SW Engineering CSC648/848 Section 02 Fall 2017

Essos

Real Estate Web Application

Milestone 4

December 8, 2017

Team 06

Alberto Mancini, Team Lead | amancini@mail.sfsu.edu

Julian Morrisette

Mayank Sachdeva

Jirat Parkeenvincha

Felix Chin

Shenliang Wang

Revisions

December 8, 2017	Initial Draft

Table of Contents

Product Summary

Essos is a real-estate web application that strives to be a simple to use listing platform for real-estate agents and home seekers. With *Essos*, agents and prospective buyers can connect easily over a new property without jumping through any unnecessary hoops.

Essos provides a simple and easy to use set of features:

- All guests can view new featured listings right from the Essos Home page
- Anyone can easily search for properties by typing in a city name or zipcode in the search bar
- Listings provide plentiful property information and images with Google Maps integration
- Registered Users may contact Agents with just the click of a button from any listing page
- Agent accounts have access to a simple to use dashboard making it simple to create new listings, delete listings, and view messages from interested buyers

The simple-to-use platform that *Essos* provides makes taking the first steps to buying any property easier - all it takes is a simple button press.

Explore the *Essos* real-estate web application at: https://sfsuse.com/fa17g06/

Usability Test Plan

Test Objective

- The objective of this test is to gauge the usability of the site's main function: Search.
- Expose flaws with the Search function by asking our testers to perform the search task with no prior instructions. The testers will need to intuit how to use the Search function in order to search for the targeted queries.

Test Plan

System setup

- Tester will log into a computer with access to the internet and access to the following accepted internet browsers: Google chrome or Firefox.

Starting point

- Tester will begin at the site homepage wherein a search button will be present.

Task to be accomplished

- Tester will attempt a search for listings in a city of their choosing.
- Tester will attempt a search for listings with a zip code of their choosing.

Intended Users

- The targeted user is broad, as any person who is searching for a property is a potential customer.
- Users will be assumed to have no technical expertise and may be a complete novice to web browsing.

Completion criteria

- Tester is satisfied that they have accomplished the search query they attempted. This is evaluated from a post test questionnaire.

URL of System

- https://sfsuse.com/fa17g06/

Usability Questionnaire

I found the Search function easy to use (Check one)

__ Strongly Disagree __ Disagree __ Neither Agree or disagree

__ Agree __ Strongly Agree

I found the listings for the City I was searching for (Check one)

__ Strongly Disagree __ Disagree __ Neither Agree or disagree

__ Agree __ Strongly Agree

I found the listings for the zip code I was searching for (Check one)

__ Strongly Disagree __ Disagree __ Neither Agree or disagree

__ Agree __ Strongly Agree

QA Test Plan

Test Objective

1. The objective of this test is to validate the functioning of one of the primary features of the website, i.e. Search.
2. The tester shall be given specific instructions to follow and the results shall be recorded as a PASS/FAIL in each of the individual cases.

Hardware and Software setup

1. The tester should have access to a device that supports recent versions of two web browsers – Google Chrome and Mozilla Firefox.
2. The devices used should be connected to internet via Wi-Fi, Ethernet, Cellular, etc.
3. The internet connection speed should be adequate (at least 2.5MBps) for flawless browsing experience.
4. And finally, the user should open the following link on either of the prescribed browsers to begin the test – https://sfsuse.com/fa17g06/

Features to be tested

This test will solely focus on the Search feature of the website. The Search feature shall be operated by inputting a query in the search field on the very top of the homepage of the website. The permitted inputs are names of cities or the zipcodes. As of December 7th, 2017; the database has 5 properties and each of our test case will seek to verify these entries.

Test #	Test title	Description	Test input	Expected output	Result (Chrome)	Result (Firefox)
1.	Zip-code search	Tester shall input the prescribed zip-code number into the search bar and the results shall be validated through comparison with the expected output.	94102	**3 results** -1 Dr Carlton B Goodlett Pl -200 Larkin Street -99 Grove Street.	PASS	PASS
2.	City search by full name	Tester shall input the prescribed name of city into the search bar and the results shall be validated through comparison with the expected output.	san francisco	**5 results** - 50 Phelan ave - 1 Dr Carlton B Goodlett Pl -200 Larkin Street -99 Grove Street -One Ferry Building	PASS	PASS
3.	City search by partial name	Tester shall input the prescribed partial name of a city into the search bar and the results shall be validated through comparison with the expected output.	francis	**5 results** - 50 Phelan ave - 1 Dr Carlton B Goodlett Pl -200 Larkin Street -99 Grove Street -One Ferry Building	PASS	PASS

Code Review

A code review was conducted on the basic implementation of search on Essos. The intent of this submitted code is to query the database of listings and return MySQL %LIKE results in an object to be rendered by a search results page.

In order to ensure clear code, indentation style is used in which code housed in blocks is indented by 4 spaces.

Feature: Search
Submitted by: Alberto Mancini

From /db/listings/index.js

```
/* Searches the database for listings LIKE the passed search query */
function search(query, callback) {
   db.query("SELECT * "+
                   "FROM listings " +
                   "WHERE city LIKE '%" + query +"%' " +
                   "OR zipcode LIKE '%" + query +"%'",
   function(err,rows,fields) {
       if(err) {
           res.status(500).json({"status_code": 500,"status_message": "internal server error"});
           return callback(err);
       } else {
           var listings = [];
           for(var i = 0; i < rows.length; i++) {

               var listing = {
                   'address' : rows[i].address,
                   'city' : rows[i].city,
                   'zipcode' : rows[i].zipcode,
                   'price' : rows[i].price,
                   'sqft' : rows[i].sqrft,
                   'type' : rows[i].type,
                   'agent': rows[i].agent_id,
                   'id': rows[i].listing_id,
                   'thumb': undefined
               }
               listings.push(listing);
           }
           return callback(listings, rows.length);
       }
   })
}
```

Called by the corresponding route:
From /routes/index.js

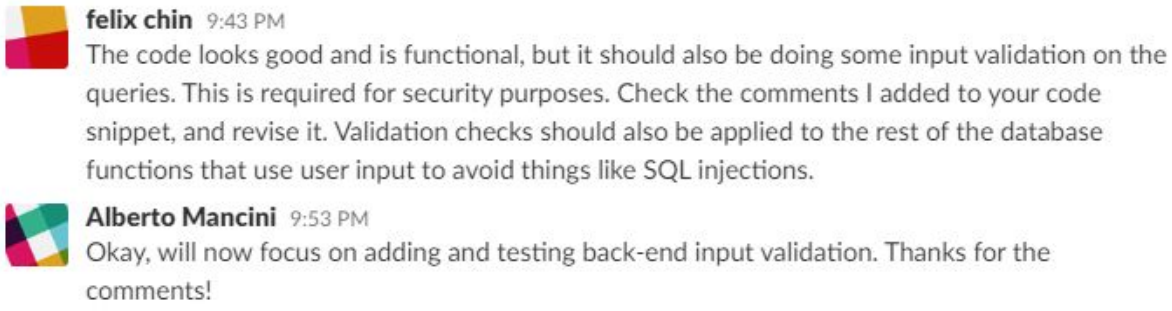
```
router.get('/test', function(req, res, next) {

   db.Listings.search(req.query.search, (searchResults, rows) => {

       /* TEST IMAGE PATH FETCH IN ISOLATION
       for(var listing of searchResults) {
           db.Images.get_listing_images(listing.id, (path) => {
               listing.thumb = path[0].img;
           });
       }*/

       res.render('test', { title: 'Results Page', search: req.query.search, results: searchResults,
                            search_val: req.query.search, resultsFound: rows});
   });
});
```

Reviewed by Felix Chin via Slack (12/7/17)




```
router.get('/test', function(req, res, next) {
   //limit max query length to 40 characters
   req.checkQuery('search', 'Exceeded maximum query length.').len(0, 40);
   //Sanatize the input by escaping harmful chars
   req.check('search','Invalid search input').escape(req.query.search);

   //checks if the string is alpha-numeric and US chars --helpful
   //req.check('search','Invalid search input').isAlphanumeric(req.query.search,[en-US]);

   db.Listings.search(req.query.search, (searchResults, rows) => {

       /*
       for(var listing of searchResults) {
           db.Images.get_listing_images(listing.id, (path) => {
               listing.thumb = path[0].img;
           });
       }*/

       res.render('test', { title: 'Results Page', search: req.query.search, results: searchResults,
                            search_val: req.query.search, resultsFound: rows});
   });
});
```

Self-check: Security

Major Assets Being Protected

- User information
    - Name
    - Email
    - Passwords

Are passwords being hashed in the database?

YES. All passwords are hashed, and no plain-text passwords are stored in the database.

Input Data Validation

- Search bar input

```
//limit max query length to 40 characters
req.checkQuery('search', 'Exceeded maximum query length.').len(0, 40);
//Sanitize the input by escaping harmful chars
req.check('search','Invalid search input').escape(req.query.search);
```

- Account registration and login fields

```
/* Field validation */
req.checkBody('fname', 'First Name field cannot be empty.').notEmpty();
req.checkBody('lname', 'Last Name field cannot be empty.').notEmpty();
req.checkBody('email', 'The email you entered is invalid.').isEmail();
req.checkBody('email', 'The email you entered is of invalid length.').len(4, 100);
req.checkBody('pwd', 'Password must be at least 8 characters long.').len(8, 100);
req.checkBody("pwd", "Password must include one lowercase character, one uppercase character, a number, and a special character.")
            .matches(/^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.* )(?=.*[^a-zA-Z0-9]).{8,}$/, "i");
req.checkBody('pwd-match', 'Password must be between 8-100 characters long.').len(8, 100);
req.checkBody('pwd-match', 'Passwords do not match.').equals(req.body.pwd);
```

Self-check: Non-functional Requirements

Current Development Status of Non-Functional Requirements

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 Google Cloud Platform 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	ON-TRACK
8	Maps showing real-estate location shall be required	ON-TRACK
9	Application shall be deployed from the team's account on Google Cloud Platform	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.	ON-TRACK
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.	ON-TRACK
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)	ON-TRACK
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).	DONE