Planning Document - H13B Eagle

Requirements

Elicitation

Joshua Chew

Email: jchew279@gmail.com

What do you turn to for teamwork-driven communication applications? Discord? Microsoft Teams?

I guess I use Discord for most of my teamwork activities.

What do you like about this application?

It is relatively easy to use, with lots of useful functions. You can add, message and call friends. You can create servers and even customise them. I think they were effective in their usability overall.

What is your biggest issue with this application? What are the challenges you encounter when using this application?

- **Messages?**
- **DM messaging?**
- **Photo uploads?**
- **Channels?**
- **UI?**

I think the main one for me are notifications and trying to learn how to control notifications to my own will, y'know.

I didn't like how I could just view one chat rather than looking at multiple at once.

But also, it was very annoying figuring out how to use bots and how to add them in and how to configure them. I think if they integrated some of the features that bots provide into their application and made it more understandable for us to use.

What if we improved this aspect? Would you consider using Streams over your current application?

If there was more notification control, plus utilisation of what bots provide and if there's a way of showing multiple chats at once, then I would consider it. However to be honest, I would still use Discord, just because I am still used to that application. But I would definitely consider Streams.

If you could change anything about this application, how would you change it?

- I reckon you should start thinking big perhaps, you can send photos or gifs.
- I know it's not a backend thing, but can you perhaps make the UI better?

What features do we do well compared to other competitors?

- The delayed message is pretty good. It's almost like scheduling an email. I don't really see that in discord other than adding a bot so I really enjoyed that feature.
- I also liked the standup feature. I don't really use it and I'm still not sure of its full capability but it's definitely not seen in Discord and so I think that's a feature that is unique.
- Statistics can be good, tracking time and activity. Shows when you're using up too much time, being productive, etc.

Jessica Tran

Email: jessicatran0210@gmail.com

What do you turn to for teamwork-driven communication applications? Discord? Microsoft Teams?

- Typically I use messenger for uni work. I don't use Teams because it seems very business orientated.
- Microsoft Teams has also never been a requirement for my courses. As a result, it's very unfamiliar and it's less accessible by the general public.

What do you like about this application (Messenger)?

- I like the popup notifications, the emoji reactions.
- There's read/received which is helpful in a group setting to see who's active at the moment.
- I like that you can call, share photos
- Can send voice messages, send videos,
- Sharing facebook posts.

What do you like about this application (Streams)?

- Looking at the profile page, I thought it was strange that the profile was located on the side bar as opposed to being at the top.
- I like the user statistics because I like seeing data from apps and its nice to see that it tracks your information, your involvement, your usage.
- I like the timer and it differentiates itself from other messaging apps. (Can schedule ahead of time)

What is your biggest issue with this application? What are the challenges you encounter when using this application? (Messenger)

- **Messages?**
- **DM messaging?**
- **Photo uploads?**
- **Channels?**
- **UI?**

- Constantly evolving so it's difficult to keep up with the features.
- Aside from that I don't have any issues with messenger.

What if we improved this aspect? Would you consider using Streams over your current application? (Messenger)

- Yeah as long as it's very clear what the new implementation does and how to use it.
- Visually concise and limited steps to achieve the result.

What is your biggest issue with this application? What are the challenges you encounter when using this application? (Streams)

- **Messages?**
- **DM messaging?**
- **Photo uploads?**
- **Channels?**
- **UI?**

- Confused on what the standups would be good for.
- Instead of displaying "9 Minutes ago" just display the time.
- Invite_member and leave_channel in an unideal place. Should be at the bottom of the page.
- When you invite people to a channel, people should have the option to accept or decline.
- Can't invite members to DMs is annoying because you'll create a bunch of redundant DM's.
- Questioning the purpose of a global owner in the context of a messaging app. Why should the first person signed up be able to see everything?
- If you provide no msg when you edit something, it deletes it? Weird functionality because you already have the "remove" msg feature.
- The symbols are confusing (i.e. promoting someone to owner looks like an add friend)
- Tagging system requiring handle_str doesn't seem right? You can't even see the user's handle_str unless you click on their name.
- Tagging doesn't have a UI. Makes it hard to tag people
- I'd like to have a list of members which shows all the members who have signed up to streams.

- A popup that occurs or some sign that shows that you have notifications.
- Should have the ability to search for messages within a specific channel/dm. The current search function searches for a msg within the entire streams application.
- Why are private channels still visible to other users. (The name and its existence)

What if we improved this aspect? Would you consider using Streams over your current application? (Streams)

The reason why I use messenger is because i directly get notifications. Streams however doesn't provide direct notifications and I need to physically be in the app to see the notifications.

Hence if this isn't a feature, I would not consider using Streams over Messenger.

If you could change anything about this application, how would you change it?

- UI needs to look a lot better
- Blocking other users.
- Have something similar to discord where it gives instructions on how to navigate the website. Very hard to understand without someone telling you what is what.

What features do we do well compared to other competitors? if so, what?

- Definitely the stats feature (transparency about tracking you)
- Could improve the stats by adding more things (amount of likes received)
- Most photos sent, etc
- Make this the point of difference between our competitors.
- Timed messages (very unique no other messaging apps have this)
- Like that standups are an option. (i don't see it being used elsewhere).
- Good in the classroom setting, have people jot down their thoughts whilst watching a video for example, and then have the standup timer end once the video ends and all their thoughts whilst watching get published. Prevents spamming.
- Have the time of when a standup message was sent. (Forces engagement(
- Everything else is just a standard message app

Max Cochrane

Email: maxcochrane23@gmail.com

What do you turn to for teamwork-driven communication applications? Discord? Microsoft Teams?

I appreciate the nuanced form in which discord helps interact with colleagues and other fellow individuals with a socially laxed environment, and Microsoft teams takes the sophistication and sets it into a professional manner (however with less features). Overall, I prefer discord much more.

What do you like about this application?

The ability to not only engage in text but also contains quite robust voice and video call feature and also the ability to make restricted subchannel and broader channel applications is quite appealing.

What is your biggest issue with this application? What are the challenges you encounter when using this application?

The auto updates checker will delay opening discord.

What features do we do well compared to other competitors? if so, what?

The ability to send messages after a select amount of time and start stand ups is great to organize meetings and other formal gatherings. As well as the ability to track the user statistics well give us a better understanding of the activity.

What features would you want us to add?

Similar to Discord, streams should allow robust voice features in order to improve communication between dm and channel members.

No live update of the stand-up raises issue in term of achieving its purpose as a meeting like chain of messages. So, in the case of a stand up, live updates of the stand-up messages appear while the meeting lasts.

Recurring messages that act as pre schedule reminders can be added that ultimately act as a reminder notification for a later date in which a stand up occurs or something along the lines of a meeting occurs at the time. The bonus of having it be recurring suggests it could act as a reminder for said event.

Analysis & Specification - Use Cases

Once you've elicited this information, it's time to consolidate it.

Take the responses from the elicitation and express these requirements as **user stories**. Document these user stories. For each user story, add user acceptance criteria as notes so that you have a clear definition of when a story has been completed.

Once documented, generate at least one use case that attempts to describe a solution that satifies some of or all the requirements elicited. You can generate a visual diagram or a more written-recipe style, as per lectures.

User Stories

As a user, I want to have more notification control, so that I don't have to get annoyed with pesky irrelevant notifications.

- Have a drop-down menu for notifications
    - All notifications
    - Only tags - @everyone and @me
    - No notifications
- This menu will be on every channel and DM
- They can also choose to disable notifications from specific users as well.

As a user, I want to display multiple chats at once, so that I don't have to constantly switch back and forth with channels and DMs.

- If the user is only viewing one channel/dm, it takes up the entire screen/window.
- However, if they were viewing two, it would get split into two.
- They can view up to four channels/dms at a time.
- The top right of each channel/dm is an 'X' button to close the chat.

As a user, I want to display photos onto messages so that I can message other things that aren't just words.
- The user could either:
    - Paste in the url of the photo onto the type box and it will be processed and then outputted onto the message
    OR
    - Copy and paste the actual photo onto the type box and it will be processed and then outputted onto the message.
- There will be a limit of 100MB on the size of the submitting photo.

As a user, I want to have instructions on how to navigate the website so that I don't feel lost whilst using the application.

- Once a user signs up with Streams, they will be greeted with an informative UI that explains what each button does.
- Each explanation can be closed by pressing the "x" button to close the popup.

- There will also be a button called "Skip Tutorial". Once the user clicks on it, all pop-ups containing explanations on how to use Streams will disappear.
- There will be a button "Help" to revisit the tutorial which when clicked, will replay the informative UI as if they are a new user.

As a user, I want to have a more appealing and informative UI so that I am able to navigate around with ease and enjoy using Streams.

- Each button also contains a logo/photo that accurately represents the action.
- There will be a "Help" button that upon being clicked by the user, will bring up an interface with information on how to use Streams.
- Users will also be able to add and change channel and DM profile pictures.
- Within the member's list of a particular channel/DM, it will display their handle string underneath their first and last name.
- There will be a button that looks like a palette which allows users to customise their interface whilst using Streams.

As a user, I want to have the ability to block users to safeguard myself from potential harassment.

- The member's list of a channel/DM is displayed once the user clicks on a specific channel/DM.
- Next to each member's name there is a button "Block".
- Once the user clicks "Block", the block button changes to "Unblock" and the blocked user can no longer contact the user through DM's, channel invites or messages.
- The user cannot block the same user twice.

As a user, I want the ability to have voice communication with my friends on UNSW Streams so that we can have more freedom in the aspect of communication.

- Near the name of the channel/dm there is a phone call symbol.
- Once the user clicks this button then he will initiate a voice call with the members of the said dm or channel.
- The person can not force other users, in channel or dm, to join the call as it is voluntary.

As a user, I want there to be live updates of messages in stand up to better fulfil the objective of a stand up as well as see first hand user participation.

- As a stand up occurs, for a set amount of time, any message into the supposed stand up will show live updates of said messages sent.
- This should take the form of a message similar to those sent to a dm or channel.

As a user, I want there to be a recurring notification for and stand up to better organize meetings as well as improve overall organisation.

- As a stand up occurs, for a set amount of time, a message will be sent in a set amount of time, which notifies that a stand up will occur.
- The start up cannot be set by a member not in a channel.
- The message is a form of a message but has limits to the number of characters.
- Does not double notify a user, i.e. if the message in said notification includes a tag.

As a user, I want there to be a method for scheduling future stand ups to better organisation of any and all members taking part and improve overall organization.

- The stand up can be set , by input, to schedule a standup to start at a specific minute.
- The startup must be set by someone in the dm or channel
- This function works with recurring message function to delivery repeating messages
- Failures arise in input such as ibnvalid months, day or times

Use Cases

Use Case 1

- **Use Case**: Block Users
- **Goal in Context**: User needs to block another user as well as their messages on mutual channels and DMs.
- **Scope**: UNSW Streams
- **Level**: Primary Task
- **Preconditions**: Both users are valid.
- **Success End Condition**: The blocked user is now blocked. All messages from the blocked user will be blocked.
- **Failed End Condition**: The user to be blocked is already blocked.
- **Primary Actor**: A user of Streams
- **Trigger**: The user clicks on the "Block" button located next to another user's name.

The blocker will not be able to see the blockee's messages as they are substituted with a "Blocked User" name and a "Blocked Message" message. The blockee will still see the blocker's messages and name.

MAIN SUCCESS SCENARIO

Step 1. Registered user logs in.
Step 2. Registered user joins channel/DM.
Step 3. Registered user clicks on the 'block' button next to the selected user.
Step 4. Server receives command and places 'blocked' user in registered user's blocklist.
Step 5. All messages from 'blocked' user will be substituted with 'Blocked Message' and their names will also be substituted with 'Blocked User'. Their profile picture will be changed back into the default image. Only the registered user will see this. The registered user cannot see the 'blocked' user's name when creating a new channel/DM.

Use Case 2

- **Use Case**: Send photos
- **Goal in Context**: User needs to send photos to channels and DMs.
- **Scope**: UNSW Streams
- **Level**: Subfunction
- **Preconditions**: User is valid and registered. User is in the channel or DM.
- **Success End Condition**: User sends a valid photo that is within the size limit in the channel or DM.
- **Failed End Condition**: User doesn't send a valid photo. User sends a photo/gif above size limit.
- **Primary Actor**: A user to Streams
- **Trigger**: If the message includes a link to a photo or they have submitted a local file of photo/gif.

MAIN SUCCESS SCENARIO 1

Step 1. Registered user logs in.
Step 2. Registered user joins channel/DM.
Step 3. Registered user clicks on the photo button.
Step 4. A File Explorer/Finder window opens up.
Step 5. Registered user selects photo
Step 6. Streams processes photo and outputs it into a message on the channel/DM.

MAIN SUCCESS SCENARIO 2

Step 1. Registered user logs in.
Step 2. Registered user joins channel/DM.
Step 3. Registered user types in a URL link to photo.
Step 4. Streams processes photo and outputs it into a message on the channel/DM.

Use Case 3

- **Use case:** I'm a user who wants a recurring message for a stand-up feature in UNSW streams.
- **Goal in Context:** Inclusion of a feature that send recurring messages at given time intervals similar to a notification for a stand up that occurs at some given interval for a set amount of time which could start more than 24 hours away from being initially set.
- **Scope:** UNSW Streams.
- **Level:** primary task.
- **Preconditions:** Connected successfully in a dm and/or channel as an authenticated user starts a stand-up within said dm or channel.
- **Success End Condition:** Before the stand-up time there should be notification sent in different time intervals.
- **Failure End Condition:** user is not a part of the channel and/or dm where the stand-up occurs.

- **Primary Actor:** An authenticated user of a stream who is also an authentic user of a dm and channel.
- **Trigger:** A person enters the stand up command for a number of seconds, enters when the stand-up will occur (i.e., date and time) and at what time intervals notification will appear as well as the message for said notification. then someone participating within the channel or dm will receive notifications until the stand-up starts at a set date and time.

MAIN SUCCESS SCENARIO

Step 1. Registered user registers/logs in
Step 2. Registered user joins channel/DM
Step 3. Registered user clicks on the schedule time button.
Step 4. Registered user inserts the time for the message to be sent, the time interval in between each recurring message and the time for the recurring messages to stop.
Step 5. Registered user confirms this selection and types their message.
Step 6. Registered user clicks the send button
Step 7. Server receives the command to recurrently send messages, starting at the time for the message to be sent and sending it within every time interval.
Step 8. Once the time reaches the time specified for recurring messages to stop, the server will stop sending recurring messages to that channel/DM.

Use Case 4

- **Use case:** I'm a user who wants to schedule a stand up for the future.
- **Goal in Context:** including a feature that allows members of a group or channel to preset times for automatic activation of stand ups. Autonomously signalling the beginning of meetings.
- **Scope:** UNSW Streams.
- **Level:** primary task.
- **Preconditions:** Connected successfully in a dm and/or channel as an authenticated user starts a stand-up within said dm or channel.
- **Success End Condition: a successful automatic start up is started at the correct given time.**
- **Failure End Condition:** user is not a part of the channel and/or dm where the stand-up occurs, start up does not start at the given time due too incorrect inputs.
- **Primary Actor:** An authenticated user of a stream who is also an authentic user of a dm and channel.
- **Trigger:** A person enters the stand up command for a number of seconds, enters when the stand-up will occur (i.e., date and time) and inputs the duration of said stand up which then will automatically activate for all those in the channel or dm at said specified time.

MAIN SUCCESS SCENARIO

Step 1. Registered user logs in.
Step 2. Registered user joins channel/DM.
Step 3. Registered user enters time (year,month,day,hour,minute) for which the stand up is scheduled
Step 4. Streams then automatically start this stand up at the given time.

Validation

With your completed use case work, reach out to the 2-3 people you interviewed originally and inquire as to the extent to which these use cases would adequately describe the problem they're trying to solve. Ask them for a comment on this, and record their comments in the PDF

Joshua Chew

That's very handy. That will allow for more effective communication allowing the sharing of images and screenshots. I can share images to friends and workers. I really liked the two options where I can send from my computer or send a link and it gets converted to an image. So I think it solves that particular problem.

Jessica Tran

I think it is well thought out and is a much needed upgrade/feature to add especially to a messaging app. This is because channel invite allows users to invite anyone within Streams. In doing so, it safeguards users from unwanted harassment. Therefore, I think it solves the problem in an easy and effective manner.

Max Cochrane

Being able to schedule standups allows you to plan in advance and makes it much easier to communicate as well as organize our stand ups as a whole and not only that it makes use of implementing a feature which makes it more succinct. This upgraded feature makes the platform more usable in a business inclined environment as well as on a personal note allowing scheduled stand ups leads to the possibility of running other recurring events olike watch parties and streams.

Design

Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
block/user/v1 Given a user's **handle_str**, block them from the authorised user. A blocked user will no longer be able to invite the user into channels or DMs. Any messages sent from the blocked user will be replaced with "Blocked Message".	PUT	**Parameters:** { token, handle_str } **Return Type:** { }	**InputError when:** ● handle_str refers to a user who is already blocked. **AccessError when:** ● handle_str does not refer to a valid handle_str ● token passed in is invalid
message/recurring/v1 **message** is the message that is being sent recurrently. **channel_id** is the channel that the message is being sent to recurrently, and is -1 if it is being sent to a DM. **dm_id** is the DM that the message is being sent to recurrently, and is -1 if it is being sent to a channel. **time_sent** is the specified time to send the initial message. If no **time_sent** is provided, **time_sent** is automatically set to immediately. **time_interval** is the time between each recurring message. **time_stop** is the time when recurring messages will stop being sent.	POST	**Parameters:** { token, message, channel_id, dm_id, time_sent, time_interval, time_stop } **Return Type:** { message_id }	**InputError when:** ● both channel_id and dm_id are invalid ● both channel_id and dm_id are -1 ● neither channel_id or dm_id are -1 ● length of the message is over 1000 characters ● time_sent is a time in the past ● time_interval is less than 0 ● time_stop is a time in the past ● time_stop is a time before time_sent **AccessError when:** ● token passed in is invalid

			● the pair of **channel_id** and **dm_id** are valid (i.e. one is -1, the other is valid) and the authorised user has not joined the channel or DM they are trying to send a recurring message to.
message/sendimage/local/v1 Given a local image saved in the computer, it sends a message with said image into channel or DM. **img** is the image that is locally saved in the machine **channel_id** is the channel in which the message is being sent to, and is -1 if it is being sent to a DM. **dm_id** is the DM in which the message is being sent to, and is -1 if it is being sent to a channel.	POST	**Parameters:** { token, img, channel_id, dm_id} **Return Type:** { }	**InputError when:** ● **img** is not a valid image (.jpeg/.jpg) ● both **channel_id** and **dm_id** are invalid ● both **channel_id** and **dm_id** are -1 ● neither **channel_id** or **dm_id** are -1 ● **img** is above 100MB in storage size **AccessError when:** ● **token** passed in is invalid ● the pair of **channel_id** and **dm_id** are valid (i.e. one is -1, the other is valid) and the authorised user has not joined the channel or DM they are trying to send a message to.
message/sendimage/external/v1 Given a URL of an image on the internet, sends a message with said image into the channel or DM. Please note: the URL needs to be a non-https URL (it should just have "http://" in the URL. **imgurl** is the URL of the image **channel_id** is the channel in which the message is being sent to **dm_id** is the DM in which the message is being sent to	POST	**Parameters:** { token, imgurl, channel_id, dm_id} **Return Type:** { }	**InputError when:** ● **imgurl** is not a valid image (.jpeg/.jpg) ● both **channel_id** and **dm_id** are invalid ● both **channel_id** and **dm_id** are -1 ● neither **channel_id** or **dm_id** are -1 ● **imgurl** is above 100MB in storage size ● **imgurl** does not start with 'http://' **AccessError when:**

			● **token** passed in is invalid ● the pair of **channel_id** and **dm_id** are valid (i.e. one is -1, the other is valid) and the authorised user has not joined the channel or DM they are trying to send a message to.
standup/schedule/v1 Given a date and time, a start up will automatically start and last for a duration specified in input, in a channel or dm. **Id** is the id of the channel/ dm **Id_type** is the id type i.e. Dm = 2 and Channel = 1 **Length** is the duration of the stand-up. **Month_day** is the day and month at which it's supposed to start given in a form of a tuple where the first input is a number from 1-12 and day it can range from 1-30 or 1-31 depending on the month. **Hour_min** is a tuple that includes what hour in the day in 24 hour time and the minute ranging from 1-60.	POST	**Parameters:** { token, id,length,id_type ,month_day,hour_min} **Return Type:** { time_finish }	AccessError when: ● Token is passed is irrelevant InputError when: ● Id type is incalid (not 1 or 2) ● Id is a Invalid dm_id or channel_id is ● Month in month_day is invalid (i.e. more than 12 less than 20) ● Day in month_day is invalid (i.e. the day is does not correspond correctly to the month, (1,30)) ● Hour in day_hour is invalid i.e. less than 1 or more than 60

Conceptual Modelling (State)

Diagram 1

- **Waiting for user to login or register**
  - User registers → **Waiting for user to initiate an action**
  - User logs in → **Waiting for user to initiate an action**

- **Waiting for user to initiate an action**
  - User clicks on already joined channel → **Displays messages, members list and waits for the user's action**
  - Creates a channel → **Displays messages, members list and waits for the user's action**
  - Creates a DM → **Displays messages, members list and waits for the user's action**
  - Joins a channel → **Displays messages, members list and waits for the user's action**
  - User clicks on already joined DM → **Displays messages, members list and waits for the user's action**

- **Displays messages, members list and waits for the user's action**
  - User clicks on the schedule time button → **Prompts the user to enter a date and time for the recurring message to start**

- **Prompts the user to enter a date and time for the recurring message to start**
  - (Invalid) → **INPUTERROR: Invalid time provided**
  - User enters a correct date and time → **Prompts the user to enter a time interval in between each recurring message**

- **Prompts the user to enter a time interval in between each recurring message**
  - (Invalid) → **INPUTERROR: Invalid time provided**
  - User enters a correct date and time → **Prompts the user to enter a date and time for the recurring message to stop**

- **Prompts the user to enter a date and time for the recurring message to stop**
  - (Invalid) → **INPUTERROR: Invalid time provided**
  - User confirms the selection → **Waits for the user to enter a message**

- **Waits for the user to enter a message**
  - User enters their message and sends it → **Displays messages, members list and waits for the user's action**
  - User enters a message longer than 1000 characters → **INPUTERROR: Message too long**

Diagram 2

- **Wait for user to login or register**
  - user registers → **Waiting for user to initiate action**
  - user logs in → **Waiting for user to initiate action**

- **Waiting for user to initiate action**
  - creates a channel → **Displays messages members list and waits for user action**
  - creates a DM → **Displays messages members list and waits for user action**
  - joins a channell → **Displays messages members list and waits for user action**

- **Displays messages members list and waits for user action**
  - Blocked member → **new state here messages from blocked user in common channels re hidden**

- **new state here messages from blocked user in common channels re hidden** ↔ **INPUTERROR: Already a blocked member**