[Requirements] Elicitation

Questions:

How often and for what purpose do you use teamwork-driven communication tools such as Microsoft Teams, Skype, etc?

For these tools what benefits have you found in using them, and are there any limitations or drawbacks you've experienced?

What are some key features or functionalities that are missing from existing teamwork-driven communication tools that you would like to add?

By adding or changing these features what do you think the impact would be on other users that also use the platforms?

First User: (Name: Jason | Email: ojfm123@gmail.com)

Occupation: Actuary at Finity Consulting

1. I use platforms such as *Slack* on a daily purpose for work related things such as communication with colleagues and task organisation.

2. The benefits I've found with using Slack is that it is very easy to find other people in my organisation and contact them. I am also able to set different statuses to myself so others know what I'm working on/doing. So far I have not come across any limitations in my time using Slack.

3. I would like the ability to set automatic statuses during certain times of the day or during certain days of the week. That way if I forget to update my status the rest of my colleagues still know what I am doing.

4. Everyone would be able to determine mine and other users' statuses at all times of the day. This would also prevent users from having the wrong status when they forget to change them manually.

Problem:

The user statuses used by *Slack* rely on the users updating them frequently meaning that when users forget to update their status it conveys wrong and outdated information to their peers and colleagues.

Potential Solution:

Develop an automatic status updating system for the users. This could be done either by automatically detecting what the user is working on or simply by users setting different statuses at different times.

Second User: (Name: Xander | Email: xandersawyerwork@gmail.com)

Occupation: Tertiary Student

1. I use teamwork-driven communication tools like *Microsoft Teams* and *Zoom* almost daily for collaborating with my classmates and instructors. We use these tools for group discussions, online presentations, lectures, etc.

2. Using these tools allows for us to communicate and collaboration with team members easily and all the time regardless of the location of others. They let us be on the same page and ensure that no one misses important updates or deadlines. One limitation is that it can be a bit hard of non-tech savvy people to use these platforms.

3. A feature that would be useful is the ability to organise and categorise messages more effectively. For instance, it would be helpful to be able to filter messages by sender, date, or keyword, to find specific messages or conversations more easily.

4. Organising and categorising messages more effectively would make it easier for users to find specific information quickly instead of having to scroll through multiple chats, users, etc. This would reduce alot of frustration when trying to find one specific thing and would probably lead to an increase in productivity. I think it would enhance the overall user experience.

Problem:

It’s difficult for users to find specific pieces of information within the communication tool when there are not proper/easily usable organisation tools available.

Potential Solution:

Create a series of tools which allow users to search based on a series of specific categories to more accurately find what they are looking for. E.g search by name, search by keyword, search by time, etc.

Third User: (Name: Presit | Email: presitkaur@gmail.com)

Occupation: General Worker

1. As it currently stands, I do not use teamwork-driven communication tools frequently. In the instance that external communication is utilised, it is usually done through email (such as Outlook) or through text/telephone.

2. A major benefit of using these tools is that it streamlines the ability to reach out to multiple people at once, ensuring that they all receive the same message. Phone calling, however, is not the best suited for this as it relies on one-to-one communication with the recipients. Within this lies the limitation of relying on two different platforms to achieve specific communication needs.

3. The ability to easily schedule tasks on a group calendar or a program within an application would be highly beneficial. This way organisation of tasks can be accessed by all, and any conflicts can be communicated with other team members. Currently this can be achieved through emails however it tends to be finicky for both the sender and receiver and difficult to share with others.

4. I believe that having a simplified scheduling program that can be easily shared can greatly impact the productivity of a team. All members would have access to a shared schedule and can use the same application to bring up any conflict. Not only that but it would reduce the need to rely on other applications to achieve needs that depend on consistency and communication. For example, a team member not being able to download another application due to storage issues on their device would struggle to maintain the same level of productivity as someone who has the application installed.

Problem:

There is no simple and intuitive way to schedule tasks on a group calendar. Current methods are both finicky and confusing to use for both sender and receivers.

Potential Solution:

Create a simple and easy to use method of sharing and scheduling tasks with multiple people. This solution would require a good UI in order for it to be easily usable.

[Requirements] Analysis & Specification - Use Cases

User	User Story
1	As a member of a collaborative workspace, I want to be able to automatically update my user status, so that my colleagues are able to determine what I am working on without requiring me to manually update my status.
2	As a student who mainly works online, I want to be able to categorise and organise messages by a filter, so that I am able to find key information both more efficiently and with less hassle.
3	As a worker, I want an easier way to schedule tasks for multiple users, so that I am able to properly communicate deadlines to others without confusion.

User	Acceptance Criteria
1	**Scenario:** The user has switched to a different task. **Given:** The user has set up automatic user status updates. **When:** The user has switched to a different task. **Then:** The system changes the user's status to the new task.
2	**Scenario:** User is trying to find all messages from a given criteria. **Given:** The user has navigated to the search page. **When:** The user has selected a search type (name, time, contains). **And:** The user has indicated what is being searched for (name, time, word, phrase) **Then:** The system returns a list of everything that contains what the user searched.
3	**Scenario:** The user wants to send a scheduled task to multiple other users. **Given:** The user has created the task (set time, title, etc). **And:** The user has indicated who is being linked to the task. **When:** The user sends the task. **Then:** The system created a new task for the sender and linked users with the task details.

User	Use Case
1	**Use Case:** Automatically updating user status **Goal in Context:** A user wants their user statuses to update automatically depending on what task they are completing **Scope:** Current user information **Preconditions:** The user is part of the system, has set up automatic user status updates, and has changed what task they are working on **Success End Condition:** The user's status has been updated to the new task being done **Failed End Condition**: The user’s status had not been updated, and an error is thrown **Primary Actor:** The user **Trigger:** The user switches what task they are working on
	Use Case List
	The user begins set up of automatic status updates on their profile User selects which tasks correspond to which statuses System updates the automatic system with new user preferences User changes from their current task to another task System determines which new status the user should be under System updates the users stored profile to be of the new status System updates the viewable user profile to show the new status

User	Use Case
2	**Use Case:** Gathering all relevant data for the user **Goal in Context:** A user wants to find all relevant information of a specific field with a given value **Scope:** Communication tool information **Preconditions:** The user is part and authenticated in the system, and begins a search for a specific field with a given value **Success End Condition:** The system returns all information from that field that contains the value given **Failed End Condition**: The system cannot get the information and throws an error **Primary Actor:** The user **Trigger:** The user uses the searching function
	- **Use Case List**
	Users uses the search function System prompts user to identify a field and parameter of information they want to collect User enters in specific field and parameter and uses function System sorts potential data by whether the user is permitted to access it or not System searches through the sorted data to find all relevant information with the keys System orders fully collected data by relevance System returns all information to the user

User	Use Case
3	**Use Case:** Sending scheduled task to multiple users **Goal in Context:** A user wants to schedule a task for multiple users at the same time **Scope:** Multiple user information **Preconditions:** The user is part and authenticated in the system, and begins creates a new task and has a list of people the task is being sent to **Success End Condition:** The system updates the schedules for all users in the list to include the new task created **Failed End Condition**: The system cannot update the schedules and returns an error **Primary Actor:** The user **Trigger:** The user sends a task to other users
	- **Use Case List**
	The user creates a scheduled task The user uses the send task to other users function The system prompts the user to enter which task and which users are being included The user enters in the task and users System grabs the task and adds it to the other users tasks System updates the database so the users updates tasks are now included Added users are able to see the new task on their task list

[Requirements] Validation

User	Comment on Use Case
1	“The use case does describe the problem that I would like to be solved, however there is still where it could update based on certain set times, and not when I just switch tasks.”
2	“I think you’ve done a great job at showing what I wanted to put into the thing. I think you could be a bit more specific about what the field and parameters were but other than that it looks great.”
3	“This is a good way of displaying how collaborating on a task would work, as long as it is easy to use and isn’t fiddly like the others.”

[Design] Interface Design

Name & Description	HTTP Method	Data Types	Exceptions

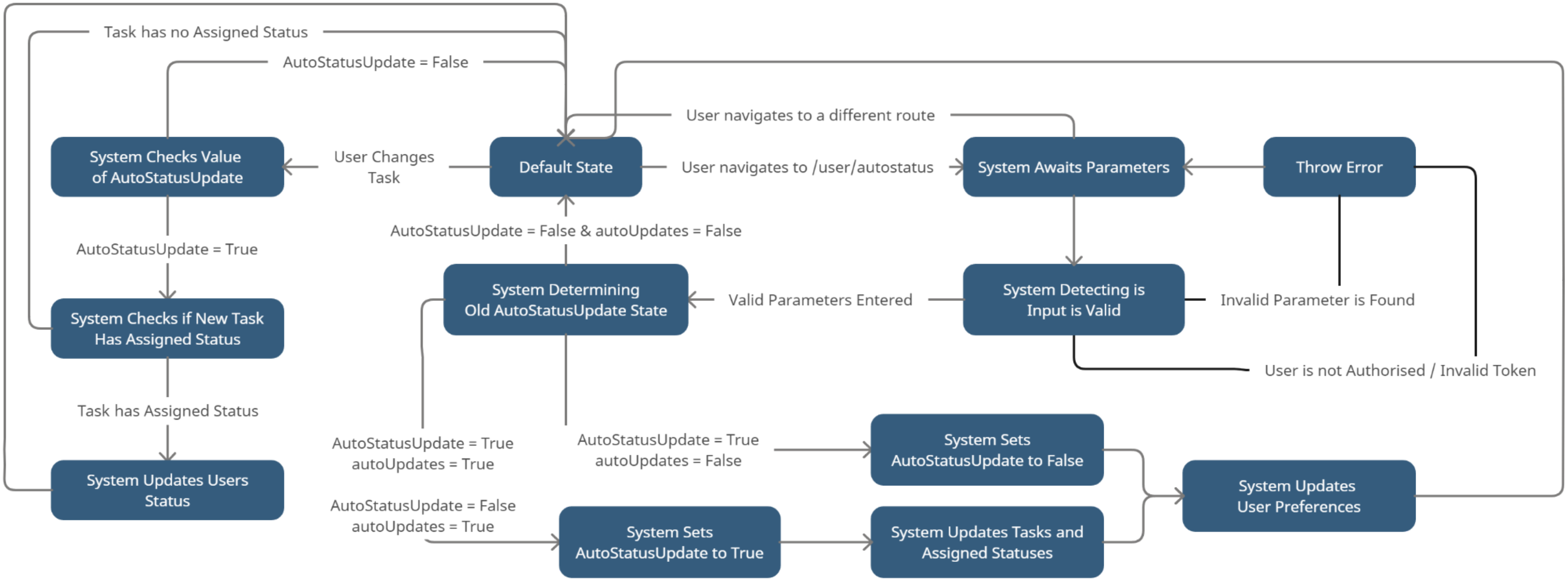
/user/autosetstatus
 	
Updates a users preferences about how and if their status will update automatically
 	
(This would also include the implementation of user statuses that would include routes such as: /user/setstatus)
	'PUT'	**Body Parameters:** {
		autoUpdates: boolean,
		tasks: {taskId, statusId}Array
		}
		
		Return if no Error: {
		autoUpdates: boolean
		}
			Err 403:
			- User is invalid
			
			Err 400:
			- tasks is empty
			- task is defined twice
			- invalid taskId
			- invalid statusId

/search/field
 	
Searches for all data within the given field that is related to the other parameter put in.
 	
field: enum of fields such as, content, timeSent, user, channel, dm
 	
Note: when searching for timeSent returns all messages within a certain tolerance.
	'GET'	**Body Parameters:** {
		field: enum,
		parameter: string,
		}
		
		Return if no Error: {
		searchResults:
		{message}Array ||
		{user}Array ||
		{channel}Array ||
		{dm}Array
		}
			Err 403:
			- User is invalid
			
			Err 400:
			- Field is empty
			- Field is an invalid field
			- parameter is empty

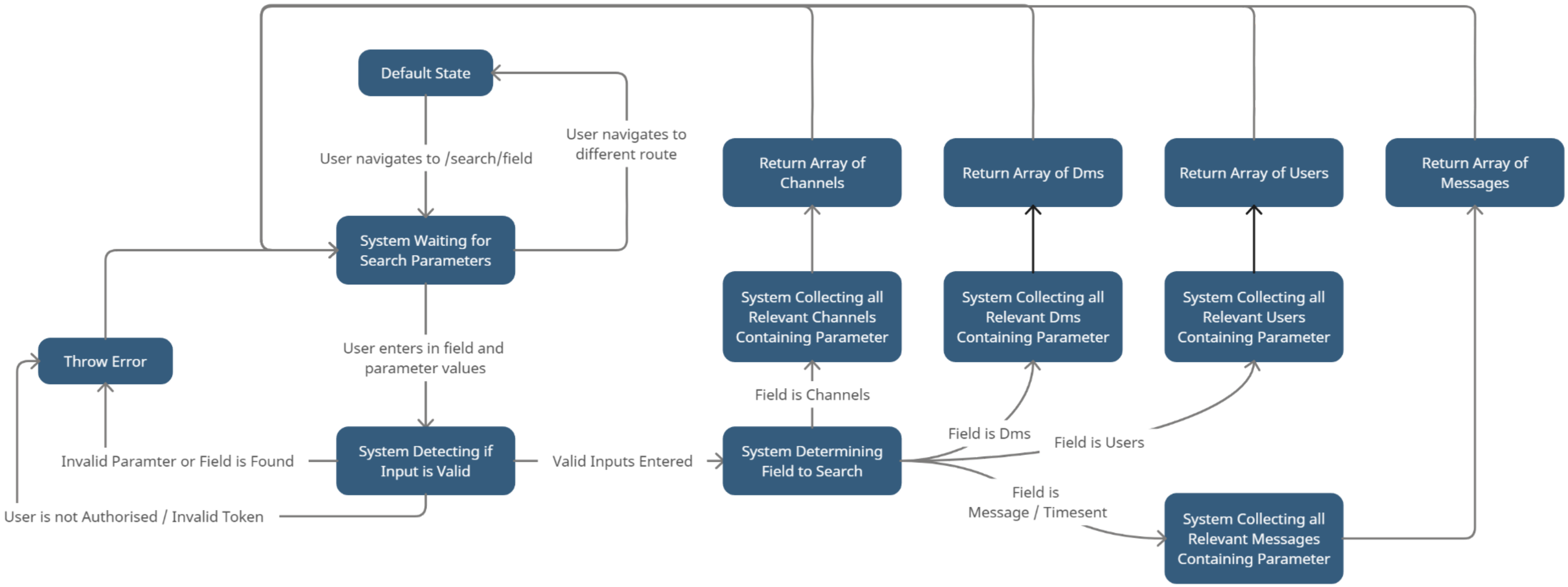
/task/share
 	
Share a task with multiple other users, meaning they all also have a duplicate task in their tasklist.
	'PUT'	**Body Parameters:** {
		task: task,
		Users: number[]
		}
		
		Return if no Error: {
		
		}
			Err 403:
			- User is invalid
			
			Err 400:
			- Task is empty
			- Users is empty
			- Users contains duplicates
			- Users contains a user that does not exist

[Design] Conceptual Modelling - State Diagrams

State Diagram User 1:


Task has no Assigned Status
AutoStatusUpdate = False
User navigates to a different route
System Checks Value of AutoStatusUpdate
User Changes Task
Default State
User navigates to /user/autostatus
System Awaits Parameters
Throw Error
AutoStatusUpdate = True
AutoStatusUpdate = False & autoUpdates = False
System Checks if New Task Has Assigned Status
System Determining Old AutoStatusUpdate State
Valid Parameters Entered
System Detecting is Input is Valid
Invalid Parameter is Found
User is not Authorised / Invalid Token
Task has Assigned Status
AutoStatusUpdate = True autoUpdates = True
AutoStatusUpdate = True autoUpdates = False
System Sets AutoStatusUpdate to False
System Updates Users Status
System Updates User Preferences
AutoStatusUpdate = False autoUpdates = True
System Sets AutoStatusUpdate to True
System Updates Tasks and Assigned Statuses

State Diagram User 2:



Default State
User navigates to /search/field
User navigates to different route
System Waiting for Search Parameters
User enters in field and parameter values
Throw Error
Invalid Paramter or Field is Found
System Detecting if Input is Valid
User is not Authorised / Invalid Token
Valid Inputs Entered
System Determining Field to Search
Field is Channels
Field is Dms
Field is Users
Field is Message / Timesent
System Collecting all Relevant Channels Containing Parameter
System Collecting all Relevant Dms Containing Parameter
System Collecting all Relevant Users Containing Parameter
System Collecting all Relevant Messages Containing Parameter
Return Array of Channels
Return Array of Dms
Return Array of Users
Return Array of Messages