WEBD 236 Lab 4

If you use an external source (i.e. a web-page, the required textbook, or an additional book) to help you answer the questions, be sure to cite that source. You should probably always be citing a source.

Problem

This is the last modification of your forum assignment. Starting from your previous lab assignment (the one with answers, clickable tag names, and data validation), you will be adding file attachments to questions, mini-markdown, and role-based access controls. You should add the following features

- Role based access controls. There should be three roles: User, Editor, and Administrator. The permissions associated with these roles are as follows:
    - User: “answer_create.” In other words, a registered user can post answers.
    - Editors: “answer_delete,” “question_create,” “question_delete,” and “question_edit.” In other words, an editor can delete anyone’s answers or questions, and can create and edit questions.
    - Administrators: “admin_page,” “user_delete,” “user_edit.” In other words, administrators can access the administrative functions page (changing permissions and group membership), and delete or edit any user.

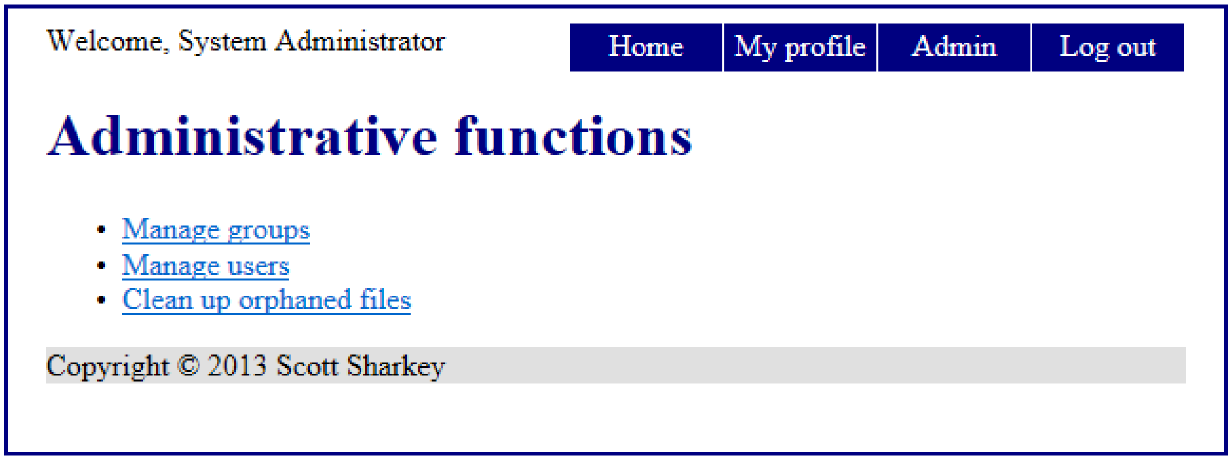
  You should use these permissions to enforce access to certain controllers and to display certain kinds of links. However, any user can manage “their own stuff.” This means that the edit user controller should work for administrators and the user himself (to change e-mail addresses, passwords, etc.) For example, a regular user can delete his own answer.

  This is an exercise in integration. The RBAC library was already demonstrated as part of the course, and is available for download, but it works in a different context. You will need to integrate this code into your previous lab assignment. Your solution should automatically generate the right permissions and store them in the database. Furthermore, you should have three users already set up (one administrator, one editor, and one user) for testing your solution. The administrator account should be a member of all three roles, the editor should be a member of editors and users, and the user should just be a member of users. You should provide the e-mail addresses and passwords for these users when you submit your assignment.

- Mini-markdown. While not nearly a complete markdown implementation, the simple markdown function presented as an example in the course should also be integrated. In particular, answers and forum questions should support the mini-markdown syntax. The answers and questions should be stored in the database in markdown and only converted to HTML as the page is being rendered through the view. Be sure to escape any HTML so that injection-based attacks aren’t possible.

- Attachments. Each question can now have one or more file attachments associated with it. Again, this is an exercise in integration as there were examples presented in class for how to handle file attachments. Only those users with the "question_edit" permission should be able to upload or delete file attachments. However, anybody (even an unregistered user) should be able to download and view the attachments.

The following screen shots should help you to understand more fully. A logged in administrator can access the administrative functions page, manage groups, users, and files. This should work just like the examples from class:



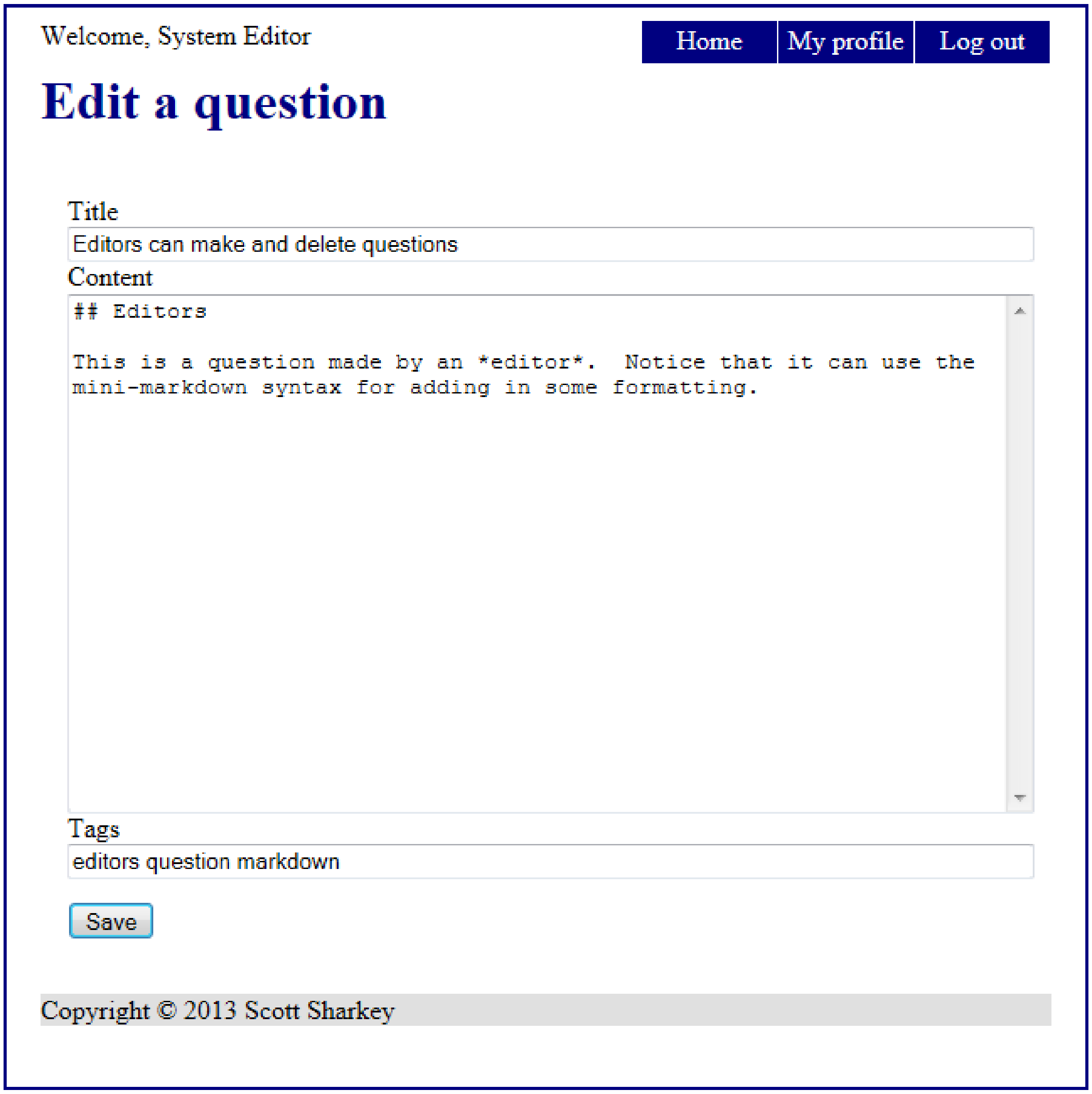
Welcome, System Administrator

Administrative functions

- Manage groups
- Manage users
- Clean up orphaned files

Copyright © 2013 Scott Sharkey

An editor can add and edit posts. Notice the use of mini-markdown in the example below:



Edit a question

Title

Editors can make and delete questions

Content

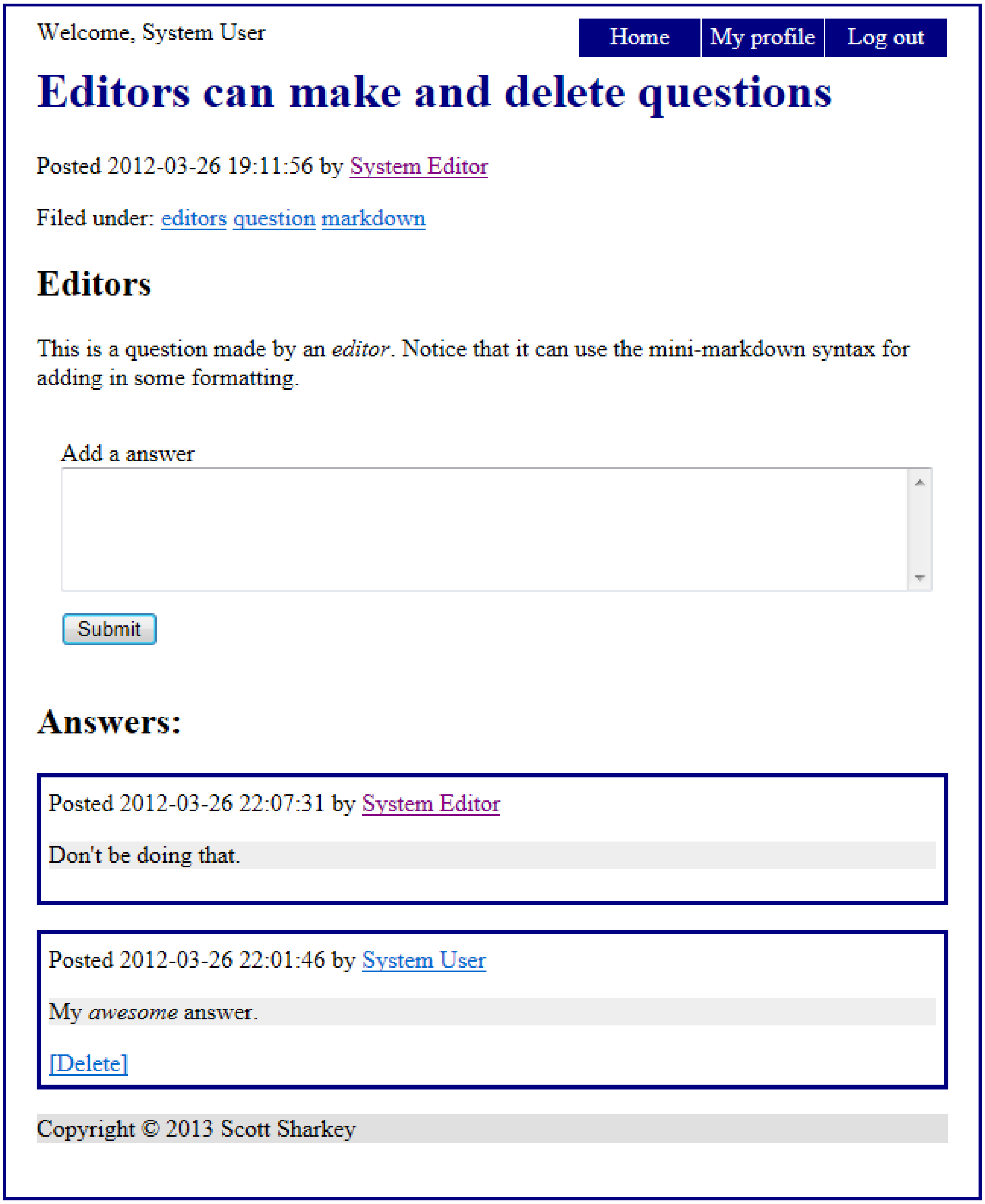
```
## Editors

This is a question made by an *editor*.  Notice that it can use the
mini-markdown syntax for adding in some formatting.
```

Tags

editors question markdown

Copyright © 2013 Scott Sharkey

Users can add answers (and delete their own answers):



Welcome, System User

Home | My profile | Log out

Editors can make and delete questions

Posted 2012-03-26 19:11:56 by System Editor

Filed under: editors question markdown

Editors

This is a question made by an *editor*. Notice that it can use the mini-markdown syntax for adding in some formatting.

Add a answer

Submit

Answers:

Posted 2012-03-26 22:07:31 by System Editor

Don't be doing that.

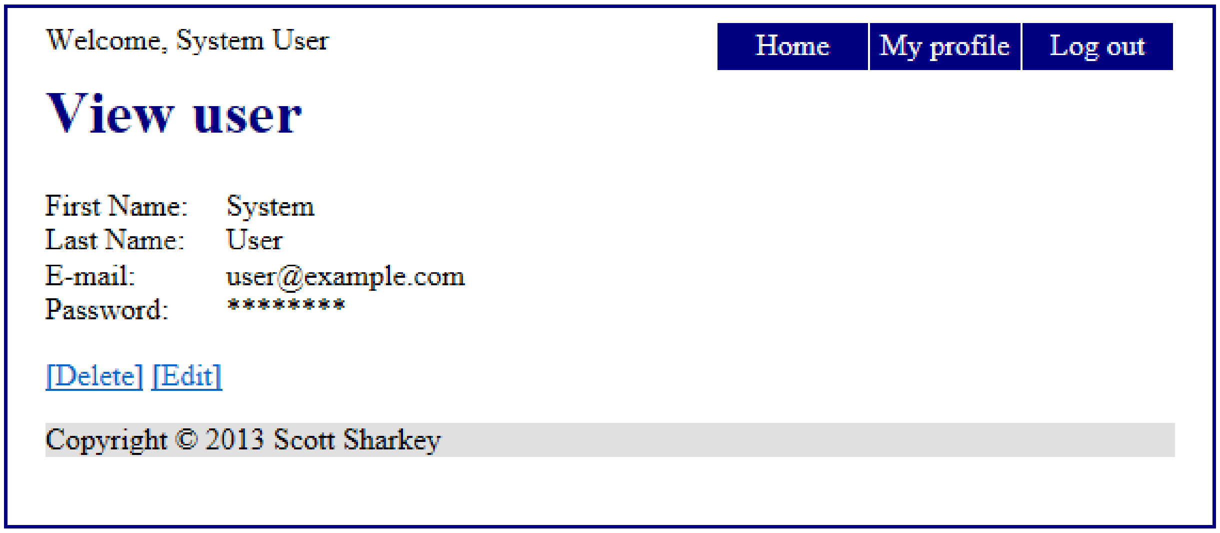
Posted 2012-03-26 22:01:46 by System User

My *awesome* answer.

[Delete]

Copyright © 2013 Scott Sharkey

Of course, users can always view, edit, and delete their accounts:



Welcome, System User

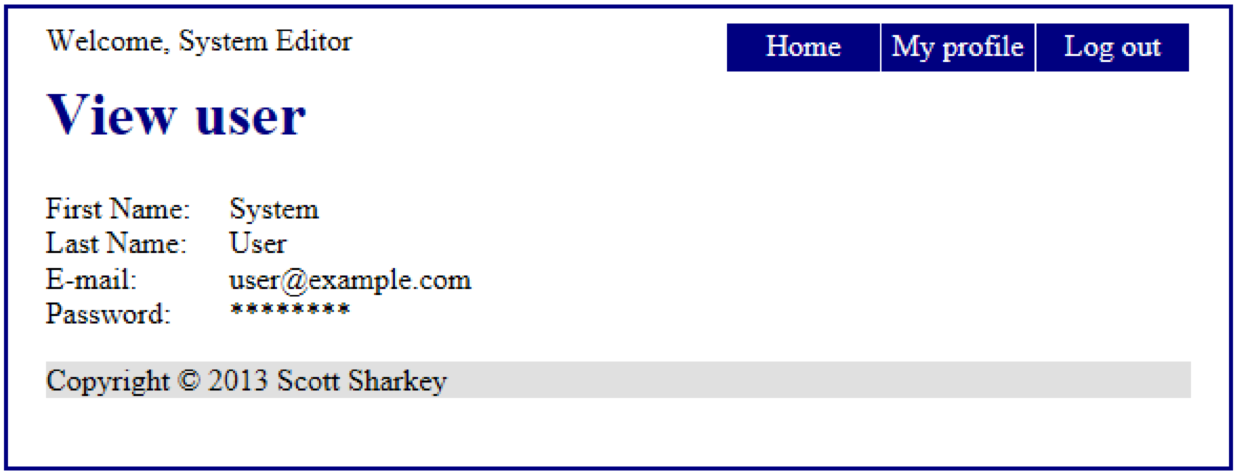
View user

First Name: System
Last Name: User
E-mail: user@example.com
Password: ********

[Delete] [Edit]

Copyright © 2013 Scott Sharkey

An administrator can edit or delete any user, but editors can only view another's profile:



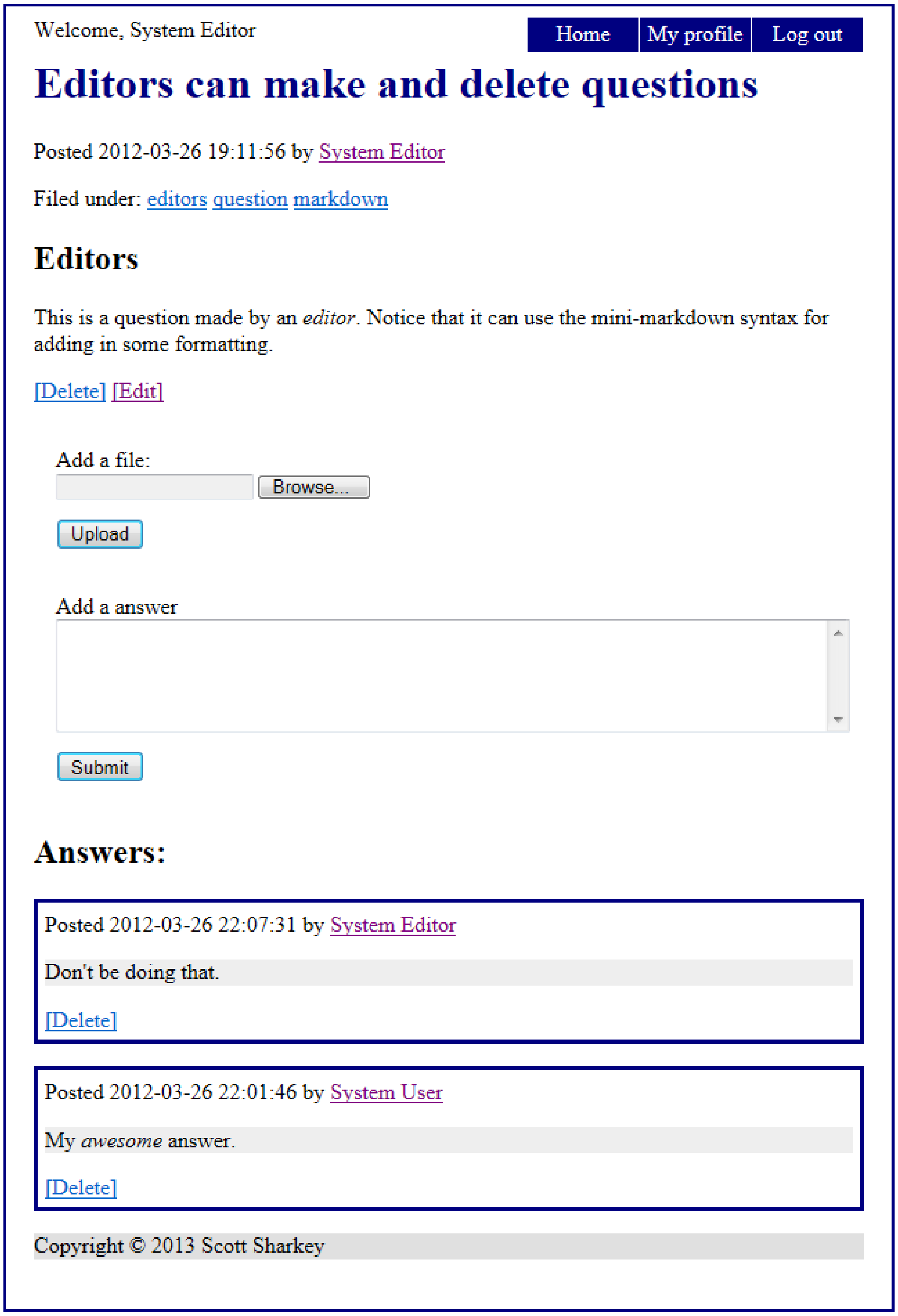
Welcome, System Editor

View user

First Name: System
Last Name: User
E-mail: user@example.com
Password: ********

Copyright © 2013 Scott Sharkey

On the other hand, editors can delete any question or answer. Notice as well below that an editor can attach a file to any question (from the view question screen):


Welcome, System Editor

Editors can make and delete questions

Posted 2012-03-26 19:11:56 by System Editor

Filed under: editors question markdown

Editors

This is a question made by an *editor*. Notice that it can use the mini-markdown syntax for adding in some formatting.

[Delete] [Edit]

Add a file:

Add a answer

Answers:

Posted 2012-03-26 22:07:31 by System Editor

Don't be doing that.

[Delete]

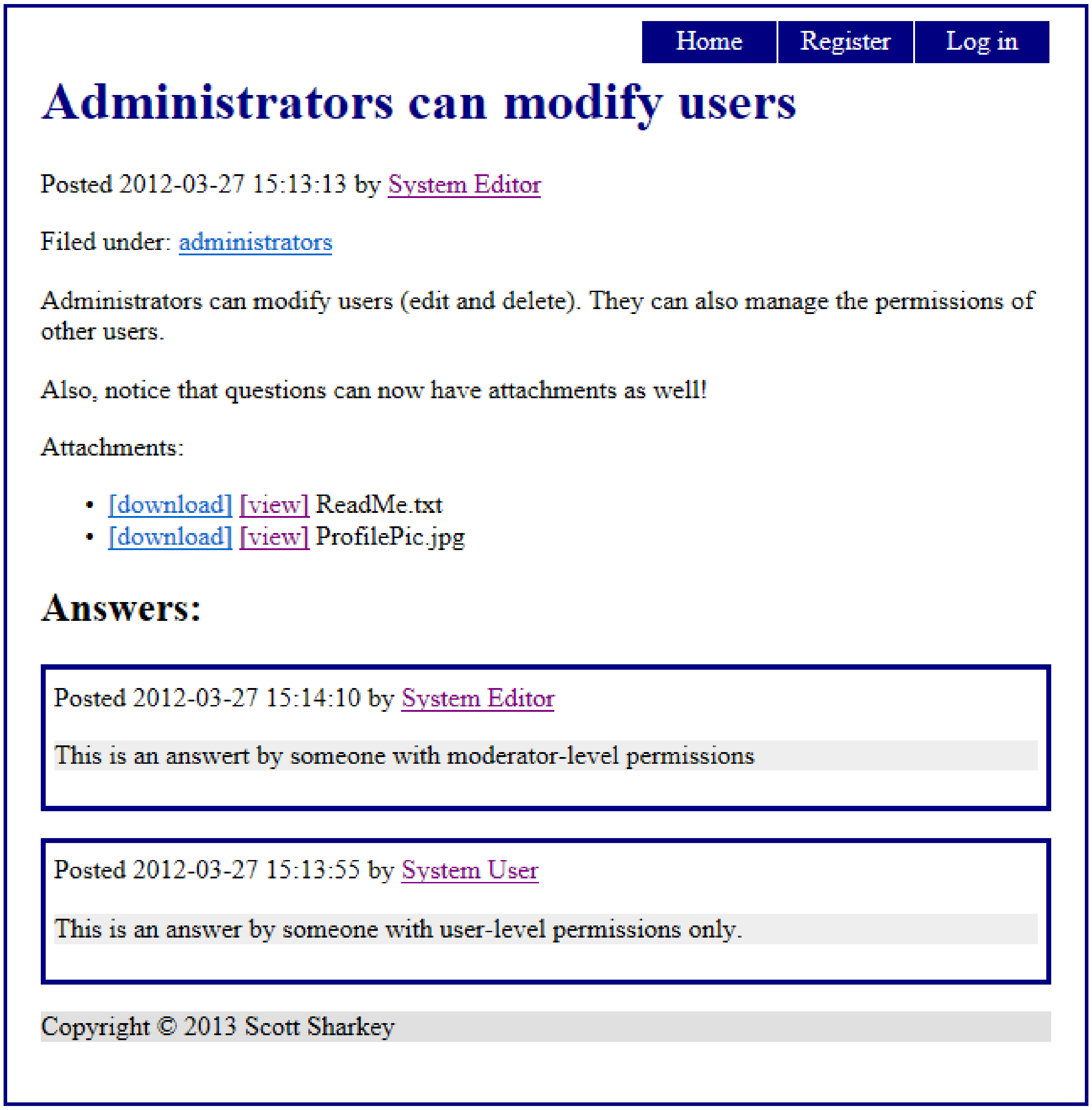
Posted 2012-03-26 22:01:46 by System User

My *awesome* answer.

[Delete]

Copyright © 2013 Scott Sharkey

Regular users (even unregistered ones as shown below) can download and view attachments. Notice that for users that are not logged in, the answer capability isn't available:



Administrators can modify users

Posted 2012-03-27 15:13:13 by System Editor

Filed under: administrators

Administrators can modify users (edit and delete). They can also manage the permissions of other users.

Also, notice that questions can now have attachments as well!

Attachments:

- [download] [view] ReadMe.txt
- [download] [view] ProfilePic.jpg

Answers:

Posted 2012-03-27 15:14:10 by System Editor

This is an answert by someone with moderator-level permissions

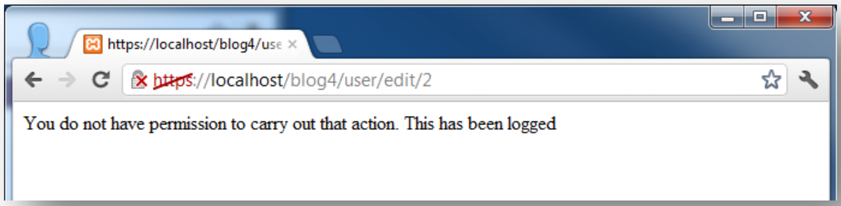
Posted 2012-03-27 15:13:55 by System User

This is an answer by someone with user-level permissions only.

Copyright © 2013 Scott Sharkey

Be sure to maintain referential integrity on everything in the database. Orphaned files should be cleaned up through the administrative interface (see the earlier screen shot).

Finally, you should prevent any URL fishing attacks by protecting your controllers (in other words, it's not sufficient to not display links in the UI, you should also prevent unauthorized actions typed directly into the address bar of the browser). All unauthorized accesses should be logged:


https://localhost/blog4/use
~~https~~://localhost/blog4/user/edit/2

```
You do not have permission to carry out that action. This has been logged
```

Basic Requirements:

- Add in RBAC for Users, Editors, and Administrators (with permissions as defined previously).
- Allow questions and answers to have mini-markdown syntax (but prevent HTML injection).
- Allow file-uploads, downloads and viewing on questions.
- Only display links in the view for operations permitted to that user.
- Prevent access to operations via URL fishing. Log all unauthorized access attempts.
- Maintain referential integrity on all tables. Orphaned files should be cleaned up via a link in the administrative interface.
- All lab 3 requirements should still also be met.
- ***If you are a WEBD student, update your personal portfolio using this project!***

Bonus

You can implement the "forgot a password" functionality discussed in week 9 for an extra 10 points.

Helpful Hints

- See the RBAC, markdown, and file-upload examples posted on the supplemental web site. This entire lab is to integrate these features into your forum project.
- It would be a very, very good idea to have your initial database auto-generated. This would involve the creation of tables if they did not exist, inserting all permissions, groups, and three initial users as well. See the Todo file uploads application in Week 11 for how this can be done.
- Logging can really help to debug.
- You should use SQLite (not MySQL) for this project.
- Make sure that your project works on any server on any directory. In other words, you should never hard-code a URL with the name or IP address of your machine. We won't be using your machine when we test it. Also, you should not hard-code a directory name in your application. It should run as http://localhost/forum/index or as http://localhost/myforum/index or any other directory URL.
- Use the MVC framework developed in class. This will help, since the project is fairly large.

Submission instructions

Create a ZIP file of your entire project contents as it is found under your C:\XAMPP\htdocs\webd236\forum-<yourname>\ directory. This ZIP file should contain your database, your PHP scripts, style files, etc. Follow proper coding conventions (indentation, commenting, etc.) that you would have learned in previous courses.

Submit the ZIP file to the dropbox for this assignment in the course.