1   REQUIREMENTS

1.1   ELICITATION

For the elicitation phase of requirements, the following general question set (see Figure 1) was developed with the intention of uncovering what problems they currently face with existing teamwork-driven communication tools. These were followed up with additional questions if more elaboration was desired.

> 1. *State your name and email please*
> 2. *Have you ever used a teamwork communication tool such as Microsoft Teams, Slack, or Google Workspace?*
> 3. *What did you use it for/what was the reason you needed to use it?*
> 4. *Take me through your experience with (or one of) the application, what are any issues you encountered while using it?*
> 5. *Were there any existing features that did not work as intended?*
> 6. *Were there any features you thought of that did not exist in the application that would have been useful to you?*
> 7. *Would you use the same tool again? If not, what is stopping you?*

Figure 1: Interview Question Set

The users interviewed were either current university students or employed in workplaces where team collaboration is ubiquitous. Furthermore, they were selected based on previous experience with existing teamwork communication tools. The interview responses are documented below in Table 1, and

Table 2.

Table 1: Response 1 - User: Manik

Q	Response
1	Name: Manik Bagga Email: manikbgg@gmail.com
2	I have used Microsoft teams and slack
3	I used them in various workplaces to communicate with my team and others within my organisation
4	Microsoft Teams is the primary communications platform used for my organisation. Almost all communications from the organisation comes through the Teams functionality or the various group chats that can be created. Bugs do occur from time to time such as high latency in calls, misregistration of I/O devices and sometimes options on messages are overlapped by the border around the chat area
5	Sharing word documents in a group chat for my organisation has been broken since implementation. Changes to a document do not appear for every user
6	The ability to launch polls within a group chat would be a feature that is useful to add
7	Yes, I would use the tool again in a professional setting

Table 2: Response 2 – User: Iona

Q	Response
1	Name: Iona Quinn Email: ionaq99@gmail.com
2	I have used Microsoft Teams.
3	I used it as a university student (primary education) for group assignments.
4	Used it to post collaborative documents and files for a group presentation.
5	Most of the features I used worked as intended. I was not able to use the webcam background features for calls as my laptop was not good enough.
6	I would love some sort of way to add others as friends to make direct messaging easier.
7	Yes, if the assignment or project would require it.

Table 3: Response 3 - User: Jared

Q	Response
1	Name: Jared Mateo Email: jared.mateo@student.unsw.edu.au
2	I have not used slack or google workspace before, but I've used Microsoft Teams.
3	Had to use it for some Uni lectures and as the main communication tool throughout a group project.
4	The overall experience was average I would say, there weren't many personalisation options. Maybe a few more themes would have been nice and the ability to change the layout of the chats or client would have been nice too, as personally I did not like the layout. I like the fact that you could have spotlighted messages that itself could have its own thread of messages underneath. This was a nice touch as you were able to organise topics and their discussions well.
5	All the features that I was exposed to worked as intended except for maybe one part of the application. They had an activity tab where you could see all the messages you missed or if someone mentioned your name. This in hindsight would have been useful but in my opinion, it was hard to navigate, confusing and thus not useful at all. Also, sometimes when you would click an item in the activity tab, it would fail to bring you to the item.
6	In terms of this application, maybe the ability to store the topics, key messages, or discussions that I will need to use or look back on in the future would be nice. Apart from that, just fixing the activity tab and adding more personalisation options to the application.
7	Maybe on some occasions I would use it as I like the organisation of messages and topics when communicating but other than that, I personally do not like the layout and look of it so for my primary chatting tool, no. But for projects and work-related things I would be more inclined to use it.

From the conducted interviews, most of the responses refer to Microsoft Teams with only one user having experience with slack. The main problem encountered is a lack of customisation to the user, and bugs surrounding important productivity features such as document sharing and the activity feed. A proposed solution is to implement some of these functionality ideas and minimize bugs to create a tool with increased flexibility and reliability than the competition. These would include:

- Customisation in activity feed, application theme, and key message storage/display.

- A 'friends list' or contacts feature to allow for more efficient communication between frequent collaborators
- The ability to launch polls within channels for team decisions.

1.2 ANALYSIS & SPECIFICATION – USE CASES

The following user stories were generated to summarise the responses seen in 1.1:

1. As a collaborator, I want an efficient method of direct messaging users I frequently interact with so that I can save time communicating with specific users.
2. As a collaborator, I want tools that make democratic decision making easier so that everyone in my team can communicate their thoughts.
3. As a collaborator, I want to personalise the application interface so that it is more suited to my needs, and style of work and I am more motivated to use it.

These user stories were broken down into criteria to clearly indicate whether the problems have been solved or not. These can be seen in Table 4 below. These were further elaborated upon in Table 5 and Table 6 by generating use cases.

Table 4: Acceptance Criteria

User Story	Acceptance Criteria
1	- Contacts are always listed somewhere easily accessible to the user - When selected, a list of the user's contacts are displayed to the user - The user can update and edit these contacts in some way - The user can search through these contacts using the search field around the top of the page - The user can only search for handles that are at least 2 characters in length
2	- The has the option to include a democratic decision-making item (a poll/vote) when posting a channel message. - This user does this by selecting an icon which includes this item in the message - Users in the channel can select one option they desire or rank their choices from 1 to n - Options cannot have a character length of greater than 100 characters - The user can set a due time following which votes can no longer be cast
3	- The user can select a colour theme from pre-defined colour palettes by clicking an icon and selecting from a drop-down menu or similar. - The user can specify what is to be shown in their activity/notifications feed from the following options: Stand-ups, Tagged messages, Direct messages, Channel Messages, Channel leaves, Channel Joins, Channel Invites, and Channel add/remove owners - The user can save posts by selection and access these at a later time in a separate tab.

Table 5: Use Cases (Background)

	USE CASES (Background)	
#	**Criteria**	
1	Use case	Organises user's existing DMs
	Goal in context	User should be able to organise a list of DMs they currently are a member of
	Scope	Seams App
	Level	Sub-function
	Preconditions	User is already registered to Seams
	Success end condition	User is a member of at least one DM
	Failed end condition	User currently is not a member of any DMs
	Primary actor	Collaborator
	Trigger	User selects "prioritise" option to organise DMs
2	Use case	Creates poll in a DM/channel
	Goal in context	User should be able to create a poll in a DM/channel where other members can vote in
	Scope	Seams App
	Level	Sub-function
	Preconditions	User is already registered to Seams
	Success end conditions	User is a member of the channel
	Failed end conditions	User is not a member of the channel
	Primary actor	Collaborator
	Trigger	User creates a message on a channel and selects the "poll" option
3	Use case	Changes the appearance of Seams interface
	Goal in context	User should be able customise the colour palette of the Seams interface and customise the activity/notifications feed shown to them
	Scope	Seams App
	Level	Sub-function
	Preconditions	User is already registered to Seams
	Success end conditions	User is logged in and is viewing their user profile
	Failed end conditions	User is not logged in or viewing their user profile
	Primary actor	Collaborator
	Trigger	User selects "personalise" icon on their user profile

USE CASES (Scenarios)
Use Case #1 Main Success Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App checks that email and password are stored in the Seams database Step 4: App retrieves which DMs/channels the user currently is a member of from the database Step 4: User selects option to prioritise DMs Step 5: App retrieves which DMs the user currently is a member of from the database Step 6: User ranks their DMs Step 7: App updates priority list of user's DMs stored on the database Step 8: App returns the list of DMs ranked according to priority Possible Failure Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App checks that email and password are stored in the Seams database Step 4: App retrieves which DMs/channels the user currently is a member of from the database Step 5: User selects option to prioritise DMs Step 6: App cannot retrieve any DMs where the member is a user from the database
Use Case #2 Main Success Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App checks that email and password are stored in the Seams database Step 4: App retrieves which channels the user currently is a member of from the database Step 5: User selects a particular channel Step 6: User creates a message and selects the "poll" option Step 7: App creates a message with a poll and updates the database Possible Failure Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App checks that email and password are stored in the Seams database Step 4: App cannot retrieve any channels where the user currently is a member of from the database
Use Case #3 Main Success Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App checks that email and password are stored in the Seams database Step 4: App retrieves which DMs/channels the user currently is a member of from the database Step 5: User selects option to "personalise" the main interface Step 6: App returns which options to personalise interface are available to the user Step 7: User selects their preferred options Step 6: App updates interface according to the user's preferences

Possible Failure Scenario: Step 1: App prompts user to login Step 2: User logs into Seams using the email and password they registered with Step 3: App cannot retrieve the user's email or password from the database	

1.3 VALIDATION

To validate our approach, the original interview subjects were asked to comment on the generated use cases and whether they encapsulate the problems they encounter currently. These can be seen in Table 7, where the consensus is that the use cases provide good coverage for the problems identified.

Table 7: Use Case Feedback

User	Feedback
Jared	"The use cases given back to me, if implemented properly, would be sufficient enough to solve my problems of having little personalisation options for the interface."
Manik	"The concept presented by the seams team adequately identifies and rectifies a glaring issue in the current Microsoft Teams app. This concept promotes ease of use for the user by providing a streamlined and simple process to create a poll"
Iona	"The use-case that has been developed encapsulates my problem but framed in a different way than I originally envisioned. I imagined a friends list feature, but my problem was really to find a way to sort dms I wanted to access quickly."

2 DESIGN

2.1 INTERFACE DESIGN

Table 8 and Table 9 specify the necessary capabilities for our solutions.

Table 8: Input/Output Types

Variable Name	Type
token	String
colour_palette	Integer between 0 and 4 (inclusive).
activity_filter	An array of strings with selections of what the user wishes to display from the following strings: Stand-ups, Tagged messages, Direct messages, Channel Messages, Channel leaves, Channel Joins, Channel Invites, and Channel add/remove owners By default activity filter contains all options.
filter	Boolean

poll	Dictionary of shape {
	    poll_id: integer,
	    poll_title: string,
	    poll_options: [strings],
	    poll_votes: [integers]
	    time_finish: integer (unix timestamp)
	}

poll_id	Integer
polls	List of type poll

poll_option	String
dm_priority	Integer
For all other Input/Output types refer to Table 6.1 in Iteration 3 readme.md	

Table 9: Interface Table

Name & Description	HTTP Method	Data Types	Exceptions
user/interface/colour/set/v1 Set an interface colour palette preference for a user to be used when generating the front-end UI. The colour palette preferences are pre-selected by the dev team and are optioned 0 – 4.	POST	**Parameters:** { token, colour_palette } **Return Type:** { }	**InputError** when: • colour_palette does not refer to a valid selection (not between 0 and 4)
notifications/prefs/v1 Set what types of activity are to be displayed in a user's filtered notifications. Selections can be made from the following options: Stand-ups, Tagged messages, Direct messages, Channel Messages, Channel leaves, Channel Joins, Channel Invites, and Channel add/remove owners	POST	**Parameters:** { token, activity_filter } **Return Type:** { }	**InputError** when: • An invalid selection is found in activity_filter
notifications/get/v2 Returns the user's most recent 20 notifications, ordered from most recent to least recent. If filter is true, only return notifications related to activity set in notifications/prefs/v1	GET	**Parameters:** { token , filter } **Return Type:** { notifications }	N/A
channel/polls/start Send a poll from the authorised user to the channel specified by channel_id. Note: Each poll should have its own unique ID, i.e. no polls should share an ID with another message, even if that other message is in a different channel.	POST	**Parameters:** { token, channel_id, poll, time_finish } **Return Type:** { poll_id }	**InputError** when: • channel_id does not refer to a valid channel **AccessError** when: • channel_id is valid and the authorised user is not a member of the channel
channel/polls/view	GET	**Parameters:**	**InputError** when:

Given a channel with ID channel_id that the authorised user is a member of, return up to 50 polls between index "start" and "start + 50". poll with index 0 is the most recent poll in the channel. This function returns a new index "end" which is the value of "start + 50", or, if this function has returned the least recent polls in the channel, returns -1 in "end" to indicate there are no more polls to load after this return.		{ token, channel_id, start } **Return Type:** { polls, start, end }	• channel_id does not refer to a valid channel • start is greater than the total number of polls in the channel **AccessError** when: • channel_id is valid and the authorised user is not a member of the channel
message/poll/vote/v1 Given a channel ID authorised user is a part of and an active poll, cast user vote on selection.	POST	**Parameters**: { token, channel_id, poll_id, poll_option } **Return Type**: { poll_votes }	**InputError** when: • poll_id does not refer to an active poll • channel_id does not refer to a valid channel • poll_option does not refer to a valid option in the poll **AccessError** when: • channel_id is valid and the authorised user is not a member of the channel
dm/list/v2 Returns the list of DM's that the user is a member of based on priority.	GET	**Parameters**: { token } **Return Type**: { dms }	**InputError** when: • dm_id does not refer to a valid DM **AccessError** when: • dm_id is valid and the authorised user is not the original DM creator • dm_id is valid and the

			authorised user is no longer in the DM
dm/setPriority/v1 Set priority of dm specified in dm_id to dm_priority. Higher priority numbers will be displayed first.	POST	**Parameters**: { token, dm_id, dm_priority } **Return Type**: { }	**InputError** when: • dm_id does not refer to a valid dm **AccessError** when: • dm_id is valid and the authorised user is not a member of the dm

2.2 CONCEPTUAL MODELLING (STATE)

To aid in developer understanding, state model diagrams were developed for each of the use cases. They can be seen in Figure 2 below.
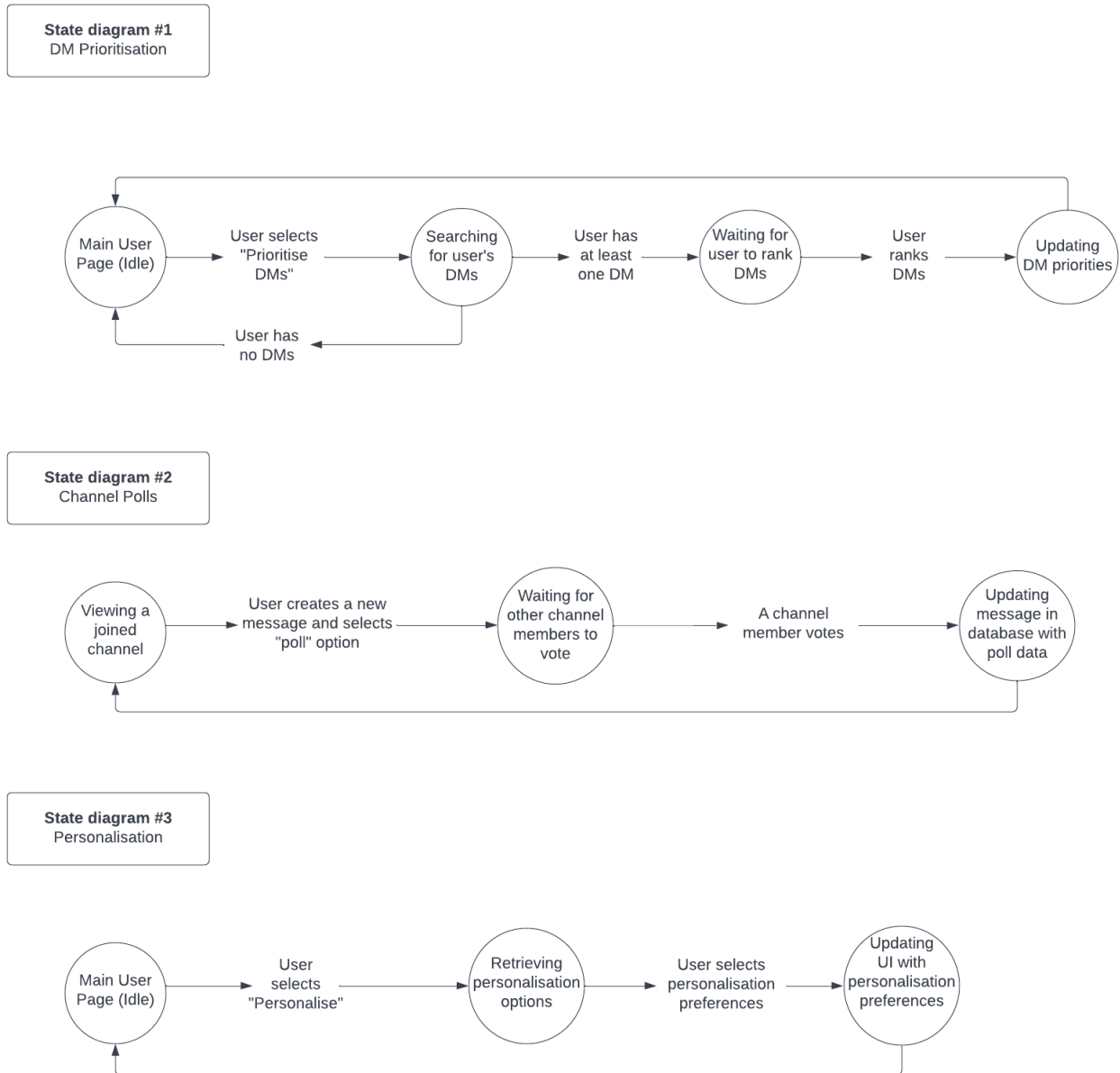

Figure 2: State Diagrams