1. Elicitation

Target User 1:
Name: Carlos Martinez
Email: carlos.martinez1432@hotmail.com

Target User 2:
Name: Ashley Kim
Email: k.ash444@gmail.com

Questions:

1. Have you ever used messaging apps like Slack, Teams, or Messenger for teamwork-driven communication?

2. If so, what did you like about these apps and what problems did you encounter?

3. In your experience, what are the most important features that a teamwork-driven messaging app should have?

4. What are some common problems that you encounter when using messaging apps for team communication, and how do you typically solve these problems?

5. Are there any specific pain points or challenges that you have encountered with existing messaging apps that a blocking feature would help to address?

Answers:

Target User 1:

1. Carlos has used Slack and Teams for team communication. He likes that these apps allow for easy collaboration and communication between team members, but he has encountered problems with message overload and difficulty finding important messages.

2. Carlos thinks that important features for a teamwork-driven messaging app include the ability to easily search for and find messages, the ability to organize conversations by project or topic, and the ability to integrate with other productivity tools such as calendars and task lists.

3. Carlos has found that message overload and difficulty finding important messages can be a common problem when using messaging apps for team communication. He typically solves these problems by using the search function to find specific messages and creating separate channels or threads for different projects or topics.

4. Carlos believes that a blocking feature would be helpful for addressing issues with unwanted messages in team communication apps.

Target User 2:

1. Ashley has used Messenger for team communication in the past, but found it difficult to keep track of conversations and messages. She has also used other productivity tools such as Trello to manage team tasks and projects.

2. Ashley thinks that important features for a teamwork-driven messaging app include the ability to easily organize conversations and messages, the ability to track team tasks and deadlines, and the ability to integrate with other productivity tools.

3. Ashley has encountered problems with difficulty keeping track of conversations and messages when using messaging apps for team communication. She typically solves these problems by creating separate chat groups for different projects or topics, and by using other productivity tools to manage tasks and deadlines.

4. Ashley believes that a blocking feature would be helpful for addressing issues with harassment in team communication apps.

Proposed Solution:

Based on the feedback gathered from Carlos and Ashley, it seems that a teamwork-driven messaging app should have features that allow for easy organization of conversations and messages, integration with other productivity tools, and task tracking. In addition, a blocking feature would be helpful for addressing issues with unwanted messages or harassment. A potential solution could be to develop a messaging app that integrates with existing productivity tools and allows for easy organization of conversations and messages by project or topic. The app could also include a blocking feature to help address issues with unwanted messages or harassment.

2. Analysis & Specification

User stories:

1. *The user would like to be able to block users to help sort through important messages*

Acceptance criteria:

- There is an option to block certain users.
- The users blocked should not be allowed to send messages to the user or invite the user to channels/dms.
- The selected users to block are hidden from the users view as well as their messages.

2. *The user would like to be able to keep track of conversations and messages to maintain productivity*

Acceptance criteria:

- When joining or creating channels the user is able to create folders of servers and add tags.
- The user can create their own tags for organization, which is only visible to the user.
- The folders are only viewable by the user but the functionality remains the same.

3. *The user would like to be able to search and find messages to help organize his servers and increase workflow.*

Acceptance criteria:

- There is an option to search through messages.
- The messages that appear in search will be relevant to the user's input.
- The messages will only be from the relevant channels the user is in.
- The user will be given a series of parameters for their search such as the timeframe, key words and users mentioned.

Use case: Blocking a user

Background information:

Use case: “Blocking” a certain user from a particular user's view so they do not see their messages.

Goal: Given a particular user/list of other users any user should have the ability to hide the list of users from appearing including their messages and invite request to channels/dms.

Scope: Black box design principles apply and the blocked users will be added in the user profile interface with a list of blocked userIDs.

Preconditions: Both the users have to exist on the server and have been registered with a userID.

Success end condition: If the users chosen by the user to block are successfully blocked, the users will be added to a list of blocked users and will remove the blocked users from the users view.

Failed end condition: The users on the block list will remain the same status for the authorized user, i.e no changes will be made to their current ‘blocked’ status.

Primary actor: Any registered user.

Trigger: The user will interact with a “block” option on the other user's profile which will be shown on the frontend.

Example use case:

User: Interacts with ‘block’ prompt on specific users profile.
System: Confirms the other user and the authorized user are existing registered members.
System: “Are you sure you want to block this user?”
User: Confirms prompt
System: Adds the blocked users userID to the authorized users list of blocked users.

3. Validation

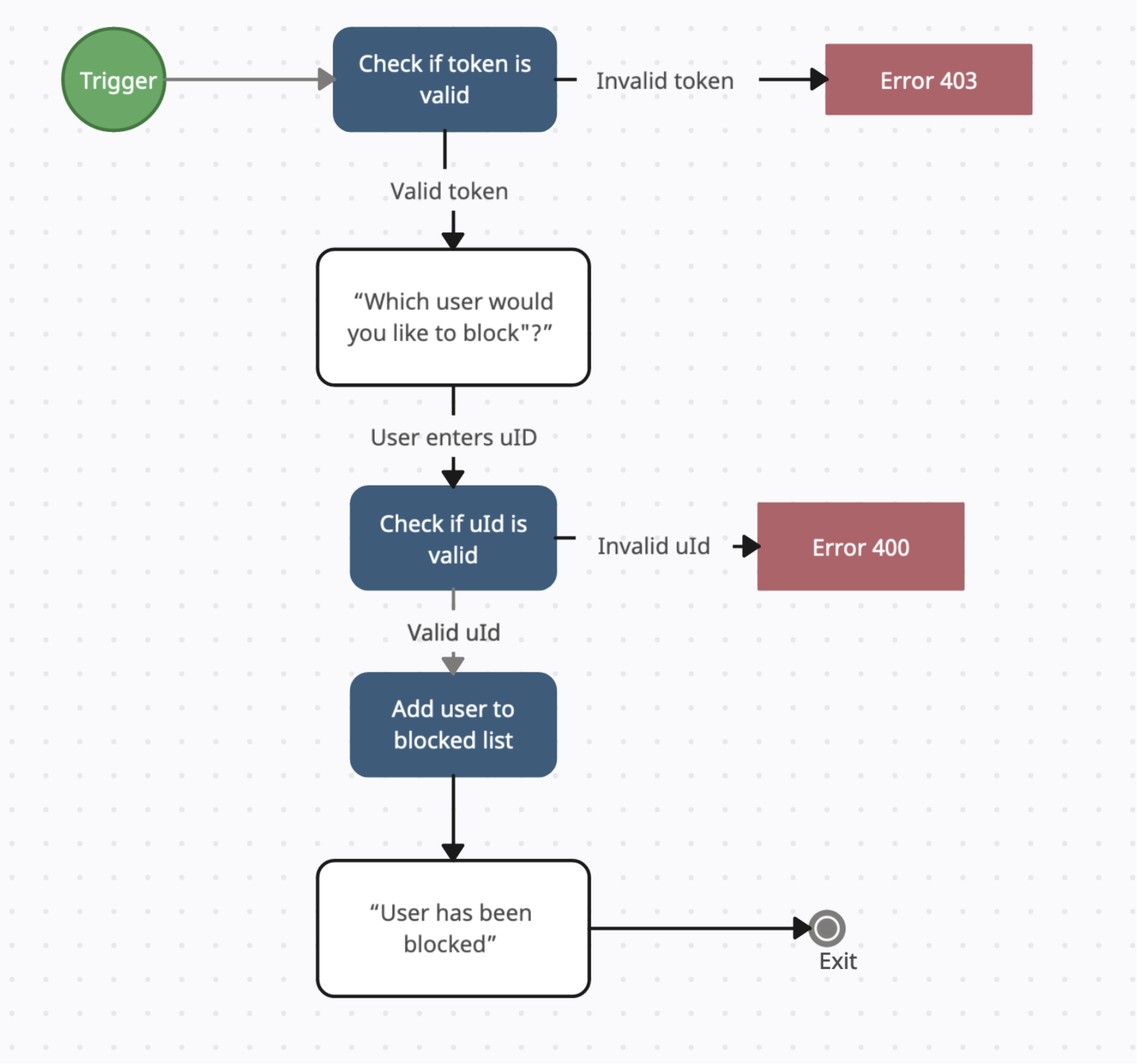
In our validation we asked Ashley again for their thoughts regarding our solution for blocking users. Ashley was happy to see that a straightforward implementation for blocking users would be put in place and that it does exactly what she wants it to do. She believes that it will increase her workflow and organization on Memes by filtering unnecessary messages and channel/dm invites.

Interface Design:

Name & Description	HTTP Method	Data Types	Exceptions

`users/block/v1`
Given a user with uId, add the user with ID to the authorized users list of blocked users.
Blocked users' messages should not be visible and users who are blocked will not be able to invite the user to channels or DMs.
Global owners are not able to be blocked.
	POST	**Body parameters:**
		`( uId )`
		Return type if no error:
		`{}`
			403 Error when:
			● `uId` refers to a global owner
			400 Error when:
			● `uId` does not refer to a valid user

State diagram:


Trigger
Check if token is valid
Invalid token
Error 403
Valid token
"Which user would you like to block"?"
User enters uID
Check if uId is valid
Invalid uId
Error 400
Valid uId
Add user to blocked list
"User has been blocked"
Exit