

H13A CAMEL PLANNING FOR ITERATION 3

[Requirements] Elicitation:

Questionnaire:

1. Have you ever been in a group project before?
2. What platform was used to communicate with your group?
3. What feature stuck out to you when using this particular platform?
4. When using this platform what feature would you like to be implemented and why?
5. How should this feature be used? I.e. a brief description of what the user would experience when using the feature.
6. Who should have access to this feature? (Eg only channel owners / admin / channel members etc)

User 1 Responses: Raymond Yu (raymond.yu2003@gmail.com)

1. Yes
2. Messenger
3. There was lots of things you can do e.g sending gifs and calling team members
4. Adding a to-do list
5. Allows the team to delegate tasks and see what tasks that need to get done.
6. Everybody

Brief solution: Implementation of a to-do list feature that would allow the users to add tasks, due dates and assign the tasks to users within the channel or dm.

User 2 Responses: Raymond Li (raymondli1212@gmail.com)

1. Yes
2. Messenger mainly, but also teams.
3. Sending emails, as I had always wanted to learn to do it.
4. Sending photos.
5. This feature would be used to better communicate with the team through diagrams or any images.
6. Everybody

Brief solution: Implementation of a feature that would allow the users to send images into channel messages or dm messages.

User 3 Responses: Burton Tan (mynameisburton@gmail.com)

1. Yes
2. Messenger and Google Docs

3. (Talking about messenger) → the feature that stuck out to me was the clarity and ease of access
4. Feature i like implemented is daily games where you compare with your friends. It can strengthen bonds and establish who the true trivia king is amongst groups
5. They play a game daily, like wordle, then share results with the team
6. Everybody

Brief solution: Implementation of a feature that would allow the user to play games as well as share their high scores to channels or dms.

[Requirements] Analysis & Specification - Use Cases:

User stories:

As a student, I want my method of team communication to have a to-do list feature so that I am able to see what tasks should be completed first.

As a student, I want a to-do list feature when communicating with my team so that I am able to delegate tasks to my teammates.

As a student, I want to be able to send images to my team so that I can communicate by using diagrams.

As a student, I want my social media communication apps to have some component of gamification so that me and my friends can have some fun everyday competing against each other

Acceptance Criteria:

1. Acceptance criteria for to-do list feature:
 - a. A to-do list feature should be implemented
 - b. Allows user to set due dates
 - c. Allows user to assign tasks to themselves or other users
 - d. Allows user to mark the task as completed
2. Acceptance criteria for sending images feature:
 - a. Allows users to send images as a message
 - b. Allows users to remove the image
 - c. Allows users to edit the message that the image was sent in
3. Acceptance criteria for daily games feature:
 - a. A daily game feature should be implemented
 - b. Allows user to send their high scores to the chat
 - c. Allows users to compete against each other in little games by comparing high scores

Use Cases:

Use Case 1: To-do list feature

1. User logs into Seams
2. User creates a channel or dm
3. User adds other users to the channel if a channel was created
4. User clicks 'create to-do list'
5. Seams asks for what tasks the user wants to add to the to-do list
6. User adds in the tasks which he needs to do
7. The user can then add the due date of the task and assign a user to be in charge of the task

8. Seams creates the to-do list
9. User finishes one of the tasks and goes to check off the task from his list
10. Seams adds a tick next to the task on the to-do list
11. Step 7 and 8 repeats until all of the tasks on the to-do list is finished
12. Seams deletes the to-do list because all of the tasks are completed

Use Case 2: Sending images

1. User logs into Seams
2. User creates a channel or dm
3. User adds other users to the channel if a channel was created
4. User clicks 'send image'
5. Seams asks for the user to attach an image file
6. User sends the message that contains the image file
7. The image is displayed in the channel / dm
 - a. The user can then delete the image (thus deleting the message)
 - b. The user can also edit a previous message and attach an image

Use Case 3: Daily games

1. User logs into Seams
2. User creates a channel or dm
3. User adds other users to the channel if a channel was created
4. User clicks 'Games'
5. User can choose from a variety of games and select one
6. The selected game could then be played by the user
7. Once the user is finished with the game they are presented with a score for their performance in the game which they are able to share to a channel or dm if they would like
 - a. If they choose to share, a message is posted in the channel or dm that lets other users know of the score that was achieved by the user that shared their score.
 - b. If they choose not to share, the game client exits and the home page of seams is displayed.

[Requirements] Validation:

Comment from Raymond Yu based on To-do list use case:

This use case describes the to-do list feature that I had in mind. I think that once a task is completed and ticked off, it should be removed from the list until eventually there is nothing in the list. That would be my one criticism but otherwise sounds good!

Comment from Raymond Li based on the Sending images use case:

Yes this use case encapsulates what I had envisioned for the photo sending feature. I think it covers all the bases and is exactly what I had in mind.

Comment from Burton Tan based on Daily games use case:

This is only one aspect of adding games that I described. I want people within the channel or dm to also play games with the user in multiplayer games. The single player ones can have high score displays but the multiplayer games should display who wins in the channel or dm messages.

[Design] Interface Design:

Our team has chosen to focus on the to-do list feature because it is both achievable and interesting. From here on, the problem we are trying to solve would be the implementation of the to-do list feature to allow for task delegation and organisation.

Input Output Types:

Variable Name	Type
'token'	string
Has substring 'id'	integer
'name'	string
Has substring 'date'	datetime
'todo_task'	Dictionary of form { 'due_date', 'assigned_u_id', 'task' }
'todo_list'	Dictionary of shape { 'name' 'todo_id' 'tasks': ['todo_task'] }

Endpoints:

NAME and DESCRIPTION	HTTP METHOD	INPUT	OUTPUT	ERRORS / EXCEPTIONS
<i>todo/create/v1</i> Given an active token and a string called 'name', creates a to-do list with the title name and returns the todo_id.	POST	{ 'token', 'channel_id', 'name' }	{ 'todo_id' }	Access Error: <ul style="list-style-type: none">- Token is invalid- Token is not a member of the channel Input Error: <ul style="list-style-type: none">- Channel_id does not refer to a valid channel- Name is less than 1 character or more than 1000 characters
<i>todo/createdm/v1</i>	POST	{ 'token', 'dm_id',	{ 'todo_id' }	Access Error:

Given an active token and name, creates a to-do list with the title name and returns the todo_id.		{ 'name' }		<ul style="list-style-type: none"> - Token is invalid - Token is not a member of the dm Input Error: <ul style="list-style-type: none"> - Dm_id does not refer to a valid dm - Name is less than 1 character or more than 1000 characters
<i>todo/delete/v1</i> Given an active token and todo_id, removes the todo list.	DELETE	{ 'token', 'todo_id' }	{ }	Access Error: <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm the todo list is in Input Error: <ul style="list-style-type: none"> - Todo_id does not refer to a valid todo list
<i>todo/list/v1</i> Given an active token and todo_id, prints the list of tasks within the todo list that corresponds to the todo_id.	GET	{ 'token', 'todo_id' }	{ 'todo_list' }	Access Error: <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm the todo list is in Input Error: <ul style="list-style-type: none"> - Todo_id does not refer to a valid todo list
<i>todo/addtask/v1</i> Given an active token, todo_id, due_date, assigned and task, adds the task to the todo list that the todo_id refers to. The due_date is specified by due_date, the person who is assigned to the task is specified by assigned and the task is specified by task. Returns a task_id corresponding to the task in that todo list.	POST	{ 'token', 'todo_id', 'due_date', 'assigned_u_id', 'task' }	{ 'task_id' }	Access Error: <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm the todo list is in Input Error: <ul style="list-style-type: none"> - Todo_id does not refer to a valid todo list - Due_date is in the past - assigned_u_id is not a user in the channel/dm

<p><i>todo/gettask/v1</i></p> <p>Given a valid token and the task id, return the details of that task</p>	GET	{'token', 'todo_id' 'task_id'}	{'todo_task'}	<p>Access Error:</p> <ul style="list-style-type: none"> - Token is invalid - User is not part of the channel or DM the todo is in <p>Input Error</p> <ul style="list-style-type: none"> - Task id does not refer to a valid todo task - Todo id does not refer to a valid list
<p><i>todo/removetask/v1</i></p> <p>Given an active token, todo_id and task_id removes the task corresponding to task_id from the todo list that corresponds to the todo_id.</p>	DELETE	{'token', 'todo_id', 'task_id' }	{}	<p>Access Error:</p> <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm the todo list is in <p>Input Error:</p> <ul style="list-style-type: none"> - Todo_id does not refer to a valid todo list - Task_id does not refer to a valid task in the valid todo list
<p><i>todo/complete/v1</i></p> <p>Given an active token, todo_id and task_id, marks a task as completed in the todo list. This should also remove it from the list.</p>	PUT	{ 'token', 'todo_id', 'task_id' }	{}	<p>Access Error:</p> <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm the todo list is in <p>Input Error:</p> <ul style="list-style-type: none"> - Todo_id does not refer to a valid todo list - Task_id does not refer to a valid task in the valid todo list
<p><i>todo/reassign/v1</i></p> <p>Given an active token, todo_id, task_id and u_id, reassigns the task task_id in</p>	PUT	{ 'token', 'todo_id', 'task_id', 'u_id' }	{}	<p>Access Error:</p> <ul style="list-style-type: none"> - Token is invalid - Token is not a member of the channel or dm

the todo list corresponding to todo_id to the user u_id.				<p>the todo list is in</p> <p>Input Error:</p> <ul style="list-style-type: none">- Todo_id does not refer to a valid todo list- Task_id does not refer to a valid task in the valid todo list- U_id does not refer to a user in the channel or dm
---	--	--	--	---

[Design] Conceptual Modelling (State):

State Diagram: To-Do List (within channel)

Assumptions

- The channel has already been created
- The user token has already been created and is valid
- The parameters put in by the users is always valid (eg. user token, channel id, character length, valid to-do list id)

