Planning The Next Problem to Solve

[Planning – Jayden Matthews]

[Requirements] Elicitation

Wilson Leung – wilsonleung008@gmail.com

- **In what environment do you use a team-based communication platform, like Streams?**

Wilson: I use team-based communication platform Microsoft Teams to communicate between team members in university and course staff to discuss homework.

- **What capabilities do other platforms have that Streams does not provide that would benefit your use of the system?**

Wilson: I would like to see user statuses, such as DnD (Do not Disturb), Idle, In-Standup, Online and Offline. This includes current applications that the user is using, like Discord Enrichment.

- **How would this [capability] improve the effectiveness/efficiency of the usage of the overall System?**

Wilson: This would allow me to estimate when a reply can be received. I would also like to see who is active in channels which I have joined. It would enable me to see team-members activity and participation.

- **For the [capability], how would you expect it to be used within the system?**

Wilson: User profile pictures can have a little icon, which is colored to indicate their current status. Alternatively, you could implement a hover-over feature, which expands out a pop-up which shows extra information, such as their status, current application they are using and whether they are currently in a standup.

- **What scenarios have you experienced that could-be/were solved by [capability]?**

Wilson: I have messaged multiple of my team members, using the fact that their status was Online, and I chose not to message them whilst their status was away/DnD, to ensure they didn't miss any information.

- **What data must be utilized as input for this [capability]**

Wilson: We will need to track the active/live data from user actions, in order to determine their status.

- **What data must be stored for this [capability]? (Persistence)**

Wilson: No data must be stored.

- **How immune would this [capability] be from data loss and/or total system failure? (Maintainability)**

Wilson: As there is no data that must be stored, system failure would not be an issue, for example, if the server dies, then once it is back up all users will be set to Idle until a user makes an action/input.

- **What are the outputs that you would expect from this [capability]?**

Wilson: What applications users are using currently. And the current status of the user, as previously mentioned DnD, Idle, In-Standup, and Online.

- **How easy would it be to expand or upgrade the system's [capability]? (Scalability)**

Wilson: There would be no problem scaling the capability, as it is applied server-wide to all users.

- **What are some cases where the [capabilities] output would change?**

Wilson: Opening an active application, clicking a different page/link (Online), No inputs being received for the past 5 minutes (Idle), User changes their status to DnD manually. If a stand-up is in progress, and the user in the channel is Online (In-Standup)

- **Does this [capability] pose an unauthorized access threat?**

Wilson: It does not expose vital information but there might be a slight breach of privacy if hackers can see what application users are on.

- **Will this capability involve static or live data capture?**

Wilson: This will use live-capture data.

- **Are there any other notes that should be considered?**

Wilson: No.

[Requirements] Analysis & Specification – User Stories, Acceptance Criteria and Use Case

User Stories and Relevant Acceptance Criteria

As a student, I want to be able to see the statuses of my fellow team members, so our communication is more efficient and consistent.

- Given I am in a channel, when I click on the channel's name in the left sidebar, then I should be taken to a channel details page where the status of channel-members can be seen.
- Given I am looking in the channel details, when I scroll through the members list, then I should see a small colored icon on each of their profile pictures, which is either green for Online, red for Do Not Disturb, Yellow for Idle, Blue for In-Standup and Grey for Offline.
- Given I have logged out of all sessions, when other channel members check channel details, they should see a grey icon on my profile picture.

As a student, I want to be able to be able to manually set my status to Do Not Disturb, when I am busy or unable to chat with the team, so that I don't get bombarded with notifications/messages.

- Given I am on my profile details page, when I click the DnD toggle switch, my status is now manually overridden to a red icon as viewed from all other users.

As a member in university projects, I want to be able to see what applications users are currently running, so that I can track whether my team are doing work or taking a break, as to not bother them.

- Given I am on the channel details page, When I hover over another team members profile pic, then a pop-up should appear with information, including user's handle, current status, and currently application in use.

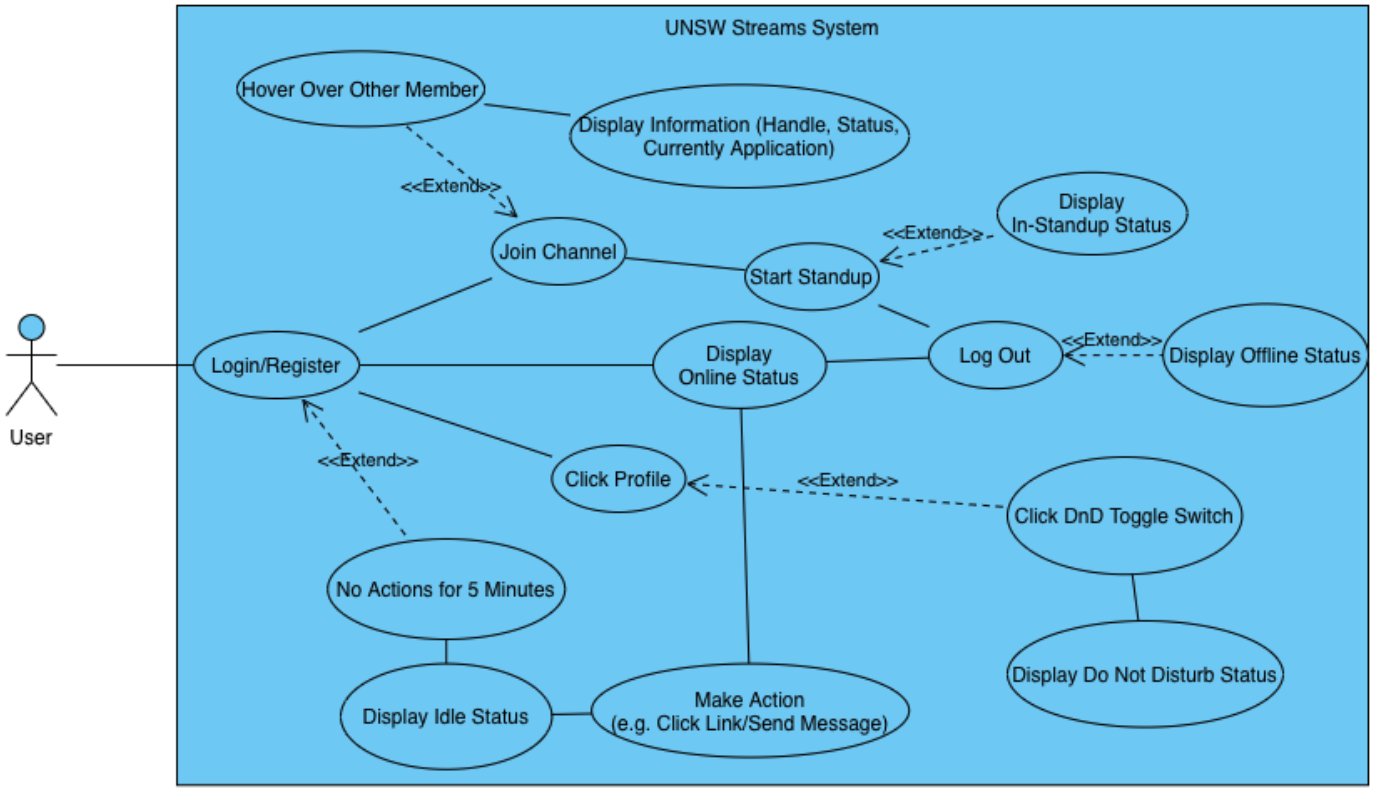
As a student, I would like to be able to automatically have an idle status after 5 minutes, so that I can let other team members known when I am not at my computer.

- Given I have not click on a link or executed an action on Streams for 5 minutes, when other members view the channel-details page, then they should see that I am currently on Idle, with a yellow icon on my profile picture.
- Given I execute an action by clicking a link or sending a message whilst being on Idle, when other members view the channel-details page, then they should see that I am currently now on Online, with a green icon on my profile picture.

As a student, I would like to be able to see when other team members are “In-Standup”, so that I don’t accidentally send irrelevant messages.

- Given a member or myself have started a standup in the channel, When I look at the channel members list, then all the members who are Online should change to In-Standup, with blue icons. Members who were previously Idle or DnD before the stand-up started will stay the same.

Use Case



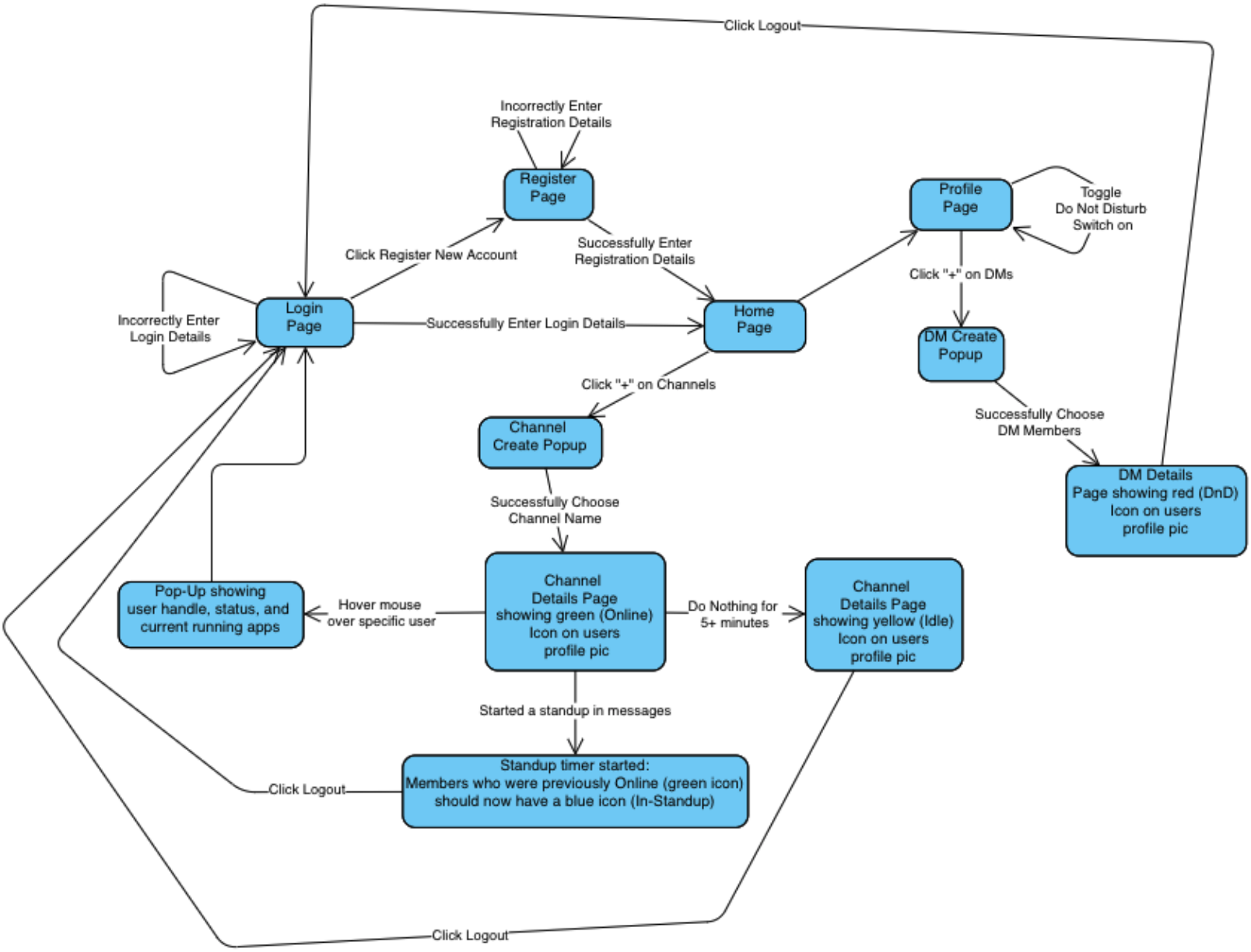
[Requirements] Validation

- Are these user-stories a sufficient summary of what I interviewed you about?
  - Wilson: Yes, each user stories sufficiently covers the different statuses I was talking about, and you also included tracking running applications, like Discord Enrichment, which is good. Overall, the reasoning for each user story is in line with my opinions.
- Are the relevant user acceptance criteria, in line with you functional/non-functional expectations?
  - Wilson: Yes, overall, the functional idea of icons is quite good, as its like most other platforms and its symbolically universal. The overall functional ideas are pretty accurate to what I wanted; however, it seems like there aren’t a lot of non-functional requirements which is reasonable given this is a purely functional capability.
- How well does this use-case diagram describe the problem we are trying to solve?
  - Wilson: This use case diagram is almost exactly in-line with how I initially interacted with Streams, which is good to see. Also, all relevant functional requirements are covered which is nice.

[Design] Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
user/status/set/v1 Sets the users status based on the following criteria: Online (1) if just registered/logged in and none of the other statuses. Idle (2) if user has been inactive for more than 5 minutes. Will require threading.timer(). In-Standup (3) if the user is Online, and a standup has started in a channel which they are a member of. Do Not Disturb (4) if the user has toggled the DnD switch on the front-end. Offline (0) if the user has logged out and has no active sessions. In () is each statuses relevant status_id. The statuses should be updated as frequently as possible, which can be achieved using threads.	POST	Parameters: {token, status_id} Return Type: {}	InputError if: - status_id does not exist - status_id is 4, and the user currrently has a “In-standup” status. Possible Edge Case: If a standup is called for more than 5 minutes, and the user does not make an action during that period, the Idle status will override the In-Standup status.
user/status/v1 Returns the status of the user specified.	GET	Parameters: {token, u_id} Return Type: {status_id}	InputError: U_id does not exist.
user/status/info/v1 Returns information on the user, including handle_str, status_id and current application. If status_id is 0 then current application should be empty string.	GET	Parameters: {token, u_id} Return Type: {handle_str, status_id, curr_aps}	InputError: U_id does not exist.

[Design] Conceptual Modelling (State)



[Planning – Benjamin Chau]

[Requirements] Elicitation

Jessica Chau – jessica.chau3@gmail.com

- **In what environment do you use a team-based communication platform, like Streams?**

In a work environment, secondary education during lockdown.

- **What capabilities do other platforms have that Streams does not provide that would benefit your use of the system?**

The ability to track attendance and participation of people using standups.

- **How would this [capability] improve the effectiveness/efficiency of the usage of the overall system?**

It would reduce the time spent on overhead (administration) associated with teaching secondary students online.

- **For the [capability], how would you expect it to be used within the system?**

Used for class activities with students and as a communication tool between staff members.

- **What scenarios have you experienced that could-be/were solved by [capability]?**

After asking a question to a group of students. I would like the ability to be able to see who has submitted an answer. Having standup track attendance and participation would allow this process to be automated.

Coordination between staff members when absentees and casual teachers are involved.
Attendance and participation tracking of standups will ensure tasks and classes are accounted for.

Proposed Solution

After a standup has concluded, a list of attendees and participants will be created and made available. Attendance will be based on who has sent messaged during the standup. Basic information for the specific standup will be displayed by clicking an icon with the option to be redirected to a more detailed page with info on all standups and the option to export data.

[Requirements] Analysis & Specification - Use Cases

User Stories and UAC

As a teacher, I want to be able to see who has sent a message during a standup so that I can get a better gauge of participation.

- Details icon will appear on the right of the sent message next to other actions (emote, pin share etc.)
- Clicking on icon will display a summary of attendees and participants.
- Attendees will list people in the standup.

- Participants will list people who have sent a message during the standup.
- Said summary window will have buttons to export data or redirect user to a more detailed statistics page.

As a teacher, I want to be able to see who attended a standup so that I can know who is aware of a task.

Scenario: Wants to see if the substitute teacher attended the standup.

Given: The user has navigated to the relevant standup.

When: The user hovers over and clicks the details icon.

Then: A window will appear with details about attendees.

User Case

Main Success Scenario

Step 1. Details icon appears next to the message after a standup is finished.

Step 2. User clicks on button.

Step 3. A window appears outlining attendance and participation of standup members.

Step 4. User obtains the necessary information.

Step 5. User closes window.

[Requirements] Validation

Jessica: These cases adequately describe my problems. Quality of life features such as a button to a more detailed stats page consisting of data from all standups and the ability to export data have also been added which is good.

[Design] Interface Design

Name & Description	HTTP Method	Data Types	Exceptions
standup/details/v1 Fetches the attendance data associated with a particular standup.	GET	Parameters: {token, standup_id} Return Type: {participants, attendance_rate}	InputError: If standup_id is not valid.
standup/details/all/v1 Fetches the attendance and participation data of all standups.	GET	Parameters: {token} Return Type: {standup_data_all} Where standup_data_all is a list of dictionaries which contain standup_id, participants, and attendance rate.	N/A
standup/details/export Stores the participation and attendance data of a particular standup in csv format.	GET	Parameters: {token, standup_id} Return Type: {}	InputError: If standup_id is not valid.

[Design] Conceptual Modelling (State)