Milestone 4

SW Engineering CSC648/848 Spring 2019

Team 101

Project Zephyr

Revisions
5/7/19

Date: May 7, 2019

Team Members: Jiannan Li, Tigist Ambaw, Bijan Mahdavi, Canbin Mei, Sydney Lou Morano, Brendan Ng, William Xie, Samuel Zaffanella

[**Email:** daiweirealrealreal@gmail.com]

Product Summary: Zephyr

List of all major committed functions (all final P1 functions we will have and test for):

1. New Listing - Landlord posts a new apartment listing to get approved by an admin
2. Search Listing - A user will be able to query the database for listings based on a keyword
3. Find all Listings - Returns all listings in the database
4. Filter Listing - Query the database based on multiple checks
5. Delete Listing - A landlord can delete one of their posted listings
6. Register - Landlords and Students can register for an account by providing a username and password
7. Login - Landlords, Students, and Admins can login by providing a valid username and matching password
8. Admin Approve - Admin can approve a listing to be posted
9. Admin Deny - Admin can refuse a listing to be posted

Unique in Zephyr: The ability for students to rate landlord (?)

URL to zephyr: http://csc648section3team101-env.bvkfmtnnnp.us-east-2.elasticbeanstalk.com

QA Test Plan:

Unit Test:
Download unit testing JavaScript framework mocha.
Create test.js file.
Export search and create listing related functions to test.js.
Make test cases.
Call functions and input test cases in parameters.
Compare expected return value.

Unit testing could not be finished in time. There were complications with parameter data types.

Integration Test: **Tested on Chrome and Firefox**

Test Case id: Posting_1
Test Case Description: Post a listing

Test Scenario: As a landlord, post a listing with a data set and then verify that it shows up on the listings page

Test Data:

Title = Tomato
Has Roommate = yes
Provide Parking = no
Allow Pets = yes
Zip Code = 666
Picture = house.jpg
Price = 300.00
Address = Elm Street
Description = Very scary place
House Size = 666
Number of Bathroom = 2
Number of Bedroom = 5
Category = House

Test Coverage: Test Data does not cover when some fields are unfilled.

Test Results:

Step #	Step Details	Expected	Actual Results	Pass / Fail / Not Executed / Suspended
1	Login as Landlord	Landlord logged in	As Expected	Pass
2	Enter Test Data in New Listing	Test Data can be entered	As Expected	Pass
3	Click Save	Leaves the page	As Expected	Pass
4	Check if listing is in Listings	Listing is among Listings with all fields	As Expected, except picture not the same	Almost Pass?

Test Case id: Search_1

Test Case Description: Search for listings

Test Scenario: Search for specific homes using filters, regardless of whether or not you have an account.

Test Data:

Keyword = Street
Choose Category = Apartment
Bedrooms = 1

Bathrooms = 2
Size range = 180 sq. ft - 700 sq. ft
Price range = $100 - $1200

Test Coverage: Test Data does not cover when some fields are unfilled.

Test Result:

Step #	Step Details	Expected	Actual Results	Pass / Fail / Not Executed / Suspended
1	Navigate to Listings page	In Listings page	As expected	Pass
2	Enter Test Data into search fields	Test Data can be entered	As expected	Pass
3	Click Search	Refreshes page	Refreshed to page error	Fail
4	Find listing that match entered data	Search results appear		Not Executed

Test Case id: SignUp_1

Test Case Description: Account creation

Test Scenario: Trying to sign up as a student without an SFSU student email.

Test Data:

First Name = Alex
Second Name = Smith
Account type = Student
Email = AlexSmith@example
Username = Alex
Password = potato
Confirm Password = potato

Test Coverage: Test did not cover account creation for landlord, admin, and pre-existing accounts.

Test Result:

Step #	Step Details	Expected	Actual Results	Pass / Fail / Not Executed / Suspended
1	Navigate to Sign up page	In Sign up page	As expected	Pass
2	Enter Test Data into Sign up fields	Test Data can be entered	As expected	Pass

3	Press Sign up	Leave page	As expected	Pass
4	Sign In with Username and Password	Should not be able to sign in	Able to sign in	Fail

Test Case id: Admin_account
Test Case Description: Admin approves listing
Test Scenario: After a landlord creates a new listing, an admin account must approve of it before it becomes public.
Test Data(subject to change):

Landlord = sam
New listing = (all the data from new listing e.g. address, house, name, size, price etc.)
Approve = yes

Test Coverage: Note - Admin listing review functionality is not fully implemented yet…
Test Result:

Step #	Step Details	Expected	Actual Results	Pass / Fail / Not Executed / Suspended
1	Login as Admin	Admin logged in	As expected	Pass
2	Look for listing to review	In review	Not there	Fail
3	Approve Listing	Page refreshes		Not Executed
4	Find Approved listing	New listing located in Listings		Not Executed

Beta Test Plan:

Beta Testing is one of the customer validation technique to evaluate the level of customer satisfaction with the product by letting it to be validated by the end users, who actually use it, for over a period of time.

Product experience gained by the end users are asked for feedback on design, functionality, and usability and this helps in assessing the quality of the product.

The objective of Beta Test

The points mentioned below can even be considered as the objectives for Beta Test and are very much required to produce far better results for a product.

1. Beta Test provides a complete overview of the true experience gained by the end users while experiencing the product.
2. It is performed by a wide range of users and the reasons for which the product is being used varies highly. Marketing managers focus on target market's opinion on each and every feature, while a usability engineer / common real user focus on product usage and easiness, technical users focus on installation and uninstallation experience, etc. But the actual perception of the end users clearly exhibits why they need this product and how they are going to use it.
3. Real world compatibility for a product can be ensured to a greater extent through this testing, as a great combination of real platforms is used here for testing on a wide range of devices, OS, Browsers, etc.
4. As a wide range of platforms which the end users are actually using, might not be available to the internal testing team during the QA, this testing also helps to uncover the hidden bugs and gaps in the final product.
5. Few specific platforms will cause the product to fail with showstopper bug which was not covered during QA. And this helps in improvising/fixing the product to be a compatible one with all possible platforms. Known Issues, which are accepted by the Product Management team, may take a great turn when the end user faces the same issue and may not be comfortable while using the product.

   In our case the intended users are SFSU student and landlord users.

Beta Test plan

- Testing approach to be followed by the participants
- Collecting feedback from customers/users
- Tools used to log buys, measure productivity, we are collecting feedback using survey or ratings.

<u>Code Review</u>:

a) Our team is using Airbnb JavaScript coding style. This style is easy to read and well documented. The way we enforce our member for the coding style is provide them with the Airbnb coding style GitHub guide.

b) Code review is based on pull request when member trying to commit to certain branch.

Dev to Release #12

Merged **Balbalnom** merged 41 commits into `release` from `Dev` 5 days ago

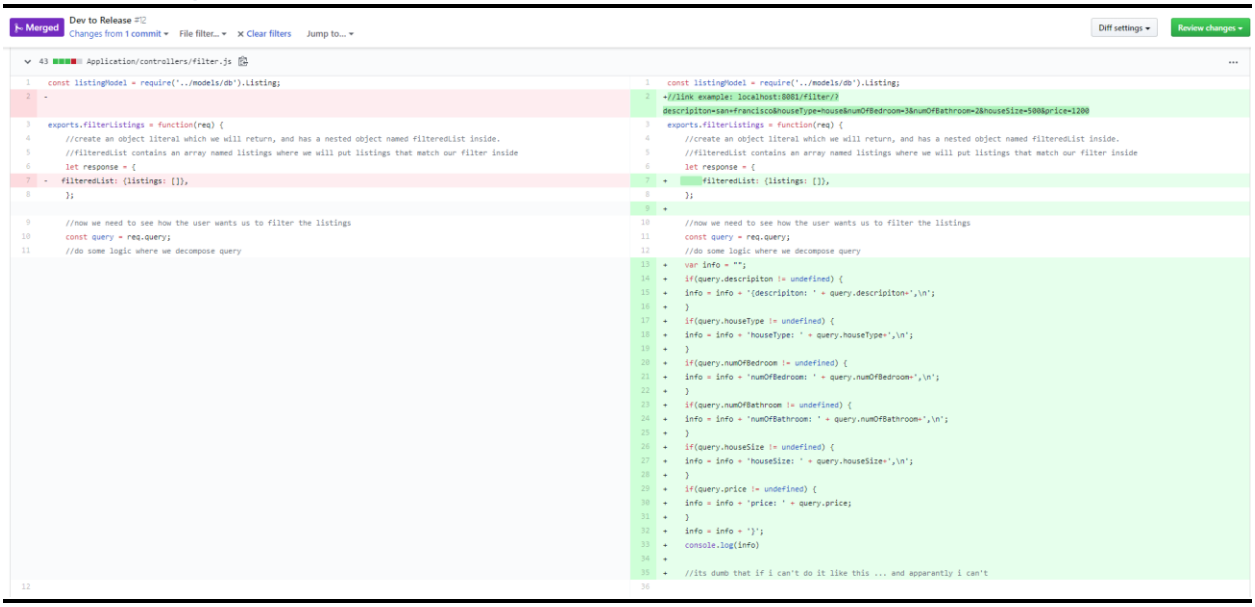
Conversation 0 | Commits 41 | Checks 0 | Files changed 495

Balbalnom commented 5 days ago

No description provided.

Bijan Mahdavi and others added some commits on Feb 28

Adding basic handlebar functionality. Will do something with it tomorrow		ba8381a
forgot this file		ccddfeb
m1 doc finished	Verified	cb4c3b5
new m1 doc update	Verified	7abe616
Mysql routing update		5b0a17d
mysql test		8161841
Search function		34caae2
search function		88585e3
Search functionality at /search		bd13aee
Sequelize setup		2a26feb
sequelize setup		a69feba
Merge pull request #3 from CSC-648-SFSU/development	Verified	660f218
schema done according to current xml		9f5188c
Merge pull request #4 from CSC-648-SFSU/development	Verified	a105021



All differences are being marked in the pull request as well as merge conflicts. It is clean and neat for team member to understand the differences between versions. Upon group member or admin approval. The code can be merge and update into certain branch.

Self-Check:

- 1. Application shall be developed, tested and deployed using tools and servers reviewed by Class TA (Nicholas Olegovich Stepanov) in M0 (some may be provided in the class, some may be chosen by the student team, but all tools and servers have to be reviewed by class TA).
    - **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. Data shall be stored in the team's chosen database technology on the team's deployment server.
    - **Done**

- 4. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
    - **On track**

- 5. Application shall be very easy to use and intuitive.
    - **On track**

- 6. No email clients shall be allowed.
    - **Done**

- 7. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
    - **Done**

- 8. Site security: basic best practices shall be applied.
    - **Done**

- 9. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator.
    - **On track**

- 10. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
    - **Done**

- 11. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2019. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).
    - **Done**