# CVWO Mid-Assignment Submission

# Lee Wei Min (A0201446R)

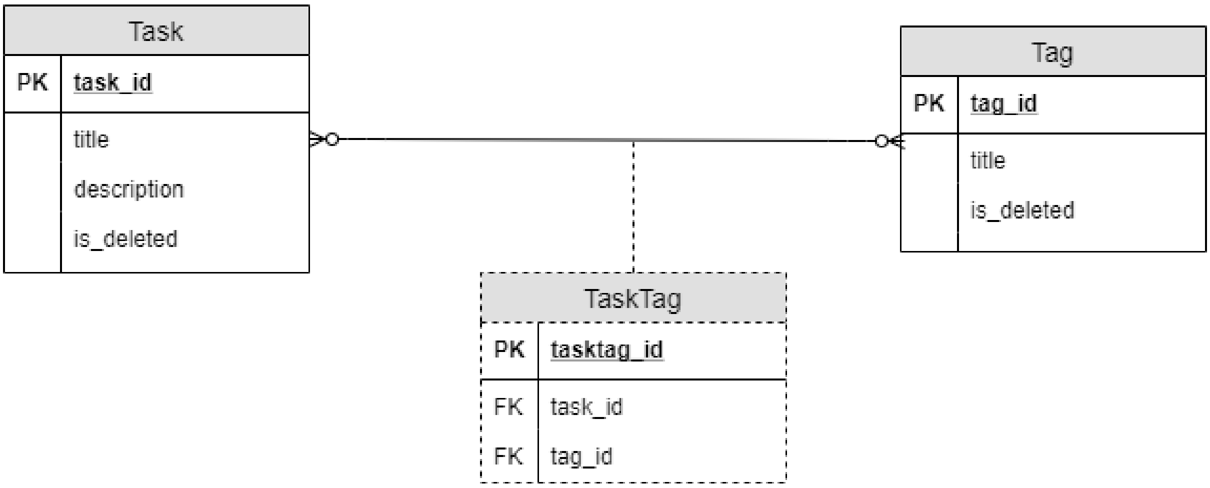
## 1 Overview

### 1.1 Architecture

The to-do app is architected using the MVC (Model-View-Controller) pattern, using PostgreSQL and Ruby on Rails for the back-end (as a data JSON API endpoint) and React for the front-end (reads data from endpoint).

### 1.2 Database

The app has two models: Task and Tag. A task can have many tags, and vice-versa. This means Task and Tag have a many-to-many relationship. This is the entity relationship diagram:



### 1.3 Design

The app is designed to be minimalistic, lightweight and fast to maximise user experience. It is inspired from the Any Do app, which is one of the most popular to-do apps in the market.

### 1.4 Use Cases

Currently, the user is able to:

- Perform Create, Read, Update and Delete (CRUD) operations for tasks

- Search for tasks
- Perform CRD operations for tags

## 2 Problems

Although I had experience in web development, I have not worked with these technologies before, so I had to spend some time to familiarise myself with the syntax. Setting up of the development environment was also not easy as I am using Windows Subsystem for Linux (Bash), which Rails does not natively support. As the program grows more complex, it becomes increasingly harder to keep a mental note of how each part of the app interacts with one another. This makes it more likely to introduce bugs.

## 3 Suggestions

Currently, I have developed a minimum viable product (MVP) of the app and deployed it on Heroku:

- Back-end: https://immense-sands-22358.herokuapp.com/api/tags and https://immense-sands-22358.herokuapp.com/api/tasks
- Front-end: https://agile-shore-02575.herokuapp.com/

By end-submission, I intend to implement the following improvements, given enough time:

- Refactoring:
  - Use TypeScript
  - Use Redux for state management instead of passing props
- Set up a Cron job to perform a daily backup of the database
- Introduce login feature
- Implement other ancillary enhancements such as deadlines and user statistics
- Introduce testing

These improvements are to improve user experience and to provide opportunities to learn new technologies.

## 4 Execution Plan

I will continue to follow established software engineering principles in developing the app. To implement the suggested features, I will also continue to consult online tutorials and apply the relevant segments accordingly. Before introducing each ancillary feature, I will consider its impact on user experience and benchmark it against industry standards.