



# UNSW

## Sydney

**COMP1531: Software Engineering Fundamentals**

**Term 3– 2020**

**5.3. Planning for the next problems to solve planning doc**

*Sienna Archer, z5208759*

*Raymond Mahadana Kawiswara, z5240051*

*Karan Mehta, z5309394*

*Madeline Younes, z5208494*

## Table of Contents

[Requirements] Elicitation: Interviews .....	1
Interview 1: .....	1
Interview 2: .....	2
Interview 3: .....	3
[Requirements] Analysis & Specification/Validation – Use cases.....	4
Tagging / Mentioning (From interview 1): .....	4
Accessibility (From interview 2):.....	5
Custom reacts (From interview 3):.....	6
Private Messaging (From interview 3):.....	8
[Design] Interface Design .....	10
[Design] Conceptual Modelling (State).....	12
Custom React .....	12
Private Messaging.....	12
Accessibility.....	12

# [Requirements] Elicitation: Interviews

## Interview 1:

**What is your email address and name?**

**Name:** Gabrielle Younes

**Email:** [gmyounes@iinet.net.au](mailto:gmyounes@iinet.net.au)

**What are the interviewees currently using?**

Slack

**What is the purpose of using the current software?**

For all society related discussion (WIT). Which includes messaging on channels, individual messaging and using shout out channel to make announcements.

**What do they like / dislike about the current app that they are using?**

- Likes
- channels, the ability to have different areas for different things,
- tagging and reacting to posts
- Dislike
- Slack bot

**Have you wanted to do something which was not possible due to the limitations of the software?**

No

**Are there any features that you would like to be added into the app?**

By adding tagging.

## Interview 2:

**What is your email address and name?**

**Name:** Isabelle Gullotta

**Email:** [gullotta8@gmail.com](mailto:gullotta8@gmail.com)

**What are the interviewees currently using?**

Discord.

What is the purpose of using the current software?

Text + audio chat, used in both gaming and social contexts.

**What do they like / dislike about the current app that they are using? Have you wanted to do something which was not possible due to the limitations of the software?**

I like that it's easy to set up friends and servers, the ping function for getting in touch with particular groups of people, ability to stream video and share screen, and integration of bots to increase the features of discord (e.g. text based game bots, music bots). I also like the degree of privacy you can enable on servers so that people cannot directly contact you.

I dislike that certain functionalities are reliant on bots (like music) when it is so widely used that it would be better to have consistently available methods integrated into discord by discord. I also dislike sharing screen from a Mac as it doesn't (as far as I know) send the audio directly from the computer (this functionality is available on windows i believe. I also would like this to be possible in the future). Maybe having some drawing abilities in app could be cool? Similar to drawing based games. Not a must though.

**Are there any features that you would like to be added into the app?**

Oooo sound like good features! Particularly the timer messages because idk if that's available in the absence of bots. Maybe it'd be good as a non-admin to have the ability to add people to the server (assuming admins allow it)? Does it have any audio based voice type stuff or is it all intended as text? Can user to user messages be sent? Maybe some accessibility stuff would be good, in terms of text to audio functionality, font size able to be adjusted, contrasted text, etc. Just chucking some ideas lol, use whatever you want.

## Interview 3:

**What is your email address and name?**

**Name:** Jack Spiggle

**Email:** jspiggle@hotmail.com

**What are the interviewees currently using?**

Slack, teams, messenger

**What is the purpose of using the current software?**

Because of the high user base, aka everyone else is using them, mandated by university

**What do they like / dislike about the current app that they are using?**

- Likes the layout of teams, dislikes that important messages get lost
- Only good thing about messenger is that it is simple

**Have you wanted to do something which was not possible due to the limitations of the software?**

- On teams you cannot pin messages which can be annoying and would solve the above problem
- On teams you also cannot set reminders which would be helpful

**Are there any features that you would like to be added into the app?**

- Private messaging
- More than one type of react/ custom reacts
- Calendar integration
- Setting nicknames only in the channel

# [Requirements] Use cases

## Tagging / Mentioning (From interview 1):

### User Story:

As a general, group focused user, I would like to be able to tag or mention users in posts so that they can be quickly notified of the post and get back to me as soon as possible.

### UATS:

- There is a special character sequence to tag or mention a user by their username in message fields.
- An email is sent to users who are tagged or mentioned
- Users should have response times to their posts shortened

### Use Case:

- Use Case: Tagging a user
- Goal in context: User needs to be sent an email when they have been tagged
- Scope: Flockr app and user's email
- Level: Subfunction
- Preconditions: Both users are registered and are members of the channel where the message is posted
- Success end condition: The User being tagged receives an email and there is an indication on the message that they have been tagged in
- Failed end condition: User2 ignores User1
- Primary Actor: User1, User2
- Trigger: User1 tags user 2

### Main success scenario:

- User1 writes a post, in which they tag User2
- Backend recognises that User2 has been tagged and sends an email to User2, informing them of who has tagged them, and on what channel
- User2 receives this email and using tremendous willpower, removes himself from his bedsheets and opens flockr.
- User2 sees an icon indicating the message in which they were tagged and responds
- User1 is happy

### User response:

Perfect, that seems to describe the tagging capability I was talking about.

## Accessibility (From interview 2):

### User Story:

As a gamer / general social user, I would like for there to be accessibility options available, so that if needed, the application can help getting visual information to the user clearly.

### UATS:

- There should be an option to open accessibility settings with an obvious, clear icon, near the logout button
- This menu should have options to zoom in, enlarge text, read text aloud, move buttons further apart, and change other graphic settings
- Users should have an easier time reading information on the screen and accessing different parts of the application

### Use Case:

- Use Case: Using accessibility options to get information from the page easier
- Goal in context: User needs to feel that information is easier to see using the accessibility options
- Scope: Flockr application frontend / backend
- Level: New feature set.
- Preconditions: User is logged into Flockr app
- Success End Condition: User finds it easier to navigate the app and read from the screen when using the accessibility options
- Failed end condition: User finds there to be no difference OR user finds that visual elements become more unclear OR user finds that though it does help, the options for such a menu are too cluttered and time-consuming to use properly
- Primary actor: User
- Trigger: User clicks on the accessibility menu

### Main Success Scenario:

- User logs on to the application and finds that the list of channels is difficult for them to read
- User clicks the accessibility icon, and a dropdown menu appears with 5 options.
- User selects the option with an icon depicting a small capital 'A' turning into a large capital 'A'
- A pop-up window appears with a slider bar allowing for 5 different text sizes greater than the current and user moves it up two points.
- Backend registers this information, and passes it to the frontend, which adjusts its font sizes across the application

- User notices that the colour contrast between the message field and the background is too low, and they usually have to click a few times to hit the message field. They open the accessibility menu again
- User selects icon depicting a square with half of it coloured light grey and the other half, a much more prominent black.
- Backend registers this and sends information to the frontend to switch the colour scheme to have more contrast between input fields and whitespace
- User opens menu again, and selects text to audio option, then clicks a message or channel name (any text) and the backend calls a function to read out the text.
- User opens menu again and selects magnifying glass option, then hovers the mouse over the screen, with a 100-pixel radius around it being zoomed in on.
- User opens menu again and clicks icon with two boxes and an arrow between them. Backend registers this and sends instructions to frontend to move all selectable fields 50 pixels away from one another on all sides
- User is satisfied with the simplicity and ease of use of this menu.

#### **User response:**

The accessibility features sound like they are easy to use and would be super helpful. I approve of this implementation.

## Custom reacts (From interview 3):

#### **User Story:**

As a frequent user of social networking tools, I would like to be able to create and use our own custom reacts (emotes) so that I can express my responses better and in a funnier manner.

#### **UATS:**

- The field to create custom reacts should be placed under the channel name and above the members list.
- The user can then add reacts by browsing their files and adding their preferred images / emotes in the custom react tab.
- Custom reacts are unique to each channel and every channel can have their own custom reacts.
- There can only be up to 5 custom reacts in a channel, an error message will be displayed to the frontend if the user tries to add more than 5 custom reacts in a channel.
- Only channel owners or flockr owners can add this custom reacts, however every member in the channel should be able to use these reacts.



**Use Case:**

- Use Case: Creating and using a custom react
- Goal in context: User and other members in the channel can react to messages there using the custom react
- Scope: Flockr app and messages (channel chat)
- Level: Subfunction
- Preconditions: Users are already registered and have joined the channel where the message will be posted and reacted too. User1 is also the owner of the channel and there have not been any custom reacts in the channel.
- Success end condition: Every member of the channel can use the custom react to react to messages and the message that gets reacted by the custom react can be viewed by all members in the channel.
- Failed end condition: Not every user can use the reacts and view the reacted message.
- Primary Actor: Users; particularly, user1 and user2.
- Trigger: User1 creates a custom react and both users can react using the custom react to a message sent by user2.

**Main success scenario:**

1. User1 opens the create custom reacts tab and adds a new react in the channel using an image called pepehands.png.
2. Backend verifies that the user making the request is a channel owner and not just a member and checks that the custom reacts in the channel has yet to exceed 5.
3. Backend verifies the image.
4. Afterwards, the backend assigns a react\_id to the custom react which is unique to the channel and adds it to the data of valid react\_ids. Afterwards, the image will then be added as a file in the frontend and should be possible to be displayed and used by the other members.
5. Flockr informs that the addition of the custom react is successful and returns a unique react id.
6. The channel members can then now use the custom react to react to all the messages in the channel.

**User response:**

It looks like it works exactly how I would want it to work, only small comment is 5 reacts seems small; I know on discord on more premium channels you can have more reacts.

NOTE: It is listed at 5 as the reacts are shown on the side of the message, if we wanted to do more, we would also have to change how the reacts are displayed. For example, how discord deals with reacts, they are listed in order under the message.

## Private Messaging (From interview 3):

As a frequent user of social networking tools, I would like to be able to message other users directly and privately using private messaging (DMs) so that I can send sensitive information privately and talk to others personally.

### UATS:

- Every user will have their own unique u\_id which numbers that will be generated by the system when they register for the first time. This u\_id is not changeable and is used by others to add them and message them privately.
- A user's u\_id will be displayed in their own respective profiles.
- A user can add other users for private messaging after adding them using their u\_id. This feature should be located under the profile icon.
- All private messages will be displayed in the 'friends' tab which is above the 'my channels' tab so that people will be able to access private messages quicker.
- Messaging will work just like messaging in channels, the same messaging features will still apply to private messaging

### Use Case:

- Use Case: Start and send a private message to another user
- Goal in context: Two users can send messages privately to each other without being in a channel
- Scope: Flockr app
- Level: Primary Task
- Preconditions: User is already registered and have obtained the other registered user's u\_id.
- Success end condition: The user can add another user as a friend and privately message them.
- Failed end condition: The user either fails to add the user's u\_id or could not send any message to them once they have been added.
- Primary actor: Users; particularly, user1 and user2.
- Trigger: User adds another user through their u\_id and sends a private message.

### Main success scenario:

1. User1 opens the add friend feature below profile and inputs the u\_id of user2.
2. Flockr requests backend to verify the u\_id. Backend informs that the u\_id is valid and adds the user as a friend.
3. User2 is recognized as a friend by user1 and user1 is recognized as a friend by user2. They should now be able to message each other directly by using the friends tab.

4. User1 opens the friends tab and sends a message to user2.
5. Flockr requests backend to verify the message and adds it into the data if the message is valid.
6. Message is sent and registered in the system.
7. Both user1 and user2 should be able to read the message when they read it from their accounts, respectively.

**User response:**

It looks like it works exactly how I would want it to work.

## [Design] Interface Design

Function name	HTTP Method	Parameters	Return Type	Exceptions	Description
/message /send	POST	(token, channel_id, message)	{message_id, tagged_users }	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>Message is more than 1000 characters</li> </ul> <b>AccessError</b> when: the authorised user has not joined the channel they are trying to post to	The existing message send description plus: Recognises a tagged user from an '@' character followed by a user's handle string. Sends email to that user to notify them. If the user is not in the channel or the user handle doesn't exist, the tagging doesn't do anything.
other/setting s/accessibility	POST	(token, settings)	{}	<b>AccessError</b> When the token is not valid.	Changes the stored settings data for channel details to then give to the front end which should be dependent on the user.
/message /react/add	POST	(token, channel_id, img_url)	{react_id}	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>The channel_id is not a valid channel.</li> <li>img_url returns an HTTP status other than 200.</li> <li>There is already more than 5 custom reacts in the channel.</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>The authorised user trying to add the custom react is not a channel owner in the channel.</li> </ul>	Given a URL of an image on the internet, generate a react_id and add the image as a valid react in the channel.
/message /react/remove	DELETE	(token, channel_id, react_id)	{}	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>channel_id is not a valid channel.</li> <li>React_id is 1 (react_id belongs to all channel) or react_id is not a</li> </ul>	Removes a valid react_id from a channel and the respective image that it corresponds to. The react_id cannot be 1 as 1 is reserved for the react_id of thumbs up in the server.

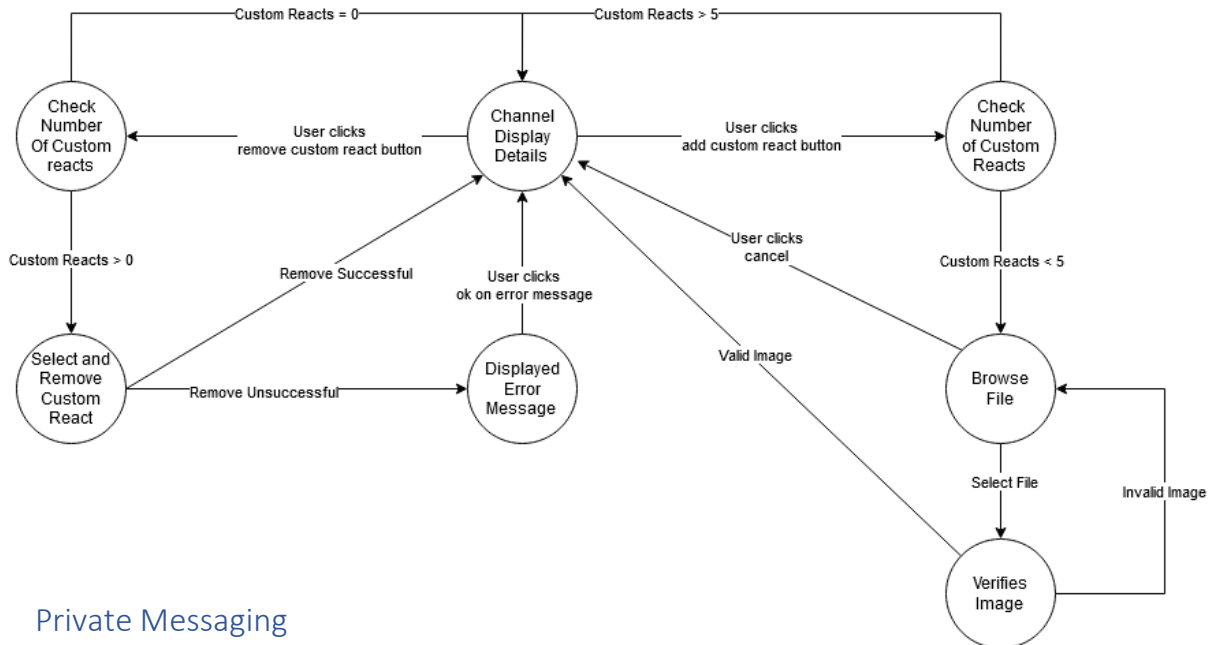
				valid react_id in the channel <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>The authorised user trying to remove the custom react is not a channel owner in the channel.</li> </ul>	
/friends/add	POST	(token, u_id)	{channel_id}	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>u_id does not belong to a valid user.</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>the token is not valid.</li> </ul>	Given a valid u_id, adds the appointed user as a friend on both ends (user requesting this function and the u_id appointed). A channel_id is returned so that message functions can be called and use to send messages as a private message.

**tagged\_users** is a list of u\_id's of the users who have been tagged in the message

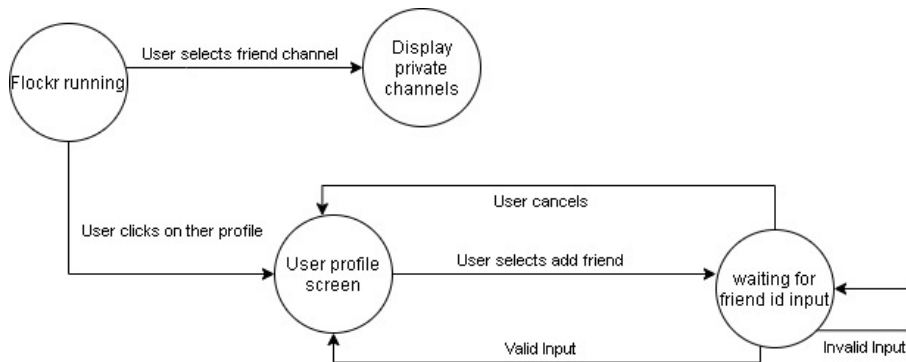
**Settings** : dictionary which contains the settings for text\_size, font, colour, icon\_size, button\_size, space, voice read text aloud and other graphic settings.

# [Design] Conceptual Modelling (State)

## Custom React



## Private Messaging



## Accessibility

