SW Engineering CSC 648/848 Fall 2014

"Listed"

Local Group: 04

Lead: Dhruvkumar Joshi joshi@mail.sfsu.edu
Gilbert Szeto gszeto@mail.sfsu.edu
Kumari Sweta ksweta@mail.sfsu.edu
Jordan Guinn jguinn@mail.sfsu.edu
Yefa Qi kevinqiyf@hotmail.com

Milestone 4 -Beta Launch

12/12/2014

History Table:

Date	Comments	Version
12/12/14	**First initial writeup by the team**	**1.0**
12/15/14	**Revisions by the team after Professor Petkovic's comments**	**1.1**

1) Product Summary:

"Listed" is a real estate web application built for today's realtors, allowing them to manage their listed properties. This application allows a particular real estate company to post the contents of residential listings, which will viewable by potential buyers. These future homeowners will then have the ability to contact the realtors about such homes.

This product is the real estate application for the experienced and inexperienced, with functionality for both. The website can be easily navigated by those without experience looking for homes. It displays the location of the properties being looked at, allowing the user to determine if the neighborhood they'd be living in fits their lifestyle. It also provides contact information for multiple realtors, more than capable of assisting with the process of buying property. Finally, it simplifies the job of the realtor with its step-by-step process of creating listings. The strong usability on boths sides is what makes "Listed" stand out among its competitors.

The following are the features Listed shall include:

1. Listings shall contain the following:
    a.) Picture(s) of the property
    b.) Property description (address, city, zip code, price)

2. Unregistered users shall be able to search listings by:
    a.) Desired zip code
    b.) Desired city

3. Unregistered users shall be able to create a personal profile on the website, thereby becoming registered users.
    a.) Profiles shall require full name & email address.

4. Registered users shall be able to specifically describe attributes of homes they find interesting.
    Attributes shall include:
    a.) Number of bedrooms
    b.) Price
    c.) Size of the house

5. Registered users shall be able to contact realtors about properties

they're seriously considering.

6. A search engine shall be available within the context of each page.

7. Sellers shall be able to provide realtors with information about their home.

8. Realtors shall be able to post listings.

9. Realtors shall be able to modify listings.
    a.) Realtors shall only have access to their listings.

10. Admin will have comprehensive access to all listings.
    a.) Admin shall have authority to edit listings if content is deemed inappropriate in any way.

11. Unregistered users shall be able to determine house locations with Maps.
    a.) Specific house shall be highlighted on map, with surrounding neighborhood viewable as well.

12. All users can read the website's privacy policy.

Our product is located at the following URL: "http://sfsuswe.com/~f14g04/".

2) Usability Test Plan:

Test Objective:

The objective of this test is to evaluate the usability of the search function and to gather information about how to provide a better user experience. This test plan will be based on the use case where an unregistered user will be looking to search the site for available homes they may want to purchase in the future. The task will be described below for an evaluator to test and assess the usability of the search function.

Test Plan:

System Setup:

- The system shall be able to run any of the following browsers with reasonable speed:
    - Mozilla Firefox
    - Google Chrome

    - Microsoft Internet Explorer
    - Apple Safari
- The system shall have access to internet.
- The system shall have a keyboard, a mouse, and a standard display.

Start Point:

- The homepage of the Listed site: http://sfsuswe.com/~f14g04/

Task to be accomplished:

- Attempt to search for available homes based on zipcode and city.
- Answer the accompanying questionnaire at the bottom.

Intended user:

- An unregistered user who has general knowledge of using a web browser and is looking to find real estate properties.

Completion criteria:

- Correctly enter a search query and end up in the search results page where the results are presented to the user.
- Fill out the questionnaire.

URL to be tested:

- http://sfsuswe.com/~f14g04/

Questionnaire:

1. The search function was easy to find. (Check one)

○ Strongly Agree ○ Agree ○ Neutral ○ Disagree ○ Strongly disagree

Comments:

2. It was easy to search for the item(s) you were looking for. (Check one)

○ Strongly Agree ○ Agree ○ Neutral ○ Disagree ○ Strongly disagree

Comments:

3. Browsing search results was easy. (Check one)

○ Strongly Agree ○ Agree ○ Neutral ○ Disagree ○ Strongly disagree

Comments:

4. The organization of the search results page was well presented. (Check one)

○ Strongly Agree ○ Agree ○ Neutral ○ Disagree ○ Strongly disagree

Comments:

3) QA Test Plan:

Test Objective:

The objective of this test is to walk through each individual step of the search function and ensure that all steps are functioning as designed. In this test, the unregistered user will search with both valid and invalid search criteria. Upon completion of a successful search, the user will then attempt to filter the results correctly as well as incorrectly. At this point, the user will get to the detailed page of a particular listing. The test will be described below in further detail.

Hardware/Software Setup:

Basics:

- The system shall requite Internet access, running on any of the following browsers:
  - Mozilla Firefox
  - Google Chrome
  - Microsoft Internet Explorer
  - Apple Safari
- The system shall have a keyboard, a mouse, and a standard display.

Starting Point:

- The tester shall begin at the home page of the Listed site: http://sfsuswe.com/~f14g04/

Tasks to be Accomplished:

- The tester shall provide invalid information into the search query.

- The tester shall look up all homes in “San Francisco”.
- The tester shall attempt to filter those results with impossible minimum and maximum prices.
- The tester shall then filter the same results with correct specifications.
- The tester shall finally reach the main page of a specific listing.

Completion Criteria:

- Correctly enter a search query and end up in the search results page where the results are presented to the user.
-

Expected Time to Completion:

- 3 minutes

Specified URL:

- http://sfsuswe.com/~f14g04/

Feature to be Tested:

The search function is the feature being reviewed in this test. The following are the components to be included within the testing:

- Search with Invalid Criteria
- Search by City
- Proper Filtering of Search Results
- Clickable search results’ images

Test Cases:

Test 1: Search with Invalid Criteria

Actions:

a. Go to http://sfsuswe.com/~f14g04/
b. Find the search bar located in the top right-hand corner of the page
c. Type in “Hello” in the search bar and press Submit.

Expected:

- No listings appear on the resulting page

Pass/No Pass:

Test 2: Search by City

Actions:

a. Go to http://sfsuswe.com/~f14g04/
b. Find the search bar located in the top right-hand corner of the page
c. Type in “San Francisco” and press Submit.

Expected:

- 5 listings appear on the resulting page
- “San Francisco” remains in the search bar on the resulting page

Pass/No Pass:

Test 3: Improper Filtering of Search Results

Actions:

a. Proform Test 2: Search for key words
b. On the resulting page, go to the “Refine Filter” table
c. Select “600,000” as the minimum price and select “$500,000” as the maximum price.
d. Press Submit.

Expected:

- No matched listings appear on the resulting page

Test 4: Proper Filtering of Search Results

Actions:

a. On this page of listings, go to the “Refine Filter” table
b. Select “300,000” as the minimum price and select “$700,000” as the maximum price.
c. Press Submit.

Expected:

- 2 listings appear on the resulting page
- The listings have respective addresses of “1600 Holloway Ave” & “244 El Campo Dr”

Pass/No Pass:

Test 5: Access Specific Listing Page

Actions:

a. Perform Test 2 or Test 3
b. Click on the picture of the listing with the address “1600 Holloway Ave”.

Expected:

- The resulting page has a larger, more descriptive view of the single listing with the address “1600 Holloway Drive”

Pass/No Pass:

QA Test Results

Test 1: Search with Invalid Criteria

Actions:

a. Go to http://sfsuswe.com/~f14g04/
b. Find the search bar located in the top right-hand corner of the page
c. Type in “Hello” in the search bar and press Submit.

Results:

- Next page has no listings displayed

Test 2: Search by City

Actions:

a. Go to http://sfsuswe.com/~f14g04/
b. Find the search bar located in the top right-hand corner of the page
c. Type in “San Francisco” and press Submit.

Results:

- Next page has five listings displayed
- “San Francisco” remains in the search bar on the page

Test 3: Improper Filtering of Search Results

Actions:

a. Perform Test 2: Search for key words
b. On the resulting page, go to the “Refine Filter” table
c. Select “600,000” as the minimum price and select “$500,000” as the maximum price.
d. Press Submit.

Results:

- Next page has no listings displayed

Test 4: Proper Filtering of Search Results

Actions:

a. On this page of listings, go to the “Refine Filter” table
b. Select “300,000” as the minimum price and select “$700,000” as the maximum price.
c. Press Submit.

Results:

- Next page has two listings displayed
- Those two listings have respective addresses of “1600 Holloway Ave” & “244 El Campo Dr”

Test 5: Access Specific Listing Page

Actions:

a. Perform Test 4
b. Click on the picture of the listing with the address “1600 Holloway Ave”.

Results:

- Next page has a larger, more descriptive view of the single listing with the address “1600 Holloway Drive”

4) Code Review:

Initial Email

Subject: M4 Search Function Code Review

From: Gilbert Szeto <gszeto@sfsuswe.com>

Date: 12/12/2014 2:33 PM

To: ksweta@sfsuswe.com

Hello Kumari,

For M4 code review can you review the search functionality code from the listings.php controller:

```
// Using the POST method
// Consider using the '/listings/search/94114 san francisco' MVC way
public function search()
```

```
        {
        /*
        * select property.name,
        *       property.price,
        *       property.size,
        *       property.num_room,
        *       property.unit_type,
        *       address.street_address,
        *       address.city,
        *       address.zipcode
        * from `listing`
        * left join `property` on listing.property_id = property.id
        * left join `address` on property.address_id = address.id;
        */
        // check for cases where people search using
sfsuswe.com/~f14g04/Listed/listings/search/94112
        if (func_num_args())
        {
        $search_string = func_get_arg(0);
        }

        if (isset($_POST["submit_search"]))
        {
        $search_string = ($_POST["search_city_zip"]);
        }

        $listings_model = $this->loadModel('ListingsModel');
        $property_image_model = $this->loadModel('PropertyImagesModel');
        $listings_search_results = $this->getResults($listings_model, $search_string);

        foreach ($listings_search_results as $listing)
        {
        $listing->images =
$property_image_model->getAllImagesOfProperty($listing->property_id);
        }
        require 'application/views/_templates/header.php';
        require 'application/views/listing/search.php';
        require 'application/views/_templates/footer.php';
        }

        // Take user input search string and either search by zip code or
        // search by city, depending on user input
        private function getResults($model, $search_string)
        {
        $pattern_zip_codes = "/(\040*)(\d{5})(\040*)/"; // Five digits with 0 or more spaces trailing
        $pattern_words = "/[^a-zA-Z\040]*/"; // Get any spaces and alpha characters
        if (preg_match($pattern_zip_codes, $search_string))
        {
        preg_match_all($pattern_zip_codes, $search_string, $zip_code);
        $results = $model->getListingByZipcode($zip_code[0][0]);
        $this->removeColumns($results);
        //var_dump($results);
        return $results;
        } elseif (preg_match($pattern_words, $search_string))
```

```
{
// remove all non-word characters except spaces
$filtered_string = preg_replace("/^[\W\040\d]*/", "", trim($search_string));
//echo "Filtered search string: " . $filtered_string;
if (empty($filtered_string))
{
return array();
}
$results = $model->getListingByCity($filtered_string);
//$this->removeColumns($results);
return $results;
} else
{
return array();
}
}
```

Thanks for your help.
-Gilbert

Response

Subject: Re: M4 Search Function Code Review

From: ksweta@sfsuswe.com

Date: 12/12/2014 3:14 PM

To: "Gilbert Szeto" <gszeto@sfsuswe.com>

Hi Gilbert,

Thanks for sending email. The code works fine. However, following are some suggestions:

1.line 226: do not use superglobal $_POST array directly. Instead try using some filtering function, in order to avoid injection attack.

2.It is recommended to keep lines at approximately 75-85 characters long for better code readability. There are multiple line with more than 90 characters for e.g line no. 235,255

3.For the following code

```
    if (func_num_args())
{
   $search_string = func_get_arg(0);
    }

if (isset($_POST["submit_search"]))
{
   $search_string = ($_POST["search_city_zip"]);
}
```

it's better to use if else.

4.I don't see any reason to include these comments in the code.

```
/*
 * select property.name,
 *        property.price,
 *        property.size,
 *        property.num_room,
 *        property.unit_type,
 *        address.street_address,
 *        address.city,
 *        address.zipcode
 * from `listing`
 * left join `property` on listing.property_id = property.id
 * left join `address` on property.address_id = address.id;
 */
```

5.I also see lot of code commented, which are not used for e.g:

```
//var_dump($results);
//echo "results class type: " . get_class($results[0]);
//fetchAll(PDO::FETCH_CLASS, "Listing")
// Need to implement class PDO:FETCH_CLASS
//echo "first: " . $results[0]->id;
```

Because we have version control, we can always get back previous code , if needed.

6.Also, for multi line comments, it's better to use block comments instead of single line comments.

It would be good if you can make these changes.

Thank you,
Kumari Sweta

Final Email

Subject: Re: M4 Search Function Code Review

From: Gilbert Szeto <gszeto@sfsuswe.com>

Date: 12/12/2014 3:40 PM

To: ksweta@sfsuswe.com

Thanks for your quick feedback.

I shall take your comments into consideration and refactor the search functionality to reflect your suggestions.

-Gilbert

5) Risk Assessment:

Schedule Risk:

This may be our most significant risk to date. With the final deadline rapidly approaching and each one of our team members involved in other scholastic obligations, scheduling meetings with all team members has proven quite difficult. To solve this issue, we have been working remotely to increase productivity.

Skills Risk:

There is a major risk with front-end development, as the team lacks any substantial experience in that field. But Dhruv and Jordan are more than willing to make up for this deficiency with plenty of hard work and quick learning.

Technical Risk:

The team has developed a substantially thorough understanding of all the software components used within this project. We do not see any technical risks at this point.

Teamwork Risk:

We do not see any teamwork risks. Everyone is very passionate about their part of the project and ready to help the rest of the team.

Legal Risk:

We are using “FREE”, “GPL”, “MIT Licensed” and “OPEN SOURCE” products in our Project. We will use the LinkedIn and Google Maps APIs, which are free. Consequently, no legal risks are currently present.