

Mid-Assignment Submission

Guo Yichao

A0204731R

Preparation:

1. Installed ubuntu(Windows OS)
2. Node.js and npm
3. Yarn
4. Installed ruby on rails with rbenv
5. Configuring git
6. Installed mysql/postgresql
7. Learned to upload project onto github.com
8. Tried example case in rails guide
9. Read up rails api
10. Tried tic tac toe example in react official site guide
11. Followed and tried another online react on rails recipe example

Process/Elements of task:

1. Create new application by command
(\$ rails new taskManager --webpack==react -d postgresql)
2. Run server by command (\$rails server)
3. Install front-end dependencies
4. Config react
5. Generate controller, model (\$rails g)
6. Migration
7. Define actions in controller in controller.rb
8. Define task as a resource in model.rb
9. Create view files and design webpages html.erb
10. Create links linking different webpages

Basic Use Cases and Execution plan for task 1 and task 2

Task 1 Basic use cases:

1. Main page list all tasks according to date need to accomplish desc. So the most urgent task show up in front.
2. Create task (def create, def new)
3. Show task (def show)
4. Edit task (def update)
5. Delete task (def destroy)

Task 2 Basic use cases:

1. All functionality of task 1
2. In Main page listing all tasks, include a select tag box which can show tasks under one certain category only
3. In create item page, besides necessary textfields and textareas, include one select option box which can be used to categorize tasks (e.g. work, family, etc...)

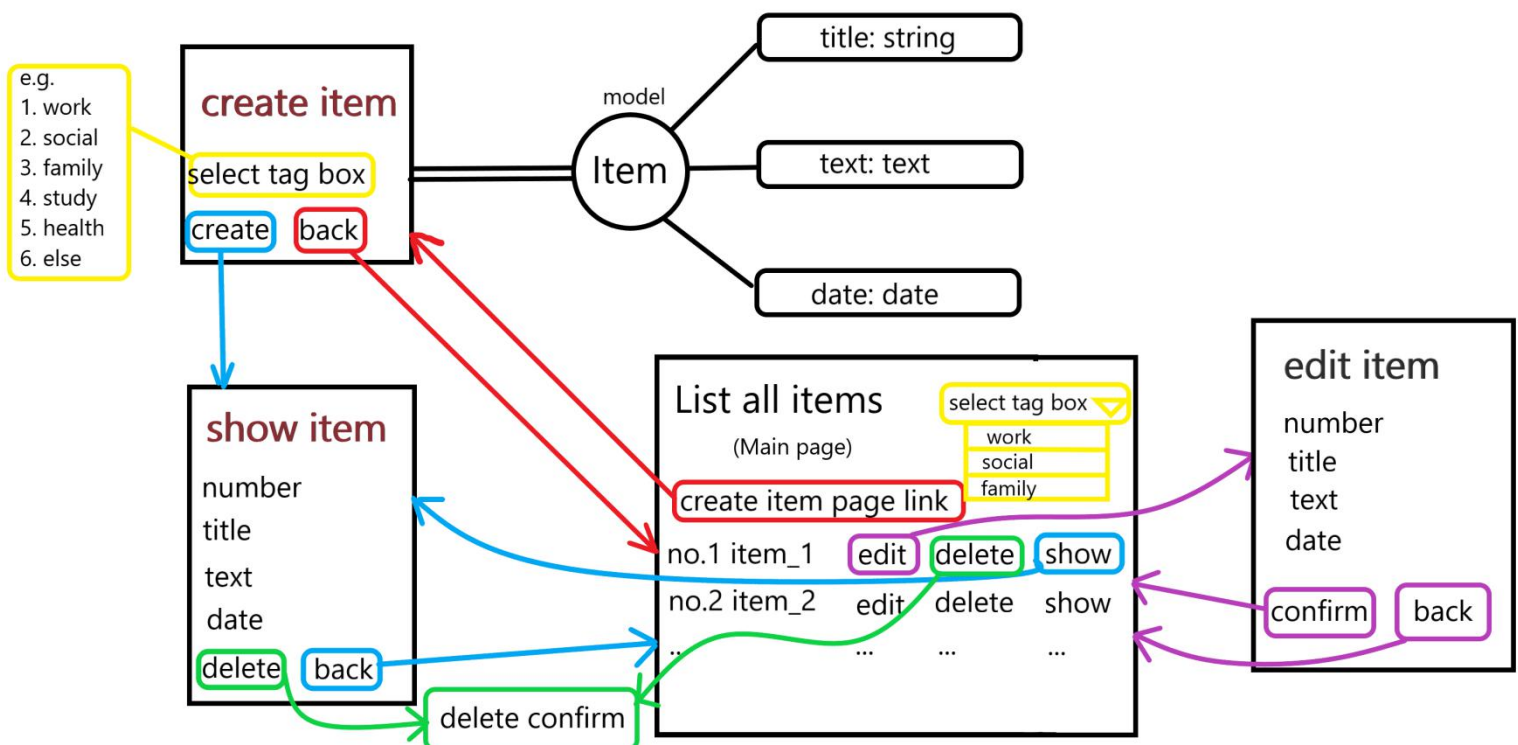
Execution plan:

Model: item

Views: index, new, show, update

Controller: new, create, update, destroy, find

Frontend: react



Possible Improvements of the application:

1. Categorize the tasks into two types: accomplished and unaccomplished. This can be achieved by setting an additional boolean property. The main page contains two tags, one tag for accomplished and another for unaccomplished.
2. Further categorize the tasks into two types: past tasks and future tasks. All outdated tasks or accomplished tasks are colored gray and upcoming tasks are colored green.

Problems:

1. Unfamiliar with React, still need read up more and practice more so as to gain a much deeper understanding in order to integrate react into rails and build pleasant and delicate user interface.
2. Unfamiliarity with linux OS causes learning of react on rails to slow down. But on the other hand, I also get to gradually picking up linux skills along the way. Such as useful commands and some common linux applications such as vim.