COMP1531 TREATS

FUTURE PROBLEMS PLANNING REPORT

1 ELICITATION

1.1 TARGET USER INTERVIEWS

1.1.1 TARGET USER 1

Name: Nick Stevens

Email: nickstevens03@gmail.com

1. How do you primarily use tools like Treats? E.g. Solely for messages, with video calls?

Well, as a Software Engineering student, when using software such as Microsoft Teams or Treats, I tend to primarily use are the voice call, message, and file transfer functions. Since most of my degree's core courses delegate students to a group to work on a major project throughout the term, I've generally relied on Microsoft **Teams** and Treats to communicate with group members.

2. What do you think is the best way to use tools like Treats to remain engaged with your team and to keep each other motivated?

I've found that the largest detractor in motivation comes from a lack of knowledge of my group members' progress, which causes unnecessary stress. I think the most reliable methods focus on keeping up with progress on tasks between group members. Regularly meeting up through voice calls and standups makes it easier to communicate updates on progress, and any problems people might have. Even though messages allow for quick and easy communication, I've found that voice calls add an element of comradery and a sense of community, improving morale and motivation which is really important.

3. When using tools like Treats to manage group projects, do you find it easy to track everyone's progress and maintain good communication throughout the project?

I think the lack of progress tracking functions makes Treats less effective for groupwork compared to other communication software like **Teams**. The best thing you can do with it is probably to make a development plan through a call or messages, but this doesn't really allow for easy modification of the plan if things should change. Although plans can be managed by other software such as Trello, I'd prefer that all team communication could be bundled in one application. For future updates, it would be great to see the addition of a Task Board or Gantt Chart feature to help with planning and progress tracking.

4. Do you prefer to work synchronously or asynchronously on a group project ie. Working at the same time or working at your own pace

I much prefer to do work on my own. I do understand the importance of collaboration on a task with group members, but I've found this kind of collaboration to be frustrating because conflicting ideas can easily lead to a sort of Frankenstein final product. But I've found that the best way to remedy this is for all members working on a singular task to dump all of their ideas. This makes it easier to quickly identify conflicting approaches to a task. The app I've been using for this is **Miro**, but it would be great for Treats to have an integrated brainstorming board to do this.

5. When working on a written project such as a presentation or report, what tools/methods do you use to best share your work and collaborate on the project?

Since Microsoft is integrated with Teams, I generally use the Microsoft Office Suite over any other software. This is mainly because of the Microsoft SharePoint, which is a cloud server for each individual Microsoft Teams Channel that lets users upload files to a Channel, where only members of that Channel can access these files by simply clicking the File Tab. I've also seen tutors and course lecturers taking advantage of this by uploading course slides and notes to the course Channel SharePoint. So, to simply share my work with other group members, I've only needed to upload whatever project I'm working on to the Channel SharePoint.

6. Which platforms have you consistently used to manage team projects, and do you believe that when deployed within professional settings, Treats would meet all of a user's/client's requirements for managing group projects? If not, which features would you like to see added to Treats?

In the past when managing team projects, I've mostly used **Microsoft Teams**, especially at uni. I think it's good because it has a very **professional feel** to it, but it lacks a lot of the features that programs like **Discord** have. For writing reports and such, I think **Google Docs** is good because it makes collaboration in real time very easy, and it's definitely better than **One-Note**, although Teams does have a **One-Note integration** which can be quite useful for sharing work with a team all in the same place. I think that Treats would not be able to meet all of a user's or client's needs because it lacks calling features, as well as the ability to send files which is definitely vital in a professional setting. And having some kind of

progress tracker bar in a channel would be very useful, as well as having something like an issue board where you can mark off tasks.

1.1.2 TARGET USER 2

Name: Zach Ahmed

Email: zachahmed02@gmail.com

1. How do you primarily use tools like Treats? E.g. Solely for messages?

I mostly use things like Treats to work on group projects, I think things like Teams are good because you can easily set voice-meetings and calls and use it on your phone, and the calendar in teams is very useful. I mostly use the messaging and call features on it.

2. What do you think is the best way to user tools like Treats to remain engaged with your team and to keep each other motivated?

I think the best way to use tools like Treats to keep everyone engaged is video calls as it allows everyone to easily share their thoughts and their progress and collaborate, and it is also an extra layer of connection, which makes it easier to stay on task, and also makes it easier to have effective breaks if you're on a call because it feels more natural and interactive than using messages.

3. When using tools like Treats to manage group projects, do you find it easy to track everyone's progress and maintain good communication throughout the project?

Unless we hold a meeting, it is difficult to track where everyone is at. I know that apps like Teams have a calendar feature where you can assign deadlines for certain people, but I don't think this is the best way to track people's progress. It doesn't really help unless you have a conversation about it either through the chat or on a call, but I think something like a progress bar would be very useful.

4. Do you prefer to work synchronously or asynchronously on a group project ie. Working at the same time or working at your own pace

I prefer a bit of both, I like being able to come together every few days, say once a week, and then using that time together to talk about where everyone is up to, what we're going to do moving forward, and I guess any issues that people have. And then, of course, going away and doing your individual sessions, because I think that is the most efficient use of everyone's time, if you get the bulk of things done outside group meetings and then you come together and you're able to sort out any issues.

5. When working on a written project such as a presentation or report, what tools/methods do you use to best share your work and collaborate on the project?

For written projects and presentations, I find software like **Google Docs** to be the most effective platform. I like having the ability to collaborate on it at the same time, and edit work together. Also, you're able to see what everyone has done, or what they're going to start doing. Also, any applications which have a **screen-sharing option** are very useful, such as **Teams**, **Discord** or **Zoom**, so you can have one document open with everyone looking at it and be on the same page.

6. Which platforms have you consistently used to manage team projects, and do you believe that when deployed within professional settings, Treats would meet all of a user's/client's requirements for managing group projects. If not, which features would you like to see added to Treats?

At the moment, I am using **GitHub** to manage a team project, and I think that the fact that you can create a **to-do list** of tasks and assign things to people makes it easy to manage the project. You're able to split it up and see who's responsible for what, and therefore you can hold people accountable, and also check up on them if you can see that they are not getting things done on time. I think Treats would be able to meet user client requirements but I think there is a little bit more that needs to get done. For example, **video calls** and more **project management tools** like progress **checks** and, similar to GitHub, a **task-list** where people can assign work to themselves and check it off as they complete tasks.

1.1.3 Proposed Solution

In review of the insights gained through target user interviews, our team has created a proposed solution to the concerns of our target users surrounding difficulties in maintaining good communication within their teams, as well as maintaining a thorough approach to project management. Several of the key features which users were interested in seeing in the future were progress trackers, issue boards and video and voice calls.

Regarding their preferred usage of tools like Treats such as Teams, both target users referenced the use of voice and video call features, as well as messaging and file transferring, and calendar tools which allow meetings to be scheduled quite easily. While Treats has a messaging option within direct messages and channels, it does not accommodate the more complex features of calling, sending files and using a calendar to schedule meetings. Both users also conveyed a strong affinity to keeping their teams motivated and engaged through calling features, suggesting that being able to chat in a more natural

environment helps to improve morale, thus highlighting the necessity for such features to be added to Treats.

Regarding the tracking of progress within their teams in past group projects, and maintaining good communication, both users expressed the difficulty they have had as platforms they have used such as Teams do not have many progress-tracking features besides the ability to assign deadlines in the calendar. They concluded that discussing their progress in a meeting was the only effective way to do keep up to date with each other's work and to plan ahead, however this does not allow for any modifications to plans that may be made, and thus a progress bar or task board would be extremely helpful.

The importance of calling and progress tracking features is further illustrated in the manner that target users prefer to work, which was asynchronously in some cases and synchronously in others. Both users find asynchronous work to be the most productive, however this work would be made more effective by progress tracking features which would allow their team members to keep up to date with their work. The users also expressed that synchronous work, such as over voice or video calls, can be very helpful as it allows for collaborative problem solving, and receiving input from others as you work, hence calling features are essential to tools such as Treats.

Both users expressed that they have preferred to use tools which allow for collaborative work in past group projects, such as the Microsoft Office Suite which allows a Channel SharePoint within Microsoft Teams, and other programs such as Google docs for collaboration on reports. Target User 2 also noted the usefulness of screen-sharing options within tools like Teams, Discord and Zoom, and such features could be integrated into Treats in the future.

In closing, the insights provided by the target users were very useful in determining how individuals prefer to work remotely within group settings, and which features they find important to maintaining good communication, tracking progress and working collaboratively. The key features that we will looks to implement for the future of Treats include voice and video calls, as well as task boards and progress bars.

2 USER CASES

2.1 EPIC: WORK ORGANISATION

User Stories	User Acceptance Criteria
As a University Student, I want to be able to create a Task Board, so that my group members and I can plan and keep up to date with each other's progress on their assigned tasks.	• An empty Task Board is available in every Channel for channel members to edit. • Task Board is structured with 3 columns, from left to right, Not Started, In Progress, Finished. • A task can be added by any channel member • Tasks can be assigned to any channel member by any channel member • Viewing a task displays its title, description, user assigned, and date due. • When assigning a Due Date, a calendar is shown to the user that they can select a date. • When creating a new task, the user can initialise its title, description, due date, and member assigned • The number of unfinished tasks is shown on the 'Task Board'. • The task can be marked as completed by any user through clicking 'Mark as Completed' when viewing the task.
As a University Student, I would like to be able to create a Gantt Chart, so that I can visualize my project as smaller subtasks and monitor progress throughout the project timespan.	• An Empty Gantt Chart is available in every Channel • Structure of Gantt chart follows, 2 columns, left column contains task, right column shows timespan of each task. • Task timespan can either be illustrated as months, or number of weeks. • A task can be added by any channel member • Tasks can be assigned to any channel member by any channel member • Viewing a task displays its title, description, user assigned, start and end date. • When assigning a Due Date, a calendar is shown to the user that they can select a start date and end date. • When creating a new task, the user can initialise its title, description, start, end date, and member assigned • Number of days remaining until the task deadline, is displayed on each Task bar.
As a University Student, I want to be able to create a Brainstorming Board, so that my channel members and I can brainstorm and map out our ideas and approaches to tasks.	• An empty brainstorming board is available in every channel • Ideas are represented as post it notes • Any channel member can add notes • Any channel member can edit notes • Notes consist of a single text body • Notes are created by double clicking an empty space on the board
As a University Student, I want to be able to upload files to the Channel, so that I can have channel members access and view my work.	• File tab is available in every channel. A different file server is created for every channel. • Server is empty when creating a new channel. • File tab is structured with 3 columns, left to right, file name, name of uploader, date of upload • File upload accepts all types • Files can only be deleted by the original uploader.

2.2 Epic: Team Communication

Use Cases	User Acceptance Criteria
As a University Student, I want to be able to host calls with channel members, so that we have a better method of communication between channel members than text messages.	• Voice calls are available for every Channel • Any channel member can initiate a voice call • User who started the voice call joins the call automatically • All channel members are notified of a voice call starting in a channel they have joined • A channel can only have 1 active voice call • A user cannot be in multiple voice calls at the same time • Voice call ends when the creator of the call ends it, or all voice call members leave. • Voice call creator can remove users from call • Users in calls can screen share, showing the current window the user is viewing • Users in calls can mute their mic

3 Validation

3.1 Target User 1

Name: Nick Stevens

Email: nickstevens03@gmail.com

To what extent do you think the Epic, Work Organization, provides an adequate solution to the issue of group collaboration, and would you have any suggestions as to how to improve the current use case?

I think it is a successful solution to the issue of group collaboration due to the number of options open to the user to increase the efficiency and transparency of their group work. This also means each group can decide which organisation method will most benefit their workflow. Group members have access to a universal task manager which can also give them the ability to hold each other accountable as they can track the progress of their group members. Similarly, the open file tab will make collaborative documents a lot easier as each member can edit and view the same piece of work in real time. Both task Boards and Gantt charts allow for project members to stay updated on one another's progress, through creation of tasks assigned to project members. The issue of inflexibility within Project plans has also been minimized, by allowing for any user to edit the title, description, due date, and user assigned of any task.

3.2 TARGET USER 2

Name: Zach Ahmed

Email: zachahmed02@gmail.com

To what extent do you think the Epic, Team Communication, provides an adequate solution to the issue regarding the lack of communication features, and would you have any suggestions as to how to improve the current use case?

I think the issue of team communication has been very well addressed by the Epic, Team Communication, because it allows users to chat in a more natural and engaged way over voice or video calls. The call notification feature will make it easy to let your other team members know when you want to start a discussion with them, which may be more effective than simply notifying them over text. The screen sharing option is definitely great to improve communication of ideas and of the work that has been done, and it will help people to inform one another on new insights or problems to tackle. The mute option will make users feel at ease with being able to turn off their mic when they need to, to block out any distracting sounds or background noise. I think that the current case could be improved by having the option to record calls for later user, which would come in handy especially in professional settings such as lecturers recording their classes, or workers in industry recording seminars.

4 INTERFACE DESIGN

All Functions return a 403 Error on Invalid Token.

Task Board			
Name & Description	HTTP Method	Data Types	Exceptions
Channel/taskboard/create Create a new task on the channel's taskboard	POST	Body Parameters: (channelId, taskName, taskDescription, dateDue) Return type if no error: { taskId }	400 Error when any of: • Invalid ChannelId • taskName length < 1 or > 200 • taskDescription length > 200 • dateDue is in the past

			• User not a member of the channel
Channel/taskboard/assign Assign a user to an existing task on the channel's taskboard	PUT	Body Parameters: (taskId, userId) Return type if no error: { }	400 Error when any of: • Invalid taskId • userId does not reference a member of the channel • User not a member of the channel
Channel/taskboard/edit Edit the details of task on the channel's taskboard	PUT	Body Parameters: (taskId, taskName taskDescription, dateDue) Return type if no error: { }	400 Error when any of: • Invalid taskId • taskName length < 1 or > 200 • taskDescription length > 200 • dateDue is in the past • User not a member of the channel
Channel/taskboard/remove Removes a task from the channel's taskboard	DELETE	Body Parameters: (taskId) Return type if no error: { }	400 Error when any of: • Invalid taskId • User not a member of the channel
Channel/taskboard/listall List all the tasks on the channel's taskboard	GET	Body Parameters: (channelId) Return type if no error: { tasks: [{ taskId, name, desc, assignee, dueDate }] }	400 Error when any of: • Invalid ChannelId • User not a member of the channel

Gantt Chart			
Name & Description	HTTP Method	Data Types	Exceptions
Channel/gantt/create Create a gantt chart for the channel	POST	Body Parameters: (channelId, chartName, chartDescription, dateStart, dateEnd) Return type if no error: { chartId }	400 Error when any of: • Invalid ChannelId • dateEnd > dateStart • ChartName length < 1 or > 200 • ChartDescription > 200 • User not a member of channel

Channel/gantt/assign Assign a user to a gantt chart	PUT	Body Parameters: (chartId, userId) Return type if no error: { }	400 Error when any of: • Invalid chartId • UserId does not reference a member of the channel • User not a member of channel
Channel/gantt/edit Edit a gantt chart's details	PUT	Body Parameters: (chartId, chartName chartDescription, dateStart, dateEnd) Return type if no error: { }	400 Error when any of: • Invalid chartId • chartName length < 1 or > 200 • chartDescription length > 200 • dateEnd > dateStart • User not a member of channel
Channel/gantt/remove Remove a gantt chart from a channel	DELETE	Body Parameters: (chartId) Return type if no error: { }	400 Error when any of: • Invalid chartId • User not a member of channel
Channel/gantt/listall List all gantt charts in a channel	GET	Body Parameters: (channelId) Return type if no error: { charts: [{ chartId, name, desc, assignees, dateStart, dateEnd }] }	400 Error when any of: • Invalid ChannelId • User not a member of channel

Brain Storming Board			
Name & Description	HTTP Method	Data Types	Exceptions
Channel/board/create Creates a note on the channel's brainstorm board	POST	Body Parameters: (channelId, noteBody) Return type if no error: { noteId }	400 Error when any of: • Invalid ChannelId • User not a member of the channel • NoteBody length < 0 or > 200

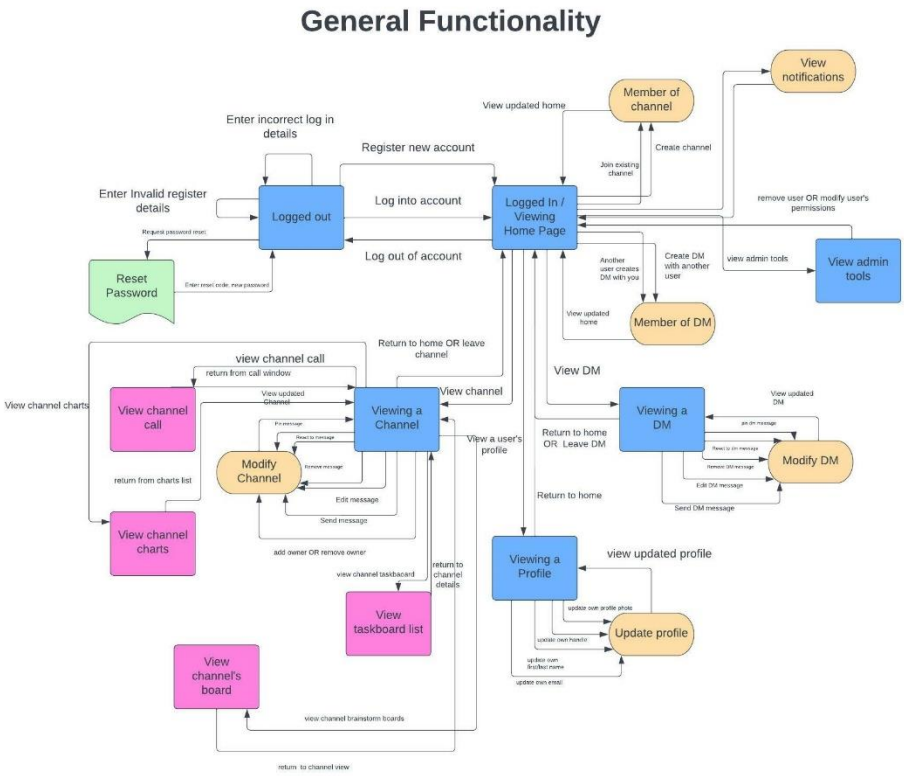
Channel/board/edit Edits a note on the channel's brainstorm board	PUT	Body Parameters: (nodeId, noteBody) Return type if no error: { }	400 Error when any of: • Invalid noteId • User not a member of the channel • NoteBody length < 0 or > 200
Channel/board/remove Removes a note on the channel's brainstorm board	DELETE	Body Parameters: (noteId) Return type if no error: { }	400 Error when any of: • Invalid noteId • User not a member of the channel
Channel/board/listall List all notes on the channel's brainstorm board	GET	Body Parameters: (channelId) Return type if no error: { notes: [{ noteId, noteBody }] }	400 Error when any of: • Invalid ChannelId • User not a member of the channel

Voice Call			
Name & Description	HTTP Method	Data Types	Exceptions
Channel/call/start Starts voice call in channel User who starts call automatically joins call as voice	POST	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • Call already active in the channel • User not a member of the channel
Channel/call/join/audio/v1 Joins an ongoing call in channel with voice	POST	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • User not a member of the channel
Channel/call/join/video/v1 Joins an ongoing call in channel with voice & video	POST	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • User not a member of the channel
Channel/call/leave/v1	POST	Body Parameters:	400 Error when any of:

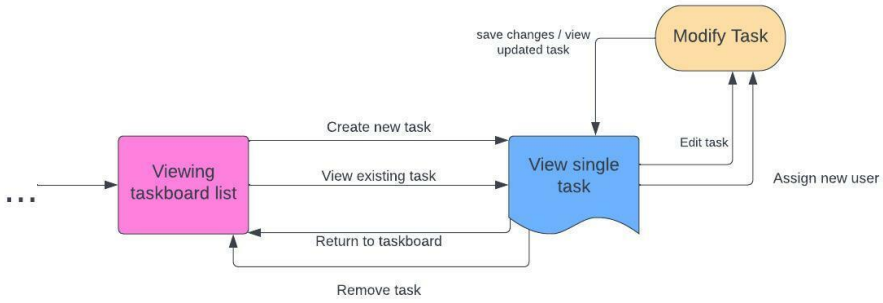
Leaves an active call in channel		(channelId) Return type if no error: { }	• Invalid ChannelId • No active call in the channel • User not a member of call • User not a member of the channel
Channel/call/mute/v1 Mutes audio input in an active call	POST	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • User not a member of call • User not a member of the channel • User already muted
Channel/call/unmute/v1 Mutes audio input in an active call	POST	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • User not a member of call • User not a member of the channel • User already unmuted
Channel/call/remove/v1 Removes a user from the active call	POST	Body Parameters: (channelId, userId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • UserId references a member not in the active call • UserId references a member who has channelOwner permissions • User does not have channelOwner permissions • User not a member of channel
Channel/call/end/v1 Ends an ongoing call in channel	DELETE	Body Parameters: (channelId) Return type if no error: { }	400 Error when any of: • Invalid ChannelId • No active call in the channel • User did not start the call and does not have channelOwner permissions

5 STATE DIAGRAMS (CONCEPTUAL MODELLING)

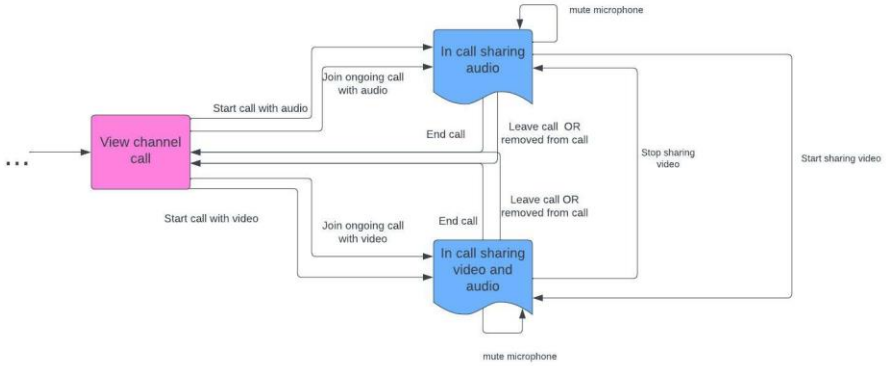
An overview of Treat's general functionality modelled as a state diagram. The new planned features, highlighted pink, have their individual state diagrams below.



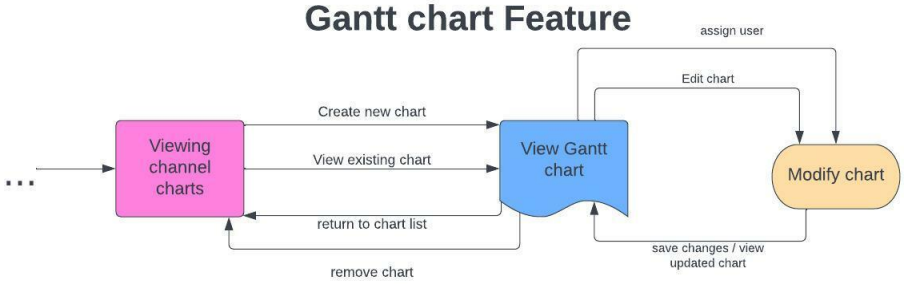
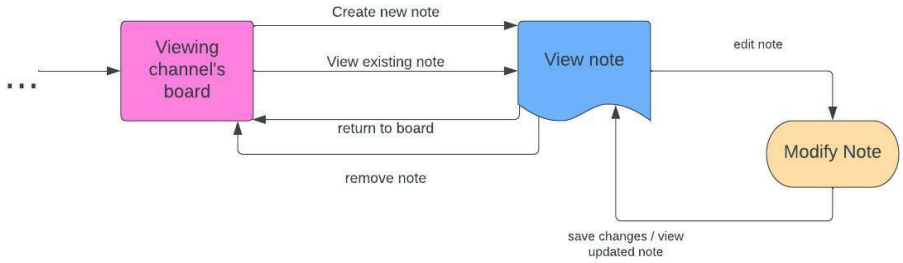
Taskboard Feature



Channel Call Feature



Gantt chart Feature



Brainstorm Board Feature