SW Engineering CSC648/848 Section 02 Fall 2017

UNLOCK.COM

Team 01

Sannuj Singhal (ssinghal@mail.sfsu.edu)
Joseph Wibowo
Paul Fontaine
Daniel Herrera
Gurchetan Johal
Steven Huynh

Date:	Comments:
12/8/17	First draft for review.

Table of Contents

1. Product Summary……... 2
2. Usability Test Plan……. 3
3. QA Test Plan…….. 4
4. Code Review…….. 6
5. Self Check on Best Practices for Security……. 9
6. Self-Check: Adherence to Original Non-Functional Specs…….. 10

1. Product Summary

The name of our product is unlock.com. Through unlock.com we aim to provide the simplest home buying experience available in the market today. Though our website does not have a unique feature to set us apart from the competition, we are sure that the simplicity of using our website will attract a large user base.

These are the basic features we have implemented for our users:

1. An unregistered buyer/seller will be able to view featured houses on the home page.
2. An unregistered buyer/seller will be able to search for a house based on city name or zip-code.
3. An unregistered buyer/seller will be able to view the number of results that are available after using the search option.
4. An unregistered buyer/seller will be able to view a property listing.
5. A listed property will have pictures of it on display for unregistered buyers/sellers to see.
6. A listed property's location will be shown on a map for unregistered buyers/sellers to see.
7. An unregistered buyer/seller will be able to sign-up for the website.
8. A registered buyer/seller will be able to do all the things described above.
9. A registered buyer/seller will be able to log-in to the website.
10. A registered buyer will be able to contact a real estate agent after viewing a property listing.
11. A registered seller will be provided a personalized dashboard.
12. A registered seller will be able to post new property listings.
13. The admin, using workbench, will be able to delete user accounts.
14. The admin, using workbench, will be able to delete property listings.

You may view unlock.com at the following link: sfsuse.com/fa17g01

2. Usability Test Plan

2.1 Test Objectives

Our main objective, in regards to testing the search function on unlock.com, is the following:

1. To ensure that the search function works properly and as expected.

2.2 Test Plan

We plan on using a Windows laptop, running Windows OS, to test our website, unlock.com, at the following web address: sfsuse.com/fa17g01.

A member of our team will be acting as an unregistered user and input a city or zip-code in the search bar. Since they, the member of our team, knows what information is contained within the database, they will be able to tell when the query result returned is correct or incorrect.

2.3 Questionnaire Form

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
1. The search bar query returns a result.					
2. The search bar clearly tells you what to input.					
3. The search bar is relatively easy to use.					

3. QA Test Plan

3.1 Test Objectives

Our main objectives, in regards to testing the search functionality of unlock.com, are the following:

1. Check whether the unregistered user can search by city.
2. Check whether the unregistered user can search by zip-code.
3. Check whether the correct property listings display when we run our query.

3.2 HW and SW Setup

***Hardware Setup**:*

The hardware being used to test the website is a Windows laptop running Windows 10 OS, and contains an Intel i5 processor.

***Software Setup**:*

The softwares being used to test unlock.com are Google Chrome Version 62.0.3202.94 and Mozilla Firefox Version 57.0.2.

3.3 Feature to be tested

The feature we plan to test is the search bar on unlock.com. We are doing this so that we may ensure the three test objectives that have listed in section 3.1 of this document work as desired and expected.

3.4 Actual Test Cases

Google Chrome

Test #	Test title (description)	Test input	Expected output	PASS/FAIL
1	Check by city (the unregistered user will input name of city in the	'oakland' 'san francisco' 'san leandro'	Property listings being displayed from Oakland, San Francisco, and San	**PASS**

	search bar)		Leandro, respectively	
2	Check by zip-code (the unregistered user will input zip-code of city in the search bar)	'94100' '94132'	Property listings being displayed from Oakland and San Francisco, respectively	**FAIL**
3	Check for listing display (the unregistered user will see the properties that have been listed, and the search result number will correlate with the number of properties that have been displayed)	'oakland' 'san francisco' 'san leandro' 'san lorenzo'	Property listings from Oakland, San Francisco, and San Leandro should display. San Lorenzo should return 0 results.	**PASS**

Mozilla Firefox

Test #	Test title (description)	Test input	Expected output	PASS/FAIL
1	Check by city (the unregistered user will input name of city in the search bar)	'oakland' 'san francisco' 'san leandro'	Property listings being displayed from Oakland, San Francisco, and San Leandro, respectively	**PASS**
2	Check by zip-code (the unregistered user will input zip-code of city in the search bar)	'94100' '94132'	Property listings being displayed from Oakland and San Francisco, respectively	**FAIL**
3	Check for listing display (the unregistered user will see the properties that have been listed, and the search result number will correlate with the number of properties that have been displayed)	'oakland' 'san francisco' 'san leandro' 'san lorenzo'	Property listings from Oakland, San Francisco, and San Leandro should display. San Lorenzo should return 0 results.	**PASS**

4. Code Review

4.1 Coding Style

Our basic coding style focuses on clean coding practices. The indented code correlates with its related bracket, thus making readability of code very easy and clear.

4.2 Peer Review of Code

The following snippet of code was emailed from Sannuj Singhal to Gurchetan Johal for review:

```
var express = require('express');
var router = express.Router();

router.get('/', function(req, res, next) {
  var word = req.query.word;
  var mysql = require('mysql');
  var connection = mysql.createConnection({
    host     : "localhost",
    user     : "fa17g01",
    password : "csc648fa17g01",
    database : "fa17g01"
  });

  connection.connect(function(err) {
    if (err) throw err;
    console.log("Connected!");

    var sql = "SELECT * FROM houses WHERE city LIKE '%" + word + "%'";
    connection.query(sql, function (err, result, fields)
    {
      res.render('results.ejs', {word: word, result: result});
    });
  });
});

module.exports = router;
```

Figure 1: search.js

The code shows how the search function was executed on unlock.com.

Gurchetan Johal reviewed the code, and offered some very good input in regards to how we may improve the security on unlock.com.

The following images will show the e-mail exchange that occurred between Sannuj Singhal and Gurchetan Johal to ensure proper review of code displayed in figure 1.

Figure 2: Initial e-mail sent by Sannuj Singhal to Gurchetan Johal

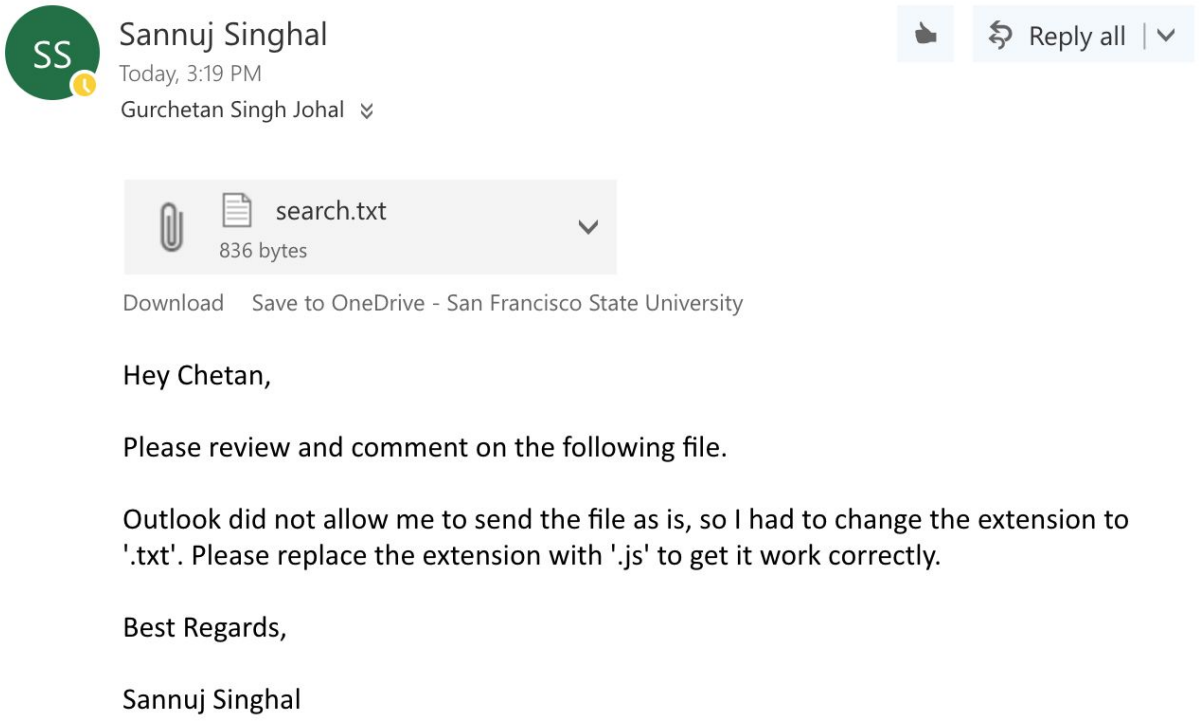
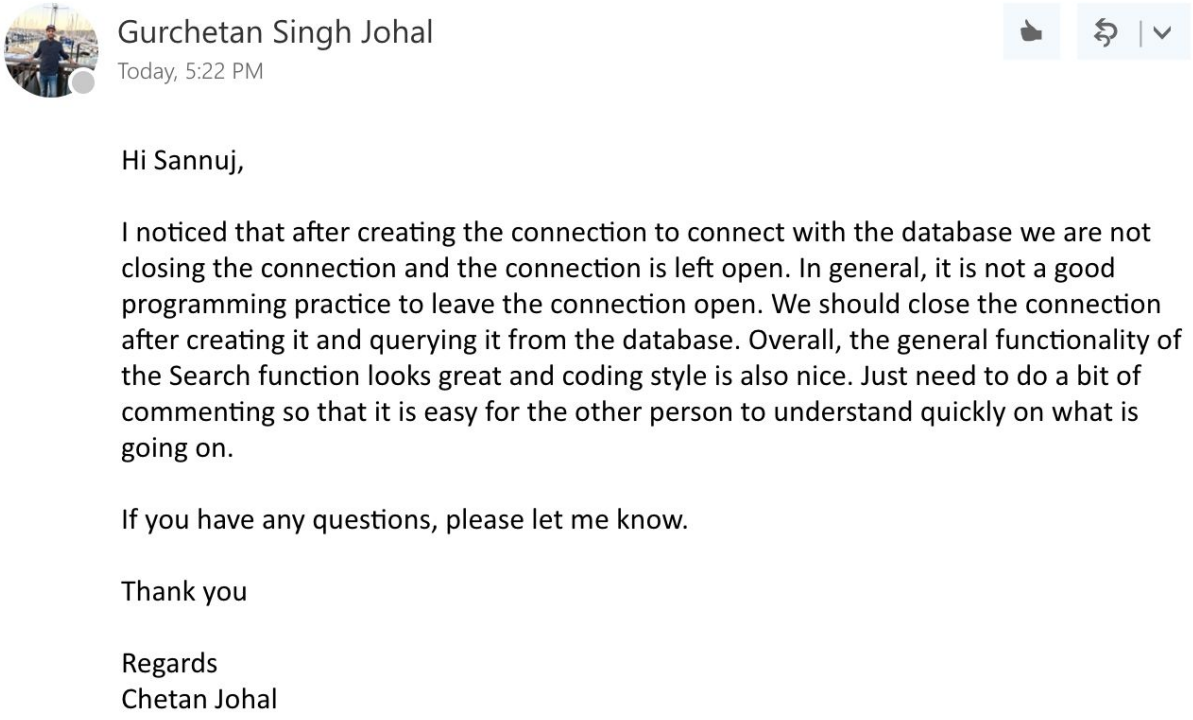


Figure 3: Gurchetan Johal's response to Sannuj Singhal



Gurchetan Singh Johal
Today, 5:22 PM

Hi Sannuj,

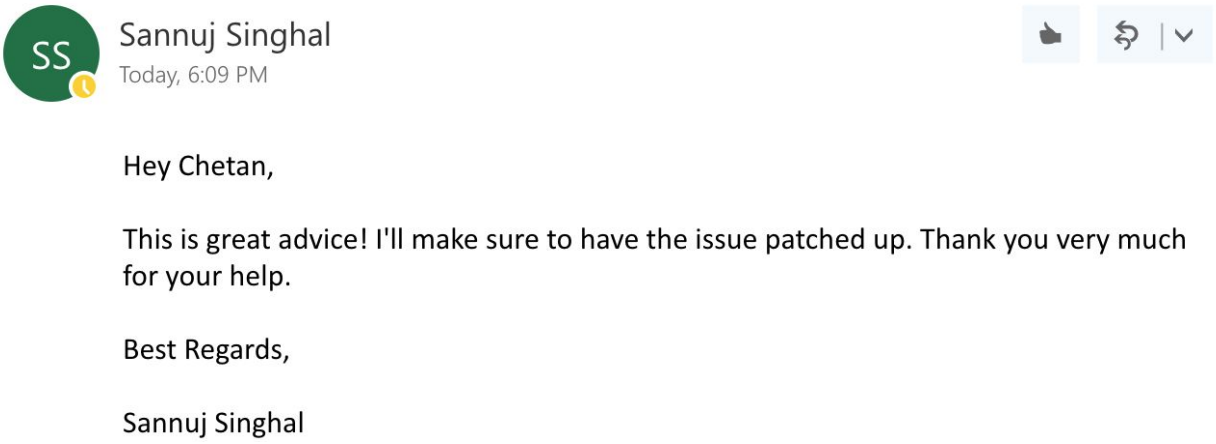
I noticed that after creating the connection to connect with the database we are not closing the connection and the connection is left open. In general, it is not a good programming practice to leave the connection open. We should close the connection after creating it and querying it from the database. Overall, the general functionality of the Search function looks great and coding style is also nice. Just need to do a bit of commenting so that it is easy for the other person to understand quickly on what is going on.

If you have any questions, please let me know.

Thank you

Regards
Chetan Johal

Figure 4: Sannuj Singhal's final response to Gurchetan Johal, signaling end of email exchange.



Sannuj Singhal
Today, 6:09 PM

Hey Chetan,

This is great advice! I'll make sure to have the issue patched up. Thank you very much for your help.

Best Regards,

Sannuj Singhal

5. Self Check on Best Practices for Security

5.1 Major assets we are protecting

We are protecting the following major assets, for the following types of users:

Registered Buyer:

1. First Name
2. Last Name
3. Email
4. Password

Registered Seller:

1. Email
2. Password

5.2 Confirmation of Password Encryption and Input Data Validation

Password Encryption:

Our implementation encrypts the registered buyer and seller's password using the Message Digest 5 (MD5) method.

Input Data Validation:

Input data is being checked in all of our forms and as well as the search bar, so that bad requests and SQL injections are not possible.

6. Self-Check: Adherence to Original Non-Functional Specs

1	Application shall be developed and deployed using class provided deployment 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 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 e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.	DONE
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	DONE
6	Data shall be stored in the MySQL database on the class server in the team's account	DONE
7	Application shall provide real-estate images and optionally video	DONE
8	Maps showing real-estate location shall be required	DONE
9	Application shall be deployed from the team's account on AWS	DONE
10	No more than 50 concurrent users shall be accessing the application at any time	DONE
11	Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.	DONE
12	The language used shall be English.	DONE
13	Application shall be very easy to use and intuitive. No prior training shall be required to use the website.	DONE
14	Google analytics shall be added	ON TRACK
15	Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services.	DONE

16	Pay functionality (how to pay for goods and services) shall not be implemented.	DONE
17	Site security: basic best practices shall be applied (as covered in the class)	DONE
18	Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development	DONE
19	The website shall prominently display the following text on all pages *"SFSU Software Engineering Project, Fall 2017. For Demonstration Only"*. (Important so as to not confuse this with a real application).	ON TRACK

Note on 3: Application is correctly hosted and deployed on class provided server. However, it is not on AWS server, as that was replaced by Professor Souza at the beginning of the semester.

Note on 9: Application is correctly hosted and deployed on class provided server. However, it is not on AWS server, as that was replaced by Professor Souza at the beginning of the semester.