


# CVWO Riding on Rails 2021/2022 Mid Assignment Submission

By Lee Zong Xun

## Overview

Groups and Steps  is a web application built using Rails and React for CVWO 2021/2022 assignment. It allows you to manage your tasks into groups, and further break them down into steps.

## Architecture

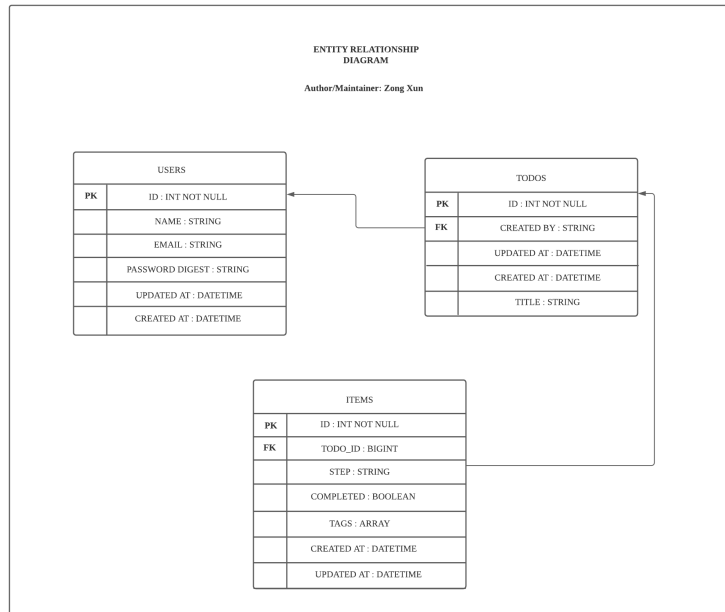
This application adopted the use of MVC architecture for the backend and Flux architecture for the frontend.

## Database Choice and design

- Currently using PostgreSQL (Implemented with Rails)
- Schema

```
1  create table "items", force: :cascade do |t|
2    t.string "step"
3    t.boolean "completed"
4    t.bigint "todo_id", null: false
5    t.datetime "created_at", precision: 6, null: false
6    t.datetime "updated_at", precision: 6, null: false
7    t.text "tags", default: [], array: true
8    t.index ["tags"], name: "index_items_on_tags", using: :gin
9    t.index ["todo_id"], name: "index_items_on_todo_id"
10 end
11
12 create table "todos", force: :cascade do |t|
13   t.string "title"
14   t.string "created_by"
15   t.datetime "created_at", precision: 6, null: false
16   t.datetime "updated_at", precision: 6, null: false
17 end
18
19 create table "users", force: :cascade do |t|
20   t.string "name"
21   t.string "email"
22   t.string "password_digest"
23   t.datetime "created_at", precision: 6, null: false
24   t.datetime "updated_at", precision: 6, null: false
25 end
26
27 add_foreign_key "items", "todos"
```

- Modeling Relationship (One to Many)



## User Story

Description	As an app user, I would like to use the application as a central knowledge and task management platform. So that i can plan my tasks effectively and efficiently.
Acceptance Criteria	Build a task management application to allow real-time tasks updates and management. Allow users to perform basic CRUDs functionalities, a tagging system to categorise tasks and a search functionality to navigate between groups.
Notes	CRUDS functionality is dependent on API availability & integration.

## Use Cases

Input	Positive Response	Negative response
Create a group	A new thread of steps will be created. Users will be prompted to start on their steps.	Unable to create a group. Error message.
Edit and Delete a group	Group's title will automatically be updated/deleted. All steps relevant to the group will also be deleted.	Error message.

Create a step	User will begin to input steps to track their tasks.	Unable to create step. Error message.
Editing and Deleting a step	Step's name will be updated/deleted accordingly.	Error message.
Search for groups	If a group exists, the user will be directed to the page containing the group's information. Otherwise, the user will be prompted to create a new group.	No error message.
Filter steps	User will be able to filter the steps according to any keyword they typed in. Successful filters will filter the steps accordingly.	No error message.
Adding tags	Tags added will be placed right beside the input box.	No error message.

## Reflection

So far, the experience has been absolutely amazing. Despite the numerous ups and downs, debugging, and internal screaming, I am really satisfied with the outcome of my mid-submission. While I had previous web programming experience (through my club activities), I had never built a full-stack web application from scratch nor have I worked with Ruby on Rails.

As a result, the majority of my time was spent learning Rails syntax, reading documentation, and constructing the backend. Setting up the Linux Environment was also difficult, and I had to use Stack Overflow to debug several of the difficulties. Adopting the MVC design for the backend and the Flux architecture for the frontend has opened up my eyes to the many software engineering paradigms, particularly the notion of state management. Using Redux has made it exceedingly simple to maintain track of app-wide state with a single source of truth. This mid-assignment has helped me to further hone my understanding of web programming, and I intend to further improve it with the following features in the coming month.

Intended improvements over the next month

1. Implement other ancillary enhancements (i.e markdown and code support)
2. Introduce testing
3. Refactor to Typescript

Frontend Repo: <https://github.com/Zxun2/cvwo-assignment-frontend>

Backend Repo: <https://github.com/Zxun2/cvwo-assignment-backend>