



Requirements and Design

UNSW Streams

T11C_CAMEL

- Blake Morris
- Redmond Mobbs
- Trissha Varman
- Vu Luu
- Tyler Gan

Requirements

Elicitation

Interview Questions

1. Which team communication software have you used in the past?
2. In what situations did this software not allow you to communicate in the way you would like to, or make it inconvenient or difficult?
3. What difficulties have you had in using the software for organisation and planning?
4. Has the scale of projects or team sizes caused any issues with communication using the software?

Interview Responses

Temmie Yao (t.yao@unsw.edu.au)

1. I've used slack for both cse tutoring and for my part time job. Also discord for assignments & hackathons

The places where i use slack i also use teams for video/voice calling

2. one common issue with slack is the need to have another service for video/voice calling. work uses teams for this, and integrates alright with slack -- you get notified of scheduled meetings etc

one issue I've noticed with both slack and discord is sometimes i read something then move to another channel then can't find it again (since i didn't see which channel the thing was in)

and one minor issue with slack is that you can't add people to dms after creating them

though in general both are pretty good for organising and coordinating stuff

also, discord's file size limit can sometimes get in the way

Slack handles uploaded files pretty well

i know you didn't ask, but I'm gonna mention some of the good parts:

- being able to start a new channel easily is quite handy. it allows people to be more independent

- similarly, automated alerts using bots/webhooks -- work uses this extensively to catch human error (e.g. forgetting to update data for a new month) and machine error (e.g. reporting automated actions so you can notice if something is going wrong)

- slack has pretty good notification functionality, such as being able to get slack to remind you of messages at a point in time, as well as disabling notifications outside a particular time range (i don't want work to grab my attention when i'm not working)

one issue with larger teams where people are working on multiple things is that it's hard to tell what's going on at a high level, but this is more of an organisational/project management thing than a chat thing

but you also don't need to be aware of everything that's going on i guess

also: slack and discord are not good for long-lived documentation. it's good for communication but terrible for preserving information when you revisit something in e.g. a year, or are onboarding a new person

I love having other apps and tools we use being interrogated with slack/discord -- teams is one example. but i think communication apps shouldn't try to solve all problems with organisation -- having some shared document store like notion or confluence is very important for large or long-running teams, and i don't think chat apps should try to solve this. maybe something like being able to link to a bunch of slack messages from confluence/etc would be helpful, i'm not sure

Luke Fisk-Lennon (luke.fisklennon@gmail.com)

Which team communication software have you used in the past?

1. I have used Slack and Discord for team communication.

2. I found that Slack's implementation of multi-workspace support made it inconvenient to join a new workspace, which requires creating an entirely new account and profile. It is also hard to manage and switch between these workspaces in the web app, which only allows you to use one workspace easily, unlike its desktop and mobile counterparts.

Discord is largely free of this issue, but has its own. Inviting new users to a server can be a convoluted and unintuitive process. For example, in COMP2521 20T2, we were allowed to use Discord for group communication. Students would send server invite links to jas, who would then join these servers with the course account. However, short default expiry times meant that many of these invites would expire before being used, leading to a negative impression of the platform.

3. While threads are a useful feature in Slack, their abstraction of synchronous and real-time conversations makes it more difficult to get a sense of what discussions are ongoing. This problem becomes more apparent as teams scale and discussions become more frequent.

Discord suffers from the opposite problem to Slack, with a lack of abstraction in the conversation history. While it also has threads, these are rarely used compared to threads in Slack, due to requiring a more involved process to create them. Resultantly, most discussions occur at the top-level, making them harder to easily gather an overview of. This issue also worsens with scale, as the volume of messages increases, to the point where it is impractical to digest asynchronously.

4. As I have only been involved in small projects and teams, scale has not become an issue. However, I can foresee problems arising, as mentioned above.

A problem identified in these responses is that it is difficult to see an overview of the communications occurring at any point.

Proposed solution

An overview page, that displays a summary of information for each channel the user is a member of:

- The current topic of discussion, which a channel owner would be able to specify and change
- The most recent pinned message
- The collation of the most recent standup
- Recently joined users
- Any recent messages with a large number of reacts

These summaries are saved, and the user can access summaries from a particular point in time with a search.

Analysis and Specification

User Stories and Acceptance Criteria

As a user of communication software, I want to find things I've previously read so I can access important information.

Given that a user has used the communication service over a long period of time
When they need to access important information from previous communications
Then they are able to find the information quickly and conveniently

As a person communicating with a team, I want to be able to keep track of high-level details of a project so that I can be informed of the context of my work.

Given that a user is working on a project in a team
When they wish to see details of the overall project
Then they are able to access this information within the communications platform

As a user of collaboration software, I want to be able to organise and access previous communications, so that I can revisit past information and inform new team members of important information.

Given that a user has used the communication service over a long period of time
When they need to access important information from previous communications
Then they are able to find the information quickly and conveniently

As a person working on multiple projects, I want to be able to switch between them easily so I can conveniently move between projects.

Given that a user is using the communication service for multiple projects
When they want to change which program they are currently working on
Then they are able to do so without having to log in again

As a user, I want to be able to see what discussions are occurring while also not having them all occur in one place, so I can understand an overview of the communications that are taking place.

Given that a user is working on a project involving many communications
When they wish to know what discussions are occurring
Then they are able to do so without having to read through long conversation histories

Use Cases

Main Success Scenario

1. User logs into UNSW Streams
2. User navigates to the 'Overview' section using the frontend
3. Application displays consolidated information about each channel the user is in
4. User reads the information to understand the high-level details of their communications
5. User clicks on a specific channel overview
6. Frontend displays that channel

Background

Goal in context: User needs to understand a high-level communications overview

Scope: UNSW Streams

Level: Primary task

Preconditions: User is a member of Streams, and of channels which have previous activity

Success End Condition: User is presented with the activity overview

Failure End Condition: There is no information to display

Primary Actor: Streams user

Trigger: User navigates to 'Overview' section using the frontend user interface

Validation

"That definitely sounds useful!"

- Temmie

"I think that would be very helpful, because it can be tedious checking each channel individually."

- Luke

Design

Interface Design

`overview/get`

Provides an overview of every channel the authorised user is a member of. If the user is not a member of any channels, the list returned is empty.

HTTP Method: GET

Parameters:

- *token* - authorisation token of the user (str)

Returns:

- *overviews* - list of dictionaries for each channel, containing:
 - *channel_id* - id of the channel this overview relates to (int)
 - *channel_topic* - current discussion topic (str)
 - *recent_pin* - id of the most recently pinned message (int)
 - *recent_standup* - id of the most recent standup message (int)
 - *recent_joins* - list of ids of any users who joined in the last 24 hours (int list)

Exceptions:

- `AccessError` raised when:
 - Token is invalid
-

`overview/getold`

Provides the overview information for a time in the past.

HTTP Method: GET

Parameters:

- *token* - authorisation token of the user (str)
- *date* - date from which to show overview information

Returns:

- *overviews* - list of dictionaries for each channel, containing:
 - *channel_id* - id of the channel this overview relates to (int)
 - *channel_topic* - current discussion topic (str)
 - *recent_pin* - id of the most recently pinned message (int)
 - *recent_standup* - id of the most recent standup message (int)
 - *recent_joins* - list of ids of any users who joined in the last 24 hours (int list)

Exceptions:

- `AccessError` raised when:
 - Token is invalid
 - `InputError` raised when:
 - Date is not in the past
 - Date is earlier than the deployment date
-

`channel/settopic`

Sets the topic message of a channel. This is displayed when viewing this channel, and in overviews. The default topic is an empty string. Setting the topic to an empty string clears it.

HTTP Method: PUT

Parameters:

- *token* - authorisation token of the user (str)
- *channel_id* - id of the channel to set the topic message in (int)
- *topic* - message to set as the channel topic (str)

Returns:

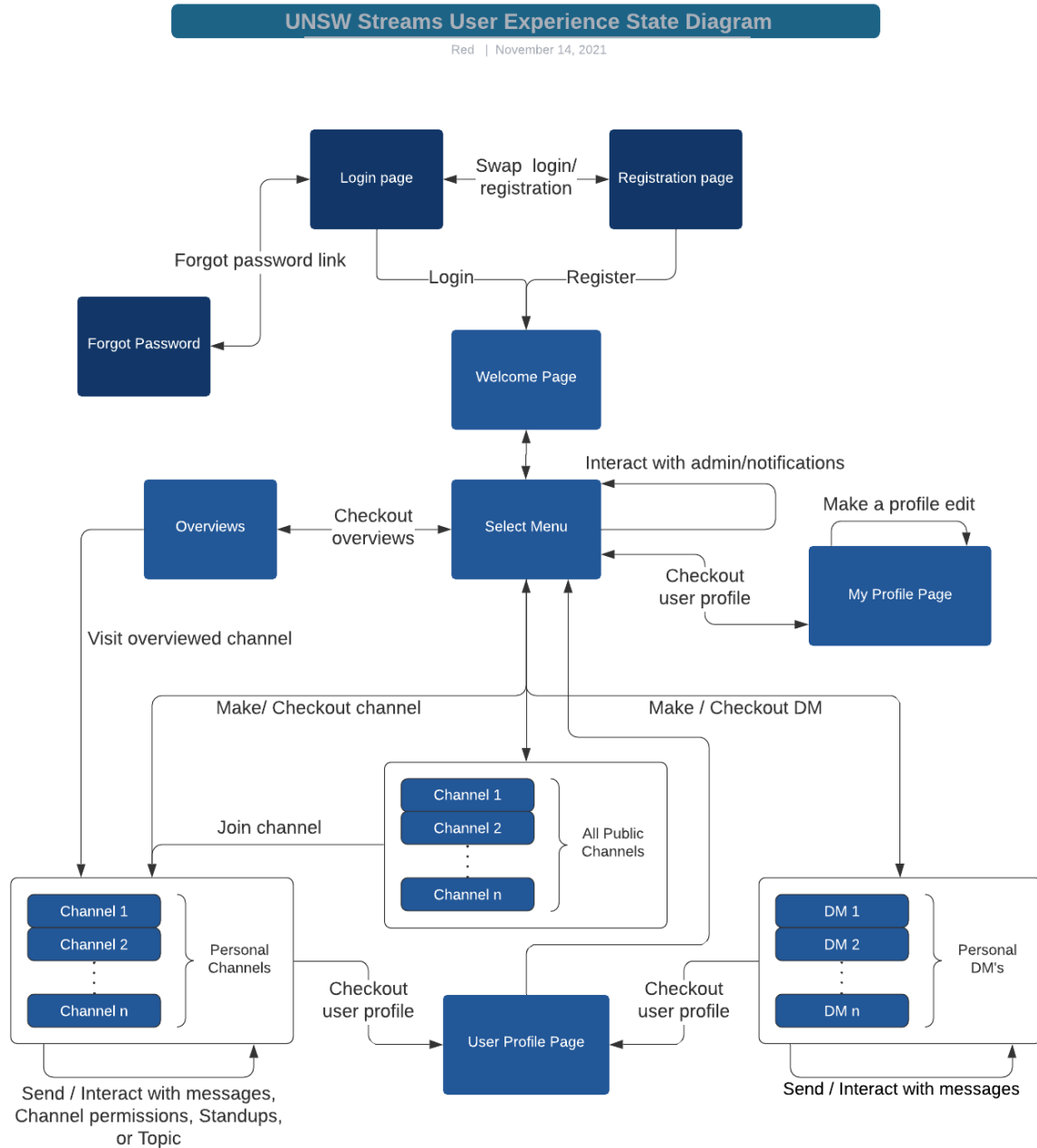
- None

Exceptions:

- `AccessError` raised when:
 - Token is invalid
 - User is not an owner of the (valid) channel
- `InputError` raised when:
 - Channel id does not refer to a valid channel
 - Length of topic message is more than 50 characters

Note: topic would also be added as a return value from `channel_details`.

Conceptual Modelling - State



Notes:

- The *Select Menu* state refers to user interactions with the side menu and header menu available once a user is logged in.
- Exit functionality is available from all states of the system (whether by logging out or by closing the connection to the server), and has been omitted for readability