SW Engineering CSC648/848 Section 01 Spring 2017

GatorTraders

Milestone 4

Team 7
Local Team
Evan Terry, Team Lead/CTO | eterry@mail.sfsu.edu
Chohee Kim | ckim4@mail.sfsu.edu
Kendrick Kwok | kkwok@mail.sfsu.edu
Brandon Chiong | bchiong@mail.sfsu.edu
Marc Panlilio | mpanlili@mail.sfsu.edu
Ramsay Wong | ramsayw@mail.sfsu.edu

Date Last Modified: 05/04/2017

1. Product Summary

The name of our product is GatorTraders. It is a site that allows San Francisco State students and faculty to sell products and message sellers to initiate transactions. What makes GatorTraders unique is it's ease of use and exclusivity to the San Francisco State community. It provides a more intimate and safeguarded network of buyers and sellers.

All Guests

1. Shall be able to visit, browse, and search the site using the categories provided by the website or typing in text onto the search bar.
2. Shall be able to create an account and login.
3. Shall provide email address and password to create an account.
4. Guests transition to registered students or faculty based on imposed email restriction based on SFSU email address.

Registered Students or Faculty

1. Shall be able to list their products for sale.
2. Shall be able to reply to a posted item for sale.
3. Shall be able to message other students.

2. Usability Test Plan

Usability tests are designed to interpret user interactions with a product and evaluate the strengths and weaknesses in from the perspective of users. It creates a better understanding of how users utilize the key functionalities of the product. The issues that are seen from the user perspective are collected and interpreted to better the product.

Test Objective

The test user's main objective is to post an item on the site. They are unregistered guests and must make an account to post. We would like to interpret the user's usage and categorize them three different areas: perceived usability of the product, actual use of the product, and time to task completion.

The user's perceived usability may differ from 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 their expectations they may have with the usage of a product. These comments will be collected which will be extrapolated and examined to determine legitimacy of issues and radar tickets created upon them exceeding a threshold of concurrence amongst testers.

The users 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 extrapolated according to page viewed as well as time between actions. It 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 is so that the user is able to easily understand how to post an item.

Test Plan

Test users will visit http://sfsuse.com/~sp17g07/gatortraders/web/welcome and with the objective of posting an item. They will be presented with the page to register, login, then be able to post.

Starting point: Using a modern web browser: Google Chorme, Firefox, or Safari and visiting the indicated page

Intended user: A San Francisco State Student or Faculty member with an email ending in @mail.sfsu.edu

Completion Criteria: To post an item for sale within a 5 minute window from loading the site. They should not experience any unnecessary page views from their intended purpose.

URL: http://sfsuse.com/~sp17g07/gatortraders/web/welcome

Questionnaire: https://goo.gl/forms/Y3NO8xgyo4rjk4iS2

3. QA Test Plan

Quality Assurance testing is a way to provide insight if features and processes are working to the intended or sometimes unintended uses cases. This plan is to determine if the feature, to post an item, is properly functioning.

Test Objectives: To create an account and attempt to post 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: Posting an item

QA Test Form: https://goo.gl/forms/o8imX7VZmxP3m2592

	Test	Test #1	Test #2	Test #3
1	Operating System	MacOS	Windows	Linux (Ubuntu, Red Hat, etc.)
2	Browser	Google Chrome (or Chromium)	Firefox	Firefox
3	Create an account	Pass	Pass	Pass
4	Log in to account	Pass	Pass	Pass
5	Click on post	Pass	Pass	Pass
6	Input text into item name field	Pass	Pass	Pass
7	Text entered into item name field	Goat Cheese	Car	Prada Dress Shirt
8	Input text into price field	Pass	Pass	Pass
9	Text entered into price field	2	200	200
10	Input text description field	Pass	Pass	Pass
11	Text entered into description field	It's Cheesy	It's a mustang	It's fall 2016 collection
12	Choose category	Pass	Pass	Pass
13	Chosen category	Other	Other	Other
14	Add a photo	Pass	Pass	Pass
15	Click post	Pass	Pass	Pass

4. Code Review

Code review has been a vital part in this project and a fundamental way for peer development. We have chosen two style types for front end and back end teams as they utilize different syntaxes. For PHP we based it off of Google's company style guide and for Twig we used the standard use case presented in their documentation.

We utilized peer review using commit messages and in-person feedback. It allowed us to adapt and change our code continuously without interruption or great time spent on corrections.

Commit messages for the app/Resources/Views/gatortrader/Post.html.twig:

Marc Panlilio: Created Container for Post Item with name, price, category. Need pull request approved

Chohee Kim: Verified functionality and within style and sytax guidelines

Evan Terry: Pull request for View_Post_intial_setup approved. Accepted.

Code from app/Resources/Views/gatortrader/Post.html.twig:

```
{% block body %}

    <body>
    <h1 align="center">Post Item</h1>
    <div class="container">
        <form name="postForm" enctype="multipart/form-data" action='postExecute' id="postForm"
method="post">
            <div class="col-md-6">
                <div class="form-group row">
                    <label class="col-md-6 col-lg-6 control-label">Item Name: </label>
                    <div class="col-md-6 col-lg-6">
                        <input type="text" class="form-control" name="itemName" placeholder="Item Name"
                               required="true">
                    </div>
                </div>
            </div>
            <div class="col-md-6 text-left-imp">
                <div class="form-group row">
                    <label class="col-md-4 control-label">Price: </label>
                    <div class="col-md-6">
                        <input type="text" class="form-control" name="price" placeholder="Price"
required="true">
                    </div>
                </div>
            </div>
            <div class="col-md-6">
                <div class="form-group row">
                    <label class="col-md-6 col-lg-6 control-label" for="category">Category: </label>
                    <div class="col-md-6 col-lg-6">
                        <select class="form-control" id="category" name="category" placeholder="Category"
required="true">
                            {% for key in category %}
                                <option name={{ key.category }}>{{ key.category }}</option>
                            {% endfor %}
                        </select>
                    </div>
                </div>
            </div>
    {% endblock body %}
```

5. Security Practices

We are protecting through encryption the following assets:

- E-mail addresses
- Passwords

Symfony by default protects and encrypts password transactions. Due to lack of funding we are not able to purchase an SSL certificate and therefor it is lacking the ability for the HTTPS protocol.

We have verified that the SFSU email constraint is in place and functioning. We are currently working on form injection based exploits on other input forms to which we will change validation constraints appropriately.

The web framework is hosted on Amazon Web Services (AWS) and can be accessed through an SSH key. While this is a good solution for development purposes, there should be further precautions taken when the product is deployed such as limiting the access to IP address and creating two-step authentication.

6. Non-functional Specs Adherence

All of the following are completed except for the ones that are stated as **(On Track)**:

1. Application shall be developed using class provided LAMP stack
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 be explicitly approved by Anthony Souza on a case by case basis.
3. Application shall be hosted and deployed on Amazon Web Services as specified in the class
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.
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
6. Data shall be stored in the MySQL database on the class server in the team's account
7. Application shall be served from the team's account
8. No more than 50 concurrent users shall be accessing the application at any time
9. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
10. The language used shall be English.
11. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.
12. Google analytics shall be added **(On Track)**
13. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.
14. Pay functionality (how to pay for goods and services) shall not be implemented.
15. Site security: basic best practices shall be applied (as covered in the class)
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
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). **(On Track)**