SW Engineering CSC648/848 Section 01 Summer 2017
San Francisco State University

Milestone 4

Warehaus Media: Beta Release

Team One

Lindsey Anne - lhogg@mail.sfsu.edu
Josh Reed
Jasmine Mann
Jason Wong
Tyler Huang
Bo Li
Raymond Feng

Submitted to Dr. Dragutin Petkovic, Anthony Souza
Department of Computer Science, SFSU
Version 1.0
Last Modified August 8, 2017

1. Product Summary

Warehaus Media is a site made by our team of 7 San Francisco State University students. Our site is a web-based service specializing in the buying and selling of users' stock images. A few of our functional specs are:

- Ability to browse site as a guest user
- Guest Users can register for an account using an email, username, and encrypted password
- Registered Users can upload media they want to sell
- Ability to search for media by title and category
- Users can message sellers to buy media
- In the backend, admin can manage the database, which includes being able to removing images and users

Our website can be accessed by visiting `http://sfsuse.com/~su17g01/warehausmedia`

2. Usability Test Plan

Test Objectives

Assign tasks to test users to assess on user experience:

- Customers browse/search for images
- Customers send messages to merchants (owner of images)
- Merchants upload images
- Users manage their account using the dashboard

Test Plan

Task: Search

- Role: Customer.
- Open your browser and navigate to `http://sfsuse.com/~su17g01/warehausmedia`
- Search for images in the Food category.
- Verify that images of food are displayed.
- Fill out the chart on the next page with feedback.

Task: Details

- Role: Customer.
- Open your browser and navigate to `http://sfsuse.com/~su17g01/warehausmedia`
- Search for images in the Food category.
- Select an image and verify that its details are displayed.
- Fill out the chart on the next page with feedback.

Task: Upload

- Role: Merchant.
- Open your browser and navigate to `http://sfsuse.com/~su17g01/warehausmedia`
- Register for an account, and login.
- Upload a sample photo using the Upload link and entering its details.
- Verify that the photo uploads to the website.
- Fill out the chart on the next page with feedback.

Test Plan Feedback

Task Name	% completed	Errors	Comments
Search			
Details			
Upload			

Questionnaire Form

Please answer the following questions with regards to the usability of the Warehaus Media website. Circle the response that you feel is most correct.

1. I thought the Warehaus Media website was easy to use.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

2. I thought there was too much inconsistency with the Warehaus Media website.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

3. I think that I would like to use the Warehaus Media website frequently.

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

3. QA Test Plan

Test Objectives

The objective of the QA test is to thoroughly test the search function of Warehaus Media in order to ensure that search results display pictures relevant to the search query. Searching by text and category will be tested.

System Setup

Processor: 2.2 GHz Intel Core i7
Memory: 16 GB 1600 MHz DDR3
Operating System: macOS Sierra Version 10.12.6
Browser 1: Safari Version 10.1.2 (12603.3.8)
Browser 2: Google Chrome Version 60.0.3112.90 (Official Build) (64-bit)

Feature to be Tested

This QA test will test the functionality of the search function from Warehaus Media's home page. The test will test for the following expected functionality:

All pictures which match the search query (text, category, or both) shall be shown.
Irrelevant pictures shall be shown if no picture match the search query.
All results should be readable and openable.
The search query shall not disappear from the search box after searching.
All results shall be the same across both browsers.

Test Cases

Test 1 - Searching by Text

Search for "food" on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test 2 - Searching by Category
Search for "architecture" on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test 3 - Searching by Random Text

Search for random text “segdfaergfsd” on home page. Validate that every expected functionality is met. If all of the expected functionality is met, the test result shall be PASS. Perform the search on both browsers.

Test Results

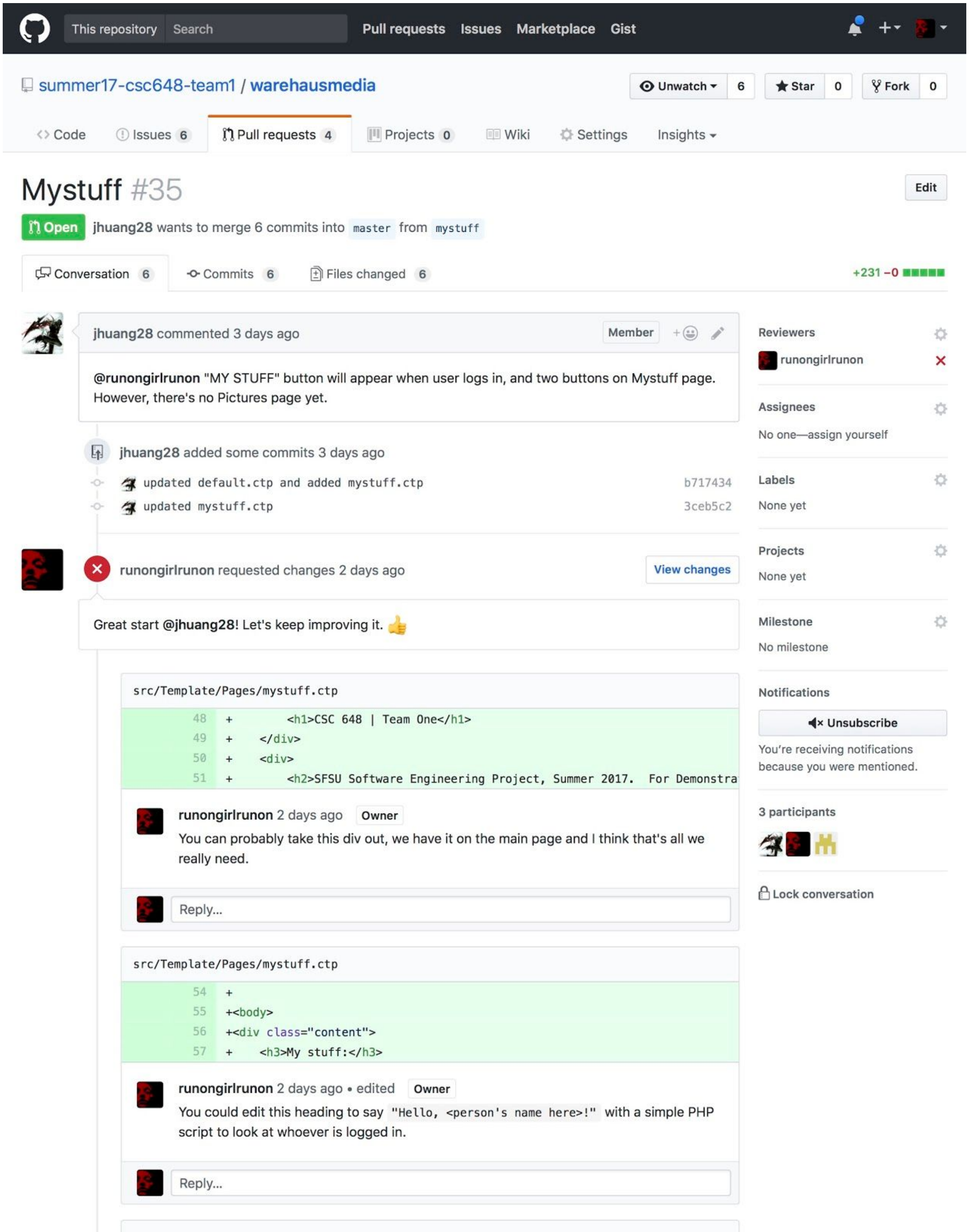
Test No.	Test Title	Test Description	Test Input	Expected Output	Test Result (Browser 1)	Test Result (Browser 2)
1	Searching by Text	Search for “food” by typing in search box	“food”	All pictures which contain “food” in the title	N/A	N/A
2	Searching by Category	Search for “architecture” by choosing it on drop-down menu	“architecture”	All pictures in the category of architecture	N/A	N/A
3	Searching by Random Text	Search for random text “akjhadf” by typing in search box	“akjhadf”	Some pictures	N/A	N/A

4. Code Review

Coding Style: To ensure that we make 100% use of the framework, we first made sure that we are all following all the naming conventions from tables names to classes and column names for posting and setting to and from the view. Second, we made sure that we use clear and meaningful variable names as well as method controller names so anyone will know what the view is going to do.

Code Review: We enforce strict review of everyone's code in order to make sure that software with a lot of obvious problems does not get into our main production branch. This process is done through Github. Once a developer is done with his/her part, the developer then pushes to a new branch on GitHub and creates a pull request with detailed comments and title specifying what he/she has done on the branch. Before merging is allowed, both the team lead and the tech lead will run his/her branch to ensure that the required functionality is functioning as requested. And if not, the reviewer can add comments on specific lines of code and request for changes.

The following page includes a sample of a pull request with code review remarks.



Mystuff #35

Open jhuang28 wants to merge 6 commits into `master` from `mystuff`

jhuang28 commented 3 days ago

@runongirlrunon "MY STUFF" button will appear when user logs in, and two buttons on Mystuff page. However, there's no Pictures page yet.

jhuang28 added some commits 3 days ago

```
updated default.ctp and added mystuff.ctp          b717434
updated mystuff.ctp                                3ceb5c2
```

runongirlrunon requested changes 2 days ago

Great start **@jhuang28**! Let's keep improving it. 👍

`src/Template/Pages/mystuff.ctp`

```
48  +        <h1>CSC 648 | Team One</h1>
49  +    </div>
50  +    <div>
51  +        <h2>SFSU Software Engineering Project, Summer 2017.  For Demonstra
```

runongirlrunon 2 days ago Owner

You can probably take this div out, we have it on the main page and I think that's all we really need.

`src/Template/Pages/mystuff.ctp`

```
54  +
55  +<body>
56  +<div class="content">
57  +    <h3>My stuff:</h3>
```

runongirlrunon 2 days ago • edited Owner

You could edit this heading to say `"Hello, <person's name here>!"` with a simple PHP script to look at whoever is logged in.

Reviewers

runongirlrunon ✕

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Notifications

Unsubscribe

You're receiving notifications because you were mentioned.

3 participants

Lock conversation

5. Self-Check on Best Practices for Security

Assets

1. Data in database is hidden from users and can only be manipulated through controller interaction or administrative privilege.
2. Web application will be used entirely in a HTTPS environment to encrypt sensitive data such as passwords and credit card numbers
3. Images are securely kept on a filesystem within the Amazon Web Service and cannot be accessed remotely.
4. Users passwords are encrypted with bcrypt password hashing technology that uses a salt to protect against rainbow table attacks as well as brute-force search attacks.
5. Input is correctly validated to protect against SQL injection and database error due to malformed queries.

6. Adherence to Non-Functional Specifications

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. **(On Track)**
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 **(On Track)**
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, Summer 2017. For Demonstration Only"*. (Important so as to not confuse this with a real application). **(On Track)**
18. Support all current versions of Chrome, Firefox, Edge, and Internet Explorer. Older versions shall not be explicitly supported. **(Done)**