

# COMP1531 TEAM CRUNCHIE

## **Elicitation:**

Find 2-3 people to interview as target users. Target users are people who currently use a tool like Treats, or intend to. Record their name and email address.

Develop a series of questions (at least 4) to ask these target users to understand what problems they might have with teamwork-driven communication tools that are currently unsolved by Treats. Give these questions to your target users and record their answers.

Once you have done this, think about how you would solve the target users' problem(s) and write down a brief description of a proposed solution.

## **Questions to ask:**

1. What problems do you have with teamwork-driven communication tools that are currently unsolved by our system?
2. How would you improve this system?
3. What is the most useful feature of this system?
4. What are some features you dislike about our system?

## **Answers:**

User 1: James Marshall (j.marsh1177@gmail.com)

1. This system seems like it has all the main features of a communication platform. However, these days all mainstream communication platforms like microsoft teams and discord have a feature to block users.
2. I would want a feature to delete an account.
3. I think the most useful features are the variety of features relating to messaging and all the different channels.
4. I dislike the fact that this system doesn't use GUI. It makes it hard to use.

User 2: Joni Granger (joni.granger24@gmail.com)

1. This system doesn't have many features and one of the most important in a communication platform, the ability to follow other users.
2. I would like a feature to follow/become friends with a user.
3. I think the most useful feature of this system is the dm feature.
4. I dislike the inability to use voicechat and call others.

## **Analysis & Specification - Use Cases:**

Once you've elicited this information, it's time to consolidate it.

Take the responses from the elicitation step and express these requirements as user stories (at least 3). Document these user stories. For each user story, add user acceptance criteria as notes so that you have a clear definition of when a story has been completed.

Once the user stories have been documented, generate at least ONE use case that attempts to describe a solution that satisfies some of or all the elicited requirements. You can generate a visual diagram or a more written-recipe style, as per lectures.

### **User stories:**

1. As a user of other communication platforms like discord and microsoft teams, I want to have a feature to block other users so that I don't get harassed.

#### **User acceptance criteria:**

- So that there is no way for blocked user to interact with you
- blocked users basically disappear from your platform
- you can unblock users
- all their messages and channels are deleted from your platform

2. As a microsoft teams user, I want to be able to delete my account so that if I don't want to use it anymore, people know I don't use it anymore.

#### **User acceptance criteria:**

- Other users cannot find deleted account anymore i.e. they cannot invite to channel, dm etc.
- When the user is deleted from system, if they wanted to they can register with the same email they used before
- deleting process must not take more than 5 minutes

3. As a facebook messenger user, I want to be able to follow/become friends with another user so that I can easily find them through my account.

#### **User acceptance criteria:**

- be able to see all your friends or followers
- be able to follow any account
- not be able to follow deleted user accounts

### **Use case:**

- **Use Case:** Be able to delete, follow, block accounts in this system
- **Goal in Context:** a user can be able to delete their own account, follow other accounts or block other accounts voluntarily and others can block and follow you.
- **Scope:** UNSW Treats
- **Preconditions:** All users are registered with an account and is logged in.
- **Success End Condition:** The user has followed, deleted, blocked an account.
- **Failed End Condition:** The user has not followed, deleted, blocked an account.
- **Primary Actor:** User
- **Trigger:** User selects option to delete their account, follow or block and account, or other user follows or blocks you.

#### **Success Scenario 1:**

1. User selects option to follow another account
2. Be able to see the account followed on the user's account
3. Account followed gets a follow notification
4. Other user follows back
5. User is shown on other user's follows

### **Validation:**

With your completed use case work, reach out to the 2-3 people you interviewed originally and inquire as to the extent to which these use cases would adequately describe the problem they're trying to solve. Ask them for a comment on this, and record their comments in the PDF.

User 1: James Marshall (j.marsh1177@gmail.com)

These features make this system more modern as most communication platforms like discord, facebook messenger, instagram all have a feature to follow and be followed, block or be blocked, and delete your account.

User 2: Joni Granger (joni.granger24@gmail.com)

The features make this system better to use as if I don't want to use it anymore, I can delete my account without ever leaving a trace. Being able to follow your friends makes this system more lively and enjoyable to use.

## Interface design:

Now that we've established our problem (described as requirements), it's time to think about our solution in terms of what capabilities would be necessary. You will specify these capabilities as HTTP endpoints, similar to what is described in 6.2. There is no minimum or maximum of what is needed - it will depend on what problem you're solving.

Name & Description	HTTP Method	Data Type	Exceptions
auth/delete/v1  Given authUserId of user, this user's account is deleted from the system	DELETE	Query parameters: { authUserId } Return type if no error: { }	400 Error: -When authUserId is not a valid one.
user/profile/block/v1  Given authUserId of user and uld of another, user blocks the other user so they don't interact anywhere on the platform	POST	Body parameters: { authUserId, uld }  Return type if no error: { }	400 Error: -When authUserId is not a valid one. -When uld is not a valid one
user/profile/follow/v1  Given authUserId of user and uld of another, user follows the other user. On user's profile it shows their following and followers	POST	Body parameters: { authUserId, uld }  Return type if no error: { }	400 Error: -When authUserId is not a valid one. -When uld is not a valid one  403 Error: -When user is already following the other user -if user is blocked

## Conceptual Modelling - State Diagrams:

Now that you have a sense of the problem to solve, and what capabilities you will need to provide to solve it, add at least ONE state diagram to your PDF to show how the state of the application would change based on user actions. The aim of this diagram is to help a developer understand the different states of the application.

