

SW Engineering CSC648/848 Fall 2019

SECTION 1 Team-06

“SFSU Buy & Sell”

Milestone 4

12/11/2019

Team Members:

Manish Patil (*Team Lead*) mpatil3@mail.sfsu.edu

Jinan Huang (Front-end Lead)

Thomas Tran (Back-end Lead)

Poulomi Banerjee (Github Master)

Ha Phuong Tran (Front-end/Full Stack Developer)

Zihao Wu (Front-end Developer)

# Product Summary

- Product Name: SFSU Buy & Sell
- All Major Committed Functions:
  - User shall be able to browse on the website
  - User shall be able to search for the products
  - User shall be able to select the products from the product categories
  - User shall be able to access all the information about the product
  - User shall be able to post a listing on the website
  - User shall be able to decide the pickup spot **(Unique)**
  - User shall be able to add product for selling on the website
  - Admin shall be required to have the right of approving the products for posting live
- URL: **TO BE DETERMINED UPON FINAL DEPLOYMENT**

# Usability Test Plan

- Test Objectives:

Team-06 tested usability on our search functionality. We specifically tested for how users will be interacting with the search bar itself. Team-06 came to an agreement that search functionality is one of the most important features to have for any user given our product design and should be a very usable part of our application. Search should be usable where any users with or without internet web application experience shall be able to search for a specific item without too many variables. The easier our search is, the better the user experience is when navigating through our application.

- Test Background and setup:

## System Setup:

- Setup via PC and Macintosh OS devices
- Tested URL: <http://54.183.203.235/searchProducts> (url will change due to AWS functionality)

## Starting Point:

- When user clicks on the URL link of the website, they start at the Home Page where they can see on the top search bar having dropdown of categories and input for text that they are intended to search.

## Intended users:

Intended users are SFSU students.

### ● Usability Task description:

- Open SFSU Buy & Sell website.
- Search for an item that is a book with the lowest price.
- Search for an item that is a chair with the highest price.
- Search for all books.
- Search for all furniture.
- Search for all electronics.
- Search for all clothing.
- Search for all sports gears.
- Search for all event tickets.

- Lickert subjective test:

	Questions	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	Searching for an item by its Category (books, furniture, clothes, event tickets, electronics, sports gears) is easy to use.					
2	Searching a Category (books, furniture, clothes, event tickets, electronics, sports gears) by sorting price range is easy to use.					
3	The website is easy to use on different browsers.					
		Too Clumsy	Difficult to work with	Very Intuitive		
4	Are there any improvements that you feel SFSU Buy & Sell could make easy to use?					

# QA Test Plan

- **Test Objective:**

- We opted to test search functionality and capability extensively, specifically our search bar's search capability. Proper search functionality is critical to our product and without it, we would have no major product to deploy. In this matter, Team-06 decided it was best that search functionality be taken through multiple different tests as we felt this functionality is pivotal to our product. Being aware of any issues within our search bar prior to public deployment (Milestone 5 Presentation) allows us options to ensure our design will not break down unexpectedly during public trials and demonstrations. By taking this precaution, we avoid any drastic bugs or errors occurring when presenting our finalized product.

- **Test Background and Setup:**

- Setup via PC and Macintosh OS devices
- Tested URL: <http://54.183.203.235/searchProducts> (url will change due to AWS functionality)
- Done with 2 different browsers: Google Chrome and Safari

- **Feature to be tested:**

Working functionality of Search filter is tested.

- Test Table:

Test #	Description	Browser and OS used	Test input	Expected output	Pass/Fail
1	Test %Like in search	Chrome, Windows	"Sofa"	Return 2 results with Sofa in the name	Pass
2	Test %Like in Search	Safari, Macintosh	"-1"	Return no results	Pass
3	Test dropdown Search	Chrome, Windows	"Furniture"	Return ALL Furniture results	Pass

## Code Review

- Coding style:

- Opted for Java coding style/best practices.
- Code snippet from QA Test:

```
searchProducts: (req, res) => {
  searchString = req.body.search;
  categoryString = req.body.categoryName;

  if( searchString != undefined && categoryString != undefined ){
    var query = "SELECT * FROM Products WHERE title LIKE '%" + searchString + "' AND categoryName LIKE '%" + categoryString + "'";
  }else if(searchString == undefined){
    var query = "SELECT * FROM Products WHERE title LIKE '%" + searchString + "'";
  }else if(categoryString == undefined){
    var query = "SELECT * FROM Products WHERE title LIKE '%" + searchString + "'";
  }else{
    var query = "SELECT * FROM Products"
  }
  // execute query
  db.query(query, (err, result) => {
    if (err) {
      res.redirect('/');
    }
    res.render(pages + '/search2', {
      title: "Welcome to SFSU | View Proucts",
      products: result
    });
  });
},
```



**Manish Patil** 5:54 PM

This is the code of search functionality (edited)

image.png ▼

```
searchProducts: (req, res) => {
  searchString = req.body.search;
  categoryString = req.body.categoryName;

  if( searchString != undefined && categoryString != undefined ){
    var query = "SELECT * FROM Products WHERE title LIKE '%" +searchString+ "' AND categoryName LIKE '%" +categoryString+ "'";
  }else if(searchString == undefined){
    var query = "SELECT * FROM Products WHERE title LIKE '%" +searchString+ "'";
  }else if(categoryString == undefined){
    var query = "SELECT * FROM Products WHERE title LIKE '%" +searchString+ "'";
  }else{
    var query = "SELECT * FROM Products"
  }
  // execute query
  db.query(query, (err, result) => {
    if (err) {
      res.redirect('/');
    }
    res.render(pages + '/search2', {
      title: "Welcome to SFSU | View Products",
      products: result
    });
  });
},
```



**Thomas Tran** 5:56 PM

yes (edited)

thank you

## Security

- Assets being protected:
  - **User passwords**
    - Protected via Encrypting user passwords. Done using Node Crypto
  - **User ID#**
    - Protected and hidden from public
  - **Seller IP address**
    - Not currently protected by certifications, but can be done
  - **User Search**
    - Validated search done via Express-Validator



- Confirm that you encrypt PW in the DB - Working
- Confirm Input data validation - Working

## **Self-Check**

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). - **DONE**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers. - **DONE**
3. Selected application functions must render well on mobile devices.-**DONE**
4. Data shall be stored in the team's chosen database technology on the team's deployment server.-**DONE**
5. No more than 50 concurrent users shall be accessing the application at any time. - **ON TRACK**

6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.- **ON TRACK**
7. The language used shall be English.-**DONE**
8. Application shall be very easy to use and intuitive. **DONE**
9. Google analytics shall be added. **ISSUE (Shortage of time)**
10. No email clients shall be allowed. **DONE**
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**
12. Site security: basic best practices shall be applied (as covered in the class). **DONE**
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.  
**DONE**
14. The website shall prominently display the following exact text on all pages “SFSU SoftwareEngineering Project CSC 648-848, Fall 2019. For Demonstration Only” at the top of the WWW page. (Important so as to not confuse this with a real application). **DONE**