



### **Team members:**

Stephen Burns, Kevin Gao, Dhairya Amin, Alan Libby, Zach Bridwell

### **Problem Statement:**

Various classes have group projects and there are no dedicated services for finding teammates. This means that there is a lot of time spent by individuals searching for groups to work with and also by groups searching for additional members. This time could be much better spent actually working on the project. TeamUp provides a service for searching students based on classes and skills.

### **Background Information:**

Finding teams for projects can be difficult, especially if you don't know many people in the class. Currently, Piazza is the go-to way to find teams. However, that isn't very efficient because you must update the post every time a group is made, resulting in some people applying to full groups and failing to fill up other empty spots. Even when you do find a team, you may find out that the skill set of current team members is narrow or that some members' schedules don't work well together. Using TeamUp, users will be able to create groups by class. Each user will be able to enter any experience they have, as well as what times they definitely cannot, maybe can, and definitely can meet so that users can make informed decisions when deciding who to team up with. On top of this, groups can be updated

in real time when someone joins a group so that users do not have to manually mark a group as full.

## Environment:

We intend to use Javascript as our programming language because it is fast, versatile and it is used everywhere in the world. For the backend, we will use MongoDB database to store groups, categories and user information because MongoDB stores data in flexible, JSON-like documents. To connect the frontend and backend, we will use an Express server.

## Functional Requirements:

Backlog ID	Functional Requirement	Hours	Status
1	As a user, I would like to be able to create a TeamUp account.	6	Planned for sprint 1
2	As a user, I would like to login to my TeamUp profile.	3	Planned for sprint 1
3	As a user, I would like to edit my TeamUp profile.	4	Planned for sprint 1
4	As a user, I would like to start a new Team.	2	Planned for sprint 1
5	As a user, I would like to remove a Team.	2	Planned for sprint 1
6	As a user, I would like to see a list of all available Teams.	3	Planned for sprint 1
7	As a user, I would like to be able to filter the list of Teams.	5	Planned for sprint 1

8	As a user, I would like to see a detailed view of the users currently in a Team when I click on it.	6	
9	As a user, I would like to view other users' profiles.	4	Planned for sprint 1
10	As a user, I would like to attach documents to a Team.	6	
11	As a user, I would like to remove documents from a Team.	3	
12	As a user, I would like to download documents to view them.	5	
13	As a user, I would like to add other users before posting the Team.	6	
14	As a user, I would like to edit a Team post.	3	Planned for sprint 1
15	As a user, I would like to be able to have notes on a Team post.	4	Planned for sprint 1
16	As a user, I would like to logout of TeamUp.	4	Planned for sprint 1
17	As a user, I would like to be able to chat with other users.	7	
18	As a user, I would like to leave reviews for other users.	6	
19	As a user, I would like to assign tags to a Team.	3	
20	As a user, I would like to be notified each time someone requests to join or joins my team.	4	
21	As a user, I would like to be notified when my team is full.	2	
22	As a user, I would like to decide whether my Team is open or restricted.	2	Planned for sprint 1
23	As a user, I would like to see a list of all skills	5	

	currently possessed by team members.		
24	As a user, I would like to leave a Team.	2	Planned for sprint 1
25	As a user, I would like to assign skill preferences for my created Team.	3	Planned for sprint 1
26	As a user, I would like to be able to kick users from my Team.	3	
27	As a user, I would like to invite other users to join my Team.	5	Planned for sprint 1
28	As a user, I would like to block other users.	4	
29	As a user, I would like to search for a specific user	6	Planned for sprint 1
30	As a user, I would like to join or request to join a team.	5	

## Non-Functional Requirements:

### Architecture:

JavaScript will be the coding language in both the client and server. We will use React for the client and the Express framework for the server. We will use Talk.js for the chat functionality. We will be using MongoDB running on AWS to store our data.

### Security:

We will be using AWS' Identity and Access Management Database Authentication which allows to use an identification instead of storing user credentials in the database.

### Usability:

The interface should be easy to use for finding, participating in, and managing teams.

## Use Cases:

Case 1: Create a team up account

Action	System Response
1. On the login page tap register	2. Display signup page
3. Enter email and password	4. Send request to AWS
	5. If success ask for username
6. Enter username	7. Store name in AWS
	8. Send user to home page

Case 2: Login to teamup profile

Action	System Response
1. On the login page select login	2. Display login page
3. Enter username and password	4. Verify information through aws
	5. Send logged in user to home page

Case 3: Edit team up profile

Action	System Response
1. On profile page select edit profile	2. Verify user

	3. Send user to edit page
4. Edit page	5. Verify changes made
	6. Submit changes
	7. Return user to newly updated profile

#### Case 4: Start a new team

Action	System Response
1. On the home page select create new team	2. Display create new team page (or display model)
3. Fill out new team information	
4. Submit new team	5. Add new team to database
	6. If successful return to team listing

#### Case 5: Remove team

Action	System Response
1. Select remove team from team page	2. Remove team from database
	3. Display success if successful
	4. Send user to home page

#### Case 6: See a list of all available teams

Action	System Response
1. Select available teams button	2. Search mongodb database

from home page	for all available teams
	3. Take user to new page and display teams

Case 7: I would like to be able to filter the list of Teams.

Action	System Response
1. From available teams page, select filters to apply	
2. Select search for teams button	3. Search database for teams that apply to filters
	4. Display teams that apply on page

Case 8: View detailed info on all team members

Action	System Response
1. Select the page of a team that user is a member of	2. Load page with information about that team and a chat window
	3. Open a connection for real-time chat among team members
4. Select detailed team info button on team page	5. Server retrieves user info from database and sorts it as needed
	6. Server displays detailed user info on team page

Case 9: View a user's profile

Action	System Response
1. Search for a user via the search bar	2. Server retrieves user list from database and filters it as needed.
	3. Server displays sorted users
4. User selects the user that they would like to view the profile of	5. Server retrieves profile info from database
	6. Server displays profile info on profile page

#### Case 10: Upload a file for a team

Action	System Response
1. Select the page of a team that user is a member of	2. Load page with information about that team and a chat window
	3. Open a connection for real-time chat among team members
4. Select the upload file button	5. Open a file picker dialog
6. Select a file	7. Upload the file and use chat connection to push the link to other users

#### Case 11: Remove an uploaded file from a team



Action	System Response
1. Select the remove button next to the file link in the chat window	2. Remove the link from the chat log
	3. Remove the file from the server

Case 12: I would like to download documents to view them.

Action	System Response
1. Select the link to the document from the chat window	2. The server will return the file to be downloaded by the browser

Case 13: I would like to add other users before posting it.

Action	System Response
1. Select Create option to start a team	2. Display the create team page
3. Fill up details about the team	4. Display textbox to input team members emails
5. Enter email of users to add to the team	
6. Select Post	7. Post is added to the team lists in app and database

Case 14: I would like to edit a Team post.

Action	System Response
1. Select the edit button next to the field to be edited	2. Make the field editable, create a button to save the changes
3. Click the save button	4. Return the field to a view-only state
	5. Update the field in the database to reflect the changes

Case 15: I would like to be able to have notes on a Team post.

Action	System Response
1. From a team page, select the option to add a note	2. Create an editable field for the body of the note
3. Select the option to create the note	4. Add the note to the database

Case 16: I would like to logout of TeamUp.

Action	System Response
1. From bar at the top of any page, select logout button	2. Ask user for confirmation they want to logout
3. Select confirm button	4. Logout user from site
	5. Return user to main page

Case 17: Chat with other members of a team

Action	System Response
1. Select the page of a team that user is a member of	2. Load page with information about that team and a chat window
	3. Open a connection for real-time chat among team members
4. Type message into chat window and select send	5. Update chat log with message
	6. Use chat connection to push message to other users currently viewing chat window

#### Case 18: Leave reviews for other users

Action	System Response
1. User navigates to profile page of another user either via search or from team page	
2. User selects the leave a review button	
3. User fills out the displayed dialog box for review information	4. Server stores review on user's profile in database

#### Case 19: Assign tags to a team

Action	System Response
1. User navigates to team settings via the team page	

2. User inputs tags into the tags field on the team settings page.	3. Server updates team information in the database.
	4. Server sends confirmation to user that settings were changed.

Case 20: Be notified when user joins or requests to join team

Action	System Response
1. User joins or requests to join a team	2. Server processes the request and updates the database accordingly.
	3. Server sends notification email to the team leader.

Case 21: Notify when team is full

Action	System Response
1. User joins a team, either directly or via accepted request	2. Server updates the database accordingly, and checks if the team is at capacity.
	3. If the team is at capacity, the server sends a notification email to the team leader specifying that the team is full.

Case 22: Set team to be open or restricted

Action	System Response
1. Select Create Team	2. Display option to make team open or restricted
3. Select open or restricted	4. Show the option as marked
5. Select Post Team	6. Save open or restricted option in the database
	7. Show button to join the team for open and show button to request for restricted

Case 23: See a list of all skills possessed by other team members

Action	System Response
1. Select 'view team'	2. Server creates list of skills possessed by team members by getting data from their profile
	3. Display the newly created list of skills

Case 24: Leave a team

Action	System Response
1. Select 'leave team' from team page	2. Ask for confirmation to leave team
3. Confirm you want to leave team	4. Remove user from team via database

	5. If successful send user 'successfully left team'
	6. Send user back to home page

#### Case 25: Assign skill preferences for the team

Action	System Response
1. Select Create Team	2. Display options to add skill preferences
3. Select skill preferences	
4. Select Post Team	5. Save skill preferences in the database
	6. Display the skill preferences on the Team post

#### Case 26: Kick other users from team

Action	System Response
1. User navigates to member list of a team that they are a part of.	
2. User selects the option to kick a particular team member next to their name.	3. Server asks for confirmation that the user wants to kick the team member.
4. User confirms that they want to kick this team member.	5. Server processes the request and updates the database, as well as sending a notification email to the kicked user.

#### Case 27: Join a team from an invite

Action	System Response
1. [Owner of a team] from a user's profile page, select invite to team	2. If user can invite people to >1 team, show user a list of teams they can invite the person to
	3. Show confirmation dialog
4. Select a team if needed, confirm invite	5. Notify recipient of pending invite
	6. Notify user that the invite was sent
7. [Invite recipient] select accept on the invite	8. Add user to list of team members in MongoDB
	9. Notify team owner that invite was accepted

#### Case 28: Block other users

Action	System Response
1. Select 'block user'	2. Show confirmation popup
3. Confirm block	4. Add the blocked user to a blocked users list for the current user

#### Case 29: Search users

Action	System Response
1. User types name or email of	2. Display list of all users with

another user into search box	that name and/or email address

### Case 30: Join a team by request

Action	System Response
1. From the list of teams detailed in case 6, select the request to join option for a team	2. Notify user that the request was sent
	3. Notify team owner of the pending request
4. [The team owner] select accept on the request	5. Add user to the list of team members in MongoDB
	6. Notify original user that their request was accepted