

CEIC1531 - Flockr

By Joshua Lee, Andreea Vissarion, Kelly Zhou, Cyrus Chow, Nick Dodd

[Requirements] Elicitation

The following are questions posed to target users, Ryan Lee (ryanleee@hotmail.com) and Mimi Lam (mimilam03@yahoo.com.au), with their answers recorded.

Are you satisfied with channels and their functionality and how permissions work with owners and regular members?

Ryan: I think they are fine with how they are. I don't really use channels for anything else.

Mimi: They are ok. Having just two roles as owner and regular members is enough without being overly complicated.

Do you feel you are able to share information easily to other users using Flockr?

Ryan: Yes, although the messaging is the bare minimum. I am happy with how it is easy to send them, however would like the option to also send photos in chat. I appreciate the ability to upload profile pictures for clear communication though.

Mimi: Yes all my messages can be separated into different channels which is useful for talking to different groups.

Are there any tools that Flockr could have that it does not at the moment that can help with productivity and teamwork?

Ryan: I think there is no calling feature. A lot of similar apps have calling features so it would be good to have voice communication.

Mimi: I agree with Ryan. To be able to call is good to be included because sometimes messaging is not as good as talking to get the message across properly.

Do you think individual calls are enough or should group calls also be implemented?

Ryan: I definitely think group calls are needed because it should be just as good as channels in talking to many people at once

Apart from calls, is there anything else that would be helpful for communication to improve Flockr?

Ryan: maybe adding video calls would be helpful to make calls more engaging

Mimi: Not just calls but to also have mute button in call would be useful so that everyone can listen to one person talk at a time.

[Requirements] Analysis & Specification - Use Cases

The following are user stories with corresponding acceptance criteria. Three use cases are then completed.

As a student I want to have voice calls because it is faster and easier to communicate information.

Acceptance Criteria:

- A voice call button is available next to each member of the Flockr
- Call invite for all involved users when a call is started
- Call invites for users disappear after accepting.
- All users engaged in call cannot have another call at the same time
- Group calls can be held as well but the button is next to each channel
- Call ends when all members leave call
- Any person who has left call can still rejoin call

As a student, I would like calls to be able to have video so that people are more engaged in meetings

Acceptance Criteria:

- During call, there is a video enable button so a user can use their video camera
- It can be turned off by pressing the button again

As a instructor, I would like users in call to also mute themselves so that people can hear one person speak at a time

Acceptance Criteria:

- During call, there is a mute button so a user can mute their voice so they do not make sound
- It can be disabled by pressing the button again

As a student, I would like to be able to send photos in the chat to allow a greater depth of idea sharing.

Acceptance Criteria:

- On the channel page there is a “attach image” button so a user can send an image as a message.
- The picture can be enlarged by clicking on the image after being sent.

USE CASES:

Case 1)

Use Case: User logs on and is able to start a video call in a channel they are part of.

Goal in Context: User is able to start, and end a video call.

Scope: Flockr

Preconditions: The user has an account with the flockr and a channel they are part of. There are multiple registered users in this channel.

Success End Condition: The user has started a video call which they are also able to end.

Failed End Condition: The user is unable to start or end a call.

Primary Actor: User

Trigger: User logs on.

MAIN SUCCESS SCENARIO:

Step 1: System asks user to log in.

Step 2: User enters email and password.

Step 3: System shows Flockr home page.

Step 4: User clicks on channel they are part of.

Step 5: System shows channel page, which includes members and messages.

Step 6: User prompts the system for a group call to start.

Step 7: System invites all members into a call.

Step 8: Users accept the invite.

Step 9: System adds the members into the call.

Step 10: System displays members currently in call.

Step 11: A user presses the enable camera button.

Step 12: System enables video for that user.

Step 13: The call starter presses end call.

Step 14: All users are removed from the call and call ceases.

Case 2)

Use Case: Users log on, go to channel, and join a video call, mute themselves, and leave call.

Goal in Context: Users are able to join a video call, mute themselves, and leave call.

Scope: Flockr

Preconditions: The users have an account with the flockr and are part of the same channel.

Success End Condition: The users have all joined the call and have been able to mute people not currently talking, as well as leave.

Failed End Condition: The user is unable to mute or leave a call.

Primary Actor: User

Trigger: User logs on.

MAIN SUCCESS SCENARIO:

Step 1: System asks user to log in.

Step 2: User enters email and password.
Step 3: System shows Flockr home page.
Step 4: User clicks on channel they are part of.
Step 5: System shows channel page, which includes members and messages.
Step 6: User prompts the system for a group call to start.
Step 7: Users join call.
Step 8: System adds the members into the call.
Step 9: System displays members currently in call.
Step 10: Users press the enable camera button.
Step 11: System enables video for that user.
Step 12: Users press mute button except for the call starter.
Step 13: System mutes all users except for call starter.
Step 14: Users unmute themselves and call starter mutes themselves.
Step 15: System mutes call starter.
Step 16: Users press leave call.
Step 17: System removes them from call.
Step 18: The last person leaves the call.
Step 19: All users are removed from the call and call ceases.

Case 3)

Use Case: User logs on, and sends a photo message to a channel they are a part of.

Goal in Context: User is able to send a photo message.

Scope: Flockr

Preconditions: The user has an account with the flockr.

Success End Condition: The user has sent a photo.

Failed End Condition: The user is unable to send a photo.

Primary Actor: User

Trigger: User logs on.

MAIN SUCCESS SCENARIO:

Step 1: System asks user to log in.
Step 2: User enters email and password.
Step 3: System shows Flockr home page.
Step 4: User clicks on channel they are part of.
Step 5: System shows channel page, which includes members and messages.
Step 6: User clicks attach image to message and clicks send.
Step 7: System updates messages and displays image.
Step 8: User clicks on photo.
Step 9: System enlarges image.

[Requirements] Validation

Enquiring Ryan and Mimi as to whether the use cases adequately describe the problem.

Here is a use case and a summary of the requirements I have learned from the last time I asked you questions. Do these help answer those needs?

Ryan: It looks like it covers all the needs and tools I would have liked to see in Flockr. I'm very happy with the use cases. It seems you added the mute button which is something Mimi wanted.

Mimi: I am also very happy with the acceptance criteria and the things that are proposed to be implemented.

[Design] Interface Design

The solution:

Function Name	HTTP Method	Parameters	Return Type	Exceptions	Description
call/create	POST	(token)	{call_id}	Access Error: Token passed in is not a valid token.	Starts either a group or private call. Adds a call_id to a list of call_ids which are dictionaries that hold the channel the call is started in, and member dictionaries containing fields 'mute' and 'video_on'.
call/invite	POST	(token, is_group, u_id, call_id)	{}	Input Error: is_group is not valid u_id is not valid call_id does not exist Access	If is_group is true, all members from the channel the call has been started in are invited. If is_group is false, only u_id is invited

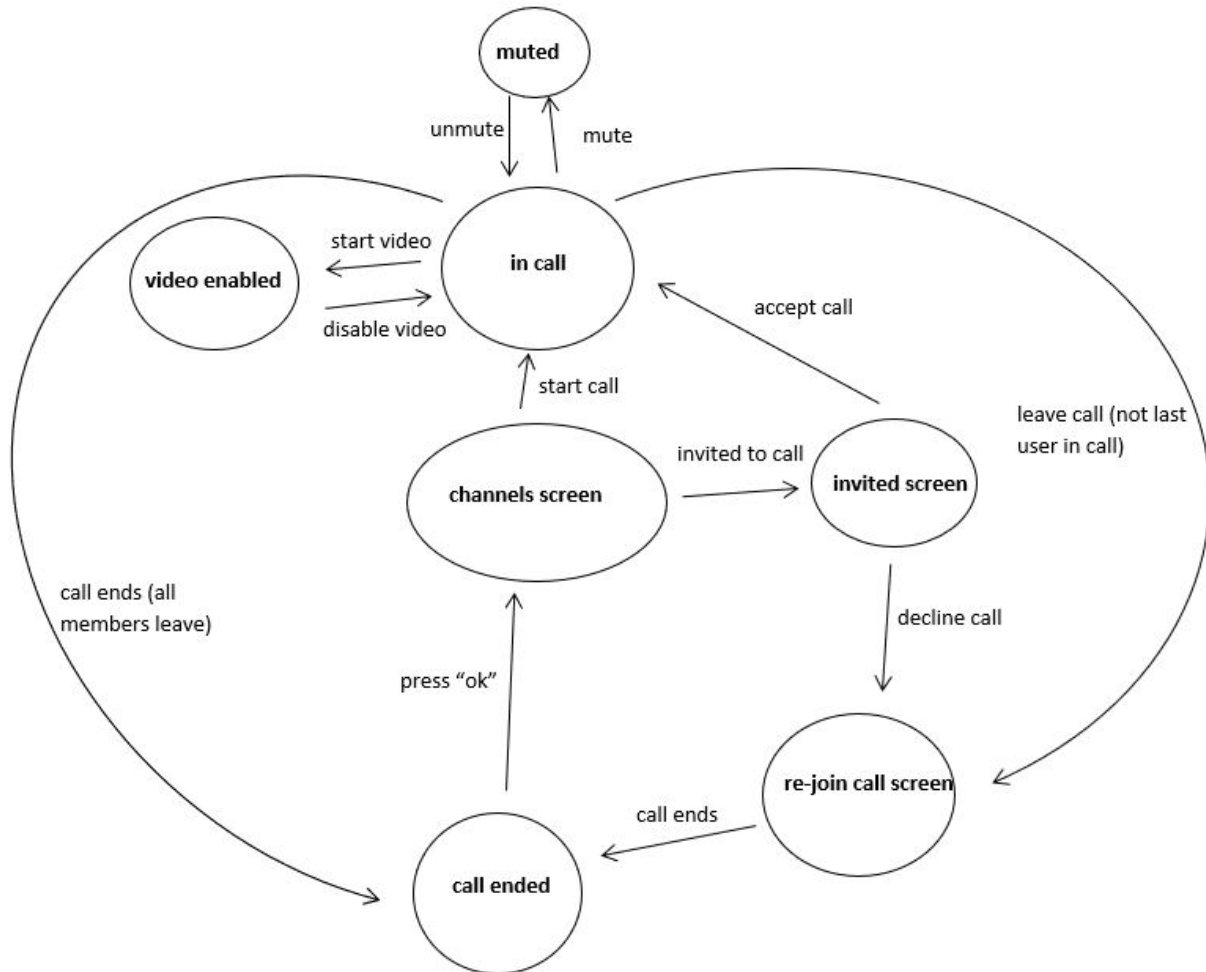
				Error: Token passed in is not a valid token.	to call. In the former case, the u_id passed in can be the u_id of the authorised user.
call/mute	POST	(token, call_id)	{}	Input Error: call_id does not exist Access Error: Authorised user is not in call Token passed in is not a valid token.	In call_id dictionary, mutes the user.
call/unmute	POST	(token, call_id)	{}	Input Error: call_id does not exist Access Error: Authorised user is not in call Token passed in is not a valid token.	In call_id dictionary, unmutes the user.
call/video	POST	(token, call_id)	{}	Input Error: call_id does not exist Access Error: Authorised user is not in call Token passed in is not a valid token.	In call_id dictionary, enables video for the user.

call/unvideo	POST	(token, call_id)	{}	<p>Input Error: call_id does not exist</p> <p>Access Error: Authorised user is not in call</p> <p>Token passed in is not a valid token.</p>	In call_id dictionary, disables video for the user.
call/leave	POST	(token, call_id)	{}	<p>Input Error: call_id does not exist</p> <p>Access Error: Authorised user is not in call</p> <p>Token passed in is not a valid token.</p>	Removes a user from the call with specified call_id. Call ceases once last person in call leaves.
call/join	POST	(token, call_id)	{}	<p>Input Error: call_id does not exist</p> <p>Access Error: Authorised user is not in call</p> <p>Token passed in is not a valid token.</p>	Adds a member to the call with specified call_id, however if user is already in a different call, do not allow join.
call/end	POST	(token, call_id)	{}	<p>Input Error: call_id does not exist</p> <p>Access Error: Authorised user is not the</p>	Ends the call

				call starter Token passed in is not a valid token.	
call/getmembers	GET	(token,call_id)	{members}	Input Error: call_id does not exist Access Error: Authorised user is not in call Token passed in is not a valid token.	Returns the members currently in the call
message/sendphoto	POST	(token,img_url)	{msg_id}	Input Error: img_url doesn't exist Access Error: Token passed in is not a valid token.	Sends an image as a message.
message/enlargephoto	GET	(token,msg_id)	{image}	Input Error: msg_id doesn't exist msg_id does not correspond to a message that is an image. Access Error: Token passed in is not a valid token.	Makes specified image take up a larger amount of the screen.

[Design] Conceptual Modelling (State)

The solution for video feature visualised with a diagram:



The solution for image sending feature visualised with a diagram:

