CVWO 2019/2020

Name: Low Jie Feng Matriculation Number: A0201747J

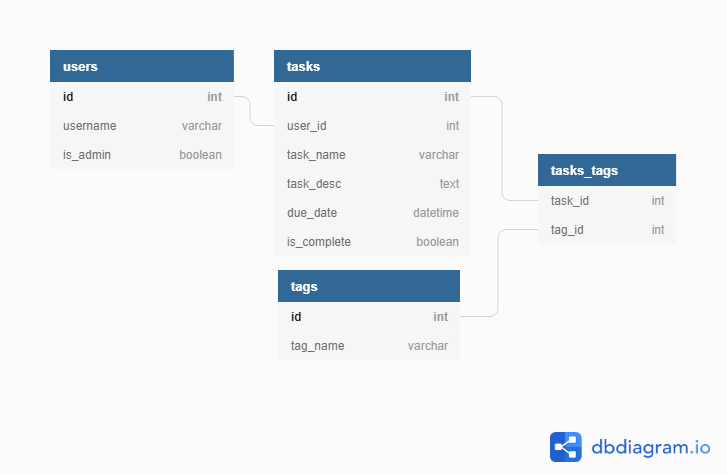
Basic Use Cases for ToDoApp

|  |  |  |
| --- | --- | --- |
| **User** | Normal | Add task |
| View a specific task |
| View all tasks |
| Update task (such as name, description, due date and status) |
| Delete task |
| Add a tag to a task |
| Delete a tag of a task |
| Search via tag |
| Search via task’s name and task’s due date |
| Admin | View a specific user |
| View all users |
| Update user (such as name and admin status) |
| Delete user |
| All operations by a normal user |

General Execution Plan

1. Add basic models and methods for Task and User
2. Add authorisation
3. Add UI for forms and index pages
4. Add Tag model and setup the association
5. Add Tag methods within each Task
6. Add searching operations
7. Streamline the index pages for users and tasks
8. Test the application
9. Setup the application for deployment
10. Add additional features

Database Model



Execution Plan – with a bit more details

Models and Controllers

1. After setting up the model for user and task, authorisation was done in simplified way – only with username.
2. Users can key in the username they used previously to retrieve their past tasks. Also, if they do not logout, the session is preserved within cookies[:user\_id].
3. Upon logging in, users can perform CRUD actions on their own tasks.
4. Admins (saved within cookies[:is\_admin]) can perform CRUD actions on users.
5. The tag model is associated with the task model via a join table. It is setup such that deleting a task would delete related ActiveRecords in the join table, but the tags are still present in the tag table.
6. Creation / linking of tags – On a task’s new, show and update pages, tags should be able to be added or retrieved, and these chosen tags would be linked to the task.
7. Searching can be done either on a task’s name, task’s due\_date or via the tag.

Views

1. Views are done via React, and the React-Bootstrap UI library.