Software Engineering CSC 648/848

Section 01 Fall 2017

Future Homes Real Estate

Team 10

***Team Lead*: Raymie Michael**
***Email*: rmichael@mail.sfsu.edu**
Sohaib Syed
Dilraj Singh
Justin Zhu
Peter Cruz
Mrinalini Garre

Milestone 4

December 8, 2017

Date Submitted	**December 8, 2017**
Date Revised	

1. Product Summary

Our website is called Future Homes, and what makes our website special is that, it is user friendly. Our website is about browsing for homes and exploring the property details and these are our P1 functions:

1. All the guests are prompted to enter a city or zip code into the search bar to look for the listings in the area.
2. If guests enter an invalid address then the search bar will give the user suggestions of the correct addresses which is similar to the invalid address.
3. Guests have access to a features bar which will further sort the listed homes.
4. Guest have access to our team names and their details who put the website together, under contact.
5. Guests can sort the homes from lowest to highest price, highest to lowest price, or most to least amount of rooms.
6. Guests can access to information about the listings' real estate agent.
7. Guests can view details about the houses such as: house picture, address, and value of the house.
8. Sellers have an inbox of forms which are submitted by potential buyers.
9. Sellers can contact the buyers who are interested in their listings by sending them a form.
10. The admin have the ability to delete any property listings.
11. Sellers are required to register before uploading any sort of listings.
12. Our website provides property details and location of a map.
13. Our website has a login and signup option for buyers and sellers.

14. Guests and registered sellers will be able to contact real estate agents.

15. Future Homes' website provides list homes, rooms for rent, pieces of land, condominiums, apartments, and townhomes that are available or have been recently sold.

2. Usability Test Plan

Future Homes' Usability Test Plan on Search Function

Test Objectives:

To obtain constructive user feedback regarding the search for listings function on our website.

To determine if the search by zip code or city feature is correctly working and gathering the relevant results.

To determine if the query has sufficient error checking.

To determine if the correct number of listings that a search required is displayed on the listings page.

Test Plan

System setup:

Laptop computers or desktops with Windows or Mac OS operating systems using latest Chrome and Safari browsers.

Starting point:

Future Homes is open. (sfsuse.com/fa17g10/)

Tasks to be Accomplished:

Attempt to enter an invalid zip code.

Attempt to enter an invalid city.

Enter San Francisco and acquire x listings in San Francisco along with listings in similarly named locations.

Enter "Sam Fransisco" and still acquire x listings in San Francisco or similar locations.

Enter 94066 and acquire x listings in San Bruno along with listings in similarly named cities.

Enter “9406” and still acquire x listings in San Bruno or similar locations.

Enter California and acquire all listings in California.

Intended User:

There will be six mock users of Future Homes; all of whom have basic computer skills. They will be representing the intended users. The intended users are people wishing to buy real estate and browse real estate.

Questionnaire:

1. It was easy to locate and use the search function.

☐ Strongly disagree ☐ Disagree ☐ Neither agree nor disagree ☐ Agree ☐ Strongly agree

2. The displayed results were relevant to my query.

☐ Strongly disagree ☐ Disagree ☐ Neither agree nor disagree ☐ Agree ☐ Strongly agree

3. The search function always understood what I was trying to type.

☐ Strongly disagree ☐ Disagree ☐ Neither agree nor disagree ☐ Agree ☐ Strongly agree

4. If you have any further comments, please write them in the space provided below.

3. QA Test Plan

Future Homes' Quality Assurance Test Plan on Search Function

Test Objectives:

To determine if the search functionality of Future Homes is working properly.

Hardware Setup:

Macbook Pro

System Type: macOS Sierra version 10.12.16

Processor: 2.4 GHz Intel® Core TM i5

Software Setup:

Google Chrome Browser version 62.0.3202.94

Feature to be Tested:

The feature to be tested is searching for listings using the provided search bar on the home page. The intended results are listings which contain the searched city or zip code or a similar interpretation. The results are to be listed on the listings page and it should indicate how many listings have appeared in the search.

	Description	**Test Input**	**Expected Output**	**Pass/ Fail**
1	To test search by zip code (Browser: Safari)	Enter “94016” in the search box and click the search button.	The search page should display searches under “Search Results” 1. 1600 Holloway Ave, San Francisco, CA 94132. Size: 1,200 square feet. 3 bedrooms, 2 bathrooms. Price: $1,200,000. 2. 2130 Fulton St, San Francisco, CA 94117, Size: 1,900 square feet, 4 bedrooms, 3 ½ bathrooms, $2,750,000.	Pass
2	To test search by city (Browser: Chrome)	Enter “Palo Alto” in the search box and click the search button.	The search page should display searches under “Search Results” 3. 3330 College Drive, San Bruno, CA 94066. Size: 990 square feet. 2 bedrooms, 1 bathroom. Price: $1,200,000.	Pass
3	To test search by State (Browser: Safari)	Enter “CA/ California” in the search box and click the search button.	The search page should display searches under “Search Results” 4. 1700 W. Hillsdale Boulevard San Mateo, CA 94402. Size: 1,100 square feet. 3 bedrooms, 2 ½ bathrooms. Price: $1,300,00	Pass

4. Code Review

Coding Styles

Four space indentation

The opening bracket is to be placed at the end of the first line

The closing bracket is to be placed on a new line, else statements excluded

Commenting throughout simple, smaller pieces of code

A header with short summary in complex pieces of code that serve important role

Camelcase for names and variables

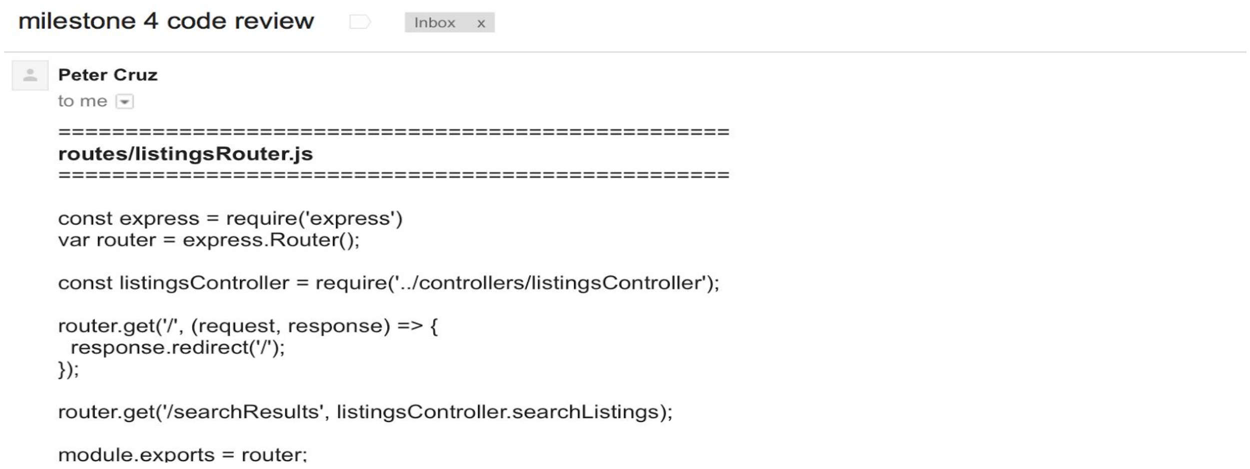
NodeJS:

Short lines are better than lines that are too long

HTML

All tags and attributes should be in lowercase

Code Review Emails:

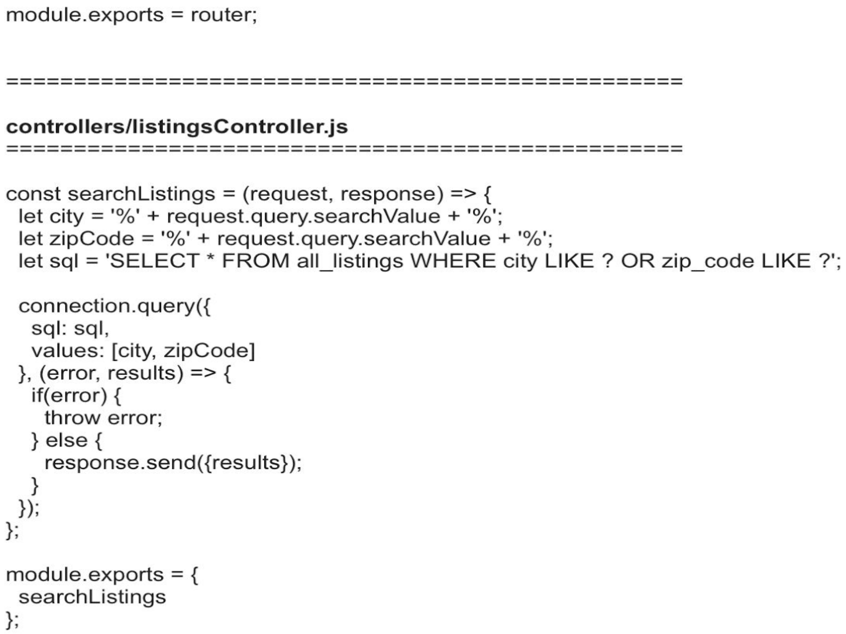


milestone 4 code review

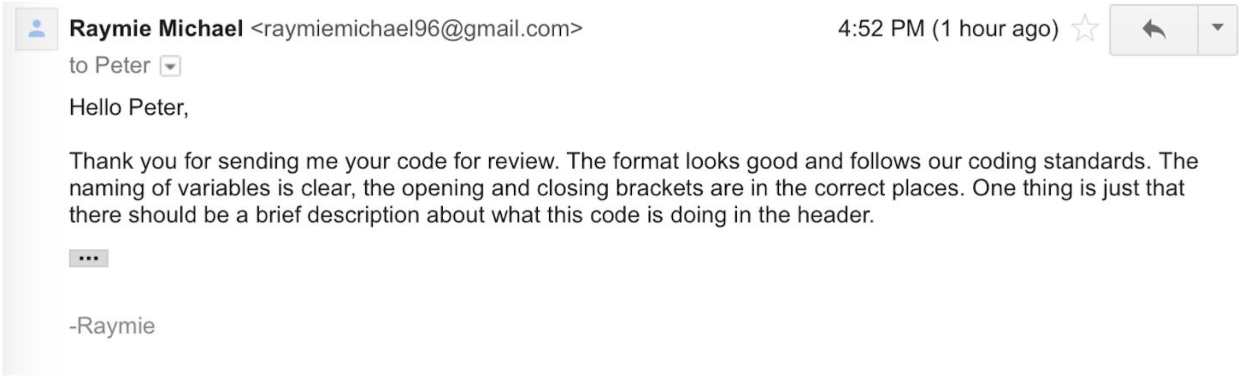
Peter Cruz

to me

```
==================================================
routes/listingsRouter.js
==================================================

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

const listingsController = require('../controllers/listingsController');

router.get('/', (request, response) => {
  response.redirect('/');
});

router.get('/searchResults', listingsController.searchListings);

module.exports = router;
```



```
module.exports = router;


==================================================

controllers/listingsController.js
==================================================

const searchListings = (request, response) => {
  let city = '%' + request.query.searchValue + '%';
  let zipCode = '%' + request.query.searchValue + '%';
  let sql = 'SELECT * FROM all_listings WHERE city LIKE ? OR zip_code LIKE ?';

  connection.query({
    sql: sql,
    values: [city, zipCode]
  }, (error, results) => {
    if(error) {
      throw error;
    } else {
      response.send({results});
    }
  });
};

module.exports = {
  searchListings
};
```



Raymie Michael <raymiemichael96@gmail.com>
to Peter

Hello Peter,

Thank you for sending me your code for review. The format looks good and follows our coding standards. The naming of variables is clear, the opening and closing brackets are in the correct places. One thing is just that there should be a brief description about what this code is doing in the header.

-Raymie

Edited Code:

===

routes/listingsRouter.js
This performs the action of displaying the listings page after a search.

===

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

const listingsController = require('../controllers/listingsController');

router.get('/', (request, response) => {
  response.redirect('/');
});

router.get('/searchResults', listingsController.searchListings);

module.exports = router;
```

===

controllers/listingsController.js
This will fetch listings from the database that are similar to user’s query.

===

```
const searchListings = (request, response) => {
  let city = '%' + request.query.searchValue + '%';
  let zipCode = '%' + request.query.searchValue + '%';
  let sql = 'SELECT * FROM all_listings WHERE city LIKE ? OR zip_code LIKE ?';

  connection.query({
          sql: sql,
          values: [city, zipCode]
  }, (error, results) => {
          if(error) {
          throw error;
          } else {
          response.send({results});
          }
  });
};

module.exports = {
  searchListings
};
```

5. Self-Check: Best Practices for Security

List major assets you are protecting

- We are protecting the realtor’s account information, including their first name, last name, username, and password.
- We are protecting the listings that are currently in the database.
- We are protecting the dashboard that the realtor can view. This is to ensure that no one but the realtor can view the listings and messages that they have with prospective homeowners.
- The privacy of all users shall be protected.

Confirm that you encrypt PW in the DB

- We have encrypted the password in the database.

Confirm Input data validation (list what is being validated and what code you used) – we request you validate search bar input;

- We have validated the input data.

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

Copy all original non-functional specs as in high level application document published at the very beginning of the class and then for each say DONE if it is done (which is expected and required); ON TRACK if it is in the process of being done and you are sure it will be completed on time; or ISSUE meaning you have some problems and then explain it.

Note: you must adhere to all original non-functional specs as published in the original high level specification document. Failure to do so may cause reduced grade

1. Application shall be developed and deployed using class provided deployment stack: ON TRACK.
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: ON TRACK.
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: 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: 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: ON TRACK.
9. Application shall be deployed from the team's account on AWS: ON TRACK.
10. No more than 50 concurrent users shall be accessing the application at any time: ON TRACK.
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: ON TRACK.
14. Google analytics shall be added: ON TRACK.
15. Messaging between users shall be done only by class approved methods and not via email 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: ON TRACK.
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.