'Assign Role' with the value 'Member'. A red box highlights the 'Create' button at the bottom of the modal, which is circled with a red arrow labeled 20.

14. Members' List Page

The screenshot shows the 'Members' list page after the new member 'Member 1' has been added. The member is listed in the table with columns 'Name', 'Groups', and 'Roles'. The 'Name' column shows 'Member 1', the 'Groups' column shows 'Groups', and the 'Roles' column shows 'Member'. A red box highlights the member row, which is circled with a red arrow labeled 21.

Use Case #1: Leader's Perspective

15. Members' Profile Page

Organization

- + Create
- 1 Organization 1
- Find

Member's Profile

Member 1

Profile

Group:	N/A
Role:	Member

Recognition

No recognitions yet



16. Members' Profile Page / Edit

Organization

- + Create
- 1 Organization 1
- Find

Member's Profile

Member 1

Profile

Group:	N/A
Role:	Admin

Recognition

Cancel **Save** 24

23

Use Case #1: Leader's Perspective

17. End - Members' Profile Page

The screenshot shows a user interface for managing members. On the left, there is a sidebar with a 'Create' button and a 'Find' button. The main area is titled 'Member's Profile' and displays a single member named 'Member 1'. The member's profile includes fields for 'Group' (N/A) and 'Role' (Admin). Below the profile, there is a section for 'Recognition' which states 'No recognitions yet'. The top navigation bar includes links for 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', and 'Highlights', along with a notification bell icon and a user profile icon.

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.

Use Case #2: Leader's Perspective

1. Start - Organization Dashboard Page

The screenshot shows the 'Organization' dashboard for 'Organization 1'. On the left sidebar, there are buttons for '+ Create' and '1 Organization 1'. Below the sidebar, a red circle labeled '1' is drawn around the 'Members' box. A red arrow points from this circle down to the 'Members' section of the main content area. The main content area contains six boxes: 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', and 'Highlights'.

2. Members' List Page

The screenshot shows the 'Members' list page. The top navigation bar includes tabs for 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', and 'Highlights'. The 'Members' tab is active. The main content area displays a table with columns: 'Name', 'Groups', and 'Roles'. A new row is being added, indicated by a red circle labeled '2' around the '+' button in the top right corner of the table header. A red arrow points from the 'Members' box on the previous screen down to this '+ 2' button.

Use Case #2: Leader's Perspective

3. Members' List Page / Add Member

The screenshot shows the 'Members' list page with a modal overlay titled 'Add a Member'. The modal contains two input fields: 'Member's ID' with the value 'Member 1' and 'Assign Role' with the value 'Member'. A red box highlights the 'Create' button at the bottom of the modal. A red arrow points from this button down to the main 'Members' list page, indicating the successful addition of the new member.

4. Members' List Page

The screenshot shows the 'Members' list page after a new member has been added. The newly added member, 'Member 1', is highlighted with a red box and a red circle labeled '5' on the 'Name' column. A red arrow points from the bottom of the page up to the 'Name' column of the newly added member, indicating the successful addition.

Use Case #2: Leader's Perspective

5. Members' Profile Page

Organization

- + Create
- 1 Organization 1
- Find

Member's Profile

Member 1

Profile

Group:	N/A
Role:	Member

Recognition

No recognitions yet

6. Members' Profile Page / Edit

Organization

- + Create
- 1 Organization 1
- Find

Member's Profile

Member 1

Profile

Group:	N/A
Role:	Member

Recognition

+

Cancel Save

Use Case #2: Leader's Perspective

7. Members' Profile Page / Edit / Add Recognition

Add Recognition

Certifications

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Recognition 1 Lorem ipsum dolor sit amet.	Title Description	Title Description

Add

8. Members' Profile Page / Edit

Member's Profile

Profile

Group: N/A

Role: Member

Recognition

Recognition 1
Lorem ipsum dolor sit amet.

Cancel **Save**

Use Case #2: Leader's Perspective

9. End - Members' Profile Page

The screenshot shows a user interface for managing members. On the left is a vertical sidebar with the following sections:

- Organization**: Includes a plus sign icon for creating new organizations, a "Create" button, and a section for "Organization 1" which has a count of "1".
- Find**: A search bar.

The main content area is titled "Member's Profile" and displays the following information for "Member 1":

- Profile**: Shows the member's profile picture and role. Details:
 - Group: N/A
 - Role: Member
- Recognition**: Shows a single recognition entry. Details:
 - Recognition 1**: An image placeholder.
 - Description: Lorem ipsum dolor sit amet.

At the top of the main content area, there are navigation links: Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. There are also notification and user profile icons in the top right corner.

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.

Use Case #3: Leader's Perspective

1. Start - Organization Dashboard Page

The screenshot shows the 'Organization' dashboard for 'Organization 1'. On the left sidebar, there are buttons for '+ Create' and 'Find'. The main area displays 'Organization 1' with a count of '1' under 'Organization 1'. Below this are four cards: 'Members', 'Groups', 'Events', and 'Tasks/Projects'. The 'Tasks/Projects' card is circled in red with a number '1' above it. A red arrow points from this card down to the 'Tasks/Projects' page.

2. Empty Tasks/Projects List Page

The screenshot shows the 'Tasks/Projects' list page. The top navigation bar includes tabs for 'Members', 'Groups', 'Events', 'Tasks/Projects' (which is active), 'Recognition', and 'Highlights'. There is also a search bar with a magnifying glass icon and a red '+' button. A red circle with the number '2' is drawn around the '+' button. A red arrow points from the bottom of the 'Tasks/Projects' card on the previous page down to this page.

Use Case #3: Leader's Perspective

3. Empty Tasks/Projects List Page / Create Task/Project

The screenshot shows the 'Tasks/Projects' creation interface. A red box highlights the input fields for 'Title' and 'Description'. A red circle with the number '3' is positioned above the 'Create' button. A red oval surrounds the entire 'Create' button area, with a red circle containing the number '4' at its bottom right corner.

Create a Task/Project

- Name:
- Description:
- Upload Content:
- Assigned to:
- Deadline:

Create

4. Tasks/Projects List Page

The screenshot shows the 'Tasks/Projects' list page after a task has been created. A red box highlights the task card for 'Title'. A red circle with the number '5' is positioned above the task card. A red line connects the 'Create' button from the previous screen to the task card on this screen.

Current Tasks/Projects

	Title	07/16/2024
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.		

Use Case #3: Leader's Perspective

5. Tasks/Projects Detail Page

The screenshot shows a user interface for managing tasks/projects. On the left is a sidebar with 'Organization' (Create, +), '1 Organization 1', and 'Find'. The main area has tabs: Members, Groups, Events, Tasks/Projects (selected), Recognition, and Highlights. A red circle labeled '6' is on the 'Edit' button. Below it, a large gray box contains a placeholder image. A red dashed arrow labeled 'Scroll Down' points to a 'Description' section with placeholder text. Another red dashed arrow points to a 'Submissions' table. The table has columns: Name, Submitted Date, and Review. Under 'Name' is 'Member 1'. Under 'Submitted Date' is '00/00/2024'. Under 'Review' is a red box containing the word 'View', with a red circle labeled '7' above it.

6. End - Tasks/Projects Submission View Page

The screenshot shows the 'Tasks/Projects Submission View Page' for 'Member 1'. The top navigation bar is identical to the previous page. The main content area displays the title 'The Name of the Task/Project'. Below it, under 'Content', there is a placeholder 'N/A'. Under 'Submitted Text', there is a text input field with the placeholder 'Type'. A red arrow points from the 'View' button in the previous page's screenshot down to the 'View' button in this page's screenshot.

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.

Use Case #4: Member's Perspective

1. Start - Organization Dashboard Page

Organization

+ Create

2 Organization 2

Members Groups Events

Tasks/Projects Recognition Highlights

Find

Bell User icon

1

2. Empty Tasks/Projects List Page

Organization

+ Create

2 Organization 2

Members Groups Events Tasks/Projects Recognition Highlights

Search

2

Task/Project 1

00/00/2024

Title

Description

00/00/2024

Title

Description

00/00/2024

Find

Bell User icon

2

Use Case #4: Member's Perspective

3. Tasks/Projects Detail Page

The screenshot shows a web application interface for managing tasks and projects. At the top, there is a navigation bar with tabs: Organization, Members, Groups, Events, Tasks/Projects (which is the active tab), Recognition, and Highlights. Below the navigation bar, there is a header section titled "Tasks/Projects Details". On the left side, there is a sidebar with a "Create" button and a list item "Organization 1". The main content area displays a task entry form for "Task/Project 1". The form includes a "Text Entry Upload" dropdown menu, a deadline field set to "00/00/2024", a "Description" text area containing placeholder text, and a "Text Entry" section with its own placeholder text. A red circle labeled "3" is positioned over the "Text Entry" section. A red arrow points from the "Upload Task/Project" button (circled with a red oval) down to the "Text Entry" section.

4. Tasks/Projects Detail Page / Upload

This screenshot shows the same application interface after a task has been uploaded. The "Tasks/Projects" tab is still active. The main content area displays the "Task/Project 1" details, including the "Text Entry Upload" dropdown, the "Text Entry" section, and the "Description" text area. A modal window is open in the center, displaying the message "Successfully Uploaded Your Task/Project." with a "Okay" button. A red circle labeled "5" is positioned over the "Okay" button. A red arrow points from the "Okay" button back up to the "Text Entry" section.

Use Case #4: Member's Perspective

5. Tasks/Projects Detail Page

Organization

Members Groups Events Tasks/Projects **6** Recognition Highlights

Tasks/Projects Details

Task/Project 1

Deadline: 00/00/2024

Description

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Text Entry

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Re-upload Task/Project

6. End - Tasks/Projects List Page

Organization

Members Groups Events Tasks/Projects **Recognition** Highlights

Tasks/Projects

Current Tasks/Projects

Title	Description	Deadline
		00/00/2024
		00/00/2024

Past Tasks/Projects

Title	Description

Task/Project 1

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut

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.

Use Case #5: Leader's Perspective

1. Start - Members' Profile Page

The screenshot shows the Member's Profile page. At the top, there is a navigation bar with tabs: Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. The Highlights tab is highlighted with a red circle containing the number 1. Below the navigation bar, the title "Member's Profile" is displayed. On the left side, there is a sidebar with sections for Organization (Create, Organization 1), Find, and a globe icon. The main content area starts with a section for "Member 1" featuring a profile picture and an "Edit" button. Below this is a "Profile" section with fields for Group (N/A) and Role (Member). The next section is "Recognition", which displays four items, each with a small image, a title, and a description. A red arrow points from the "Highlights" tab on the navigation bar down to the "Highlights" section on the page.

2. Empty Highlights List Page

The screenshot shows the Highlights list page. At the top, there is a navigation bar with tabs: Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. The Highlights tab is highlighted with a red circle containing the number 2. Below the navigation bar, the title "Highlights" is displayed. On the left side, there is a sidebar with sections for Organization (Create, Organization 1), Find, and a globe icon. The main content area shows a search bar with a magnifying glass icon and the word "Search". Below the search bar, the text "No Tasks/Projects Created" is displayed. A red arrow points from the "Highlights" tab on the navigation bar down to the "Highlights" section on the page.

Use Case #5: Leader's Perspective

3. Empty Highlights List Page / Create Highlight

The screenshot shows the 'Highlights' section of a web application. A red box highlights the 'Create a Highlight' form. A red circle labeled '3' points to the 'Name:' field containing 'Highlight 1'. Another red circle labeled '4' points to the 'Create' button at the bottom of the form.

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Search

+

1 Organization 1

Create

Find

Highlights

Create a Highlight

Name: Highlight 1

Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Upload Content: File.jpg

Upload File

Tag Members: Type member's name, everyone, or group name

Member 1 Member 2 Member 3

Create

4. End - Highlights List Page

The screenshot shows the 'Highlights' section of the application after a highlight has been created. It displays a single highlight card for 'Highlight 1', which was published on 07/16/2024. The card includes a placeholder image and a link to 'Edit'.

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Search

+

1 Organization 1

Create

Find

Highlights

Highlight 1

Published: 07/16/2024

Edit

Tagged Members

Member 1 and others

Find

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.

Use Case #6: Leader's Perspective

1. Start - Empty Event List Page / Create Event

The screenshot shows the 'Create an Event' form. A red box highlights the input fields for Name, Upload Content, Description, Event Type, Location or Link, Date & Time, and Hosted by. A red circle labeled '1' points to the 'Events' tab in the top navigation bar. A red circle labeled '2' points to the '+' button in the top right corner. A red circle labeled '3' points to the 'Create' button at the bottom of the form. A red arrow points from the 'Create' button down to the 'Events' list page.

Create an Event

- Name: Event 1
- Upload Content: File.jpg
- Description: Lorem ipsum dolor sit amet.
- Event Type: In-Person
- Location or Link: Location Address
- Date & Time: 07/16/2024 1 pm to 07/16/2024 2 pm
- Hosted by: Type member's name or group name

Create

2. Event List Page

The screenshot shows the 'Events' list page. A red box highlights the event card for 'Event 1'. A red circle labeled '5' points to the event card. A red arrow points from the event card back up to the 'Create an Event' form.

Events

Upcoming Events

Event 1
Lorem ipsum dolor sit amet.
07/16/2024/1pm – 2pm

Use Case #6: Leader's Perspective

3. Event Detail Page

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Description
Lorem ipsum dolor sit amet.

Attendees

Name	Joined Date	Group
Member 1	00/00/2024	N/A
Member 2	00/00/2024	N/A

Set Up Groups 7

4. Event Detail Page / Set Up Group

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Create Groups

Number of Groups: 2

Group Assign Type: Assign Automatically

Attendees

Name	Joined Date	Group
Member 1	00/00/2024	N/A
Member 2	00/00/2024	N/A

Set Up Groups 8

Use Case #6: Leader's Perspective

5. Event Detail Page / Assign Group

Assign Groups

Group 1
Member 1

Group 2
Member 2

Attendees

Save 9

Name	Joined Date	Group
Member 1	00/00/2024	N/A
Member 2	00/00/2024	N/A

6. End - Event Detail Page

Description

Member 1

Member 2

Attendees

Set Up Groups

Name	Joined Date	Group
Member 1	00/00/2024	Group 1
Member 2	00/00/2024	Group 2

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.

Use Case #7: Leader's Perspective

1. Start - Empty Tasks/Projects List Page / Create Task/Project

The screenshot shows a user interface for managing tasks and projects. On the left, there's a sidebar with a '+ Create' button and a '1 Organization 1' section. The main area is titled 'Tasks/Projects'. At the top, there are tabs for 'Members', 'Groups', 'Events', 'Tasks/Projects' (which is highlighted with a red circle labeled '1'), 'Recognition', and 'Highlights'. To the right of the tabs is a search bar and a user profile icon. In the center, there's a 'Current Tasks/Projects' section with a placeholder image and some placeholder text. Below it is a 'Create a Task/Project' form with fields for 'Name' (with 'Title' as a placeholder), 'Description' (with placeholder text about Lorem ipsum), 'Upload Content' (with a file input field containing 'File.jpg' and a 'Upload File' button), 'Assigned to' (with a text input field containing 'Type member's name, everyone, or group name' and a 'Member 1' entry with a delete button), and 'Deadline' (with a date input field set to '07/16/2024'). At the bottom of the form is a large black 'Create' button. A red box labeled '3' encloses the entire form area. A red arrow labeled '4' points downwards from the 'Create' button towards the bottom of the page.

2. Tasks/Projects List Page

This screenshot shows the same interface after a task has been created. The 'Tasks/Projects' tab is still selected. The 'Current Tasks/Projects' section now displays a single task card. The card features a placeholder image, the title 'Title', a description placeholder text about Lorem ipsum, and a deadline of '07/16/2024'. The rest of the interface remains the same, with the sidebar, other tabs, and the overall layout.

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.

Use Case #8: Member's Perspective

1. Start - Sign In Page

TeamMate

Welcome, Mate!

Email Address

Password

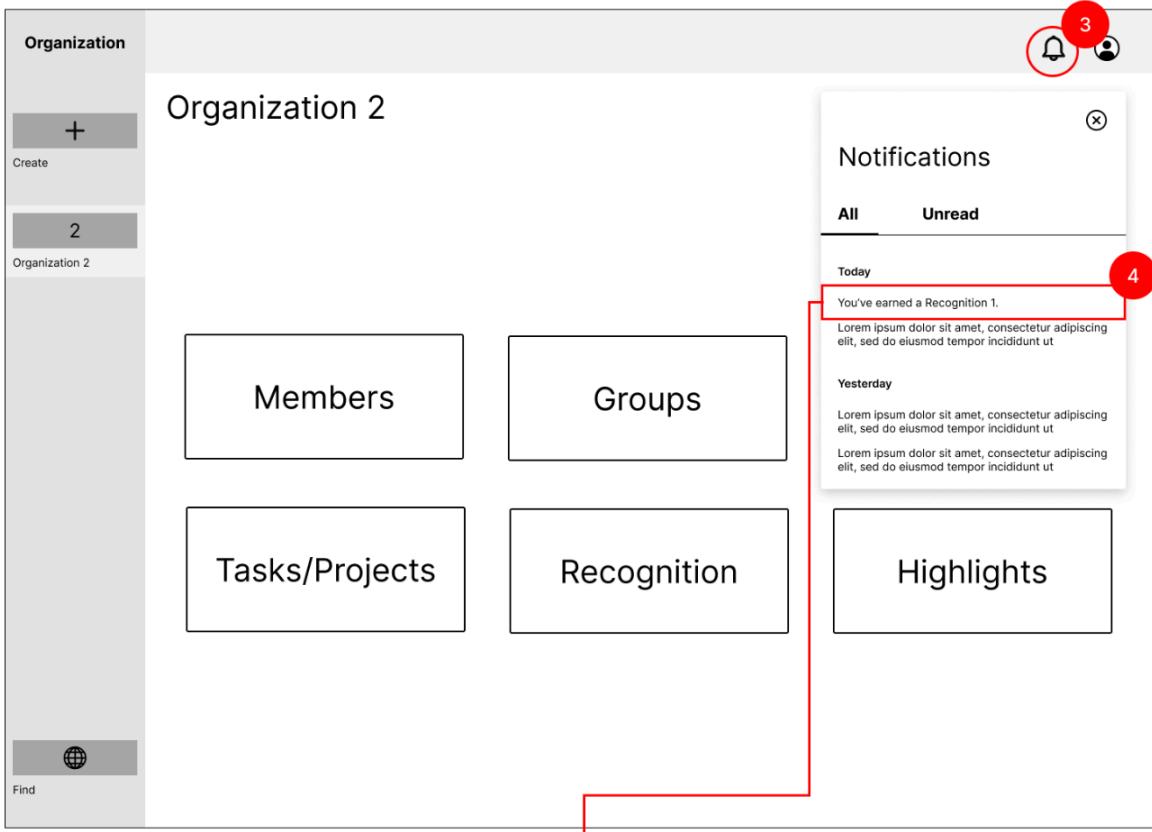
Remember Me [Forgot Password?](#)

Sign In

Or

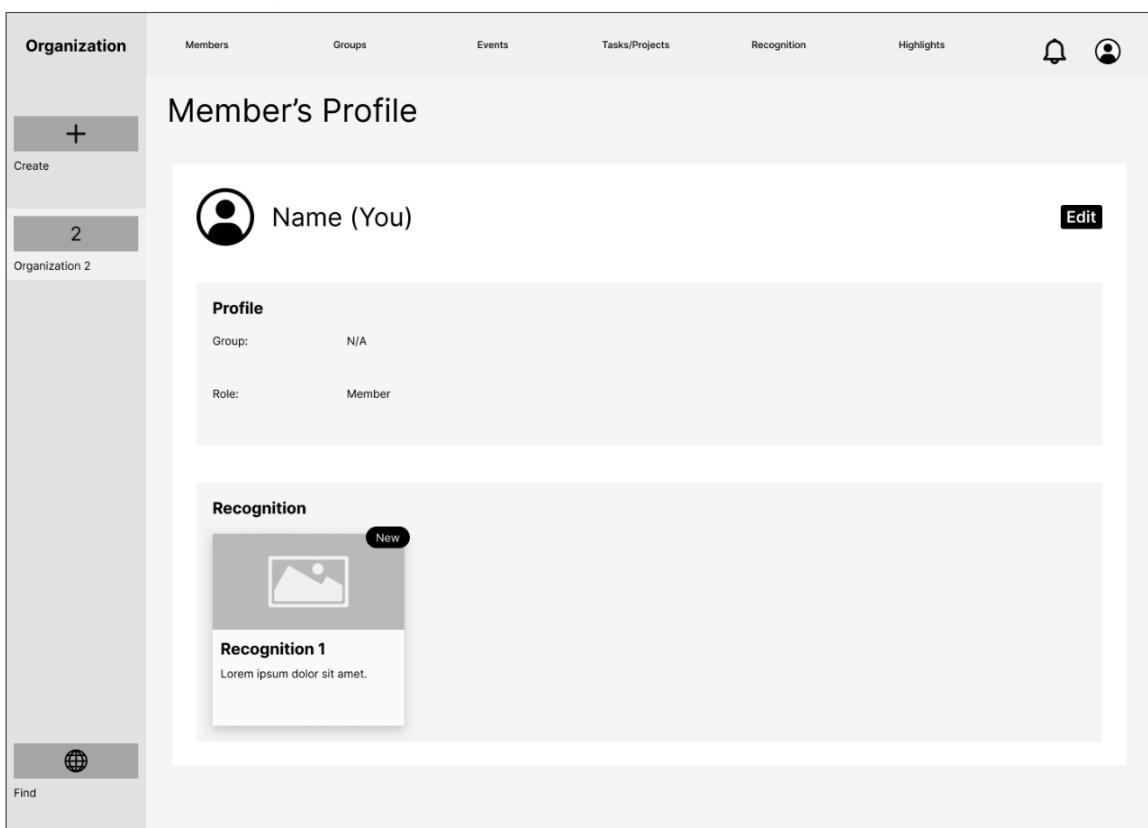
[Create an Account](#)

2. Organization Dashboard Page / Notofication



Use Case #7: Member's Perspective

3. End - Member's Profile Page / New Notification



The screenshot shows a user interface for a member's profile page. On the left, there is a sidebar with a 'Create' button and a 'Find' button. The main content area is titled 'Member's Profile' and displays the user's name ('Name (You)') with an 'Edit' button. Below this, there are sections for 'Profile' (Group: N/A, Role: Member) and 'Recognition' (Recognition 1, which is a placeholder for 'Lorem ipsum dolor sit amet.' and has a 'New' button). The top navigation bar includes links for Members, Groups, Events, Tasks/Projects, Recognition, and Highlights, along with a notification bell icon and a user profile icon.

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.

Use Case #9: Member's Perspective

1. Start - Groups List Page

The screenshot shows the 'Groups' tab selected in the top navigation bar (circled in red with number 1). The sidebar on the left includes a 'Create' button (circled in red with number 2) and an 'Organization 2' section. The main area displays a search bar and a message: 'You're not in any groups'. Below this, there are sections for 'My Groups' and 'All Groups'. Under 'All Groups', three cards are shown, each with a thumbnail, title, and status (e.g., 'Invite Only', 'Full'). The third card, 'Group 1', is highlighted with a red box and circled with number 2. A red arrow points from the 'Join' button in the Groups Detail Page below to this card.

2. Groups Detail Page

The screenshot shows the details for 'Group 1'. The top navigation bar is identical to the previous page. The sidebar includes a 'Find' button. The main content area shows the group's title 'Group 1', its description ('Anyone Can Join | 4 Members | 5 Members Limit'), and a 'Description' section with placeholder text. A 'Join' button is highlighted with a red box and circled with number 3. Below this, there is a 'Members' section with a table:

Name	Joined Date	Role
Member 1	00/00/2024	N/A
Member 2	00/00/2024	N/A
Member 3	00/00/2024	N/A

A red arrow points from the bottom of the Groups List Page to the 'Join' button on this page.

Use Case #9: Member's Perspective

3. Groups Detail Page / Join

The screenshot shows the Groups Detail Page. On the left sidebar, there are buttons for 'Create' (with a plus sign) and 'Organization 2'. At the bottom are 'Find' and a magnifying glass icon. The main area has a title 'Groups' and a sub-section 'Group 1'. Below it are status indicators: 'Anyone Can Join', '4 Members', and '5 Members Limit'. A 'Description' box contains placeholder text. A modal window titled 'Join the Group?' is open, containing the button 'Yes, I'll Join' which is circled in red and labeled '4'. Below the modal is a table titled 'Members' with three entries: Member 1, Member 2, and Member 3, all listed under 'Joined Date: 00/00/2024' and 'Role: N/A'.

4. Groups List Page

The screenshot shows the Groups List Page. The left sidebar is identical to the previous page. The main area has a title 'Groups' and a search bar. Below it is a section titled 'My Groups' featuring 'Group 1' with a 'Joined' button. Underneath is a '5/5' rating icon. Below this is a section titled 'All Groups' showing three groups: 'Title' (status 'Invite Only'), 'Title' (status 'Full'), and 'Group 1' (status 'Joined'). The 'Joined' button for 'Group 1' is circled in red and labeled '4'.

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.

Use Case #10: Leader's Perspective

1. Start - Tasks/Projects Submission View Page

The Name of the Task/Project

< Member 1 >

Content
N/A

Submitted Text
Type

Organization

Create

1 Organization 1

Find

Members Groups Events Tasks/Projects Recognition Highlights 1 Request Revision Complete

2. Empty Highlight List Page

Highlights

Search

+ 2

No Tasks/Projects Created

Organization

Create

1 Organization 1

Find

Members Groups Events Tasks/Projects Recognition Highlights + 2

Use Case #10: Leader's Perspective

3. Highlight List Page / Create Highlight

The screenshot shows the 'Create a Highlight' form. The 'Name' field contains 'Highlight 1'. The 'Description' field contains a long block of placeholder text about labor rights. The 'Upload Content' section shows a file named 'File.jpg' and a 'Upload File' button. The 'Tag Members' section has a search bar and three member tags: 'Member 1', 'Member 2', and 'Member 3'. A large red oval surrounds the entire form area, and a red circle labeled '4' is positioned over the 'Create' button at the bottom.

4. End - Highlight List Page

The screenshot shows the 'Highlights' page after creation. It displays a card for 'Highlight 1', which was published on 07/16/2024. The card features a placeholder image, an 'Edit' button, and a 'Tagged Members' section showing 'Member 1 and others'. Below the card is a paragraph of placeholder text.

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.

Use Case #11: Leader's Perspective

1. Start - Recognition List Page / Create Recognition

The screenshot shows the 'Recognition' tab selected in the top navigation bar. A modal window titled 'Create a Recognition' is open. Inside the modal, several input fields are visible: 'Recognition Name' (containing 'Recognition 1'), 'Description (5/50 words)' (containing 'Lorem ipsum dolor sit amet.'), and 'Type of Recognition' (containing 'Option1'). At the bottom of the modal is a large black button labeled 'Create'. A red arrow points from the 'Create' button down to the 'Recognition List Page' below.

Organization

Members Groups Events Tasks/Projects Recognition **1** Highlights

Search

Recognition

Create

1 Organization 1

+ Find

Create a Recognition

Recognition Name

Recognition 1

Description (5/50 words)

>Lorem ipsum dolor sit amet.

Type of Recognition

Option1

Create **4**

2. Recognition List Page

The screenshot shows the 'Recognition' tab selected in the top navigation bar. A single recognition item is listed under the heading 'Option 1'. The item includes a thumbnail image, the title 'Recognition 1', and a brief description 'Lorem ipsum dolor sit amet.'. A red arrow points from the 'Recognition 1' item down to the 'Recognition List Page' below.

Organization

Members Groups Events Tasks/Projects Recognition **1** Highlights

Search

Recognition

Create

1 Organization 1

+ Find

5

Option 1

Recognition 1

Recognition 1
Lorem ipsum dolor sit amet.

Use Case #11: Leader's Perspective

1. Start - Recognition Detail Page / Add Member

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Recognition Details

Recognition 1

Description

Add a Member

Member's ID

Member 1

Members

Create

No members yet

Share

Edit Add Member

2. Recognition Detail Page

Organization

Members Groups Events Tasks/Projects Recognition Highlights

Recognition Details

Recognition 1

Description

Member 1

Members

Name Issued Date

Member 1 00/00/2024

Share

Edit Add Member

Use Case #11: Leader's Perspective

3. Member's Profile Page

The screenshot shows the Member's Profile page. On the left is a sidebar with 'Organization' (1), 'Create', and 'Find' buttons. The main area has tabs for Members, Groups, Events, Tasks/Projects, Recognition, and Highlights. The 'Recognition' tab is active. It displays a profile for 'Member 1' with a placeholder image and the role 'Member'. Below this is a 'Recognition' section with a placeholder image and the text 'Recognition 1' and 'Lorem ipsum dolor sit amet.' A red circle highlights the 'Edit' button in the top right corner of the main content area.

4. End - Member's Profile Page / Edit

The screenshot shows the Member's Profile page in edit mode, indicated by the 'Cancel' and 'Save' buttons in the top right. The 'Profile' section now contains dropdown menus for 'Group' (set to 'N/A') and 'Role' (set to 'Admin'). The 'Recognition' section remains the same with its placeholder image and text.

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.

Use Case #12: Member's Perspective

1. Start -Organization Dashboard Page / Notification

The screenshot shows the Organization Dashboard Page. On the left, there is a sidebar with a '+ Create' button, a '2 Organization 2' section, and a 'Find' button. The main area is titled 'Organization 2' and contains five cards: 'Members', 'Groups', 'Tasks/Projects', 'Recognition', and 'Highlights'. In the top right corner, there is a 'Notifications' icon with a red badge containing the number '1'. A red arrow points from this icon to the 'Notifications' section on the right.

Notifications

All **Unread**

Today

You've earned a Recognition 1.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut

Yesterday

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut

2. Member's Profile Page / New Notification

The screenshot shows the Member's Profile page. The top navigation bar includes tabs for 'Members', 'Groups', 'Events', 'Tasks/Projects', 'Recognition', and 'Highlights', along with a notification bell and user profile icons. Below the navigation, the title 'Member's Profile' is displayed, followed by a profile card with a placeholder image and the text 'Name (You)' with an 'Edit' button. The 'Recognition' tab is selected, showing a card for 'Recognition 1' with the text 'New' and a small image placeholder. A red box highlights this card, and a red arrow points from it down to the bottom of the page. The bottom of the page features a footer with a 'Find' button.

Profile

Group: N/A

Role: Member

Recognition

New

Recognition 1
Lorem ipsum dolor sit amet.

Use Case #12: Member's Perspective

3. End - Recognition Detail Page

The screenshot shows a user interface for managing organization recognition. On the left, there is a sidebar with a 'Create' button and a 'Find' button. The main area displays a 'Recognition Details' card for 'Recognition 1'. The card includes a 'Description' section with placeholder text and a 'Members' section listing two members: 'Name (You)' and 'Member 2', each with their respective profile icons and issue dates.

Organization

Members Groups Events Tasks/Projects Recognition Highlights

+

Create

2

Organization 2

Recognition Details

Recognition 1

Description

>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do.

Members

Name	Issued Date
Name (You)	00/00/2024
Member 2	00/00/2024

Share

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 at most 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 many comments.
- 1.10. A registered user can make 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 many qualification instances.
- 2.7. A member can receive many award instances.
- 2.8. A member can give many qualification instances.
- 2.9. A member can give many award instances.
- 2.10. A member shall have an organization.

- 2.11. A member shall have 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 many groups.
- 3.6. An organization shall have many events.
- 3.7. An organization shall have one or many members.
- 3.8. An organization shall have 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.
- 5.6. An award type will have an associated org

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.
- 7.6. A qualification type will have an associated org

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 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.

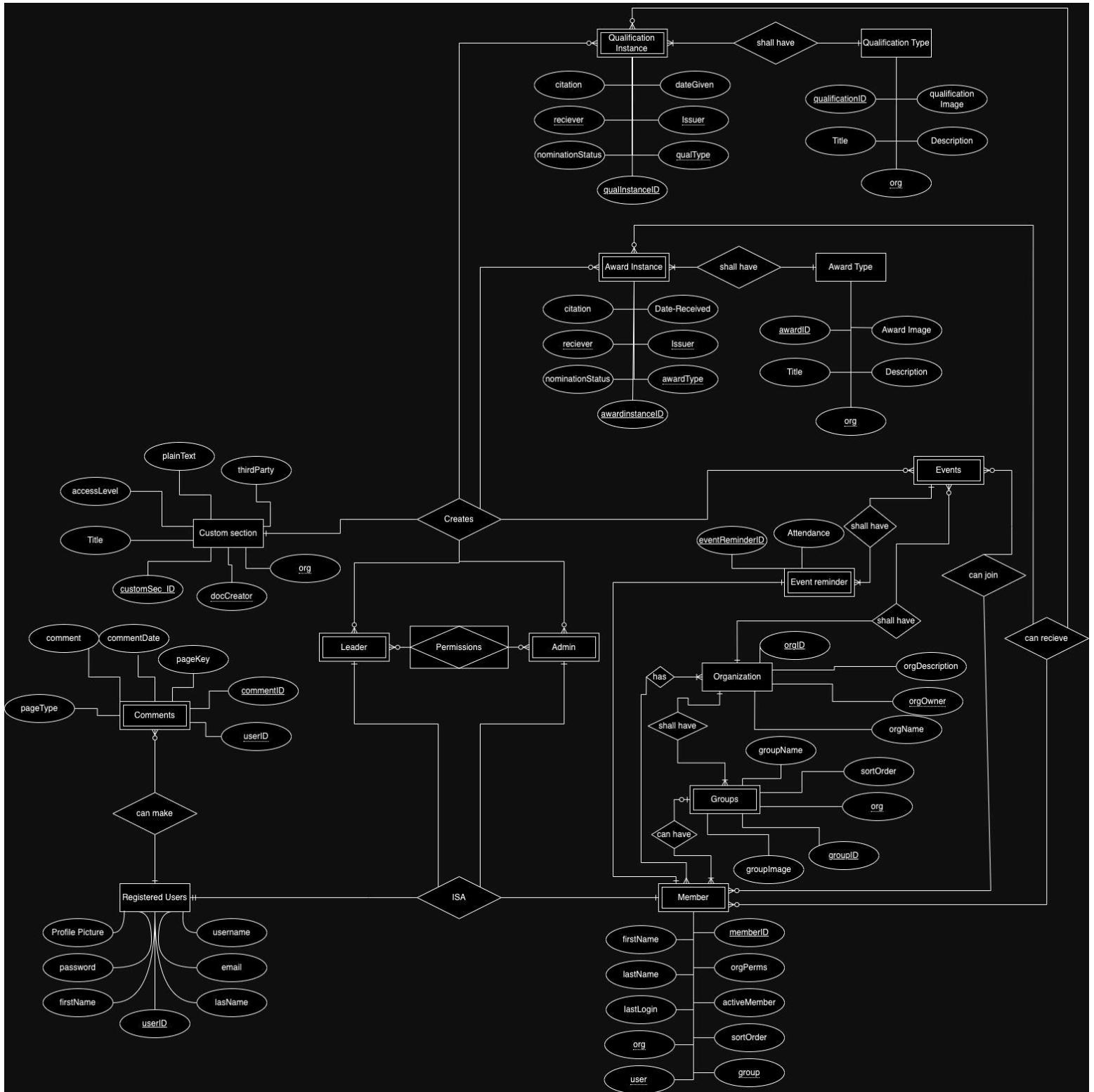
11. Event reminder

- 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).

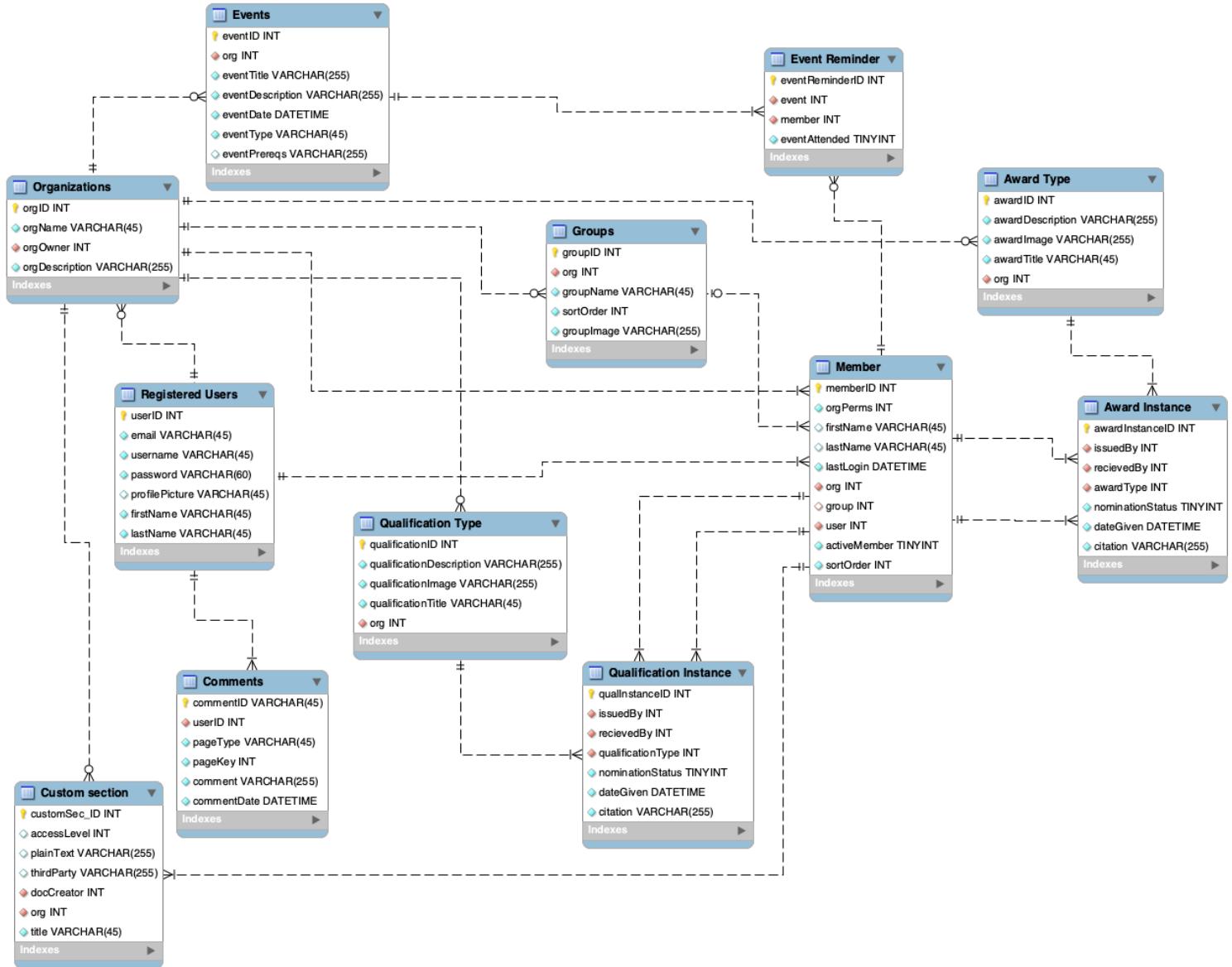
12. Groups

- 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

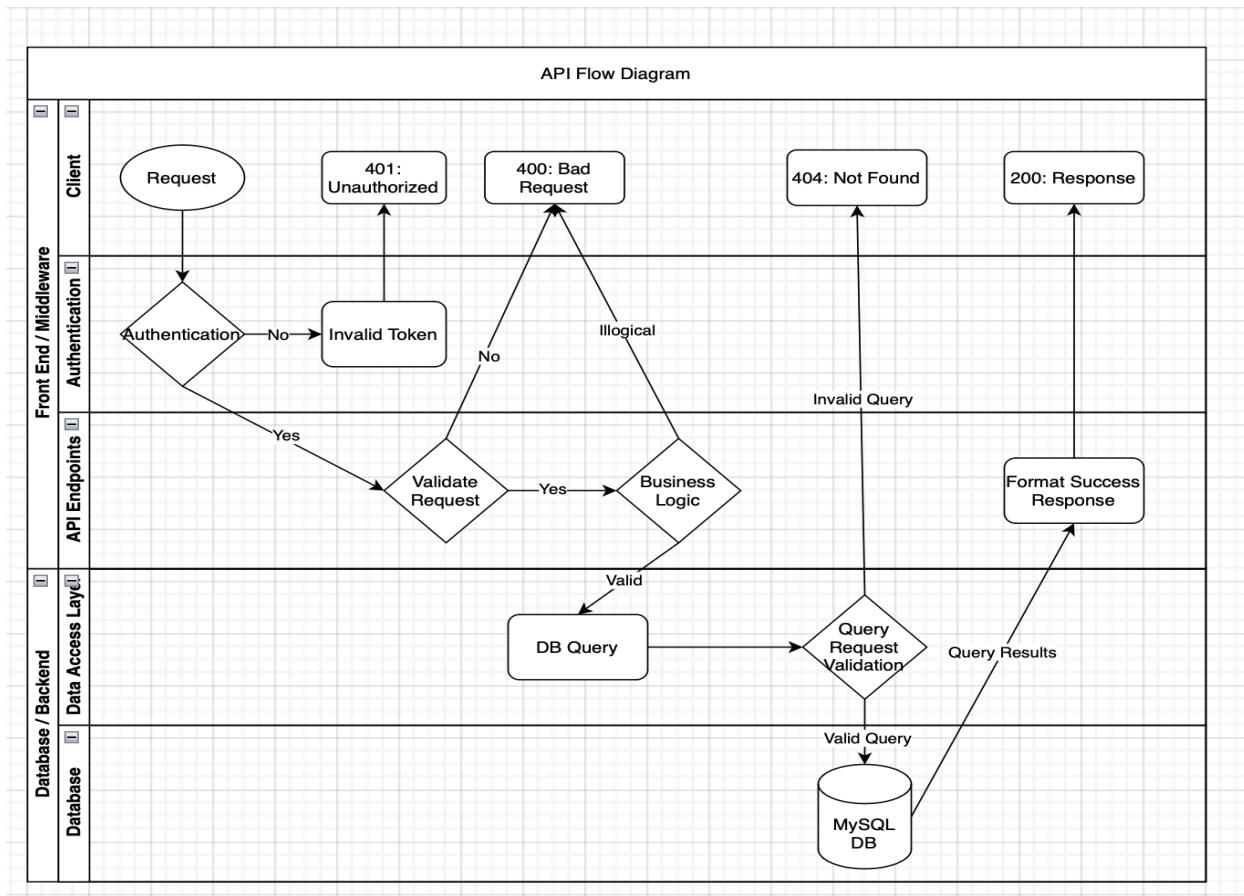
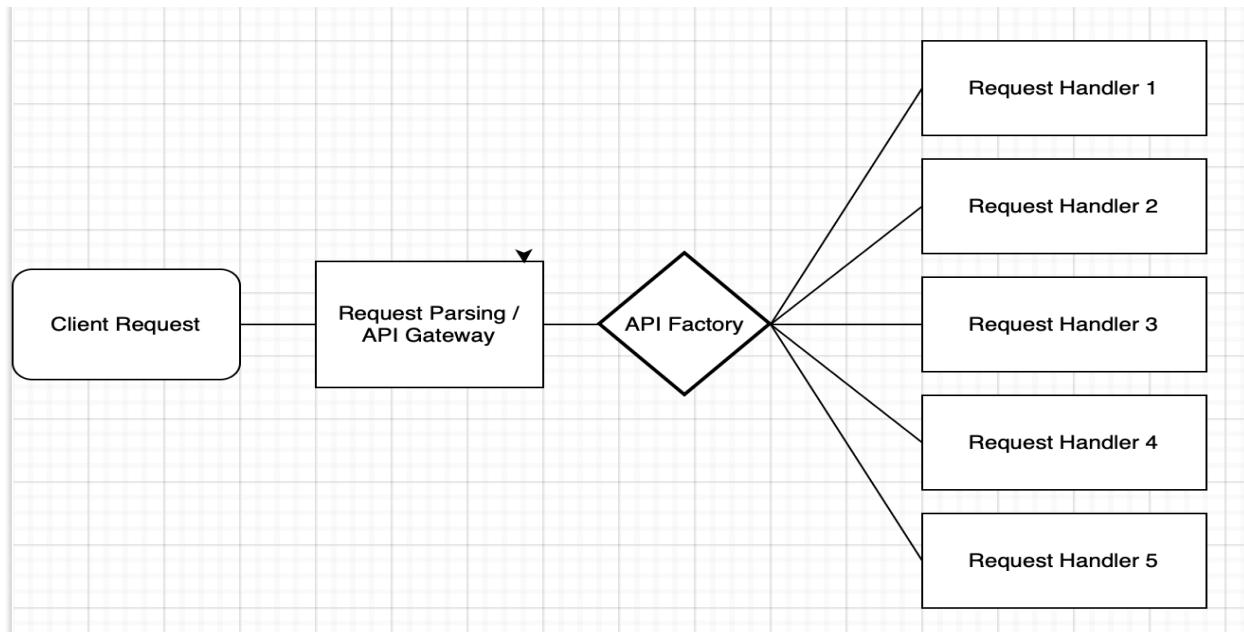
API HIGH LEVEL ARCHITECTURE

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.

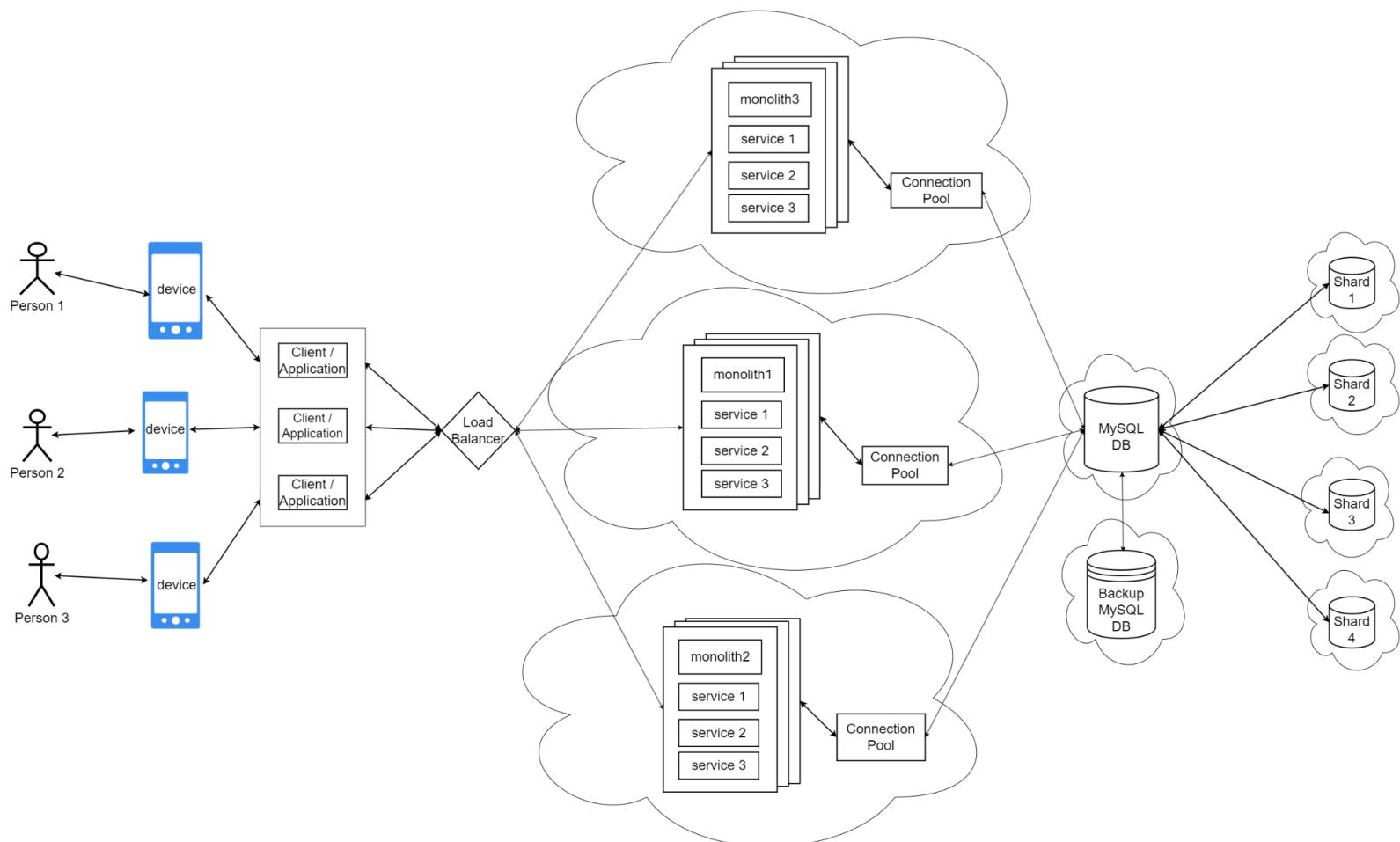
Users will interact with the platform through a React application deployed on EC2. This React front-end will communicate with a back-end server implemented using Express and TypeScript, also hosted on EC2. This setup ensures seamless interaction and maintains cross-language compatibility between the 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.

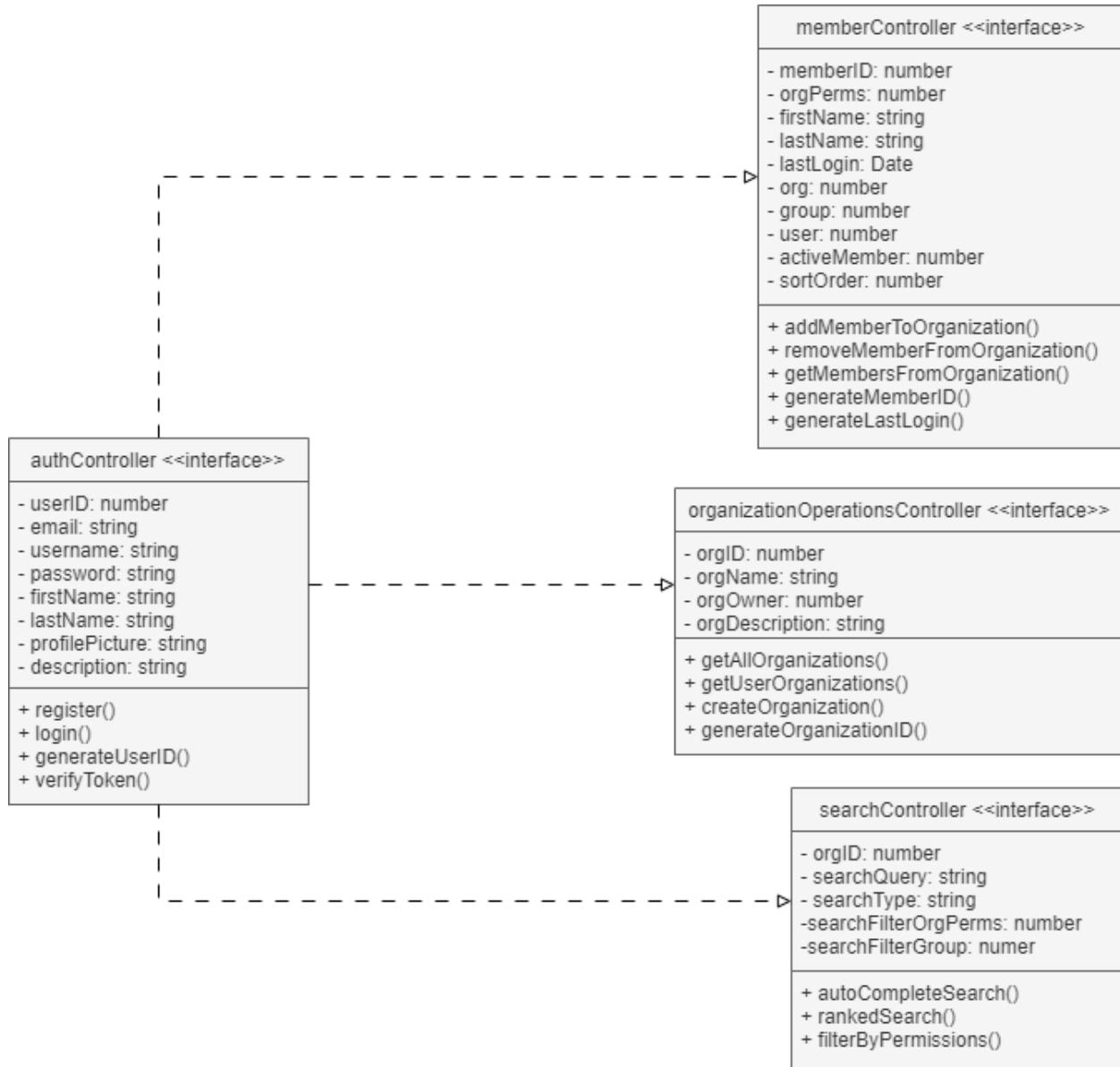
Database operations will be optimized using a connection pool, reusing connections for future database queries to reduce redundancy and manage resources efficiently. As the service grows, the database will be sharded, partitioning large datasets across multiple database nodes. This sharding strategy 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.

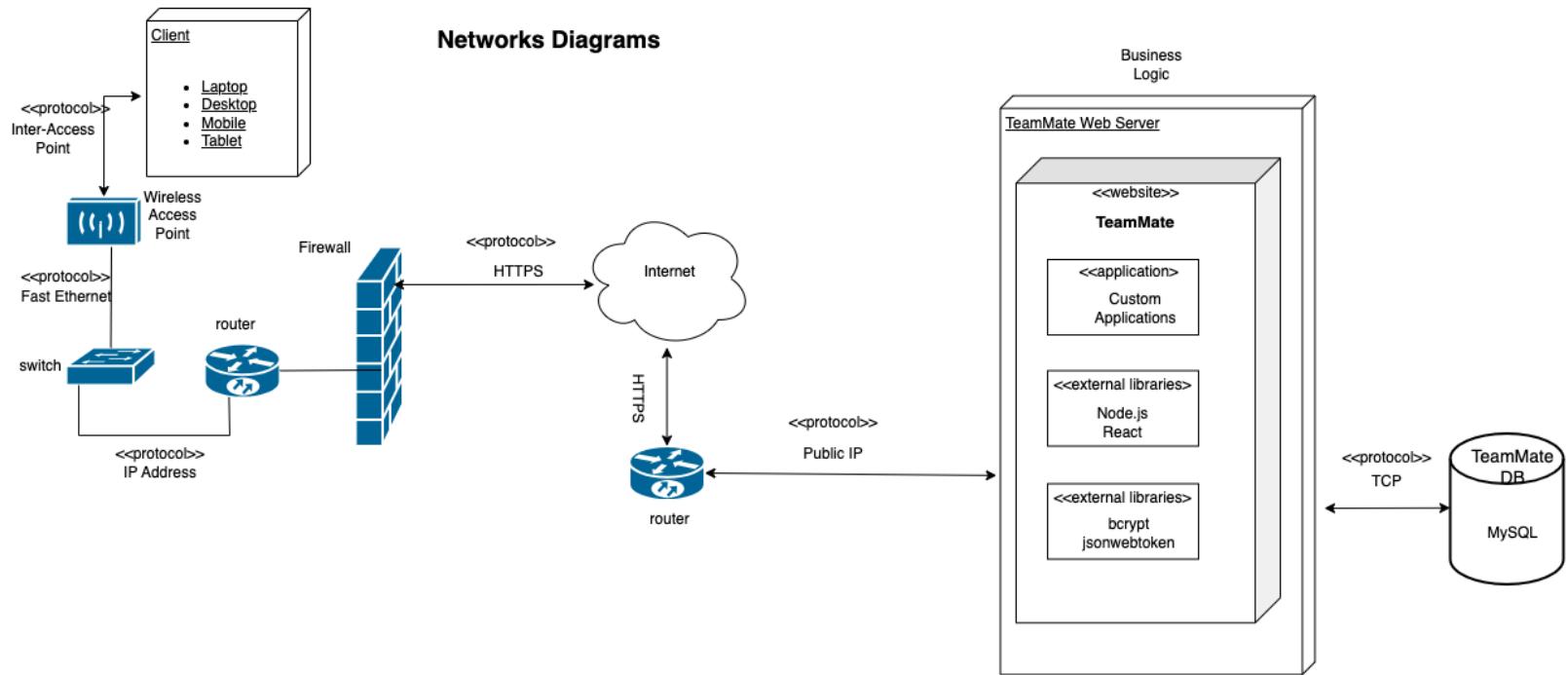
By integrating horizontal and vertical scaling methods, utilizing robust database management, and implementing message queuing for efficient communication, we can ensure that our monolithic architecture remains scalable, cost-effective, and maintainable. This design leverages AWS's capabilities to provide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



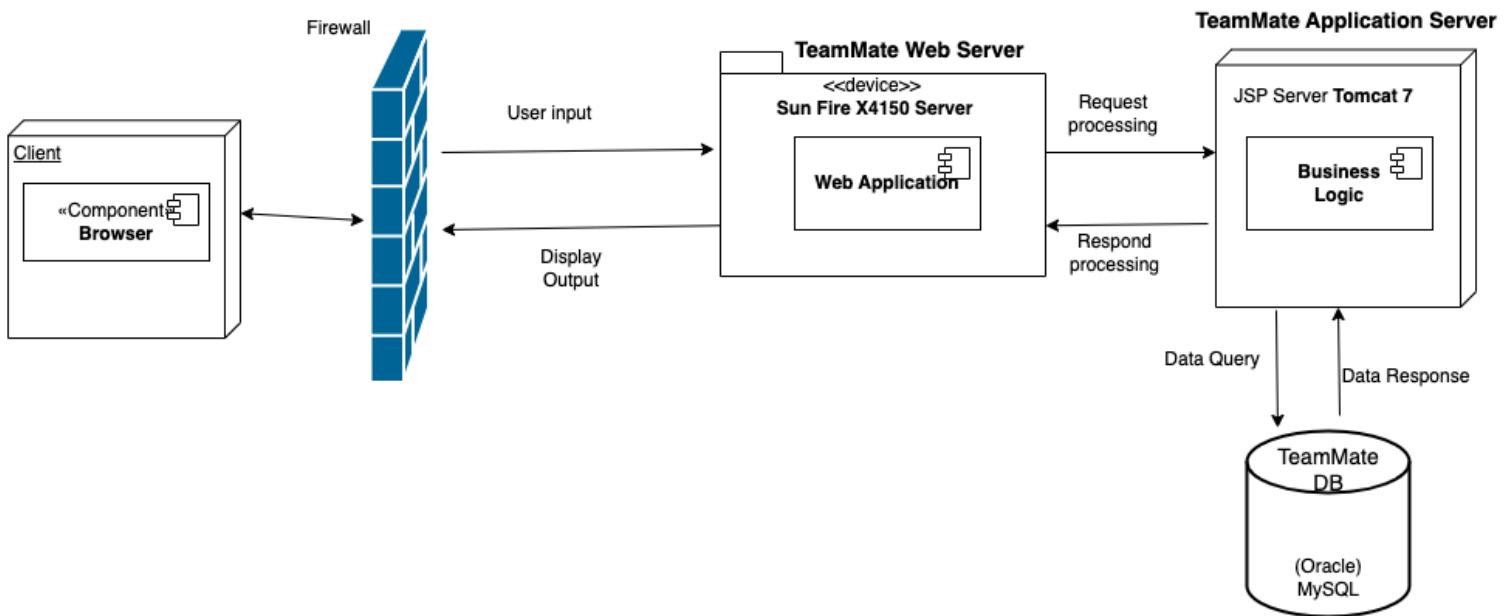
We are primarily using interfaces rather than classes. The interactions between the interfaces and the database tables are identical, and the database tables are well described in the EER (Enhanced Entity-Relationship) diagram. Below are the UML diagrams for our controllers and interfaces:



High Level Application Network and Deployment Design



Deployment Diagram



Horizontal Prototype Feedback

“On the Home Page, the three images look like buttons that the User may want to click.”
Restyled these elements to not look like buttons.

“Maybe we can add a Free Demo that people can try without registering/logging in.”
Free Demo added as a low-priority task to be tackled after completing the app.

“Add Back button to Sign-in and Register pages.”
Added a back button to the corresponding pages.

“Show password requirements on the registration page before they make a mistake.”
Added requirements as “placeholder” value in textbox.

“On the Create Organization Form, replace select field that have options 1, 2, 3.”
Removed this field from the form.

“Dashboard needs to navigate to a newly created org rather than asking to refresh.”
Fixed routing to ensure the user is navigated to their org when they create it.

“Remove WIP Buttons where not required.”
Removed Implement Later wherever those features have been completed.

“Revisit website color scheme.”
Restyled the overall website theme and added shades of purple.

“Search Bar can be longer.”
Search bar extended to take up more room.

“Group item number of people and max limit is confusing.”
Replaced vague numbers with more descriptive text.

“Don't stretch the placeholder images.”
Replaced with different images that are appropriately sized where able.

Team Contributions

- **Tharun Krishna**
 - Documentation revisions.
 - General front-end development.
 - Front-end styling in-line with Figma wireframes.
 - GitHub supervision.
 - AWS synchronization.
- **Tyron Cheung: Contributions Score 10**
 - Created all figma wireframes and prototypes.
 - Contributed to front-end development and design.
 - Oversaw other front-end developers progress.
 - Refined static pages.
- **Randolf Gabrielle Uy: Contributions Score 10**
 - Leading efforts on Backend of the Prototype.
 - Oversaw other back-end developers progress.
 - Mentored other team members.
 - Created Organization Creation Backend.
 - Created Member Addition Backend.
- **Carlos Hernandez: Contributions Score 10**
 - Refined Entity Relationship Diagram (ERD).
 - Refined Entity Establishment Relationship (EER).
 - Assisted Backend team.
 - Created SQL Queries for backend integration.
 - Interfacing front-end components with backend-controllers.

- **Xuefeng Guan (Tony): Contributions Score 7**
 - Contributed to tasks/projects pages.
 - Contributed to static pages.
 - Contributed to events pages.
 - Contributed to sidebar.
- **Naw San: Contributions Score 8**
 - Created Initial Recognition Page.
 - Created Initial Header and Popups.
 - Contributed to peripheral static pages.
 - Created Footer.
 - Refined Footer.
- **Yasmine Sucu: Contributions Score 7**
 - Contributed to backend APIs.
 - Contributed to Event Controller.
 - Contributed to static pages.
- **Jannelly Hernandez Umanzor: Contributions Score 6**
 - Contributed to backend APIs.
 - Contributed to Event Controller.
 - Contributed to static pages.

SW Engineering CSC648-848-05 Summer 2024

TeamMate Personnel Management Team 05 - Concat

Name	E-Mail	Role
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 4
July 30 2024**

Version History

Revision	Date Submitted
M1	13 June 2024
M1V1	20 June 2024
M1V2	09 July 2024
M2V1	09 July 2024
M2V2	23 July 2024
M3V1	23 July 2024
M3V2	30 July 2024
M4V1	30 July 2024

Table of Contents

<u>Product Summary</u>	3
<u>Committed Functions</u>	4
<u>Application</u>	5
<u>Usability Test Plan</u>	6
<u>Use/Test Case 1 (Organization Creation)</u>	6
<u>Use/Test Case 2 (Group Creation)</u>	6
<u>Use/Test Case 3 (Award Creation)</u>	7
<u>Use/Test Case 4 (Event Creation)</u>	7
<u>Use/Test Case 5 (Member Operations)</u>	8
<u>Test Table</u>	8
<u>Questionnaire</u>	9
<u>Responses</u>	9
<u>QA Test Plan</u>	17
<u>QA Test 1 (Password Security)</u>	17
<u>QA Test 2 (Member Addition)</u>	18
<u>QA Test 3 (Search Bar)</u>	19
<u>QA Test 4 (Error Logging)</u>	20
<u>QA Test 5 (Concurrent Users)</u>	21
<u>Code Review</u>	22
<u>Internal Code Review</u>	22
<u>External Code Review</u>	30
<u>Security Self-Check</u>	31
<u>Non-Functional Specs Self-Check</u>	34
<u>Additional Comments</u>	37
<u>Changes to the DBMS</u>	37
<u>Team Contributions</u>	38

Product Summary

TeamMate is a comprehensive personnel management solution designed for small to mid-sized groups, such as non-profits and community organizations. It consolidates essential management functions into a single, user-friendly platform, enabling easy registration, member tracking, event management, and award issuance.

TeamMate's standout and superior feature is its Automated Recognition and Progress Tracking System. This system automatically tracks member participation, awards achievements, and updates progress, providing timely recognition and broad overviews of individual and group accomplishments. This holistic view is valuable for performance reviews, planning, setting future goals along with motivating the members of the organization.

Discord focuses on communication and lacks recognition and structured progress tracking, **Airtable** and **Google sheets** require extensive manual setup to achieve similar functionality, whereas **TeamMate** offers an out-of-the-box solution that simplifies administrative tasks. **PERSCOM** provides good tracking but demands more manual effort for recognition, while **Rippling HCM** caters to larger organizations with complex and costly systems, making it less suitable for smaller groups.

Committed Functions

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 create a new organization.
- Accounts shall join an existing organization.
- Accounts shall view all organizations they have 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.
- 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 a roster page that shows all members in the organization separated by groups.
- 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.
- Admins shall directly assign awards.
- Award instances shall have a date received.
- Award instances shall state the issuing member.

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.
- 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.
- Admins shall assign members to an existing event.
- Group leaders shall assign members to an existing event.
- Members shall be able to join an event.
- Events shall have a title.
- Events shall have a description.
- Events shall have a start datetime.

Application

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/>

Usability Test Plan

Use/Test Case 1 (Organization Creation)

Objective:

Create an organization. While conducting this test, the user will also go through the registration and login process. This feature is being tested to determine if the user is intuitively able to navigate the home page and sidebar of the dashboard.

Description:

The horizontal prototype already supports these features, the user will start from the website's home page. The intended user is a first time user of the website, with no prior context as to what it is. We are measuring the time and mouse movements of the user, to see how long it takes the user to complete the test, and where all they navigate while completing the test, in order to determine if the website design makes intuitive sense to a first time user.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/>

Use/Test Case 2 (Group Creation)

Objective:

Create a group. This feature is being tested to determine if the user is intuitively able to navigate the header and corresponding pages intuitively.

Description:

The beta prototype will be supporting this feature, the user will start from the dashboard after logging in. The intended user is a new user that has just created an organization. We are measuring the time taken and mouse movements of the user to determine if they can quickly navigate to the required pages intuitively, to validate our dashboard design.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/organizationDashboard/Testing%20Org>

Use/Test Case 3 (Award Creation)

Objective:

Create an award. This feature is being tested to determine if the user is able to find the awards page and create an award.

Description:

The beta prototype will be supporting this feature, the user will start from the dashboard after logging in. The intended user is a relatively new user that has created an organization. We are measuring the time taken and mouse movements & page navigation to determine if they are able to figure out that awards are part of the recognitions tab, and can use the creation form to make an award instead of certification.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/organizationDashboard/Testing%20Org>

Use/Test Case 4 (Event Creation)

Objective:

Create an event. This feature is being tested to determine if the user is able to navigate to the events page and follow through the event creation form.

Description:

The beta prototype will be supporting this feature, the user will start from the dashboard after logging in. The intended user is a relatively new user that has created an organization. We are measuring the time taken and mouse movements to determine if they are able to intuitively navigate from the dashboard to the event page and the events creation form and complete it.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/organizationDashboard/Testing%20Org>

Use/Test Case 5 (Member Operations)

Objective:

Add a member. This feature is being tested to determine if the user can figure out how to add members to their organization and assign them attributes.

Description:

The beta prototype will be supporting this feature, the user will start from the dashboard after having created a few attributes. The intended user is a user that has explored the dashboard for some time and created various attributes. We are measuring the time taken and mouse movements to determine if they are able to intuitively navigate to the members page and invite another user to their organization.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/organizationDashboard/Testing%20Org>

Test Table

Test/Use Case	% Completed	Efficiency	Errors	Comments
Org Creation	100%	4-5 Minutes	None	Older users unable to figure out that the create bar on the sidebar is how you make an org. Inflated efficiency due to users coming up with registration info.
Group Creation	100%	<1 Minute	None	Creation complete.
Award Creation	100%	<1 Minute	None	Creation complete.
Event Creation	100%	<1 Minute	None	Creation complete.
Member Add	100%	1-2 Minutes	None	It's not clear how to find a user's ID to add into the org. Can also add duplicate members to org.

Questionnaire

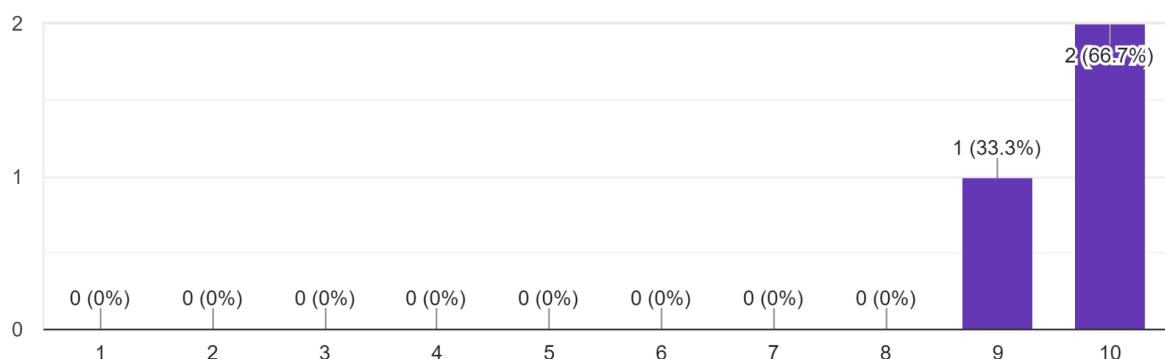
1. I found it easy to navigate from the home page to the dashboard.
2. I found it easy to create an account and login.
3. I was able to intuitively navigate the dashboard to create an organization.
4. The dashboard header was difficult to understand.
5. The group's page layout confused me.
6. I was able to intuitively navigate the dashboard to create a group.
7. I struggled to find the awards page.
8. The layout of the awards page was consistent with the other pages.
9. I was able to intuitively navigate the dashboard to create an award.
10. The event page's layout was consistent with the rest of the application.
11. The event details page layout contained all the information I wanted it to.
12. I was able to intuitively navigate the dashboard to create an event.
13. The layout of the members page made intuitive sense.
14. Using the add button in the members page was confusing.
15. I was able to intuitively navigate the dashboard to add a member.

<https://forms.gle/Xq8RFURWbQ7Rh4gp9>

Responses

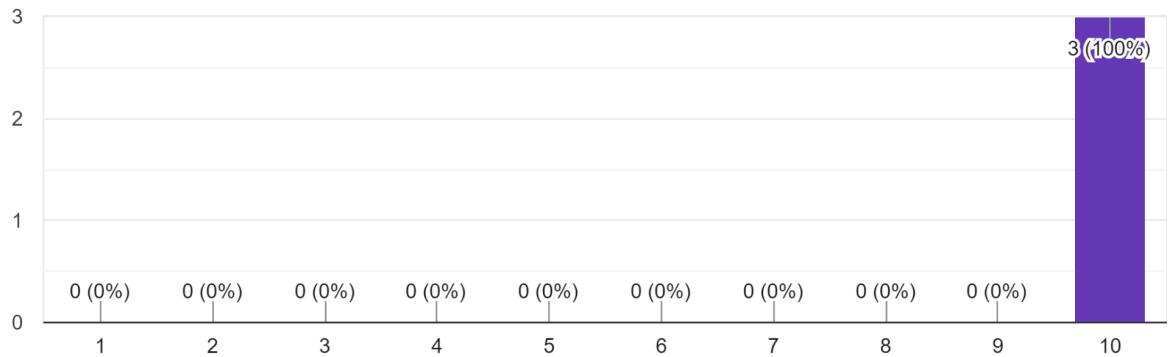
I found it easy to navigate from the home page to the dashboard.

3 responses



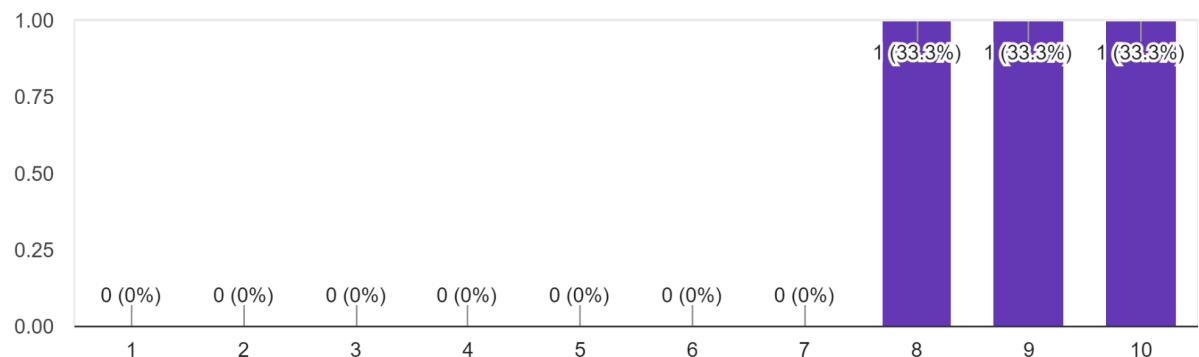
I found it easy to create an account and login.

3 responses



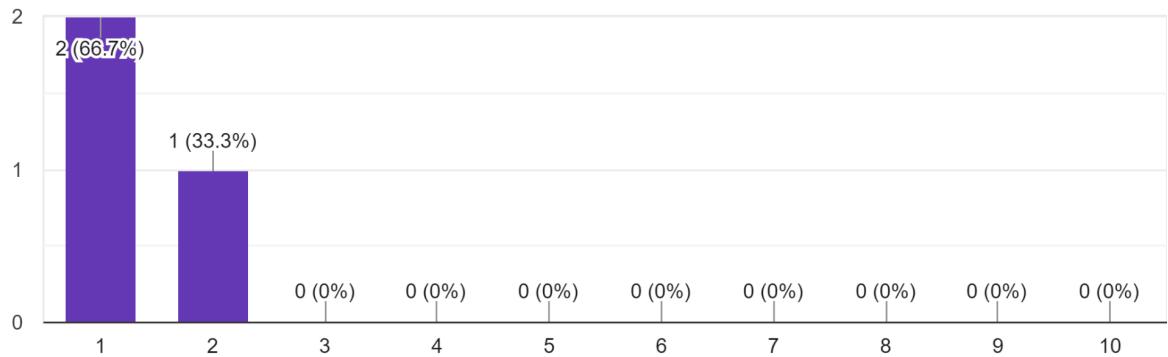
I was able to intuitively navigate the dashboard to create an organization.

3 responses



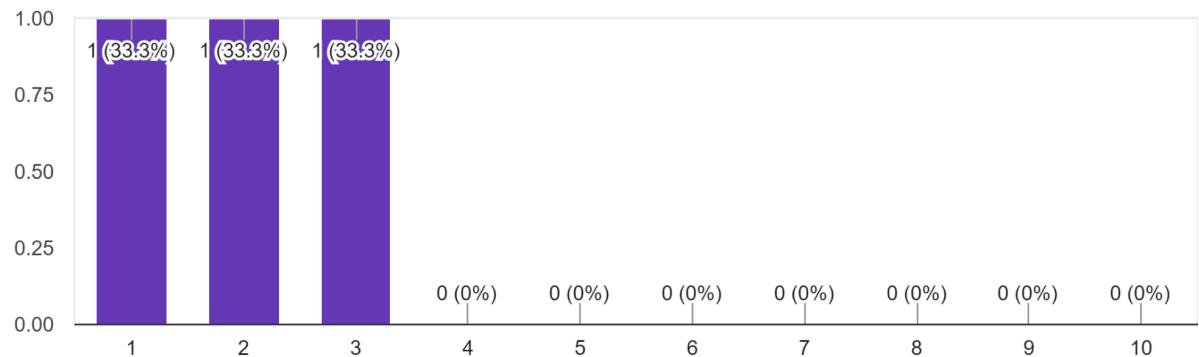
The dashboard header was difficult to understand.

3 responses



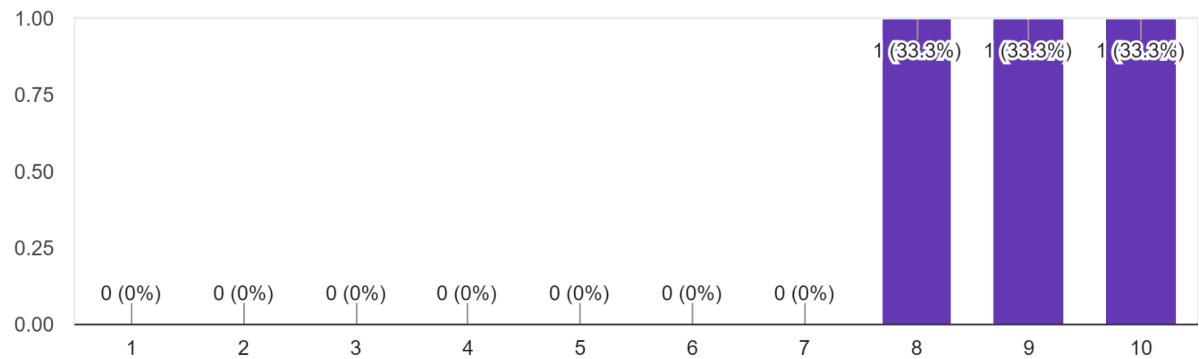
The group's page layout confused me.

3 responses



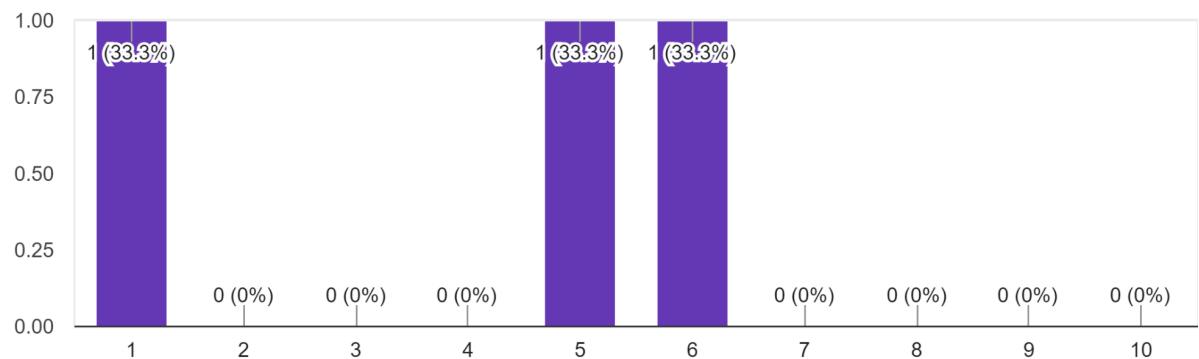
I was able to intuitively navigate the dashboard to create a group.

3 responses



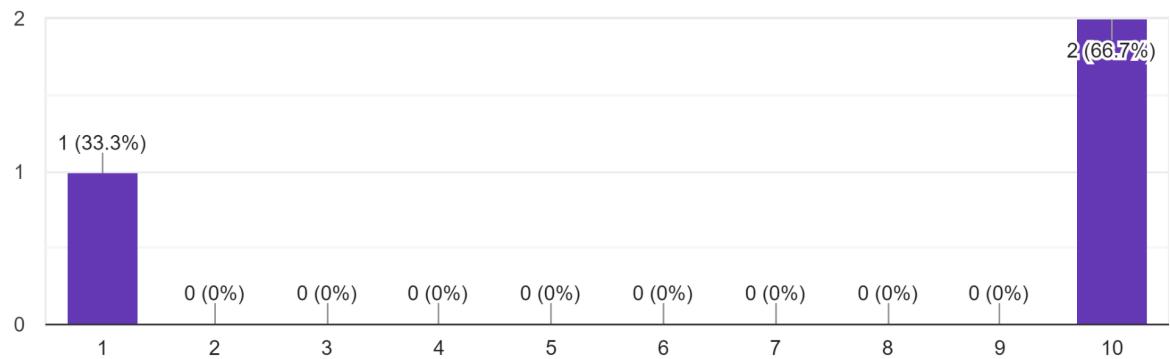
I struggled to find the awards page.

3 responses



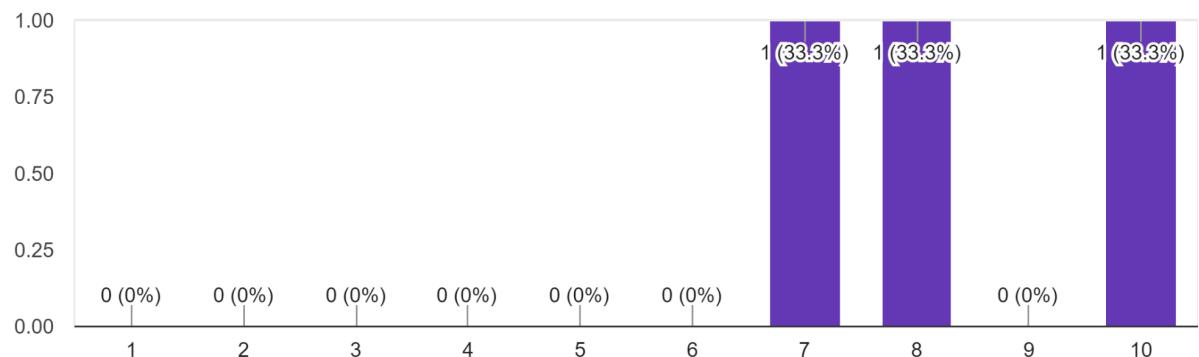
The layout of the awards page was consistent with the other pages.

3 responses



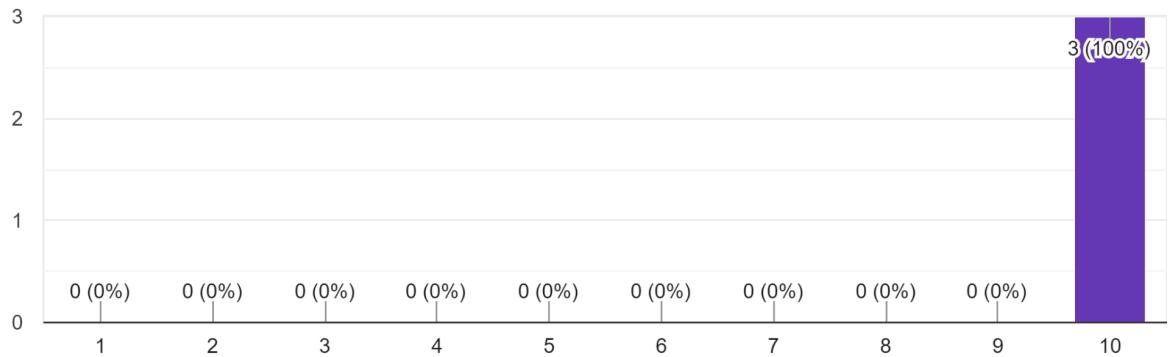
I was able to intuitively navigate the dashboard to create an award.

3 responses



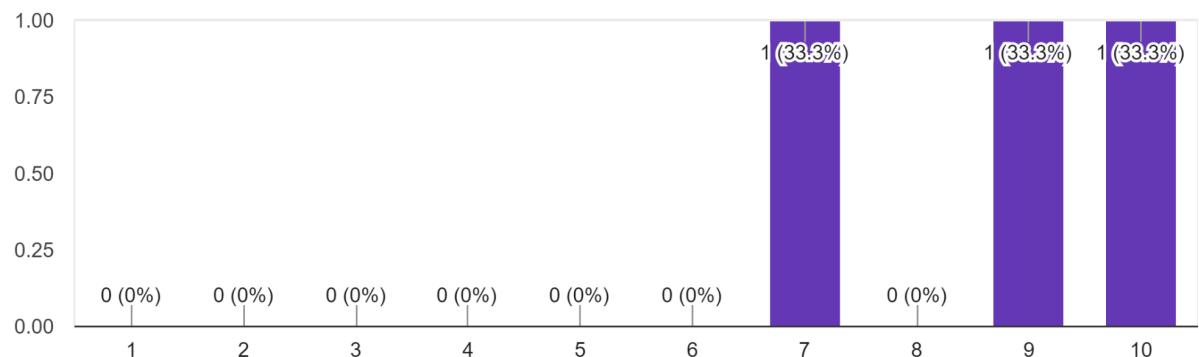
The event page's layout was consistent with the rest of the application.

3 responses



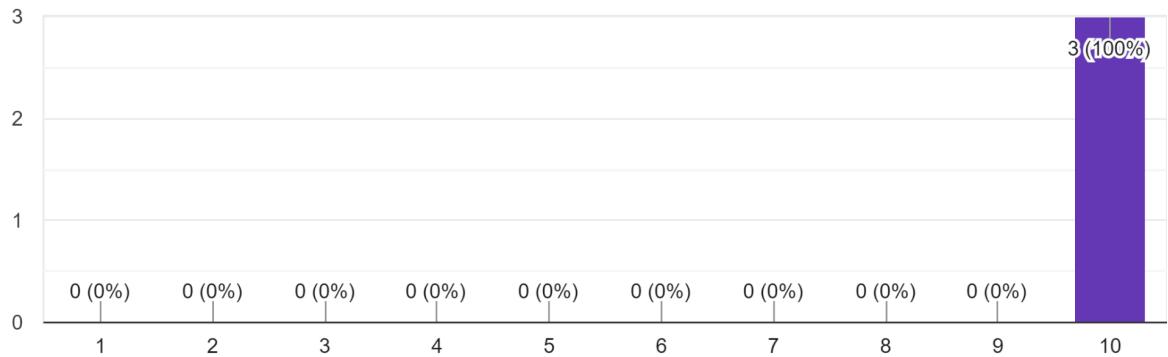
The event details page layout contained all the information I wanted it to.

3 responses



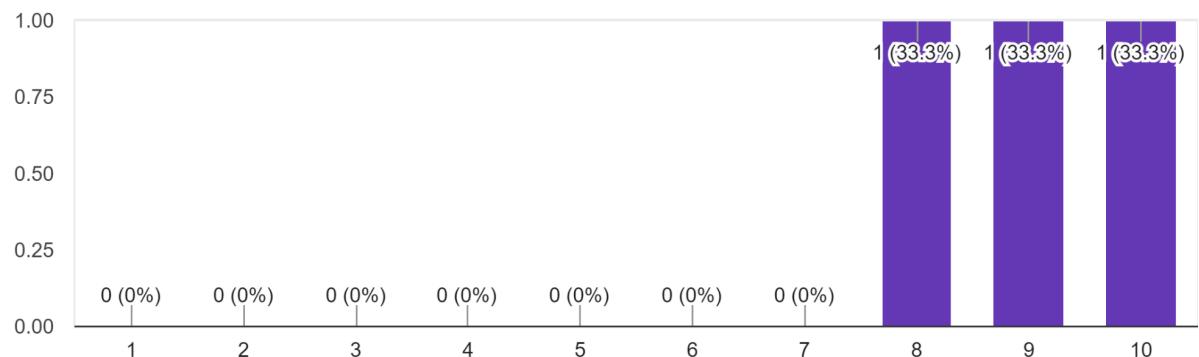
I was able to intuitively navigate the dashboard to create an event.

3 responses



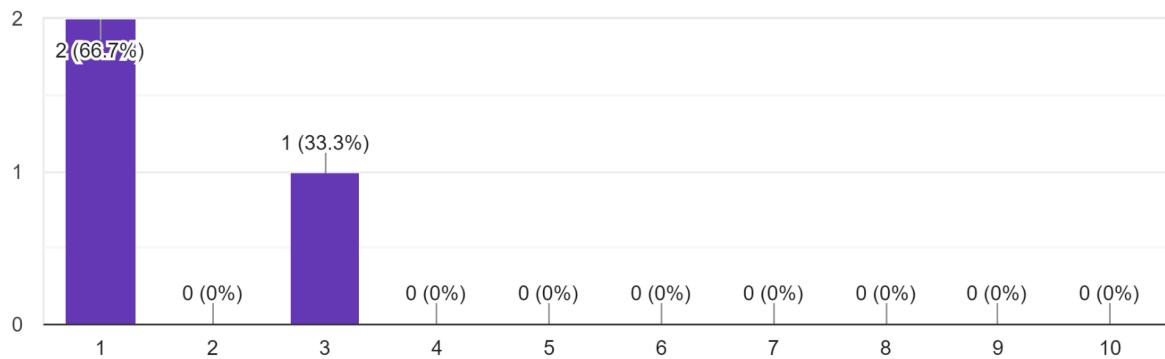
The layout of the members page made intuitive sense.

3 responses



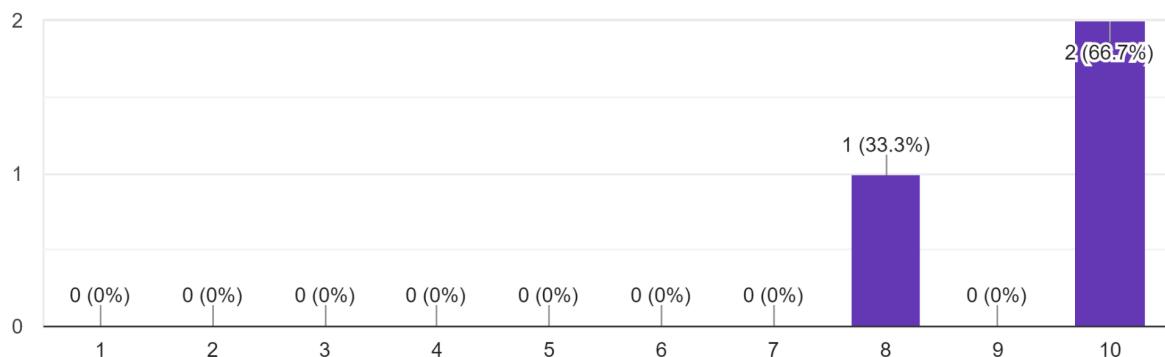
Using the add button in the members page was confusing.

3 responses



I was able to intuitively navigate the dashboard to add a member.

3 responses



QA Test Plan

QA Test 1 (Password Security)

Objective:

Create an account with a variety of passwords, valid and invalid. Ensure that password validation during registration is following modern security standards.

Plan:

The horizontal prototype already supports these features, the user will start from the website's registration page. The three tests will be to try a password that's too short, a password that is long enough, but doesn't have special characters, and a valid/secure password.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/register>

Number	Description	Test Input	Expected Output	Pass/Fail
1	Short Password	“Test”	Password must have 8 characters	Pass
2	Weak Password	“Password123”	Password must contain special characters	Pass
3	Strong Password	“Password@123”	No errors	Pass

QA Test 2 (Member Addition)

Objective:

Add three members to your organization using their Member IDs. Ensure that member addition is following good usability standards and error handling.

Plan:

The horizontal prototype already supports these features, the user will start from the website's dashboard after logging in and creating an organization. The three tests will be to add a member that doesn't exist, a member that does exist, and a member that is already in the organization..

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/Members>

Number	Description	Test Input	Expected Output	Pass/Fail
1	Invalid Member	1234	Member does not exist.	Pass
2	Valid Member	2017	Member added to list.	Pass
3	Existing Member	9141	Member is already in the organization.	Fail

QA Test 3 (Search Bar)

Objective:

Search for three members from the members page. Ensure the search bar is following along with our expected load standards and requirements.

Plan:

The horizontal prototype already supports this feature, the user will start from a populated members page. The three tests will be to search for an existing member, searching for a member that doesn't exist, and a blank search bar.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/Members>

Number	Description	Test Input	Expected Output	Pass/Fail
1	Valid Search	“John Doe”	Member list with one item	Fail
2	Invalid Search	“TestUser”	Empty member list	Fail
3	Empty Search	“ “	Unfiltered member list	Fail

QA Test 4 (Error Logging)

Objective:

Check if your actions are being logged in the console. Ensure the browser console is populating with appropriate messages in-line with our fault tolerance standards and requirements.

Plan:

The horizontal prototype already supports this feature, the user will start from an empty members page. The three tests will be to try to add a member that is valid, a member that does not exist, and the user themselves. The browser's console will be monitored during the test to check for results.

<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/Members>

Number	Description	Test Input	Expected Output	Pass/Fail
1	Valid Member	2017	“Member added successfully”	Pass
2	Invalid Member	1234	“Member not found”	Pass
3	Add Self	9141	“Member already in organization”	Fail

QA Test 5 (Concurrent Users)

Objective:

Create an account and organization while the application is under load. Ensure the application can support many concurrent users without significant degradation in performance in-line with our performance standards and requirements.

Plan:

The horizontal prototype already supports the features we will be stress testing. We will be using Postman to simulate network traffic. The user will start from the homepage. The user will try registering, logging in, and creating an organization while the database and backend are under stress.

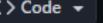
<http://ec2-44-223-28-76.compute-1.amazonaws.com:3000/>

Number	Description	Test Input	Expected Output	Pass/Fail
1	Registration	New Account	Minimal Delay	Pass
2	Login	Existing Account	Minimal Delay	Pass
3	Create Organization	New Organization	Minimal Delay	Pass

Code Review

Internal Code Review

modified file eventController and modified the #20

Edit 

 Open JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannelly](#) 

Conversation 19 Commits 10 Checks 0 Files changed 4 +119 -13

JannellyHU9301 commented 2 weeks ago

createEventReminderMember file
Got cookies to verify authentication
createEventReminderMember,
Generate event reminder ID
Create query and throw error if query fails

modified file eventQueriesfile and added the
function getEventQuery

KaiserMighty and others added 2 commits 3 weeks ago

- Merge pull request [sfsu-joseo#18](#) from sfsu-joseo/development cf767f0
- modified file eventController and modified the e28e337

JannellyHU9301 requested a review from randolfuy01 2 weeks ago

randolfuy01 reviewed 2 weeks ago

View reviewed changes

application/backEnd/backend-app/src/controllers/eventController.ts Outdated

```
2      2      import { getCookies, verifyToken } from "./authController";
3      3
4      4      import { Request, Response } from "express";
5      5      import { verify } from "jsonwebtoken";
6      6      import { getMemberByUserID } from "../database_queries/memberQueries";
7      - // interface Event {
```

Reviewers

- 001CarlosH
- randolfuy01

Still in progress? [Convert to draft](#)

Assignees

No one—[assign yourself](#)

Labels

None yet

Projects

None yet

Milestone

No milestone

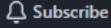
Development

Successfully merging this pull request may close these issues.

None yet

Notifications

Customize

 Subscribe

You're not receiving notifications from this thread.

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 

```
2      2      import { getCookies, verifyToken } from "./authController";
3      3
4      4      import { Request, Response } from "express";
5      5      import { verify } from "jsonwebtoken";
6      6      import { getMemberByUserID } from "../database_queries/memberQueries";
7      - // interface Event {
7      + import { getEventQuery } from "../database_queries/eventQueries";
```



randolfuy01 2 weeks ago

getEventQuery should be grouped with the other internal imports like Event, EventReminder, not with verify, request, response



Reply...

[Resolve conversation](#)

Notifications

Customize

 [Subscribe](#)

You're not receiving notifications from this thread.

4 participants



 Maintainers are allowed to edit this pull request.

 [Lock conversation](#)



randolfuy01 reviewed 2 weeks ago

[View reviewed changes](#)

```
application/backEnd/backend-app/src/controllers/eventController.ts
132 +      try {
133 +          let newEvent = await getEventQuery(eventID);
134 +      } catch (Error) {
135 +          return res.status(500).json({
```



randolfuy01 2 weeks ago

Rather than a query error, if we cannot find the event. This should be an error that the event does not exist



Reply...

[Resolve conversation](#)

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 

[Resolve conversation](#)



 **randolfuy01** reviewed 2 weeks ago

[View reviewed changes](#)

randolfuy01 left a comment

...

Looks good so far, only a few minor issues.



 **randolfuy01** reviewed 2 weeks ago

[View reviewed changes](#)

application/backEnd/backend-app/src/controllers/eventController.ts

```
136 +         message: "Error querying event"
137 +     })
138 +   }
139 + }
```



randolfuy01 2 weeks ago

...

now that we verify that the event exists, make sure to create another query to add that eventReminder to the database.



JannellyHU9301 2 weeks ago

[Author](#)

...

Gabe,

Can you please clarify regarding the query to add eventReminder to the database:

Which database table will this record be written?

What attributes will be written to the database?



randolfuy01 2 weeks ago

...

there should be an eventReminder, reference the table diagram inside milestone 2. If a user is creating an event reminder, presumably you would want to write all attributes to that table

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 

there should be an eventReminder, reference the table diagram inside milestone 2. If a user is creating an event reminder, presumably you would want to write all attributes to that table into a single row



randolfuy01 2 weeks ago

...

there isn't an actual query for it yet, you can create one. It would be similar to createMember or createOrganization just with different fields but the syntax should be relatively similar



JannellyHU9301 2 weeks ago

Author

...

By 'Table Diagram' do you mean the EER (Enhanced Entity-Relationship) diagram in the Milestone 2 document?



 Reply...

 Resolve conversation

 in the file of eventController I modified and 

3aef8bf

  **JannellyHU9301** requested a review from **randolfuy01** last week



JannellyHU9301 commented last week

Author

...

I modified in the eventQueries file I added insertEventReminderQuery

In the eventController I created the function generateEventReminderID() and created query to check if the event exists and

throw an error if event does not exist in the database



 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3](#) Jannelly...



randolfuy01 reviewed last week

[View reviewed changes](#)

application/backEnd/backend-app/src/controllers/eventController.ts

```
1 + import { Event, EventReminder, Member, Cookie } from "./models";
2 + import { getEventQuery } from "../database_queries/eventQueries";
3 + import { insertEventReminderQuery } from "../database_queries/eventQueries";
4 +
5     import { getCookies, verifyToken } from "./authController";
```



randolfuy01 last week

...

This needs to be grouped with the other internal imports



JannellyHU9301 last week

Author

...

Gabe, I am confused about which line needs to be grouped with the other internal imports?



randolfuy01 last week

...

Do u see how getCookies and verifyToken have 1 line in between them and the insertEventReminderQuery? The getCookies and verifyToken should be grouped with them.



Reply...

[Resolve conversation](#)



randolfuy01 reviewed last week

[View reviewed changes](#)

application/backEnd/backend-app/src/controllers/eventController.ts

Outdated

Show resolved

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 

ts

Outdated 



randolfuy01 reviewed last week

[View reviewed changes](#)

application/backEnd/backend-app/src/database_queries/eventQueries.ts

```
40 + // This query inserts an event reminder into the EventReminder table in the data
41 + async function insertEventReminderQuery(eventReminder: EventReminder): Promise<...
42 +     return new Promise((resolve, reject) => {
```



randolfuy01 last week

Have Carlos review the database logic behind this



JannellyHU9301 last week

Author

...

[@001CarlosH](#) Gabe said if you can review the logic behind this function



001CarlosH last week

...

the query for the get events is correct



001CarlosH last week

...

also make sure to have the db set up on your own system, you need to be capable of testing your own functions you create. its difficult for me to run it because i have 3-classes and personal responsibilites.



Reply...

[Resolve conversation](#)



randolfuy01 reviewed last week

[View reviewed changes](#)

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 



 **randolfuy01** reviewed last week

[View reviewed changes](#)

randolfuy01 left a comment

...

Looks promising, needs to be tested and working and updated a little more



 **JannellyHU9301** and others added 2 commits [last week](#)

-  Fixed the imports in file eventController ... baf8665
-  Merge branch 'sfsu-joseo:master' into back_end_branch_m3_Jannelly b4b8a1a



JannellyHU9301 commented last week

Author ...

Fixed the imports in EventController file

Added triple equal sign, to check for explicit equivalence in lines 116-118 of the eventController file

Asked Carlos to check the database logic behind the query for the get events and he said it was correct



  **JannellyHU9301** requested a review from **randolfuy01** last week

 **JannellyHU9301** added 5 commits [last week](#)

-  Added the eventRoutes file to post the event ... da0b009
-  Updated files b935930
-  Merge branch 'back_end_branch_m3_Jannelly' of <https://github.com/Jannelly> ... 3158553
-  Added the event routes file ... 3dbb64b

 Open

modified file eventController and modified the #20

JannellyHU9301 wants to merge 10 commits into [sfsu-joseo:back_end_branch_m3](#) from [JannellyHU9301:back_end_branch_m3_Jannel...](#) 

JannellyHU9301 commented last week

Author

Fixed the imports in EventController file

Added triple equal sign, to check for explicit equivalence in lines 116-118 Of the eventController file

Asked Carlos to check the database logic behind the query for the get events and he said it was correct



 JannellyHU9301 requested a review from **randolfuy01** last week

 JannellyHU9301 added 5 commits last week

-  Added the eventRoutes file to post the event  da0b009
-  Updated files b935930
-  Merge branch 'back_end_branch_m3_Jannelly' of <https://github.com/Jann...>  3158553
-  Added the event routes file  3dbb64b
-  Uptaded insertEventReminderQuery function in the  b279278

Add more commits by pushing to the [back_end_branch_m3_Jannelly](#) branch on [JannellyHU9301/csc648-848-05-sw-engineering-su24-T05_2](#).

Review requested

[Show all reviewers](#)

Review has been requested on this pull request. It is not required to merge.

[Learn more about requesting a pull request review.](#)

 1 pending reviewer



This branch has no conflicts with the base branch

Merging can be performed automatically.

[Merge pull request](#)

You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

External Code Review

Aaron Jacob Saeteurn Rayray

To: Tharun Krishna

Reply | Reply all | Forward | ...

Tue 7/30/2024 11:18 AM

Hey Tharun,

Our team has analyzed your code and left the following:

- Likes the well organization within each files
- Likes the use of using TS for better consistency
- Liked how the passwords were being hashed before storage, leading to better security
- Error handling could be improved, instead of error registering user, could specify where error is actually coming from to improve
- Integrating more comments to help testers who don't understand what's going on to still be able to acknowledge the information

Thank you for reviewing our code and we look forward to seeing your guys work at the end of the semester!

- Aaron, Team 01

From: Tharun Krishna <tkrishna1@sfsu.edu>

Sent: Tuesday, July 30, 2024 9:47 AM

To: Aaron Jacob Saeteurn Rayray <arayray@sfsu.edu>

Subject: Re: CSC648 External Code Review

Hey Aaron,

Our backend lead has reviewed your files and shared with me the following feedback:

- There could be specific comments that communicate what routers do what to a first-time reader.
- Authentication and security is spread out over 3 different files, this makes it harder to track.
- There could be more decoupling within the code. Everything is so interdependent on each other that making changes to the one class could affect everything else.
- The user class could be separated out, I feel that it's a little dense at the moment, that affects its readability.
- It appears that there is a split between functional programming and object oriented programming, you could try to consolidate the styles for consistency.

We appreciate you taking the time to conduct a code review with us, and wish you good luck with the rest of the project!

Thanks,
Krishna
Team 05

From: Aaron Jacob Saeteurn Rayray <arayray@sfsu.edu>

Sent: Monday, July 29, 2024 8:21 PM

To: Tharun Krishna <tkrishna1@sfsu.edu>

Subject: Re: CSC648 External Code Review

Hi Tharun,

Attached is our sign up module that deals with authentication and email verification. Would be appreciated if you guys could review the email verification portion of our signup process. I will get back to you either by tonight or tomorrow morning with your code review, thank you! (code will be sent over discord as it got blocked on here)

From: Tharun Krishna <tkrishna1@sfsu.edu>

Sent: Thursday, July 25, 2024 2:20 PM

To: Aaron Jacob Saeteurn Rayray <arayray@sfsu.edu>

Subject: CSC648 External Code Review

Hi Aaron,

I saw your message about the external code review and would like to take you up on your offer! I've attached a package that contains our user authentication logic, with the main file being authController.ts, for review. I've included three additional files that authController.ts imports for your reference as needed. Please feel free to reply to this email with a package or files of your own, and we will review them and get back to you ASAP!

Thanks,
Krishna
Team 05

Security Self-Check

In-line with modern standards, our application does not store unsecured plaintext passwords, but instead stored a secure hashed password to keep users' accounts safe.

When the backend receives the password, our controller calls a hashing function.

```
// HASH AND VERIFY HASHING PASSWORD
const hashedPassword:string = await hashPassword(registerUser.password);
const verify = verifyPassword(registerUser.password, hashedPassword);
if (!verify) {
    return res.status(400).json({
        error: 'Password Hashing unsuccessful'
    });
}
```

The hashing function uses bcrypt, and goes as follows:

```
import bcrypt from 'bcrypt';

// HASHES PASSWORD
async function hashPassword(password: string): Promise<string> {
    return await bcrypt.hash(password, 8);
}

// VERIFIES PASSWORD
async function verifyPassword(password: string, hashPassword: string): Promise<boolean> {
    return await bcrypt.compare(password, hashPassword);
}
```

This hashed password is what is passed in the MySQL query and stored in the DB:

	userID	email	username	password	profilePicture	firstName	lastName
▶	9141	testing@email.com	JohnDoe	\$2b\$08\$UFJrQUcykSWjhQNAYzR4D./30625/63mgTZUGnHPJajNyZqGnGKUy	NULL	John	Doe
*		NULL	NULL	NULL	NULL	NULL	NULL

Our search bar input data are validated to make sure we are searching the right tables in the database:

Our autocomplete function is as follows:

```
export const autoCompleteSearch = async (req: Request, res: Response) => {
  try {
    try {
      let token = req.cookies.token;
      let decoded = jwt.verify(token, JWT_SECRET_KEY);
      if (decoded === null) {
        throw new Error('Unauthorized User');
      }
    } catch (Error) {
      res.status(401).json({
        error: 'Unauthorized User'
      });
    }
    let searchQuery: string = req.body.searchQuery;
    // Query words from the database
    let members: Member[] = await
getMembersFromOrganizationQuery(req.cookies.orgID);
    let queryWords: string[] = members.map((members) =>
members.firstName + " " + members.lastName);
    const completer = new Trie();
    for (const word of queryWords) {
      completer.insert(word);
    }
    const prefix: string = searchQuery;
    const result: string[] = completer.search(prefix);
    res.status(200).json({
      queryResults: result
    });
  } catch(Error) {
    res.status(400).json({
      error: 'Error searching for topic'
    });
  }
}
```

Our ranked search function is as follows:

```
export const rankedSearch = async (req: Request, res: Response) => {
    try {
        let cookie: Cookie = await getCookies(req);
        let verified = verifyToken(cookie.token);
        if (verified === null) {
            return res.status(401).json({
                error: "Unauthorized User"
            });
        }
        let searchQuery: Search = req.body;
        let org = req.cookies.orgID;
        if (searchQuery.searchType === "members") {
            let members: Member[] = await
getMembersFromOrganizationQuery(org);
            let rankedMembers: any = rankSearched(searchQuery.searchQuery,
members.map((member) => member.firstName + " " + member.lastName));
            let sortedMembers: Member[] = rankedMembers.map((rankedMember:
{ item: string, score: number }) => {
                return members.find(member => (member.firstName + " " +
member.lastName) === rankedMember.item);
            }).filter((member: Member | undefined): member is Member =>
member !== undefined);
            return res.status(200).json({
                message: "Search for members",
                queryResult: sortedMembers
            });
        }
        // Truncated for formatting, but we have an else if for every case
    } catch(Error) {
        return res.status(400).json({
            error: 'Error searching for topic'
        });
    }
}
```

Non-Functional Specs Self-Check

Performance

- **DONE** - The application shall respond quickly to user prompts with minimal latency.
- **DONE** - The application shall be able to host/handle up to 1,000 accounts without any performance problems.
- **DONE** - The application shall support many active-concurrent users without significant degradation in performance.

Security

- **DONE** - The application shall only store hashed passwords.
- **ON TRACK** - The application database shall be backed up regularly.

Coding Standard

- **DONE** - The application shall follow consistent coding and naming conventions throughout.
- **DONE** - The application shall gracefully handle errors without detrimentally affecting the rest of the user experience.
- **ON TRACK** - The application shall be well documented for easy understandability for other developers and admins alike.

Media Storage

- **ISSUE** - The application shall restrict profile pictures to 500x500 pixels and 5 megabytes or less.
 - Images are not part of our priority 1 requirements, and as such, are not included in our prototype.

Privacy

- **DONE** - The application shall hide organizations, preventing users not in the organization from viewing it.
- **DONE** - Data shall be anonymized and pseudonymized wherever possible to minimize damage in the event of a leak.

- **DONE** - Only the minimum amount of data required shall be stored in order to minimize the risk of security breaches.

Scalability

- **DONE** - The application shall be designed to scale efficiently to handle an increasing number of users and data without performance degradation.
- **DONE** - The database shall scale down if the workload is low to remain cost effective and optimize resource utilization.

Reliability

- **DONE** - The application shall ensure high availability, with a 99.9% uptime, and incorporate robust error-handling mechanism to maintain stability.

Usability

- **DONE** - The application shall provide an user friendly interface, ensuring that users can easily navigate and use all features without extensive training.

Compliance

- **DONE** - The application shall comply with relevant industry standards and regulations, such as for data protection and privacy.

Expected Load

- **DONE** - The application shall respond to user queries within 500 milliseconds for requests.
- **DONE** - The application shall handle a minimum of 1000 transactions per second during peak hours.
- **DONE** - The application shall horizontally scale to support up to 10,000 concurrent users without degradation in response time.

Fault Tolerance

- **DONE** - The application shall recover from a server failure within 15 minutes without loss of data or interruption of service.
- **DONE** - Critical components of the application shall have a backup system that can take over within 5 minutes of a primary system failure.
- **DONE** - The application shall log all errors and provide automatic recovery for non-critical errors within 60 seconds.

Database High Level Specs

- **DONE** - The database shall use appropriate indexes to optimize query performance, especially for frequently accessed data.
- **DONE** - The database should be capable of storing up to 1 terabyte of data efficiently, with automatic data compression for storage optimization.
- **DONE** - All sensitive data stored in the database should be encrypted.

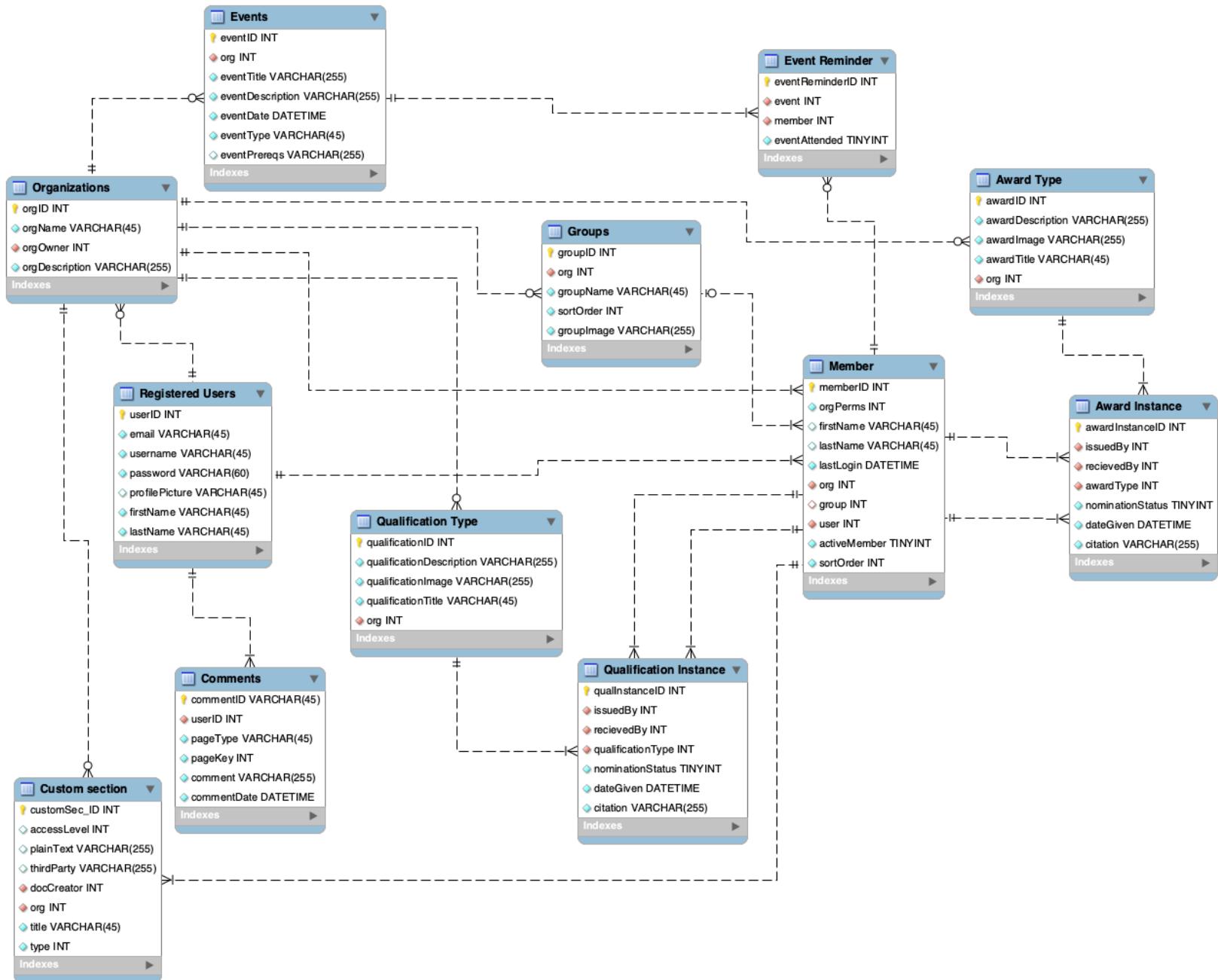
Browser and OS Support

- **DONE** - The application shall support Chrome Stable 126 (126.0.6478.153).
- **DONE** - The application shall support Windows 10 22H2 (19045.4651).
- **DONE** - The application shall support Windows 11 23H2 (22631.3737).
- **DONE** - The application shall support macOS Sonoma (14.5).
- **DONE** - The application shall support Amazon Linux (2023.4.20240611.0).

Additional Comments

Changes to the DBMS

Adding a Title (VARCHAR 45) and Type (INT) field to the Custom Section entity in order to differentiate between Projects/Tasks and Highlights.



Team Contributions

- **Tharun Krishna**
 - Revised Product Summary Section of Documentation.
 - Usability Test Plan Section of Documentation.
 - QA Test Plan Section of Documentation.
 - Security Self-Check Section of Documentation.
 - Non-Functional Requirements Self-Check Section of Documentation.
 - Group page frontend of Beta Prototype.
 - Recognitions page frontend of Beta Prototype.
- **Tyron Cheung: Contributions Score 10**
 - Prototype revisions as per horizontal prototype feedback.
 - General restyling of all pages.
 - Oversaw other front-end developers progress on static pages.
 - Refined static pages.
 - Tasks/Projects page frontend of Beta Prototype.
 - Events page frontend of Beta Prototype.
 - Highlights page frontend of Beta Prototype.
- **Randolf Gabrielle Uy: Contributions Score 10**
 - Leading efforts on Backend of the Prototype.
 - Revised search controller of Beta Prototype.
 - Created recognition controller of Beta Prototype.
 - Created custom sections controller of Beta Prototype.
 - Revised password hashing functions.
 - Added various routing components for the backend.
 - Miscellaneous bug fixing on the backend component.
 - Completed Internal Code Review.
 - Complete External Code Review.
- **Carlos Hernandez: Contributions Score 10**
 - Revised custom sections queries of Beta Prototype.
 - Revised custom sections controller of Beta Prototype.
 - Adjusted DBMS files (EER and .sql files).
 - Revised group queries of Beta Prototype.
 - Miscellaneous bug fixing on the backend component.

- **Xuefeng Guan (Tony): Contributions Score 8**
 - Revised Case Studies page of Beta Prototype.
 - Revised Reviews page of Beta Prototype.
 - Revised Updates page of Beta Prototype.
 - Revised Server Status page of Beta Prototype.
 - Revised Features page of Beta Prototype.
 - Revised Pricing page of Beta Prototype.
 - Revised About page of Beta Prototype.
- **Naw San: Contributions Score 8**
 - Revised Contact Us page of Beta Prototype.
 - Revised Chat Support page of Beta Prototype.
 - Revised Report a Bug page of Beta Prototype.
 - Revised Careers page of Beta Prototype.
 - Revised Culture page of Beta Prototype.
 - Revised Blog page of Beta Prototype
 - Revised Help Center page of Beta Prototype
- **Yasmine Sucu: Contributions Score 6**
 - Product Summary Description Component of Documentation.
 - Product Summary Superior Features Component of Documentation.
 - Product Summary Competitors Component of Documentation.
- **Jannelly Hernandez Umanzor: Contributions Score 6**
 - Product Summary Committed Functions Component of Documentation.
 - Contributed to Internal Code Review.
 - Miscellaneous contributions to backend API.

Post Analysis

During the project, our main challenges were overcoming inconsistencies in the coding aptitude of team members. Some team members were very capable, while others struggled with trivial tasks. We ended up dividing up the work in such a way that trivial tasks were handed off to those that were less capable; but as we progressed, we ran out of trivial tasks to handoff. We tried giving these members more sophisticated and impactful tasks, but a lot of the contributions made were inconsistent with what our mockups and wireframes depicted. At some point the more capable team members decided that it would just be more efficient to do most of the work ourselves, rather than having to revise everything multiple times.

A good portion of the team also lacked many of the baseline fundamental web development skills, and we struggled with team members submitting code that was more-likely-than-not generated by ChatGPT. These files needed to be redone, which reduced productivity of our more capable team members because they had to redo the work of others. It felt unfair to simply deny work to some members of the team, so I opted to give the opportunities to prove themselves with the understanding that their work would likely have to be revised anyway. A lot of time and bandwidth was expended trying to mentor team members through their development tasks, which were fairly trivial and simple in all actually. It's possible that with a smaller team, we might have been more efficient and productive, since we wouldn't have had to hold the hands of others as we moved along.

If I had to recreate the project from scratch, I would definitely include Tailwind for CSS in our tech stack. Our stylesheets had a lot of similar code that became tedious to write out. Tailwind, in conjunction with traditional CSS, would have made life a lot easier. We could have used traditional CSS and stylesheets for reused elements, and Tailwind for unique one-of-a-kind stylings to keep the stylesheet readable and small. Without tailwind, the stylesheets became very unwieldy and bloated; many times a stylesheet change intended for a certain component would end up changing something in another page entirely. Tailwind would have definitely boosted our productivity and efficiency.

Another issue was project bloat, due to the tight deadlines around summer semester, we had to move through a lot of the initial setup hastily. Less than ideal choices and foundations were created which worked at the time, but created issues during milestone 3. The free AWS EC2 instance did not have enough memory to actually compile our project. I ended up upgrading to a paid version because of the tight deadline, but if we had followed good practices early on, it's likely that we could have gotten away with just using the free AWS EC2 instance.

Team Contributions

- **Tharun Krishna**
 - Document editing and formatting for all milestones.
 - Contributed use cases, requirements, and data entities to milestone 1.
 - Created, managed, and migrated AWS cloud instances.
 - Notion creation and updating for each milestone.
 - Revised dashboard pages in-line with Figma wireframes.
 - Merging separate stylesheets for core functions into shared stylesheets.
 - Groups and recognition page backend connection.
 - Nearly all sections of the milestone 4 documentation.
- **Tyron Cheung: Contributions Score 10**
 - Contributed use cases, all diagrams, and comp analysis to milestone 1.
 - Created all storyboards and mockups for milestone 2.
 - Created all wireframes for milestone 3.
 - Coordinated frontend team with prototypes, static pages, and revisions.
 - Created home, signing, dashboard, and other new user facing pages.
 - Created all changes related to the milestone 3 prototype feedback.
 - Events, tasks/projects, and highlights backend connection.
- **Randolf Gabrielle Uy: Contributions Score 10**
 - Created all backend/system/scalability/API design summaries and diagrams for milestone 2.
 - Restructured repo and application in-line with tech stack definition.
 - Coordinated and mentored other developers with backend interfaces.
 - Conducted internal and external code review for milestone 4.
 - Created extensive authentication and search backend components.
 - Created all other backend interfaces, controllers, and routings.
 - Enhanced application security with tokens, hashing, and cookies.
- **Carlos Hernandez: Contributions Score 10**
 - Created Database definitions, requirements, ERDs, EERs, and forward engineering for DBMS milestone 2.
 - Revised DBMS EER and forward engineering as needed for milestones 4.
 - Database to backend controller interfacing.
 - Created and revised database SQL queries for all prototypes.
 - Various bug fixes in backend controllers and models.
 - Coordinating with and mentoring other backend developers.
 - Assisted with connecting backend interfaces with the frontend.

- **Naw San: Contributions Score 7**
 - Contributed use cases, requirements, and comp analysis to milestone 1.
 - Created all network diagrams for milestone 2.
 - Created initial header, footers, pop ups, and corresponding styling.
 - Revised front page footer to final version.
 - Contributed to the initial recognitions page.
 - Contributed to contact us, chat support, report a bug, careers, culture, blog, and help center static pages.
- **Xuefeng Guan (Tony): Contributions Score 5**
 - Contributed use cases, requirements, and comp analysis to milestone 1.
 - Contributed to database ERD diagram in milestone 2.
 - Created initial dashboard sidebar.
 - Contributed to initial dashboard, members, tasks, and events page.
 - Contributed to case studies, reviews, updates, server status, features, pricing, and about static pages.
- **Yasmine Sucu: Contributions Score 3**
 - Contributed use cases, requirements, and comp analysis to milestone 1.
 - Polishing and editing the backend section of milestone 2 documentation.
 - Contributed to the project summary section of milestone 4 documentation.
 - Contributed to user profile backend API.
- **Jannelly Hernandez Umanzor: Contributions Score 3**
 - Contributed use cases, requirements, and comp analysis to milestone 1.
 - Contributed to Internal Code Review for milestone 4.
 - Contributed to the project summary section of milestone 4 documentation.
 - Contributed to organization backend API.