San Francisco State University

SW Engineering CSC648/848 Spring 2020
"Gator Crawl"

Section 02 Team 01

Francis Cruz fcruz2@mail.sfsu.edu
Zain Khan
Jugal Bhatt
Jeffery Wan
George Freedland
Huy Phan

Milestone 4

M4 Date Submitted	M4 Date Revised
4/21/20	4/21/20

1) Product summary (e.g. how would you market and sell your product – about ½ page)

Name of the product

Gator Crawl

Explicit itemized list of ALL major committed functions

- Users can login and register
- Users can post, search, view, close, and buy items
- Users can bid on items if product permits
- Users can build profiles
- Users can message each other
- Admin can delete, reject, and accept items for sale

URL to your product accessible to instructors, on deployment server

Link may be subject to change. AWS changes IP of server when stopped and started again.
Current URL for Milestone 3: http://ec2-3-133-87-174.us-east-2.compute.amazonaws.com/

2) Usability test plan - Search /Filter Functionality

Test Objective

We are going to discuss the usability test plan for our search and filter functionality. We are focussing it on this test plan because search is of the main features of many online shopping stores. The search and filtering functionality is important. Since it helps the users find their products within seconds of looking. If we don't have a good search or filter function implemented in our website; it will be hard for the user to find the item they are looking for. Which would lead to a bad user experience and interface design. Having a bad user experience and interaction will cause the users to be upset with our website because it won't be as efficient as we like. We want to ensure if the user types in “computer” in the search bar; that they are only presented with all products that have the keyword computer in their name or description. Or if they filter by “electronic”, then all electronic results will be displayed.

Test Background and Setup

System Setup for Search: To set up for our system to run the Search system is pretty simple. First we need to ensure that our search functions are up and running on our website. The second thing we need to set up is to ensure our text forum is allowing for user input. The third step is to ensure the system takes what was captured from the text forum and search our database.. The system will take what the user entered as an input and try to match it with the best products. The final step up for search, the system will retrieve all items matching that text forum input and display the results to the user. After finding the best match products; it will display those products to the user.

System Setup for Filter: To set up for our system to run the Filter system is similar to the Search. First we need to ensure that our filter functions are up and running on our website. The second thing we need to set up is to ensure our drop down menu is allowing for user clicks.. The

third step for the setup, is to ensure the system takes what was captured and filters our database for items matching that description.. The final step up for the filter is, when the system will retrieve all items matching that filter input and display the results to the user.

Starting point : Our starting point is to ensure that our website is taking in user inputs from the search bar and the drop-down menu.

Intended Users : This function is going to be intended for all users who have access to the site. Both users and admins will be allowed access to this functionality.

URL : http://localhost:3000/searchresults

Usability Task Description

The first step to test the Search and filter functionality is to go to our website and log into your account.

. For ***Search -***. The first step is to click on the search tab on the top ribbon. The next step is to enter an item you are looking for in the search bar. Then hit enter to finalize that search. (e.g If you want to look for shoes; then type in shoes in the search bar and press enter). This will show all matches with the keyword of shoes.

For ***Filter -*** The first step is to click on search on the top ribbon. The next step is to click on the drop down menu. Then select one of the following options and press enter.(e.g If you are looking for clothes on our website, then select "Apparel" from the drop down menu and hit enter). This will show all results matching apparel.

We would measure effectiveness based on the search results generated by the search and filter functions. If you can find what you want by your search. Then it had strong effectiveness. Otherwise it has poor effectiveness.

We would measure efficiency based on time spent looking for a product. If it took a shorter time to find the product you are looking for by searching or filtering than it has high efficiency.

However it took a shorter time by looking through the products one by one then it has low efficiency.

Lickert Subjective Test :

1. The search function was easy to use? (Strongly Disagree to Strongly Agree)
2. The filter function was easy to use? (Strongly Disagree to Strongly Agree)
3. You found what you were looking for with the search results? (Strongly Disagree to Strongly Agree)

3) QA test plan - max 2 pages

For the same function you chose for the usability test, write a QA test plan (check class slides), with sections as follows:

- Test objectives:
    - Product Search and filter functionality
- HW and SW setup (including URL):
    - Mac using using Google Chrome and Safari
- Feature to be tested
    - Product Search and filter functionality
- QA Test plan:

Test #	Test Title	Test Description	Test Input	Expected Output	Test Results
1	Search All	Test find all functionality	none	Get 4 products	PASS
2	Search with Category Filter	Test search functionality with just Category Filter	Category dropdown menu with the category “clothing” selected.	Get 1 product with the name “Nike Air Max”	PASS
3	Search with Category Filter and Query	Test search functionality with Category Filter and Query	Category dropdown menu with category “Electronics” and Query “TV”	Get 1 product with the name “Old TV”	PASS
4	Search All (safari)	Test find all functionality	none	Get 4 products	PASS
5	Search with Category Filter (safari)	Test search functionality with just Category Filter	Category dropdown menu with the category “clothing” selected.	Get 1 product with the name “Nike Air Max”	PASS
6	Search with Category Filter and Query	Test search functionality with Category	Category dropdown menu with	Get 1 product with the name “Old TV”	PASS

	(safari)	Filter and Query	category “Electronics” and Query “TV”		

4) Code Review:

- a) By this time you should have chosen a coding style. In the report say what coding style you chose.

- b) Chose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code under review (or part of it – 2 pages or so MAX) for this function to be peer reviewed, and document this as follows:
    1. One team member should submit code to other team member(s) for peer review.
    2. Peer review should be performed by other group member(s) (1 review is OK).
    3. Peer review is to be done by e-mail and comments are to be included in the code
    4. Submit the e-mail containing or screen shot of the peer review and commented code and e-mail communication related to this in your Milestone 4 document
- Important: It is critical that code reviews are friendly and helpful, intended to help and education, and not to criticize. It is strongly suggested that you use peer review in the development of the whole system. Reviewers should also check for at least minimal code header and in-line comments and flag this as a problem if this is not adequate

Note: peer review must include checking for basic header and in-line comments

a) Our sole coding language for front end and back end is JavaScript. For the backend we use NodeJS and for the frontend we use React with Redux. For development we use the Atom IDE on which we install the linter-eslint package along with eslint-config-airbnb for rules. This essentially gives every developer the same errors not just for syntactical mistakes but for general indentation and structure. All the rules are in the .eslint files.

Search and filter

ZK Zain Khan <zainkhanx93@gmail.com>
To George William Freedland

Reply | Reply All | Forward | ...
Tue 4/21/2020 7:55 PM

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

import MainNavBar from '../Nav/MainNavBar';

const SearchResults = () => {
 const [query, setQuery] = useState('');
 const [filter, setFilter] = useState('');
 const [categories, setCategories] = useState([]);
 const [products, setProducts] = useState([]);

 const handleSearch = () => {
  if (query) {
   if (filter) {
    // code for searching with Search and Filter function ( look up prod by name and category)
    axios.get(`/api/products/${query}/${filter}`).then((res) => {
     setProducts(res.data);
    });
   } else {
    // code for searching with Search function (Can search for anythig by name)
    axios.get(`/api/products/${query}`).then((res) => {
     setProducts(res.data);
    });
   }
  } else if (filter)
   // code for searching with  Filter function ( Eletronics, Apperel, etc)
   axios.get(`/api/products/all/${filter}`).then((res) => {
    setProducts(res.data);
   });
```

Search and filter

ZK Zain Khan <zainkhanx93@gmail.com>
To George William Freedland

Reply | Reply All | Forward | ...
Tue 4/21/2020 7:55 PM

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

import MainNavBar from '../Nav/MainNavBar';

const SearchResults = () => {
 const [query, setQuery] = useState('');
 const [filter, setFilter] = useState('');
 const [categories, setCategories] = useState([]);
 const [products, setProducts] = useState([]);

 const handleSearch = () => {
  if (query) {
   if (filter) {
    // code for searching with Search and Filter function ( look up prod by name and category)
    axios.get(`/api/products/${query}/${filter}`).then((res) => {
     setProducts(res.data);
    });
   } else {
    // code for searching with Search function (Can search for anythig by name)
    axios.get(`/api/products/${query}`).then((res) => {
     setProducts(res.data);
    });
   }
  } else if (filter)
   // code for searching with  Filter function ( Eletronics, Apperel, etc)
   axios.get(`/api/products/all/${filter}`).then((res) => {
    setProducts(res.data);
   });
```

Nav folder should be renamed to Navigation, let's use full variable names when possible so it's easy to manage/go through the application

Comment which component are we sending the products response to

spelling

Every axios request should also have a .catch(err => {}) added after the .get() so in case there is a bad request we can display it neatly on the front instead of giving an error to the browser.

spelling, list all categories

.catch(err => {});

5) **Self-check on best practices for security – ½ page**

- List major assets you are protecting
    - Passwords
- Say how you are protecting each asset (1-2 lines of text per each)
    - We are hashing the user passwords in the backend before saving to the database
- Confirm that you encrypt PW in the DB
    - We are encrypting PW in the database
- Confirm Input data validation (list what is being validated and what code you used) – we request you validate search bar input for up to 40 alphanumeric characters;
    - Inputs are validated in the frontend

6) Self-check: Adherence to original Non-functional specs – performed by team leads

Copy all original non-functional specs as in high level application document published at the very beginning of the class. Then for each say either: DONE if it is done; ON TRACK if it is in the process of being done and you are sure it will be completed on time; or ISSUE meaning you have some problems and then explain it. Note: you must adhere to all original non-functional specs as published in the original high level specification document. Failure to do so may cause reduced SE Product grade

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

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.

9. Google analytics shall be added

   ON TRACK

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)

    ON TRACK

13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

    ON TRACK

14. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only"* at the top of the WWW page. (Important so not to confuse this with a real application).

    DONE