

1531 - Planning for next problems to solve M13A_BOOST

Report Overview

This report will include a short description of the user problems and stories our team, M13A_BOOST has identified to solve in a hypothetical future iteration of UNSW Treats. After interviewing several potential users of our program, we have identified several key features to include in future iterations, as well as completing basic modelling and interface design for these features.

Contents

[Interviews](#)

[Analysis & Specification - Use Cases](#)

[Validation](#)

[Interface Design](#)

[Conceptual Modelling - State Diagrams](#)

Interviews

We asked each of our interviewees 4 questions to gather useful data about future developments of UNSW Treats. The questions are as follows:

Q1: What app are you currently using, and what is one unique feature you like about this app?

Q2: What is the feature you use most commonly in the app you're currently using?

Q3: What features don't currently exist that you would like to see implemented in your app?

Q4: How would this be useful for your profession?

The table of interviewees is below.

<i>Name</i>	<i>Email</i>	<i>Profession</i>
Aleisha Lawrence	aleishlawrence02@gmail.com	High school coach
Daniel O'Dea	tlx494@gmail.com	Software reliability engineer
Gautama Desaraju	gautamades@gmail.com	Tertiary student

INTERVIEW 1:

Name - Daniel O'Dea

Email - tlx494@gmail.com

Profession - Software reliability engineer - Atlassian

Q1: What app are you currently using, and what is one unique feature you like about this app?

As a part of my role with Atlassian, I currently use Slack for regular communication with my team. Slack is super important to me, as I work from home and use Slack to communicate with my team and boss on a daily basis. One unique feature I particularly like about Slack is the ability to split my projects and messages across different workspaces. Being able to have a different workspace for developers, the reliability team and the whole company to work between means that there are clear boundaries between different workspaces

Q2: What is the feature you use most commonly in the app you're currently using?

The feature I currently use the most is the mute notifications feature on different channels. As I get so many different messages a day from various channels, being able to mute notifications from channels that are not essential to keep up to date in is very important.

Q3: What features don't currently exist that you would like to see implemented in your app?

I would like for a live call feature to be added into slack, so that I don't have to switch between different accounts for zoom, microsoft teams and google meet for different meetings.

Q4: How would this be useful for your profession?

This would make keeping up in communication with coworkers a lot easier. Currently, I have to switch between different accounts and have many applications open at the same time, which is very inconvenient. Being able to natively make calls through slack would improve my time usage.

INTERVIEW 2:

Name - Aleisha Lawrence

Email - aleishalawrence02@gmail.com

Profession - High school coach

Q1: What app are you currently using, and what is one unique feature you like about this app?

I currently use emails as my primary form of communication for my job as a high school coach. The feature I like most about emails is probably the notifications I get each time a new email is received, because it lets me stay on top of communications from my students and from my boss.

Q2: What is the feature you use most commonly in the app you're currently using?

I mostly commonly use the send email feature to my students to let them know about any changes to class, or to send through notes after a class is complete. This works well, but is not

incredibly convenient as I have to open and send a new email every time I want to send through a message to my students.

Q3: What features don't currently exist that you would like to see implemented in your app?

I would like to see some sort of entertainment feature added to my app, that allows me to send fun games as an attachment or function of emails to make my students smile and have a fun time when engaging with the messages that I send.

Q4: How would this be useful for your profession?

This would be very useful for my profession, as my students are often disengaged with content or not interested in the subject matter that I am teaching. However, adding more interactive and fun elements to emails will make students more likely to read the emails, and take in the content that I send, as well as enjoy the fun game or interactive element that is attached to the email.

INTERVIEW 3:

Name - Gautama Desaraju

Email - gautamades@gmail.com

Profession - Tertiary student

Q1: What app are you currently using, and what is one unique feature you like about this app?

As a university student at UNSW, I commonly use Microsoft Teams to communicate with team members for group assignments. I like the ability to have calls within the app, as it saves the need for starting a separate Zoom or Google Meet call.

Q2: What is the feature you use most commonly in the app you're currently using?

I most commonly use the different channels feature, as I have multiple simultaneous group projects running for different courses as well as for a society that I am a part of. Being able to keep track of all the messages and attachments that have been sent across the various channels is extremely helpful in tracking progress and staying updated for my many projects.

Q3: What features don't currently exist that you would like to see implemented in your app?

I currently don't make use of the set user status function. The ability to set my status as either active, away, online or busy etc. means that I am able to show how I am displayed to other group members.

Q4: How would this be useful for your profession?

This is very useful as I don't always want to be seen as available to all my different team members. For example, if I am working on a uni group assignment late at night, I don't want my society president to see that I'm online and DM me with other work to do, as I'm currently too busy. By setting my status as away, it will show the president that I'm not available right now to get any tasks done.

Proposed solution:

From the interviews conducted, we have identified 3 key functions to add to UNSW Treats.

1) Set user status

Allowing a user to set various statuses viewable to other users. This may include settings such as away, busy, do not disturb, online and offline.

2) Hangman game

Allows a pop-up of a hangman game within a channel, with the logic and functionality of the game handled by a series of backend API calls. The game must be solved within a 3 minute timer.

3) Mute notifications setting

Allowing an individual to turn off notifications, meaning that they will not receive any tags or notifications from mentions in channels.

The proposed backend implementation of these functions will be described in detail and shown below in the interface design.

Analysis & Specification - Use Cases

Below are a series of user stories, to clearly illustrate the reason to create each of the functionality described above. For each user story, a series of acceptance criteria has been outlined, so that there is a clear definition of when the story is completed. Additionally, each of the user stories has use cases that describes the solution

1) Set user status

User story: As a student, I want to manually display my user status, so that people will know not to bother me when I'm busy.

User Acceptance criteria:

- The user is valid and logged in
- The default status for an online user is 'online', and an offline user is 'offline'
- The status for other users is displayed in the frontend when the cursor clicks a user's profile
- The status for the current user is displayed in the top right of screen
- A status indicator is displayed next to the status - green for online, red for offline, and orange for all other states
- A user can change their displayed status to all other options
- The user can change the status by clicking a drop down menu of options when hovering over their status

Use case:

- User logs in

- User status is 'online'
- User switches status to away when they are taking a break
- Status is displayed as 'away' to other users

Goal in context: To externally display to others the online status of a user

Scope: This applies to all users within UNSW Treats

Preconditions: All users must first be logged in, and have an account. The default status upon login is 'online'

Success end condition: The status of the user is changed upon request

Failed end condition: Status remains the same

Primary actor: The user who is logged in

Trigger: Frontend API call to select a status from a dropdown menu of statuses

2) Hangman game

User story: As a young school student using UNSW Treats, I want to be entertained while I use the app so that I stay more engaged with learning.

User Acceptance criteria:

- User must be valid and logged in
- A hangman game must not currently be in existence
- User can enter guess through a search bar interface
- The search bar will accept queries from the message tab
- A correct guess will reveal the letters for that guess
- An incorrect guess will lose a life for the hangman
- Upon success, a success message will play, the interface will close
- Upon failure, a failure message will play, the interface will close

Use case:

- The user is logged in
- The user is a member of a channel
- A user presses a button on the display labelled hangman
- A 5-10 letter word is selected at random
- Hangman interface appears in the channel display screen
- All users within the channel can make hangman guesses
- There is a timer ticking down from 180 seconds
- Users must attempt the game before time runs out

Goal in context: To play a hangman game within a given channel

Scope: Within the frontend of a given channel. The permissions and logic is handled by the backend

Preconditions: The channel and user both exist, a hangman game is not already being played

Success end condition: All the letters of the hangman guess are correctly guessed

Failed end condition: The 'hangman' runs out of lives

Primary actor: The users within a given channel play the game, an individual user starts the game

Trigger: A 'start hangman' button within the channel interface

3) Mute notifications setting

User story: As a software engineer using UNSW Treats, I want to be able to mute notifications, so that I do not receive notifications from non-relevant channels.

User Acceptance criteria:

- User must be valid and logged in
- Channel to mute notifications in must exist
- User will no longer receive notifications from tags in a given channel
- The tags will not be shown

Use case:

- User logs in
- User sees many notifications from irrelevant channels
- User navigates to that channel
- User hits the mute notifications button
- Notifications for future pings in that channel do not show up for the user
- User wants to see notifications again
- User clicks the unmute notifications button
- User can now see future notifications

Goal in context: To enable a user to mute notifications from undesirable channels

Scope: A channel (channels) for a given user

Preconditions: The channel and user must both exist and be valid, the user must not have notifications muted for that channel

Success end condition: User is able to mute notifications

Failed end condition: Error condition is met

Primary actor: User within a channel

Trigger: Hitting a 'mute notification' button on the frontend

Validation

To determine the extent to which the use cases described above will adequately describe the problem we are trying to solve, we reached out to the users we originally interview, (Daniel, Aleisha and Gautama), to ask them how satisfied they are with our design.

Daniel

Q - Having read the use cases, would you be satisfied in using these features to solve the mute notifications problem you outlined in our original interview?

Yes, if implemented correctly I am satisfied that these features would be able to let me ignore mentions in channels that I don't want to receive notifications from. This would be extremely

helpful for my job as a reliability engineer, as I could prioritise messages by only receiving tags for the most essential messages, so that the only the most relevant and urgent messages receive immediate attention. I will then be able to sift through the other messages later, to catch up on tags for less immediate cases. However, I am concerned that there is no case to mute notifications from a DM, as I may want to ignore certain DMs

Aleisha

Q - Having read the use cases, would you be satisfied in using these features to solve the entertainment problem you outlined in our original interview?

I think that implementing a hangman function into the channel feature would be an interesting and unique feature that I could use to have a bit of fun with my class. However, I think that the hangman feature may not draw further attention to the messages which I want to convey, but instead distract from the messages. Overall, perhaps a smaller windowed version of the game, or a graphic that is more relevant to the message being sent. However, hangman is an interesting and good concept for me to use with my students.

Gautama

Q - Having read the use cases, would you be satisfied in using these features to solve the set user status problem you outlined in our original interview?

This would be a very helpful solution for me to be able to set a status for other people to view. This would mean that when I am online working with one particular group, I can set my status as away or busy, so that other groups and friends know not to message and bother me with other things. As long as every time I log back in, the status resets to online, I am very happy with this function as it means that I will not accidentally set my status to away, and then forget to change it back at a late date.

Interface Design

Having described the problem requirements and considered the user stories, the below interface describes the backend capabilities to implement the functionality described. These are specified as HTTP endpoints, similar to the existing specification for UNSW Treats.

NOTE - many of the functions also require frontend implementation. This interface design will only handle the backend API calls, rather than the frontend.

See interface design on next page.

Name & Description	HTTP Method	Data Types	Exceptions
/user/setStatus/V1	POST	Body Parameters: { status } Return type if no error: {}	400 Error when user sets status to the current status 403 Error: invalid token 400 Error: status passed in is not one of the available options
/other/hangman/V1	POST	Body Parameters: { channelId } Return type if no error: { finishTime }	403 Error: invalid token 400 Error: hangman game is already playing for user 400 Error: channel does not exist 403 Error: the user is not a member of the channel
/message/muteNotificationsV1	PUT	Body Parameters: { channelId } Return type if no error: {}	403 Error: invalid token 403 Error: the user is not a member of the channel 400 Error: channel does not exist 400 Error: notifications are already muted
/message/unmuteNotificationsV1	PUT	Body Parameters: { channelId } Return type if no error: {}	403 Error: invalid token 403 Error: the user is not a member of the channel 400 Error: channel does not exist 400 Error: notifications are not muted

Conceptual Modelling - State Diagrams

The following diagram indicates ONE of the user journeys a user could potentially take while using UNSW Treats. It shows the chain of operations for a user, depending the input that they give.

