SW Engineering CSC 648/848 Section 02

Spring 2017

Gator Store

A one-stop e-shop where San Francisco State University Students can buy and sell used and new items.

Milestone 4

May, 05 2017

Team 09 (*local*)

Charles Williams (aasim@mail.sfsu.edu)
Jason Huang
Myat Min Maung
Krishnan Ramakrishnan
Leopoldo Rodriguez
Grant Gaviglio
Wilson Le

Revision history

Date	Description
May, 05 2017	Rough draft for in-class feedback

Product Summary

Gator Store: is a site for SFSU student to sell items such as books, computers, furniture, clothing and more, directly to other SFSU students. Our website will initially cater to San Francisco State University students, but as the service grows, we could expand to other schools and offer the same services to them as well. The unique feature our website offers is the one-click Ordering option.

What we offer (Gator Store):

- We are dedicated to serve the students of San Francisco State University.
- With Nearly 30000+ students attend SFSU who are always looking to buy and sell their books as well as other various products at a cheaper price.
- With our feature that lets the buy
- Fully searchable and easily browse listings.
- Communication system to enable messaging between the seller and the buyer.
- **Unique Feature:** With our One-Click ordering places your order automatically and lets you skip the shopping cart.
- We will offer the books as well as other various products listed on our site to the students of SFSU which will shell be listed by student so students can set their own competitive price.

You can visit 'Gator Home' at: http://www.sfsuse.com/~sp17g09/gatorstore/

Usability Test plan

Test Objectives:

The Usability Test Plan focuses on testing the feature for its ease of use. We select the **'Search' feature** to analyze in detail from the testing standpoint.

Test Plan:

System Setup:
User will need access to the internet and a web browser. In the web browser running a recent version of either Chrome, or Firefox in which the user would need to type the URL provided which would navigate them to the homepage of the website.

Starting Point: Navigate to: http://www.sfsuse.com/~sp17g09/gatorstore/

Tasks:
1. From the webpage, search for a chair
2. From the webpage, search for a cup
3. From the webpage, search for a laptop
4. From the webpage, search for a phone

Intended Users: The intended users for the website are SFSU students seeking to buy and sell there items to other SFSU students.

Completion Criteria: User performs a search using a valid keyword to search for and find an item for each step.

Questionnaire form:

Likert scale questions to gauge usability of the search function are:

1. It was easy to locate search function on the website from any page.

○ Strongly Agree
○ Agree
○ Neutral ○ Disagree
○ Strongly Disagree

2. It was easy to use search and find results to my satisfaction.

○ Strongly Agree

○ Agree

○ Neutral

○ Disagree ○ Strongly Disagree

3. 2. It was easy to narrow down the search scope with the given category search filters.

○ Strongly Agree

○ Agree

○ Neutral

○ Disagree ○ Strongly Disagree

4. The search bar wasn’t too obstructive as it communicated the kind of query expected.

○ Strongly Agree

○ Agree

○ Neutral

○ Disagree ○ Strongly Disagree

Comments: (Optional)

Quality Assurance Test Plan

Test Objectives:

Ensure the usability and functionality of searching products and searching products with a filter, along with the ability to post, and remove items.

Hardware and Software Setup:

Website supports web browsers (Chrome, FireFox, and IE). Internet connection is required to interact with website.

Hardware:

Processor: AMD FX-8359 Eight-Core 4.00GHz
RAM: 8GB
OS: Windows 7 64-bit

Feature To Be Tested:

Test to be repeated by all browsers:

We are targeting the ‘search’and functionality for QA testing. Its features to be to ensure the stability of the **functionality and usability** of the search feature, requires numerous tests. The tests started searching products in general. By default, the search bar had filters are set to **all** as a result, some results were not expected. For example, if the user searched “book”, the user expects the results to be books. However the results also included “Macbook charger”. By change the search filter to books this makes the search being entered to be more narrow scope, so when the user types in a more broad word like “book”. The “Books” category list ensures that only books would show up and this goes for the same for the other categories. Next we test the posting function which lets the user make a new posting to the site. Additional Backend validation to prevent SQL injection which strips all semicolons from within the query. This is required since we use GET parameter that pass the search query through the URL which could attempt to input malicious backdoor queries.

Test Number	Description	Test Input	Expected Output	PASS/FAIL Google Chrome	PASS/FAIL Firefox

1.	Open the webpage: http://www.sfsuse.com/~sp17g09/gatorstore/ Change the category from “All” to “Books” and search with an empty search bar	**Category:** Books **Search:** “ “ **Sort by:** “Best Match”	Products that fall in the “Books” Category. **7 Results**	Fail	Fail
2.	Following Test #1, we change “Category” from “Books” to “All”.	**Category:** “All” **Search:** “ “ **Sort by:** “Best Match”	Products that fall in the “All” Category returned got more than books. **8 Results**	Fail	Fail
3.	With the “All” category selected, enter “physic” into the search bar.	**Category:** All **Search:**“cup” **Sort by:** “Best Match”	Products that contain the word “cup”. products, **2 Results**	Pass	Pass

Code Review

Coding guidelines:

As responsible developers, we adhered to the coding style of the set framework {cakephp} which uses the model, view, and controller model. The following resources were used as references and tests to ensure that all our code conformed to the set model.

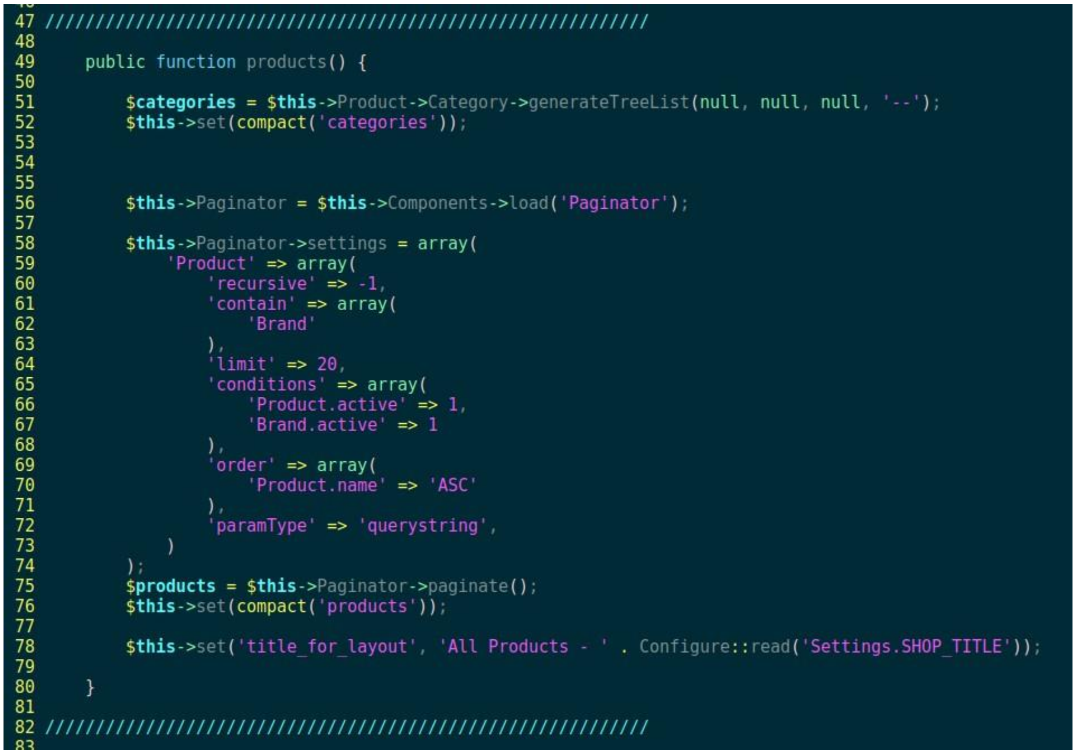
https://www.w3.org/html/
HTML is the standard markup language used to create web pages and its elements form the building blocks of all websites.

https://cakephp.org/
CakePHP makes building web applications simpler, faster, while requiring less code. A modern PHP 7 framework offering a flexible database access layer and a powerful scaffolding system that makes building both small and complex systems simpler, easier and, of course, tastier. Build fast, grow solid with CakePHP.

https://www.php.net/
PHP (recursive acronym for *PHP*: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

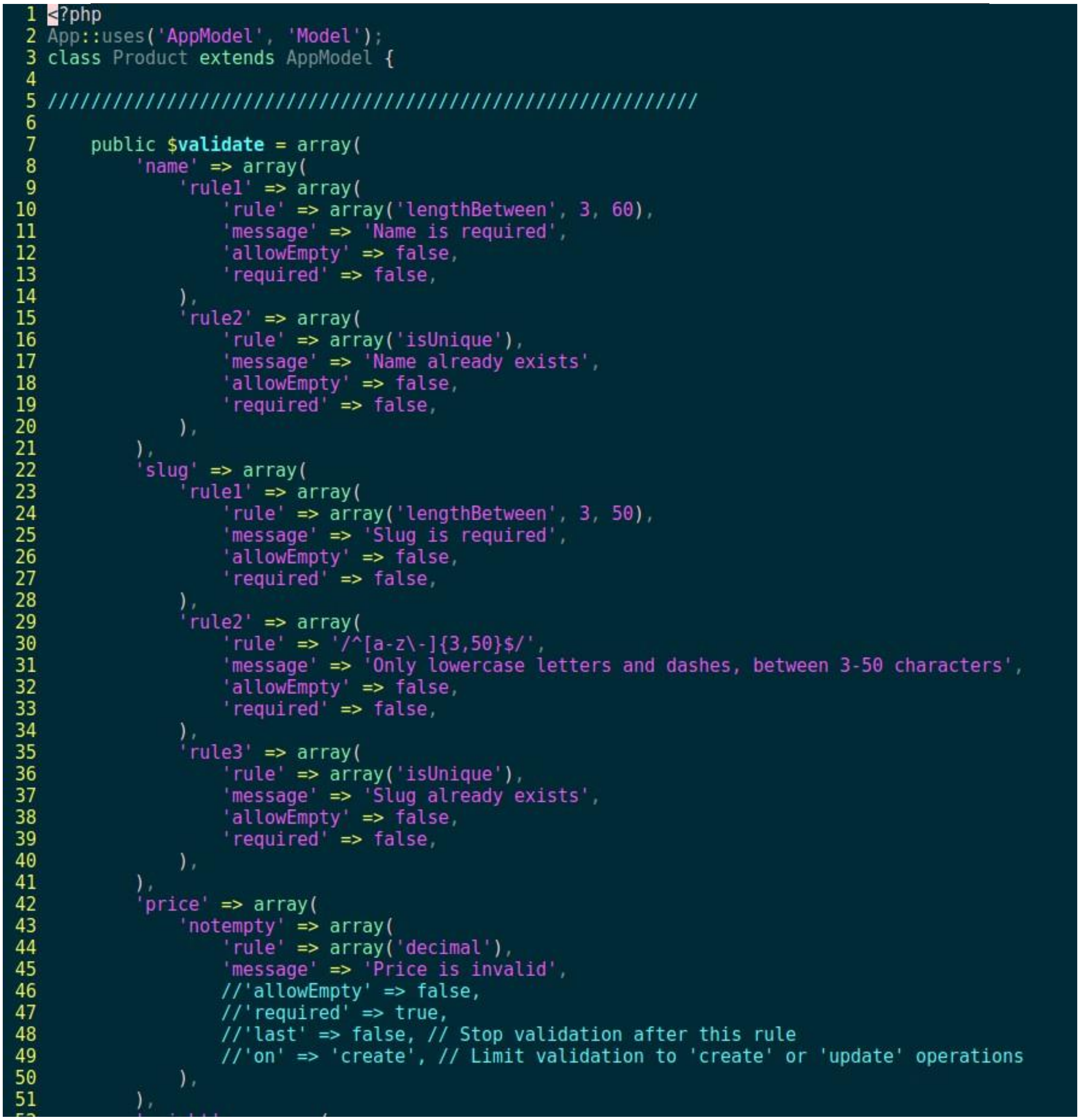
The listed screen shots are examples of our coding model adherence, note that these are only snippets of code and not the complete functions provided within each class.

Controller



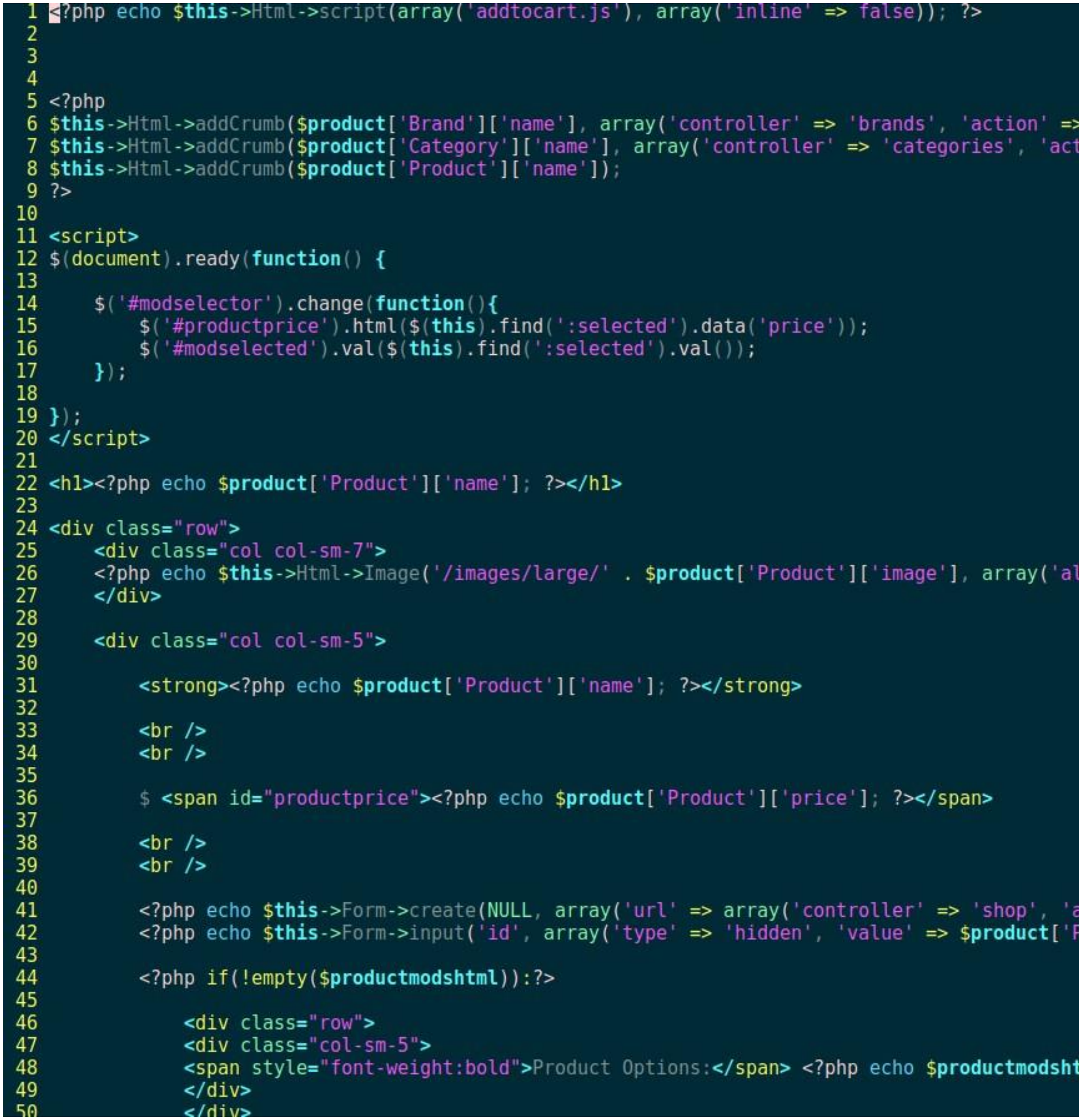
```
////////////////////////////////////////////////////////////

    public function products() {

        $categories = $this->Product->Category->generateTreeList(null, null, null, '--');
        $this->set(compact('categories'));



        $this->Paginator = $this->Components->load('Paginator');

        $this->Paginator->settings = array(
            'Product' => array(
                'recursive' => -1,
                'contain' => array(
                    'Brand'
                ),
                'limit' => 20,
                'conditions' => array(
                    'Product.active' => 1,
                    'Brand.active' => 1
                ),
                'order' => array(
                    'Product.name' => 'ASC'
                ),
                'paramType' => 'querystring',
            )
        );
        $products = $this->Paginator->paginate();
        $this->set(compact('products'));

        $this->set('title_for_layout', 'All Products - ' . Configure::read('Settings.SHOP_TITLE'));

    }

////////////////////////////////////////////////////////////
```

Model



```php
<?php
App::uses('AppModel', 'Model');
class Product extends AppModel {

////////////////////////////////////////////////////////////

    public $validate = array(
        'name' => array(
            'rule1' => array(
                'rule' => array('lengthBetween', 3, 60),
                'message' => 'Name is required',
                'allowEmpty' => false,
                'required' => false,
            ),
            'rule2' => array(
                'rule' => array('isUnique'),
                'message' => 'Name already exists',
                'allowEmpty' => false,
                'required' => false,
            ),
        ),
        'slug' => array(
            'rule1' => array(
                'rule' => array('lengthBetween', 3, 50),
                'message' => 'Slug is required',
                'allowEmpty' => false,
                'required' => false,
            ),
            'rule2' => array(
                'rule' => '/^[a-z\-]{3,50}$/',
                'message' => 'Only lowercase letters and dashes, between 3-50 characters',
                'allowEmpty' => false,
                'required' => false,
            ),
            'rule3' => array(
                'rule' => array('isUnique'),
                'message' => 'Slug already exists',
                'allowEmpty' => false,
                'required' => false,
            ),
        ),
        'price' => array(
            'notempty' => array(
                'rule' => array('decimal'),
                'message' => 'Price is invalid',
                //'allowEmpty' => false,
                //'required' => true,
                //'last' => false, // Stop validation after this rule
                //'on' => 'create', // Limit validation to 'create' or 'update' operations
            ),
        ),
```

View



```
<?php echo $this->Html->script(array('addtocart.js'), array('inline' => false)); ?>



<?php
$this->Html->addCrumb($product['Brand']['name'], array('controller' => 'brands', 'action' =>
$this->Html->addCrumb($product['Category']['name'], array('controller' => 'categories', 'act
$this->Html->addCrumb($product['Product']['name']);
?>

<script>
$(document).ready(function() {

    $('#modselector').change(function(){
        $('#productprice').html($(this).find(':selected').data('price'));
        $('#modselected').val($(this).find(':selected').val());
    });

});
</script>

<h1><?php echo $product['Product']['name']; ?></h1>

<div class="row">
    <div class="col col-sm-7">
    <?php echo $this->Html->Image('/images/large/' . $product['Product']['image'], array('al
    </div>

    <div class="col col-sm-5">

        <strong><?php echo $product['Product']['name']; ?></strong>

        <br />
        <br />

        $ <span id="productprice"><?php echo $product['Product']['price']; ?></span>

        <br />
        <br />

        <?php echo $this->Form->create(NULL, array('url' => array('controller' => 'shop', 'a
        <?php echo $this->Form->input('id', array('type' => 'hidden', 'value' => $product['P

        <?php if(!empty($productmodshtml)):?>

            <div class="row">
            <div class="col-sm-5">
            <span style="font-weight:bold">Product Options:</span> <?php echo $productmodsht
            </div>
            </div>
```

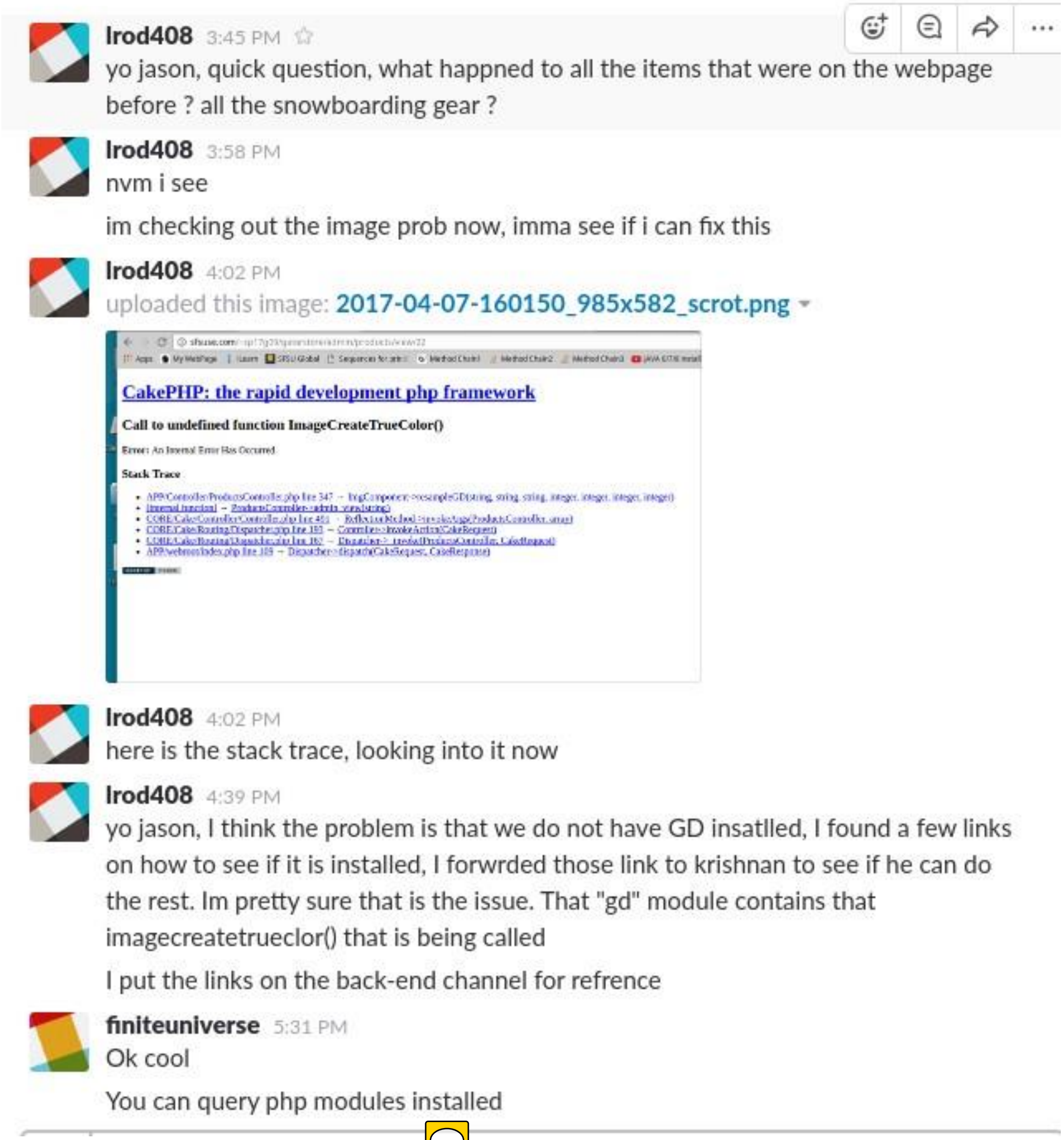
Peer Reviews:

We used git and github as a platform for peer review, we resolved conflicts through face-to-face and slack communication. We used github to track issues and bugs when a specific entity presented itself, a long with issue tracking we used github to assign specific tasks to each member. Milestones were also established through github, because this application was written with a small scale deployment in mind, we deployed this application through github instead of heroku or other deployment tools. The only issue we had was granting permissions to certain files and folders during the integration process.

All the github features may be found here: https://github.com/features

---

Below are some screen shots of our interactions through face-to-face and slack communications.



lrod408 3:45 PM

yo jason, quick question, what happned to all the items that were on the webpage before ? all the snowboarding gear ?

lrod408 3:58 PM

nvm i see

im checking out the image prob now, imma see if i can fix this

lrod408 4:02 PM

uploaded this image: 2017-04-07-160150_985x582_scrot.png

CakePHP: the rapid development php framework

Call to undefined function ImageCreateTrueColor()

Error: An Internal Error Has Occurred.

Stack Trace

- APP/Controller/ProductsController.php line 347 → ImgComponent->resampleGD(string, string, string, integer, integer, integer, integer)
- [internal function] → ProductsController->admin_view(string)
- CORE/Cake/Controller/Controller.php line 491 → ReflectionMethod->invokeArgs(ProductsController, array)
- CORE/Cake/Routing/Dispatcher.php line 193 → Controller->invokeAction(CakeRequest)
- CORE/Cake/Routing/Dispatcher.php line 167 → Dispatcher->_invoke(ProductsController, CakeRequest)
- APP/webroot/index.php line 109 → Dispatcher->dispatch(CakeRequest, CakeResponse)

lrod408 4:02 PM

here is the stack trace, looking into it now

lrod408 4:39 PM

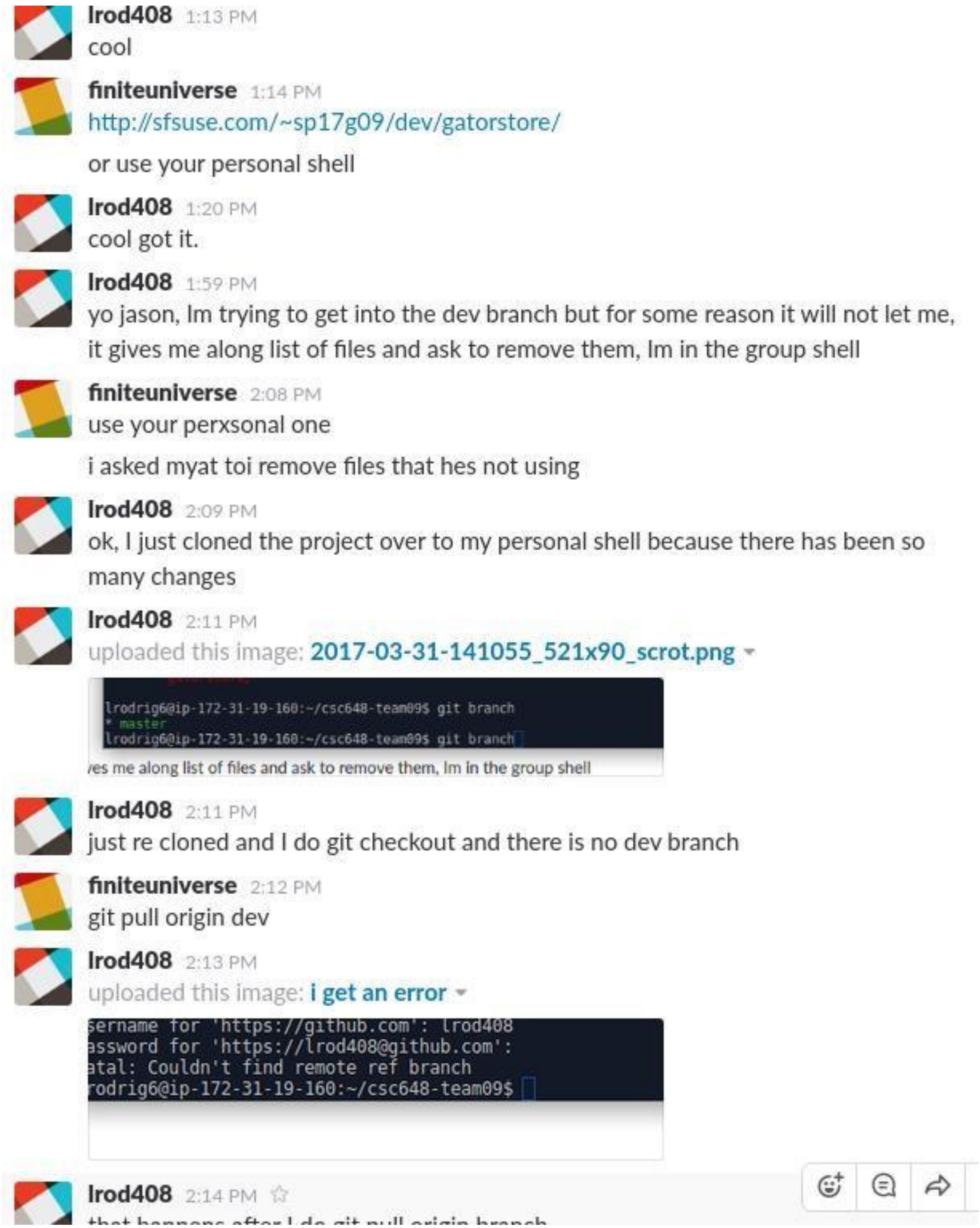
yo jason, I think the problem is that we do not have GD insatlled, I found a few links on how to see if it is installed, I forwrded those link to krishnan to see if he can do the rest. Im pretty sure that is the issue. That "gd" module contains that imagecreatetrueclor() that is being called

I put the links on the back-end channel for refrence

finiteuniverse 5:31 PM

Ok cool

You can query php modules installed



lrod408 1:13 PM
cool

finiteuniverse 1:14 PM
http://sfsuse.com/~sp17g09/dev/gatorstore/

or use your personal shell

lrod408 1:20 PM
cool got it.

lrod408 1:59 PM
yo jason, Im trying to get into the dev branch but for some reason it will not let me, it gives me along list of files and ask to remove them, Im in the group shell

finiteuniverse 2:08 PM
use your perxsonal one

i asked myat toi remove files that hes not using

lrod408 2:09 PM
ok, I just cloned the project over to my personal shell because there has been so many changes

lrod408 2:11 PM
uploaded this image: 2017-03-31-141055_521x90_scrot.png

```
lrodrig6@ip-172-31-19-160:~/csc648-team09$ git branch
* master
lrodrig6@ip-172-31-19-160:~/csc648-team09$ git branch
```

ves me along list of files and ask to remove them, Im in the group shell

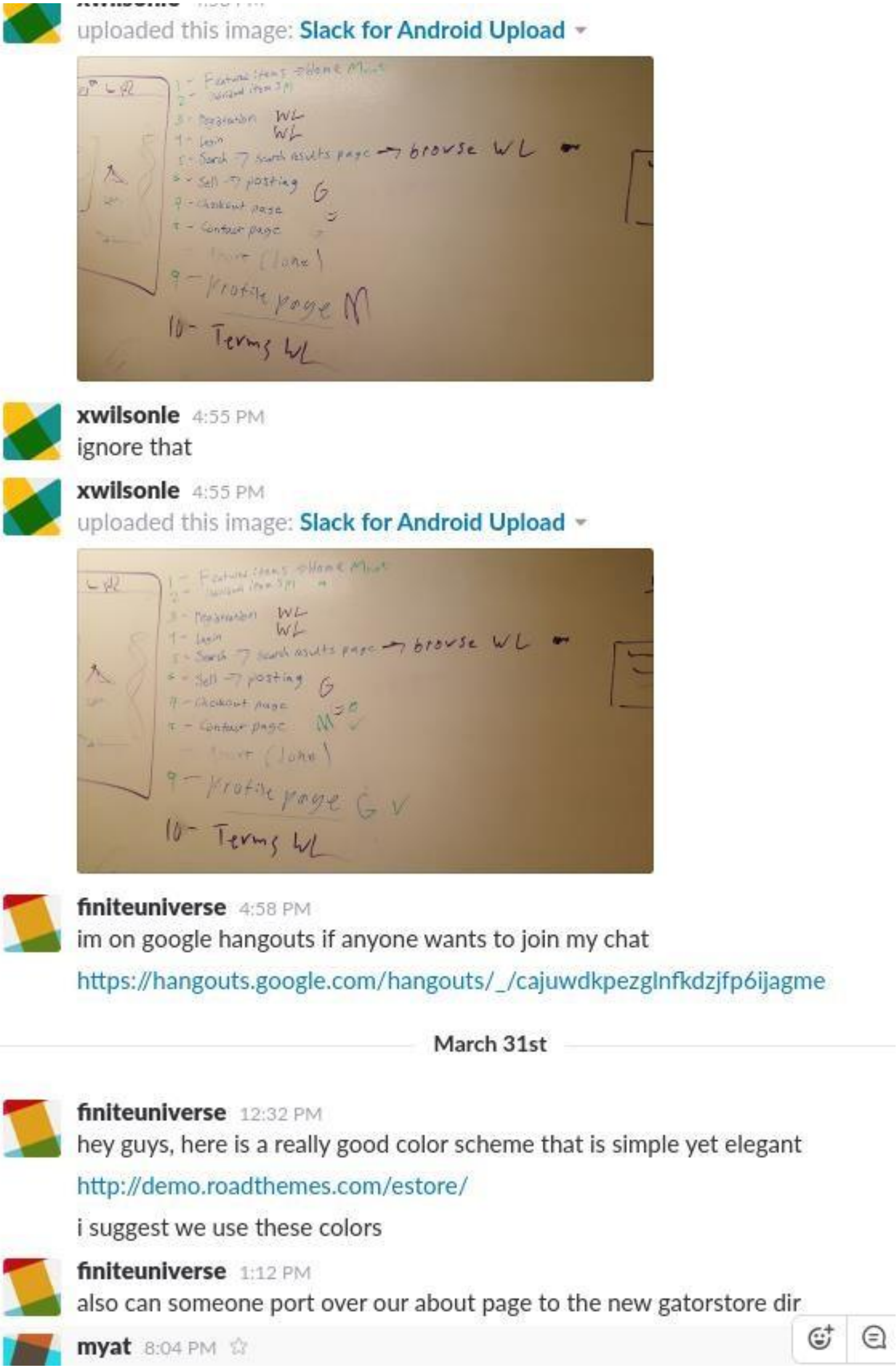
lrod408 2:11 PM
just re cloned and I do git checkout and there is no dev branch

finiteuniverse 2:12 PM
git pull origin dev

lrod408 2:13 PM
uploaded this image: i get an error

```
sername for 'https://github.com': lrod408
assword for 'https://lrod408@github.com':
atal: Couldn't find remote ref branch
rodrig6@ip-172-31-19-160:~/csc648-team09$
```

lrod408 2:14 PM
that happens after I do git pull origin branch



uploaded this image: **Slack for Android Upload**

xwilsonle 4:55 PM

ignore that

xwilsonle 4:55 PM

uploaded this image: **Slack for Android Upload**

finiteuniverse 4:58 PM

im on google hangouts if anyone wants to join my chat

https://hangouts.google.com/hangouts/_/cajuwdkpezglnfkdzjfp6ijagme

March 31st

finiteuniverse 12:32 PM

hey guys, here is a really good color scheme that is simple yet elegant

http://demo.roadthemes.com/estore/

i suggest we use these colors

finiteuniverse 1:12 PM

also can someone port over our about page to the new gatorstore dir

myat 8:04 PM

Self-check on best practices for security

Major assets we aim to protect include:

- User’s confidential information such as his/her email address and password.
- We protect against invalid listings being posted by requiring the user to register and login to post a new listing.
- We protect the storage server against upload of images greater than 10 MB by attempting to resize larger images.

Encryption of passwords in the Database:

Passwords are encrypted before being stored in the database using the functions recommended by PhP foundation for hashing and verifying passwords. It uses the Bcrypt algorithm in the password_hash function to generate a fixed 60 character hash.

We use the inbuilt password_verify() to verify for password matches.

Form input validation

1. **Registration form** is validated by using JQuery Validation and it will check if the user filled in the required fields: First Name, Email Address, User Name, Password, and Confirm Password. For certain fields, numbers are not allowed in first name, emails must be in an email format, and confirm password must be the same as the password field before it.

2. **Creating a posting** is validated using JQuery Validation and it will check if the user filled in the required fields: name, price, category, Image, size of listing, and description.

3. **The search bar** in the header uses a custom regex that will allow letters, numbers. Any incorrect input from the registration, create listing pages, and the search bar. Will give error messages to the user next to the form field that tell them what they must correct and what input is valid.

Adherence to original non-functional spec

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, and 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 served from the team's account**(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. **(On track)**

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**(Done) //Todo We need to add the line of code to the PHP**

13. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services. **(On track)**

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)**

The Team

Name	**Role**
Charles Williams	(**Team lead**) **QA**
Jason Huang	(**Tech lead**) **Backend**
Krishnan Ramakrishnan	**Backend**
Leopoldo Rodriguez	**Backend**
Myat Min Maung	**Frontend**
Grant Gaviglio	**Frontend**
Wilson Le	**Frontend**