SW Engineering

CSC 648 Section 01

Spring 2017

GatorYard

Team 1:

Ericka Murphy

Aura Davis

Cynthia Chao

Danny Gonzalez

Jimmy Chung

Michael Menlikalew

Phinehas Sridhar

Milestone 4

History Table of Revisions

1. Product Summary:

Hoping to make things easier for San Francisco State University Students, GatorYard aims to provide SFSU students an environment to sell and buy items between each other.

Once GatorYard launches, SFSU students will be able to browse items that other SFSU students have posted and register for an account. Registering for an account allows users to post items for sale and contact one another. However, if a user doesn't have an account, they can still browse what items are for sale, and if they see something they like and they choose to contact the seller, they will be prompted to register or login. When a user views an item for sale, they will see a photo, price, the seller's choice of meeting up on campus for delivery, and a description of that item.

GatorYard also provides sellers with a unique option: the ability to inform buyers that they're able to meet up with them on campus to give them their item.

You can view GatorYard's deployment from www.sfsuse.com/~sp17g01/sp17g01/web.

2. Usability Test Plan:

Test Objectives:

Users will be tasked to use GatorYard's search function and category filter. The users do not need an account to browse item listings, so they will not be prompted to register an account with GatorYard.

Thus, the objectives are:

1. Find the search bar.
2. Type a word into the search bar.
3. View the results.
4. Change the category filter.
5. Type a new word into the search bar.
6. View the narrowed results given by using the category filter.

Test Plan: Post an Item Listing

System setup
The test monitor will be in the same room with the user, but at a distance from the user to give them an atmosphere that they are alone and browsing item listings on their own. The test monitor will observe the user's actions and take notes, but the primary collection of data comes from the Likert scale questions the user completes based on the task.

Starting point
The user will start on GatorYard's homepage, where they will try and use the search function and category filter to find items.

Task to be accomplished
The main task for the user to complete is to identify whether the users can browse the item listings with the search bar and category filters.

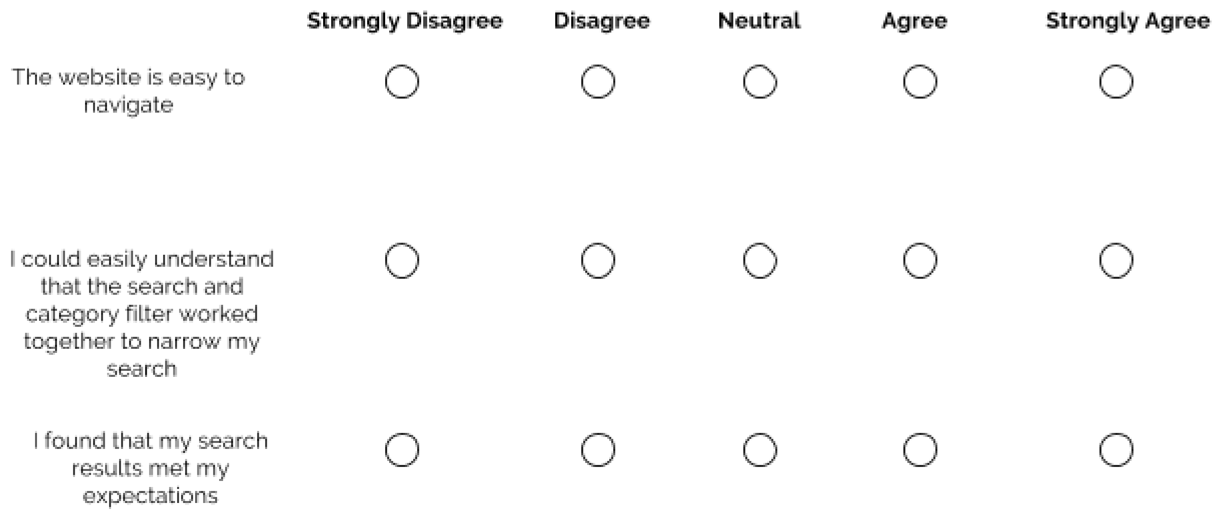
Who is the intended user
The intended users are San Francisco State University students who do not have an account with GatorYard, but will not need an account to browse GatorYard's item listings.

Completion criteria
To complete the task, the user is able to search for items with the search bar and use a category filter to narrow their search. Once the user has finished searching for items, they will answer 3 Likert Scale questions based on their experience.

URL of the system to be tested
www.sfsuse.com/~sp17g01/sp17g01/web

Questionnaire Form:

Below are statements based on the task you completed or did not complete. Please read each statement and answer the extent to which you disagree or agree with the statements.



	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The website is easy to navigate	○	○	○	○	○
I could easily understand that the search and category filter worked together to narrow my search	○	○	○	○	○
I found that my search results met my expectations	○	○	○	○	○

3. QA Test Plan:

Test Objectives:

The user will follow given instructions to see whether or not they get the correct output when they use the search function and category filter. If the user searches for an item without adjusting the category filter (default is 'All'), the search function will look through all the categories that fit the text the user has input into the search bar. However, if the user changes the category (e.g. Electronics) and enters text into the search bar, the search function will look for items that match that category.

HW and SW Setup:

Hardware

- Windows 7 and 10
- Linux
- Mac OS

Software

- Firefox V53.0
- Google Chrome V 58.03
- Internet Explorer V11.1

Feature to be tested:

This QA will test GatorYard's search function and category filter. The search function displays items for sale in the database, and the category filter allows users to narrow their search based on categories GatorYard has in the database.

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results
1	Search	General Search	Samus	Samus Amiibo	Pass
2	Search	Partial Entry	Sam	Samus Amiibo	Pass
3	Search	Consistency Check	Samus	Samus Amiibo	Pass
4	Search	Invalid Input	&*/	Invalid Input	Fail
5	Search	No results	Nukes	No Results	Pass
6	Category Filter	General Search with Category Filter	Category: Furniture Text: Sofa	Sofa	Fail
7	Category Filter	Category Filter	Category: Furniture	Sofa, Small Sofa, Inflatable Bed, Chair	Pass

4. Code Review

Code Being Reviewed

```php
class ItemRepository extends EntityRepository
{
   /**uses $text object to search database for item object using SQL queries to find
a match for a a then returns it. */

   /**
    * @return Item[]
    */
   public function search ($text)
   {
       return $this->createQueryBuilder('item')
           ->andWhere('item.name LIKE :name')
           ->setParameter('name', '%' .$text. '%')
           ->getQuery()
           ->execute();
   }
}

$task = new search();
$task->setSearchText('');

$form = $this->createFormBuilder($task)//This is for the search bar
->add('searchText', TextType::class)
   ->add('save', SubmitType::class, array('label' => 'Search'))
   ->getForm();

if ($request->isMethod('POST')) {
   $formData = $request->request->get('form');
   $items = $repository->search($formData['searchText']);
}
```

Review of Code

Hello Aura,

I am very impressed with all the hard work you've put into the project. I've reviewed the code you've sent me. I am wondering why you named the variable $task, task, when it is a class that holds the search text. Would it be possible to create another search function one that uses both categories, by using another 'where' statement in the query builder? something like ->where(item.category = cat) , where cat would be the passed in category.

another idea is we could just use one search function that takes in two parameters, ($cat and $text). first we would check if (cat == 'all categories') if so we would just return what you have in your code. If not we would return the query with the extra 'where' statement. That is just an idea.

Though the search bar works thanks to you. I will be happy to work with you this thursday. Keep up the amazing work.

Thanks,
Danny

5. Self-Check:

Assets we are protecting:

1. User Data
2. Server Security

Best Practices for Security

Encrypting passwords into the database is currently underway. At the moment, users cannot sign up for GatorYard, so if a user tries to sign up, the information they give does not go into the database.

Input data validation is also currently underway, but GatorYard will ensure that users cannot inject SQL statements to access our database (i.e. search bar input) and that email registrations will check for the string "sfsu.edu."

6. Non-Functional Specs:

Spec	Status
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 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, Chrome.	ON TRACK
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	ON TRACK
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	ON TRACK
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
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)	ON TRACK
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development	ON TRACK
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