

## ELICITATION

| Questions   | Eric<br><a href="mailto:eric.lin4@hotmail.com">eric.lin4@hotmail.com</a>   | Joseph<br><a href="mailto:jh.tran1998@gmail.com">jh.tran1998@gmail.com</a>  | Jessica<br><a href="mailto:jessica.chan5377@gmail.com">jessica.chan5377@gmail.com</a>  |
|---|--|---|--|
| What kind of apps/sites did you use to communicate with your team for group projects?           | Trello   | Trello/Flock  | Slack  |
| Did you find it easy to collaborate with your group members using these apps?                   | No, everyone convinces themselves they're going to be so organised and keep up to date but then no one uses it anymore | Most of the time, yes.  | Yeah   |
| What about a task-oriented service like Trello? Is that more effective than Flock? How?         | It depends because you have to commit to use it and get everyone to remember to use it                                 | Yes, it has all the things you do in one spot, able to move them into different categories. Can physically see teammates progress                                       |  |
| Did you find it easy to keep track of tasks that you and your group have achieved or are doing? | There were no updates until group meetings   | Yes   | Yeah   |
| Did you find it hard to coordinate with your team to carry out tasks?                           | Not really   | Not hard to coordinate with my team   | No   |
| What challenges did you face when using these kinds of apps/sites?                              | Forgets to update progress on trello/forgets to check trello   | Not an everyday thing, won't use it outside of project that often- you don't keep it in mind all the time and remembering to go back to check it is difficult sometimes | I don't check slack by itself, only if i get an email when there's been an update<br>Slack wasn't active, but whenever it was it's important |

|   |  |   |   |
|---|--|---|---|
| Do you have any suggestions that will help overcome these challenges?   | System that gives reminders easily, daily reminder to check up on trello   | Setting reminders, having a desktop app- which has notifications that remind not daily but often  | More reliable notifications that things have changed, Would like to get notifications but daily reminders would be really annoying  |
| What are your thoughts on using the tagging feature?  |  |   | Nothing special about tagging people if everyone sees it anyways. It's not team-building, no one posts anything unless it's important anyways   |
| What kind of features do you think will improve the experience of using these sort of apps for team projects? | Notifications, option to choose how often reminders<br>Reminders to check the app<br>Reminders before due dates  | By having a task system inside commonly used programs, when ur doing the project it's easier to access than using an external website/app | Separating work and fun stuffs within channels instead of having just one text channel makes it more accommodating to team building<br>Different channel types - able to categorise into professional, fun, etc.<br>Could also implement more fun features  |
| Were there any features you wished (app) had?   | Users being able to set the percentage of how much work they've done<br>It allows them to visualise how close they are to finishing a task and motivates users to keep increasing the percentage bar | Back in my day, we used screenshots! Nowadays, there's screenshare which is much more convenient  | There's no meeting system for scheduling, it was mainly sorted by text so importing a calendar that has due dates and stuff- tied to that specific channel, and being able to mark availabilities for meetings<br>Attached workspace - placed to keep shared files, to-do list that updates over time |

## **USER STORIES**

1. **SCREENSHARE (Joseph):** As a uni student using Flockr, I want screensharing so that I am able to get help or assist groupmates when dealing with problems

User Acceptance Criteria (assuming there are calls in Flockr)

- User needs to be in a call in order to screen share
- Button is placed to the left of the exit call to enter/exit screen sharing
- When clicked by user, option to share whole screen or just an application pops up
- Screenshare starts once the user selects an option
- Multiple users can screenshare at the same time

### **USER CASES**

User Case: Sharing their screen

Goal in Context: User in a call shares their screen or an application

Scope: Flockr

Level: Subfunction

Preconditions: User already has an account and is in a call within a channel

Success End Condition: User shares their screen

Failed End Condition: User is unable to share their screen

Primary Actor: Flockr User

Trigger: User joins a call and wants to screenshare

### **MAIN SUCCESS SCENARIO**

Step 1. User presses the screenshare button

Step 2. Website asks for whether they want to share their screen or application

Step 3. User selects the option to share their screen

Step 4. Successful, so website overlays user's screen on top of the messages feed

2. **REMINDERS (Eric):** As a uni student, I want constant reminders so that our group can stay on track of our tasks

User Acceptance Criteria

- Users are able to set reminders about upcoming tasks or due dates by choosing when to be reminded (eg. 1 hr before, 24 hrs before, etc)
- Users are able to set reminders to check their account then receive an email from Flockr

### **USER CASES**

User Case: Set Task Reminder

Goal in Context: User chooses a time to remind them of a task's due date

Scope: Summary

Level: Primary Task

Preconditions: User is part of the channel

Success End Condition: Flockr sends an email at the chosen time

Failed End Condition: Flockr does not send an email

Primary Actor: Flockr User

Trigger: User clicks on the set reminder button in the channel

### MAIN SUCCESS SCENARIO

- Step 1: User asks to set a reminder from Flockr
- Step 2: System asks user what type of reminder
- Step 3: User chooses the reminder for a specific task
- Step 4: System checks if they are in channel and task was assigned to them
- Step 5: System asks when they would like to be reminded
- Step 6: User sets a time to get a reminder from Flockr
- Step 7: System checks time is valid and queues an email to be sent at the specific time
- Step 8: When time arrives, email is sent

### USER CASES

User Case: Set Checkup Reminder

Goal in Context: User chooses how often they should check on Flockr

Scope: Summary

Level: Primary Task

Preconditions: User has a Flockr account

Success End Condition: Flockr sends an email at the chosen time at the chosen frequency

Failed End Condition: Flockr does not send an email

Primary Actor: Flockr User

Trigger: User clicks on the set reminder button in their profile

### MAIN SUCCESS SCENARIO

- Step 1: User asks to set a reminder from Flockr
- Step 2: System asks user what type of reminder
- Step 3: User chooses to get checkup reminders from Flockr
- Step 4: System asks user how often they would like to receive these emails
- Step 5: User selects daily reminders
- Step 6: System queues an email to be sent everyday to user
- Step 7: When time arrives, email is sent

3. **INTEGRATION WITH OTHER APPS (Joseph):** As a uni student, I want integration with more common apps such as Google Drive and Trello so that it will be more convenient for our group

#### User Acceptance Criteria

- User can access files from Google Drive or the task board from Trello by clicking a tab labelled 'APPS' near the scroll bar
- Clicking on the tab will expand to show the Google Drive and Trello logo
- Clicking on the respective apps will open them in a new tab

### USER CASES

Use Case: Access App

Goal in Context: Channel member opens an app in a new tab

Scope: Flockr

Level: Summary

Preconditions: User is a member of the channel

Success End Condition: User's chosen app is opened in a new tab

Failed End Condition: User's chosen app is not opened in a new tab

Primary Actor: Channel member

Trigger: Channel owner clicks on the 'APPS' tab

#### MAIN SUCCESS SCENARIO

Step 1: Channel member opens the 'APPS' tab

Step 2: System checks which app to open

Step 3: Channel member clicks on the Google Drive logo

Step 4: System checks if they are a member of the channel

Step 5: Successful, so Google Drive opens in a new tab

4. **CHANNEL UNIQUE CALENDAR (Jessica):** As a uni student, I want a calendar so that I can keep track of due dates for that specific project

#### User Acceptance Criteria

- A mini calendar symbol button
- When clicked, a calendar showing the current month pops up
- Due dates can be assigned and are marked
- Availabilities can be set for certain dates in order to determine meeting times
- Calendar can be marked via colour code or symbols

#### USER CASES

Use Case: Accessing a calendar

Goal in Context: Assigning and checking due dates, and setting availabilities in the calendar

Scope: Flockr

Level: Primary Task

Preconditions: User is a member of the channel

Success End Condition: User sets a due date and availabilities

Failed End Condition: User is unable to open the calendar

Primary Actor: Channel member

Trigger: User clicks on the calendar button

#### MAIN SUCCESS SCENARIO

Step 1: User clicks on the calendar icon

Step 2: System checks the user is part of the channel

Step 3: System opens the calendar in month format with due dates marked

Step 4: User selects a date

Step 5: System shows the specific date and any due dates/availabilities

Step 6: User asks to set a due date

Step 7: System asks for details such as task name, time and assigned people

Step 8: User gives due date details

Step 9: System sets the due date and marks the day on the calendar

Step 10: System returns back to the calendar in month format

Step 11: User selects another date

Step 12: System shows the specific date and any due dates/availabilities

Step 13: User asks to set their availabilities for the day

Step 14: System asks when user is available that day

Step 15: User inputs their availability that day

Step 16: System marks it into the calendar

5. **THREADS WITHIN CHANNELS (Jessica):** As a uni student, I want different message threads so that I can separate chats based on the topic of the thread

User Acceptance Criteria

- Next to the list of channels panel, there will be a panel for different threads
- To make a new thread, there will be a 'ADD' button on the top of the panel and clicking it will show a new thread and prompts the user to input the name of the new thread

USER CASES

Use Case: Create thread

Goal in Context: Channel owner creates a new thread in channel

Scope: Flockr

Level: Primary Task

Preconditions: User is an owner of the channel

Success End Condition: User creates a new thread

Failed End Condition: User does not create a new thread

Primary Actor: Channel owner

Trigger: Channel owner clicks on the 'ADD' in the threads panel

MAIN SUCCESS SCENARIO

Step 1: Channel owner asks to create a new thread

Step 2: System checks if they are a channel owner

Step 3: System asks for thread name and thread type

Step 4: Channel owner inputs details

Step 5: System creates a new thread shown on the threads panel

USER CASES

Use Case: Delete thread

Goal in Context: Channel owner deletes thread in channel

Scope: Flockr

Level: Primary Task

Preconditions: User is an owner of the channel and thread should exist

Success End Condition: User deletes thread

Failed End Condition: User does not delete thread

Primary Actor: Channel owner

Trigger: Channel owner clicks on the bin icon next to the thread's name

#### MAIN SUCCESS SCENARIO

Step 1: Channel owner asks to delete a thread

Step 2: System checks if they are a channel owner

Step 3: System asks what thread is to be deleted

Step 4: Channel owner selects a thread

Step 3: System deletes thread and removes it from the threads panel

- 6. PERCENTAGE BAR (Eric):** As a uni student, I want a task percentage bar so that my team and I can easily visualise my progress for a task

#### User Acceptance Criteria

- Users are able to see their own percentage bars for each task assigned to them
- Users can see other channel members' percentage bars
- Percentage bars go from 0 to 100% and are set by the user assigned to the task

#### USER CASES

Use Case: Task percentage bar

Goal in Context: User can set tasks as a percentage bar and set progress of tasks assigned to them

Scope: Flockr

Level: Primary Task

Preconditions: User is a member of the channel

Success End Condition: User updates their percentage bar

Failed End Condition: User is unable to create a task percentage bar

Primary Actor: Channel member

Trigger: Channel owner clicks on the bin icon next to the thread's name

#### MAIN SUCCESS SCENARIO

Step 1: User asks to create a new task

Step 2: System asks user for the name of the task and who it is assigned to

Step 3: User inputs the details

Step 4: System creates a percentage bar from 0 to 100% and displays it to the user

Step 5: User asks to adjust the progression of a task

Step 6: System checks that the user is the one assigned to the task

Step 7: System confirms and asks the user how they would like to adjust the percentage

Step 8: User updates the progress on the percentage bar

Step 9: System visually updates the percentage bar for that task

## VALIDATION

Eric: You can be more specific about the fact that you can choose days to be reminded of in user stories. For example, throw in all the details about how the feature could work and with percentage you could even tie that into reminders like (oh you've only indicated that you've done 20% of the task and you've only got half the time left! update the percentage or try to do more of this task now!). Everything else is actually super nice, I love it! It's structured really well.

Joseph: Everything seems good, it addresses my problems pretty well except for the integration of apps. I was under the impression that you could view/edit stuff from inside Flockr, it would also be good to include the option of being able to open them in another tab though.

Jessica: It covers everything, satisfies my needs for a calendar and having these options are really nice.

## INTERFACE DESIGNING

| Variable Name  | Type                                       |
|----------------|--|
| Availabilities | List of booleans for hours in a week       |
| Calendar_dict  | Dictionary of due dates and availabilities |

| Function Name               | HTTP Method | Parameters  | Return Type | Exceptions   | Description  |
|-----------------------------|-------------|---|-------------|--|--|
| channel/<br>screenshare     | PUT         | (token,<br>channel_id,<br>screenshar<br>e_option) | {}          | <b>InputError</b> when any of: <ul style="list-style-type: none"><li>Channel ID is not a valid channel</li></ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"><li>Token is invalid</li><li>User is not in channel</li><li>User is not in an active call</li></ul> | Allows the user to share their screen, either their entire screen or just an app, in the call  |
| channel/<br>set<br>reminder | PUT         | (token,<br>channel_id,<br>reminder_i<br>d)        | {}          | <b>InputError</b> when any of: <ul style="list-style-type: none"><li>Channel ID is not a valid channel</li></ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"><li>Token is invalid</li><li>User is not in channel</li></ul>                                       | Allows the user to set a reminder for an upcoming task by choosing either reminder_id:<br>1 - 1 hour before<br>2 - 1 day before<br>3 - 1 week before<br>And receives an email at the chosen time |



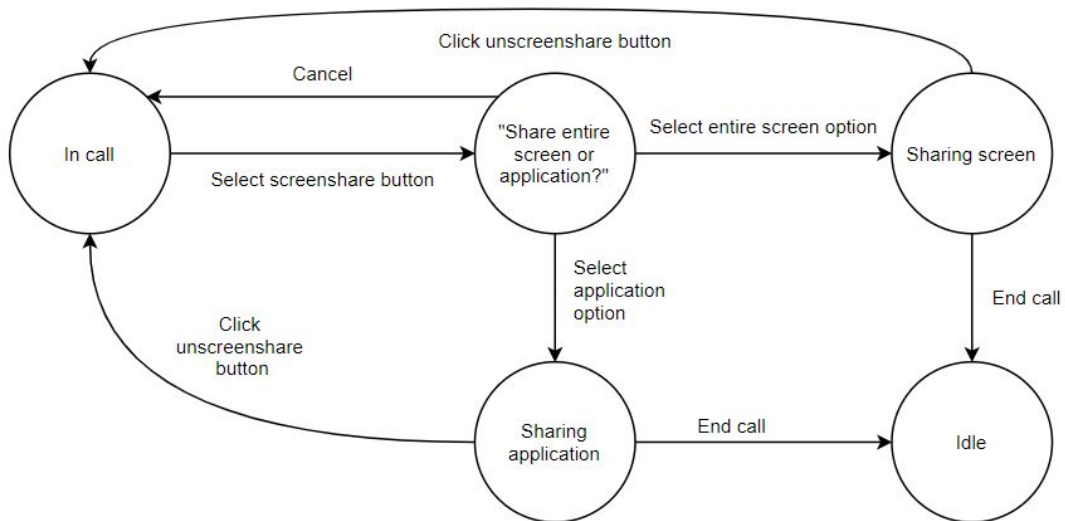
|                                    |      |  |                 |   |  |
|------------------------------------|------|--|-----------------|---|--|
| user/set<br>checkup                | PUT  | (token,<br>checkup_id<br>)   | {}              | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Checkup id is invalid</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> </ul>   | Allows the user to set a reminder to check Flockr by choosing either checkup_id: 1 - Daily<br>2 - Once every 3 days<br>3 - Weekly<br>And receives an email at the chosen times |
| channel/<br>openapp                | POST | (token,<br>channel_id,<br>app_id)  | {}              | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Channel ID is not a valid channel</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> </ul>   | Allows the user to open either Google Drive (app_id 1) or Trello (app_id 2) in a new tab   |
| channel/<br>open<br>calendar       | POST | (token,<br>channel_id)   | {calendar_dict} | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Channel ID is not a valid channel</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> </ul>   | Opens up a calendar showing the current month with due dates and availabilities marked   |
| calendar/<br>set<br>duedate        | POST | (token,<br>channel_id,<br>task_id,<br>assigned_<br>members,<br>time_duedate) | {}              | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Channel ID is not a valid channel</li> <li>• When due date is in the past</li> <li>• No assigned members</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> <li>• User setting due date is not in the channel</li> <li>• Assigned members aren't all members of the channel</li> </ul> | Assigns a task to a due date in the calendar and marks the due date. The due date also includes a list of members assigned to the task.  |
| calendar/<br>set<br>availabilities | POST | (token,<br>channel_id,<br>availabilities)                                    | {}              | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Channel ID is not a valid channel</li> <li>• When setting availabilities in the past</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> <li>• User is not in channel</li> </ul>  | Sets the user's availabilities for a week in hour blocks   |
| channel/<br>createthread           | POST | (token,<br>channel_id,<br>thread_name)                                       | {thread_id}     | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>• Channel ID is not a valid channel</li> <li>• Thread name is over 20 characters long</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>• Token is invalid</li> <li>• User is not in channel</li> </ul>   | Creates a new thread in the channel with the thread_name   |

|                           |        |  |           |   |  |
|---------------------------|--------|--|-----------|---|--|
|                           |        |  |           | <ul style="list-style-type: none"> <li>User is not an owner of channel</li> </ul>   |  |
| channel/<br>deletethread  | DELETE | (token,<br>channel_id,<br>thread_id)                 | {}        | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> <li>Thread ID does not exist</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>Token is invalid</li> <li>User is not in channel</li> <li>User is not an owner of channel</li> </ul>  | Deletes the thread from the channel  |
| channel/<br>createtask    | POST   | (token,<br>channel_id,<br>task_name)                 | {task_id} | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>Token is invalid</li> <li>User is not in channel</li> </ul>   | Creates a task for the user and automatically defaults percentage bar for it to 0% |
| channel/<br>setpercentage | PUT    | (token,<br>channel_id,<br>task_id,<br>percentage_id) | {}        | <b>InputError</b> when any of: <ul style="list-style-type: none"> <li>Channel ID is not a valid channel</li> <li>Task ID does not exist</li> <li>Percentage isn't between 0 and 100</li> </ul> <b>AccessError</b> when any of: <ul style="list-style-type: none"> <li>Token is invalid</li> <li>User is not in channel</li> <li>User is not the one who set the task</li> </ul> | Increases or decreases the percentage bar, by a given amount, of a user's task     |

## CONCEPTUAL MODELLING

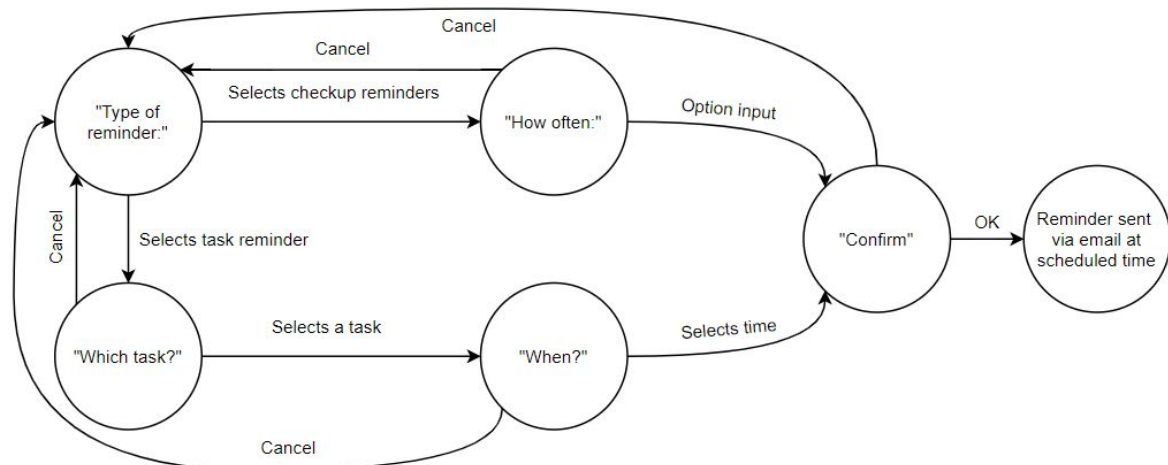
State Diagram 1: Screenshare

### Screensharing



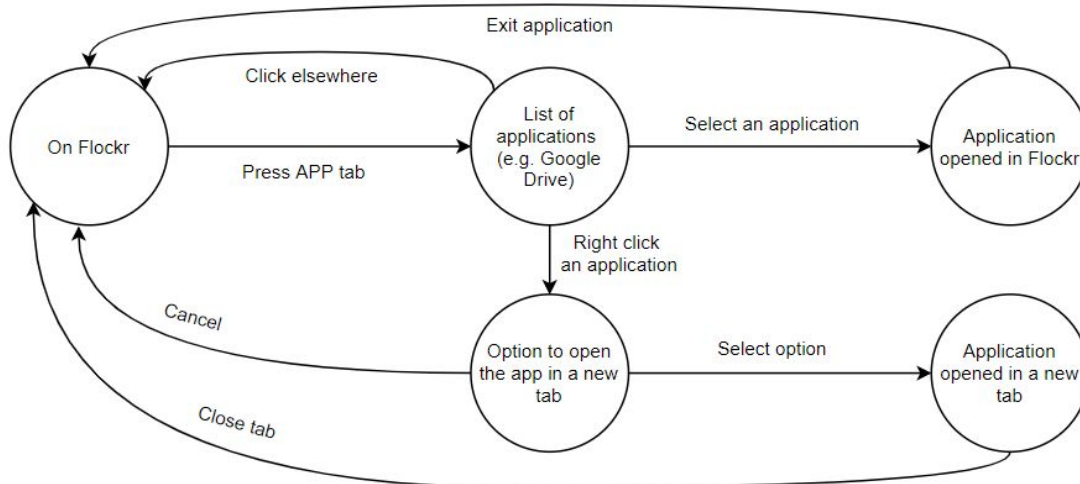
State Diagram 2: Reminders

### Reminders



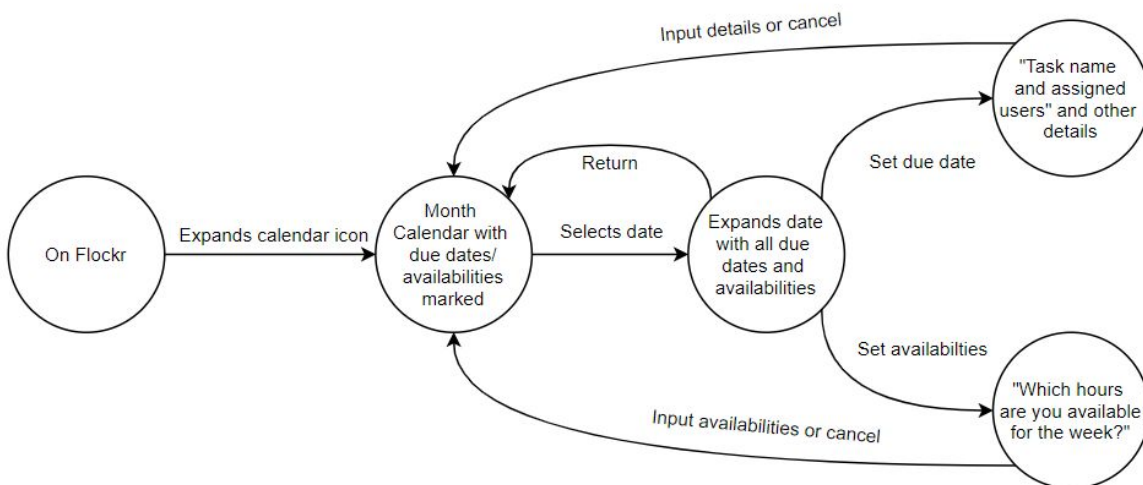
State Diagram 3: Integration with apps

## App Integration



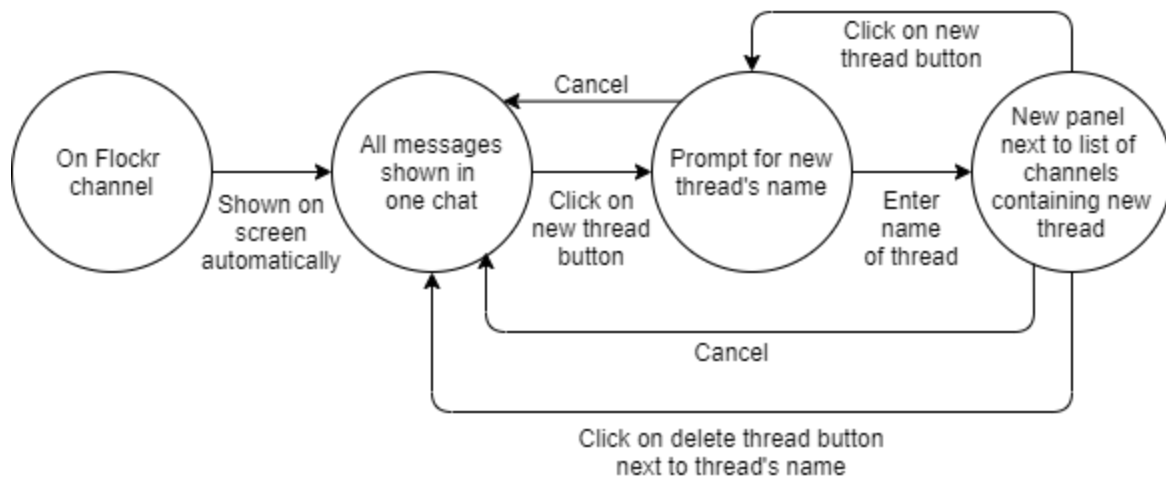
State Diagram 4: Personal Calendar

## Calendar



State Diagram 5: Thread in Channels

# Thread in Channels



State Diagram 6: Percentage Bar

# Percentage Bar

