Team Lead
Andrew Andrawos aandrawo@mail.sfsu.edu

Team Members
Scott Penn
An Dao
Anita Zhen
Brandon Tong
Nicholas Stepanov

SW Engineering CSC648/848 Summer 2018
StockOverflow
Team 4
Milestone 4
08/05/2018

1) StockOverflow

StockOverflow is a stock photo website where you can download and upload photos for free. We are using a minimalist design to keep our site easy to use. Our site is designed to be something pleasantly simple which we used focus groups to help accomplish.

Link: https://csc648-stock-overflow.herokuapp.com/

Committed Functions:

- Search bar with categorical search.
- Category images with categorical search.
- Upload capabilities with custom description.
- Individual image pages that lists image information.
- About page with group descriptions.
- Login
- Sign Up
- Admin page with admin database manipulation capabilities.

2) Usability test plan – 2 pages max

Usability testing of Upload Image function:

-Test objectives:

Determine the usability of Upload Image function to an average user without technical skills. This will involve evaluation of the overall effectiveness of our website design, efficiency of implemented functions and individual user satisfaction. The usability test is to be completed by several users who are represented by use cases to form an overall "average user" opinion.

-Test plan:

Problem statement:

After arriving on the main page of the website, please upload a given image to the category Nature with image name "Forest" and description "Wild forest in the morning".

User profile:

The users performing usability testing shall not be experienced programmers and website designers. Perfect User profile would include students, designers and faculty.

Method and test design:

The user will be given a problem statement and upon completion will be asked to evaluate 3 statements by choosing an answer on Lickert scale. The average answers will be computed together with other statistical information such as variance to observe the usability testing results.

Test environment and equipment:

The test shall be performed inside a supported browser standard hardware such as a cheap laptop or stationary computer.

Evaluation measures:

Users' answers will be evaluated statistically by determining the answers to questionnaire. Important measures to consider are: mean values for each question will report overall effectiveness, efficiency and satisfaction of the users; variance results will determine how divided are the average users on the above metrics.

-Questionnaire:

1.The Upload Image page was easy to find and use.

Strongly agree----Agree------Neither agree nor disagree------Disagree-----Strongly disagree

2. The process of uploading an image intuitively conveyed all the required steps.

Strongly agree----Agree------Neither agree nor disagree------Disagree-----Strongly disagree

3. It was simple to upload an image.

Strongly agree----Agree------Neither agree nor disagree------Disagree-----Strongly disagree

4. The process of uploading an image to the website was enjoyable.

Strongly agree----Agree------Neither agree nor disagree------Disagree-----Strongly disagree

QA Test Plan:

-Test Objectives:

We wish to verify the proper implementation of validation for our Upload Image page. Currently users must enter a category and name as required fields, and may only upload images when logged in. We also wish to verify that the above functions work on at least two major Browsers.

-Hardware and Software Specifications:

Hardware Specification:
Computer: MacBook Pro (Retina, 13-inch, Early 2015)
OS: MacOS High Sierra Version 10.13.6

Software Specification:
Browser 1: Mozilla Firefox Version 61.0.1 (64-bit) (Latest)
Browser 2: Google Chrome Version 68.0.3440.84 (Official Build) (64-bit)

-Features to Test:

Form Validation:

In the upload form there are three required fields: Image Name, Category, and the Image File. We will test for the Image Name field to see if this requirement is being enforced.

User Permissions:

Only users and admins are allowed to upload images to Stock Overflow. If a guest attempts to submit the upload form, they should be sent to the login screen. We will test this requirement.

Database Integration:

Once the above two conditions are met, the user can submit the form. This will upload the image to our server's file system and add its data to the database. In the final build, all images will be uploaded as 'Pending' and require admin approval to see on the main site. However, for the QA test we will relax these restrictions and view the image directly after upload.

Test Cases

Test #	Test Title	Test Description	Test Input	Expected Output	Results
1	Attempt Upload with no Image Name Firefox	Make sure that the form does not go through without the required field “Image Name”	Submit upload form without Image Name field filled in.	Browser will prompt user to fill out name field.	PASS
2	Attempt Upload with no Image Name Chrome	Make sure that the form does not go through without the required field “Image Name”	Submit upload form without Image Name field filled in.	Browser will prompt user to fill out name field.	PASS
3	Attempt Upload while not logged in Firefox.	Make sure that guests cannot upload to the site	Submit form with Image Name : “Bear”, Description : “Test”, and Bear.jpg image file while not logged in.	User will be redirected to login page.	PASS
4	Attempt Upload while not logged in Chrome.	Make sure that guests cannot upload to the site	Submit form with Image Name : “Bear”, Description : “Test”, and Bear.jpg image file while not logged in.	User will be redirected to login page.	PASS
5	Search for Upload Firefox	Confirm that the image was uploaded to the site. (Currently test images are uploaded immediately)	Type “Bear” on the main page in the search bar.	3 results, all images of a bear. The last one will display the uploaded image when clicked.	PASS
6	Search for Upload Chrome	Confirm that the image was uploaded to the site.	Type “Bear” on the main page in the search bar.	3 results, all images of a bear. The last one will display the uploaded image when clicked.	PASS

4) Code Review:

Style: The coding style for the back end is based around functionality. We have many functions in order to keep the functionality separate. Every request for data is passed through middleware functions in order to separate responsibility of data management and view rendering.

****Code Review Comments will be in Blue**

```
function validateSearch(req, res, next) {
 req.hasSearched = true  //What is importance of hasSearched?
 req.isValid = false;
 req.message = "";
 var searchTerm = req.query.search;
 if (searchTerm == undefined) {
   searchTerm = "";
   req.hasSearched = false;
 }
 if (searchTerm.length > 50) {
   req.message = "Search term must be no more than 50 characters long. Please search
again with a shorter query.";
   req.isValid = false;  //Might not be necessary since it was already set false
   next();
 }
 var pattern = /^[0-9a-zA-Z\s]*$/;
 if (!searchTerm.match(pattern)) {
   req.message = "Search term must contain only alphanumeric characters or spaces. Please
search again with a valid query."; //would not use “query” to make it more    user friendly
   req.isValid = false; //Might not be necessary since it was already set false
   next();
 }

 req.isValid = true;
 next();
}

// This function is an intermediate function that passes the results of a database query
```

```
// to the renderer. Currently it checks if the category is valid, then runs a query depending
// on the result. In the future, we plan on handling the errors in a better way than ignoring
them.
function search(req, res, next) {
 if (!req.isValid) {
   req.searchResult = "";
   req.searchTerm = "";
   req.category = "";
   next();
 }
//The user's search term
 var searchTerm = req.query.search;
 //The user's selected category
 var category = req.query.category;

//What will displayed from this query in the statement below?
 if (category === undefined || category === "") {
   database.query('SELECT Image, ID FROM Posting WHERE Name ~~* $1', ['%' +
searchTerm + '%'], (err, result) => {
          if (err) {
      req.searchResult = "";
          req.searchTerm = "";
          req.category = "";
          next();
          }

          //The results are parsed as JSON into the image column String that points to a file.
          req.searchResult = result.rows.map(x => String(x.image));
          req.photoID = result.rows.map(x => String(x.id));
          req.searchTerm = searchTerm;
          req.category = "";
          next();
   });
 } else {
//What will displayed from this query in the statement below?
 database.query('SELECT Image, ID FROM Posting WHERE Category = $1 AND Name
~~* $2', [category, '%' + searchTerm + '%'], (err, result) => {
          if (err) {
      req.searchResult = "";
```

```
            req.searchTerm = "";
            req.category = "";
            console.log(err, "search second query")
            next();
            }

            if (result != undefined) {
        req.searchResult = result.rows.map(x => String(x.image));
            req.photoID = result.rows.map(x => String(x.id));
            } else {
        req.searchResult = "";
             }
            req.searchTerm = searchTerm;
            req.category = category;
            next();
    });
  }
}
```

Overall, the code is architecturally sound. There maybe a few unnecessary lines, and also some more lines of comments are needed for code clarity.

5) Self-check on best practices for security – ½ page

Some major assets we are protecting are:

- **User information**
- **Passwords**
- **Categories of pictures stored in the database**
- **Whether image is declined, pending, or approved**

To confirm password validation, we use javascript and whether the password is at a certain length of the requested field. In the password section, we first have user input a password then we have an input as confirm password. There the database will check if the two input field match each other. If the two input fields do not match then it will alert the user. If the user forgot their password, it will redirect the user to reset their password and replace their password in the database.

In validating our search bar input, we check what the user searched if it is in the database. If so, it will show images the user requested and searched for. If the image the user requested is not in our database, the user will be redirected to a page where it will narrow its search content and provide its image in the closest category field.

6) Self-check: Adherence to original Non-functional specs

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).

DONE

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.

ON TRACK

3. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**

4. No more than 50 concurrent users shall be accessing the application at any time

DONE

5. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

ON TRACK

6. The language used shall be English.

DONE

7. Application shall be very easy to use and intuitive.

DONE

8. Application shall render well on mobile devices (UI shall be responsive)

ON TRACK

9. Google analytics shall be added

ON TRACK

10. No e-mail clients shall be allowed

DONE

11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.

DONE

12. Site security: basic best practices shall be applied (as covered in the class)

ON TRACK

13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

DONE

14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2018. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

ON TRACK