Legacy CodeReport

Project Name: Genetic Dataset Curation Website

Platform: Ruby on Rails
Tools: Github Heroku RSpec&FactoryGirl
Main Goal: Provide an online website platform for users to submit these answers to the questionnaires about the relationships between disease name and gene NO. And support administration for administrators on the results.
Function Description:
The root-URI page is the log-in page. It offers options for creating new user, which will lead to a new user setting page. The top layer of the website offers some useful panels for quick access. They are Home, Help and Log in tags. The bottom layer of the website offers some development information, which are links to the information about the website developers.
After correctly login or submitting new user request, the website will be directed to user profile page. And there are two situations. For users, they will only see the results of the questions they've already submitted their answers, namely the "submissions". A button "Start" is offered to the users to direct to the disease page, where they can make their choices on relevance and reason and submit the answers. For administrators, they can not only do what the users could do, but look at the users' statistics, answers distribution as well. Besides, under every circumstance the DataSet value appears, there will be a hyperlink to an external website contains the information about this DataSet.

Learning Codes:
Scheme of the tables stored in database: 3 tables
Users: name email created_at updated_at password_digest remember_digest admin accuracy
Disease: disease accession created_at updated_at related unrelated closed
Submissions: disease_id user_id is_related created_at updated_at reason
Top View of the Website Design: 3 models(User, Disease, Submission).
Details: Use sessioncontroller to control the root-URI (the login page) and check and store the user login information in session[], and finally direct to the User model.
The User model has two controllers-the admincontroller and the usercontroller. These controllers take care of the Routes concerned with the users. There are ways in this model to connect the other two models.
The Disease model has only one diseasecontroller. This controller's work is mainly getting data from the database and support a user interface for users to submit answers.
The Submission model doesn't have code in the controller. This model belongs to the disease and user. Inside this model, the work is mainly interacting the submission results of a specific disease and user with the database records.
Previous stories improvement:
Make the website much easier to maintain.