F13B DINGO - Aisha, Aleesha, Jenson, Van

[Requirements] Elicitation - Aisha + Aleesha

Target scope: university students

For our elicitation, we created a Google Form with a combination of questions that elicit both inconveniences or issues with online communication and possible solutions from the target user, and opinions on features that we think could improve these problems.

Google form:
https://docs.google.com/forms/d/e/1FAIpQLSdzEHkvF2z5_BVz6eLS5B5u3VVQTZ9RvWSjQ0GCupI2hfHZFA/viewform

Responses:

https://docs.google.com/forms/d/1pvtCb3IxQK_JrakKQdotPwO292T5rcDyqe4_0miMFoY/edit#responses

From the responses we received, university students seem to find organising group meetings and the deletion of message history to be their biggest problems with using online communication tools.

Proposed solution:

For organising group meetings, we can implement a feature that's similar to our current standup feature where each member of a channel or dm can provide and adjust their availability over a given timeframe. This can be done by any user regardless of owner status and the availability of specific users can be requested. Another feature that would be beneficial for this would be to implement statuses for each user to know who is currently available or not.

To tackle message history being deleted accidentally or on purpose to clear space, we can implement a function where the users can save all/select messages to a page on the server or to an external data storage location such as Google Drive.

[Requirements] Analysis & Specification - Use Cases - Aleesha + Aisha

User Stories

Since all users we interviewed were university students that use, or plan to use, an online communication tool like Seams, the user stories will be written using the name of each student. Any shared suggestions will be written in a generic university student user story.

As Kevin, I want to be able to save chat messages and email them to all members that have joined the channel or dm so that everyone has a copy of group discussions.

Scenario: The user wants to save all/select messages

Given: The user knows the start and end indexes of the range of messages they want to save

When: A save command is run in the chat or a save button is clicked

Then: The messages will be saved to a page that can be viewed on Seams

Given: The user is still in the channel/dm

When: The user opens/navigates to the specific messages page

Then: The user will be able to view the contents of each message, the user who sent the message, and when the message was sent

Scenario: The user wants to email all/select messages to each member directly from Seams

Given: The user has saved the messages (above scenario)

When: The user views the saved messages on Seams

And: A send email button is clicked

Then: The system sends the contents of the messages page in an email to each member

Given: A member received the email

When: The email is opened

Then: The member will be able to view the contents of each message, the user who sent the message, and when the message was sent

As a university student, I want to be able to share my availability with my group members so we can organise team activities/meetings more easily

Scenario: The user wants to upload a screenshot of their timetable

Given: The user has a screenshot of their timetable

When: An image attachment button in the message input box is clicked

Then: The user can upload the screenshot with an optional message

Given: There are other users in the channel or dm and/or the user who uploaded the screenshot is still in the channel or dm

When: The other users or the user who uploaded the photo opens the channel or dm page

Then: They will see the latest message that contains an image

Scenario: The user wants to gather the availability of all/some members over a certain timeframe

Given: The user is a member of the channel or dm

When: The user clicks the ‘When to Meet’ button

Then: The user will be prompted to input a time limit [xx hrs, xx mins, xx sec], and a link for a shared calendar will be sent in the channel or dm

Given: A user wants to add their availabilities

When: The user opens the calendar link

Then: They can highlight the times when they are available to meet over the next 7 days

Given: A user needs to adjust the availabilities they have already set

When: The user opens the calendar link

Then: They can edit the times when they are available to meet over the given time limit

Given: The availability timer has ended and/or all members have added their availabilities

When: A generate button is clicked on the availability common window

Then: A timetable highlighting availabilities using shades of colour is generated showing availabilities of all users who filled in the calendar(from light to dark of the same colour illustrates not available to most available)

As Felix, I want to be able to draw diagrams with digital ink with drawing features in channels and dm so I could communicate complex tasks using visual illustration to students.

Scenario: The user wants to create an illustration

Given: The user has ideas about diagrams they wish to draw

When: An drawing feature button in the message input box is clicked

Then: The user is directed to empty canvas page

Given: The user is in empty canvas page

When: The user wishes to select drawing features such as pen type or colour

And: Feature buttons representing colour or thickness is clicked

Then: The user could select desired features and create drawings on the canvas

Scenario: The user wants to save or send the created drawing to a channel/dm

Given: The user has finished and wishes to send or save the drawing

When: The user has finished drawing illustration and get prompted with button containing two selections[save image, save and send image] or cancel

Then: image is saved and sent, otherwise all is aborted

Given: The user has pressed save and send image

When: The user selects from all list of channel or dm he is in and click specific one

Then: Drawn illustration will be sent to selected channel or dm

Given: Other users in the channel/dm received the sent illustration

When: They long pressed the image

Then: They could select if they wish to save or illustrate on the image and proceed

As May, I want to be able to upload chat history to a drive or cloud storage with the click of a button so we don't lose chat history that is accidentally deleted or is cleared for space.

Scenario: The user wants to upload chat history to an external storage service

Given: The user has a Google account associated with the email they connected to Seams

When: An upload command is run in the chat or an upload button is clicked

Then: The messages will be saved to a file

And: The file will be uploaded in the root of the user's Google Drive storage (if this tutorial is followed: https://www.geeksforgeeks.org/uploading-files-on-google-drive-using-python/)

Given: The user is still in the channel or dm

When: The user opens their Google Drive storage

Then: The user will be able to view the file which contains the contents of each message, the user who sent the message, and when the message was sent

Use Cases

For message storage, file can be saved, then sent to members via email and/or be uploaded to Google Drive

- The user long press any message in channel/dm to bring up the 'select'
- The user gets prompted with a page with selection tabs on all messages inside the channel/dm
- User selects any messages or files inside channel/dm by clicking on the message, message will be highlighted if selected
- Selection will be withdrawn if click again, or save all messages using 'all' button
- User clicks 'finish' button and get prompted with notification tabs with [save, send, cancel]
- User could click save to upload selected messages to google drive account(specified in profile info)
- Click 'send' then choose 'email' or 'chat'
- Clicking 'chat' goes to a page with a list of channels or dms, select any one with a prompted message saving "send [chat history] to [channel/dm]?", click send to send, cancel to abort command and return to the message selection page
- Clicking 'email' goes to the email entering box
- The user enters any additional email in the inbox then clicks confirm
- The page will bring up error message if email is not input correctly, otherwise the selected message will be sent to the input email address and all members in the channel/dm of the execution

For generation of availability timetable, users could start whentomeet command for a certain duration of time

- The user clicks ‘when to meet’ button in the message tab
- The user is prompted to enter the duration of time in hours minutes and second for where to meet this command [xx hrs, xx mins, xx sec] and X number of days to include on the calendar
- The user clicks ‘start’ after entering the time then a link is sent to the specified channel or dm
- The user clicks the link and is brought to a page with an empty timetable
- The user is able to click any blank grid space in the timetable to specify personal availability
- The selected space will be highlighted green to represent free time to meet, clicking again will abort the previous command
- The user clicks ‘send’ button after finishing filling the timetable, then his/her availability is being saved to this Whentomeet standup
- After the set time duration, the link will stop accepting more responses and auto generates a timetable with collaborative availability for all uses that have submitted a response, the extent of availability will be shown using different shades of the same colour from light to dark representing ‘not available ’ to ‘mostly available’

[Requirements] Validation - Aisha

We have created a second Google Form that we sent out to the three students we interviewed previously asking for their feedback on our two proposed use cases. For each use case, we asked the students to rate how well the use case solved the problems that we elicited on a scale from 1 to 5. We also prompted the students to provide any extra comments about our proposed solution.

Google form:

https://docs.google.com/forms/d/e/1FAIpQLSdWyGi6jaJUgyYVOtFxK_Hj-_MBz7UaCBK3ONHfwQXsXwauow/viewform

Responses:

https://docs.google.com/forms/d/10iqiNplII_KtHe13Evaw3_UMKrXr3g75seVkYebgnBc/edit?usp=sharing

From the given responses, it could be concluded that the use cases have described and solved the problems well. However, valuable advice is given to further functionality of our interface.

For use case 1 that describes the storage of messages to google forms or send to members by email, it is suggested that the process could be automated every single time that a message is sent, new messages could be synced with google drive message storage. We could set the refresh time to be one minute and that saving feature is executed every minute to make sure that all messages are backed up. This may cause frustration if sent by email though, so manual optional for email could still be kept. What is more, it is suggested that not only email could be used, more popular chatting apps could be connected to Seams in the future, that messages in Seams could be sent across applications to make sharing more convenient for various platforms. Overall, 100% of responses chose 5 on a scale of 1 to 5 saying that this use case successfully describes and solves their problem very well.

For use case 2 that describes the interaction between users with the whentomeet form, most participants have left very positive comments saying that the use case perfectly describes and solves the problem. There was an additional comment advising that the feature could be combined with the notification feature in Seams, that certain user could be tagged to be reminded, and the notice will not subside until chosen users have filled in the response. This will further the functionality of this feature to avoid the case that timetable is faulted since users have forgotten to fill in the form.

Overall, our written use cases have described and solved the brought up problems well, indicating that the team is on a good track for tackling current problems and coming up with creative solutions.

[Design] Interface Design - Jenson

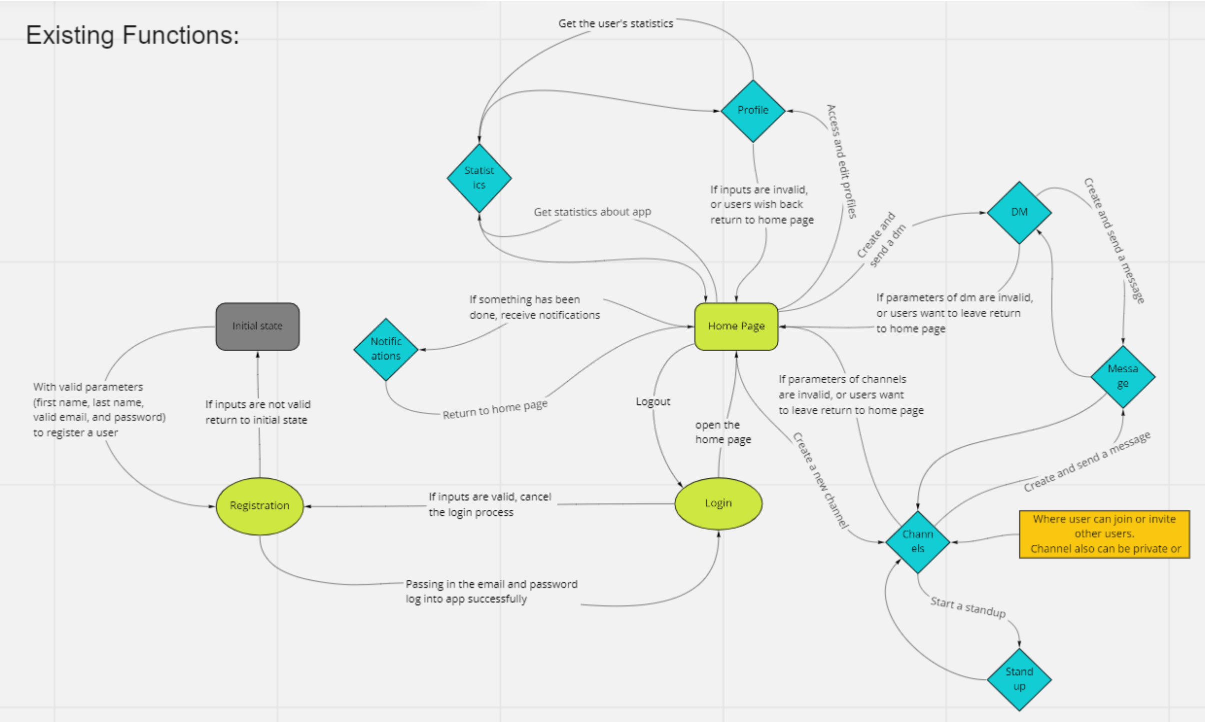
We have generated the following HTTP endpoints that would be suitable for our solutions summarised in our case studies.

Route	Description	Method	Input Parameters	Return object	Exceptions
whentomeet/start	The user selects an option of how many days to include on the calendar, and hours to run the timer for, and then creates a timetable to allow other members to select their availability.	POST	Parameters: {token, channel_dm_id, calendar_days, hours}	Returns a URL to the page generated by the function, which contains the calendar_days that can be filled in by all members.	AccessError: - Invalid token InputError: - Invalid format for parameters - Invalid date or time
whentomeet/add	The users invited to the meeting can select their available days/times, which then updates the meeting so all invited users can see their choices. A 30-min block will be filled in from the given time.	POST	Parameters: {token, channel_dm_id, day, time}	{}	AccessError: - Invalid token InputError: - Invalid day/time selected
whentomeet/send	When timer has ended or all users have input their availabilities, update the calendar	POST	Parameters: {token, channel_dm_id}	{}	AccessError: - Invalid token
messages/save	A user can select a range of messages, and save them. The messages will be stored on a separate page, and can be accessed by the user only. The save will contain the content of the message, the sender/receiver, and a timestamp. The content will be stored as a txt file.	POST	Parameters: {token, channel_dm_id, start_of_index, end_of_index}	Returns a URL to a page containing a txt file containing the information.	AccessError: - Invalid token
messages/sendlog	The user can forward the txt file from messages/save to all members in the channel using their account emails or upload the file	POST	Parameters: {token, channel_dm_id, URL}	{}	AccessError: - Invalid token - Invalid channel_dm_id InputError:

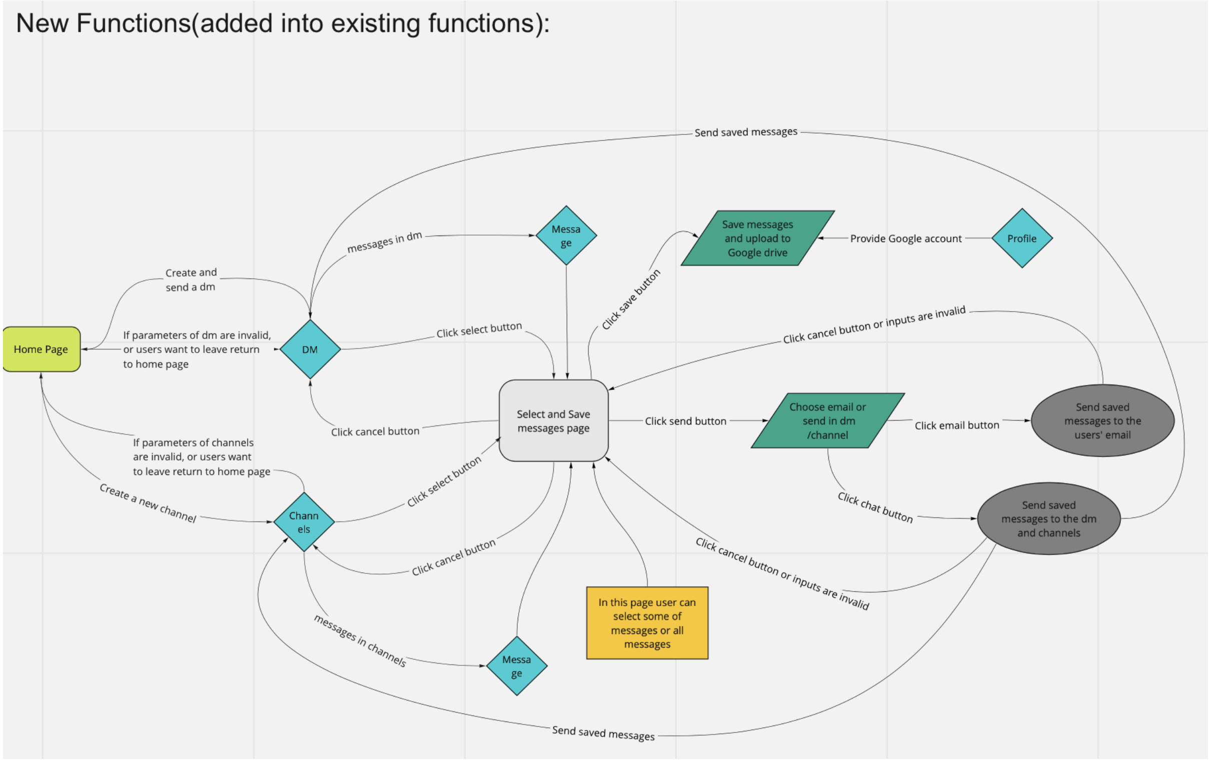
	to a google drive (assuming their email is connected to a google drive). If email is not connected to a Google Drive account, no errors are raised and the file is not uploaded.				- Invalid URL
message/sendimage	The user can send images in the message section. Given the URL is valid, the image is downloaded and sent.	POST	Parameters: {token, channel_dm_id, URL}	Returns the message_id	AccessError: - Invalid token - Invalid channel_dm_id InputError: - Invalid URL

[Design] Conceptual Modelling (State) - Van

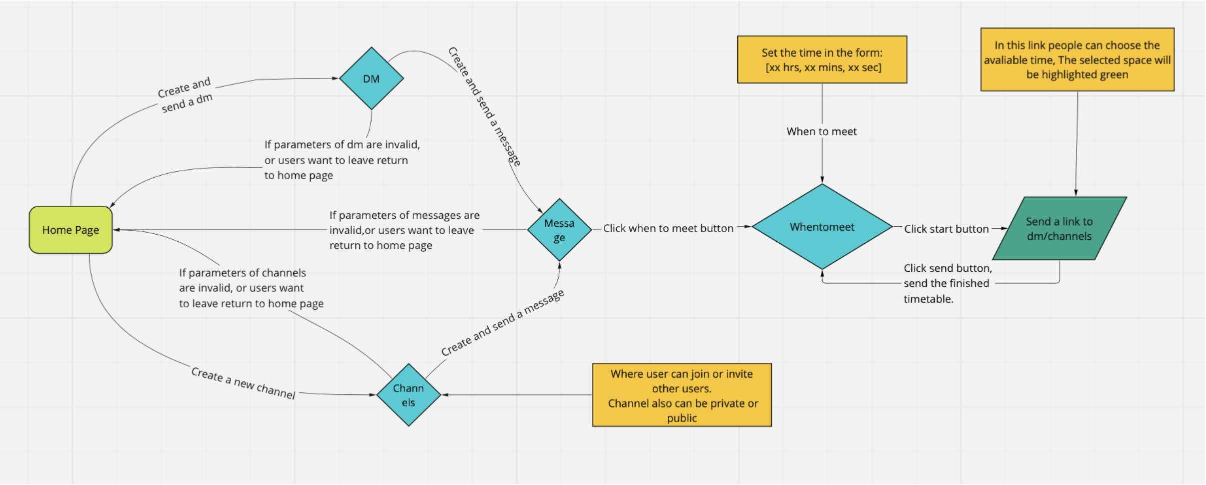
State diagram for our existing functions:


Existing Functions:
Get the user's statistics
Profile
Access and edit profiles
Statistics
Get statistics about app
If inputs are invalid, or users wish back return to home page
Create and send a dm
DM
Create and send a message
If something has been done, receive notifications
Initial state
Notifications
Home Page
If parameters of dm are invalid, or users want to leave return to home page
With valid parameters (first name, last name, valid email, and password) to register a user
If inputs are not valid return to initial state
Return to home page
Logout
open the home page
If parameters of channels are invalid, or users want to leave return to home page
Message
Create a new channel
Create and send a message
Registration
If inputs are valid, cancel the login process
Login
Channels
Where user can join or invite other users.
Channel also can be private or
Passing in the email and password log into app successfully
Start a standup
Standup

State diagram of our new functions for saving and sending message history:


New Functions(added into existing functions):
Send saved messages
messages in dm
Messa ge
Save messages and upload to Google drive
Provide Google account
Profile
Create and send a dm
Click save button
Home Page
If parameters of dm are invalid, or users want to leave return to home page
DM
Click select button
Click cancel button or inputs are invalid
Select and Save messages page
Click send button
Choose email or send in dm /channel
Click email button
Send saved messages to the users' email
Click cancel button
If parameters of channels are invalid, or users want to leave return to home page
Click select button
Create a new channel
Chann els
Click chat button
Send saved messages to the dm and channels
Click cancel button
Click cancel button or inputs are invalid
In this page user can select some of messages or all messages
messages in channels
Messa ge
Send saved messages

State diagram for our Whentomeet function:


DM
Create and send a dm
Create and send a message
Set the time in the form: [xx hrs, xx mins, xx sec]
In this link people can choose the avaliable time, The selected space will be highlighted green
If parameters of dm are invalid, or users want to leave return to home page
When to meet
Home Page
If parameters of messages are invalid,or users want to leave return to home page
Messa ge
Click when to meet button
Whentomeet
Click start button
Send a link to dm/channels
If parameters of channels are invalid, or users want to leave return to home page
Click send button, send the finished timetable.
Create and send a message
Create a new channel
Chann els
Where user can join or invite other users. Channel also can be private or public