## Team Name: F15A Crunchie

## Elicitation

### Questionnaire

1. What teamwork-driven communication tools do you use?
2. How often do you use these particular tools?
3. What do you usually use teamwork-driven communication tools for?
4. Do you face any problems when using teamwork-driven communication tools? If so, what problems do you usually face?
5. What are your suggestions on solving these problems?
6. Are there any features that you would like to see in a teamwork-driven communication tool?

### Responses

- Name: Hockeen Liaw

  Email: hockeen.liaw@gmail.com

  1. MS Teams, Messenger
  2. Everyday
  3. University
  4. Inability in refreshing messages quick, audio and video problems in meetings
  5. Improve on message refreshing, improve audio and video quality
  6. Since Teams do not have a meeting feature, it could be implemented

- Name: Charles Hang

  Email: darkbox118@gmail.com

  1. MS Teams,
  2. Every week
  3. Group assignments at university
  4. Poor performance, extremely slow to find files and navigate between channels and folders on Teams. better connection
  5. Create a simpler interface that is friendlier to message and respond to messages. Currently, the new conversation and reply feels like writing an email, it should feel more friendlier and carefree to message. Basically be more like Discord/Slack.
  6. As a programmer, I also like an easy way to write code snippets. Discord implements the code snippet UI well with the triple ```, I'd like to see something similar that is easy to access as well.

- Name: Angelica Hii

  Email: hiiangelica@hotmail.com

  1. MS Teams, Zoom
  2. Every week
  3. Group meetings, lectures
  4. Bad connection but usually that's on my end, video / sound can get quite choppy
  5. better connection
  6. Zoom's 40 min limit should be extended (unlimited timing) for all

**Proposed solution**

Two of our respondents mentioned about meetings. Therefore, since UNSW Treats does not have a meeting feature, we could add a meeting feature.

## Analysis & Specification - Use Cases

**User Stories**

1. As a student, I want to be able to start a meeting on a channel so that my teammates can join the meeting to communicate.

User acceptance criteria:

- The user of a channel can start a meeting in the channel
- Given that the user have started a meeting, the user join the meeting as part of creation and a timer is started to record meeting duration

2. As a student, I want to be able to join a meeting so that I can have discussions with my teammates.

User acceptance criteria

- The user of a channel can join any meeting which is created in the channel
- Given that the user have joined the meeting, the user is able to communicate with all members in the meeting

3. As a office worker, I want to be able to schedule a meeting so that my colleagues and I can be aware of the meeting happening.

User acceptance criteria:

- The user of a channel can schedule a meeting in the channel
- The scheduler of the meeting can add details for the meeting, namely meeting name, meeting time, and meeting members
- Given that a meeting is scheduled, invited members are notified
- Given that a meeting is scheduled, members of the channel can view the details of the meeting
- Given that a meeting is scheduled, a reminder will be sent to all meeting members 1 hour before the meeting

4. As a student, I want to be able to create a calendar so that I can keep track of all my meetings

User acceptance criteria:

- The user can create a calendar which includes all upcoming meetings
- Given that the calendar is created, the user is able to view all upcoming meetings

**<u>Use case</u>**

Use case: Starting a meeting

- Goal in context: A user of a channel can start a meeting
- Scope: UNSW Treats
- Level: Primary Task
- Preconditions: The user has registered an account and is logged in,  the user should also be a member of the channel
- Success End Condition: The user is in the meeting and can send a messages in the meeting
- Failed End Condition: The user is not in the meeting and cannot send messages in the meeting
- Primary Actor: User
- Trigger: User selects option to start a meeting

Success Scenario

1. User selects option to start a meeting in a channel
2. UNSW Treats lists user as a member of the meeting
3. UNSW Treats starts timer to record duration of meeting
4. User sends a message to the meeting
5. UNSW Treats displays the sent message

Use case: Joining a meeting

- Goal in context: A user can become a member of a meeting either by their own initiative or by invitation

- Scope: UNSW Treats
- Level: Primary Task
- Preconditions: The user has registered an account and is logged in,  the user should also be a member of the channel
- Success End Condition: The user is in the meeting and can send a messages in the meeting
- Failed End Condition: The user is not in the meeting and cannot send messages in the meeting
- Primary Actor: User
- Trigger: User selects option to join a meeting, or user is invited by another user to a meeting

Success Scenario 1

1. User selects option to join a meeting in a channel
2. UNSW Treats displays all messages sent in the meeting
3. UNSW Treats lists user as a member of the meeting
4. User sends a message to the meeting
5. UNSW Treats displays the sent message

Success Scenario 2

1. User is invited by another user to join a meeting in a channel
2. UNSW Treats displays all messages sent in the meeting
3. UNSW Treats lists user as a member of the meeting
4. User sends a message to the meeting
5. UNSW Treats displays the sent message

Use Case: Scheduling a meeting

- Goal in context: A user can schedule a meeting in the channel
- Scope: UNSW Treats
- Level:
- Preconditions: The user has registered an account and is logged in, the user should also be a member of the channel
- Success End Condition: A meeting is scheduled in the channel with details visible to all members
- Failed End Condition: Meeting is not scheduled
- Primary Actor: User
- Trigger: User selects option to schedule a meeting

Success Scenario

1. User selects option to schedule a meeting
2. User enters details of the meeting
3. UNSW Treats displays details of the meeting
4. Notifications are sent to all invited members

Use Case: Creating a calendar

- Goal in context: A user can create a calendar listing all the meetings of the user
- Scope: UNSW Treats
- Level:
- Preconditions: The user has registered an account and is logged in
- Success End Condition: A calendar is created
- Failed End Condition: Calendar not created
- Primary Actor: User
- Trigger: User selects option to create calendar

Success Scenario

1. User selects option to create a calendar
2. UNSW Treats displays a calendar with details of all meetings of the user

## Validation

**Comments**

Hockeen Liaw : The proposed solution is great as meeting feature is one of the most important and most-used feature in any communication platform

Angelica Hii: Creating meeting feature is definitely a good start to solving all the other problems related to meetings.

## Interface Design

Notes: All routes that require a token should raise a 403 Error object when the token passed in is invalid.

| Name and description | HTTP Method | Data Types | Exceptions |
|---|---|---|---|
| meeting/start | POST | Parameters: ( token, channelId, meetingName )<br>Return type if no error: { meetingId } | **400 Error** when any of:<br>● channelId does not refer to valid channel<br>**403 Error** when any of: |

| | | | |
|---|---|---|---|
| | | | ● channelId is valid and the authorised user is not a member of the channel |
| meeting/join | POST | Parameters: ( token, meetingId )<br>Return type if no error: { } | **400 Error** when any of:<br>● meetingIld does not refer to a valid meeting<br>● the authorised user is already a member of the meeting<br>**403 Error** when:<br>● authorised user is not a member of the channel where the meeting is held |
| meeting/invite | POST | Parameters: ( token, meetingId, uId )<br>Return type if no error: { } | **400 Error when any of:**<br>● meetingId does not refer to a valid channel<br>● uId does not refer to a valid user<br>● uId refers to a user who is already a member of the meeting<br>**403 Error when:**<br>● authorised user is not a member of the channel where the meeting is held |
| meeting/schedule | POST | Parameters: ( token, channelId, meetingName, | **400 Error when any of:** |

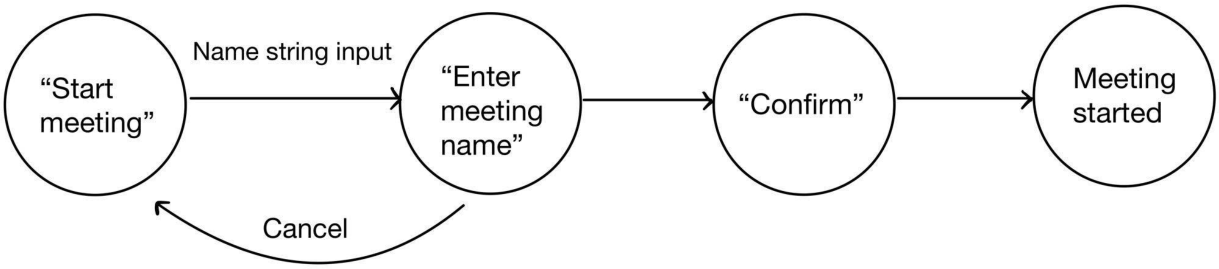
|  |  |  |  |
|---|---|---|---|
|  |  | meetingTime, uIds )<br>Return type if no error:<br>{ } | - channelId does not refer to a valid channel<br>- uId does not refer to a valid user<br><br>**403 Error when:**<br>- authorised user is not a member of the channel where the meeting is scheduled |
| calendar/create | POST | Parameters: ( token )<br>Return type if no error:<br>{ calendar } |  |

**New variables**

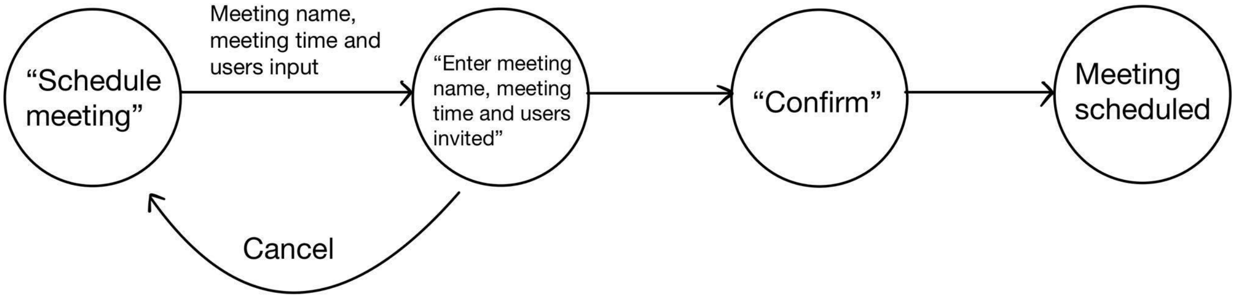
| **Variable Name** | **Input / Output** |
|---|---|
| (Outputs only) named exactly **calendar** | Array of objects, where each object contains types { meetingName, meetingTime } |

## Conceptual Modelling - State Diagram

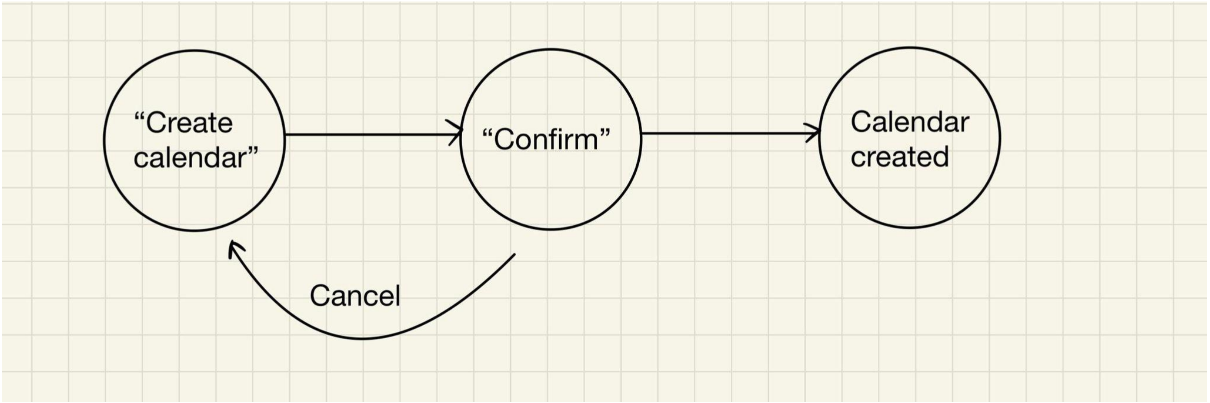
Starting a meeting



Scheduling a meeting


"Schedule meeting"
Meeting name, meeting time and users input
"Enter meeting name, meeting time and users invited"
"Confirm"
Meeting scheduled
Cancel

Creating a calendar


"Create calendar"
"Confirm"
Calendar created
Cancel