

# SW Engineering CSC 648/848 Spring 2017

## Section 01

### SFSU Congre-Gator Market Team 04 (local)

Ishan Taha ([ishan@mail.sfsu.edu](mailto:ishan@mail.sfsu.edu))

Prateek Gupta  
Darrel Ogbonna  
Shane Cota  
Kevin Chu  
Mark Tompong

**Milestone 4**  
**3<sup>rd</sup>, May 2017**

#### History Table

Initial Draft of Milestone 4	05/03/2017	V1.0

## **Product Summary**

Congre-Gators is a website for SFSU students to buy and sell goods. Congre-Gators is intended to be a hub for students in need of offloading goods for cash to link up to other students in need of picking up used goods at an affordable price.

The home page at Congre-Gators displays the most recently posted items. Users may browse these items or explore other item listings via the search bar without being logged-in. This functionality is akin to “window shopping” and allows user to peruse the website on an informal trial basis before deciding to register if they haven’t already done so. Items can be sorted on price, date and name. Each posted item presents an opportunity to purchase it by presenting an easily visible button labeled ‘buy’. Upon attempting to purchase an item, if a user is not already logged in, they will be informed that they must log-in in order to buy an item.

Users are also invited to sell items via the button “sell” displayed in the top right corner. Selecting sell will redirect a user to a page where they may upload item images, and write descriptions of their items and set a price. As with buy, when a user who is not logged-in attempts to upload any data they will be informed they must log-in in order to do so.

Users will have easy access to either logging-in or registering with the website. Upon logging in, a visitor will be redirected back to the most recently visited page before logging in. Upon registration, a user will also be redirected back to the most recently visited page before registration.

Logged-in visitors have access to additional functionality general users don’t have. In addition to being able to freely buy and sell items, a user can check their notifications page which records conversations with other users. The notifications page offers the option of helping coordinate a meetup place between parties by accessing Google Maps.

Finally, a logged-in user may access a drop-down called “My Account” which provides links to a profile page where a user may edit information about themselves as well as a report lightbox which allows one to report misconduct to website administrators, and finally the simple ability to logout.

Congre-Gators is visible to everyone, and items may be browsed by anyone. However, full functionality is offered only to SFSU students at this time. This visibility and limited functionality will allow relevant users to explore the site and ultimately give it a trial before registration, akin to window shopping.

## **Usability Test Plan**

### **Test Objective:**

The objective for the usability test for the search feature in Conre-Gators is to ensure that the search bar is easy to find, use and that it yield relevant search results. Congre-Gators works because of its ability to bring relevant goods to a user in need. Some may end up being uncommonly found. It is therefore essential that the search feature be easy to use such that if a user is looking for something that happens to have been posted to Congre-Gators, they are able to find it.

### **Test Plan:**

System Setup: The user will go to to the Congre-Gator homepage by visiting the following link: <http://sfsuse.com/~sp17g04/Home/view/home>.

Starting Point: The user will be given login credentials and log-in.

Task to be Accomplished: The user will be given an item to search on as well as given free reign to search for whatever they like. The shall do so by specifying categories as well as a general search.

Completion Criteria: The user shall receive the relevant results to their search.

Intended User: Any SFSU students who are interested in flipping unused goods for a profit, as well as students who are in need of cheap items to help furnish their life in college.

### **URL:**

<http://sfsuse.com/~sp17g04/Home/view/home>

### **Questionnaire:**

Question:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Search bar is easily visible and user input is easily entered					
Drop-down menu for categories is easily visible					
Search results are relevant					

Comments:	
-----------	--

## **Quality Assurance Test Plan**

### **Test Objective:**

This quality assurance test is to make sure that the search feature is designed and working properly. This is not a test for usability.

### **HW and SW Setup:**

The website shall be viewable in a standard desktop/laptop browser and shall render correctly on the latest version of all major browsers: Mozilla Firefox, Google Chrome and Safari.

### **Feature to be Tested:**

Search bar/categories.

### **Actual Test Cases:**

Number	Description	Test Input	Expected Output	Pass/Fail
1.	General Search – no specific category used	Search on “likely” posted items	Relevant results displayed	Pass
		Search on “nonsense” items	Invalid search message displayed, followed by popular items for sale	Pass
2.	Category Search – specific category used	Search on an item related to the selected category	Relevant results displayed	Pass
		Search on an item unrelated to the selected category	No search results, followed by popular items for sale	Pass
		Search on “nonsense” item	Invalid search message displayed, followed by popular items for sale	Pass



Shane Cota &lt;shanecota@gmail.com&gt;

## CSc 648, Request for Code Review

2 messages

**Shane Cota** <shanecota@gmail.com>

Sun, Apr 30, 2017 at 2:00 PM

To: ihsan@mail.sfsu.edu

Hello Ihsan,

Following is a portion of the code from views/common/registered\_navbar.php. Since we are using our search feature for usability test and QA, we should also use a portion of related code for code review. Please go through it, and make sure are adhering to coding style. Make comments where necessary. If you have any questions you can hit me up, or refer to section 4 of Milestone 4.

Thanks!  
Shane

```

<!-- Centered Category Search & Input Search -->
<form class='form-inline mr-auto fix-align' style='padding-top: 4px' action="<?php echo base_url() .
"home" ?>" id="searchSubmit" method=GET>
  <label class='sr-only' for='inlineFormInputGroup'>Search</label>
  <div class='input-group mb-2 mr-sm-2 mb-sm-0'>
    <div class='form-group'>
      <select class='custom-select mb-2 mr-sm-2 mb-sm-0' id='categories' name='category'
onchange='document.forms["searchSubmit"].submit()'>
        <option value="">All</option>
        <?php
          // Populates category list
          foreach ($categories as $category) {
            // If user chose category, mark that option as selected
            $selector = "";
            if ($currentCategory == $category->category_id)
              $selector = " selected='selected'";

            // Adds category option to list
            print "<option value=$category->category_id$selector>$category->category_
name</option>\r\n";
          }
        ?>
      </select>
    </div>
  <!-- attempting to call controller for form validation on search -->
  <?php
    // echo form_open modified by scota
    //echo form_open('Users/search');
    $data = array(
      'class'      => 'form-control',
      'id'         => 'inlineFormInputGroup',
      'name'       => 'search',
      'placeholder' => 'Search ...',
      'type'       => 'search',
      'style'      => 'height: 40px',
      'value'      => $searchTerms
    );

```

```
    echo form_input($data);
    ?>

    <input type='hidden' name='sort' id='sort'>
    <button class='btn btn-success' style='cursor: pointer; height: 40px' type='submit'><i class='fa
fa-search' aria-hidden='true'></i></button>
  </div>
</form>
```

---

**Ihsan M Taha** <ihsan@mail.sfsu.edu>  
To: Shane Cota <shanecota@gmail.com>

Sun, Apr 30, 2017 at 4:51 PM

Hey Shane, thanks for sending the code. While it is functional, there are only minor issues with the indentation, alignment of a CodeIgniter form element, and comments. I have edited the code and highlighted in red where these changes are made.

#### Indentation:

1. In line four (now five), the div tag appeared nested when it should have been right underneath the label tag. This effect cascades to everything following it so I just highlighted the first line.
2. In line seven (now eight), the option tag is a part of the select tag, so we should indent it to appear as a part of the select tag.

#### Alignment:

1. In lines 26-35 (now 30-41), the arrows were not aligned and the echo statement was way off alignment-wise as well.

#### Spacing:

1. Since we generally try to avoid spacing the first and last line of a nested element, I have edited the spacing to appear the way it is below.

## PHP Comments:

1. Line nine, 11, 16, and 24 (now 11, 13, 19, and 28) the sentences should end with a period.

## HTML Comments:

1. Line 23 (now 26), 'attmempting' is misspelled and needs to change to 'Calls' anyways because the code works now.

```

<!-- Centered Category Search & Input Search -->
<form class='form-inline mr-auto fix-align' style='padding-top: 4px' action="<?php
echo base_url() . "home" ?>" id="searchSubmit" method=GET>
  <label class='sr-only' for='inlineFormInputGroup'>Search</label>

  <div class='input-group mb-2 mr-sm-2 mb-sm-0'>
    <div class='form-group'>
      <select class='custom-select mb-2 mr-sm-2 mb-sm-0' id='categories'
name='category' onchange='document.forms["searchSubmit"].submit()'>
        <option value=''>All</option>

        <?php
          // Populates category list.
          foreach ($categories as $category) {
            // If user chose category, mark that option as selected.
            $selector = "";

            if ($currentCategory == $category->category_id)
              $selector = " selected=\"selected\"";

            // Adds category option to list.
            print "<option value=$category->category_id$
selector>$category->category_name</option>\r\n";
          }
        ?>
      </select>
    </div>
  </div>

```

```
<!-- Calls controller for form validation on search -->
<?php
    // echo form_open modified by scota.
    // echo form_open('Users/search');
    $data = array(
        'class'      => 'form-control',
        'id'
=> 'inlineFormGroup',
        'name'        => 'search',
        'placeholder' => 'Search ...',
        'type'        => 'search',
        'style'
=> 'height: 40px',
        'value'
=> $searchTerms
    );

    echo form_input($data);
?>

<input type='hidden' name='sort' id='sort'>

<button class='btn btn-success' style='cursor: pointer; height: 40px'
type='submit'><i class='fa fa-search' aria-hidden='true'></i></button>
</div>
</form>
```

---

**From:** Shane Cota <[shanecota@gmail.com](mailto:shanecota@gmail.com)>  
**Sent:** Sunday, April 30, 2017 2:00:26 PM  
**To:** Ihsan M Taha  
**Subject:** CSc 648, Request for Code Review

[Quoted text hidden]



## **Self-Check on Best Security Practices**

### **Assets:**

User information including passwords and associated email

### **User password:**

Yes.

### **Input validations:**

Login and registration accept a limited number of alphanumeric characters. Registrations enforces uses of email ending in '@mail.sfsu.edu'. Search bar accepts alphanumeric characters as well as spaces.

## **Adherence to Non-functional Specs**

√ = Done

- √ The application shall be developed using the LAMP stack provided by the class CTO.
- √ The application shall be developed using an approved PHP Framework from the class CTO.
- √ The application shall be developed using GUI technologies.
- √ The application shall be hosted and deployed on Amazon Web Services provided by the class CTO.
- √ The application shall be optimized for standard desktop/laptop browsers.
- √ The application shall render correctly on the two latest versions of at least three major browsers such as Mozilla, Safari, and Chrome.
- √ The application shall have responsive UI code so it can be adequately rendered on mobile devices, but no mobile native app is to be deployed.
- √ Data shall be stored in MySQL database on the class server in the team's account.
- √ Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
- √ No more than 50 concurrent users shall be accessing the application at any time.
- √ The language used shall be English.
- √ The application shall be very easy to use and intuitive. No prior training shall be required to use the website.
- √ Google Analytics shall be added.
- √ Messaging between users shall be done only by class approved methods to avoid issues of security with email services.

- √ Pay functionality shall not be implemented on the website.
- √ Site security shall be applied according to best practices.
- √ Modern SE processes and practices shall be used including collaborative and continuous SW development.
- √ The website shall prominently display the following text on all pages “SFSU Software Engineering Project, Spring 2017. For Demonstration Only.”
- √ The team shall use Github to maintain the website codebase.