UNSW Beans: User Requirements & Forward Modification Analysis

Authored by Boosted (H13A_BOOST)

Rui Chen, Dominic Cheung, Jessica Gu, Nathan Liu, Raine Wang

Table of Contents

User Elicitation

Concurrent to development of the software solution, 3 target users, henceforth “subjects”, were contacted regarding their usage of a Beans-like tool, as well as providing feedback following a beta testing session of UNSW Beans. The subjects, including names and email addresses, are listed hereinafter.

Subjects’ Details

Name	Email
Emily Ting	espeoncharshire@gmail.com
Isabella Lam	isabella.lam.work@gmail.com
Hyanh Nguyen	hyanh.nguyen2@gmail.com

The aforementioned subjects were presented with several questions, as follows, upon completion of their beta testing session, and their responses recorded. The responses gathered throughout the process are depicted below in an anonymised format to ensure confidentiality of responses, in accordance with principles outlined in the *UNSW Student Privacy Statement* and *UNSW myExperience Survey Procedure*. This further reflects *Principle 6* issued by the UN Statistics Division within the Development of National Statistical Systems, regarding the confidentiality of surveyed data.

1. Would you be able to name a previously utilised communication tool and a relevant valued feature within it, which you feel UNSW Beans may be lacking?
2. How intuitive or user-friendly do you perceive the current user interface of UNSW Beans to be, and what improvements can be made?
3. What do you typically utilise communication tools for, and can you identify any flaws within the use of UNSW Beans in such use case?
4. What other teamwork utilities would you like to see from UNSW Beans?

Subjects’ Responses

Q.	Responses
1	● Discord/Teams - Voice & Video functions ● Discord - Ability to type/send messages without scrolling down, Add users to DMs, Rename channels and DMs ● Ability to delete channels
2	● Could be improved aesthetically (colour palette) ● Mostly straightforward, some badly-named and unintuitive features (Step/Live), some confusing elements (Other channels list) ● Slightly confusing, inability to rename or organise channels prompting confusion,

	self-populated dm is a weird concept, comparable to using teams/zoom for the first time
3	● Talking to friends/about uni work/societies, inability to rename and categorise channels and DMs, inability to send images ● For gaming, lack of voice call functionality ● Communication with friends, work colleagues, common interests
4	● Dark mode, background image/themes (similar to Instagram), embeds, link interactivity ● dark mode, greater default pfp variety, adjustable accessibility features (font size, contrast), Wingdings ● Dark mode, emotes

Upon analysis of recorded responses and collation of relevant information, a solution was proposed to solve the following problems identified:

- Editing channel//DM names
- Move channels/DMs
- Delete channels
- Dark Mode/Other themes
- Voice call functions
- Use of emotes and reactions

The solution is as follows: The implementation of functions to allow the editing and moving of channel and DM names, delete channels, and commence voice calling would enable the addressing of various problems. These would utilise a simple function to alter the database for the first two changes, in addition to slight frontend modifications for buttons to complete such interactions. Through appending frontend changes to modify the thematic colouration of pages within the application, it is possible to allow the use of dark modes as well as other visual themes. The appending of emotes and reactions would require extending of the current message reaction system, combined with alterations for the frontend to allow for the use of other reactions beyond the current thumbs up emoji. On the other hand, the implementation of voice call functions would require the establishment of a voice server to receive and transfer vocal signals, as well as substantial alteration to the frontend and backend to allow for users to interact as required for the transmission of audio.

Analysis & Specifications - Use Cases

User stories were constructed to express the listed requirements for the solutions planned, acceptance criteria were also formed to identify the completion of such alongside use cases, depicting the manner through which accurate function can be verified..

Case	User Story	User Acceptance Criteria	Use Cases
1	As a user,I want to edit channel and DM names so then I can organise my workspace clearly	● A button is placed next to the channel name named “EDIT NAME” ● The inputted name must be between 1 and 20 characters inclusive	1. The user clicks on the “edit name” button next to the channel name 2. A prompt pops up on the users screen asking for a new name 3. If the user inputs a name over 20 characters it will give them an error 4. Otherwise, if successful, the channel will be renamed.
2	As a user,I want to move channels and DMs so then I can organise my workspace clearly	● Upon dragging a channel/DM on the sidebar, it will be relocated to the dropped position ● Dropping the channel/DM outside of the channel/DM subsection of the screen will not move the channel/DM	1. The user creates a new channel 2. The user decides to drag the channel up to it’s related channels 3. The user drags it outside of the channel subsection, causing nothing to change 4. The user drags it to the top of the channel list. 5. The channel moves to the top, reorganising the list of channels.
3	As a user, I want to delete channels cause I might make channels by mistake	● A button is placed next to the leave channel button named “REMOVE CHANNEL” ● A confirmation message will appear upon clicking remove, which a user must also agree with	1. User accidentally creates a new channel 2. User decides to remove channel by pressing the button “remove channel” 3. A confirmation button will pop up, to which the user will either agree or disagree to 4. If the user disagrees, the channel will remain 5. If the user agrees, the channel will be deleted
4	As a user, I want dark mode and themes as I appreciate my eyes not dying from light mode and would appreciate customisability of the application	● A button is placed in the top right hand side corner, right of the logout button to allow users to change between light and dark mode. ● The text “LIGHT MODE” is displayed when in dark mode and the text “DARK MODE” when in light mode	1. User registers, and successfully joins Beans 2. User is blinded by the light mode and thus wants to change the theme 3. The user presses the button “dark mode” in the top right hand side corner 4. The theme changes to dark mode, and the button “dark mode” changes to display the words “light mode”.
5	As a user, I would like voice call functions as I enjoy contacting and interacting with others in a vocal manner	● Within the channel, there will be a button located top right which begins a voice/video call between all members of the channel. This will be the same	1. The user clicks on the connect voice call button in a channel 2. The user is connected to the voice call of the channel

		for DMs. - The text “START CALL” will be displayed while there is no active call in the given channel, this will change to “JOIN CALL” for any unconnected users and “LEAVE CALL” for any connected users	3. The user participates in conversation, but then clicks on the mute button 4. The user is muted 5. A while later, the user comes back and unmutes themselves, rejoining the conversation 6. The user finished their conversation and clicks on the disconnect button in the channel 7. The user is disconnected from the voice call of the channel
6	As a general user, I want to be able to send emotes to friends	- In the send message input box, there should be an emote button next to the send message button - When clicked, the emote button should show an emotes menu, allowing the user to choose from different emotes available - Once an emote has been chosen, the emote should appear in the message box, appended to the current message typed by the user	1. The user wants to send an emote as a reaction to something another user said in the channel 2. The user selects the emote button next to the send message button 3. An emote menu pops up, giving the user many different options 4. The user clicks on the emote, causing it to pop up in the message box. 5. The user adds a message with the emote, and sends it. 6. Both the emote and message pop up into the channel messages.

Validation

Further feedback was acquired from the subjects regarding the extent to which the constructed use cases reflected their initial feedback for problems within the UNSW Beans application. Findings are similarly recorded hereinafter in an anonymised form.

Subject	Validation Feedback
1	The features were mostly wanted and reflected suggested changes from feedback previously given. No further suggestions were made.
2	All features are wanted and do indeed address problems identified with UNSW Beans. Persistence of dark/light modes has been suggested, with all others being fine
3	Most features were wanted, and all were considered beneficial. Including a character limit display was recommended for renaming. Implementation for most features was considered reasonable.

Interface Design

Various capabilities were decided upon the analysis of the requirements and proposed solution. The routes below detail expected inputs and outputs, accompanied by brief explanations and descriptions of key components.

Name & Description	**HTTP Method**	**Data Types**	**Exceptions**

channel/name/edit/v1
Allows the authorised user to edit the name of a channel.
	PUT	Parameters: {
		channelId,
		channelName
		}
		Returns: {}
			channelName can not be empty
			channelName must be between 1-20 characters inclusive
			channelId must belong to a valid channel
			Authorised user must be an owner of the channel

dm/name/edit/v1
Allows the authorised user to edit the name of a DM.
	PUT	Parameters: {
		dmId,
		dmName
		}
		Returns: {}
			dmName can not be empty
			dmName must be between 1-20 characters inclusive
			dmId must belong to a valid DM

channel/remove/v1
Removes an existing channel with the given channelId. A confirmation prompt is given to the user
	DELETE	Parameters: {
		channelId
		}
		Returns: {}
			Authorised user must be an owner of the channel
			channelId must belong to a valid channel

dm/invite/v1
Invites a user of the given uId to the specified dm. If successful, the invited user automatically joins.
	POST	Parameters: {
		dmId,
		uId
		}
		Returns: {}
			dmId must belong to a valid DM
			uId must belong to a valid user
			The user being invited must not already be in the DM

channel/rearrange/v1
Allows the user to rearrange their list of joined channels.
	PUT	Parameters: {
		channelId,
		position
		}
		Returns: {channelsOrder}
			channelId must belong to a valid channel
			channelId must belong to a channel that the authorised user is a member of
			position must be a valid index of the list of the channels the authorised user is a member of

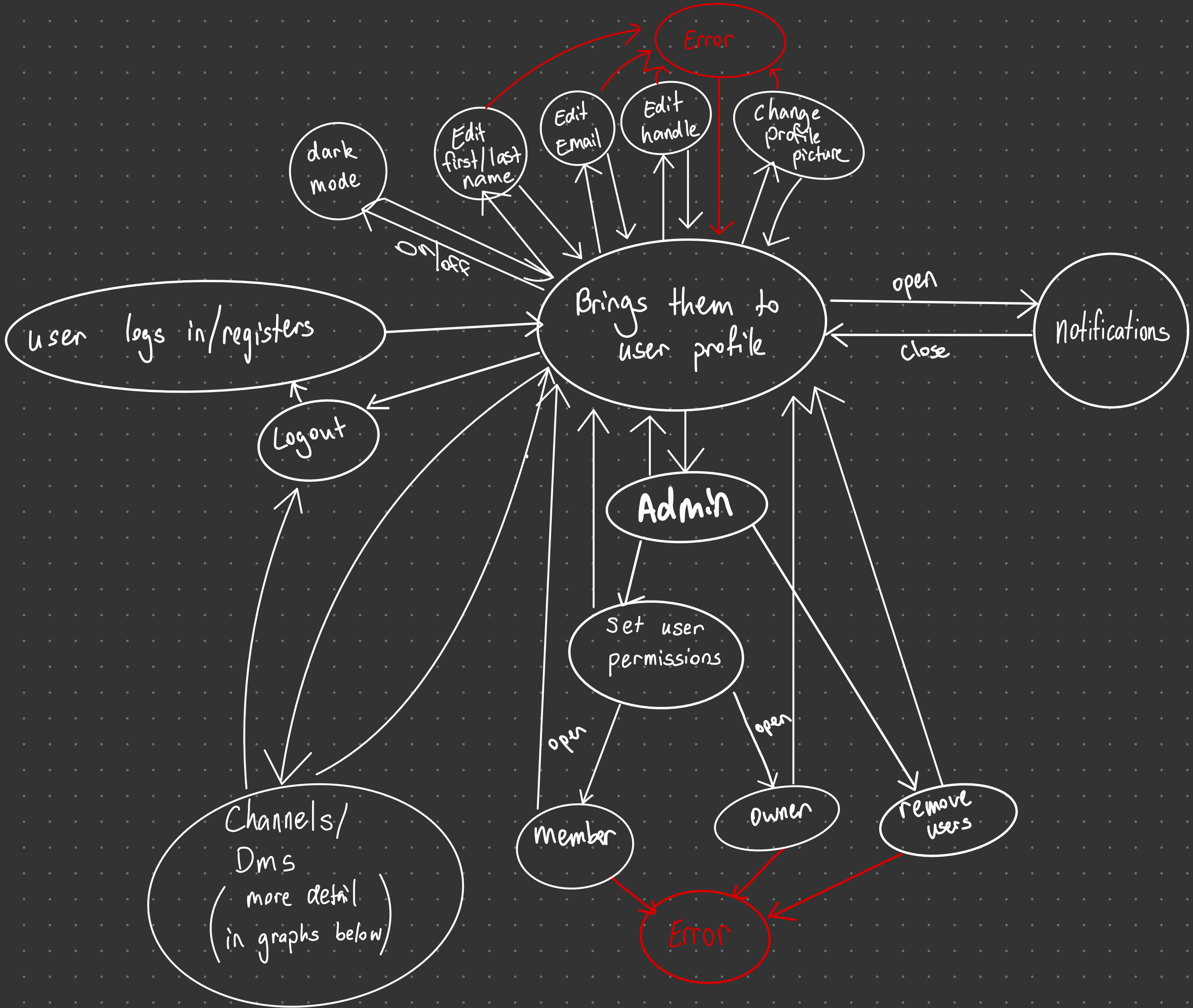
dm/rearrange/v1
Allows the user to rearrange their list of dms.
	PUT	Parameters: {
		dmIId,
		position
		}
		
		Returns: {dmsOrder}
			dmId must belong to a valid DM that the authorised user is a member of
			Position must be a valid index of the list of dms the authorised user is a member of

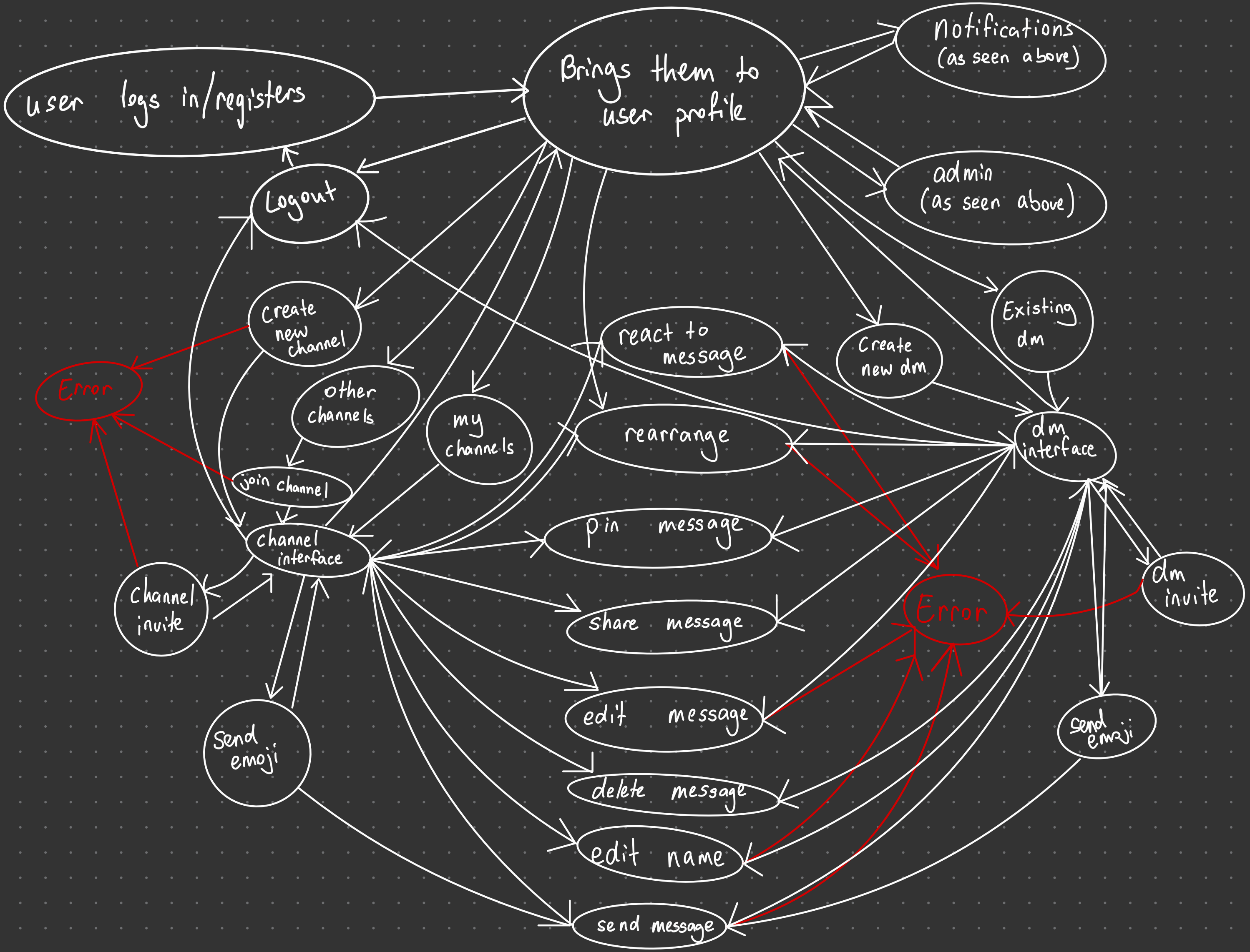
message/sendemoji/v1
Allows the user to pick and send an emoji from a provided range of emojis to the channel specified by the given channelId. Emojis can be removed but not edited
	POST	Parameters: {
		channelId,
		emojiId
		}
		
		Returns: { messageId }
			channelId must belong to a valid channel
			emojiId must be a valid emoji on the frontend

dm/create/v3
Creates a new DM including the users that correspond to the given uIds. The authorised user becomes the owner of the DM.
	POST	Parameters: {
		uIds
		}
		
		Returns: { dmId }
			All given uIds must belong to a valid user
			There should be no duplicate uIds given
			NEW
			Users should not be able to create a self-populated DM

Conceptual Modelling - State Diagrams

The following diagram(s) depict the necessary changes regarding the states of the application upon


Error
dark mode
Edit first/last name
Edit Email
Edit handle
Change profile picture
On/off
user logs in/registers
Brings them to user profile
open
close
Notifications
Logout
Admin
Set user permissions
open
open
Channels/ Dms
(more detail in graphs below)
Member
Owner
remove users
Error


user logs in/registers
Brings them to user profile
notifications (as seen above)
admin (as seen above)
Logout
Create new channel
Error
other channels
my channels
join channel
channel interface
Channel invite
Send emoji
react to message
rearrange
pin message
share message
edit message
delete message
edit name
send message
Create new dm
Existing dm
dm interface
dm invite
Error
send emoji