

CVWO Assignment: Riding on Rails

Mid-assignment Write-up

By Liu Zechu (Ao188295L)

Basic Use Cases

Initial: the user chooses whether he wants to log in to his existing account or he is new to the application, in which case he needs to create a new account.

Create new account: the user enters his username, password and password confirmation to set up an account. After creating an account, he is automatically logged in to the application where he can create a new task on his to-do list. There is an option for him to log in using an existing account instead.

Login: the user enters his username and password and submit them. He is then redirected to the page that shows all his uncompleted tasks. On the login page, there is an option for him to be redirected to the page where he can create a new account.

Create: on the page that shows all the user's to-do tasks, there is a field to enter a new task and choose tags to describe this task or different priority levels. Once this is done, submit the form so that it is saved to the database and it will be available to be viewed on the show page.

Read: on the show page, the tasks will be divided by their due dates and displayed under "Overdue", "Today", "Tomorrow", "Next 7 Days" etc. The user can click on a button that redirects him to another page showing his completed tasks and dates of completion. From here, the user can also go back to the page that shows uncompleted tasks.

Edit: on the show page, beside each task there is an Edit button. The user can click on this button and a text field will show up, where he can edit the task and submit the changes. Afterwards, the user will be redirected to the show page with the edited tasks.

Mark as complete: at any time, on the page that shows all the uncompleted tasks, the user can click a button to change the status of a particular task to "completed". This task will then be removed from the list of uncompleted tasks. The user can view this task on the page that shows all the completed tasks. The user can also choose to change the status of the task back to "uncompleted", in which case the task will be shown on the uncompleted task page again.

Delete: the user can delete any task marked as uncompleted or completed by clicking on the Delete button. The user will be asked to confirm this action.

Logout: the user can click on the logout button and the application will redirect to the initial page where the user can either create a new account or log in using an existing account.

Execution Plan

For the user sign-in feature, I will use a gem (possibly Clearance) to achieve this feature. This generates the User model and its associated migration. A Task model will be generated. User is associated with Task through has_many and each task belongs to a user. Each task will also have an attribute that denotes its tag and a status attribute to indicate whether it is uncompleted or completed.

After authentication, the user will interact with the Task model through actions defined in Task Controller. These actions include show_uncompleted, show_completed, new, edit, create, update, mark_as_complete, destroy_uncompleted, destroy_completed etc. In View, show_uncompleted will display all tasks whose status is uncompleted and arrange them based on their due dates. Show_completed displays completed tasks on another page. New allows the user to create a new task which will be saved to the database through create. Edit allows the user to edit a task item which will then be updated. Mark_as_complete will change the status of a task object from uncompleted to completed. Finally, the destroy actions will delete completed and uncompleted tasks permanently but the user will be prompted to confirm the deletion.

The application will be styled possibly using a front-end framework such as Bootstrap.