

Planning

2020/11/10

Elicitation

Miriam Tuvel, z5257404

Questions to ask:

1. What group messaging system do you currently use for teamwork communication?
2. How often do you use the messaging system?
3. Do you tend to use this app for small or big groups?
4. Does it work effectively for your group size?
5. Do you think it would work well for a different group size?
6. Do most participants of the groups you are in actively participate? If not, why do you think they don't?
7. Do you think the app you're using effectively allows your team to work on multiple tasks at once? If not, why do you think that?
8. Does the app have features that help your group set out different tasks? E.g boards
9. Do you think your app is effective at allowing collaborative work such as sharing files and images?
10. Is it easy for you to go back and find information about a particular question you raised in the group?
11. Do you think the app is good at starring or pinning messages to highlight to the group important messages?
12. Do you feel like your message has been seen when sent e.g are there good ways for people to respond to show they have read and understood your message?
13. What features do you think would enhance communication between all team members?
14. What other problems do you have with the apps?

Response (Deborah Bui, deborah.bui@ad.unsw.edu.au)

1. FB messenger
2. Every day
3. Small (less than 10)
4. No because with more than a few participants its very hard to have a congruent conversation especially with multi-faceted projects
5. Yes maybe for a smaller group
6. Yes
7. No, fb messenger can't create different channels in a group to deal with different things, and it's hard to keep track of several discussions about different topics that aren't happening at once

8. No but would make a big difference if able to set out tasks on the apps that everyone can have easy access to.
9. It is alright
10. Yes - you can search the group directly which is good.
11. No
12. Yes - reactions are very useful and read receipts are effective
13. Channel creation, ability to subject or flag importance of certain messages, allow someone to send through a task and you to indicate you've started on it without having to type out message
14. As mentioned before not being able to allocate tasks clearly and to see everyone's progress as they work on it.

Response (Racheli Jacobs, racheli.jacobs@ad.unsw.edu.au)

1. Teams
2. Everyday
3. Both-sometimes 4, sometimes 20
4. yes
5. yes but for bigger groups it works well only when it's giving opinions on a particular task. However, it doesn't work as well when you have many subgroups in one group.
6. mostly. However, it may be helpful to have different ways for easy responses like polls.
7. no- there is no way to split up conversations like threads so it can be confusing when different groups within the groups both need to speak.
8. no- which means when allocating tasks it's usually done as a normal message which is not very effective.
9. yes, there is a section for sharing files
10. yes, you can search a keyword and it clearly shows all options on the side and what group they are from.
11. yes-there are options to send an urgent message and gives the choice to notify continuously if very important.
12. yes-you are able to react with emojis letting them know you read it
13. It would be helpful to have a 'to-do' list that can be accessed in the chat. It would be even better if it could be continuously edited so people can add things and also demonstrate when something has been done.
14. When working with many people and separating into different tasks we usually have to make separate groups and then send our ideas back to the main group to get their opinions. It would be better if there was a way for bigger groups to work and communicate in a more clearer way.

Two features to resolve:

1. To make it better for bigger groups in Flockr that have big projects is to create separate channels/threads, where all members of the (main channel) project team have access to. This way different tasks can be split into different groups so conversations can be easily followed and found and different sub-groups can talk at the same time with congruent messages.

2. To make teamwork more effective we can make a to-do list which is a pinned message at the top of each channel that can be edited by all participants. Hence, all team members can click on the to-do list to see what tasks need to be done and things can be easily added and removed throughout the day as well as showing progress.

Uses Cases

Yiyang Huang, z5313425

User Story with UAC - Issue 1

As a user of Flockr, I want to have a project team that allows us to create multiple channels within the project team, so that we can assign each channel to a specific aspect of the project and related communications can be posted in that channel for future reference whereas the project team is the main channel.

- When a project team is created, users can create multiple channels within the project team.
- Each channel can be assigned with names.
 - Duplicate names of channels are not allowed.
 - Maximum length of the channel's name is 30 characters.
 - Channels with no name assigned will be set to default "channel_<date_created>".
- Users can view each channel as sub groups in the main project team(group) and allows messages to be sent across channels at the same time.

User Story with UAC - Issue 2

As a user of Flockr, I want to have a to-do list that pins messages at the top of each channel which can be edited by all members, hence our team can easily see the upcoming tasks that need to be done and daily progress of our team.

- To-do list is at the top of each channel.
-
- When messages are pinned, users can insert it into the to-do list at the top of each channel.
- When a message is pinned, it is no longer a message, it becomes a to-do list item that stores: due dates, topic, people assigned to this task, extra information.
- Users can edit, add, remove items in the to-do list.
- Users can see the completed or removed to-do items.
- When users click on the to-do list item, it will show related information about this task. (e.g. due dates, people assigned etc.)

Use Cases - Issue 1

- Use Case: Create a project team (main channel) and have multiple channels within it
- Goal in Context: Users need to create channels related to a specific aspect of the project after creating a project team.
- Scope: Flockr
- Level: Primary function
- Preconditions: User is logged in, no project team (main channel) is created.
- Success End Condition: User is logged in, a project team is created, channel(s) within the project team is(are) created.
- Failed End Condition: No project team is created.
- Primary Actor: User
- Trigger: User clicks on the button “+” on the section that labels the name of the project team in the sidebar.

Main Success Scenario - Issue 1

1. User click on the ‘+’ button on the top left corner
2. Flockr asks user for name of the project team
3. User enters the name
4. User clicks on the button ‘+’ on the side bar where the project team is located
5. Flockr asks user for name of the channel
6. User enters the name
7. Flockr creates a channel within the project team with the given names

Use Cases - Issue 2

- Use Case: Pin message as to-do list items
- Goal in Context: Users need to pin a message as a to-do list item on the top of the channel and stores additional details.
- Scope: Channel within the Flockr’s project team
- Level: Subfunction
- Preconditions: User is logged in, a channel is created within a project team, no message pinned on the to-do list of the channel.
- Success End Condition: A message is pinned on the to-do list of the channel, shows at the top of the window, and stores additional details: due dates, people assigned to the task, topic, extra information.
- Failed End Condition: No message is pinned on the to-do list of the channel.
- Primary Actor: User
- Trigger: Users click on the ‘pin’ button of a message in the channel.

Main Success Scenario - Issue 2

1. User clicks on the ‘pin’ icon on a message

2. Flockr saves the message to the to-do list of that channel an item appear at the top of the window
3. Flockr asks user what additional details they wish to add
4. User enters the due date and assignees
5. Flockr adds the due date and related members to the to-do list item

Validation

Miriam Tuvel, z5257404

Response (Deborah Bui, deborah.bui@ad.unsw.edu.au)

The use of threads solved my issue of working on multifaceted projects in large groups. It allows many conversations to be had between people that can be easily found and followed. A to-do list also solves the original issue of not having anything on the app that can be easily used to allocate tasks. Being able to see past to-do lists is also very helpful as it allows me to see if tasks have already been completed and by whom. Therefore, I feel like my two main issues have been very adequately addressed.

Response (Racheli Jacobs, racheli.jacobs@ad.unsw.edu.au)

The solutions given solve my biggest issues when working on Teams as the first solution of adding the option of threads to the channel solves the issue of having to create sub-groups when working in big groups and having to repeat ourselves in the main group as all the members can see every thread. Also, adding a to-do list makes the app a lot more effective for dividing up team-work and being able to continuously edit and setting deadlines will make the to-do list very useful.

Interface Design

Mia Bueno, z5210209

Requirements:

1. Users need to create a project team and within the project team users can create multiple channels for specific use.
2. Users need to pin a message as a to-do list item on the top of the channel and stores additional details.

In addition to all the channel functions that are already described in the spec, the following functionality is needed to address the users' problems and feed the requirements.

Note:

- Since it is required that users can view each channel as sub groups in the main project team (group) and messages are allowed to be sent across channels simultaneously, *channels/create* feature of 'is_public' is redundant
- *message/pin* and *message/unpin* functionalities are already set up to mark messages as "pinned" and "unpinned" to be given special display treatment by the frontend. This special treatment is adding it to the view of the to-do list pinned at the top of respective channels.

Function Name	HTTP Method	Parameters	Return type	Exceptions	Description
team/create	POST	(token, name, is_public)	{team_id}	InputError when any of: <ul style="list-style-type: none">• Name is more than 20 characters long	Creates a new project team in which users can create multiple channels within
team/invite	POST	(token, team_id, u_id)	{}	InputError when any of: <ul style="list-style-type: none">• channel_id does not refer to a valid channel.• u_id does not refer to a valid user AccessError when <ul style="list-style-type: none">• the authorised user is not already a member of the channel	Invites a user into a project team and is authorised to invite themselves into any existing channel under the project team
message/broadcast	POST	(token, team_id, message)	{message_id}	Input error: <ul style="list-style-type: none">• Message is more than 1000 characters AccessError:	Sends a message across all channels

				<ul style="list-style-type: none"> when: the user is not a member of the project team 	
channels/create (modified)	POST	(token, team_id, name)	{channel_id}	Input error: <ul style="list-style-type: none"> name already exists name length > 30 characters AccessError: <ul style="list-style-type: none"> when: the user is not a member of the project team 	<p>Creates a new channel accessible to everyone that's a member of team project</p> <p>Channels with no name assigned will be set to default "channel_<date_created>"</p>
todo/create	POST	(token, team_id, channel_id)	{todo_id}	Input error: <ul style="list-style-type: none"> todo list in channel already exists name length > 10 characters AccessError: <ul style="list-style-type: none"> when: the user is not a member of the project team 	creates a new empty todo list in specified channel
todo/list	GET	(token, todo_id)	{items}	Input error: <ul style="list-style-type: none"> todo_id does not refer to active/valid to_do list AccessError: <ul style="list-style-type: none"> when: the user is not a member of the project 	Provide a list of all incomplete items in specified todo list (and their associated details) that

				team	the authorised user is part of
todo/listall	GET	(token, todo_id)	{items}	Input error: <ul style="list-style-type: none"> • todo_id does not refer to active/valid to_do list AccessError: <ul style="list-style-type: none"> • when: the user is not a member of the project team 	Provide a list of all (incomplete + complete) items in specified todo list (and their associated details) that the authorised user is part of
item/add	POST	(token, todo_id, is_complete, assignees, time_due)	{item_id}	Input error: <ul style="list-style-type: none"> • todo_id does not refer to active/valid to_do list AccessError: <ul style="list-style-type: none"> • when: the user is not a member of the project team 	Creates a new list item to specified to do list and returns a globally unique item id
item/details	GET	(token, item_id)	{status, assignees, time_due}	Input error: <ul style="list-style-type: none"> • item_id does not refer to existing todo list item AccessError: <ul style="list-style-type: none"> • when: the user is not a member of the project team 	Given an item_id, provide basic details about the to do list item
item/update	PUT	(token, item_id, is_complete)	{}	Input error: <ul style="list-style-type: none"> • item_id does 	Given item_id,

				not refer to existing todo list item AccessError: <ul style="list-style-type: none"> when: the user is not a member of the project team 	update status of item given is_complete boolean
item/edit	PUT	(token, item_id, edit)	{}	Input error: <ul style="list-style-type: none"> item_id does not refer to existing todo list item AccessError: <ul style="list-style-type: none"> when: the user is not a member of the project team 	Given item_id, update task description using edit parameter

Continue next page

Conceptual Modelling (State):

Yue (Yuelanda) Dai, z5310546

auth.py

