

# REQUIREMENTS

## Interview 1

### Elicitation

Interviewee: Nathan Auckett

Email: [nauckett@gmail.com](mailto:nauckett@gmail.com)

Occupation: Game developing

**1. *What communications app do you currently use?***

Discord.

**2. *What do you use this app for?***

Communication for gaming and work related subjects

**3. *What do you like about this app?***

The call quality is high and you can screen share.

**4. *Are there any problems you have found using this app?***

I would like to be able to set deadlines and have a tool to plan out sprints for the game so me and Anton (game developing partner) can both keep track of what's happening and when.

**5. *What are some aspects that could be improved on this app and or added to solve this problem?***

Adding a to do list to the app would be a very helpful addition.

**6. *What would you use this addition for?***

I would use a to do list to keep track of deadlines as well as breaking down tasks and reminding me when I need to complete them.

# Analysis & Specification - Use Cases

## Epic - To Do List

### *User Stories*

- As a game developer I want to add deadlines for tasks on my to do list in order to create sprints for myself.
- As a busy person I want my to do list to send me notifications so I can remember what I am supposed to be doing and keep on task.
- As a programmer most of my tasks involve multiple aspects and I would like to be able to divide tasks up into smaller parts.

### *User acceptance Criteria*

- When you are on a channel there is a bar on the right with a to do list button, when clicked it comes up with all the current to do lists and a + button in the corner to add a to do list.
- There is an add button on the to do list as well to add a task, When you create a task it has a name and completed by, field.
- When you click the completed field it drops down a calendar and once you select a date it drops down a list from 00:00 - 24:00 to specify time.
- When you click on a task in a to do list it opens a new page with the tasks information
- On the task page it had a reminders list on the left hand and an add reminder button at the bottom.
- when you click on add reminder a pop up box comes up and had two fields (name and time)
- The name is entered in the box next to the name field and the time field works the same as the completed by field from the criteria above.
- On the task page it has a header with the main task with bullet points of the smaller tasks underneath
- The smaller tasks have little check boxes next to them and when clicked have a tick next to them.
- There is a button at the bottom labeled (add sub-task) which when clicked will come up with the task fields and then enter to add the task.
- The to do list, tasks and subtasks all have a bin icon next when clicked it asks if you are sure you want to delete, select yes to continue or no to cancel.

### *Use Case - Add reminder for task*

- Step 1. User adds to do list
- Step 2. App asks for to do list information
- Step 3. User enters information
- Step 4. User adds a task
- Step 5. App asks for task information
- Step 6. User enters task information
- Step 7. User clicks on task
- Step 8. App Displays page with tasks information
- Step 9. User clicks add reminder
- Step 10. App asks for reminder name and time
- Step 11. User enters name and time for reminder
- Step 12: App adds reminder to reminder list
- Step 13: App send notification to user when time entered arrives

### **Validation**

#### *Interviewee Comment*

The aspects of the to do list look good, however I would also like a weekly performance review so i can see what tasks I did on time and what tasks I didn't.

# DESIGN

## Interface Design - To Do List

Added data types

Variable name	Type
list_id	int
list_name	string
task_id	int
task_name	string
completed_by	timestamp
lists	A list of dictionaries
list	A dictionary containing { list_id, list_name, tasks}
tasks	A list of task dictionaries
task	A dictionary containing {task_id, task_name, completed_by, sub-tasks}
reminder_name	string
reminder_time	timestamp
sub_task_name	string
sub_tasks	A list of dictionaries containing {task_id, task_name, completed_by}
is_completed	Boolean

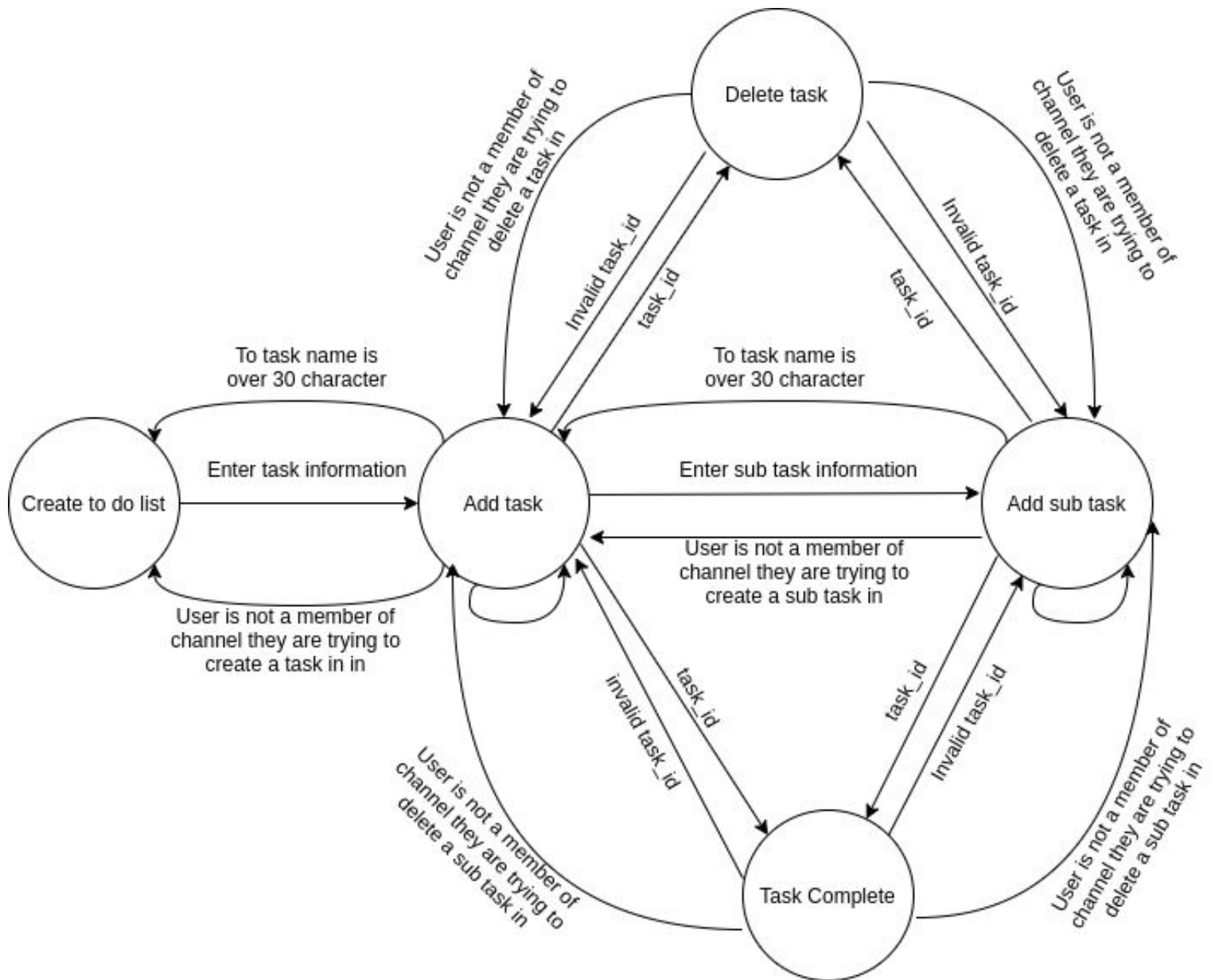
## Endpoints

Function name	HTTP Method	Parameters	Return type	Exceptions	Description
create/todolist	POST	(token, channel_id, list_name)	{list_id}	<b>InputError</b> when: -Name is more than 30 characters long  <b>AccessError</b> when: the authorised user has not joined the channel they are trying to create a to do list in	Creates a new to do list with the name given for the channel specified with the channel_id
add/task	POST	(token, list_id, task_name, completed_by)	{task_id}	<b>InputError</b> when: -Name is more than 30 characters long -When they enter a time that has already passed -list_id does not refer to a valid list  <b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit to do list in	Adds a task to the to do list with given list_id
task/reminder	POST	(token, task_id, reminder_name, reminder_time)	{}	<b>InputError</b> when: -Name is more than 30 characters long -When they enter a time that has already passed -task_id does not refer to a valid task  <b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit to a task in.	Sets a reminder time for task with given task_id and sends notification to user when reminder_time arrives
tasks/list	GET	(token, list_id )	{tasks}	<b>InputError</b> when: -list_id does not refer to a valid task  <b>AccessError</b> when: the authorised user has not joined the channel they are trying view the to do list tasks in	Lists all the tasks in a to do list
todolist/list	GET	(token, channel_id)	{lists}	<b>InputError</b> when: -channel_id does not refer to a valid task  <b>AccessError</b> when:	Lists all the to do lists in a channel

				the authorised user has not joined the channel they are trying view the to do list in	
task/info	GET	(token, task_id)	{}	<b>InputError</b> when: -task_id does not refer to a valid task	For a valid task return the information about the task_id, name, completed by, reminders, sub_tasks
task/subtask	POST	(token, task_id, sub_task_name, completed_by)	{task_id}	<b>InputError</b> when: -task_id does not refer to a valid task -Name is more than 30 characters long -When they enter a time that has already passed  <b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit a task in	Adds a sub task to a task
task/setname	PUT	(token, task_id, task_name)	{}	<b>InputError</b> when: -task_id does not refer to a valid task -Name is more than 30 characters long	Update the name of a task
task/settime	PUT	(token, task_id, completed by)	{}	<b>InputError</b> when: -task_id does not refer to a valid task -When they enter a time that has already passed	Update the time task needs to be completed by
todolist/setname	PUT	(token, list_id, list_name)	{}	<b>InputError</b> when: -list_id does not refer to a valid task -Name is more than 30 characters long	Update to do list name
task/delete	DELETE	(token, task_id)	{}	<b>InputError</b> when: -task based on task_id no longer exists  <b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit a task in	Given task_id for task, this task is deleted from the to do list.

task/complete	POST	(token, task_id)	{}	<p><b>InputError</b> when: -task bases on task_id no longer exists</p> <p><b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit a task in</p>	Given a task within a to do list mark it as 'completed' to be given special display treatment byu the frontend.
todolist/delete	DELETE	(token, list_id)	{}	<p><b>InputError</b> when: -list based on list_id no longer exists</p> <p><b>AccessError</b> when: the authorised user has not joined the channel they are trying to edit a list in</p>	Given list_id for a to do list, this list is deleted from the channel.
subtasks/list	POST	(token, task_id)	{sub_tasks}	<p><b>InputError</b> when: -task_id does not refer to a valid task</p> <p><b>AccessError</b> when: -the authorised user has not joined the channel they are trying view the sub tasks in</p>	Provides a list of all the sub_tasks in a task with given task_id
task/performance	POST	(token, channel_id)	{task_percentage}	<p><b>InputError</b> when: -channel_id does not refer to a valid channel</p> <p><b>AccessError</b> when: -the authorised user has not joined the channel they are trying view performance in</p>	<p>Provides a percentage of tasks completed in a channel. e.g</p> <p>Tasks completed</p> <hr/> <p>All tasks in all the to so lists</p>

## Conceptual Modelling (State)





# REQUIREMENTS

## Interview 2

### Elicitation

Interviewee: Samuel Pott

Email: sampott1[@gmail.com](mailto:sampott1@gmail.com)

Occupation: Full time student

- 1. *What communications app do you currently use?***
  - a. Messenger.
- 2. *What do you use this app for?***
  - a. Communication with peers for study purposes
- 3. *What do you like about this app?***
  - a. Accessible as almost everyone has a Facebook account.
- 4. *Are there any problems you have found using this app?***
  - a. I would like to be able to create milestones so I can track the the progress of our projects on a larger scale as opposed to that of a smaller scale using a task board
- 5. *What are some aspects that could be improved on this app and or added to solve this problem?***
  - a. Adding a milestone functionality which details everything that needs to be done before a milestone can be reached would be very helpful
- 6. *What would you use this addition for?***
  - a. I would use this addition in keeping track of my overall progress in a project

## **Analysis & Specification - Use Cases**

### **Epic - To Do List**

#### *User Stories*

- As a user I want to be able to create milestones for my project so I can track my overall progress within the project.
- As a user I want to be able to create milestones so I can break down a big project into smaller sections
- As a user I want to be able to create a estimated time of completion for each milestone so that I can plan my work load accordingly

#### **User acceptance Criteria**

- When you click on a milestone it lists all the tasks that need to be completed before the milestone can be completed
- Tasks within a milestone are labelled as completed or uncompleted
- Tasks can be added or removed from a milestone

#### *Use Case - Create milestone*

Step 1. User creates milestone

Step 2. App asks for tasks which need to be completed for the milestone to be reached

Step 3. User enters task information

Step 4. App asks for a estimated completion time

Step 5. User adds an estimated completion time

## **Validation**

#### *Interviewee Comment*

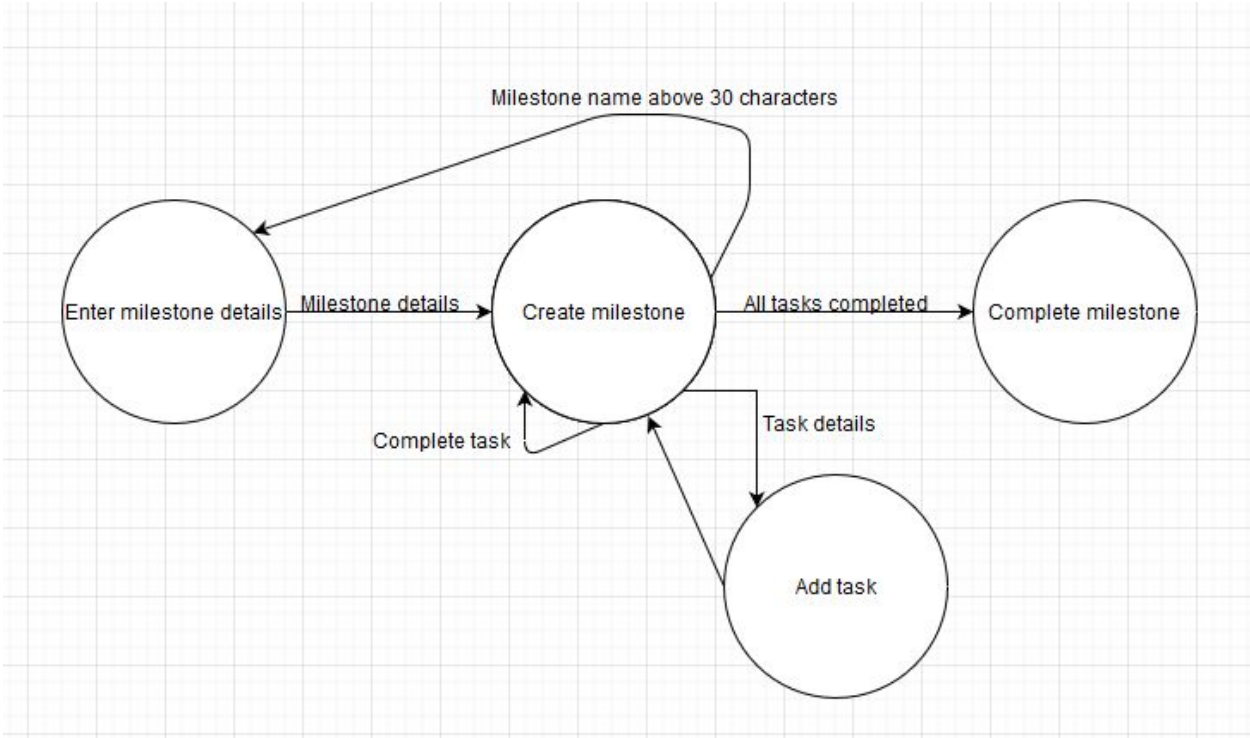
Yes this is what I want, however I would also like it if the milestones could be color coded so that I can easily tell when a due date is upcoming very soon without having to read the details.

## Endpoints

Function name	HTTP Method	Parameters	Return type	Exceptions	Description
milestone/create	POST	(token, tasks, milestone_name, etc, completed)	{}	<b>InputError</b> when: -milestone_name above 30 characters -etc is invalid epoch time	Creates a milestone using milestone_name with a list of tasks that need to be completed.  Also contains an estimated time of completion
milestone/complete	POST	(token, milestone_id)	{}	<b>InputError</b> when: -milestone_id does not refer to valid milestone	Sets the 'completed' field of a milestone to be true
milestone/taskadd	POST	(token, milestone_id, task_name)	{}	<b>InputError</b> when: -milestone_id does not refer to valid milestone -task_name above 30 characters	Adds a new task to the milestone with name task_name
milestone/taskcomplete	POST	(token, milestone_id,	{}	<b>InputError</b> when: -milestone_id does not refer to	Completes a task that is part of

		task_id)		valid milestone -task_id does not refer to task within the milestone	milestone with milestone_id
milestone/all	GET	(token)	{milestones}		Returns a dictionary of all milestones

Conceptual Modelling (State)



# REQUIREMENTS

## Interview 3

### Elicitation

Interviewee: Patrick Nguyen

Email: [rubikscuber99877@gmail.com](mailto:rubikscuber99877@gmail.com)

Occupation: Full time student

***What communications app do you currently use?***

Messenger.

***What do you use this app for?***

I use this app generally to message and sometimes use it to call my friends too.

***What do you like about this app?***

It is very convenient as it is linked to facebook, and also my phone number, so that I can reach the people I know very easily.

***Are there any problems you have found using this app?***

The biggest issue I have with the app is displaying my status. Messenger only shows online and offline, and this depends on if I have the app open or not. I want to be able to display other statuses such as a do not disturb, or an away option.

***What are some aspects that could be improved on this app and or added to solve this problem?***

Being able to display my status to my friends on messenger would be a feature that I would like to be implemented.

## **Analysis & Specification - Edit Status**

### **Epic - Edit Status**

#### ***User Stories***

- As a user I want to be able to edit my status to let my colleagues/peers/friends/family know if I am online or away from the app.

#### ***User Acceptance Criteria***

- When you go to user settings, there is an edit status option
- When you click on edit status, a smaller window pops up that presents the status options
- When you choose a status, you click update and exit the window
- The display status changes
- This display status can be updated as many times as needed

#### ***Use Case - Editing the user's status***

Step 1. User goes to user settings

Step 2. User goes to edit status

Step 3. App displays a window that asks the user to click on one of the given status settings

Step 4. User chooses a status.

Step 5. App updates the user's status

## Validation

### *Interviewee Comment*

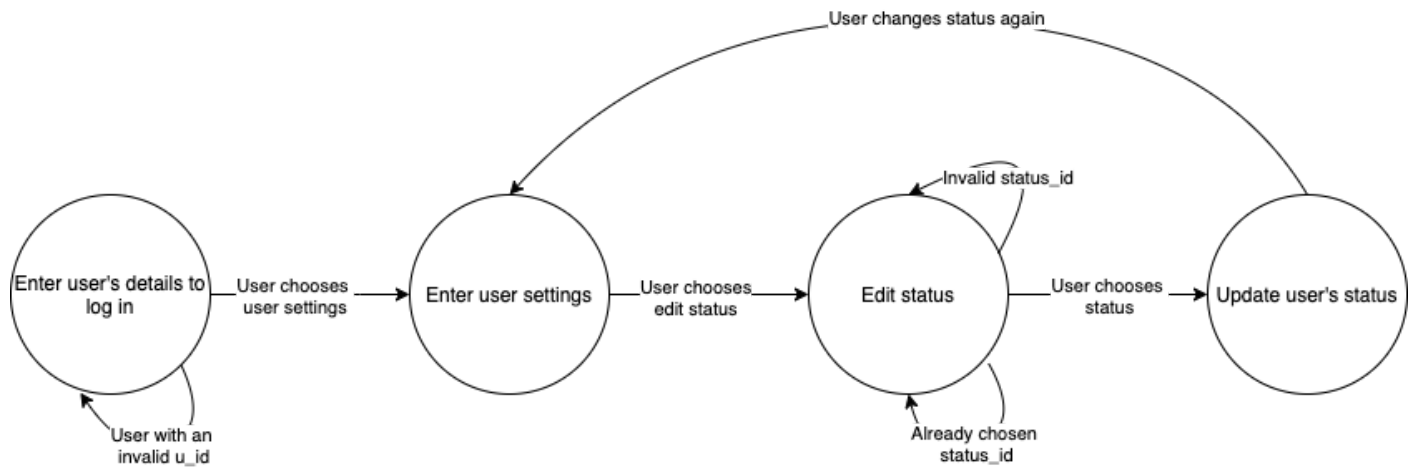
The options that I have to edit my status seem great. However, I would also like to be able to add in my own specific customised status messages as the ones given by the app are a bit limited.

## Interface Design

### Endpoints

Function name	HTTP Method	Parameters	Return type	Exceptions	Description
user/profile/status	POST	{token, u_id, status_id}	{}	<b>InputError</b> when any of: -user with u_id is not a valid user -status_id is not a valid Status ID. -user with u_id has already chosen the same active status ID	Updates the authorised user's status

## Conceptual Modelling (State)





# Requirements

## Elicitation

Interviewee: Jack Smith

Email: jacksmith97605@gmail.com

Occupation: University Student

### ***What communications app do you currently use?***

Flock

### ***What do you use this app for?***

Communication during group projects, as well as communicating with others in a course chats

### ***What do you like about this app?***

The interface is pretty intuitive, and it has a decent amount of useful features for organisation such as a to do list, and the ability to both voice and video call.

### ***Are there any problems you have found using this app?***

You can't block other users like you can in most other messaging apps like Facebook. There have been occasions in the class chat for certain courses where I really would have liked to block particular users from messaging me, but found myself unable to, and having to put up with them.

### ***What are some aspects that could be improved on this app and or added to solve this problem?***

A feature that prevents certain users from messaging me could be implemented to fix this.

# Analysis & Specification - Blocking

## Epic - Blocking

### *User Stories*

- As a user I want to be able to prevent certain users from messaging me at any time.

### *User Acceptance Criteria*

- When you click on a user's profile/name, there will be an option that appears which says 'Block'.
- When you click on the 'Block' option, the other user will no longer be able to select you as a target for a message.
- The other user's name will be greyed out to you, to show that you have blocked them.
- You can click on a blocked user's profile/name and the 'Block' option will have been replaced by an 'Unblock' option.
- Clicking on the 'Unblock' option will ungrey the user's name, and they will now be able to select you as a target for messages.

### *Use Case - Blocking another user*

Step 1. User goes to target user's profile/name

Step 2. App displays option to block/unblock target user depending on if target user is already blocked

Step 3. User chooses to block/unblock target user

Step 4. App updates the target user's range of recipients for messages to either include or exclude the user who has blocked/unblocked them

Step 5. App updates user's vision of target user's name to be grey for 'blocked' or not grey for 'unblocked'

## Validation

### *Interviewee Comment*

The option to block and unblock users like this seems to be pretty good. Although, I feel it might be a bit more convenient to be able to block more than one person at a time.

## Interface Design

### Endpoints

Function name	HTTP Method	Parameters	Return type	Exceptions	Description
users/block	POST	(token, u_ids)	{}	<b>InputError</b> when any of: -user with u_id is not a valid user -u_id is already among list of people blocked	Updates a list of people the user has blocked, preventing them from being a recipient of those users' messages.
users/unblock	POST	(token, u_ids)	{}	<b>InputError</b> when any of: -user with u_id is not a valid user -u_id is not among list of people blocked	Removes target users from list of user's blocked people, letting them message the user again.

## Conceptual Modelling(State Diagram)

