

Homework 4: 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.

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 three members, then ignore Task 4.

IMPORTANT: Be sure to add your university username (e.g., mine is “sdfming”) in a comment at the top of your model class file. (Otherwise, how will we know which work is yours?)

How to submit your team’s work

Before you can submit, all team members must have merged their code into the master branch and pushed the updates to GitHub. If a team 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.
 - Use `rails g scaffold ...`
- Add attribute validations to the class as specified. Here is a page to help you:
 - http://guides.rubyonrails.org/v5.0.0.1/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 to help you:

- [http://guides.rubyonrails.org/v5.0.0.1/active_record_basics.html - crud-reading-and-writing-data](http://guides.rubyonrails.org/v5.0.0.1/active_record_basics.html-crud-reading-and-writing-data)
- <http://guides.rubyonrails.org/v5.0.0.1/testing.html>

Task 1

Model class to create:

Actor
name : string gender : string homepage : string age : integer

Validations:

- length
 - **name** can be no longer than 60 characters.
- format
 - **homepage** must follow the general format of a basic HTTP or HTTPS URL.
- inclusion
 - **gender** must be "male", "female", or "non-binary category".
- uniqueness
 - **name** must be unique.
- presence
 - **name** and **gender** must have values.
- allow_blank
 - **age** and **homepage** may be blank.
- numericality
 - **age** must be at least 2 (babies must be at least 2 to act) and no greater than 125 (no human has ever been confirmed to have lived that long).

Task 2

Model class to create:

Film
title : string genre : string release_year : integer imdb_url : string

Validations:

- length
 - **title** can be no longer than 120 characters.
- format
 - **imdb_url** must follow the general format of a HTTP URL for the imdb.com domain.
- inclusion
 - **genre** must be one of the literary genres listed on wikipedia (https://en.wikipedia.org/wiki/List_of_genres): "Adventure", "Action", "Comedy", etc.
- uniqueness
 - **imdb_url** must be unique.
- presence
 - **title** and **genre** must have values.
- allow_blank
 - **release_year** and **imdb_url** may be blank.
- numericality
 - **release_year** must be at least 1889 (year of first film) and no greater than the current year.

Task 3

Model class to create:

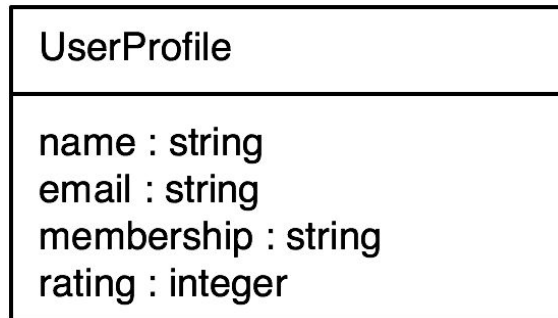
Reviewer
handle : string phone : string status : string years_active : integer

Validations:

- length
 - **handle** can be no longer than 16 characters.
- format
 - **phone** must follow the general format of a 10-digit telephone number (e.g., "123-456-7890").
- inclusion
 - **status** must be "top", "second tier", or "amateur".
- uniqueness
 - **handle** must be unique.
- presence
 - **handle** and **phone** must have values.
- allow_blank
 - **status** and **years_active** may be blank.
- numericality
 - **years_active** must be at least 1 and no greater than 125 (no human has ever been confirmed to have lived that long).

Task 4

Model class to create:



Validations:

- length
 - **name** can be no longer than 30 characters.
- format
 - **email** must follow the general format of an email address.
- inclusion
 - **membership** must be "platinum", "gold", "silver", or "bronze".
- uniqueness
 - **name** must be unique.
- presence
 - **name** and **email** must have values.
- allow_blank
 - **membership** and **rating** may be blank.
- numericality
 - **rating** must be at least 1 and no greater than 5.