Elicitation

User 1:
Name and email address:

      Wanny Sun, z5324152@ad.unsw.edu.au

Problems:

Since channel names are decided by the owners, he can't keep track of them, and sometimes names are repeated, which makes it hard to use Teams.

Solutions:
1. Add a function that can modify channel names for only the user who calls it

User 2:
Name and email address:

      Kevinnesan VP, z5275718@ad.unsw.edu.au

Problems:

Has bad wifi and so cannot always read through the messages in a chat. He'd like to be able to read previous messages even without wifi.

Solutions:
1. Add a function that can download a chat so that the user can read through messages without wifi

User 3:
Name and email address:

      Peter Alexander, z5368178@ad.unsw.edu.au

Problems:

Has many chats and so is unable to find a particular one, making it harder to work in groups

Solutions:

1. Add a searching function on the channels interface:

      When users enter "channel: <keyword>", return a list of channels with the name of the channels including the keyword.

      When users enter "user: <keyword>", return a list of channels where the name of users in the channel including the keyword.

When users enter "message: <keyword>", return a list of channels where the message in the channel including the keyword.

Analysis and Specifications - Use Cases

User 1:
User Story:
As a teamwork app user (uni student), I want to be able to change channel names, to make managing my groups easier.

Acceptance Criteria:
- Change channel names for myself
- Can see the original channel name

Use Case:
- User selects channel to change name of
- User enters name they want to see instead
- Channel name is changed to the new name the user entered for the user
- Original channel name can still be checked

User 2:
User Story:
As a teamwork app user (uni student), I want to be able to look through previous messages without needing wifi, to make work faster.

Acceptance Criteria:
- Download chat messages automatically
- Turn off automatic message downloading

Use Case:
- User selects option to automatically/manually download chat messages
- Messages are downloaded for offline access
- User views previous messages when offline
- User unselects option to automatically download messages
- Messages stop being downloaded automatically

User 3:
User Story:
As a teamwork app user (student), I want to be able to find specific chats, so I can work faster and more efficiently.

Acceptance Criteria:

- Search individual chats by name, users or messages

Use Case:
- User searches for a chat using the name, users or messages in the search bar
- List of chats with that name, users or messages is returned
- User accesses the necessary chat

Validation

User 1:
What extent:
This use case would perfectly solve my problems because I would be able to much better organise my groups and channels without getting confused between the different assignments or subjects.

User 2:
What extent:
This use case covers most of my problems but I want to also be able to choose up till which message gets downloaded as often my chats have a lot of messages and these might end up taking up too much space on my computer.

User 3:
What extent:
This use case would cover all the problems I'm currently facing and make it easier for me to access specific chats as well as find information from particular chats when I don't remember what it was called.

Validation

chats - a dictionary containing **channels** and **dms**

Name & Description	HTTP Method	Data Types	Exceptions
channel/setname/v1 Update the channel name for the authorised user	PUT	**Parameters:** { token, channel_id, channel_name } **Return Type:** {}	**InputError** when: ● length of channel name is not between 1 and 50 characters

			inclusive
message/download/chat/v1 Download the messages from a channel or dm	GET	**Parameters:** { token, chat_id} **Return Type:** {messages}	**InputError** when: ● chat_id does not refer to a valid channel/dm **AccessError** when: ● chat_id is valid and the authorised user is not a member of the channel/dm
search/chat/v1 Find all the channels/dms the authorised user is a part of with a particular channel/dm name, some specific users, or a particular message	GET	**Parameters:** { token, channel_name / users / message} **Return Type:** {chats}	N/A

Validation

User 1: