

Project Development Report

Summary

T13A BOOST has developed and implemented a backend JavaScript server for a new communication platform, UNSW Beans. It is a tool that allows users to communicate, share, and collaborate virtually. With each iteration, we develop and test more features according to the specifications and needs of the user. This report will focus on understanding problems, developing requirements, and drafting some early designs from user feedback.

Elicitation

In this section, we ask a series of questions to target users to understand problems with the current UNSW Beans. A proposed solution is then developed to focus on the feedback given.

Table of interviewees:

Name	Profession	Email
Jessica Fan	Sponsorship coordinator	fan-jessica@gmail.com
Dean Luo	University Student	d.luo2002@outlook.com
Callum Parsons	University Student	calparsons8442@gmail.com

Interview 1 – Jessica Fan:

Question 1. Do you currently use any collaborative communication platforms similarly to UNSW Beans? and what do you like about the app?

My workplace commonly uses Microsoft Teams and Outlook to communicate with other co-workers. Microsoft Teams is very versatile and allows me to communicate with my team members daily. It also integrates perfectly with other Microsoft products such as Word and Excel that I use all the time.

Question 2. What is the most common feature that you use in the platform?

I mainly use the message feature because it's my preferred method of communicating with my team.

Question 3. What feature from UNSW Beans hasn't been implemented that you would like to see?

I would like to see a feature that allows me to set my activity status. For example, online, busy, away offline, etc.

Question 4. Why is this implementation important and how does it affect your usage of the platform?

In Microsoft Teams, it usually sets my status automatically depending on how active I am via usage of keyboard strokes and mouse activity. It also sets my status as busy during a meeting if it is scheduled in the calendar. You can also manually set the status yourself as well. This is a great feature as it indicates to my team that I am not able to respond to their inquiries quickly if I am set to busy or away. Which is why I think it is a great feature to add to UNSW Beans.

Interview 2 – Dean Luo:

Question 1. Do you currently use any collaborative communication platforms similarly to UNSW Beans? and what do you like about the app?

Yes, I use Discord, which is a communication platform built for gamers, Microsoft Teams which I use for school related projects, and Outlook as my main email service provider. Discord is a great platform because it lets friend communicate directly via voice, video, or text, and join servers with different communities can interact with each other.

Question 2. What is the most common feature that you use in the platform?

The most common feature I use is probably message and voice chat. I use Discord a lot so I can chat to my friends and communicate with each other while we play Valorant.

Question 3. What feature from UNSW Beans hasn't been implemented that you would like to see?

After getting familiar with the functionality of UNSW Beans, I believe that it's missing a channel/delete feature.

Question 4. Why is this implementation important and how does it affect your usage of the platform?

I believe that if a channel is created, then the channel owner also can delete it. This helps with management as the channel owner can just remove old channels that doesn't serve a purpose and help navigate through more important channels easily. The channels are more concise, relevant, and tidier to the user.

Interview 3 – Callum Parsons:

Question 1. Do you currently use any collaborative communication platforms similarly to UNSW Beans? and what do you like about the app?

Discord is my main platform that I use for communication. I like easy it is to use and it has a lot of features that are relevant to my lifestyle. Watching my friends stream on the app is great and it doesn't feel slow when I do use it.

Question 2. What is the most common feature that you use in the platform?

Messaging is the most common feature that I use in Discord.

Question 3. What feature from UNSW Beans hasn't been implemented that you would like to see?

I think you need a mute function because you're going to get notified every time a message is sent so it can get annoying and distracting when you're trying to focus.

Question 4. Why is this implementation important and how does it affect your usage of the platform?

Because you'll need to be notified when someone sends a message so you can reply. However, often it is mainly not directed at you or could be a random conversation in a channel. Receiving notifications when it's not related to you can be annoying if you're trying to focus so having the ability to mute can help.

Proposed solution:

We have identified three key features gathered from the interviews to implement to UNSW Beans. These are:

1. User activity status – A user can set their status that is shown to other users.
2. Channel remove – a channel owner can delete an existing channel
3. Mute notifications – a user can not be notified for any mentions in channels.

Analysis and Specifications – Use Cases

Use cases is a sequence of events that defines the interaction between a user and system to achieve a goal. Below is a series of user stories created from the proposed features to be implemented.

User Story 1: User activity status

An employee must run off to a meeting and set her status to 'away' so that her team members know that she is not able to respond to them quickly.

User Acceptance Criteria:

1. Scenario: Wants to set their activity status.
2. Given: 'Online' status (indicated with a green icon) is given when user is logged in and 'Offline' (indicated with a grey icon) when user is logged out.
3. When: User can set their status via a drop-down menu with status options under their profile name. Busy, away, Invisible etc.
4. Then: User activity status is updated and is displayed.
5. When: User sets status to busy.
6. Then: Status icon is the colour red.
7. When: User sets status to away.
8. Then: Status icon is the colour yellow.
9. When: User sets status to invisible.
10. Then: Status icon is the same as offline.

Use case: User is going to be away from the computer.

Goal in Context: User wants to set status to 'Away' to let team members know.

Scope: UserId.

Preconditions: User's status is online when logged in and offline when logged out.

Successful End: User's status is set to 'Away' and icon is yellow.

Failed End: User's status remains the same.

Primary Actor: User.

Trigger: clicking on the status option button.

Validation

Question: From the previous feedback, is the acceptance criteria and use case for the proposed feature satisfactory?

Jessica: Yes, that's exactly how I would like it to be.

User Story 2: Channel Delete

A channel created for a recently completed project is not irrelevant and is taking up space. The channel owner would like to delete the channel so it can be cleared up.

User Acceptance Criteria:

1. Scenario: Wants to delete a channel
2. Given: User navigated to the channels page
3. When: User clicked on the delete channel button.
4. And: System checks if the user is the channel owner.
5. Then: System then deletes the channel if user is channel owner.

Use case: Channel is irrelevant and is no longer needed.

Goal in Context: User wants to delete the channel.

Scope: UserId, channelId.

Preconditions: User is the channel owner.

Successful End: Channel is deleted.

Failed End: Channel is not deleted.

Primary Actor: User.

Trigger: clicking on the delete channel button.

Validation

Question: From the previous feedback, is the acceptance criteria and use case for the proposed feature satisfactory?

Dean: If the functions allow me to delete channels, then it is good to me.

User Story 3: Mute notifications

A student is studying for their exam and wants to mute because they are receiving notifications quite frequently and is getting distracted.

User Acceptance Criteria:

1. Scenario: User wants to mute their notification.
2. Given: User navigated to channel that needs to be muted.
3. When: User clicks on the mute button.
4. Then: Channel stops notifying the user.

Use Case: User clicks on the mute button of a channel.

Goal in Context:

Scope: Channel to be muted.

Preconditions: Channel must exist.

Successful End: Channel stops notifying user.

Failed End: Channel still notifies user.

Primary Actor: User.

Trigger: clicking on the mute button.

Validation

Question: From the previous feedback, is the acceptance criteria and use case for the proposed feature satisfactory?

Callum: Yes, I am satisfied with these features if they are implemented correctly and allows me to not receive any notifications.

Interface Design

Name and Description	HTTP Method	Data Types	Exceptions
/user/userStatus/V1 Given a valid status the user status is set to status.	POST	Body Parameters: { status } Return type if no error: { }	400 Error: status passed is not a valid option.
Channel/channelDelete/V1 Given a channelId and token. Checks if user is the channel Owner of the channel and delete.	POST	Body Parameters: { uld, channelId } Return type if no error: { }	400 Error: ChannelId is invalid. uld is not a valid user. uld is not a member of the channel. 403 Error: ChannelId is valid and uld is not owner of the channel but is a member.

Channel/channelMute/V1	PUT	Body Parameters: { uld, channelId } Return type if no error: {}	400 Error: ChannelId is invalid. uld is not valid. 403 Error: uld is not a member of the channel.
------------------------	-----	---	---

Conceptual Modelling - State Diagrams

This is a diagram that showcases a user's journey within the application given the appropriate inputs.

