[Requirements] Elicitation

Interviewee:

- **Eric Ning - enin7193@uni.sydney.edu.au**
- **Oscar Wen - zwen0504@uni.sydney.edu.au**
- **PeangYao Wang - pengyao.wang@student.monash.edu**
- **Sen Du - sam@umei.group**

What hurdles do you experience in a team environment that you would like to see solved in UNSW streams

Eric: For a teamwork driven communication tool, it is very hard to get everyone on the same page. Ideally, I would like to have everyone be able to share their progress or what they are doing in real time.

Oscar: Collaboration is a very important aspect of working in a team environment so maybe having more features that help facilitate this would be good.

PengYao: Instantly communicating and task solving is crucial in teamworks. Quick response to events and auto setting status would be very helpful.

Sen: Separation of teamwork causes trouble, there are always complaints about having too much work or too less to do.

What can UNSW Streams do to help members collaborate more efficiently?

Eric: The importance of everyone being aware of their roles and what everyone else is doing will go a long way in getting the team to be more efficient and reduce confusion.

Oscar: Communication is essential to allowing for more effective collaboration as by having people giving their inputs in a discussion, it allows for a better understanding of what everyone wants.

PengYao: A desktop component can help. This tool or to say diagram can display what others are doing currently and auto updating what everyone has achieved .

What is a problem that you face working as a team?

Eric: It is really hard to know when everyone is available other than just seeing them type in chat and knowing what people are present at a given time is important information for all team members.

Oscar: Disagreements can occur quite regularly in a team environment and there are rarely any functions that help with these conflicts

PengYao: When communicating with group members, typing and sending messages can take a lot of time and there is no function for sending voice messages or inviting specific members into a phone call.

Summary:

Communication problems:

- Voice/video call to enable users to meet more frequently, facilitating collaboration
- Screen share to convey visual / technical information and online documents editing.
- Having a channel announcement to make so that important issues are addressed.

Collaboration problems:

- In-built issue board to delegate tasks and set deadlines for these tasks
- Upload files to make them public and allow others to give feedback or share information
- Schedule meetings by having a work calendar which tells others what time slots they are free
- Scheduled periodic reports from everyone to collecting group members' status

[Requirements] Analysis & Specification - Use Cases

1. As a student at USYD, I see a list of time slots available so that I can plan my meetings more easily.
    - Everyone has a personal calendar that all other users can view
    - When a user tries to schedule a meeting, a list of timeslots will be shown for the user to choose from.
    - If APP is unable to find any timeslots, APP will use an algorithm to find time slots where most users can attend
    - When a meeting is set, users are prompted to confirm their availability.
    - Users will receive a notification when the meeting is scheduled and a one day notification if the meeting is scheduled more than a day in advance.
    - Meeting attendance is recorded as part of user stats to incentivise attendance.

2. As a user, I want an inbuilt Issue board so that I can set and assign tasks to allow for more organised collaboration.
    - Users can reassign and remove themselves from a task
    - UNSW streams sends a notification to the user who is assigned a task
    - Can set an optional deadline: (emails get sent if not completed on the day)
    - Tasks can be categorised depending on their state (e.g completed, doing, todo) with confirmation screen
    - Tasks can be ordered by priority

3. As a user, I want to be able to make team wide announcements for important issues so that they will be addressed.
    - Every user will have associated role/s
    - Only users with global permissions will be able to tag all members
    - User can tag a role to send role-specific announcements
    - A user must be added to a role by a person with the role currently

4. As a user I would like to be able to share my screen to my team members so that my team members can see what I am doing
    - I have to be apart of a voice call to screen share

- I can screen share to my channel members
- Only those in the channel can join
- I can end the stream and still be apart of the voice call
- Can choose to share screen with sound or not
- Owners have permission to stop people from sharing screens

5. As a user I would like to be able to use voice chat with other members using streams so that we can communicate more easily
    - I can start a voice chat with any member of the channel
    - They can choose to pick up or decline the voice chat
    - Multiple members can be inside the voice chat
    - Members get notified when a voice chat is started
    - The notification for the voice chat disappears after a minute
    - If one person is in the call by themselves for too long, the voice chat ends
    - Profile picture lights up when someone speaks
    - There is a mute option as well as a screen share function
    - Anyone can leave the voice chat or join in the channel at any time

6. As a user, I want to upload files to share with my team members so that we can easily share information
    - I can upload a file as an option when sending a message
    - Anyone can download the file that I sent
    - There is a limit to the size of the file being uploaded
    - Only members of the channel can download the file

Use Case:

- **Use Case**: Schedules meeting
- **Goal in context**: Users need an automated way to schedule meeting based on everyone's availability
- **Scope**: UNSW Streams
- **Level**: Primary task
- **Preconditions**: User is a part of the channel
- **Success End Condition**: A meeting time is created and confirmed by included members via email
- **Failed End Condition**: N/A
- **Primary Actor**: User
- **Trigger**: User selects an option to create a meeting OR User is changing the

meeting time

Success Case 1.

1. Users are prompted to select a list of members required to attend the meeting.
2. App calculates available time slots based on the member’s public schedules.
3. Users are prompted to select time slots from a dropdown menu.
4. User sets a time slot for next week.
5. Meeting is posted as an announcement in the channel.
6. Included members are tagged and the app sends a notification to meeting members.
7. App asks meeting members to confirm.
8. Time of meeting is marked as BUSY for all meeting members’ schedules.
9. A day prior to the meeting, members of the meeting are tagged again with a meeting reminder.

Success Case 2

1. Users are prompted to select a list of members required to attend the meeting.
2. App calculates available time slots based on the member’s public schedules.
3. No available time slots due to timetable clashes.
4. UNSW Streams calculates the time slot with most members available.
5. Owner confirms the meeting.
6. Members selected are tagged and the app sends them a notification.
7. App asks meeting members to confirm.
8. Time of meeting is marked as BUSY for all meeting members’ schedules.
9. A day prior to the meeting, members of the meeting are tagged again with a meeting reminder.

Fail Case 1

1. Users are prompted to select a list of members required to attend the meeting.
2. App calculates available time slots based on the member’s public schedules.
3. No available time slots due to timetable clashes.
4. App calculates the time slot with most members available.
5. App is unable to determine appropriate time due to major timetable clashes.

Use Case:

- **Use Case**: Create a task on the issue board
- **Goal in context**: User needs to assign and remove tasks on Issue board to track team progress.

- **Scope**: UNSW Streams
- **Level**: Primary task
- **Preconditions**: User is a part of the channel
- **Success End Condition**: User creates a task on the issue board. User can change the state of an existing task
- **Failed End Condition**: N/A
- **Primary Actor**: User
- **Trigger**: User selects option to create a task on the issue board, OR User is changing the state of an existing task.

Success Case 1:

1. User creates a task that is to be completed
2. The task is delegated to a team member
3. UNSW streams sends a notification to that team member

Success Case 2:

1. User creates a task that is to be completed
2. The task is delegated to a team member
3. UNSW streams sends a notification to that team member
4. Deadline is set for the task to be completed by a user
5. User is not finished with the task
6. UNSW streams sends an email to the user

Success Case 3:

1. User creates a task that is to be completed
2. The task is delegated to a team member
3. UNSW streams sends a notification to that team member
4. The task is set to a lower priority
5. User removes themselves from the task
6. Confirmation screen pops up
7. Task is still on the issue board but with no one assigned

Success Case 4:

1. User checks what tasks are on the issue board
2. User changes the status of the task
3. Task status is changed

[Requirements] Validation

The use cases seem to address their problems with effective collaboration

Overall comments:

Issue Board:

Q: What issues did these use cases solve?

A: By having the issue board, there will be tasks assigned to people in the team that everyone can see, including progress, what needs to be done and who is doing what.

This helps reduce confusion in a team environment since users can easily view their own tasks

Q: Were there any issues that the use cases missed?

A: However, there should be more categories where the tasks can be placed in such as on hold or need help etc.

Also, it would be useful to have the ability to create comments on issues to highlight specific parts of the tasks which have issues.

Meetings:

Q: What issues did these use cases solve?

A: For the meetings, having everyone's schedule available and scheduling based on that helps a lot with getting smooth meetings

It's good to see that it accounts for unpredictability by allowing users to change meeting times and for the issue board allowing changing members, priorities.

Q: Were there any issues that the use cases missed?

A: I would like to have a record function especially for meetings to have a record of what happened as well as to show other people in the future.

Also, it would be helpful to have functionality to “block out” recurring periods of the week, so people are aware of your lunch breaks or days where you are busy.

[Design] Interface Design

Name & Description	HTTP Method	Data Types	Exceptions

meetings/create
 	
User schedules a meeting with members with IDs, given by the list 'u_ids'.
 	
Depending on the available time slots based on the work calendar for each user, a meeting is scheduled at a time within the specified time period.
	POST	Parameters: {token, meeting_name, time_period, u_ids}
		
		Return type: {meeting_id}
			Input Error:
			- Meeting time is a time in the past
			- Any of the user IDs entered is valid
			- No non-conflicting timeslots can be found in the specified time period
			- The start of time_period is a time in the past
			Access Error:
			- The user IDs passed in are valid and user trying to create the meeting is not a global owner

meetings/add_members
 	
Given a meeting ID, the meeting owner can add a member who is not already in the meeting, to the meeting.
	POST	Parameters: {token, meeting_id, u_id}
		Return types:
		{}
			Input Error:
			- 'meeting_id' does not exist
			- 'u_id' does not exist
			- Member being added is already a part of the meeting
			Access Error:
			- 'meeting_id' exists but user trying to add a member to the meeting is not the meeting owner

meetings/details Given the meeting id, a user can see the details of the	GET	Parameters: {token, meeting_id} Return:	Input Error: - 'meeting_id' does not exist Access Error:

meeting		{meeting}	- ‘meeting_id’ exists but user trying to add a member to the meeting is not the meeting owner

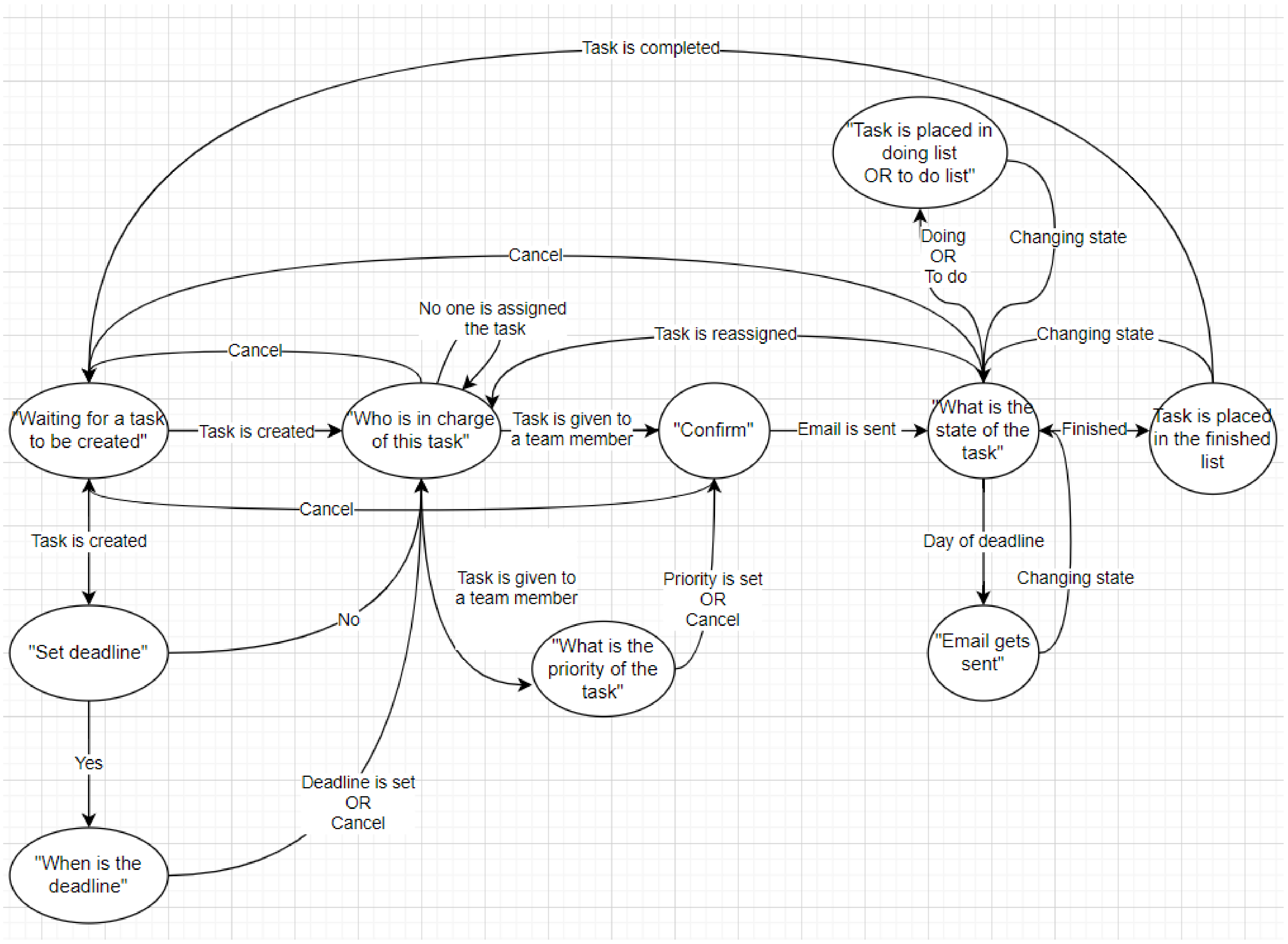
meetings/remove
 	
Given the meeting id, a user can remove the meeting
	GET	Parameters:
		{token, meeting_id}
		
		Return:
		{}
			Input Error:
			- ‘meeting_id’ does not exist
			Access Error:
			- ‘meeting_id’ exists but user trying to add a member to the meeting is not the meeting owner

channel/issueboard/create
 	
Given the token of a given member, they can create a task on the issueboard
	POST	Parameters:
		{token, channel_id}
		Return:
		{task_id}
			Input Error:
			- ‘Channel_id’ does not exist
			Access Error:
			- ‘Channel_id’exists but user trying to create the issueboard is not a member of the channel

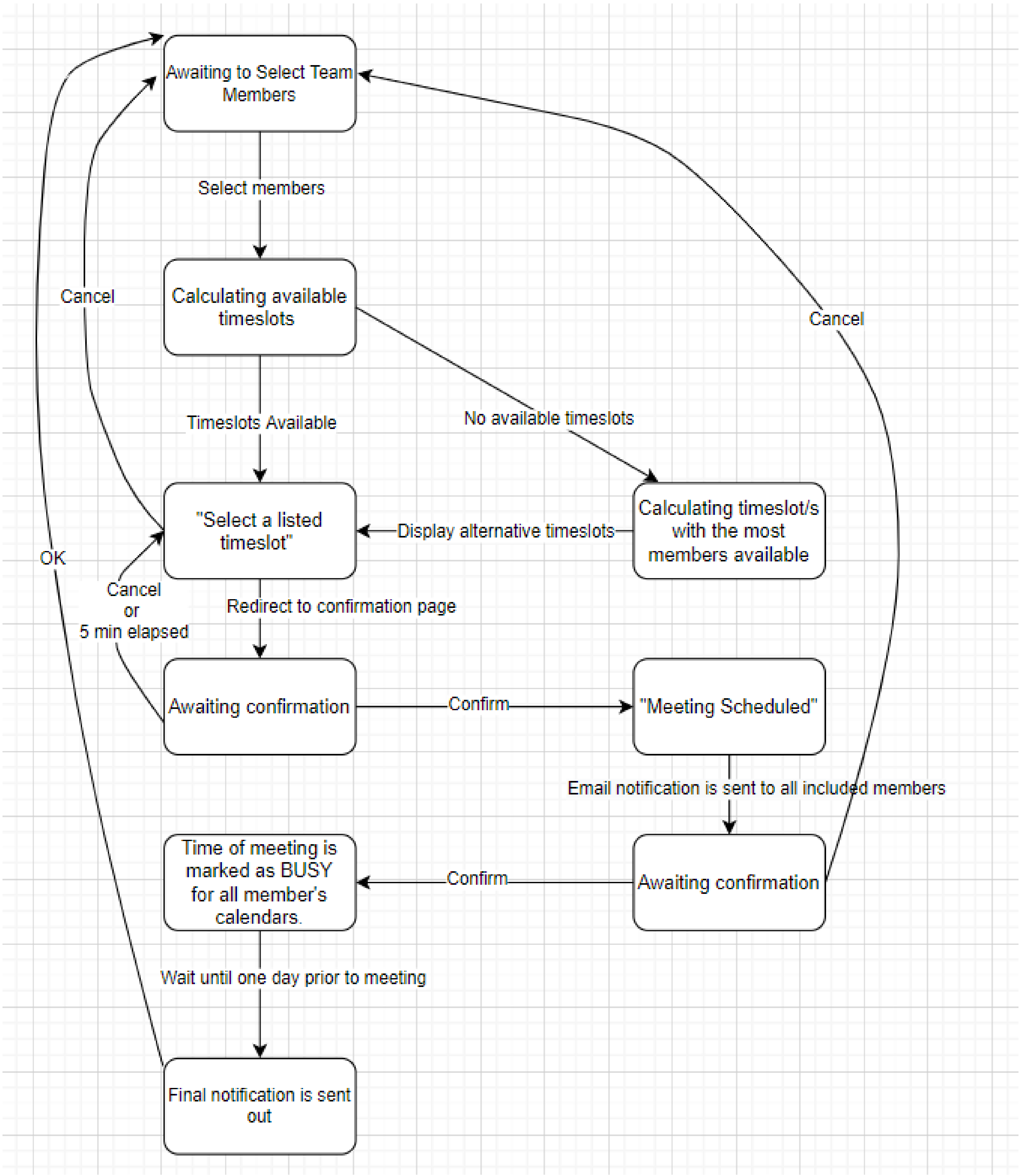
channel/issueboard/state
 	
Given the task id, they can change the state of a task depending on its state
	POST	Parameters:
		{token, task_id, channel_id}
		Return:
		{}
			Input Error:
			- ‘Channel_id’ does not exist
			- The ‘task_id’ does not exist
			- ‘Task_id’ does not refer to valid task within a channel that a user has joined
			Access Error:
			- ‘Channel_id’exists but user trying to change the task is not a member of the channel

[Design] Conceptual Modelling (State)

Issue board state diagram:


Task is completed
"Task is placed in doing list OR to do list"
Doing OR To do
Changing state
Cancel
No one is assigned the task
Cancel
Task is reassigned
Changing state
"Waiting for a task to be created"
Task is created
"Who is in charge of this task"
Task is given to a team member
"Confirm"
Email is sent
"What is the state of the task"
Finished
Task is placed in the finished list
Cancel
Task is created
Day of deadline
Task is given to a team member
Priority is set OR Cancel
Changing state
"Set deadline"
No
"What is the priority of the task"
"Email gets sent"
Yes
Deadline is set OR Cancel
"When is the deadline"

Meetings scheduler state diagram:


Awaiting to Select Team Members
Select members
Calculating available timeslots
Timeslots Available
No available timeslots
Cancel
Cancel
"Select a listed timeslot"
Display alternative timeslots
Calculating timeslot/s with the most members available
OK
Cancel or 5 min elapsed
Redirect to confirmation page
Awaiting confirmation
Confirm
"Meeting Scheduled"
Email notification is sent to all included members
Time of meeting is marked as BUSY for all member's calendars.
Confirm
Awaiting confirmation
Wait until one day prior to meeting
Final notification is sent out