

# Planning for New Features

Team 4 Orange

## Elicitation

Eddie Ma

info@vibewire.org

1. Do you use team-based chat apps
  - a. Uses slack
2. What do you use team-based chat apps for? -> identify the TYPE of user
  - a. Management at coworking space
  - b. Uses it to communicate with team about sponsorships, sharing links, etc.
3. Is there anything you find annoying about slack
  - a. its difficult to order & categorise channels outside of renaming them. you don't get access to channels unless you created them, its difficult to go through all the conversations to find what your looking for (search function helps) and there's no interface integration with other software (have to use #commands)
  - b. In regards to channels more specifically:
    - i. can't create team/department channels automatically or add people to them by department
  - c. Trello integration
    - i. Command line instructions are hard to learn with bots
4. What would help with integrating external tools
  - a. Would like to make a GUI so that its easier to navigate and is more intuitive
  - b. Buttons to trigger tasks in slack
5. What would you need with channel categorisation
  - a. i would like categories for Admin, Management, Seniors, Juniors, General, Marketing, etc and then channels within them. Also a Projects category, and within that are channels for all projects and even subchannels for topics within the Project. Like LEVELS
6. How many levels would you want?
  - a. i'd say 3 levels should be enough. any more and it gets rowdy

Hannah Ahn

hannah.ahn12@gmail.com

1. 1. Do you use team-based chat apps
  - a. Uses slack
2. What do you use it for?
  - a. Communication when working on projects in teams
  - b. Shares links, stores files, manages project progress
  - c. As a team member
3. Is there anything you find annoying about slack?
  - a. Lack of project management features
  - b. Have to use separate apps to project manage
  - c. Would be nice to have a tab where to-do tasks were managed within slack
4. What would you need out of the project management features?
  - a. Ability to assign due dates and users to deadlines
  - b. Reminders in specific slack channels
  - c. All consolidated in one page in slack

Matthew Marston  
matthew.marston98@gmail.com

1. Do you use team-based chat apps
  - a. Uses slack
2. What do you use it for?
  - a. Meeting Updates
  - b. Updating people on information from meetings and using it to schedule meetings
  - c. Get messaged by people on slack and just show up
3. What stops you from using Slack more?
  - a. Outdated UI
  - b. No voice capabilities
  - c. Slower than discord
4. What about the UI is outdated?
  - a. Colour scheme
  - b. Fonts
5. Would you use it if the theme could be changed?
  - a. Yes
  - b. Just a bit meh about slack

# Analysis & Specification - Use Cases

For Eddie:

As a team manager and workspace administrator, I want to be able to create channel levels so I can organise my teams and conversations more effectively.

User Acceptance Criteria:

1. Needs to reduce the time taken to create new channels with teams in them
2. Needs to reduce the time taken to navigate between different channels
3. Need to be able to create folders in the sidebar
4. Folders need to be able to contain folders within them
5. Folders need to be able to contain channels within them
6. Folders need to be able to be assigned users
7. Users assigned to folders can be made an administrator
8. Folder administrators can create new channels and sub-folders
9. Channels inside folders need to contain users assigned to all folders above them
10. Folders need to be able to be collapsed.

Use Case:

Given: that the user wants to create a new folder

When: they would click a button to create a new folder:

Then: a new channel folder would be created

Given: the user wants to add people to a folder

When: they click a button to add team members

Then: Open up a UI to assign users and admin permissions to the folder

Given: that new users and admin have been added to a folder

When: any user checks any channel in the folder

Then: give the user access to the channel

For Hannah:

As a team member liaising with clients I want to be able to assign and track tasks inside the app so I can keep my project management in one app.

User Acceptance Criteria:

- A new tab in the sidebar for tasks assigned
- Tasks need to be able to be assigned in every channel
- Users need to be able to assign due dates to tasks
- Users need to be able to assign users to tasks
- Tasks need to be able to be sorted by due date
- Tasks need to be able to be sorted by assigned user
- Tasks need to remind users on specific channels or chats

Use Case:

Given: That there is a task that needs to be assigned to a user

When: The user clicks on a message

Then: A user can turn a message into a task with a due that and assigned user

Given: That a user wants to view the tasks in a workspace

When: The user clicks a tab in the sidebar

Then: An Interface with a list of tasks already assigned is presented

Given: That a task has already been assigned to a user and a channel

When: The task is becoming due

Then: The channel and user are altered to the task becoming due

For Matthew

As a member of slack channels, I want to be able to change the colour scheme and font of the frontend to make it more visually appealing to use

User Acceptance Criteria

- An ability to change the overall colour scheme of the frontend
- An ability to change the font on the frontend
- Colours and fonts must be visually appealing

Use Case:

Given: That the user wants to change the colour scheme

When: The user presses a button on the sidebar

Then: A user interface to change font and colour pops up

Given: That the colour interface has been chose

When: The user presses on a colour or font choice

Then: Immediately change the user interface colour and font for the frontend

# Validation

For Hannah

1. Does the use case resolve your problem?
  - a. Yes
2. Do you have anything to add?
  - a. No

For Matthew

1. Does the use case resolve your problem?
  - a. Yes
2. Do you have anything to add?
  - a. Not really
  - b. Not particularly passionate about chat apps

For Eddie

1. Does the use case resolve your problem?
  - a. Sounds about right
2. Do you have anything to add?
  - a. Want to add departments
  - b. E.g. Greg is a Manager from Accounting
    - i. So he would get auto assigned to the Accounting Department folder channels
    - ii. And also Management if there is one
3. What would be the difference of adding folders to users and users to folders?
  - a. You'd be able to add several departments to a folder group

Therefore, the User Acceptance Criteria and Use Case for Eddie needs to be modified to:

User Acceptance Criteria:

1. Needs to reduce the time taken to create new channels with teams in them
2. Needs to reduce the time taken to navigate between different channels
3. Need to be able to create folders in the sidebar
4. Folders need to be able to be collapsed on the sidebar
5. Folders need to be able to contain folders within them
6. Folders need to be able to contain channels within them
7. Folders need to be able to be assigned users
8. All users of multiple existing folders can be added to new folders
9. Users assigned to folders can be made an administrator
10. Folder administrators can create new channels and sub-folders
11. Channels inside folders need to contain users assigned to all folders above them

Use Case:

Given: That the user wants to create a new folder

When: They would click a button to create a new folder:

Then: A new channel folder would be created

Given: A folder admin wants to add people to a folder

When: They click a button to add team members

Then: Open up a UI to assign users and admin permissions to the folder

Given: That new users and admin have been added to a folder

When: Any user checks any channel or folders in the folder

Then: Give the user access to all channels and folders underneath specified folder

Given: That an admin wants to add a group of users to a folder

When: The admin can opens a UI to add users

Then: The user is presented with options to add all members of existing folders



# Interface Design

Moving forward with Eddie's User Story.

## Data Types

In order to add channel folders, we would need to make folders recursive.

Named exactly:

folder: A dictionary containing: {folder\_id, name, is\_public, folders, all\_members, owner\_members, channels}

folders: a list of folder dictionaries

root: a folder which describes the raw sidebar. This is analogous to /home/ in linux. This root folder has a folder\_id of 0.

Outputs Only:

Named Exactly **folder**:

Dictionary containing:

{folder\_id, name, is\_public, folders, all\_members, owner\_members, channels}. Note that 'folders' contain more folders inside them

## Interface

Function Name	HTTP Method	Parameters (as a JSON object)	Return type	Exceptions	Description
folders/create	POST	{token, name, is_public, folder_id}	{folder_id}	<p>InputError when any of:</p> <ul style="list-style-type: none"><li>folder_id is not valid</li><li>Name is more than 20 characters long</li></ul> <p>AccessError when:</p> <ul style="list-style-type: none"><li>User is not an owner of the folder with folder_id</li></ul>	<p>Create a new folder within the folder of folder_id with a name that is either public or private. The user who made the folder is assigned as owner of the folder.</p> <p>Returns the folder_id of the created folder</p>

folders/list	GET	{token}	{root}	N/A	Return the a root folder only containing the folders which the user has access to
folders/listall	GET	{token}	{root}	N/A	Return the a root folder
folder/details	GET	(token, folder_id, layers)	{name, is_public, folders, all_members, owner_members, channels}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>Folder_id is not a valid ID</li> <li>Layers is not a positive integer</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>Authorised user is not a member of folder with folder_id</li> </ul>	<p>Given a folder with ID channel_id that the authorised user is part of, provide basic details about the channel.</p> <p>Only return details of lower folders corresponding with layer number. For example, layers = 1, will give details of folders in folders, but not the folders within the folders.</p> <p>Layers = 0 will return all the folders underneath the given folder</p> <p>If layers is larger than the number of layers in stored folders, treat it as layers = 0.</p>
folder/invite	POST	(token, folder_id, u_id)	{}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>folder_id does not refer to a valid folder.</li> <li>u_id does not refer to a valid user</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>the authorised user is not already an owner of the folder</li> </ul>	Invites a user with user_id u_id to join a folder with ID folder_id. Once invited, the user is added to the folder immediately. They will also be added to all folders and channels stored underneath that folder.

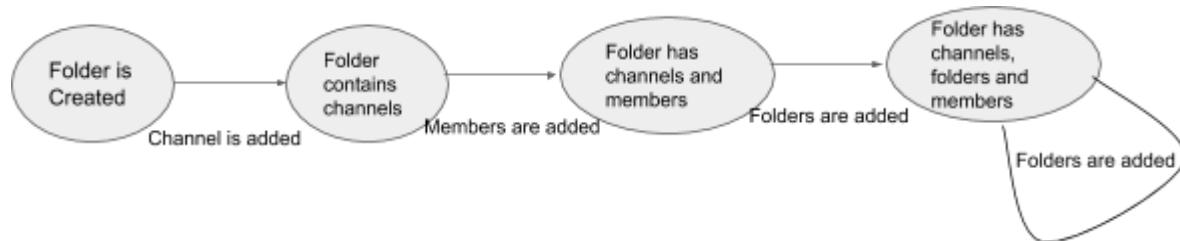
folder/remove	POST	(token, folder_id, u_id)	{}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>folder_id does not refer to a valid folder.</li> <li>u_id does not refer to a valid user</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>the authorised user is not already an owner of the folder</li> </ul>	Removes a user with user_id u_id from a folder with ID folder_id. Once removed, the user is removed from all folders and channels stored underneath that folder.
folder/leave	POST	(token, folder_id)	{}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>folder_id does not refer to a valid folder.</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>the authorised user is not already a member of the folder</li> </ul>	Removes authorised user from a folder with ID folder_id. Once removed, the user is removed from all folders and channels stored underneath that folder.
folder/addowner	POST	(token, folder_id, u_id)	{}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>folder_id does not refer to a valid folder.</li> <li>u_id does not refer to</li> </ul>	Make a user with user_id u_id to become an owner of a folder with ID folder_id. They will also be added as an owner to all folders and channels stored underneath that folder.

				<p>a valid user</p> <ul style="list-style-type: none"> <li>When user with user id u_id is already an owner of the channel</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>the authorised user is not already an owner of the folder</li> </ul>	
folder/removeowner	POST	(token, folder_id, u_id)	{}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"> <li>folder_id does not refer to a valid folder.</li> <li>u_id does not refer to a valid user</li> <li>When user with user id u_id is not already an owner of the channel</li> </ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"> <li>the authorised user is not already an owner of the folder</li> </ul>	Remove a user with user_id u_id as an owner of a folder with ID folder_id. They will also be removed as an owner to all folders and channels stored underneath that folder.
folder/addfoldermembers	POST	(token,	{}	<b>InputError</b> when	Add all_members and

		input_folder_id, adding_folder_id)		<p>any of:</p> <ul style="list-style-type: none"><li>• Input_folder_id does not refer to a valid id</li><li>• Adding_folder_id does not refer to a valid id</li></ul> <p><b>AccessError</b> when</p> <ul style="list-style-type: none"><li>• the authorised user is not already an owner of the Input folder</li><li>• the authorised user is not already an owner of the Adding Folder</li></ul>	owner_members of the adding_folder as members and owners of the input_folder.
channels/create	POST	(token, name, is_public, folder_id)	{channel_id}	<p><b>InputError</b> when any of:</p> <ul style="list-style-type: none"><li>• Name is more than 20 characters long</li><li>• folder_id does not refer to a valid folder</li></ul> <p><b>AccessError</b> when:</p> <ul style="list-style-type: none"><li>• Authorised user is not an owner of specified folder</li></ul>	Creates a new channel with that name that is either a public or private channel. This is stored within the folder with folder_id specified.

# Conceptual Modelling

The proposed feature contains very few changes in state. This is the change in state for the backend. This process is repeated



In the frontend:

