Boot Camp Homework 4: Rails MVC Model

For this homework, you will create simple model classes in Rails, write tests to run them, and practice using the basic version control features of Git.

You will do this homework as a team; however, each member of your team will be responsible for the completion of a particular task.

Like the last homework, each team member will choose one task from the list below to complete. All team members must do a different task. If your team has only four members, then ignore Task 5. IMPORTANT: Be sure to add your university username (e.g., mine is "sdflming") in a comment at the top of your model class file. (Otherwise, how will we know which work is yours?)

How to submit your squad's work

Before you can submit, all squad members must have merged their code into the master branch and pushed the updates to GitHub. If a squad member does not complete his/her work on time, you may submit without his/her contribution.

To submit your team's work, you must "tag" the current commit in the master branch:

```
$ git tag -a hw4v1 -m 'Tagged Homework 4 submission (version 1)'
$ git push origin --tags
```

To grade your work, I will check out the appropriate tag, and run it on my machine.

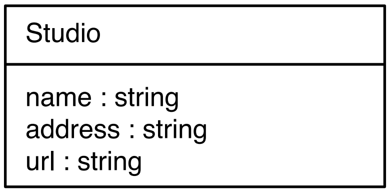
Note that if for some reason you need to update your submission, simply repeat the tagging process, but increment the version number (e.g., hw4v2, hw4v3, hw4v4, etc.).

The Tasks

The tasks are given on the following pages. For each of the following tasks:

- Create a model class as per the given UML class diagram.
- Add attribute validations to the class as specified. Here is a page that may help you:
    - http://guides.rubyonrails.org/v4.2.0/active_record_validations.html
- Write unit tests as follows. For each attribute write one test for a valid value of the attribute. For each validation of the attribute write one test that violates the validation rule. For example, if a string attribute has a length validation and a presence validation, you will have 3 tests for that attribute: one with a valid string, one with a string of invalid length, and one with an empty string. In the case of an "allow_blank" validation, include a test for a blank value. Here are couple pages that may help you:
    - http://guides.rubyonrails.org/v4.2.0/active_record_basics.html#crud-reading-and-writing-data
    - http://guides.rubyonrails.org/v4.2.0/testing.html

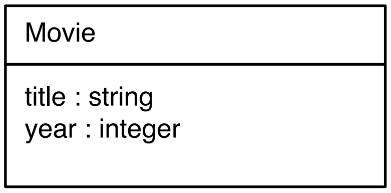
Task 1
Model class to create:



Validations:

- Name must be present ("presence").
- Allow address and URL to be blank ("allow_blank").
- Name must be 50 characters or less.
- Address must 100 characters or less.
- URL must be 80 characters or less.

Task 2
Model class to create:



Validations:

- Title and year must be present ("presence").
- Title must be 100 characters or less.
- Year must have a value of 1890 or more.

Task 3
Model class to create:

Theater
name : string address : string phone : string

Validations:

- Name must be present (“presence”).
- Allow address and phone to be blank (“allow_blank”).
- Name must be 50 characters or less.
- Address must 100 characters or less.
- Phone must be 20 characters or less.

Task 4
Model class to create:

Review
stars : integer title : string commentary : string

Validations:

- Stars, title, and commentary must all be present (“presence”).
- Stars must be a value from 0 to 5.
- Title must 100 characters or less.
- Commentary must be 10,000 characters or less.

Task 5
Model class to create:

Person
first_name : string last_name : string dob : date

Validations:

- Last name and DOB (short for date of birth) must be present (“presence”).
- Allow first name to be blank (“allow_blank”).
- First name and last name must be 50 characters or less.
- DOB must be a date later than Jan 1, 1890 and before the current date.