Ryan Haines

CSC 415

Assignment #4

4/1/16

Use cases:

1. User uploads a photo

* User logs in using credentials
* User selects the upload photo button on the dashboard
* User selects the file string that needs to be uploaded
* User gets a confirmation and the photo should be added to the page

2. User downloads a photo

* User logs in using credentials
* User selects a photo to download
* User selects the download button on the screen
* User gets confirmation that photo has been downloaded

3. User edits a photo

* User logs in using credentials
* User selects a photo to edit
* User selects from a list holding different editing options
* User inserts fields required to edit photo
* User gets confirmation that photo has been edited and should show up on the dashboard

4. User renames a photo

* User logs in using credentials
* User selects the photo to rename
* User selects the rename button and enters a new file name
* User gets confirmation that the file name has been changed

5. User changes the file type of photo

* User logs in using credentials
* User selects photo to convert
* User then selects a convert button and has the option to change the name of the file extension
* User gets confirmation that the file has been converted and will show up on the dashboard

6. User deletes a photo

* User logs in using credentials
* User selects the photo to be deleted
* User selects a delete button
* User gets confirmation that the photo has been deleted and reflect on the dashboard

Project testing:

Integration testing-

Integration testing is a good way to bring together different working parts of code to make sure that they all work together as designed. Using a top down integration approach will most likely be the best approach to testing the application. Since most of the functionality is going to be involved with the dashboard it would be a good idea to start with that and slowly work in other pieces to make sure everything works properly.

Security testing-

Having a login system will create the need to run some security testing. This is necessary because the users information will need to be secure if someone tries to break into the account. One type of security test that I can run is XSS Me to test to see if there are any vulnerabilities within the login system of the application.

Debugging tools-

Since this system is going to have quite a few working parts, it will be necessary to have some debugging tools handy. It will be important to find errors in the early stages of developing this application. One of the debugging tools I can and will use is ruby-debug. It is a command line base debugger that can be combined with the IRB shell to help detect problems with the system.

Functionality Outputs

|  |  |  |  |
| --- | --- | --- | --- |
| **Functionality Tested** | **Inputs** | **Expected Output** | **Actual Output** |
| Upload photo | Photo filename | Confirmation that photo was uploaded |  |
| Delete photo | The photo selected | Confirmation that photo was deleted |  |
| Edit photo | The photo selected and required field for specific editing option | Confirmation that photo has been edited and the photo was added to the dashboard |  |
| Change photo name | The photo selected and the new photo name | Confirmation that the photo name has been changed |  |
| Download a photo | The photo selected | Confirmation that the photo has been downloaded and the file saves to specified location |  |
| Change photo extension | The photo selected and the new file extension | Confirmation that the photo file extension has been changed and the photo with the new extension was added to dashboard |  |