

T13B_ALPACA

Future Plan

Contents

Elicitation

- Survey Questions

Use Cases

- User Stories and Acceptance Criteria

Validation

- Interviews and comments

Interface Design

- The solution

Conceptual Modelling (State)

- State diagram

Survey Questions:

- 1) Does the software fulfill its function as a means of teamwork-driven communication tool? If not, what are some requirements for the software that are not yet implemented.
- 2) What is your opinion on the level of organisation the streams app provides? How could it be improved?
- 3) What major quality of life improvements do you feel the software needs?
- 4) Did any of the features feel unintuitive?
- 5) Are there any features missing on this service, not already mentioned in previous answers, that you feel should be implemented?

Interview 1:

Name: Eve

Email: pinkycat022@gmail.com

1. It doesn't have some basic features of other communication apps like Microsoft teams including voice chats. You also can't stream what you are currently working on to other team members in a group.
2. There is almost no organisation within the channels, only a single message feed. There should be a way to split the messages into separate channels to organise according to topics.
3. There should be a way to customize the interface. At the very least having the ability to switch to a dark mode. There should be the ability to change how your username appears in a channel.
4. The stand-up feature as there was no way to access the feature without typing a command. It doesn't really make sense why the first and last name are joined together and with capital letters.
5. Different chat channels and voice chat are the major ones that should be added.

Interview 2:

Name: Aya

Email: peko.chuan@gmail.com

1. There are no voice chats or any way to split the messages up into separate chats according to topic. Not being able to split the messages means groups of members can't text chat in separate chats when communicating on separate things and you can't organise by topics The same goes for voice chats.
2. Like I already wrote, not having different text chats is a major issue as all the message feed goes into the one text chat, not being sorted by topic.
3. Customization. Should add dark mode.
4. It doesn't make sense why you start in a channel.
5. No way to turn off notifications. Should also have it so notifications can be switched off or on for individual chats.

Proposed Solution:

- 1) We can add a sidebar in the channel and let users discuss different topics in separate chats but remain in the same channel.
- 2) Add voice chat system
- 3) Add whiteboard and share screen
- 4) Add Dark mode

User Stories and Acceptance Criteria

Story 1: (2) As a uni student, I want to organise the messages into various topics.

Acceptance Criteria:

A new sidebar is introduced (front end) next to all the channels, which is associated with the channel that is being accessed, each having a different message history.

This sidebar contains a list of all the text chats which appear under the heading Text Chats. Messages are stored in the text chat they are sent from.

By default, there is a general chat that is created when the channel is created.

Owner members are able to add new text channels and change the name of these text channels.

An option appears when an owner member clicks on the Text Chats heading called “Add new text chat”.

This option is greyed out for users that are not owners.

Users have the ability to alter the names of text channels and remove them, which deletes the message history.

Story 2: (2) As a uni student, I want to be able to actually talk to my team members in a voice channel.

Acceptance Criteria:

Underneath the previously mentioned text chats, there will be voice chats a user may join.

When a user joins a voice channel with a working microphone any other users in the voice channel will be able to hear them speak.

User profiles appear underneath the voice channel when they join the channel to indicate that they are in said channel.

If a person mouses over the profile details such as the user's handle will appear.

Like text chats a default general chat is created when the channel is created.

Story 3: (1) As a uni student, I want to stream what I am currently working on to my team members and bring up a shared whiteboard to visually express concepts.

Once a user is in a voice chat they may click on their profiles under the channel title and access a stream option.

Once this is clicked they will be given the option to share one of their screens.

If a user is sharing their screen another user in the chat will be able to click on the user's profile and then click on the view stream option.

A whiteboard icon will be located next to the text voice channel.

Any user in the chat can click on the whiteboard which will bring up a pop-up whiteboard.

Users can then simultaneously draw on the whiteboard in a colour which is not currently taken.

A reset button is located at the top left of the whiteboard.

The whiteboard does not reset when there is no longer a user in the channel

Story 4: (2) As a user, I want to have less strain placed on my eyes by the white background.

Acceptance Criteria:

Under the profile, there will be a series of options under the title "Settings".

One of these settings allows the user to toggle on and off dark mode.

Light mode which is already implemented is the default.

Dark mode will switch the background to a dark grey.

Use Case:

The channel dictionary no longer has a messages list but a list called “text_chat”.

“Text_chat” contains the keys “title” which stores the chat name, “messages” which is a list of messages, and “text_chat_id”

Use Case: (User story 1)

Step 1: User that is an owner of the channel clicks on the “Add new text chat” option.

Step 2: A pop up appears that allows the user to input a name for the text chat.

Step 3: The user inputs the name and clicks “Create text chat”

Step4: The name of the chat is initialised to the input title. In the front end a new chat name will appear at the end of the chat names.

Step5: The user clicks on the title of the text chat they just created which brings up the new text channel with a completely empty message history.

Step6: The user types a message in the chat box and presses the return key.

Step7: A new message appears in the new chat with a time sent and handled next to it like any other message. The user has access to all other features available in any other text chat such as creating standups, deleting and editing messages.

Step8: Another user that is a member of the text channel sends another message in the same way.

Step9: The second user's message appears under the first user's message.

Step10: The user clicks on the title of the “general chat” created when the channel was made.

Step11: The user is shown the message history of the “general chat”. They can no longer see the previous two messages.

Step12: The user types a new message into the chat box and presses the return key.

Step13: The sent message is shown to the user in the “general chat”, and it is the first one to appear.

Step14: The user has decided that they no longer need the text chat they created earlier and clicks on the chat name in the sidebar bringing up the option “delete text chat”.

Step15: The user clicks on “delete text chat ”.

Step16: The chat is deleted , removing the message history for the chat in the process.

Step17: The title of the chat disappears from the front end interface for the user. Any users looking at the chat messages in that chat now view the text channel next up in the list.

Validation: The extent to which these use cases would adequately describe the problem they're trying to solve

Eve's comment : The proposed implementation for the text chat seems to be an intuitive way to implement multiple chats.

Aya's comment : The use case is a bit hard to digest but the idea for the implementation is great.

Interface Design

User Story 1

Name & Description	HTTP Method	Data Types	Exceptions
<p>topic/create</p> <p>Creates a new topic in the channel for users to discuss or chat about the topic.</p>	POST	<p>Parameters: {token, name ,channel_id }</p> <p>Return Type: {topic_id}</p>	<p>InputError when:</p> <ul style="list-style-type: none">- length of name is less than 1 or more than 20 characters- channel_id does not refer to a valid channel <p>AccessError when:</p> <ul style="list-style-type: none">- channel_id is valid and the authorised user is not a member of the channel
<p>topic/messages</p> <p>Given a topic with ID topic_id that the authorised user is a member of, return up to 50 messages between index "start" and "start + 50". Message with index 0 is the most recent message in the topic. This function returns a new index</p>	GET	<p>Parameters: {token ,topic_id, start }</p> <p>Return Type: {messages, start, end }</p>	<p>InputError when any of:</p> <ul style="list-style-type: none">- topic_id does not refer to a valid topic- start is greater than the total number of messages in the channel <p>AccessError when:</p>

"end" which is the value of "start + 50", or, if this function has returned the least recent messages in the topic, returns -1 in "end" to indicate there are no more messages to load after this return.			<ul style="list-style-type: none"> - topic_id is valid and the authorised user is not a member of the channel
<p>message/topicsend</p> <p>Send a message from the authorised user to the channel specified by channel_id and topic_id. Note: Each message should have its own unique ID, i.e. no messages should share an ID with another message, even if that other message is in a different channel.</p>	POST	<p>Parameters: {token , channel_id, topic_id, message }</p> <p>Return Type: {message_id}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - channel_id or topic_id does not refer to a valid channel - length of message is less than 1 or over 1000 characters <p>AccessError when:</p> <ul style="list-style-type: none"> - channel_id and topic_id is valid and the authorised user is not a member of the channel

User Story 2

Name & Description	HTTP Method	Data Types	Exceptions
--------------------	-------------	------------	------------

<p>voicechannel/create</p> <p>Creates a new voicechannel in the channel/dm.</p>	POST	<p>Parameters: {token ,channel_id, dm_id, name}</p> <p>Return Type: {voicechannel_id}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - length of name is less than 1 or more than 20 characters - channel_id or dm_id does not refer to a valid channel/DM <p>AccessError when:</p> <ul style="list-style-type: none"> - channel_id is valid and the authorised user is not a member of the channel
<p>voicechannel/join</p> <p>Given a voicechannel_id of a voicechannel that the authorised user can join, adds them to that channel.</p>	POST	<p>Parameters: {token , voicechannel_id }</p> <p>Return Type: {}</p>	<p>InputError when any of:</p> <ul style="list-style-type: none"> - voicechannel_id does not refer to a valid voice channel - the authorised user is already a member of the voicechannel
<p>voicechannel/leave</p>	POST	<p>Parameters: {token , voicechannel_id }</p> <p>Return Type: {}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - voicechannel_id does not refer to a valid voicechannel <p>AccessError when:</p> <ul style="list-style-type: none"> - voicechannel_id is valid and the authorised user is

			not a member of the channel/DM where the voice channel is
--	--	--	---

User Story 3

Name & Description	HTTP Method	Data Types	Exceptions
<p>voicechannel/sharescreen</p> <p>Allow user to sharescreen</p>	POST	<p>Parameters: {token ,voicechannel_id}</p> <p>Return Type: {sharescreen_id}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - voicechannel_id does not refer to a valid voice channel <p>AccessError when:</p> <ul style="list-style-type: none"> - voicechannel_id is valid and the authorised user is not a member of the channel
<p>voicechat/whiteboard</p> <p>Pop up a whiteboard in the voice channel.</p>	POST	<p>Parameters: {token , voicechannel_id}</p> <p>Return Type: {whiteboard_id}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - voicechannel_id does not refer to a valid voice channel <p>AccessError when:</p> <ul style="list-style-type: none"> - voicechannel_id is valid and the authorised user is not a member of the channel

<p>voicechannel/react</p> <p>Allow user to react for example raise hand to notify the speaker and ask question</p>	POST	<p>Parameters: {token , voicechannel_id, react_id }</p> <p>Return Type: {}</p>	<p>InputError when:</p> <ul style="list-style-type: none"> - voicechannel_id does not refer to a valid voicechannel - react_id does not refer to a valid react ID <p>AccessError when:</p> <ul style="list-style-type: none"> - voicechannel_id is valid and the authorised user is not a member of the channel/DM where the voice channel is

User Story 4

We can add Dark mode at Frontend.

State diagram

