Requirements and Design

[Requirements] Elicitation

Interviewee 1

Name: Detian Duan

Email Address: karinaduan6@gmail.com

Background: UNSW Commerce Student, currently using the app “Teamwork Chat” for business group projects.

Q & A

Question: What are some problems you faced when navigating the interface of Teamwork Chat?

Detian: Teamwork Chat does not feel user-friendly as there are insufficient messages to prompt successful implementation of function.

Question: What are some teamwork communication functionalities that you would like to see that are currently not present in Teamwork Chat?

Detian: I would like to have the ability to send private messages to other users directly without having to create a group. This can help to ease up the process of private messaging and speed up the ways to communicate with one another.

Detian: I would also like to have the ability to send voice notes to other users in the group. Currently Teamwork Chat can only send word messages, but having the ability to send voice messages can help users to quickly communicate their ideas with one another without the hassle of typing in long messages.

Detian: Teamwork Chat lacks the ability to send images in groups to other users. As a picture speaks a thousand words, we often need to share images in a group project discussion group to seek help quickly from other users.

Question: Is there any quality-of-life features that you would like to see that are currently not present in flockr?

Detian: Teamwork Chat can introduce status information for each user to show if a user is online or offline in the group. This can help to better coordinate work and let you know who is available to help you in a team setting.

Interviewee 2

Name: Jack Li

Email Address: jackli3691@gmail.com

Background: UNSW undergraduate student, year 3, currently majoring in Actuarial Studies

Q & A

Question: Have you ever used any team chat or collaboration applications in learning and studying? What are they?

Jack: Yes, I did use some team chat apps when I was doing a teamwork project. I used DingTalk in order to chat and discuss with my teammates when I have teamwork projects.

Question: What problems or difficulties have you encountered when using these apps?

Jack: Firstly, sometimes when I realized that I had been assigned with tasks, the allocating tasks message was hundreds of messages before the current ones. I had to search through messages even though I typed in some key words. Compared with file searching, searching for messages is not that time-consuming. When I need to search for public files, I have to search through all files so as to find the correct one. This becomes extremely tedious when hundreds of files come along with complicated file names.

Questions: What solutions would you suggest in order to solve these problems?

Jack: To solve the first difficulties, a functionality of group notice or announcement that is easy to access should be implemented. For the second one, a file system with file tags will be a great help to distinguish different files.

[Requirements] Analysis & Specification - Use Cases

Interviewee 1

Use Case 1

Use Case: Sending pictures

Goal in Context: User needs to send pictures in a channel

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be an owner or an member of the channel where the channel notice post in

Success End Condition: The user successfully sent pictures in channel

Failed End Condition: The user can not send pictures in channel

Primary Actor: User

Customer Trigger: User clicks the "sending picture" button then clicks the "select picture" button. In the end, the user click "send"

Main Success Scenario:

Step 1. User logs in with correct email and password
Step 2. User clicks the correct channel button
Step 3. User clicks "sending picture" button
Step 4. User can now view all local images
Step 5. User select the image he/she want to send and click send button
Step 6. User successfully send the image in channel

Use Case 2

Use Case: Send private messages

Goal in Context: User needs to leave send messages directly to another user

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in

Success End Condition: The user sent the message to another user

Failed End Condition: The user failed to sent the message to another user

Primary Actor: User

Customer Trigger: User clicks another user's name tag button then click "message" button

Main Success Scenario:

Step 1. User logs in with correct email and password

Step 2. User clicks the correct user's name tag

Step 3. User clicks "message" button

Step 4. User writes the message

Step 3. User clicks "send" button

Step 5. User successfully sent the message to target user

Use Case 3

Use Case: Get status of other users

Goal in Context: User needs to access user status to obtain current status of other users.

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and

Success End Condition: The user viewed the other user's status

Failed End Condition: None

Primary Actor: User

Customer Trigger: User clicks another user's name tag button then click "status" button

Main Success Scenario:

- **Step 1**. User logs in with correct email and password
- **Step 2**. User clicks the target user
- **Step 3**. User clicks “status” button
- **Step 4**. User successfully view the status of the other user

Use Case 4

Use Case: Edit User Status

Goal in Context: User needs to edit user status to return updated information

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be the authorised user making this request

Success End Condition: The user status has been edited by the user

Failed End Condition: None

Primary Actor: User

Customer Trigger: User clicks “User Details” button then click “User Status” button and lastly click “Edit User Status” button

Main Success Scenario:

- **Step 1**. User logs in with correct email and password
- **Step 2**. User clicks “User Details” button
- **Step 3**. User clicks “User Status” button
- **Step 4**. User clicks “Edit User Status” button
- **Step 5**. User edits the status
- **Step 6**. User clicks “Save” button
- **Step 7**. A “You have successfully edited your status” message prompt out

Interviewee 2

Use Case 1

Use Case: View channel notice

Goal in Context: User needs to access channel notice to acquire necessary information

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be an owner or an member of the channel where the channel notice post in

Success End Condition: The user viewed the channel notice

Failed End Condition: None

Primary Actor: User

Customer Trigger: User clicks "Channel Notice" button

Main Success Scenario:

Step 1. User logs in with correct email and password

Step 2. User clicks the correct channel button

Step 3. User clicks "Channel Notice" button

Step 4. User successfully view the channel notice

Use Case 2

Use Case: Edit Channel Notice

Goal in Context: User needs to edit channel notice to acquire necessary information

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be an owner of the channel where the channel notice post in

Success End Condition: The channel notice has been edited by the user

Failed End Condition: The user is not an owner of the channel

Primary Actor: User

Customer Trigger: User clicks "Channel Notice" button then click "Edit" button

Main Success Scenario:

Step 1. User logs in with correct email and password

Step 2. User clicks the correct channel button

Step 3. User clicks "Channel Notice" button

Step 4. USer successfully views the channel notice

Step 5. User clicks "Edit" button

Step 6. User edits the channel notice

Step 7. User clicks "Save" button

Step 8. A "You have successfully edited the channel notice" message prompt out

Use Case 3

Use Case: Upload channel file

Goal in Context: User needs to upload a file to the channel

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be an owner or a member of the channel where the file uploads in

Success End Condition: The user successfully uploaded the file they intended to upload

Failed End Condition: The file is oversize

Primary Actor: User

Customer Trigger: User inserts the file, edit “tag”, then clicks “Upload” button.

Main Success Scenario:

Step 1. User logs in with correct email and password

Step 2. User clicks the correct channel button

Step 3. User clicks “Upload File” button

Step 4. User successfully insert the file

Step 5. User edit “Tag” for the file

Step 6. User click “Save” to save the tag

Step 7. User clicks “Upload” button

Step 8. A “You successfully upload a file” message prompts out

Use Case 4

Use Case: Download channel file

Goal in Context: User needs to download a file from the channel

Scope: The web app of flockr

Level: Secondary Task

Preconditions: The User has logged in and has to be an owner or a member of the channel where the file download from

Success End Condition: The user have successfully downloaded the file they intended to upload

Failed End Condition: None

Primary Actor: User

Customer Trigger: User inserts the key word of “tags” and clicks on a tag, then choose a file and click “Download File”

Main Success Scenario:

Step 1. User logs in with correct email and password

Step 2. User clicks the correct channel button

Step 3. User clicks “Channel File” button

Step 4. User types in a keyword in “tag”, then click “search file” button

Step 5. User chooses the file they intended to download

Step 6. User clicks “Download” button

Step 7. A “You successfully download a file” message prompts out

[Requirements] Validation

Interviewee 1

Detian: With the inclusion of the ability to send voicemail, send direct messages and send pictures to other users, it has allowed flexibility in the way I communicate online with others through Flockr. I find these functions very handy and look forward to seeing more functionalities built to simplify my teamwork communication.

Interviewee 2

Jack: These functionalities are all fairly practical functionalities and they successfully solve the problems I met. I will be very glad to see these functions being implemented on Facebook Messengers and your project, Flockr!

[Design] Interface Design

Interface of functionalities inspired by Interviewee 1

Function Name	HTTP Method	Parameters	Return Type	Exceptions	Description
message/privatesend	POST	(token, user_id, message)	{message_id}	**InputError** when any of: Message is more than 1000 characters User ID is not a valid user_id	Send a message from authorised_user to the user specified by user_id
user/status/view	GET	(token, user_id)	{status_id}	**InputError** when any of: User ID is not a valid user_id	Given a User with ID channel_id, provide the status of the user
user/status/edit	PUT	(token, status_id)	{}	**InputError** when any of: Status ID is not a valid status_id	Given a status with status_id, update the status of the current authorised user
channel/sendimage	POST	(token, img_url. channel_id)	{}	**InputError** when any of: img_url returns an HTTP status other than 200. Image uploaded is	Given a URL of an image on the internet, send an image to the channel with channel_id that the authorised user is part of

				not a JPG **AccessError** when any of: the authorised user is not already a member of the channel	

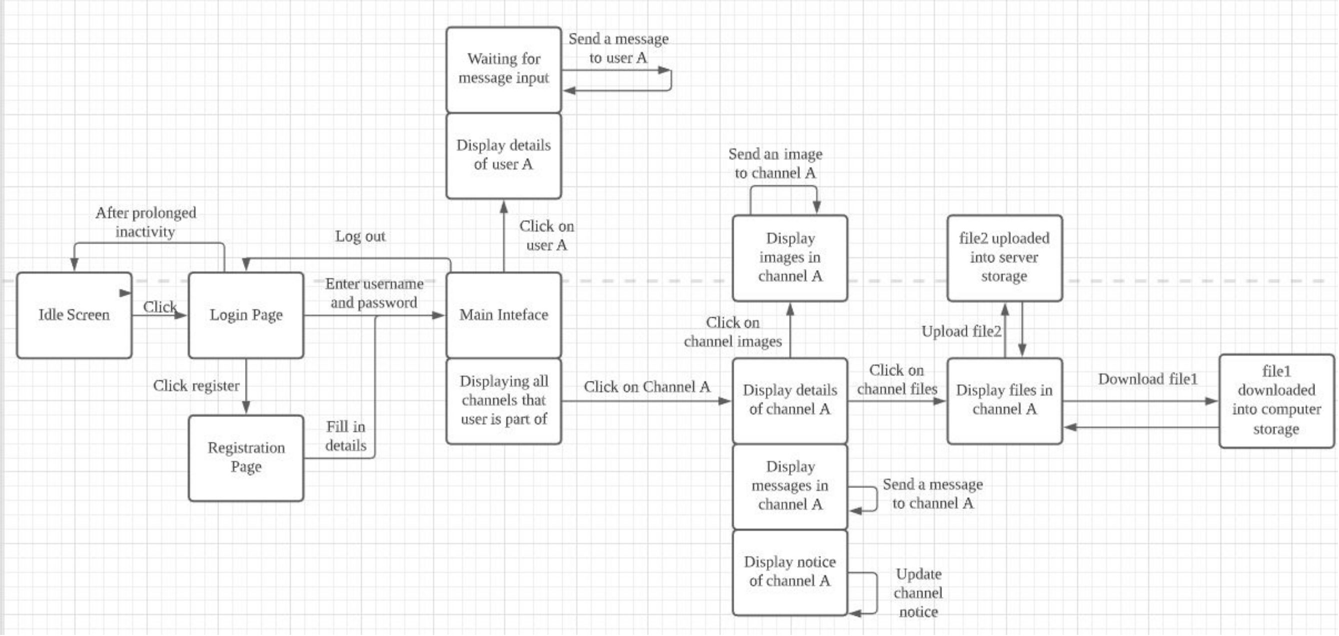
Interface of functionalities inspired by Interviewee 2

Function Name	HTTP Method	Parameters	Return type	Exceptions	Description
channel/notice/view	GET	(token, channel_id)	{notice_message}	**InputError** when any of: channel_id is not a valid channel_id **AccessError** when any of: User is not an member or owner of the channel with channel_id	Given a Channel with ID channel_id that the authorised user is part of, provide the channel notice of the channel
channel/notice/edit	PUT	(token, channel_id, notice_mes	{is_new_notice_success}	**InputError** when any of: channel_id is not a	Given a Channel with ID channel_id that the authorised user is part of, as well as a

		sage)		valid channel_id Notice contains more than 1000 words **AccessError** when any of: User is not an owner of the channel with channel_id	new notice message, change the channel notice with the new notice message
channel/file/upload	POST	(token, channel_id, tag_name, file)	{is_upload_success}	**InputError** when any of: channel_id is not a valid channel_id tag_name is greater than 20 words **AccessError** when any of: User is not an member or owner of the channel with channel_id The size of file is greater than 1 GB	Given a Channel with ID channel_id that the authorised user is part of, with a file and a tag name, upload the file to the database with a tag name
channel/file/viewfile	GET	(token, channel_id, tag_name)	{file_list}	**InputError** when any of: channel_id is not a valid channel_id **AccessError** when any of: User is not an	Given a Channel with ID channel_id that the authorised user is part of, and a tag name, provide a list of all file names that are under the tag that contains input tag name

				member or owner of the channel with channel_id	
channel/file/download	GET	(token, channel_id, file_name)	{file, is_download_success}	**InputError** when any of: Channel_id is not a valid channel_id No file is under the name file_name **AccessError** when any of: User is not an member or owner of the channel with channel_id	Given a Channel with ID channel_id that the authorised user is part of and a file name, download the file under the file name file_name

[Design] Conceptual Modelling (State)


Waiting for message input
Send a message to user A
Display details of user A
After prolonged inactivity
Log out
Click on user A
Send an image to channel A
Display images in channel A
file2 uploaded into server storage
Idle Screen
Click
Login Page
Enter username and password
Main Inteface
Click on channel images
Upload file2
Click register
Displaying all channels that user is part of
Click on Channel A
Display details of channel A
Click on channel files
Display files in channel A
Download file1
file1 downloaded into computer storage
Registration Page
Fill in details
Display messages in channel A
Send a message to channel A
Display notice of channel A
Update channel notice