

COMP 1531 planning - Iteration 3

Elicitation

	Name: John Smith Email: john.smith@outlook.com.au	Name: Jane Zhang Email: jane.zhang@gmail.com
What platforms do you use to communicate with others?	LinkedIn, Zoom, Facebook	WhatsApp, WeChat, Facebook, Zoom
Are there any specific features or functionalities that you believe are missing from these platforms that would improve your user experience?	Sometimes important messages get lost in the clutter of messages, where it can be hard to keep track. I would like to add a feature which adds a personal tag to messages, and is able to search the message by tags.	It would be nice if there was a feature that could automatically translate the messages into different languages
What are some of the challenges you have faced while using your platforms and would like to be fixed?	The platform has difficulty in dealing with unwanted messages or spam from a specific user.	One of the biggest challenges is managing notifications and alerts. Sometimes I get overwhelmed with notifications and it can be hard to stay focused on my work.
What would you change about the design of the platform to make it more appealing?	I would like to see the active status of other friends on the chat list without click into the chat. (Have different icon and colour for each status)	It would also be great to have more customization options for chat theme colours.

How simple would it be for a person who has no prior experience with your platforms to navigate and understand how to use them?	It is not simple for a person who has no prior experience with the platforms to navigate and understand how to use them. The platforms have complex interfaces with many features and options, which can be overwhelming for new users.	The learning curve for new users on the platforms can be steep, as there can be many different settings and features to understand. Without proper guidance or tutorials, it may take some trial and error for a new user to figure out how to navigate and use the platform.
---	---	---

How to solve the problem and a brief description of a proposed solution	
Problem	Solution
Hard to keep track of messages, and unable to put personal tags on messages.	The platform could introduce a feature that allows users to tag their messages with personalized labels or keywords, which would make it easier to find and organize important messages. Additionally, a search function and search bar could be implemented that allows users to quickly find messages by searching for specific tags.
It is difficult for user that speaks different language to efficiently understand the messages	To help users communicate with people who speak different languages, the platform could introduce a feature that translates messages into the recipient's preferred language and display them in the chat.
It is hard to deal with unwanted messages and spams	The platform could introduce better tools, including blocking and reporting, to dealing with unwanted messages or spam from specific users.
Unable to manage notification and alert efficient	The platform could provide users with more control over their notifications and alerts, allowing them to customize their settings and reduce distractions. (Such as not receive any alert from tag or reaction)
Unable to view user active status with a glance	Add an active status for each user, and the platform could display the user's activity status as a small icon next to their profile picture. (Statuses such as online, offline ...)

Unable to customise display options such as colour theme of each chat	To improve the visual appeal of the platform, users could be given more customization options with the colour theme of each chat.
Difficult it figures out how to navigate the platform and features for new users	When the user first registers, the platform displays a guidance to walk them through the features and settings with detailed description.

Analysis and Specifications – Use Cases

User Stories and User Acceptance Criteria

Tag Messages with Personalized Labels

User Story:

As a user, I want to be able to tag my messages and search for them using these tags via a search bar, so that I can easily find and organize important messages.

User Acceptance Criteria:

- The platform should allow users to add custom labels to messages in the chat.
- The user should be able to view and edit these labels in the message history.
- The search feature should allow the user to search for messages using tags that have been assigned to them.
- The search field should be clearly labelled and have a placeholder text, "Search for messages by tags", explaining how to use it.
- The placeholder disappears once the user starts typing.
- The search feature should trigger once the user clicks the "Search" button.
- The user must type in a valid tag.
- After the search request has been entered, return a list of messages with the specified tags.
- The list is ordered chronologically.
- The search bar should limit the user to 200 characters or less to ensure efficient searching. (The custom tag also has a length limit of 200 characters)

Enable Message Translation

User Story:

As a user, I want a feature that translates messages on click into different languages so that I can communicate with people who speak different languages. (Attached below the original message in the chat).

User Acceptance Criteria:

- The platform should allow users to translate messages into different languages by clicking a translation button located in the chat message.
- The translation button should be easily recognizable and accessible to the user.
- The translation feature should support multiple languages.
- After clicking the translation button, the user is prompted with a list of languages to translate to.
- The translated message should be accurate and provide a reasonable translation of the original message.
- The user is prompted with a confirmation message once they have clicked the translation button.
- The translated message should appear in the chat conversation below the original message.

Display Friend Activity Status

User Story:

As a user, I want to see the active status of other friends to easily tell who is available to chat.

User Acceptance Criteria:

- The platform should indicate the online or offline status of friends on the chat list with different coloured icons. (Placed next to the profile photo of the user).
- In profile settings, the user can set their active status from a set of statuses. (Online, offline...)
- The user can only change their own active status
- The user should be able to easily identify the status of their friends at a glance.

Manage Notifications and Alerts

User Story:

As a user, I want more control over my notifications and alerts so that I can reduce distractions and stay focused on my work.

User Acceptance Criteria:

- The platform should allow users to snooze notifications and alerts for a set period.
- The user should be able to easily access the snooze feature from the notification settings menu. (A snooze button in the notification setting menu)
- Once the snooze button is selected, the user is prompted with list of times, including “1 hour, 24 hour and Always On”
- The user is prompted with a confirmation message when they snooze notifications and alerts.
- The user should not receive any notification while the snooze feature is on

Provide New User Onboarding

User Story:

As a user, I want tutorials or guidance to help me navigate and understand the platform more easily.

User Acceptance Criteria:

- When a new user registers, a page is prompted to the user that covers the platform's most important features, settings, and navigation.
- There would be a “close” button at the bottom of the page, which would return to the home page on click.
- Existing users should not be prompted with such a guide.

Report Spam or Inappropriate Messages

User Story:

As a user, I want better tools for dealing with unwanted messages or spam from specific users so that I can have a more pleasant chatting experience.

User Acceptance Criteria:

- The platform should allow users to block all messages from a specific user by clicking a blocking button located next to the message of the block user.
- The user is prompted with a confirmation message when they block another user.
- The user can unblock such a user by re-clicking the block button.
- The user should not receive any notification of the blocked user in the current chat, while the blocking status is on.

Customise Chat Theme

User Story:

As a user, I want more customization options for chat colours, so that I can personalise the platform to my liking.

User Acceptance Criteria:

- The platform should allow users to choose and customise the chat colour theme from a list of available themes in the chat setting.
- The user is prompted with a confirmation message when the colour of the chat is changed.
- The platform should ensure that the customised chat theme is saved and applied across all devices that the user is logged in to.

Use Cases

Use Case: Tag Messages with Personalized Labels

- **Goal in context:** A user can tag messages with personalised labels or keywords and search for them using the tags.
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account, is logged in, and has messages in their chat history.
- **Success End Condition:** The user can easily tag messages with personalised labels and search for them using the tags.
- **Failure End Condition:** The user cannot tag messages with personalized labels or search for them using the tags.
- **Primary Actor:** User
- **Trigger:** The user selects the option to add a tag to a message in the chat

Success Scenario 1: Add a tag to a message.

1. User selects a message in the chat that they want to tag.
2. Chat platform displays the message details.
3. User selects the "Add Tag" option.
4. Chat platform displays a text box where the user can enter a personalized label or keyword.
5. User enters a valid keyword in the text box.
6. User clicks the "Save" button.
7. Chat platform saves the tag to the message and displays it next to the message.

Success Scenario 2: Search for messages using a tag.

1. User selects the search option in the chat platform.
2. Chat platform displays a text box where the user can enter a personalized label or keyword.
3. User enters a valid keyword in the text box.
4. User clicks the "Search" button.
5. Chat platform displays a list of messages that match the search query based on the specified tags.

Success Scenario 3: Edit or delete a tag.

1. User selects a message in the chat that has a tag they want to edit or delete.
2. Chat platform displays the message details and the tag associated with it.
3. User clicks on the tag.
4. Chat platform displays a text box where the user can edit the label or keyword.
5. User edits the label or keyword in the text box.
6. User clicks the "Save" button.
7. Chat platform updates the tag on the message.

Use Case: Message Translation

- **Goal in context:** A user can translate messages into different languages in the chat.
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account, is logged in, and has received messages in a different language.
- **Success End Condition:** The user can translate messages into different languages in the chat.
- **Failure End Condition:** The user cannot translate messages into different languages in the chat.
- **Primary Actor:** User
- **Trigger:** The user clicks the translation button located in the chat message.

Success Scenario:

1. User clicks the translation button located in the chat message.

2. Chat platform prompts the user with a list of languages to translate to.
3. User selects the language they want to translate the message to.
4. Chat platform displays a confirmation message.
5. Chat platform translates the message into the selected language.
6. The translated message appears in the chat conversation below the original message.
7. The translated message is reasonably accurate.

Use Case: Display Friend Activity Status

- **Goal in context:** A user can see the activity status of their friends in the chat list
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account, is logged in, and has friends in their chat list.
- **Success End Condition:** The user can see the activity status of their friends in the chat list.
- **Failure End Condition:** The user cannot see the activity status of their friends in the chat list.
- **Primary Actor:** User
- **Trigger:** The user views their chat list

Success Scenario 1: View friend activity status

1. User opens the UNSW Memes.
2. Chat platform displays the user's chat list.
3. Chat platform displays the activity status of each friend in the list (e.g. online, offline, away) next to their profile image.

Success Scenario 2: Update friend activity status

1. Users change their activity status to "Away" or "Do not disturb" in user profile settings.
2. Chat platform displays the updated activity status next to the user's profile image for all the user's friends and the user.

Use Case: Manage Notifications and Alerts

- **Goal in context:** A user can snooze notifications and alerts for a set period of time
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account, is logged in, and has received notifications and alerts.
- **Success End Condition:** The user can snooze notifications and alerts for a set period of time.
- **Failure End Condition:** The user cannot snooze notifications and alerts for a set period of time.
- **Primary Actor:** User
- **Trigger:** The user receives a notification or alert

Success Scenario:

1. User selects the "Snooze" button in the notification setting.
2. Chat platform displays a list of snooze options (e.g. 1 hour, 24 hours, Always on).
3. User selects a snooze option.
4. Chat platform snoozes the notification or alert for the selected time period.
5. Within the snooze period, no notification appears.

Use Case: Manage Spam and Unwanted Messages

- **Goal in context:** A user can block and report spam and unwanted messages
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account, is logged in, and has received spam or unwanted messages.
- **Success End Condition:** The user can block and report spam and unwanted messages.
- **Failure End Condition:** The user cannot block and report spam and unwanted messages.
- **Primary Actor:** User
- **Trigger:** The user receives a spam or unwanted message

Success Scenario:

1. User receives a spam or unwanted message.
2. Chat platform displays the message.

3. User selects the "Block and Report" option.
4. Chat platform displays a confirmation message.
5. Chat platform blocks the future messages from the sender.
6. Users wish to unblock, and re-selects the "Block and Report" option.
7. Chat platform unblocks the future messages from the sender.

Use Case: Customise Colour Theme for Chat

- **Goal in context:** Allowing users to customise the colour theme of different chat and sync across all devices.
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** The user has an account and is logged in.
- **Success End Condition:** A user can customise the colour theme of different chat.
- **Failure End Condition:** A user cannot customise the colour theme of different chat Primary Actor: User
- **Trigger:** The user selects the option to customise the appearance of the chat platform

Success Scenario:

1. User selects the "Settings" option in a chat.
2. User selects the change colour theme option.
3. Chat platform displays a list of theme colours.
4. User selects the colours they want.
5. Chat platform saves the user's customization theme colours.
6. Chat platform displays the chat colours with the user's selected theme colours.

Use Case: New User Onboarding

- **Goal in context:** To provide new users with tutorials and guidance to navigate and understand the platform.
- **Scope:** UNSW Memes
- **Level:** Primary Task
- **Preconditions:** A new user has registered an account and has logged in for the first time.

- **Success End Condition:** The new user is provided with tutorials and guidance to navigate and understand the platform.
- **Failure End Condition:** The new user is not provided with tutorials and guidance to navigate and understand the platform.
- **Primary Actor:** New User
- **Trigger:** A new user registers an account and logs in for the first time.

Success Scenario:

1. A new user registers an account and logs in for the first time.
2. Chat platform displays a page that covers the most important features, settings, and navigation of the platform.
3. The page includes a "Close" button at the bottom of the page.
4. When the user clicks the "Close" button, the page disappears, and the user is returned to the home page.

Validation

John: These use cases are very helpful in addressing the problems I've been experiencing with the UNSW Memes chat platform. It provides the feature tag messages with personalised labels and search for them using the tags, allowing me to organise crowded messages. Also, message translation use cases resolve my difficulties of communicating with people who speak different languages. In addition, the use case of display friend active status enables me to check if other users are available to chat with ease. Hence, the use cases resolve my problem to a great extent.

Jane: The use case is comprehensive and well-thought, which resolves my problem to a great extent. The snooze notifications will be particularly useful for me, as I often find myself getting distracted by constant notifications while working. I'm also glad to see the use case for blocking messages from specific users, which resolves my issue of receiving unwanted messages and spam. In addition, use cases resolve the inability to change chat colour themes, which makes the platform more engaging to use. However, the new user guide can be more interactive and further assist the problem from new users to navigate the platform.

Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
<p>message/tag/create/v1</p> <p>Given a message sent to or sent by the authorised user, and a custom tag string, add a custom tag to the message. Only this user can read the message tag.</p>	POST	<p>Body Parameter: {messageId, tagStr}</p> <p>Return type if no error: {tagId}</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> · Invalid message id · Length of tag str is greater than 200 or less than 1 <p>403 Error when any of:</p> <ul style="list-style-type: none"> · Authorised user is not in the channel/dm that send the message · Invalid token
<p>message/tag/viewall/v1</p> <p>Given a tag string and channelId, return all messages with such a tag for the authorised user in the current chat. If no messages exist with such a tag for the authorised user, return an empty array.</p>	GET	<p>Body Parameter: { tagStr, channelId }</p> <p>Return type if no error: {messages}</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> · Length of tag str is greater than 200 or less than 1 · Invalid channelId <p>403 Error when any of:</p> <ul style="list-style-type: none"> · Invalid token

<p>message/tag/edit/v1</p> <p>Given a tag, and a new custom tag string, change the authorised user's tag for this specific message.</p>	PUT	<p>Body Parameter: { tagId, tagStr }</p> <p>Return type if no error: { }</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> · Invalid tag id · Length of tagStr is greater than 200 or less than 1 <p>403 Error when any of:</p> <ul style="list-style-type: none"> · Authorised user did not create the tag · Invalid token
<p>message/tag/delete/v1</p> <p>Given a tag, and a new custom tag string, delete authorised user's tag for this specific message.</p>	DELETE	<p>Body Parameter: { tagId }</p> <p>Return type if no error: { }</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> · Invalid tag id <p>403 Error when any of:</p> <ul style="list-style-type: none"> · Authorised user did not create the tag · Invalid token
<p>message/translate/v1</p> <p>Given a message that can be viewed by the authorised user, returned a translated message with input language type.</p>	POST	<p>Body Parameter: { messageId, languageType }</p> <p>Return type if no error: { }</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> · Invalid message id · Invalid language type <p>403 Error when any of:</p> <ul style="list-style-type: none"> · Authorised user is not in the channel/dm that send the message · Invalid token

user/setstatus/v1 Given new active status, set the authorised user' status to this input new active status.	POST	Body Parameter: {statusType} Return type if no error: { }	400 Error when any of: · Invalid status type 403 Error when any of: · Invalid token
user/getstatus/v1 Get and return the active status of the user with the uld. (such user is friend with the authorised user)	GET	Body Parameter: { uld } Return type if no error: {statusType}	400 Error when any of: · Invalid uld 403 Error when any of: · Authorised user is not a friend with uld user · Invalid token
setting/snooze/start/v1 Start a period of snooze based on the input snooze time, where the authorised user would not receive any notifications.	POST	Body Parameter: { snoozeTime } Return type if no error: { }	400 Error when any of: · Invalid soozeTime selection 403 Error when any of: · Invalid token
setting/snooze/active/v1 Return whether a snooze is active for the authorised user.	GET	Body Parameter: { } Return type if no error: { }	403 Error when any of: · Invalid token

<p>user/block/set/v1</p> <p>Give a messageId, block all future messages and notification from the user that sends the message (in the channel with such message) if the blockStatus is true.</p>	POST	<p>Body Parameter: {messageId, blockStatus}</p> <p>Return type if no error: {}</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> Invalid message id <p>403 Error when any of:</p> <ul style="list-style-type: none"> Authorised user is not in the channel/dm that send the message Invalid token
<p>channel/theme/set/v1</p> <p>Given a theme colour, set the channel's theme colour to the new colour.</p>	PUT	<p>Body Parameter: {themeColour, channelId}</p> <p>Return type if no error: {}</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> Invalid themeColour Invalid channelId <p>403 Error when any of:</p> <ul style="list-style-type: none"> Invalid token
<p>auth/register/tutorial/v1</p> <p>Display a tutorial page for important features on UNSW Memes, and display to newly registered users.</p>	POST	<p>Body Parameter: {}</p> <p>Return type if no error: {tutorialInfo}</p>	<p>400 Error when any of:</p> <ul style="list-style-type: none"> Token is from a user that already received the tutorial <p>403 Error when any of:</p> <ul style="list-style-type: none"> Invalid token

Variable Name	Type
Has suffix id	integer

Has suffix str	string
named exactly messages	Array of objects, where each object contains types { messageId, uid, message, timeSent }
languageType	string
snoozeTime	One of the follow: {"1 hour", "24 hour", "Always on"}
statusType	One of the follow: {"online", "offline", "doNotDisturb", "invisible"}
themeColour	string
tutorialInfo	Array of objects, where each object contains types {featureNameStr, descriptionStr}
blockStatus	Boolean

Conceptual Modelling - State Diagrams



