

# SW Engineering CSC648/848 Summer 2020

## Chillow

## Team 6

### Team Members:

Valdemar Puente-Gonzalez (Team Lead)

[vpuentegonzalez@mail.sfsu.edu](mailto:vpuentegonzalez@mail.sfsu.edu)

Daniel Belmeur (Front-End Lead)

Sean Ellis

Philip Smith (Back-End Lead)

Salah Ali

Karan Gurung (Git Master)

### Milestone 4

3 August 2020

Revisions:
5 August 2020

---

## ***Table of Contents***

---

Product Summary.....	3
Usability Test Plan .....	4
QA Test Plan.....	5
Code Review.....	7
Self-Check: Security .....	9
Self-Check: Adherence of Non-functional Specs.....	10

---

## *Product Summary*

---

Chillow

Final P1 Functions:

- Users shall be able to browse homes on our marketplace
- Users shall be able to search and browse by categories (Pets, Distance from Campus, etc)
- Users shall be able to filter by price
- Users shall be able to filter by address
- Users shall be able to filter by zip code
- Users shall be able to filter by numbers of bedrooms
- Users shall be able to filter by number of bathrooms
- Users shall be able to filter by square footage
- Users shall be able to filter by type of view (City, Waterfront View, Suburb, etc)
- Users shall be able to filter by type of parking
- Users shall be able to see rental listing information
- Users shall be able to register for an account
- SF State students/faculty shall be able to send messages to landlords
- Registered users have all the functionality of a non-registered user
- Registered users shall be able to login to their account
- Registered users shall be able to post listings
- Registered users shall be able to reset their username or password
- Registered users shall have all personal sensitive data save on our servers encrypted
- Admin shall be required to approve all posts
- Admin shall be able to ban suspicious user
- Admin shall be able to block suspicious post

<http://chillo-env.eba-btvmjin2.us-west-1.elasticbeanstalk.com/>

---

## *Usability Test Plan*

---

### **Test Objectives:**

For our usability test we are testing the search function. We chose the search function because of the filters that have been created. Each filter in itself is a specific feature, in which we have confidence in their reliability but wish to confirm this. This will be accomplished through a series of simple instructions that describe to the user what to do. Each test will assess the efficiency of the search function through user inputs.

### **Test Background and Setup:**

The setup for the test will be simple single room setup with a working computer or laptop connected to the internet. There will be an observer present to evaluate and problem solve any technical issue.

The starting point for the test will be to receive a set of tasks and descriptions. The user will then follow the descriptions to the best of their knowledge.

This is the URL to the website that will be tested. <http://chillo-env.eba-btvmjin2.us-west-1.elasticbeanstalk.com/>. Here the intent is to test the functionality of the search function. This will be measured by the effectiveness of the search and its filters. Each filter provides its own search functionality without the need to enter in text. The text input of the search function should return results related to the entered text. The results of the test will be measured by a Likert Test.

### **Usability Task Description:**

1. Navigate to the listings page. Type in the search field “San Francisco”.
2. Navigate to the listings page Click on the more function and type “10” in the distance filter.
3. Navigate to the listings page. Click on the max amount field and type in “3000”

Here we want to test effectiveness by the completion rate of the task.

Second would be by how easily the description are to follow.

Lastly, we will measure it by the search results.

The efficiency will be measured by the time to complete the tasks.

How many clicks it takes to complete the task.

Lastly, the number of pages one must navigate to complete the task.

Likert Scale Questions:

- I found the instructions easy to follow (check one)

☐ Yes ☐ No

- I found the site easy to navigate (check one)

☐ Strongly disagree ☐ Disagree

☐ Neither agree or disagree

☐ Agree ☐ Strongly agree

- I found search function easy to use (check one)

☐ Strongly disagree ☐ Disagree

☐ Neither agree or disagree

☐ Agree ☐ Strongly agree

---

## ***QA Test Plan***

---

### **Test Objectives:**

For our usability test we are testing the search function. We chose the search function because of the filters that have been created. Each filter in itself is a specific feature, in which we have confidence in their reliability but wish to confirm this. This will be accomplished through a

series of simple instructions that describe to the user what do. Each test will assess the efficiency of the search function through user inputs.

### **HW and SW Setup:**

Hardware setup will include a laptop or PC with internet access. This system may use which ever preferred browser they would want to use (e.g. Chrome, IE, Firefox, more). The set up will also include the use of a single room with monitors readily available to help if any technical issues should arise. Finally, they will be given a set of tasks to complete alongside the site URL:

<http://chillo-env.eba-btvmjin2.us-west-1.elasticbeanstalk.com/>

### **Feature to be Tested:**

The search function

### **QA Test Plan:**

<b><u>TEST#</u></b>	<b><u>TEST TITLE</u></b>	<b><u>TEST DESCRIPTION</u></b>	<b><u>TEST INPUT</u></b>	<b><u>EXPECTED CORRECT OUTPUT</u></b>	<b><u>TEST RESULTS (Pass/Fail)</u></b>
<b><u>1</u></b>	Text Test	Test % like. In search for name in field	Type “San Francisco” Into the search field	Return 3 results, each with “San Francisco” in the name field	1. PASS 2. PASS
<b><u>2</u></b>	Distance Test	Test the distance from the school using the distance filter	Type in “10” into the distance filter	Returns 2 results, that are within 10 miles	1. PASS 2. PASS
<b><u>3</u></b>	Price Test	Test % like. In the search for numbers in field	Type in “3000” in the max price field	Returns 2 results, that are listed under 3000	1. PASS 2. PASS

---

## *Code Review*

---

**From:** Phillip Anthony Smith <psmith4@mail.sfsu.edu>  
**Sent:** Monday, August 3, 2020 10:55 PM  
**To:** Valdemar Puente-Gonzalez <vpuentegonzalez@mail.sfsu.edu>  
**Subject:** Re: Code Review

The search algorithm is written well and organized. There is a header that explains the feature and describes what the module accomplishes. There are good inline comments on blocks that need explanation and I would suggest adding more within the '/search' route since the function block is very long and could be difficult to follow for others. Another suggestion I would make is to ensure tab consistency with the rest of the project which is 2 space length for javascript files typically. The search is a pivotal feature for the application and it is clear from the complexity and care put into the code that good work has been put into it with only minor stylistic improvements to be made. Lastly, to lighten the load, let's make sure we are removing any unused code where we can and shrink the module for best readability and conciseness.

- Phillip Smith

---

**From:** Valdemar Puente-Gonzalez <vpuentegonzalez@mail.sfsu.edu>  
**Sent:** Monday, August 3, 2020 10:43 PM  
**To:** Phillip Anthony Smith <psmith4@mail.sfsu.edu>  
**Subject:** Code Review

Hey Phillip,

I have included some code for you to review. This is code of our search algorithm and would really appreciate your timely feed back. Please follow the guidance below on how to do it.

Code Review:

- a) By this time you should have chosen a coding style. In the report say what coding style you chose.
  1. Chose the code (substantial portion of it) related to the feature you used for QA and usability test. 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. What to submit? Submit the e-mail containing review or screen shot of the peer review and commented code

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

Thank you,

Valdemar



---

## *Self-Check: Security*

---

Major Assets: -

- User information
- Messaging access
- Passwords
- Listings data

Major Threats: -

- Unauthorized dashboard access
- Unauthorized database access
- Unauthorized message access
- Sending unauthorized messages
- Inappropriate listing data appearing on page

Protection Plan: -

- User information cannot be accessed through site functions unless authentication is established. Routes and API functions are protected by passport authentication middleware.
- Form fields are validated and limited as appropriate. emails must be formatted as standard email structure. Only emails containing “@mail.sfsu.edu” can gain verified status to access messaging lists. Passwords must be 8 characters.
- Passwords are hashed when sent over http and before entering the database. No plaintext passwords will be stored.
- All listings must be approved by an administrator before being available to be viewed by users
- All listings, filing systems with images will be protected through an admin block in the database

---

## *Self-Check: Adherence of Non-functional Specs*

---

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agree in M) (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 - DONE
7. The language used shall be English (no localization needed) - DONE
8. Application shall be very easy to use and intuitive - DONE
9. Google analytics shall be used - DONE
10. No e-mail clients shall be allowed. Interested users can only message to seller's via in-site messaging. One round of messaging (from user to seller) is enough for this application – ON TRACK
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) for main data items - DONE
13. Media formats shall be standard as used in the market today - DONE
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development - DONE

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