

COMP1531 Iteration 3

5.6 Planning for the next problems to solve

By T17B_ANT: Angie Liu, Chelsea Arezo, Florenz Elvin Manaois, and Harsh Bhatia

[*Requirements*] Elicitation

User 1:

Name: Alvaro Arezo

Email: alvaro@arezo.org

User 2:

Name: Zane Artesi

Email: zane.artesi@hotmail.com

User 3:

Name: Aleeya Bano

Email: Aleeyabano@gmail.com

Interview

What is a feature you like seeing in your collaboration tools such as Microsoft Teams, Skype, and Discord? e.g. voice channels, reacting to messages

User 1: "Video calling, group calls, presentation mode and screen sharing, online presence"

User 2: "Creating and leaving chats, message reacts, file sharing and co-editing, screen sharing and screen control"

User 3: Microsoft Teams allows access to collaboration documents and PowerPoints, reacting to message features on Skype, ability to make multiple communication channels easily on Discord.

What is a feature you dislike in collaboration tools that you use?

User 1: "Teams whiteboard is clunky, it would be great to have a feature-rich whiteboard capability"

User 2: "I dislike when Teams messages are automatically being marked as read when I tab into the app to view the message. Also the search functions in teams don't allow you to see messages sent before or after your search like in Facebook Messenger."

User 3: "Lack of personalisation, I like to be able to have multiple communication channels even if it with the same people but only Discord offers this feature, I like to be able to modify the looks of my collaboration tools (changing themes ect) all the current options on Skype ect are limited."

What will you primarily be using Seams for?

User 1: "Presenting to my team, messaging, voice calls, team chat"

User 2: "Collaboration with other students for group projects"

User 3: "Communication Channel for university (working on assignments, collaborating on quizzes etc)"

Are there any features you'd like to see in Seams?

User 1: "It must have an immersive user experience, it can be feature-rich but the top features must be easy to use and intuitive. Custom video backgrounds, maybe a rolling

giphy style background would be interesting. Integration with other tools like SharePoint would also be good.”

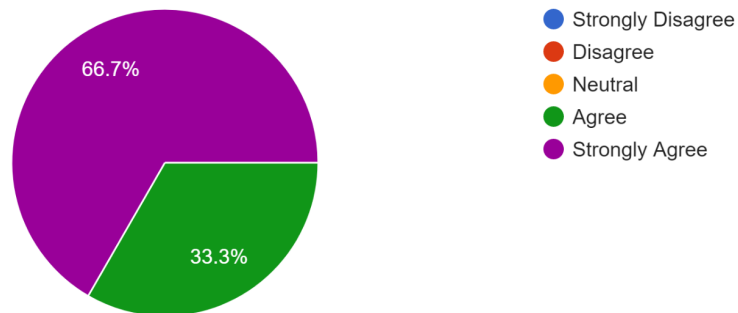
User 2: “Voice messages and the ability to view messages before they are marked as read and task setting and tracking abilities”

User 3: “Personalisation, customising the appearance on Seams, ability to create multiple communication channels, ability to work on a shared document (notes page?).”

For the next few questions, please select an option depending on whether or not you agree with the statement provided:

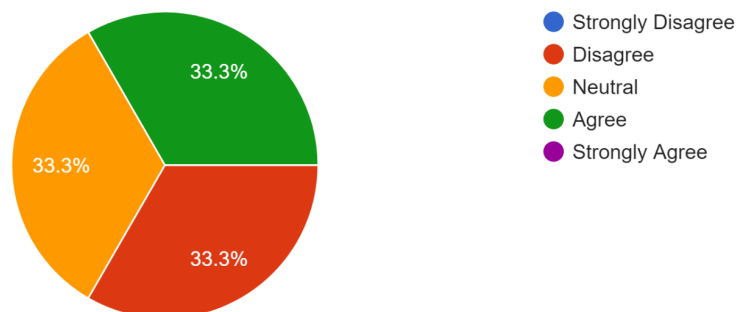
Apps such as Seams should have designs and artwork such that it is pleasing to the eye

3 responses



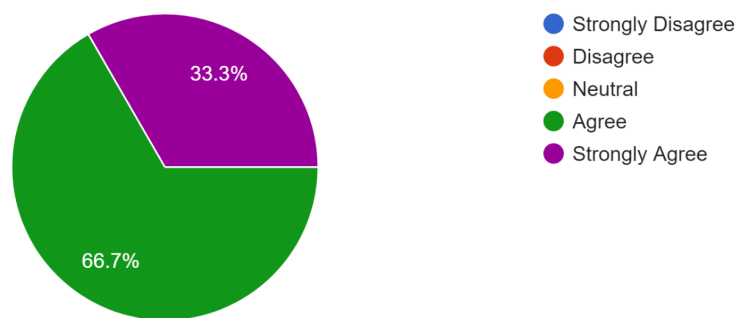
Business related applications such as Seams should employ a minimalistic theme in order to remain professional

3 responses



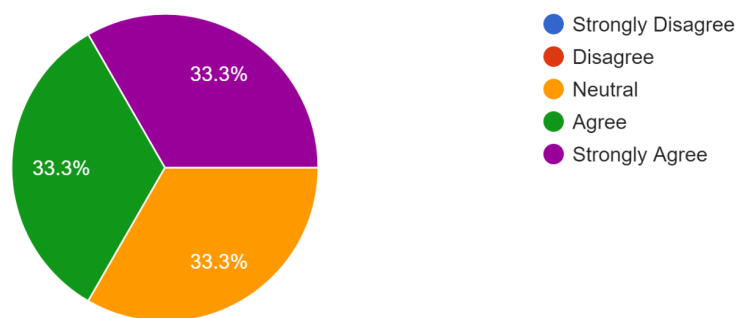
Darker themes are better than lighter themes

3 responses



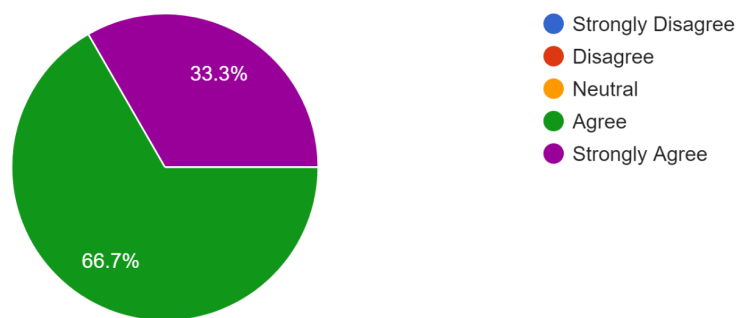
I like being able to customize the appearance of apps that I use such as Seams

3 responses



Apps such as Seams should not be 100% dedicated to work- I want there to be other features that make the experience more fun.

3 responses



Our Solutions

User	Problem	Solution
1	<ul style="list-style-type: none"> - Needs a whiteboard that isn't clunky - Incorporate external tools such as Sharepoint 	<ul style="list-style-type: none"> - Implement a function that would generate a whiteboard which will undergo a large amount of testing to reduce bugs. - Make Seams compatible with Sharepoint accounts
2	<ul style="list-style-type: none"> - Preview messages without it being marked as read - Search for a message and brings up the conversation attached to the message (messages before and after) - Voice messages - Task setting/tracking features eg due dates and checklists 	<ul style="list-style-type: none"> - Incorporate action listeners that only trigger a message being read when a user clicks any point on the screen with a dm / channel open, this will prevent any accidental reading when a user alt tabs into Seams. - When searching for a message, also show surrounding messages, as messages are already in chronological order this can be done by returning more than just the selected message - Provide an option that allows users to record and send a message, which would be handled like a regular text message with a few changes - Implement a calendar on Seams home screen that also allows users to add specific tasks on different dates, allowing them to track their tasks
3	<ul style="list-style-type: none"> - Option of sub-channels with same members in a given channel (for organising different topics in university) - Ability to modify the theme of Seams - Ability to work on shared documents through Seams eg notes page 	<ul style="list-style-type: none"> - Given a valid channel provide an option to create new sub channels where the authorised user enters a sub channel name and all team members can now communicate on - Users can go into settings and select the themes section where they can choose different options and preview the theme and confirm their selection. - Make an inbuilt document editor that is shared amongst users in a channel where they can edit and view the document.

[Requirements] Analysis & Specification - Use Cases

User stories

User 1:

As an Enterprise Architect, I want to directly message my team and other stakeholders, so that I can present my work to those who are overseas and be able to stay in the loop of any updates relevant to myself.

ACCEPTANCE CRITERIA:

Scenario: Sending a message to team member

Given: The user has logged in with a valid account

When: The user has selected a valid dm with the colleague he wishes to communicate with

And: Entered a valid message to a valid user

Then: The dm is created and the message is sent

Given: The colleague has received a notification of the message

When: The colleague opens the specified dm

Then: The colleague can act on the message and reply

User 2:

As a USYD student, I want to be able to leave channels that I was invited to from different subjects/projects when it is the end of my semester, so that I can maintain an organised setup with only relevant, active channels remaining, allowing me to keep track of the chats.

ACCEPTANCE CRITERIA

Scenario: Leaving a channel from a completed project

Given: The user has logged in with a valid account

When: The user goes to select the existing channel

And: Enters a valid channel

Then: Proceeds to leave the channel

Given: The user who left is not the owner of the channel

When: The user left the channel

Then: The remaining members of the channel are the users who have not yet left with all messages remaining

User 3:

As a student, I want to be able to create separate sub-channels for each subject/project when my semester begins, so that I can easily communicate with each team and therefore finish projects more efficiently.

ACCEPTANCE CRITERIA

Scenario: Creating a channel with new team for new subject

Given: The user has logged in with a valid account

When: The user goes to create a new channel

And: Enters a valid name

Then: Creates the channel and proceeds to invite his team

Given: The channel created is valid

When: The user invites his team to join the channel

And: Enters valid user accounts

Then: The team are now all members in the channel and can all receive notifications and communicate

When: The user goes to create sub-Channels within channel

And: Enters a valid sub channel name

Then: The team now has a new sub channel where they can collaborate on

Use cases

Use case 1

CREATE CHANNEL SUCCESS SCENARIO

Step 1: Once user opens app, it will ask for login details

Step 2: User enters valid credentials / or creates an account with valid credentials

Step 3: Once logged in user will be directed to the home page

Step 4: User selects to create new channel from the side panel

Step 5: Page navigates to a form that asks for the name of channel, and if it is to be secret or not secret

Step 6: User enters valid name and creates channel

Step 7: User selects created channel

Step 8: User invites a valid user to become member of channel

Step 9: User sends message to user through channel

Step 10: Other user receives notification of message

FURTHER FEATURES SCENARIO

Step 11: User goes to settings and can go to the themes section where they can browse through different options and pick out a theme they like

Step 12: User goes on to their created channel

Step 13: User selects to create new sub channels

Step 14: User enters a valid name and creates a sub channel to separate some topics

Step 15: User selects to create new collaborative document within channel

Step 16: Other users can edit and view this document

Step 17: Users can select and choose drawing tools on this document to simulate a whiteboard like experience.

Use case 2

LEAVE CHANNEL SUCCESS SCENARIO

Step 1: Once user opens app, it will ask for login details

Step 2: User enters valid login credentials

Step 3: Once logged in user will be directed to the home page

Step 4: User selects an existing channel that they are a member of from the side panel

Step 5: User selects "Leave Channel"

Step 6: User is no longer member of channel

Step 7: Channel disappears from "My Channels"

[*Requirements*] Validation

User 1: The scenario presented meets my requirements to not only message but actively collaborate with my teams globally. The ability to change the look and feel through customisable themes also meets the brief well. The further features scenario also meets my requirement to use shared whiteboards inside the product to assist in brainstorming and strategy development with my teams. Overall the scenarios will meet my requirements.

User 2: The scenario relating to successfully leaving a channel accurately describes the original issue I discussed about leaving a channel. Much like other communication platforms, once I submit to leave a channel I expect it to no longer be visible, accessible or able to notify me of changes. Therefore, I agree that the use case I was shown adequately reflects a solution to the issue of leaving a channel.

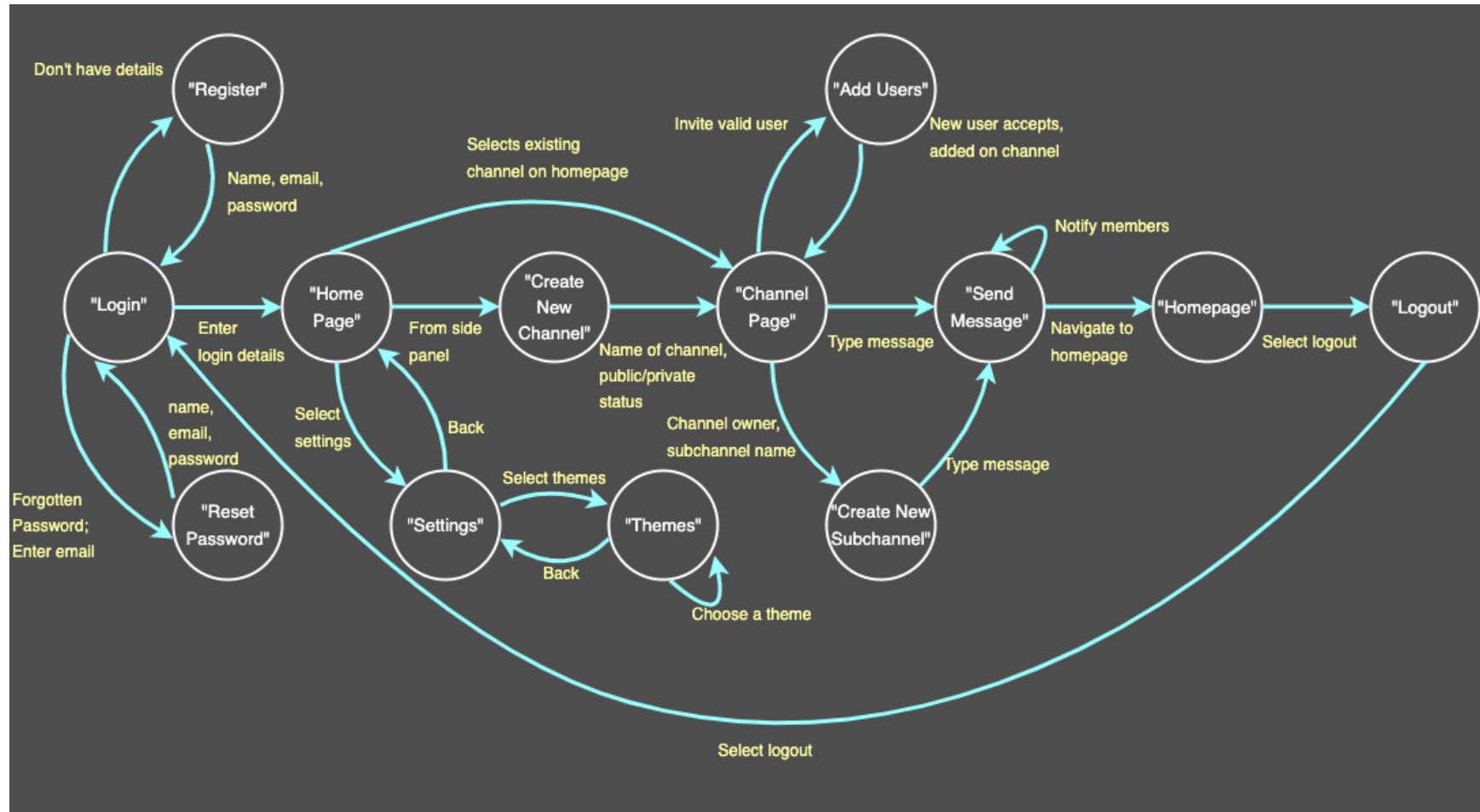
User 3: The seams channel create success scenario matches the requirements of a basic team communication application. It solves the need for an application where I can easily add team members and begin sending messages to my team where my team members are notified and can reply back with ease. The further features scenario expands on the core requirement and solves the problems which were initially raised. I now have the ability to create sub channels to differentiate a range of topics, for example a notes sharing channel, questions channel, and a general chat channel. As a student this helps manage projects better. The feature to change themes provides a solution to the problem of customisable of different messaging platforms, and makes an overall better experience. Finally as a student the ability to share documents with ease is an appealing feature that this use case has solved. Overall, I am satisfied with the use case and it solves the range of problems/ features I had initially voiced.

[Design] Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
<p>channel/newsubchannel/v1</p> <p>Given a channel with ID <code>channel_id</code> where the authorised user has owner permissions, creates a new sub channel within the existing sub channel with the given name. All members of the channel have access to view this new sub channel</p>	POST	<p>Parameters: {token, channel_id, name}</p> <p>Return type: {}</p>	<p>Input Error:</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>Access Error:</p> <ul style="list-style-type: none"> channel_id is valid and the authorised user does not have owner permissions in the channel Auth user does not exist
<p>channel/whiteboard/v1</p> <p>Given a channel with ID <code>channel_id</code> where the user is a member of the channel, generates a whiteboard in meetings. All members in the channel have access.</p>	POST	<p>Parameters: {token, channel_id}</p> <p>Return type {}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Channel_id does not refer to a valid channel <p>AccessError:</p> <ul style="list-style-type: none"> User is valid but is not member of channel
<p>messages/messageread/v1</p> <p>Given a message id <code>message_id</code> where the user is a member of the channel or dm associated with that message id, shows that the user has clicked on the message through action listeners.</p>	GET	<p>Parameters: {token, message_id}</p> <p>Return type {message_read (boolean)}</p>	<p>InputError:</p> <ul style="list-style-type: none"> Message_id is not a valid message <p>AccessError:</p> <ul style="list-style-type: none"> User is not a member of the channel/dm that the message id is associated with.
<p>conersationsearch/v1</p> <p>Given a query string, return a collection of messages in all of the channels/DMs that the user has joined that contain the query</p>	GET	<p>Parameters: {token, query_str}</p> <p>Return type {messages}</p>	<p>InputError:</p> <ul style="list-style-type: none"> length of query_str is less than 1 or over 1000 characters

(case-insensitive) and the messages before and after the highlighted message. There is no expected order for these messages.			
<p>messages/messagesendvoice/v1</p> <p>Send a voice message from the authorised user to the channel specified by channel_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, message }</p> <p>Return type {message_id}</p>	<p>InputError when any of:</p> <ul style="list-style-type: none"> channel_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 is valid and the authorised user is not a member of the channel
<p>messages/messagesendvoicedm/v1</p> <p>Send a voice message from authorised_user to the DM specified by dm_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 or DM.</p>	POST	<p>Parameters: { token, dm_id, message }</p> <p>Return type {message_id}</p>	<p>InputError when any of:</p> <ul style="list-style-type: none"> dm_id does not refer to a valid DM length of message is less than 1 or over 1000 characters <p>AccessError when:</p> <ul style="list-style-type: none"> dm_id is valid and the authorised user is not a member of the DM
<p>calendar/v1</p> <p>Calendar that is created when a user registers an account (auth/register/v1). The user can add optional due dates and checklists.</p>	POST	<p>Parameters: { token, time_due, task_str (string)}</p>	<p>InputError when any of:</p> <ul style="list-style-type: none"> Time_due does not refer to a valid date on the calendar Length of task_str is less than 1 or over 1000 characters <p>AccessError when any of:</p> <ul style="list-style-type: none"> Token does not refer to a valid user

[Design] Conceptual Modelling (State)



Draw.io link - <https://tinyurl.com/yz9wt2sa>