SW Engineering CSC648/848 Section 01 Summer 2017

"PixSale"

Local Team
Team Number 04

Manuel Duran jdceren@mail.sfsu.edu
Stephen Josey SFSU Student
Ryan Jung SFSU Student
Youssef Hakkou SFSU Student
Jeremy Tan SFSU Student
Yoezhou Yu SFSU Student

Milestone 4

08/05/2017

Revision History

Version	Action	Approval	Date
1.0	Creation of original Milestone 4 document for submission	Team Lead/Editor	08/04/2017
2.0	Updated Milestone 4 with professor's feedback	Team Lead/Editor	08/07/2017

Content and structure for Milestone 4 document for review:

Product summary

PixSale

1. Guests can browse and search through media without being registered users.
2. Users can browse through media by category.
3. Users are able to register and login to their own accounts.
4. Passwords are encrypted on the database.
5. All user information is stored on the database.
6. Registered users are able to upload photos to the website.
7. Registered users are able to make purchases on the website.
8. Admins are able to remove any media from the database.
9. Admins are able to remove any user from the database.

URL of the system to be tested: http://sfsuse.com/~su17g04/

Usability Test Plan

1. Test Objectives

Finding images fast and easy is an important aspect of PixSale. The following tests focus on finding images that meets exactly what potential customers are looking for and whether or not this function is easy to use. Users can search for photos and videos by typing the category they are looking for on the search bar. Users can also search images with the search engine by just filtering and selecting the category of their choice to find the the images or videos they want.

2. Test Plan

- *System Setup:* Computer that is running a recent version of Safari, Chrome and Firefox.

- *Starting Point:* Homepage of PixSale.

- *Task to be Accomplished:*

    1. Use search engine to find images or videos in database
        a. Find images of animals
        b. Find images of cars

    2. Select images results using categories
        a. Find an images of flowers
        b. Find an image of a car

- *Intended User:* Registered user, SFSU student.

- *Completion Criteria:* User searched or selected an images category and found a photo or video that met their criteria. The returned values are accurate and satisfactory. As a result, users are able to search and find images or a website response at each step.

- *URL of the system to be tested:* http://sfsuse.com/~su17g04/

Questionnaire

Please circle the answer that best describes your feelings to the following statements.

Question 1*: It was easy to navigate through the homepage to find images.*

Strongly Agree Agree Neutral Disagree Strongly Disagree

Comments:

Question 2*: The returned search engine results were accurate/useful.*

Strongly Agree Agree Neutral Disagree Strongly Disagree

Comments:

Question 3*: The filters by category on the site were convenient to find an images/videos.*

Strongly Agree Agree Neutral Disagree Strongly Disagree

Comments:

Question 4*: I would recommend PixSale to friends, family members, and other SFSU students.*

Strongly Agree Agree Neutral Disagree Strongly Disagree

Comments:

QA test plan

Test Objectives:

- To make sure the search functionality returns the correct items

HW and SW setup:

- PixSale does support all modern browser: Chrome, Firefox and Safari. Ensure you have an internet connection and a computer that supports modern browsers.

Feature to be tested:

- Test the search function and filters for each category on all browsers. Make sure to test the functionality, stability and usability. The search feature should return the specific results that were either selected from the category drop-down are typed in the search bar by the user. If the user searches for elephant in the search bar and selects animals from the dropdown category, then it should return an elephant. This category drop-down menu will help narrow the search results.

Test No.	Description	Test input	Expected Output	Pass/Fail
1)	Open the webpage at http://sfsuse.com/~su17g04/	Search: nothing Category: nothing Filter: nothing	Should display all images	PASS Chrome PASS Firefox
2)	Use the category filter and select flowers from the dropdown menu	Search: nothing Category: Flowers Filter: nothing	Should show 1 result of a picture of flower	PASS Chrome PASS Firefox
3)	Use search bar and type random set of numbers	Search: “124132” Category: nothing Filter: nothing	Should display 0 search results	PASS Chrome PASS Firefox
4)	Use the media type filter and select images from the filter	Search: nothing Category: nothing Filter: images	Should display all 7 images	PASS Chrome PASS Firefox
5)	Use search bar and type filter and category.	Search: car Category: cars Filter: images	Should display 3 results of cars	PASS Chrome PASS Firefox

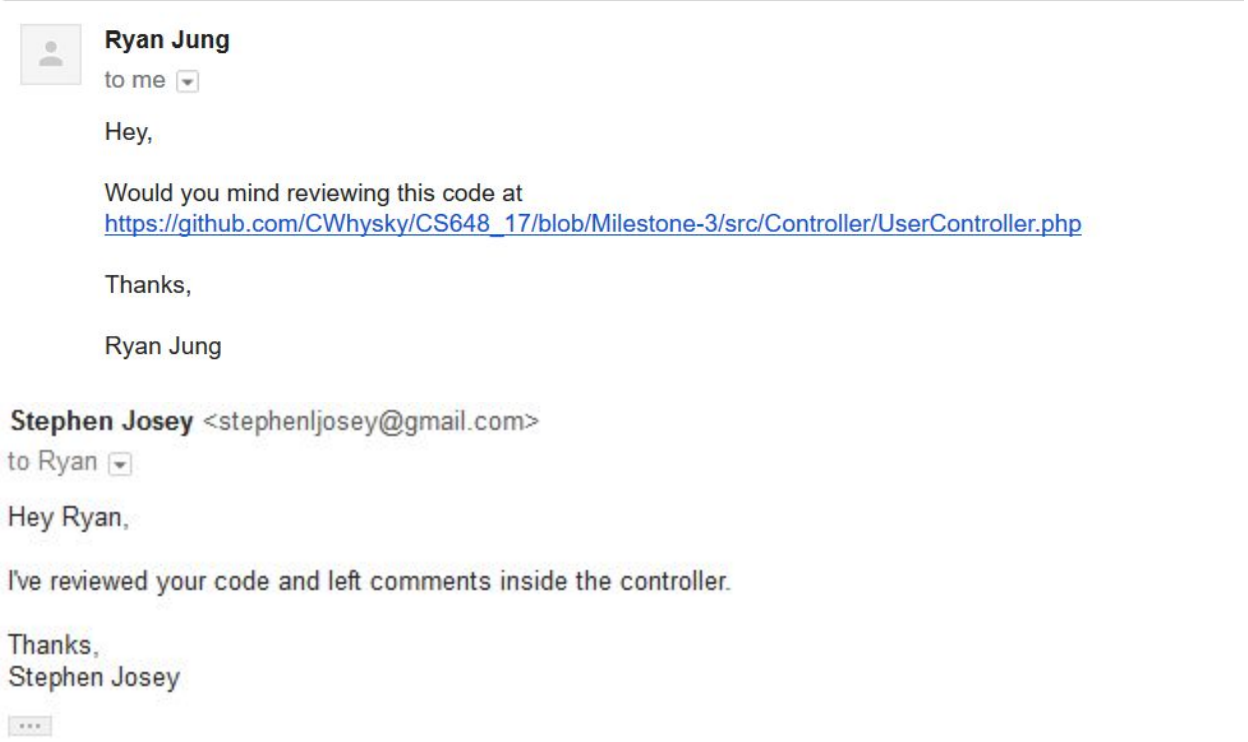
Code Review

By this time you should have chosen a coding style. In the report say what coding style you chose.

Chose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code (or part of it – 2 pages or so MAX) for this function to be peer reviewed, and document this as follows:

Peer review for AppController (screenshots below):

+-----Peer Review------+
Great use of comments to show what is being handled.
Spaces between if statements and () and {} would be better.
Everything is self-explanatory. I understood what the code was doing by skimming through it.



Ryan Jung
to me

Hey,

Would you mind reviewing this code at
https://github.com/CWhysky/CS648_17/blob/Milestone-3/src/Controller/UserController.php

Thanks,

Ryan Jung

Stephen Josey <stephenljosey@gmail.com>
to Ryan

Hey Ryan,

I've reviewed your code and left comments inside the controller.

Thanks,
Stephen Josey

```php
<?php
/*
    Handles the Messages page
*/
namespace App\Controller;

use App\Controller\AppController;
use Cake\ORM\TableRegistry;


/*
    +-----Peer Review------+
    Great use of comments to show what is being handled.
    Spaces between if statements and () and {} would be better.
    Everything is self-explanatory. I understood what the code was doing by skimming through it.

*/

class UserController extends AppController{

    /*
        Handles the landing page
    */
    public function index(){
        //get id of logged in user
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');
            $this->loadModel('Messages');
            $this->loadModel('MediaItems');
            $messages  = $this->Messages->find('all')->where(['Messages.id' => $user_id]);
            $products = $this->MediaItems->find('all', array('fields' =>
                array('id','title','price','file_path')))->where(['MediaItems.registered_user_id' => $user_id]);
            //send data to index.ctp
            $this->set(compact('messages'));
            $this->set(compact('products'));
            $this->set('id', $user_id);
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }



    }
```

```php
    public function profile(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');
            $this->loadModel('RegisteredUsers');
            $registered_user = $this->RegisteredUsers->find('all', array('fields' => array('username','email','first_name','last_name')))
                        ->where(['RegisteredUsers.id' => $user_id])->first();
            $this->set(compact('registered_user'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }

    }

    /*
       Handles the order history page
    */
    public function messages(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');

            $messages= TableRegistry::get('Messages');
            $message = $messages->find();
            $orders = $messages->find()->select([
                                        'Messages.id',
                                        'Messages.receiver',
                                        'Messages.media_items_id'
                                    ])
                                ->contain([
                                        'MediaItems' => [
                                           'fields' => [
                                               'MediaItems.file_path',
                                               'MediaItems.price',
                                               'MediaItems.media_type',
                                               'MediaItems.title'
                                           ]

                                        ]
                                ])
                                ->where(['Messages.sender' => $user_id]);


            $this->set(compact('orders'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }
    }
```

```php
        Handles the order requests page
    */
    public function orderRequests(){
        if($this->Auth->user('id')){
            $user_id = $this->Auth->user('id');

            $messages= TableRegistry::get('Messages');
            $message = $messages->find();
            $order_requests = $messages->find()->select([
                                        'Messages.id',
                                        'Messages.sender',
                                        'Messages.media_items_id'
                                    ])
                                ->contain([
                                        'MediaItems' => [
                                           'fields' => [
                                               'MediaItems.file_path',
                                               'MediaItems.price',
                                               'MediaItems.media_type',
                                               'MediaItems.title'
                                           ]

                                        ]
                                ])
                                ->where(['Messages.receiver' => $user_id]);


            $this->set(compact('order_requests'));
        }else{
            return $this->redirect(['controller' => 'login', 'action' => 'index']);
        }


    }
}
```

Self-check on best practices for security

- **Assets to be protected**:

    1) Registered User's account
    2) Registered User's uploaded media items.

- Password are encrypted on the database using CakePHP's hash function

- CakePHP's find() method prepares SQL statements to prevent SQL injection from occurring. We use CakePHP's find() method with the user's input from the search bar and let CakePHP handle preparing the SQL statements. We follow CakePHP guidelines for database queries and that protects us from most SQL injections. The search bar will also check for string text not to exceed 30 alphanumeric characters. The search bar will also check for empty text in the search bar as well as any unselected category or media type. If no filters are selected then the search will display everything.

Adherence to original Non-functional specs

1. Application shall be developed using class provided LAMP stack. **DONE**
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis. **DONE**
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class. **DONE**
4. Application shall be optimized for standard desktop/laptop browsers, and must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. **DONE**
5. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed. **DONE**
6. Data shall be stored in the MySQL database on the class server in the team's account. **DONE**
7. Application shall be deployed from the team's account on AWS. **DONE**
8. No more than 50 concurrent users shall be accessing the application at any time. **DONE**
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
10. The language used shall be English. **DONE**
11. Application shall be very easy to use and intuitive. No prior training shall be required to use the website. **DONE**
12. Google analytics shall be added. **ON TRACK**
13. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services. **DONE**
14. Pay functionality (how to pay for goods and services) shall not be implemented. **DONE.**
15. Site security: basic best practices shall be applied (as covered in the class) **DONE**
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development **DONE**
17. The website shall prominently display the following text on all pages "SFSU

Software Engineering Project, Spring 2017. For Demonstration Only". (Important so as to not confuse this with a real application). **DONE**