

# SW Engineering CSC648/848 Section

## 01 Spring 2017

## Student Swap

Group 06 is a Local Team

Alex Gaesser: *agaesser@gmail.com*

Gregory Ruffin

Anthony Ma

Robin Ma

Leanna Pangan

Avery Chua

Elric Dang

### Milestone 4

**04/27/2017**

### Revisions

Date	Description	Version
2/28	Created	1.0
3/10	First feedback	1.5
3/11	M1 Frozen	2.0
3/18	Second feedback	2.5
3/25	M2 Frozen	3.0

# 1) Product summary (e.g. how would you market and sell your product)

Product: Student Swap

Student Swap is a San Francisco State University oriented website dedicated for the mediation of student driven interactions for the sole purpose of creating a community through our application. Student Swap is an application designed to help students become accustomed to the challenges of college life by bringing them together when in search of school related products and services. Student Swap is the application students and faculty will be able to use when searching for anything relating to our school.

Our website is a posting service where users from SFSU wanting to buy and sell to others can do it in relative safety knowing that the receiver or seller is part of the SFSU community. Our focus is to provide a way for users of the SFSU community can efficiently and safely undergo transactions with relative safety.

Few key features of our website is to be able to search for items through a simple and yet effective category keyword system in which items will be organized in category and finally based on the description, be able to pinpoint whether other users are selling in our local community. The organization of Student Swap is completely focused on ease of use as well as oversighted from our instructor, Professor Petkovic, and his assistant to ensure that our website is both user oriented and efficient in terms of facility resources.

In accordance with user safety and California Law, our website policy on information on our database is effectively protected and encrypted to ensure our users do not become the victims of fraud. With our site only being a posting service, our website does not provide facilitating the transactions of the people involved. Our website is dedicated to helping students find one another for their time at SFSU.

Finally, what is unique to our website is the offerings of student facilitated tutoring as a service provided for the users of Student Swap. Our users will be able to list and offer tutoring to those who need assistance in the courses listed by the user. Tutors will be able to list a set of classes that they have taken or are proficient in and are able to be a source of help for other students in need.

Student Swap Link: [http://sfsuse.com/~sp17g06/student\\_swap/web/](http://sfsuse.com/~sp17g06/student_swap/web/)

# Usability Test (Student Swap)

**Objectives:** The focus of this test is to ensure the quality and the ease of use that Student Swap developers want to encapsulate. Be sure to give honest feedback as well as any suggestions to help provide an even greater product for final development.

**Problem Statement:** We want to determine if our website is easy to find items students are looking for.

**Target Audience:** Students and Faculty of SFSU

**System Setup:** Student Swap is coded to work on a multitude of different Web Browsers including mobile. They are coded using a combination of twig and html files with an MVC design pattern and run with Amazon Web Services.

**Environment:** Chrome, Firefox, Safari, Edge, and mobile devices. (Whichever the subject has available and handy)

**Method and Test Design:** Our subject will be tasked with looking for a book "Gone With the Wind" by Margaret Mitchell as a student for your class.

## **Task Description:**

<b>Task:</b>	<b>Criteria for Pass:</b>
1. Categories appropriately utilized	Item properly found under appropriate category
2. Search Bar Results	Item in search field properly matched with item "description"
3. Time to Complete Task	Found within 20 seconds, on appropriate item page
4. Interface Effectiveness	Immediate use of search bar with category

## **Questionnaire:**

Please fill out the table about your experience using our website

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The UI was easy to read and understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The Search bar was intuitive and effective	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Results were exactly as I expected them to be once I searched an item	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall, I would use the search function for Student Swap again	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## **Additional Comments:**

--

### **3) QA test plan**

**Objectives:** Test our Search Functionality for Student Swap and ensure that all forms of error are caught as well as the appropriate error message prompts for our users. QA subjects will act as a user on Student Swap and use the search options to test varying inputs and recording the appropriate response from the website.

**System Setup:** Student Swap is designed to work on a variety of browsers, so any of the following on any OS should be appropriate:

<b>Browser:</b>	<b>Version Number</b>
Google Chrome	58.0.3029
Mozilla Firefox	53.0.1
Safari	10.1
Microsoft Edge	<b>40.15063</b>
mobile devices	Most updated

**Primary Feature Tested:** Test cases focusing around Student Swaps SEARCH function will be evaluated. All test subjects will begin at the homepage of Student Swap and follow the prompts and record based on expected results and bugs regarding on responses from the website. Failed/Pass will be determined based on the expected results detailed in the following report sheet. Display bugs or concerns will be recorded at the end of the report.

**Secondary Tests:** Any concerns with website displays or bugs regarding formatting of the website.

**Test Case #1 (Items):** Our subject will be tasked with looking for a book “Gone With the Wind” by Margaret Mitchell as a student for your class.

Task:	Criteria for Pass:	Pass/Fail
Select “Books” in the Drop Down Category	Category is consistent through whole search test (highlighted left side + in search bar)	
Search “Gone With the Wind” with highlighted category	Item is Found promptly and is displaying the correct item	
Time to Complete Task	Search Took around 5 sec	
Change Category to “Furniture”	“Item not found, try _____” Error Message - Suggestion for item in “books” Category	
Search Bar “a;lskdfnksjdfb”	“Item cannot be found, invalid search term” Error Message - For niether category or description matches	

**Additional Report Space:**

Any bugs regarding display issues and error messages will be recorded here

Case:	Step:	Display/Bug Issue:
Case:	Step:	Display/Bug Issue:
Case:	Step:	Display/Bug Issue:
Case:	Step:	Display/Bug Issue:

## 4) Code Review:

**Greg Ruffin**  
☆ | O away | @gruffin

Greg, I've reviewed your code for the ItemController.php file you created. Please find my notes in the below snippet:

**Alex Gaesser** 12:20 PM  
added this PHP snippet: [ItemController.php review](#)

```
1 <?php
2
3 namespace AppBundle\Controller; //REVIEWED: Correctly follows namespace autoloading as per PSR-1
4
5 use FOS\UserBundle\Model\UserInterface;
6 use Sensio\Bundle\FrameworkExtraBundle\Configuration\Route; //REVIEWED: this isn't used, maybe remove it?
7 use Symfony\Bundle\FrameworkBundle\Controller\Controller;
8 use Symfony\Component\Security\Core\Exception\AccessDeniedException;
9
10 class ItemController extends Controller
11 {
12
13     /**
14      * @Route("/listings", name="listings")
15      */
16     public function listingsAction()
17     {
18         $repository = $this->getDoctrine()
19             ->getRepository('AppBundle:Item');
20
21         $items = $repository->findAll();
22
23         return $this->render('default/listings.html.twig', array('items' => $items));
24     }
25
26     /**
27      * @Route("/openOrders", name="openOrders")
28      */
29     public function openOrdersAction()
30     {
31
32         $user = $this->getUser();
33         if (!is_object($user) || !$user instanceof UserInterface) {
34             throw new AccessDeniedException('This user does not have access to this section.');
```

+ Message Greg Ruffin



```
59
60     $query = $repository->createQueryBuilder('item')
61         ->where('item.category = :category')
62         ->setParameter('category', 'book')
63         ->getQuery();
64
65     $books = $query->getResult();
66
67     return $this->render('default/books.html.twig', array('books' => $books));
68 }
69
70
71 /**
72  * @Route("/clothes", name="clothes")
73  */
74 public function clothesAction()
75 {
76     $repository = $this->getDoctrine()
77         ->getRepository('AppBundle:Item');
78
79     $query = $repository->createQueryBuilder('item')
80         ->where('item.category = :category')
81         ->setParameter('category', 'clothes')
82         ->getQuery();
83
84     $clothes = $query->getResult();
85
86     return $this->render('default/clothes.html.twig', array('clothes' => $clothes));
87 }
88
89 /**
90  * @Route("/tech", name="tech")
91  */
92 public function techAction()
93 {
94     $repository = $this->getDoctrine()
95         ->getRepository('AppBundle:Item');
96
97     $query = $repository->createQueryBuilder('item')
98         ->where('item.category = :category')
99         ->setParameter('category', 'tech')
100         ->getQuery();
101
102     $techs = $query->getResult();
103
104     return $this->render('default/tech.html.twig', array('techs' => $techs));
105 }
106 } // REVIEWED: code enclosed by trailing bracket
107
108
109 /** code review notes
110  * We are adhering to PSR-1 coding standards for PHP code
111  * Please see inline comments (denoted with REVIEWED: ) for additional feedback
112  *
113  * This file causes side-effects (generates output or changes) and does not declare symbols
114  * - in compliance
115  *
116  * This file uses only <?php and <?= tags
117  * - in compliance
118  *
119  * Classes are declared ins StudlyCaps
120  * - in compliance
121  *
122  * File uses UTF-8
```



Message Greg Ruffin





```
121 *
122 * File uses UTF-8
123 * - in compliance
124 *
125 * Namespaces and classes follow autoloading PSR
126 * - in compliance
127 *
128 * Class constants declared in upper case with underscores for spaces
129 * - no class constants. If you add any, make sure they adhere to this spec.
130 *
131 * Method names declared in camelCase
132 * - all method names are in compliance
133 *
134 *
135 * Readability notes (not a part of PSR-1)
136 * It's a good idea to put a quick summary of the inputs and outputs of each function
137 * as a comment before the function declaration, especially when the function isn't
138 * self-explanatory. Twig has a nasty habit of using confusing terminology.
139 * for example, someone new to the code will wonder what
140 *
141 *         $repository = $this->getDoctrine()
142 *         ->getRepository('AppBundle:Item');
143 *
144 * is doing. Please go back and add some summaries as time permits.
145 *
146 * All things considered, this is straightforward code that is 100% in compliance with our chosen
147 * coding standard, PSR-1. I don't really have a lot of feedback to give you other than
148 * use more comments!
149 *
150 */
```

Collapse ↑

ⓘ This snippet was truncated for display: [see it in full](#).



**Alex Gaesser** 12:22 PM

Please let me know if you have any comments or questions on my feedback. Overall, this file is compliant with our coding standard, and internally consistent in style (code blocks are the same, naming convention makes sense, etc)



Message Greg Ruffin



## **5) Self-check on best practices for security**

- Major Assets we are protecting:
  - Codebase is stored on private repository
  - Production code deployed to server
  - Development code tested on individual accounts and heavily tested before deployment to live site
  - Personal information on users is restricted to bare minimum required for site operation, and is encrypted in the database
- Passwords are encrypted in the database
- Data input is validated
  - Email address domain must be @mail.sfsu.edu
  - Invalid characters not accepted in search field
  - Database requests sanitized

## **6) Adherence to original Non-functional specs**

<b>Specification:</b>	<b>Status:</b>
1. Application shall be developed using the LAMP stack.	
2. Application shall be developed using pre-approved set of software development and collaborative tools.	
3. Application shall be hosted and deployed on Amazon Web Services.	
4. Application shall be optimized for desktop/laptop browsers of Chrome,Safari,Internet Explorer.	
5. The application shall render correctly on the latest versions of all major browsers.	
6. The application shall adequately render on browsers of mobile devices.	
7. Data shall be stored on team MySQL database.	
8. Application shall be hosted on team account	
9. Application shall not support more than 50 concurrent users.	
10. Application shall appropriately protect privacy of users and notify users appropriately.	
11. Application shall have a Terms of Use page prior to registration.	
12. The language of application shall be English.	
13. The application shall be intuitive to use and easy to use.	
14. The application shall include Google Analytics.	

<b>15.</b> Messaging between users shall be done by class-approved methods.	
<b>16.</b> Pay functionality shall not be implemented on the application.	
<b>17.</b> Website security practices shall be applied.	
<b>18.</b> Modern software practices including collaboration shall be applied.	
<b>19.</b> The website shall prominently display “SFSU Software Engineering Project, Spring 2017. For Demonstration Only” on all pages.	

**e-mail subject line:** Must be “CSC648-848 Section M Spring 2017 Milestone4 TeamN” in the subject line (M is section 01 or 02, N is a team number).



