SW Engineering CSC 648/868
Section 01 Summer 2017

Milestone 4

Picturesque

Team 5
Tiffany Ku, Team Lead | tyku@mail.sfsu.edu
Di Mao (Andy), Tech Lead | dmao1@mail.sfsu.edu
Haotian Zhang (Ivan)
Teng Yu
Zhiyang Cheng (Andrew)
Cody Jackson | cjackso5@mail.sfsu.edu
Calvin Ip | ip@mail.sfsu.edu

Last Modified
August 5, 2017
August 8, 2017

1. Product Summary

Our product *Picturesque* is a site that allows users to buy and sell stock images and videos. What makes our site unique is its wide range of categories relevant to today's world and culture along with its ease of use. The ability to message a seller to initiate transactions provides a more intimate and secure network between buyers and sellers.

All Guests
1. Media Search: Shall be able to search for the products they are looking for
2. User Registration: Shall be able to register a new account or log in to existing account
3. Shall be able to purchase a product (as guest)
4. Shall provide email address and password to create an account

Registered Guests
5. Messaging feature: Shall be able to leave message to product's owner initiating transaction.
6. Items for sale form: Shall provide the title, price, description, and preview image for the item they wish to sell.
7. Shall have access to post new product
8. Shall have access to the photos and videos they own and are selling
9. Shall be able to delete/modify their item listings

Administrators
10. Shall be able to visit, browse, and search the site.
11. Shall have access to all users posts
12. Shall be able to delete posts
13. Shall be able to delete accounts

URL: http://sfsuse.com/~su17g05/

2. Usability Test Plan

Purpose:

This test is designed to interpret user interactions with our product and to evaluate the strengths and weaknesses from the perspective of the users. This will evaluate the effectiveness, efficiency, and satisfaction on functions of our site. It creates a better understanding of how users will utilize key functions of our product. The issues that are seen from the user perspective are collected and interpreted to better the product.

Test Objectives:

The user's main objective will be to purchase an item on the site. They are unregistered guests and can choose to contact the seller about the transaction as such or after registering. From this test we would like to interpret the user's usage and categorize them by: perceived usability of the product, actual use of the product, and the time it takes to complete the task.

The user's perceived usability may be different from their actual usage and can create a correlation to actual issues with usage or perhaps lack of overall understanding of technology. It assists with understanding the user's intuition when using computers and the expectations they may have with the usage of our product. Comments will be collected in order to assist us in determining the legitimacy of issues and changes can be made upon these examinations.

The user's actual use of the product will be examined by the number of clicks and page views that are along the path to accomplish their intended objective. This will be monitored through analytical software to measure time from page load, keyboard input, cursor movement, and clickable actions. These analytics will create a correlation for verification of the user's perceived usability to the user's actions of usability.

The test will be timed and data will be collected according to pages viewed as well as time between actions. This will help determine the amount of time the user was on a page without performing an action to correlate it to the perceived usability of the product.
Overall, the purpose of this test is so that the user can easily understand how to purchase an item.

Test plan:

The users will visit sfsuse.com/~su17g05 with the objective of purchasing an item. They will be shown the home page with the options to login and register.

Starting Point:
With a modern web browser (Google Chrome, Safari, or Firefox) the indicated page of the site will be visited.

User Task: Create an account and log in. Find and purchase an image of a beach.

Intended Users:
A creative person seeking to utilize stock images and photos for a project or seeking to sell their original content.

Completion Criteria:
To purchase an item for sale, under the category "beach," as an unregistered guest within a 5 minute time frame upon loading the site. They should not experience any unnecessary pages views that will prevent them from their intended purpose.

URL: http://sfsuse.com/~su17g05/

Questionnaire: https://goo.gl/forms/KxYHcExoeJ2fK7G93

Picturesque Usability Questionnaire

Thank you for participating in our event. We hope you had as much fun testing our site as we had creating it.

We would like to hear your feedback so that we can keep improving our product. Please complete this quick survey and let us know your thoughts.

The scale is a range between strongly disagreeing to strongly agreeing, from 0 to 5 respectively.

* Required

Finding the message seller button was easy. *

	0	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	○	Strongly Agree

The given ability to custom create the transaction message to the seller was satisfactory. *

	0	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	○	Strongly Agree

The overall process to purchase was easy.

	0	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	○	Strongly Agree

Any additional comments regarding the site?

Your answer

SUBMIT

3. QA Test Plan

QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements. This plan is to determine if the feature, creating an account and purchasing an item under the "beach" category as an unregistered guest, is properly functioning.

Test Objectives: To create an account and attempt to purchase an item

Hardware and Software Requirements: Latest version of Chrome, Safari, or Firefox running on Windows 7 or higher or Mac OS 10.9 or higher
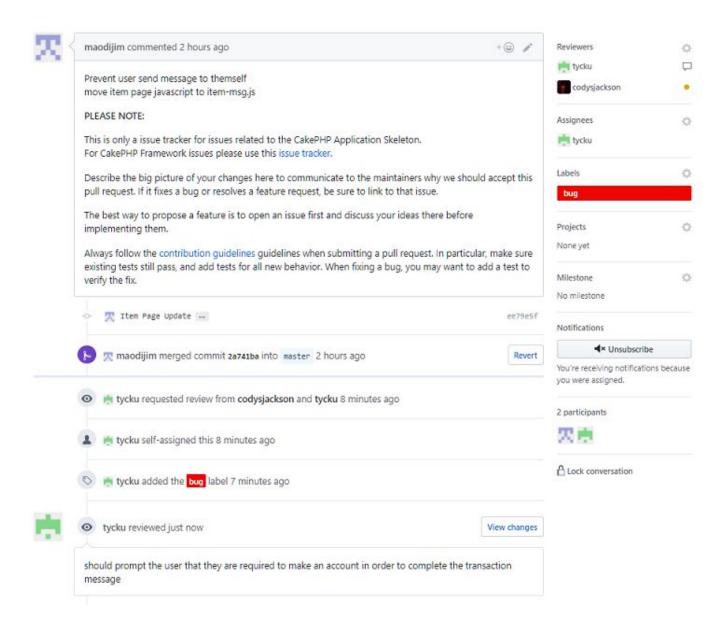
Feature to be Tested: Purchasing an item
QA Test Form: https://goo.gl/forms/7TU3E2Il8VJhInDS2

	Test	Test #1	Test #2	Expected
1	Operating System	Windows	Mac OS	Meets non functional specs for OS
2	Browser	Google Chrome	Safari	Meets non functional specs for OS
3	Creating an account	Successful	Successful	Able to create an account with ease and without conflict
4	Log in to account	Successful	Successful	Able to log in without any issues
5	Found image of beach	Manually searched "beach"	Selected "beach" under categories list	Able to find an available image
6	Click on post for details	Successful - found button easily	Successful - found button easily	Successfully able to view details of post, found the button's location without any issues
7	Click "contact owner"	Successful - found button easily with no issues	Successful - found button easily with no issues	Successfully able to contact owner with ease and with no confusion as to where the button was located
8	Leave message to owner	Successful - found button easily with no issues	Successful - found button easily with no issues	Successfully leave a message to the owner without experiencing any conflicts

4. Code Review

Code review has been a means for the team to effectively manage and organize the development on this project. For both backend and frontend the styles both teams use closely follow the standard use case presented in the documentation of CakePHP.

To ensure code review we used peer review tools on GitHub and reinforced communication using proper commit messages and feedback in person and over our private Slack channel. This helped us better understand each other's work processes and allowed us to be able to properly address changes that needed to be made.

Changes from all commits ▾   Jump to... ▾   +45 −30 ■■■■■

Unified | Split   **Review changes ▾**

36 ■■■■■ src/Template/Item/index.ctp   View 🖥 ⌄

```
          @@ -137,35 +137,11 @@ $this->layout = 'default_no_menu';
137  137          </div>
138  138          <!-- Suggestion -->
139  139          <script>
140       -          if (login !== '') {
141       -              $('#contact-button').click(function () {
142       -                  //e.preventDefault();
143       -                  if ($('#msg-box').val().length > 0) {
144       -                      $.ajax({
145       -                          url:'<?= $this->Url->build(['controller'=>'Messages','action'=>'newMsg'],['fullBase' => true])?>',
146       -                          type: 'POST',
147       -                          data:
148       -                          {
149       -                              sender_id: login,
150       -                              receiver_id:<?= $user->user_id ?>,
151       -                              media_id: <?= $item->media_id ?>,
152       -                              message_content: $('#msg-box').val()
153       -                          },
154       -                              success:function(){
155       -                                  $.notify('Message Sent', {position: 'top left', style: 'bootstrap', className: 'success
156       -                                  socket.emit('messages', {id:<?= $user->user_id ?>, content: 'new message'});
157       -                              }
158       -                      });
159       -                  }
160       -              });
161       -
162       -
163       -          } else {
164       -              $('#contact-button').click(function () {
165       -                  $.notify('Please Log in to send message', {position: 'top left', style: 'bootstrap', className: 'info'});
166       -              });
167       -          }
168       -
169       -      </script>
     140  +      var url = '<?= $this->Url->build(['controller'=>'Messages','action'=>'newMsg'],['fullBase' => true])?>';
     141  +      var receiver = <?= $user->user_id ?>;
     142  +      var media = <?= $item->media_id ?>;
     143  +      var self ='<?php if($this->request->session()->read('Auth.User.user_id') == $item->author_id){echo 'true';}else{echo 'f
     144  +      </script>
     145  +      <?= $this->Html->script('item-msg')?>
```

5. Self-Check on Best Practices for Security

Major assets we are protecting:
1. User media
2. User account data
3. Server security

Best practices:

        Passwords are encrypted in the database by hashing. Users are able to register for an account and login to existing accounts, the account information is saved to the database.

        During registration, all user inputs are being validated and an error message shall appear if the inputs do not match our validation rules. User emails are being checked using JavaScript to ensure the string the user input matches an email format and an error will occur if the input is not an authentic email address. Once user's input passes javascript validation the data is sent to the Model controller and validated again by CakePHP validator to ensure the data received is in correct format. Passwords must be 8-20 characters long with a capital letter and the user is prompted to confirm their password.

        The search bar input is also being checked by using a percent like algorithm to compare the input to the media in our database.

6. Adherence to Original Non-Functional Specs

- Application shall be developed using class provided LAMP stack **DONE**
- 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**
- Application shall be hosted and deployed on Amazon Web Services as specified in the class **DONE**
- 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**
- 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**
- Data shall be stored in the MySQL database on the class server in the team's account **DONE**
- Application shall be deployed from the team's account on AWS **DONE**
- No more than 50 concurrent users shall be accessing the application at any time **DONE**
- Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **DONE**
- The language used shall be English. **DONE**
- Application shall be very easy to use and intuitive. No prior training shall be required to use the website. **DONE**
- Google analytics shall be added **DONE**
- 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**
- Pay functionality (how to pay for goods and services) shall not be implemented. **DONE**
- Site security: basic best practices shall be applied (as covered in the class) **DONE**
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development **DONE**
- 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). **ON TRACK**