

## [Requirements] Elicitation

Find 2-3 people to interview as target users. Target users are people who currently use a tool like flockr, or intend to. Collect their name and email address.

Develop a series of questions to ask these target users to understand what *problems* they might have with teamwork-driven communication tools that are currently unsolved by flockr. Give these questions to your target users and record their answers.

### Target User 1

Name: Xiaojia Qiu

Email address: [cherryqiu1225@gmail.com](mailto:cherryqiu1225@gmail.com)

Q1: What would you usually use a teamwork-driven communication tool like Flockr for?

A1: I like to use them for formal situations such as work, course projects, society workshops. I wouldn't normally use them unless they are required by the course or if my group members choose to use them.

Q2: What features would you expect or like to see from Flockr?

A2: I think a team board that organizes tasks or to-do lists would be really helpful especially when working with a large group. For example, when a team leader posts a notification or a task in the group, members have to click a button to confirm that they've read and understood it. This way the team leader can see who has or hasn't checked their messages and take further actions to make sure that everyone is on the same page.

Breakout rooms or smaller groups within a large group would also be cool features to have, so that small groups can work separately on different tasks that are part of the big task. Being able to communicate with the main room is very important. I think this would increase teamwork efficiency as the main room won't be spammed with messages talking about different tasks. The team leader can keep track of where each group is up to easily by going into different breakout rooms, and make announcements or comments in the main room so that everyone can see them.

Other features like polls, private messages, reminders can also be quite helpful.

Q3: What problems, if any, would you anticipate arising when using a tool like Flockr?

A3: I'm worried about people sending inappropriate messages into the chat. I think communication tools should have a system that checks messages being sent and blocks inappropriate words. They could also give a warning to group members when sensitive

messages are being sent. For example, messages relating to money transfers or passwords. Otherwise they could have an age restriction on public groups or features to report and block an inappropriate user.

Q4: Have you used any similar teamwork-driven communications tools in the past? If so, what features did you like and what features did you dislike?

A4: Yes, I've used tools like zoom, Microsoft Teams, DingTalk and many more. I liked the screen sharing and onscreen whiteboarding feature of zoom which is great for brainstorming. The active status feature - being able to view team member's online status - can be good and bad at the same time. Also I don't like how I could be added into a random channel without my confirmation.

Q5: What would you have personally improved if you were developing a teamwork-driven communications tool?

A5: I would spend more time on ensuring the safety of user information, for example, add the ability to set personal information only visible to certain people, and require confirmation when being added into a channel.

## **Target User 2**

Name: Mitchell Getros

Email address: [mitchwas@gmail.com](mailto:mitchwas@gmail.com)

Q1: What would you like to use Flockr for?

A1: I'd like to use Flockr for collaboration on whatever I might be doing. It could be work or education related, such as a group assignment or project, or it could be for me and a couple friends trying to plan a trip away.

Q2: What features would you expect or like to see from Flockr?

A2: The key thing I'd like to see is a way to organise messages into categories and potentially even a collaborative to do list would be useful. A working collaborative whiteboard would also be a nice addition

Q3: What problems, if any, would you anticipate arising when using a tool like Flockr?

A3: Hardware incompatibilities would be a key one, a lot of people I know use mac based systems where Windows based systems are easier to develop for, so I'd like to see it be compatible on both systems. A potential way around this would be to make it a website/browser extension, but they're known for being slow and sluggish so the problem there would be program optimisation.

Q4: If you were developing Flockr, what features would you add?

A4: For starters it needs an inbuilt video call feature. A way to show if you're online or offline, and a way of non-verbal communication, so basically text messages. On top of that I'd make it easy to navigate between multiple different groups, so you can participate in more than one team at a time

Q5: Have you used any similar teamwork-driven communications tools in the past? If so, what features did you like and what features did you dislike?

A5: Microsoft teams and discord are both similar tools. I like the way that when presenting a PowerPoint on Microsoft teams you're able to either follow the presenter or go through the slides at your own pace. I like the way that on discord you can organise chats in such a way that it is easy to keep a track of where everything is

Q6: Have you used any similar teamwork-driven communications tools in the past? If so, how would you have personally improved them?

A6: Microsoft teams feels a little intrusive and heavy, and can be difficult to navigate. Discord is very barebones, all it really has is the communications tools and nothing more, so I'd like to see it start to implement presentation modes, previews of documents and features that could be useful in an education context

### Target User 3

Name: Allan Goodman

Email address: [epicallrounder.contact@gmail.com](mailto:epicallrounder.contact@gmail.com)

Q1: What would you like to use Flockr for?

A1: I would like to use Flockr for project management communication in order to effectively organise and delegate uncompleted tasks

Q2: What features would you expect or like to see from Flockr?

A2: Chat options, inbuilt git support, possibly video/voice calls. Cloud file storage would be great to see too.

Q3: What problems, if any, would you anticipate arising when using a tool like Flockr?

A3: I don't know

Q4: If you were developing Flockr, what features would you add?

A4: see question 2

Q5: Have you used any similar teamwork-driven communications tools in the past? If so, what features did you like and what features did you dislike?

A5: I have used Confluence/Bitbucket and Slack. Confluence/Bitbucket were good, however they were two separate products that I would have liked to have been one compact product. Slack was good, however it seemed to be too chat heavy, and didn't have many other features.

Q6: Have you used any similar teamwork-driven communications tools in the past? If so, how would you have personally improved them?

A6: following on from question 5, I would improve confluence/bitbucket by combining them, and i would improve slack by adding more features, e.g. git support and calendars

## [Requirements] Analysis & Specification - Use Cases

Take the responses from the elicitation and express these requirements as **User Stories**. Document these user stories. For each user story, add User Acceptance Criteria as notes so that you have a clear definition of when a story has been completed.

Once documented, generate at least one use case that attempts to tell a story of a solution that satisfies the requirements elicited. You can generate a visual diagram or a more written-recipe style, as per lectures.

### User Stories

1. As a user, I want to make todo lists and create tasks in the channel so that everyone knows what needs to be done.
  - Only members of a channel can see the create list button
  - Lists will be placed in a column on the right side of channel page, tasks will be placed as boxes inside the lists
  - When creating a list, the user clicks the 'create list' button, types in a name for the list and choose if the list is public or private
  - The user can create multiple lists
  - When creating a task, the user clicks the 'create task' button, types in a name for the task, descriptions of the task and have the option to choose a due date
  - A calendar will be displayed for the user to choose from
  - Due dates that have already passed will be displayed in red colour
  - All channel members can create public or private todo lists and tasks
  - Private lists and tasks within them can only be viewed and edited by the user who created them
  - Public lists and tasks within them can be viewed and edited by all members of the channel
  - Only members who can view the lists can see the create task button
  - List name and task name cannot be more than 20 characters long
  - Task description cannot be more than 1000 characters
2. As a team leader, I want team members to confirm the tasks added to todo lists so that I know who has read the tasks.
  - All tasks from public to-do lists have a confirm button on the right that is initially greyed out
  - When confirm button is clicked it turns into green
  - A name list will be generated when people confirm tasks
  - Only channel owners can view name lists of tasks by clicking on the task names
  - The creator of tasks automatically confirms them
3. As a user, I want to set my user profile to be visible to certain people so that my friends can see my name and email but strangers can't.

- In the profile page, the user can choose from the users list which users to add into the trust list.
  - The trusted users can view the user's profile but all other users cannot.
  - The default setting is that only the user himself can see his profile until he adds people into the trust list.
4. As a user, I want to be informed before being added to a channel, so that I won't be added into random channels that I don't want to join.
- When a user invites another user to join a channel, an invitation is sent by email.
  - The invitation email contains the inviter's name and the invited channel name: "[Inviter name] has invited you to join [Channel Name], please click the button to confirm invitation".
  - A 'confirm' button containing a link to the flockr webpage is included in the email
  - Once confirm button is clicked, the user is redirected to flockr webpage and added to the invited channel, a message "Invitation confirmed!" is printed.
  - If no action is taken, the user will not be added into the channel.
5. As a user, I want to be able to report users so that people who send inappropriate or rude messages in channels can be removed.
- Messages are checked by the application before sending in order to block some inappropriate words
  - Each message has a 'report' icon on the bottom right
  - When an user is reported, an email will be sent to channel owners containing the message, sender name, channel name and reporter name.
  - Channel owners can then take actions to remove or give warnings to the sender
6. As a potential user, I want separate, easily navigable text channels so that I can effectively organise and work on multiple things at once.
- A list of all channels the user is in will appear on a sidebar on the left
  - Clicking on a channel name will navigate the user to that channel
  - The channel the user is currently in will be highlighted
7. As a potential user, I want a collaborative to-do list so that I can effectively organise and delegate uncompleted tasks.
- A to-do list button will appear above the channel list
  - Clicking the button will navigate to the to-do list, which will be multiple labelled columns containing all things to do related to that column

- There will be a button which allows users to add more items and select who it is assigned to and what category it falls under and when it is due
- Users can only assign tasks to other users who are in channels with them
- There will be a button to change view, changing from multiple column lists to a calendar instead, with to do items appearing on the calendar under the date/time they are due at
- Upon clicking on a listed to do item, there will be an option to mark it as “complete”

8. As a potential user, I want a collaborative whiteboard so that I can easily brainstorm and share ideas with teammates.

- On the top bar of each channel, there will be a button which opens a whiteboards section
- In this section, there will be an option to create a new whiteboard
- There will also be an option to delete any whiteboards you have made, or any whiteboards if you are a channel owner.
- The whiteboard will be a big rectangular section with a similar interface to ‘Paint’, allowing any users in the channel to draw on it
- Users will be able to pick what colour they are drawing with from a colour wheel
- Users will be able to pick their line thickness from a dropdown menu next to the colour wheel
- There will be an undo and redo button next to the colour wheel

9. As a potential user, I want hardware compatibility across different systems so that my Mac based associates can also use Flockr.

- Flockr will work on Mac

10. As a potential user, I want an inbuilt video call feature so that I can see and talk to group members.

- Next to the whiteboard button, there will be a video call button
- This will initiate a video call and send a prompt in that channel’s chat for other users to join
- The video call will display a grid of the faces of everyone currently in the call, with users not using webcams having their names displayed instead
- On the toolbar at the bottom of the video call, there will be a button that toggles the muting of the user’s microphone
- Next to the microphone toggle button, there will be a button to leave the video call
- The call leave button will instead be a call end button for the user who initiated the call, which will end the video call
- Users will be able to see and talk with other users in the channel

11. As a potential user, I want to be able to see when other users are using Flockr or not so that I can tell if they are currently available.

- Underneath each user in the user list, there will be status message that either reads “online” or “offline” depending whether that user is currently using Flockr or not
- The user list will be accessed by a button on the top bar

12. As a potential user, I want powerpoint integration so that I can follow the powerpoint with the presenter or go through it at my own pace.

- On the top bar, there will be a button that begins a powerpoint presentation
- This allows the user to upload the powerpoint they wish to present
- Upon starting the presentation, a prompt will appear in the channel’s chat for other users to join
- Upon joining, other users will be able to follow the presenter, or move freely through the powerpoint
- There will be a button to toggle whether the presenter is followed or not in the toolbar at the bottom
- There will be a button on the toolbar at the bottom to leave the presentation
- If the presenter leaves the presentation, the presentation ends

13. As a potential user, I want to be able to preview a variety of documents so that I don’t have to open them with another program to view them.

- Users will be able to paste a variety of documents into the chat
- Clicking on the document will bring up a preview of the document
- There will be buttons to navigate and close the preview in the toolbar at the bottom

14. As a potential user, I want inbuilt git support so that I can directly view git projects through Flockr.

- Posting a git link in a channel will bring up a preview of that project
- This preview will be a window equivalent to bringing the project up in a web browser

15. As a potential user, I want Flockr to provide cloud storage so that project related files can be stored and accessed without the need of an external program.

- There will be a button on the top bar which opens a menu where all uploaded documents for that channel reside
- Users will be able to upload documents to the channel in this menu
- Users will be allocated 50GB each
- When a user views their profile, there will be a section where they can view all their uploaded files



- This section will show the information on the files, including the file size, the channel it was uploaded in, and the date it was uploaded
- This section will also allow the user to remove any files they have uploaded

## Use Cases

### Use Case 1

- **Use Case:** Create Todo List
- **Goal in Context:** Users need to create todo lists within channels
- **Scope:** Flockr, team communication tool
- **Level:** Primary Task
- **Preconditions:** The user is a member of Flockr and a member of the channel
- **Success End Condition:** The list name is of valid length
- **Failed End Condition:** The list name is too long
- **Primary Actor:** User
- **Trigger:** User clicks the 'create list' button

#### Main success scenario 1

Step 1: Channel member clicks 'create list' button

Step 2: Flockr asks user to input a name of the list

Step 3: User types in a name

Step 4: User clicks 'create' button

Step 5: Flockr checks that task name is of valid length

Step 6: Flockr adds the list into the channel

### Use Case 2

- **Use Case:** Create Task
- **Goal in Context:** Users need to create tasks within todo lists of channels
- **Scope:** Flockr, team communication tool
- **Level:** Primary Task
- **Preconditions:** The user is a member of Flockr and a member of the channel, and has permission to view and edit the selected list
- **Success End Condition:** Task name and descriptions are of valid length
- **Failed End Condition:** Task name or description is too long
- **Primary Actor:** User
- **Trigger:** User clicks the 'create task' button

## **Main success scenario 2**

Step 1: User clicks 'create task' button within a todo list

Step 2: Flockr asks user to input a name and a description of the task

Step 3: User types in a name and a description

Step 4: User clicks 'due date' button

Step 5: Flockr displays a calendar for the user to choose from

Step 6: User selects a date

Step 7: User clicks 'create' button

Step 8: Flockr checks that task name and description are of valid length

Step 9: Flockr adds the task into the todo list, task is automatically marked as confirmed

## **Use Case 3**

- **Use Case:** Confirm Task
- **Goal in Context:** Members need to confirm tasks within todo lists of channels
- **Scope:** Flockr, team communication tool
- **Level:** Primary Task
- **Preconditions:** The user is a member of Flockr and a member of the channel, and has permission to view and edit the selected list
- **Success End Condition:** Task has not been confirmed yet
- **Failed End Condition:** Task has already been confirmed
- **Primary Actor:** User
- **Trigger:** User clicks the 'confirm task' button

## **Main success scenario 3**

Step 1: User clicks 'confirm task' button of a task that needs to be confirmed

Step 2: Flockr checks if the task has already been confirmed

Step 3: Flockr adds the user into a list of users who has confirmed

Step 4: Flockr displays confirm button as green

## **Use Case 4**

- Use Case: Video call
- Goal in Context: User has a video call with other users
- Scope: Flockr, video call software
- Level: Primary task
- Preconditions: User is logged in and is in a channel
- Success End Condition: User has and finishes video call
- Failed End Condition: User does not finish user call
- Primary Actor: User
- Trigger: User presses video call button

### **Main success scenario 4**

Step 1. User presses video call button

Step 2. A video call is started with the user as the sole member

Step 3. A join video call prompt appears in the channel

Step 4. Other users press the video call prompt

Step 5. Other users are joined to the video call

Step 6. Users press the microphone mute button

Step 7. Users microphones are muted

Step 8. Users press the microphone mute button

Step 9. Users microphones are unmuted

Step 10. Other users press the call leave button

Step 11. Other users are disconnected from the call

Step 12. User presses the call end button

Step 13. The call is ended

## **[Requirements] Validation**

With your completed use case work, reach out to the 2-3 people you interviewed originally and inquire as to the extent to which these use cases would adequately describe the problem they're trying to solve. Ask them for a comment on this, and record their comments in the PDF.

### **Comments from Target User 1:**

I think the use cases describe my needs accurately and seem to solve my problems very well. The chatting tools I use don't really have similar todo list features so I'm looking forward to using them. They would be really helpful for my team work or even just organizing my personal things. I like how the todo lists can be set to private so that no other channel members can view them. Perhaps it can be improved so that people can switch between public and private status anytime, or be able to hide just a particular task instead of having to set an entire list to private.

### **Comments from Target User 2:**

Video call functionality is good, perhaps add a button to turn on/off camera as well as the microphone.

Todo list is good, maybe check for appropriate characters or format along with the valid length check.

Taskboard is good, no problems as of yet.

### **Comments from Target User 3:**

The implementation of a video call is the essence of what I would want from a team communication tool, as it effectively lets us communicate efficiently, in a way that allows us to mimic a face to face scenario.

The implementation of a todo list is a secondary priority as an addition to a communication tool as it doesn't necessarily facilitate the communication itself, however it does allow for a greater understanding of ideas, when supplemented with a primary form of communication, either by video, voice, or text messaging systems

Allowing people to create tasks in order to organise workflow is another useful supplement to communication methods provided by a team based communication tool

Additionally, confirming the completion of these tasks can be used as a form of communication itself, letting other team members know that one or more tasks has been completed.

## [Design] Interface Design

Now that we've established our *problem* (described as requirements), it's time to think about our *solution* in terms of what capabilities would be necessary. You will specify these capabilities as HTTP endpoints, similar to what is described in 6.2. There is no minimum or maximum of what is needed - it will depend on what problem you're solving.

### Interface

Function name	HTTP Method	Parameters	Return type	Exceptions	Description
todolists/ create	POST	(token, channel_id, name, is_public)	{ todolist_id }	<b>InputError</b> when any of: <ul style="list-style-type: none"><li>• Name is more than 20 characters long</li><li>• Channel ID is not a valid channel</li></ul> <b>AccessError</b> when: The authorised user has not joined the channel they are trying to create list to	Creates a new to-do list in channel with channel_id with that name that is either a public or private to-do list
todolists/ list	GET	(token, channel_id )	{ todolists }	<b>InputError</b> when: <ul style="list-style-type: none"><li>• Channel ID is not a valid channel</li></ul> <b>AccessError</b> when: The authorised user has not joined the channel with that channel ID	Provide a list of all to-do lists (and their associated tasks) within the channel that the authorised user can view
todolist/ edit	PUT	(token, todolist_id, name)	{ }	<b>InputError</b> when: <ul style="list-style-type: none"><li>• Todo list does not exist</li></ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"><li>• The todo list is public</li><li>• The todo list was created</li></ul>	Given a todo list, update its name with the new name. If the new name is an empty string, the todo list is deleted.

				by the authorised user making this request	
todolist/ remove	DELETE	(token, todolist_id)	{ }	<b>InputError</b> when: <ul style="list-style-type: none"> <li>• Todo list does not exist</li> </ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>• The todo list is public</li> <li>• The todo list was created by the authorised user making this request</li> </ul>	Given a todolist_id for a todo list, this todo list and its associated tasks are deleted from the channel
todolist/ tasks	GET	(token, todolist_id)	{ tasks }	<b>InputError</b> when: <ul style="list-style-type: none"> <li>• Todolist ID is not a valid todo list</li> </ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>• The todo list is public</li> <li>• The todo list was created by the authorised user making this request</li> </ul>	Given a todo list with ID todolist_id that the authorised user can view, return all the tasks within the todo list.
tasks/ create	POST	(token, todolist_id, name, description, due_date)	{ task_id }	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Name is more than 20 characters long</li> <li>• Description is more than 1000 characters</li> <li>• Todolist ID is not a valid to-do list</li> </ul>	Creates a new task in to-do list with todolist_id with that name, description and due date.

				<b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>• The todo list is public</li> <li>• The todo list was created by the authorised user making this request</li> </ul>	
tasks/ listall	GET	(token, todolist_id)	{ tasks }	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Todolist ID is not a valid todo list within a channel that the authorised user has joined</li> </ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>• The todo list is public</li> <li>• The todo list was created by the authorised user making this request</li> </ul>	Provide a list of all tasks (and their associated details) within the to-do list
task/ confirm	POST	(token, task_id)	{ }	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Task ID is not a valid task within a public todo list</li> <li>• Task with task ID is already confirmed by the authorised user</li> </ul>	Given a task within a todo list that the authorised user can view, mark it as 'confirmed' to be given special display treatment by the frontend
users/ confirmed	GET	(token, task_id)	{ users }	<b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Task ID is</li> </ul>	Return a list of all users (and their associated

				<p>not a valid task within a public todo list</p> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"> <li>The authorised user is not an owner of this channel</li> </ul>	<p>details) who has confirmed the task with ID task_id</p>
task/ remove	DELETE	(token, task_id)	{ }	<p><b>InputError</b> when:</p> <ul style="list-style-type: none"> <li>Task ID is not a valid task within a todo list that the authorised user can view</li> </ul> <p><b>AccessError</b> when none of the following are true:</p> <ul style="list-style-type: none"> <li>The todo list that the task is within is public</li> <li>The task was created by the authorised user making this request</li> </ul>	<p>Given a task_id for a task, this task is removed from the todo list</p>
task/edit/ name	PUT	(token, task_id, name)	{ }	<p><b>InputError</b> when:</p> <ul style="list-style-type: none"> <li>Task ID is not a valid task within a todo list that the authorised user can view</li> </ul> <p><b>AccessError</b> when none of the following are true:</p> <ul style="list-style-type: none"> <li>The todo list that the task is within is public</li> <li>The task</li> </ul>	<p>Update the given task's name</p>



				was created by the authorised user making this request	
task/edit/description	PUT	(token, task_id, description)	{}	<b>InputError</b> when: <ul style="list-style-type: none"> <li>Task ID is not a valid task within a todo list that the authorised user can view</li> </ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>The todo list that the task is within is public</li> <li>The task was created by the authorised user making this request</li> </ul>	Update the given task's description
task/edit/duedate	PUT	(token, task_id, due_date)	{}	<b>InputError</b> when: <ul style="list-style-type: none"> <li>Task ID is not a valid task within a todo list that the authorised user can view</li> </ul> <b>AccessError</b> when none of the following are true: <ul style="list-style-type: none"> <li>The todo list that the task is within is public</li> <li>The task was created by the authorised user</li> </ul>	Update the given task's due date
videocall/s	POST	(token,	{}	<b>InputError</b> when	Creates a video

start		channel_id )		<p>any of:</p> <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> </ul> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"> <li>The authorised user has not joined the channel they are trying to start a video call in</li> </ul>	call in a channel. If there is already a video call active, joins it instead.
videocall/join	POST	(token, channel_id )	{}	<p><b>InputError</b> when:</p> <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> <li>There is not currently a video call in the channel</li> </ul> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"> <li>The authorised user has not joined the channel they are trying to join a video call in</li> </ul>	Joins the given user to a call if one is currently in progress in the given channel
videocall/leave	DELETE	(token, channel_id )	{}	<p><b>InputError</b> when:</p> <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> <li>The user is not currently in a call</li> </ul> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"> <li>The authorised user has not joined the channel they are trying to leave a video call in</li> </ul>	Leaves the given channel's call the given user is currently in
videocall/ends	DELETE	(token,	{}	<b>InputError</b> when:	Ends the given

nd		todolist_id)		<ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> <li>The user is not currently in a call</li> </ul> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"> <li>The authorised user has not joined the channel they are trying to leave a video call in</li> <li>The user ending the call did not make the call</li> </ul>	channel's call the given user is currently in
----	--	--------------	--	--	---

## [Design] Conceptual Modelling (State)

Now that you have a sense of the problem to solve, and what capabilities you will need to provide to solve it, add at least one state diagram to your PDF to show how the state of the application would change based on user actions. The aim of this diagram is to understand the different states of the user or application.

## State Diagram

