

# Todo Manager Final Report

## User-guide

### How to use

**This todo manager is designed for team cooperation. Users can create team, join other's team and share tasks within team members.**

**For assignment, team act somehow like the category required in Task 2.**

### Sign up

**Whenever not logged in, you will see "sign up" in home page. Click on it, and you will be redirect to the sign up page. Fill in your username, email(must be unique), and your password(min 6 digits), and click on create my account.**

### Log in

**In the log in page(top-right corner of your web view), you can log in with your password and email. Please note that your password were never stored in the database, so you do not need to worry about data leakage. If you choose "remember me on this computer option", you will be permanently remembered until you manually log out, otherwise your session would end when you close your browser.**

### All teams

**All teams page is where all teams on the website is listed. You can create your own team with the button bellow or join other teams with the invitation code of those teams. The invitation code is created by the team owner when the team was created.**

### My teams

**My teams page is where you can see all teams you joined. Team details link and delete option is provided here if you are the admin of the team.**

### My tasks

**In my tasks page, you will see all tasks from your personal task list, as well as all team task lists you joined. The column "From" indicates which task list(a personal one or a team one) this task is from.**

### Team view & Task view

**These are where you can see details of the team/task and corresponding edit button.**

### Accessibility

**As a logged in user, you are only allowed to view the following pages: all users, all teams, my teams, my tasks, task pages which belongs to you or your teams, team pages which you joined. You are only allow to edit/delete the following: tasks from your personal list or your team, teams you created, your personal profile.**

## Code analysis

### Things need to be improved

#### Modularity and abstraction

Since it's my first time do web programming, I find it very hard to do abstraction before I write my code. For example, I did not use any partial views in new and edit view, and I also keep using a statement to check whether a user is team admin or not instead of using helpers.

#### Data validation

Some of my data validation is done in model and some are in controller. I think it is not a good design because in this way the data stored in the database may not be "secure" or "full" when some implementations of some controllers went wrong. All validations should be done in model.

#### Access control

Access control should be done using `before_action` in controllers. However I did some of these access control inside controller actions.

#### Documentation and comments

I did not write any documents (some but when I actually code this website I found my design totally useless) or comments in my program. I think it's because I am not familiar with ruby and rails yet, so I also need to try a lot when I code. It would be improved in my next project.

#### Bad error and success notice

I followed rails tutorial first that I used `object.errors.messages` to get the error messages sent back from the model. However some of my validations are inside controllers. So I also used flash to notice the user. It is better to combine this two together in one error box or something like that.

#### Poor user interaction

The user interface is not very friendly that it lacks necessary instructions (e.g. password length). And also it is not very beautiful.

## Code reference

RailsTutorial Sample\_app\_4<sup>th</sup>\_ed written by Michael Hartl

Link: [https://bitbucket.org/railstutorial/sample\\_app\\_4th\\_ed](https://bitbucket.org/railstutorial/sample_app_4th_ed)

Built on top of branch: updating-users

## My work

**Other than** User; static\_pages controller, views; sessions, and shared layouts are my independent work. Modification was done to those parts above to fit them into my implementation.

## Learning journey(reflection)

I started to learn rails when I went back China and start my vacation. I was busy

travelling and doing other stuffs, so I did not really spent much time on learning rails. I started with reading ruby on rails guides and then with a book called ruby on rails tutorial. I started following the book to build sample apps. It was really really hard at the beginning since I only know how to follow the steps but nothing else. After I finish reading the book, I tried to develop my own todo manager. The steep learning curve of rails made my first step out extremely hard. However it became much better after I finish writing tasks resource. Rails is hard because the rails guides is not clear enough to write code. I need to google a lot to clarify my doubts. To conclude, it was hard to learn rails at the beginning, tiring at the middle, and interesting now.

Appendix:

Test account: [dukesun99@icloud.com](mailto:dukesun99@icloud.com) (with some example data created)

Test password: 990520

Link: <https://obscure-retreat-16228.herokuapp.com/>

Development & Test environment: Windows 10

Production deployed on Heroku