Planning: fri15 mango group 3

1. User's name: Xinglei Zhang Email: z5237609@unsw.edu.au

User's story:

As a student in group meeting, I want to reply a certain message in the channel so that other students in the meeting channel will not confused about who this message is send to.

User Acceptance Criteria:

- Scenario: Want to reply a certain message.
- Given: user has attended in a channel.
- When: user press the reply button on the certain message.
- And: user enter the message that he or she want reply.
- Then: system will send a message to the channel by the user's name.
- And: the message should be the combination of the original message and new message

User case: reply certain message.
Goal in Context: users need to reply.
Scope: flock.
Level: Primary Task.
Preconditions: The user has an account with flock, login and attend a channel.
Success End Condition: The message user to reply is in the channel, and the message user want to send is valid.
Failed End Condition: The message has been deleted or not in this channel, or the message user want to send is too long
Primary Actor: user.
Trigger: user press the reply button on the message.

Validation:

This user case fully described the feature I would like to have in the Flock application. The two InputError and the AccessError are well-considered. However, I would like to add "channel id is not invalid channel id" as the third InputError as well as slightly change the definition of Message in the first InputError. It should be cleared that this message is the combination of the original message and new message.

2.. User's name: Zhao Huang Email: z52865206@ad.unsw.edu.au

User's story:

As a student, I want to submit the materials I used in the group discussions that my team members will have the permission to view these files.

User Acceptance Criteria:

- Scenario: Want to submit materials for group meeting.
- Given: The submit file button is on the left side of "submit message" button.
- when: Submit file starts once the user clicks "Submit File".
- And: User need to pull the files that he or she wants to the window inside the flock.
- Then: Submitting File is performed if a user pulls a valid file
- And: The Flock's window will show the size of the size.
- Then: By clicking the submitting button again to make sure you want to submit all these files you pull.
- And: Other people inside this channel can see this file but they cannot edit it.

Use Case: Submitting the file into the belonged channel.
Goal in Context: User need to submit the files in the channel he or she belongs.
Scope: user's account, (eg. registering user, change the password, etc.), channel
Level: Primary Task
Preconditions: The user is already in the channel he wants to submit the files
Success End Condition: The user submits the files he wants in the channel.
Failed End Condition: The user does not submit the files he wants in the channel.
Primary Actor: Flocker user
Trigger: User clicks on the "Submitting files" button.

Validation:

Q: Based on the conversation we had, I made the use case and UAC. Do you think it is enough to satisfy the needs you want? Or do you have anu suggestions to give to make this functionality great?
A: Maybe I want to give a suggestion that when I have submitted the files, the members have the permission to give comments on these files.

UAC: after validation

- Scenario: Want to submit materials for group meeting.
- Given: The submit file button is on the left side of “submit message” button.
- when: Submit file starts once the user clicks “Submit File”.
- And: User need to pull the files that he or she wants to the window inside the flock.
- Then: Submitting File is performed if a user pulls a valid file
- And: The Flock’s window will show the size of the size.
- Then: By clicking the submitting button again to make sure you want to submit all these files you pull.
- And: Other people inside this channel can see this file but they cannot edit it.
- And: Member inside the channel can make the comment about these files.
- Then: On the right side of these files, it will have “Comment” button.
- When: By clicking “Comment” button, the file will show to the user.
- When: By choose specific line of the file, user can add comment to this line.
- And: Everyone inside this channel has the permission to comment on the files.
- Then: After commenting, the system will update the comment file on this channel, which is on the top of the channel window called “Comment file on XXX files”

Use Case: Submitting the file into the belonged channel and other member have the permission to make comments on this file.
Goal in Context: User need to submit the files in the channel he or she belongs.
Scope: user’s account, (eg. registering user, change the password, etc.), channel.
Level: Primary Task.
Preconditions: The user is already in the channel he wants to submit the files.
Success End Condition: The user submits the files he wants in the channel and other people can comment on these.
Failed End Condition: The user does not submit the files he wants in the channel or other people cannot make comments on these files.
Primary Actor: Flocker user.
Trigger: User clicks on the “Submitting files” button.

3. User's name: Bennett Tang Email: 3378634@student.uts.edu.au

User's story:

As a gamer, I will always use a communication tools to start a voice chat with friends. I believe that having the feature of voice chat in a teamwork-driven communication tools is very important, since in a voice chat it is very easy to say what you want to say, while typing sometime you had to type a long sentence is very inconvenience.

User Acceptance Criteria:

- Every member can start a voice chat.
- Every member can join the voice chat room.
- Start the voice chat by member clicking on the phone logo.
- Other user can join the voice chat by clicking on green bar at topside.
- User can see which user is in the voice chat and how many users is inside.
- When there is no one in the voice chat, then the voice chat room will be deleted automatically.
- User will still be able to send message in the channel even they join the voice chat.

Use Case: Start a voice chat
Goal in Context: A member in the channel can start a voice chat and every other member can join in to discuss.
Scope: Channel, voice chat room.
Level: Primary Task.
Preconditions: The user is the member of the channel
Success End Condition: The member created a voice chat room; all other user can join in if they want.
Failed End Condition: The user fail to start a voice chat because he is not a member in the channel.
Primary Actor: Flocker's user.
Trigger: The member in the channel clicks on the phone logo.

Validation:

Q. This is the use case and UAC I developed from your replied, is this all the feature you want in the voice chat, and any extra feature I did not cover up?
A. Yes this is nearly all the feature I want for the voice chat room, there's one extra feature I want to add is the owner can mute the member in the voice chat room, so they can't talk. But owner cannot mute another owner.

UAC: after validation

- Owner can mute any member.
- Owner mutes a user by right clicking their name, and then click mute. Same as unmutes.
- Owners are not allowed to mute another channel owner and Flockr owner.
- When user is muted, they cannot speak, but they can still send message, or listen to another user.
- If a user is muted, they should be muted by default if they rejoin the voice chat room. Unless the voice chat room is recreated.

Extra use case for new subfunction feature:

Use Case: Mute a member in the voice chat room.

Goal in Context: An owner in the channel can mute the member in the voice chat room, so they cannot speak.

Scope: Channel, voice chat room.

Level: Subfunction.

Preconditions: The user is the owner of the channel, the owner can only mute member, but they cannot mute another owner.

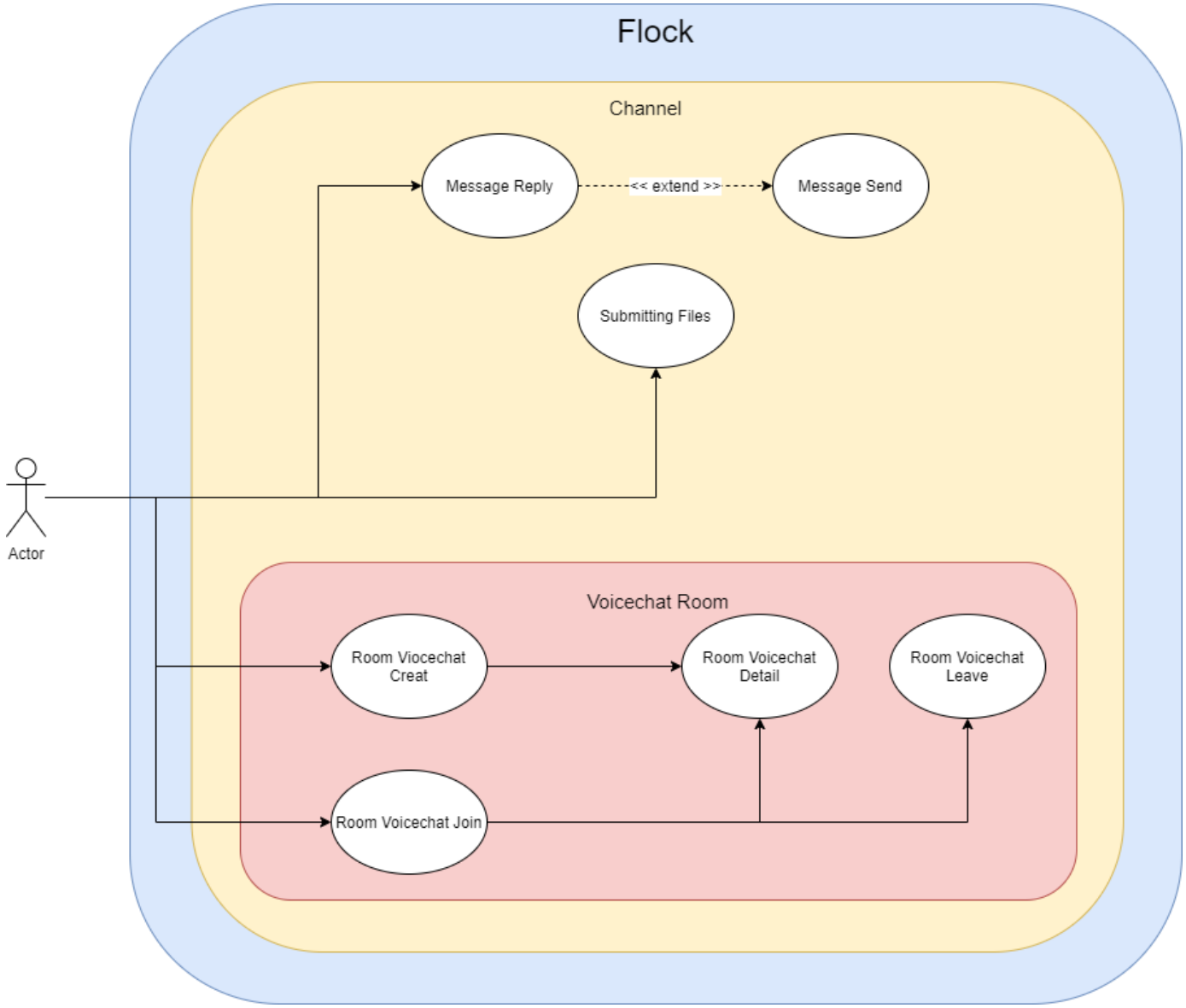
Success End Condition: The member is muted by the owner.

Failed End Condition: The user failed to mute another user.

Primary Actor: Flockr's user.

Trigger: The member clicks on the mic button next to the name of another member.

visual diagram for use cases:



Interface:

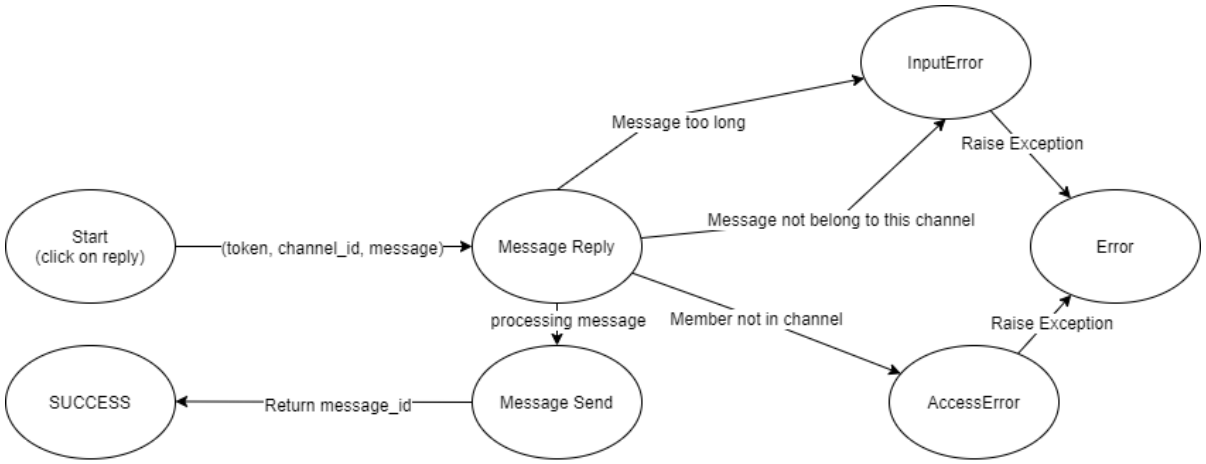
Function Name	HTTP Method	Parameters	Return type	Exceptions	Description
Message/reply	POST	(token, channel_id, message_id, message)	{message}	InputError when any of: Original message is more than 1000 characters The message user wants to reply with message_id not in the channel with channel_id AccessError when: the authorised user has not joined the channel they are trying to post to	Send a reply message from authorised_user to the channel specified by channel_id, which contain the message that authorised_user want to reply to. The final message reply sent to channel contain the original message with sender's name together with the message that authorised_user want to send.
Submitting/files	POST	(token, channel_id, local_files)	{}	InputError when any on them happened: The local_file is invalid. AccessError when: the authorised user has not joined the channel they are trying to post to	Given a list of files and submits them as message in the channel the user joins.
room/ voicechat/ create	POST	(token, channel_id)	{room_id}	InputError: when Channel ID is not a valid channel AccessError: when the authorised user is not a member in this channel	Create a voice chat room that every member in the channel can talk inside

room/ voicechat/ detail	**GET**	**(token, channel_id, room_id)**	**{all_members, Total_Member}**	**InputError: when Channel ID is not a valid channel** **when Room ID is not a valid voice chat room** **AccessError: when the authorised user is not a member in this channel**	**Given a voice chat room with ID room_id that the authorised user is part of, provide basic details about the voice chat room**
room/ voicechat/ join	**POST**	**(token, channel_id, room_id)**	**{}**	**InputError: when Channel ID is not a valid channel** **when Room ID is not a valid voice chat room** **AccessError: when the authorised user is not a member in this channel**	**Given channel_id of a channel and room_id of room that the authorised user can join, adds the user to that voice chat room**
room/ voicechat/ leave	**POST**	**(token, channel_id, room_id)**	**{}**	**InputError: when Channel ID is not a valid channel** **when Room ID is not a valid voice chat room** **AccessError: when the authorised user is not a member in this channel**	**Given the authorised user's token and remove the authorised user from the voice chat room, when the user intends to leave.**

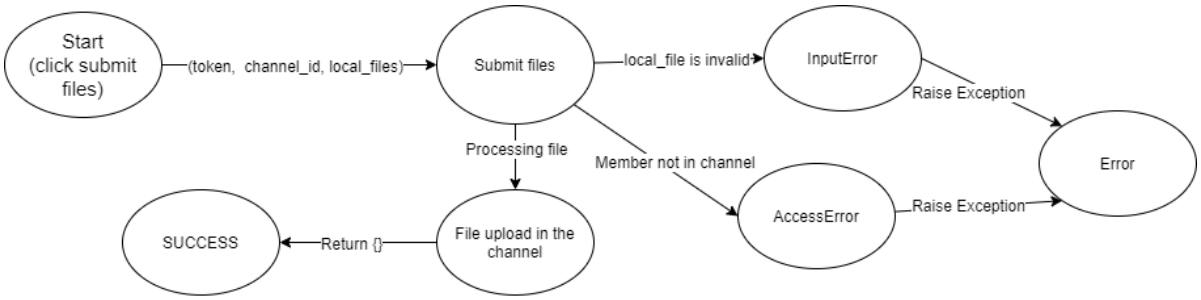
room/ voicechat/ mute	**POST**	**(token, channel_id, room_id, u_id)**	**{members that are muted}**	**InputError: when Channel ID is not a valid channel** **when Room ID is not a valid voice chat room** **When u_id does not belong to a user** **AccessError: when the authorised user is not an owner in this channel**	**mute a user (with user id u_id) inside the voice chat room with ID room_id. Once the user is muted, then the user cannot speak anymore.**
room/ voicechat/ unmute	**POST**	**(token, channel_id, room_id, u_id)**	**{}**	**InputError: when Channel ID is not a valid channel** **when Room ID is not a valid voice chat room** **When u_id does not belong to a user** **AccessError: when the authorised user is not an owner in this channel**	**unmute a user (with user id u_id) inside the voice chat room with ID room_id. Once the user is unmuted, then the user can speak again**

state diagram:

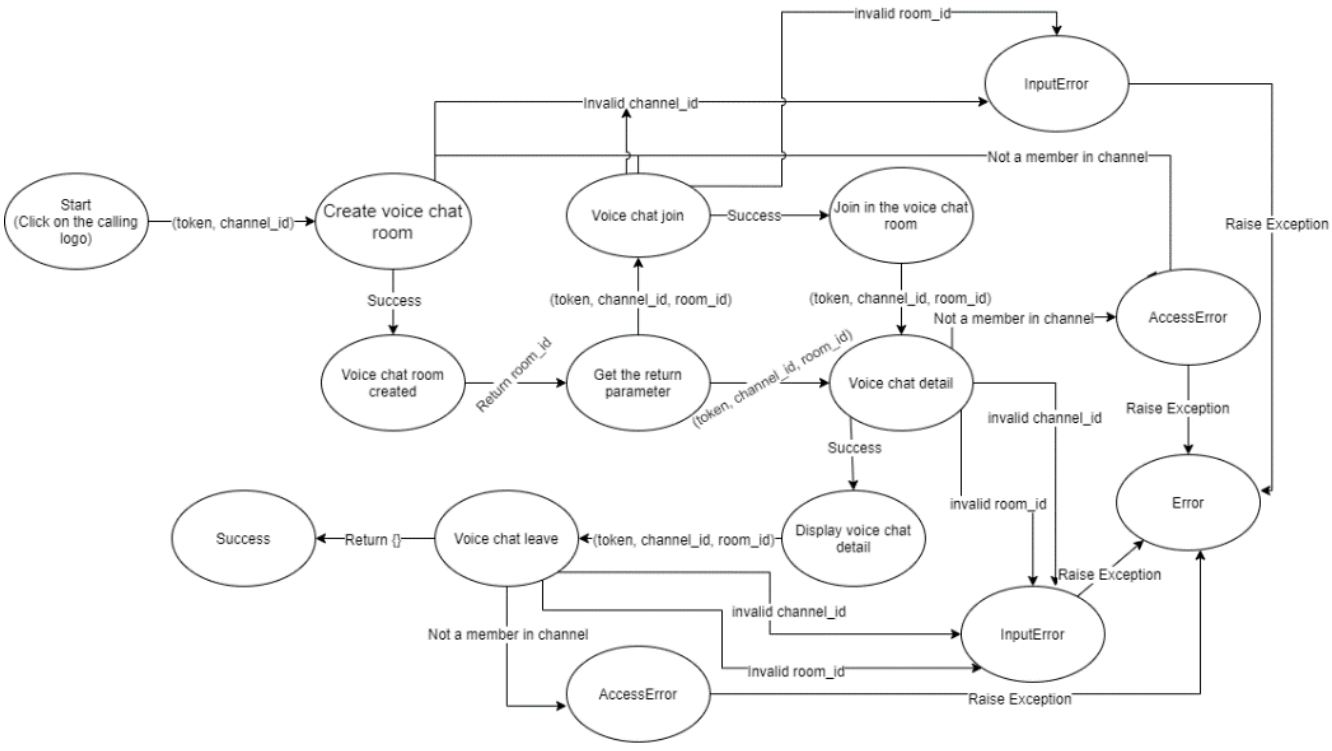
message/reply:

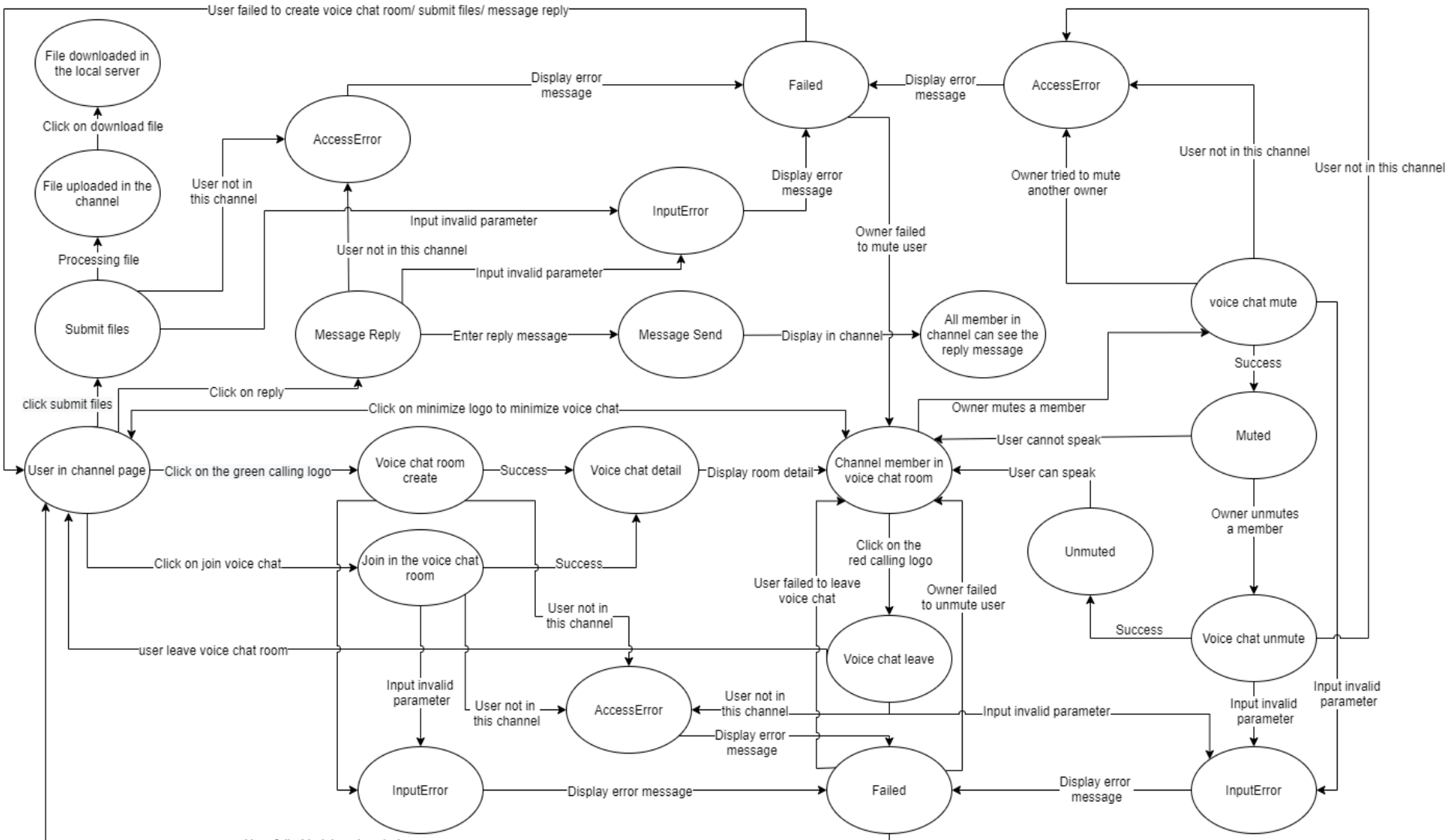


submit/file



room/voicechat:




User failed to create voice chat room/ submit files/ message reply
File downloaded in the local server
Click on download file
File uploaded in the channel
Processing file
Submit files
click submit files
User not in this channel
AccessError
Display error message
Failed
Display error message
AccessError
User not in this channel
User not in this channel
Owner tried to mute another owner
Input invalid parameter
InputError
Display error message
Owner failed to mute user
User not in this channel
Input invalid parameter
Message Reply
Enter reply message
Message Send
Display in channel
All member in channel can see the reply message
voice chat mute
Success
Click on reply
Click on minimize logo to minimize voice chat
Owner mutes a member
User cannot speak
Muted
User in channel page
Click on the green calling logo
Voice chat room create
Success
Voice chat detail
Display room detail
Channel member in voice chat room
User can speak
Owner unmutes a member
Click on join voice chat
Join in the voice chat room
Success
User not in this channel
Click on the red calling logo
User failed to leave voice chat
Owner failed to unmute user
Unmuted
Success
Voice chat unmute
user leave voice chat room
Voice chat leave
Input invalid parameter
User not in this channel
AccessError
User not in this channel
Input invalid parameter
Input invalid parameter
Display error message
Input invalid parameter
InputError
Display error message
Failed
Display error message
InputError
User failed to join voice chat room