

Stage 1 Elicitation

Interview One:

Prashanna K C, email: prashanna.kc08@gmail.com

Question: What function do you think would be a good addition to the UNSW meme?

Answer: A contact list of everyone who is in the same channels and direct messages as me, a history record of all the messages that I have sent, and a to-do list of the things I need to do to stay organised.

Question: How do you think the functionality of a to-do list can integrate with the existing features that we have already implemented?

Answer: I think that because all the features are based around team communication, such as channels and direct messages, a to-do list is a good way to integrate. One way to do this is to have a to-do list for the entire channel where members can collaborate and update the list as they go.

Question: What do you think would be the main benefits of having an extra to-do list feature?

Answer: As I mentioned before, having a collaborative to-do list would allow the team to keep track of what needs to be done in real-time as other members update the list. This would be helpful in a teamwork situation where we have a set deadline for the group, and we have broken up the big tasks into smaller chunks. Having a to-do list would allow us to see what still needs to be done, and if members need more help.

Question: You also mentioned a contact list feature. What do you think will be the main benefits of having a contact list?

Answer: I think a contact list will be good because it allows me to see very clearly the list of people that I have potential connections with. If I ever need assistance with anything, it also gives me an idea of the range of people that I can reach out to.

Question: The feature I am not particularly sure about is the history of all the messages sent because they are already being kept track of in each channel and direct message. What would be the additional benefit of having it all collated centrally?

Answer: When I first thought of it, having everything centralised would mean easier access, and I could better keep track of everything I have said in all my different direct messages and channels. But now that you mention it, maybe it is less useful than the other features.

Interview Two:

Shizun Han (Austin), shizun0321@gmail.com

Question: What function do you think would be a good addition to the UNSW meme?

Answer: A calendar that shows all the upcoming events and all the deadlines would be beneficial since it could keep me more organised and productive.

Question: How do you think the functionality of a to-do list can integrate with the existing features that we have already implemented?

Answer: I think the to-do list should be integrated with the group channel and allow all the users of the channel to edit it so we can set goals as a group and cross them off as we progress.

Question: What do you think would be the main benefits of having an extra to-do list feature?

Answer: A to-do list will keep the group more organised since everyone in the group can see what needs to be done and when the deadline is. This will in turn increase productivity as well as encourage better time management. Also, a to-do list can help break down bigger tasks into smaller sections and assigning the sections to members will be easier as well.

Stage 2 Analysis & Specification

1. As a student, I want to be able to see all the users who are a part of the same DM and channels as me in a concise list, so that I have a list of the group of people that I have connections with.

in the Meme home page, there is a section where the user can access basic information about the users that are in the same channel/dm as them

the information listed for each user connection should include name, email and what channel/dm they have in common with user

if user joins/leaves channel/dm this information should be updated automatically

2. As a student I want to be able to keep track of all the tasks and progress of my group in the form of a to do list so that as a group we can better plan out any group projects

every member in channel/dm has authority to add fields to to-do list

every member in channel/dm has authority to tick off fields in to-do list

all ticked off to-do's can still be viewed and be added back onto the to-do list

to-do's can be assigned to any member in the channel/dm

each to-do should contain information about the task and the users it is assigned to

members of channel/dm get a notification when to-do is ticked off

Use Case – create to-do list

Scope: UNSW Meme

Preconditions: user/s have registered, logged in and channel/dm has been created

Success End Condition: to-do list created and visible to all channel/dm members

Failed End Condition: user not a part of any channel/dm and cannot create a to-do list

Primary Actor: User (channel/dm member)

Trigger: user selects option to create to-do list and inputs required information

Stage 3 Validation

Validation Two (Prashanna):

Question: Does the use case outlined above adequately implement the to-do list?

Answer: It looks pretty good to me, as it covers the basic requirement of users in channels/dms being able to create a to-do list for all member to collaborate on, and all the functions of a to do list are included. A potential addition to the list could be being able to delete entire lists instead of manually ticking off every box.

Validation Two (Austin):

Question: Does the use case outlined above adequately implement the to-do list?

Answer: I think it does since the main goal for the to-do list is to implement it inside the channel to keep everyone organised. And I think the use case can work as an implementation for creating and crossing off to-do list.

Stage 4 Interface Design

Input/Output type

Variable Name	Type
Title	string
ChannelId/dmId	number
assignTo	string[], where each string is specifically a user handleStr
description	string
todoListId	number
todold	number
TodoList (output)	{title: string, todo[]}
ToDo (output)	{description: string, createdBy: uid, assignedTo}

Interface – user token is passed through the header

Name & Description	HTTP Method	Data Types	Exceptions
Todo/create Creates a new empty to-do list with input title. todoListId are unique across all channels/DMs.	POST	Body parameters: (title, channelId/dmId) Return type if no error: { todoListId }	400 Error when: length of title is less than 1 or more than 50 characters ChannelId/dmId is invalid 403 Error when ChannelId/dmId is valid but user is not a part of said channel/dm
Todo/add Adds a new to-do to the to-do list. Returns a todold – not unique between to-do lists	POST	Body parameters: (todoListId, assignTo, description) Return type if no error: {todold}	400 Error when: length of description is less than 1 or more than 500 characters AssignTo array is empty

			<p>TodoListId does not refer to valid to-do List</p> <p>403 Error when: TodoListId is valid but user is not a member of channel/dm that todolist is connected to</p>
Todo/remove Removes an uncompleted to-do, removed to-dos cannot be recovered. Completed to-dos cannot be removed, removed todolds cannot be reused	DELETE	Body parameters: (todoListId, todold) Return type if no error: {}	<p>400 Error when: TodoListId does not refer to valid to-do List Todold is invalid for given to-do list</p> <p>403 Error when: Todold is for a completed todo TodoListId is valid but user is not a member of channel/dm that todolist is connected to</p>
Todo/revive Revives a completed to-do, revived to-do will reappear on the list view, reviving does not change createdBy in todo	POST	Body parameters: (todoListId, todold) Return type if no error: {}	<p>400 Error when: TodoListId does not refer to valid to-do List Todold is invalid for given to-do list</p> <p>403 Error when: Todold is for an uncompleted to-do TodoListId is valid but user is not a member of channel/dm that todolist is connected to</p>
Todo/list Returns information about a to-do list in the channel – a list of uncompleted to-dos	GET	Body parameters: (todoListId) Return type if no error: { todoList }	<p>400 Error when: TodoListId does not refer to valid to-do List</p> <p>403 Error when: TodoListId is valid but user is not a member of channel/dm that todolist is connected to</p>
Todo/edit Edits an existing to-do, only uncompleted to-do's can be edited	PUSH	Body parameters: (todoListId, todold, assignTo, description) Return type if no error: {}	<p>400 Error when: length of description is less than 1 or more than 500 characters AssignTo array is empty TodoListId does not refer to valid to-do List</p>

			<p>Todold is invalid for given to-do list</p> <p>403 Error when: TodoListId is valid but user is not a member of channel/dm that todolist is connected to Todold is for a completed todo</p>
Todo/complete Ticks off completed to-do, if all the to-dos are ticked off the entire list is automatically removed from view. Single completed to-dos will be removed from list view once they are completed	PUSH	Body parameters: (todoListId, todold) Return type if no error: {}	<p>400 Error when: TodolistId does not refer to valid to-do List Todold is invalid for given to-do list</p> <p>403 Error when: Todold is for a completed todo TodoListId is valid but user is not a member of channel/dm that todolist is connected to</p>
Todo/delete Deletes entire to-do list even if not all to-dos are completed. Only owner of channel or DM has authority to delete. Deleted todoListId's cannot be reused.	DELETE	Body parameters: (todoListId) Return type if no error: {}	<p>400 Error when: TodolistId does not refer to valid to-do List</p> <p>403 Error when: User is not a channel/dm owner</p>

Stage 5 State Diagram:

