

CVWO Mid-Assignment

Name: Tian Fang

Matriculation Number: A0194568L

Use cases & execution plans

Use case name	Description	Execution plan
Add a new task	On the task index page, the user can add a task by clicking the “New task” button. The user will see a new webpage for creating a new task, where they can enter the title and the description of the new task (the title cannot be blank). Alternatively, the user can choose not to create a new task by clicking the “back” button	Create a new task in the database by Task.create
Show a particular task and associated tags	On the task index page, the user can click the “show” button next to a particular task to show detailed information and tags of the task.	Render Task’s show.html.erb
Edit a particular task	On the task index or show pages, the user can edit a task by clicking the “edit” button to edit the title or the description of the task	Update the task’s title and description by Task.update
Delete a particular task	On the task index page, the user can click the “destroy” button next to a particular task to delete the task	Delete the task by Task.destroy
Add a new tag	On the tag index page, the user can add a tag by clicking the “New tag” button. The user will see a new webpage for creating a new tag, where they can enter the name of the new tag. Alternatively, the user	Since a task can have multiple tags and a tag can be shared by multiple tasks. Both model (task.rb and tag.rb) should contain <i>has_and_belongs_to_many</i> . <pre>class Tag < ApplicationRecord has_and_belongs_to_many :tasks validates :tagname, presence: true end</pre>

	can choose not to create a new tag by clicking the “back” button. If the new tag has the same name as a tag created before. The task will be automatically linked to the tag created before (not completed yet).	<pre>class Task < ApplicationRecord has_and_belongs_to_many :tags validates :title, presence: true end</pre> <p>Additionally, a table named “tags_tasks” is created. Thus, to create a new tag associated with a particular task. @task.tag.create should be used.</p>
Show a particular tag and associated tasks	On the tag index page, the user can click the “show” button next to a particular tag to show detailed information of the tag and associated tasks.	Render Tag’s show.html.erb
Edit a particular tag	On the tag index or show pages, the user can edit a tag by clicking the “edit” button to edit the name of the tag	Update the tag’s name by Tag.update
Delete a particular tag	On the tag index page, the user can click the “destroy” button next to a particular tag to delete the tag	Delete the tag by Tag.destroy
Go back to the tasks index page	On pages dealing with tags, the user can click on the “Tasks” button to go back to the tasks index page.	<p>Add a line of code in Tag’s show.html.erb</p> <pre><%= link_to 'Tasks', tasks_path %></pre>
Go back to the tags index page	On pages dealing with tasks, the user can click on the “Tags” button to go back to the tags index page.	<p>Add a line of code in Task’s show.html.erb</p> <pre><%= link_to 'Tags', tags_path %></pre>

Additional functions:

A complete task manager should do much more. Here are some functions that I plan to add later:

1. A user system. Right now, anyone can use the app(<https://tf-taskmanager.herokuapp.com>). However, all tasks and tags are stored in the same database. Thus, terrible results such as accidentally changing other people’s tasks can happen.
2. Add a form to the “create task” page for users to enter the due date.

3. Add a “check” button to complete a task and a button to show all the completed tasks.
4. Add a function to group (or create a project) to organize different tasks.
5. Add a function to add a flag sign to tasks that are very important.

Problems:

Frankly, there is still a lot for me to learn. The problems listed below might actually be extremely easy to solve after I learn more about rails. Anyway, here are the problems I am currently facing.

1. I could not properly link a task with an existed tag (when the user tries to create a new tag that has the same name as a tag created before). I tried using code below but failed.

```
if Tag.find_by(tagname: tag_params[:tagname]).nil?
  @tag = @task.tags.new(tag_params)
else
  @tag = Tag.find_by(tagname: tag_params[:tagname])
  @tag.tasks << @task
end
```

2. When I create the many-to-many association between the two models (Task and Tag). If the additional table containing task_id and tag_id is named tasks_tags, the association could not be successfully created. However, if the additional table is named tags_tasks, rails can correctly identify the many-to-many association. Does this problem relate to the order in which the two models are created?
3. The way I create a new tag associated with a particular task has a problem: whenever a user clicks on the button “show” next to a task, a new tag associated with the task is created even if there is no input from the user. This is because `<% @tag = @task.tags.create %>` was used in Task’s show.html.erb. I wrote some code before based on different Actions View form helpers to get the input for associated tags. However, most of the code failed to achieve the expected result. Right now, I adopt this solution as a temporary solution, and I am sure that I will find a better way to address this problem.