

CVWO Mid-assignment Submission

User Requirements

In our day to day lives, we have many things we want to do at any point in time, so many that it can be hard to keep track of them all. This is what makes a task management application useful. A user of the application should be able to offload their to-dos on the application and the application should remember them for the user. Then, the user may return to the application at a later time to view the to-dos recorded. The user should also be able to edit the to-dos and delete them when no longer needed.

Since the number of to-dos may become quite numerous, it would also be useful to have features to help organise them as well as find to-dos effectively. Each to-do should optionally be given a category. Then, the user should be able to search for to-dos using their names, details, or their categories.

Below is a summary of the features the task management application will have.

- Create new to-dos (consisting of a name, details, and a category)
- Read all stored to-dos
- Update to-dos (edit the to-do itself or change its completion status)
- Delete to-dos
- Search for to-dos

Execution Plan

The to-do app will feature a frontend built with React and a backend using Rails. Rails will be used as a REST API endpoint and the React frontend will read data from there. If possible, the application will be hosted on Heroku.

The front page of the to-do app will feature the to-do list. There will be buttons for adding new to-dos or editing existing to-dos which take the user to another page. There, the user can fill out a form to create or update a to-do. The main page will also have delete buttons to delete to-dos.

There will be a search bar at the top of the page for users to enter search terms and look for specific to-dos. Only the to-dos matching the search query will be displayed.

To hide less important information and make the to-do list less crowded, the rows for each to-do will be able to expand or collapse. When collapsed, the row will show the to-do's name, category, and completion status. When expanded, the to-do's details as well as the edit and delete buttons will be revealed. This also helps ensure the user does not click on the wrong edit/delete button by mistake since it will be easier to see which to-do they are acting on.

A mock-up of the main page of the to-do app is shown below.

To-do	Category	
<input checked="" type="checkbox"/> Borrow books from Central Library for research	NUS	+
<input type="checkbox"/> Print stuff CS1234 cheat sheet for midterms and MA3141 assignment	NUS	- <input type="button" value="Edit"/> <input type="button" value="Delete"/>
<input type="checkbox"/> Watch that Netflix series everyone's been talking about	Recreation	+
<input type="checkbox"/> Find Waldo	Miscellaneous	+

And the following is the planned hierarchy for the React components in the main page.

- App
 - SearchBar
 - SearchButton
 - AddToDoButton
 - TodoTable
 - TodoTableHeader
 - TodoTableRow
 - TodoCheckmark
 - TodoRowExpandButton
 - TodoEditButton
 - TodoDeleteButton

For the backend, a single rails model will be generated to represent a Todo. A Todo consists of a name (string), details (text), a category (string), and a done status (boolean). The details and category are optional. Rails will be configured as an API for the frontend to request for the to-dos data.