Comp1531 - Future Planning - WED13A Cactus

Requirements

| Elicitation

Target Users: University students working on group projects/society work and other people involved in teamwork driven work environments.

Target Users

Jess Qiao | jessicaqiao2@gmail.com
Simran Thind | st56011@gmail.com
Rachael Ng | rachaelng02@gmail.com

Aim

- Gaps in the service that Dreams is providing
- Find features which users would be engaged with and find satisfying to use
- Discover what is currently wrong with competing platforms and improve upon current designs

Questions

1. What platform do you currently use for Team based work for communication?
2. Do you believe teamwork communication platforms should maintain a more formal or informal approach?
3. What are the features you use the most when using this platform, are there areas that are lacking in those features?
4. What features don't you like about the platform?
5. Are there any features that the platform does not currently support that you would want it to support?
6. What do you think the next feature to implement into UNSW Dreams is?

Jess Qiao's Response

1. I currently use Microsoft Teams and Facebook Messenger for teamwork based communication, Microsoft Teams for more formal communication and Facebook Messenger for informal communication to make plans for meetings etc.

2. I prefer the formal approach to teamwork as it maintains a certain quality of work and incentivizes work to be completed on time. I don't mind if it isn't necessarily fully serious but the platform itself should promote formal work practice.

3. The features that I use the most are **Texting**, **Sending Links**, **Sharing Files**. In my view it is very important for teams to be on the same page, be able to see each other's work and to share resources with each other.

4. Facebook Messenger often has a fast reply time but lacks formal communication and easy ways to collaborate with each other. Microsoft Teams often have slow reply times as people generally ignore messages from there as it is deemed less urgent.

5. I would like a **poke button**, or way to alert others of any urgent message that has been sent. I also think as a student it would be nice to have a collaborative **text editor** so that for projects we are able to write collaboratively while remote for note taking, report writing etc.

6. **Poke Button**, **Collaborative Text Editor.**

Simran Thind's Response

1. I use Slack and email for my job as a Research and Development intern at Invast Global, Facebook Messenger and Microsoft Teams for group projects and other teamwork activities such as case competitions.

2. I believe that there should be a mixture. Teams need both formal and informal communication, so using different platforms for those approaches will provide the best communication within the team

3. Slack: **personal messages**, **group channels**, **video** and **voice calls** The main area that's lacking in these features is due to the layout, important messages can often get lost in between unimportant replies and comments.

   Email: emails are used for more formal events or functions at the company or any important **announcements**. However, it is often easy to forget to check emails and sometimes emails can get lost or confusing if there are many different email chains

   Facebook Messenger: **messaging** is very fast and often response times are fast, there is little worry of saying the wrong thing as well. A little bit too informal, things can get a little too banterous on there.

   Microsoft Teams: **calls**, **messages**, **group channels**. Very similar to my criticisms of Slack, where the format is quite bad and usually, maybe because of the scenarios in which I am using Microsoft Teams, but the **response time is quite slow**.

4. (See Above)

5. A way to make **messages less linear**, possibly closer to the setup of emails. This would be a **toggle on/off feature** that would be activated for some channels and deactivated for others. I would also like to have a way to do **collaborative work** with groupmates / colleagues.

6. Definitely **message replies**, **collaboration tools** and different **channel modes**.

1. Facebook/Messenger for chatting, Zoom/MS Teams for calls/teamwork and Trello for goal tracking primarily for society work and for group projects.

2. Formal approach -> specific structure means easier usage and clarity for the user
   More reliable as there is a set standard for the organisation

3. Files tab where all documents are stored, similar to Google Drive -> convenient how everything is in a **centralised location**

   Private messaging and the calling functions are also very useful. Sometimes I dislike messenger because things can get **lost and cluttered**.

4. MS Teams has a similar group editing function as Google Docs but is very slow and doesn't pick up on spelling/grammatical errors quickly. Messages in messenger and microsoft teams are **sometimes easily lost** as well

5. More personalisation with **colour themes**, an easy method for **centralised files** and messages to be integrated together to **reduce clutter** and for **increased centrality**.

6. **Colour Themes**, **Message Replies**, **File Storage**

| Analysis & Specification

User Story - Poking Users

As a **university student working on group projects** I want to **be able to remind group members of deadlines and urgent Messages** so that **workflow and communication can be consistent.**

Scenario: UserA wants to get UserB to interact with a message
Given: Both users are in the same channel or dm
When: UserA selects the 'Poke' feature
And: UserA selects a message or dm
Then: UserB receives a Poke notification to interact with the message or dm

Use Case - Poking Users

- **Use Case:** A user (UserA) wants to poke another user (UserB)
- **Goal in Context:** UserA wants UserB to read / interact with a message/file/reply
- **Scope:** Network application
- **Level:** Subfunction
- **Preconditions:** UserA and UserB has an account with UNSW Dreams
- **Success End Condition:** UserB successfully receives a notification from UserA
- **Failed End Condition:** UserB does not receive a notification from UserA
- **Primary Actor:** UserA
- **Trigger:** UserA pokes UserB

User Story - Channel Modes (Functional)

As a **corporate worker** I want to **be able to change the formality of channels that I own** so that **we are able to have more formal announcement channels.**

Scenario: A User wants to make a channel specifically for announcements
Given: The user owns the channel
When: The user changes the channel settings
And: The user changes the setting "Announcement"
And: The channel changes to an announcement channel
Then: Non-owner members cannot make main-thread messages in that channel

Use Case - Channel Modes (Functional)

- **Use Case:** Changing a channel to an announcements channel
- **Goal in Context:** For a user to change their channel to make other users unable to make top-level messages.
- **Scope:** Network application
- **Level:** Subfunction
- **Preconditions:** user is an owner of the channel
- **Success End Condition:** other non-owners are unable to make top-level messages
- **Failed End Condition:** non-owners are able to make top-level messages
- **Primary Actor:** owner-user
- **Trigger:** owner-user wants to change a channel to an announcements channel

User Story - Channel Modes (Visual)

As a **university student** I want to **be able to change the theme of the platform** so that **the platform is more visually appealing.**

Scenario: A User wants to change to dark mode
Given: The user has navigated to the user settings page
When: The user has selected the "Change Theme" option
And: The user changes the setting theme to "Dark Mode"
Then: The theme of Dreams changes

Use Case - Channel Modes (Visual)

- **Use Case:** Changing UNSW Dreams to Dark Mode
- **Goal in Context:** For a user to change their visual theme of UNSW Dreams to Dark Mode
- **Scope:** Network application
- **Level:** Subfunction
- **Preconditions:** user has a UNSW Dreams account
- **Success End Condition:** user has successfully changed the theme to Dark Mode
- **Failed End Condition:** UNSW Dreams' theme is not Dark Mode

- **Primary Actor:** user
- **Trigger:** user changes the settings

User Story - Message Replies

As a **university student** I want to **be able to respond to messages directly** so that **communication between parties is clearer.**

Scenario: A User wants to respond directly to a message
Given: The User has navigated to that channel or dm
When: The user has selected the reply feature
And: The user writes a message
Then: A reply is displayed beneath the parent message

Use Case - Message Replies

- **Use Case:** Replying to a Message
- **Goal in Context:** Replying to a message as a subheading or reply, not as a completely new message.
- **Scope:** Network application
- **Level:** Subfunction
- **Preconditions:** User has authorization to message in the same channel the message they want to reply to is in.
- **Success End Condition:** a reply is successfully displayed underneath the message being replied to
- **Failed End Condition:** a reply is not successfully displayed
- **Primary Actor:** user
- **Trigger:** user presses reply button underneath a message

User Story - File Storage

As a **university student** I want to **be able to share files** so that **resources between group mates can be shared.**

Scenario: UserA wants to send files to channel/dm
Given: UserA has navigated to a channel or dm
When: UserA has selected a file to send
And: UserA writes a descriptive message for the file
Then: The file is available for all Users in the channel or dm

Use Case - File Storage

- **Use Case:** Sharing a File
- **Goal in Context:** For a user to share a file in a channel
- **Scope:** Network application

- **Level:** Subfunction
- **Preconditions:** user has access to that channel
- **Success End Condition:** A file is shared and is able to be accessed by everyone in that channel
- **Failed End Condition:** The file cannot be accessed by all members of the channel
- **Primary Actor:** user
- **Trigger:** user messages a file to a channel

| Validation

Jess Qiao's Comments

I think the poking function is really good, especially in scenarios where people may see the message but forget to reply or forget to interact. Additionally, I completely forgot to mention file sharing in my first response but I'm definitely glad you've included it here as well. I think the themes were a good addition as well because it's easier to work when what you're working on has an appealing aesthetic. I think your solutions to the issues I came up with solve them to quite a large extent, but am disappointed a collaborative text editor didn't make the cut.

Simran Thind's Comments

I think the function mode changing is quite useful and satisfies my ask of having a less linear channel mode so that we are able to maintain a formal and informal method of communication in one platform. Furthermore, the ability to reply to messages was much needed in your application. I think your solution to the problems I had is satisfactory.

Rachael Ng's Comments

I am [so excited] that you guys are implementing the visual color themes. It's something that I can't actually get excited about a platform if it doesn't have it. My eyes just can't adjust to light backgrounds anymore. The storage of files and message replies are also present which I am glad to see. I'm happy that all the features I suggested seem to have been included. The extent to which your use cases solve my problems are definitely high.

| Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
/message/poke/v1 Given a user_id and a channel_id creates a notification for the user referencing the first 20 characters of the message associated with message_id.	POST	**Parameters:** (token, u_id, channel_id, message_id) **Return Type:** {}	**InputError** when any of: ● u_id does not refer to a valid user ● channel_id does not refer to a valid channel ● message_id does not refer to a valid message **AccessError** when ● The authorised user is not a member of the channel
/message/dmpoke/v1 Given a user_id and a dm_id creates a notification for the user referencing the first 20 characters of the message associated with message_id.	POST	**Parameters:** (token, u_id, dm_id, message_id) **Return Type:** {}	**InputError** when any of: ● u_id does not refer to a valid user ● dm_id does not refer to a valid dm ● message_id does not refer to a valid message **AccessError** when ● The authorised user is not a member of the dm
/channel/changemode/v1 An owner of a channel can change the mode of a channel they own according to mode_id.	PUT	**Parameters:** (token, channel_id, mode_id) **Return Type:** {}	**InputError** when any of: ● channel_id does not refer to a valid channel ● mode_id does not refer to a valid mode

			AccessError when any of: ● The authorised user is not a member of the channel ● The authorised user is not an owner of the channel
/channel/details/v3 Given a Channel with ID channel_id that the authorised user is part of, provide basic details about the channel	GET	**Parameters:** (token, channel_id) **Return Type:** { name, is_public, mode_id, owner_mem bers, all_members }	**InputError when any of:** ● Channel ID is not a valid channel **AccessError when any of:** ● Authorised user is not a member of channel with channel_id
/user/profile/changemode/v1 A user can change the UNSW Theme according to mode_id.	PUT	**Parameters:** (token, mode_id) **Return Type:** {}	**InputError** when: ● mode_id does not refer to a valid mode
/user/profile/v3 For a valid user, returns information about their user_id, email, first name, last name, and handle, mode	GET	**Parameters:** (token) **Return Type:** {user}	**InputError** when: ● User with u_id is not a valid user
/message/reply/send/v1 Given a message_id creates a reply for that message which has its own unique message_id.	POST	**Parameters:** (token, message_id, message) **Return Type:** {message_id }	**InputError** when any of: ● Message is more than 1000 characters **AccessError** when: the authorised user has not joined the channel the original message is in
/message/reply/delete/v1	DELETE	**Parameters:** (token,	**InputError** when any of:

|			
Given a message_id given that it is a reply, remove that message if the authorised user is the original poster of the main thread (cannot remove messages from owners however) or from an owner, or from the poster of the reply.		message_id) **Return Type:** {}	• Message (based on ID) no longer exists • Message is not a reply **AccessError** when none of the following are true: • Message with message_id was sent by the authorised user making this request • The authorised user is an owner of this channel (if it was sent to a channel) or the **Dreams** • Authorised user making the request is the original poster of the message being replied to and the message being deleted was not written by an owner of the channel or the **Dreams**
/message/replies/get/v1 Given a Channel with ID channel_id that the authorised user is part of and a Message with ID message_id that is part of the Channel, return up to 50 replies between index "start" and "start + 50". Message with index 0 is the most recent message 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 messages in the channel, returns -1 in "end" to indicate there are no more messages to load after this return.	GET	**Parameters:** (token, message_id, channel_id) **Return Type:** {messages, start, end}	**InputError** when any of: • Channel ID is not a valid channel • Message ID is not a valid message • Message ID is not part of the Channel • start is greater than the total number of messages in the channel **AccessError** when any of: • Authorised user is not a member of channel with channel_id

/message/file/send/channel/v1 Sends a file to a channel denoted by channel_id with an attached message. The file denoted by the file_path is displayed as a message in Dreams.	POST	**Parameters:** (token, file_path, channel_id, message) **Return Type:** {message_id}	**InputError** when any of: ● Channel ID is not a valid channel ● Message is over 1000 characters ● File_Path does not point to a valid file ● File pointed to by File_Path exceeds 128 MB **AccessError** when any of: ● Authorised user is not a member of channel with channel_id
/message/file/send/dm/v1 Sends a file to a dm denoted by dm_id with an attached message. The file denoted by the file_path is stored as a message.	POST	**Parameters:** (token, file_path, dm_id, message) **Return Type:** {message_id}	**InputError** when any of: ● DM ID is not a valid DM ● Message is over 1000 characters ● File_Path does not point to a valid file ● File pointed to by File_Path exceeds 128 MB **AccessError** when any of: ● Authorised user is not a member of dm with dm_id

| Variable Name	Type
(outputs only) named exactly **notifications**	List of dictionaries, where each dictionary contains types { channel_id, dm_id, notification_message } where channel_id is the id of the channel that the event happened in, and is -1 if it is being sent to a DM. dm_id is the DM that the event happened in, and is -1 if it is being sent to a channel. The list should be ordered from most to least recent. Notification_message is a string of the following format for each trigger action: ● tagged: "{User's handle} tagged you in {channel/DM name}:

	{first 20 characters of the message}" ● reacted message: "{User's handle} reacted to your message in {channel/DM name}" ● added to a channel/DM: "{User's handle} added you to {channel/DM name}" ● poked in a channel/DM: "{User's handle} has poked you, interact with {first 20 characters of the message}
(outputs only) named exactly **user**	Dictionary containing u_id, email, name_first, name_last, handle_str, profile_img_url, mode_id

| mode_id	channel	user
1	Normal	Normal
2	Announcements Only owner members can make top-level messages, normal members can still reply	Dark Mode
3	File Storage Only files can be sent to this channel, images, links, files allowed	Peach/Red Color Palette
4	Important Messages sent to a channel marked in this mode automatically tags all users in the channel	Blue/Grey Color Palette

| Conceptual Modelling

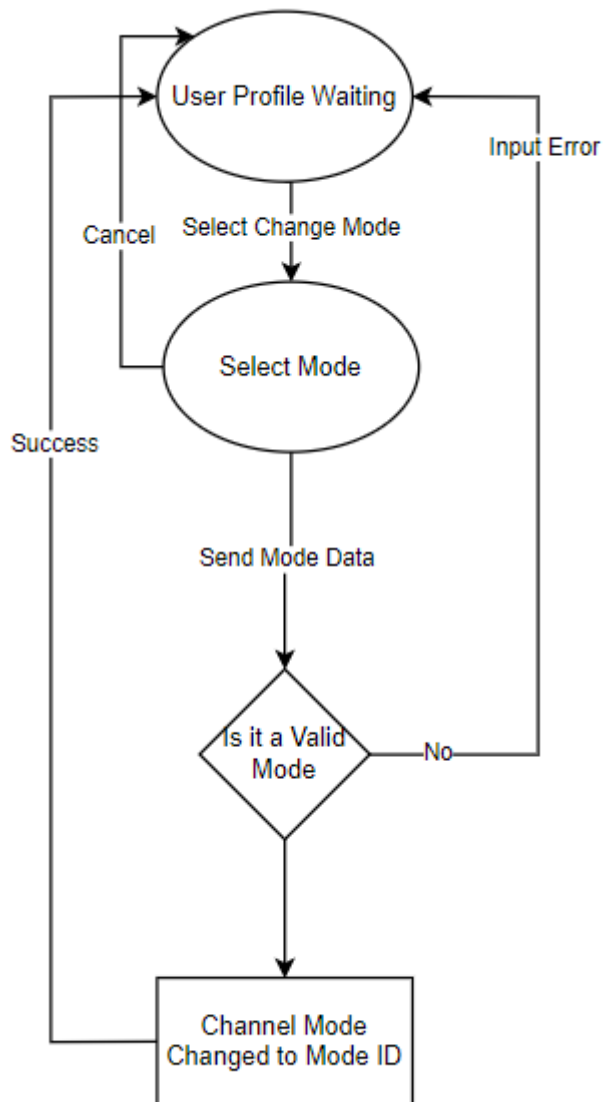Model for sending a poke for a message in a channel:
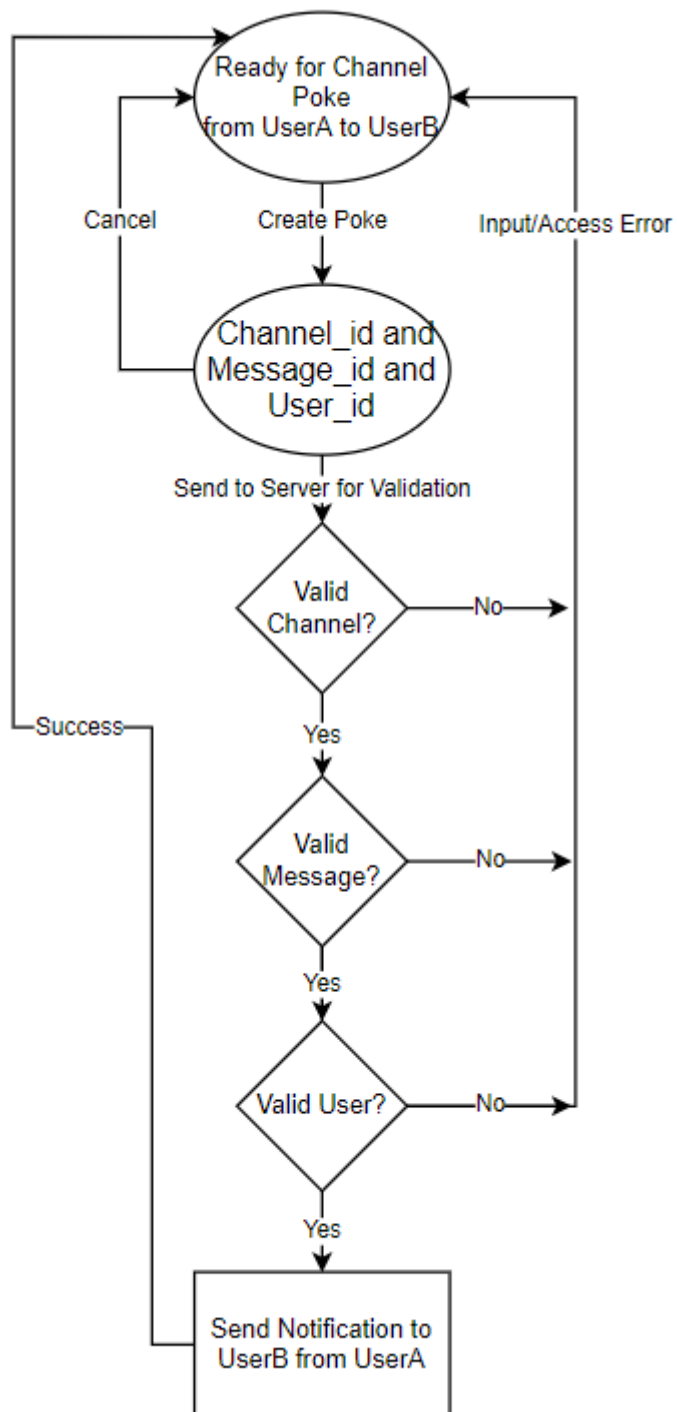
Model for sending a Poke in a DM:
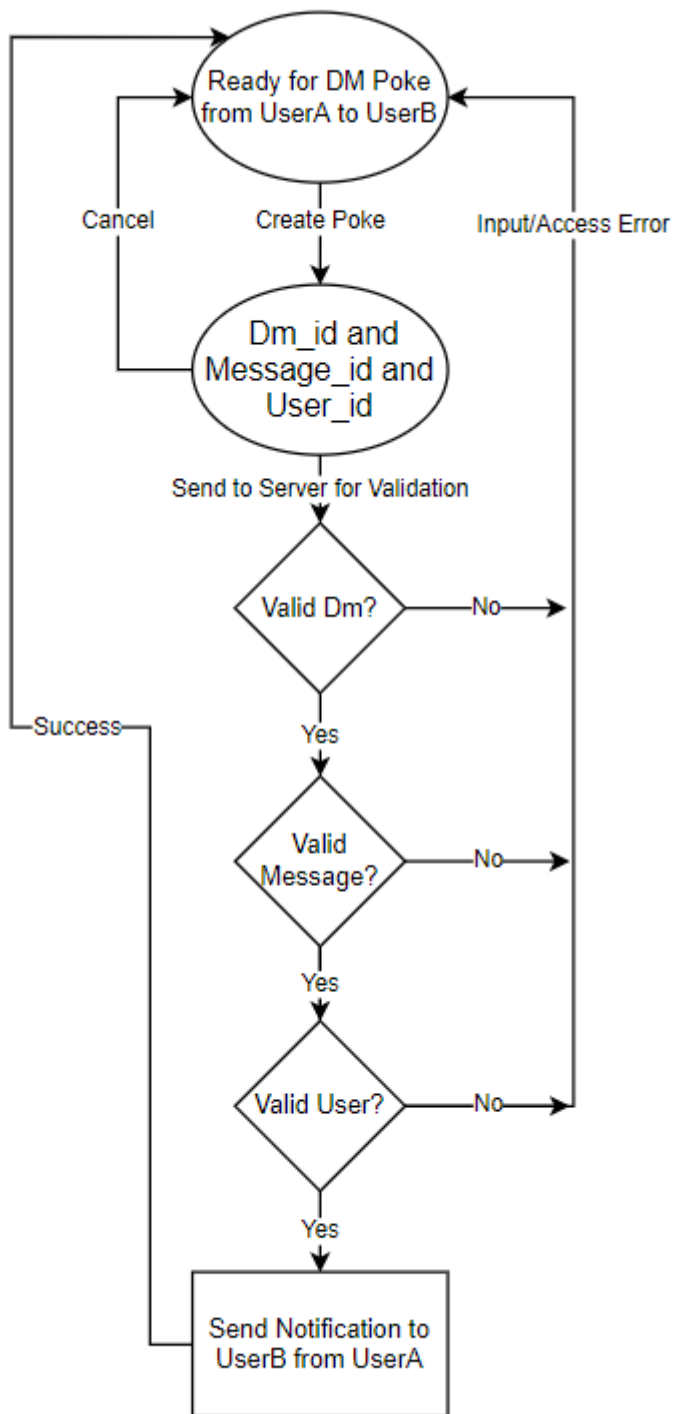
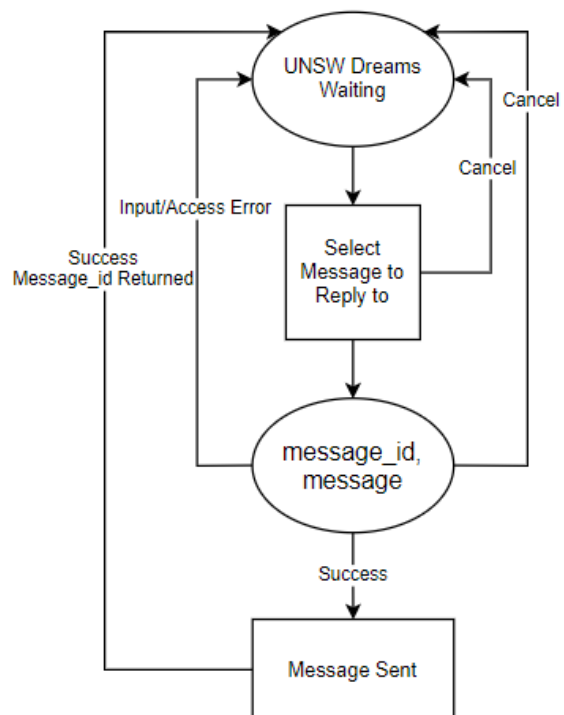Model for Changing a Channel Mode:
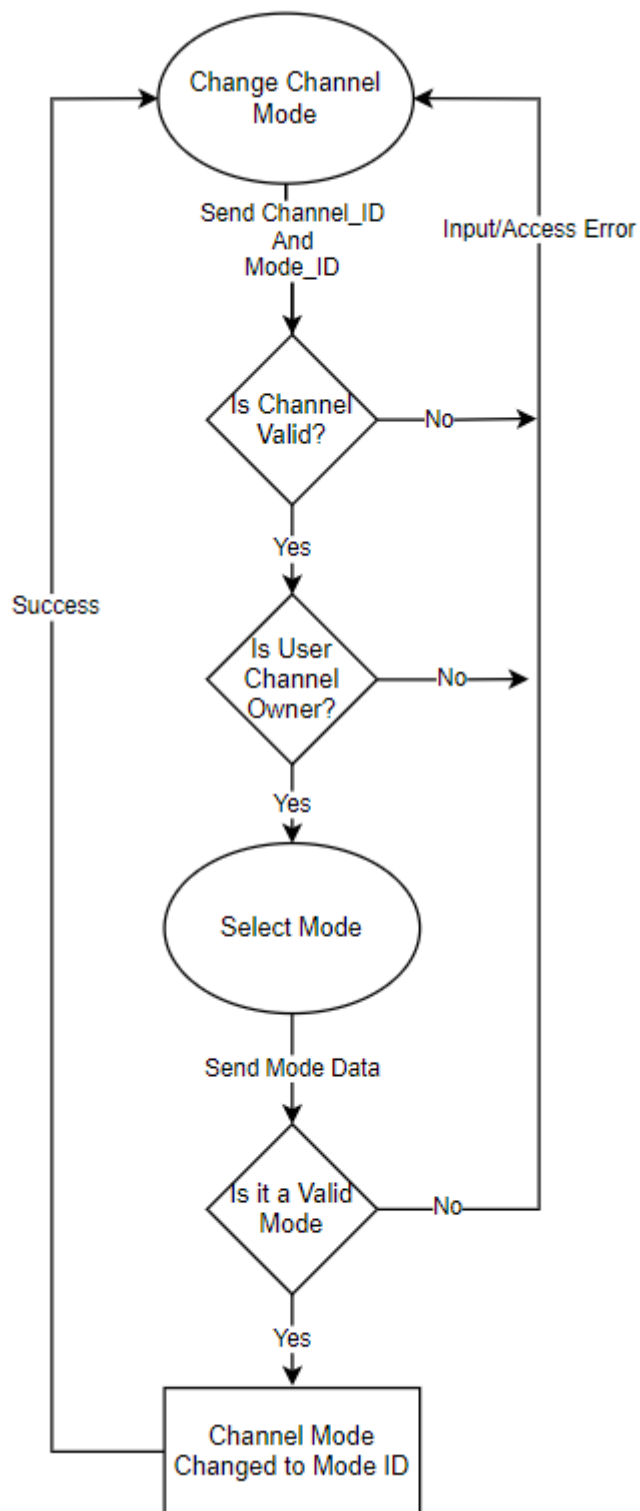
Model for changing User Theme:

Model for Reply Sending:

Model for Reply Deletion: