SW ENGINEERING CSC 648/848 FALL 2022

EZRent

By CyberDesign

Team 4

Members:

Team Lead : Devon Dy-Liacco ddyliacco@mail.sfsu.edu

Github Master: Praise O Eubany

Frontend Lead: Youssef Hammoud

Backend Lead: Issa Shihadeh

Database Master: Tung Nguyen

Frontend Engineer: Ricardo Lopez

Milestone 4

12/01/2022

History Table:

Version	Latest revision
M4V1	12/01/2022
M4V2	12/15/2022

1. Product summary

EZRent

Itemized List of major committed functions:

Criterion	Requirements
1. User Registration	1.a. Users must be able to register an account to become a Registered User 1.b. The system must allow Users to agree to a Privacy Policy 1.c. Users must be able to browse the website without an Account 1.d. The system must allow Users to provide their personal information 1.e. The system must allow Users to create a username for their account 1.f. The system must allow Users to change their personal information 1.g. The system must allow Registered Users to become a Landlord
2. User login	2.a. The system must allow Registered Users to log in their account. 2.b. The system must allow Registered Users to reset their password. 2.c. The system must allow Registered Users to browse the website 2.d. System must allow Registered Users to log out
3. Website Header	3.a. The Website Header must include the search feature 3.b. The Website Header must be able to link to the Registration Page 3.c. The Website Header must be able to link to the Login Page 3.d. All Registered Users must be able to navigate to their Profile 3.e. All pages must include the Website Header
4. Home Page/Navigation	4.a. The Home Page must be accessible to all Users 4.b. Users must be able to navigate to the Home Page from any page
5. Website Footer	5.a. The Website Footer must allow Users to see the publisher's information 5.b. All pages must include the Website Footer

6. Search	6.a. Users must be able to search for Property Listings 6.b. Users must be able to search for Landlords 6.c. Users must be able to search Listings by address 6.d. Users must be able to search Landlords by name
7. Search Filters	7.a. Users must be able to filter their searches 7.b. Users must be able to filter Property Listings by price 7.c. Users must be able to filter Property Listings by address 7.d. Users must be able to filter Property Listings by Amenities 7.e. Users must be able to filter Property Listings by number of rooms 7.f. Users must be able to filter Property Listings by number of bathrooms 7.g. Users must be able to filter Property Listings by size 7.h. Users must be able to filter Property Listings by rating
8. Search Results	8.a. Users must be able to visit Property Listing pages from the search results page 8.b. Users must be able to visit Landlord Profile pages from the search results page 8.c. Users must be able to sort their search results 8.d. Users must be able to sort their Listing search results by price 8.e. Users must be able to sort Listing search results by rooms 8.f. Users must be able to sort Listing search results by distance 8.g. Users must be able to sort Listing search results by rating 8.h. Registered Users must be able to sort Renter search results by rating
10. Ratings	10.a. Renters must be able to rate Properties 10.b. Renters must be able to rate Landlords 10.c. Users must be able to see Ratings
11. Reviews	11.a. System must allow Renters to write Reviews 11.e. System must allow Registered Users to rate Reviews 11.g. System must allow Users to read Reviews 11.i. System must allow Users to view Ratings 11.j. System must allow Registered Users to change their ratings to Reviews

12. Property Listing	12.a. Landlords must be able to post Listings 12.b. Landlords must be able to delete Listings 12.c. Landlords must be able to mark Listings as currently rented 12.d. Landlords must be able to unmark Listings as currently rented 12.e. Landlords must be able to post images to Listings 12.f. Landlords must be able to write descriptions to Listings 12.g. Landlords must be able to set locations to Listings 12.h. Landlords must be able to set a price to Listings 12.i. Landlords must be able to set a size for Listings 12.j. Users must be able to see Listings 12.k. Users must be able to find the Landlord page of the one who posted the Listing 12.l. Users must be able to post Reviews to Listings 12.m. Registered Users must be able to post Reviews to Listings 12.n. Renters must be able to save Listings to favorites
13. Landlord Profile	13.a. System must provide a Profile for Landlords upon Account creation 13.b. Landlords must be able to set a bio to their Profile 13.c. Users must be able to see a Landlord Profile 13.d. Users must be able to see Reviews about a Landlord on the Landlord Profile 13.e. Users must be able to see a Landlord's rating on their Profile 13.f. Users must be able to see a Landlord's contact information 13.g. Users must be able to see a Landlord's name 13.h. Users must be able to see a Landlord's email 13.i. Registered Users must be able to post Reviews to Landlord Profiles
14. Renter Profile	14.a. System must provide a Profile for Renters upon Account creation 14.b. Renters must be able to set their Account to Prospective Renter 14.c. Renters must be able to set a bio to their Profile 14.d. Users must be able to see public Renter Profiles 14.e. Renter Profiles must show Renter ratings 14.f. Renters must be able to edit their Profile 14.g. Renter Profiles must show Renter history

Finding a good landlord in this market is exhausting and extensive. One shouldn't have to guess the type of relationship they might have with a potential landlord. Wouldn't it be easier to be able to see the past relationships between landlords and previous tenants? With our web

application we provide a full transparent relationship between property owners and their tenants. The struggle of finding good, appropriate landlords would diminish because our services encourage proper communication between all parties. When a landlord doesn't uphold certain expectations, tenants are able to leave a detailed review on said landlord on our site. This review is attached to the property owner's name and in doing so, it also gives other interested renters knowledge of how appropriate the landlord might be. Users are able to leave a review based on the communication of the owner, the initial behavior as well as their overall experience with said person. Not only is this applicable for landlords, but for actual properties as well. All previous tenants are able to search through the profile page of a landlord and find a listing that they had previously occupied in order to leave a review and rating of their overall experience. This is then made public for all interested general users to observe and determine if a potential property or property owner fits within their criteria. Also, the overall ratings given to these objects at hand are also available. They are easily viewable in the shape of stars with the maximum amount being 5 stars, which indicates a high satisfaction for the topic at hand, whether is being the landlord or the actual property.Landlords are able to create a public profile here as well. They are able to post properties in hopes of attracting potential renters. Their information will be apparent within their designated bio where interested parties can utilize in reaching them for the interest of housing accommodations. So whether you are a future renter looking for a new housing experience or you're a previous tenant who wishes to report issues with a previous or current property; or even a landlord wishing to attract renters for their properties, Our site has the ability to accommodate all of those needs.

2.Usability test Plan

1.Post review

- Test objectives per function

- The post a review function is being tested because or website allows the user to leave a review for both either a landlord or a posting. This is meant to ensure quality between landlords and renters. Without this function it wouldn't be possible to have clarity amongst the users and property owners. This also provides users to observe the renter experience on landlords as well as observing the relationships between all the parties involved. This is also the main objective of

the website so it is very imperative that this function is easily accessible to all users who desire to use it.

- Test description per function
- The system setup for posting a review resides within the profile page of a landlord as well as the listing. There is a button that indicates to a user that they will be allowed to leave reviews based of the experience. The renters will have the ability to post how their living experience was with the landlord. Button wille be accessible to all users for viewing but will only be available for registered users to actually use. This is due to the system verifying that the review is actually legit. However with this being said, general website users will be able to view the reviews left after a registered user posts one. The available posts can be found either on a landlord page or on the actual listing page.
- To use this feature a user can visit the website, upon scrolling around the website, the featured reviews that are seen on the website have the ability to be clicked on for a more intense observation. By clicking on said feature review, the user will then be transported to the landlord profile page in which all left reviews will be apparent. At this point users should be able to see all left reviews made inspired by other registered users. On the left hand side of the screen there will be a button that is titled Post a review. That button will then give access to registered users to the post a review page. Upon being transported to the page, the users will then have the ability to leave comments on tha landlord.
- The starting point for this function is at the profile page of a landlord or on the property listing page. When visiting those pages, a button will be seen which will allow registered users to leave a comment review based on their living experience while living in a property posted by the landlord. Once posted, users will have the ability to see the comment based on the location the review was posted. For example, if the review was posted in means of the landlord profile page, then the updated review should be apparent on the refreshed profile page of the landlord. It will be organized within the section designated for reviews. The same is also said for leaving a review on a property. Once the users posts their review on a specific property, the comment can be viewed in the area designated for reviews on the listing page.
- The url for this feature is http://localhost:8080/profilepage

2.Post a listing

- Test objectives per function
- We are testing the post listing function because we believe it is one of the most important functions for the customers. It is the main reason why people will be using our website and we need to make sure it is working properly at launch for all of the users. The goal is for registered users to be able to post their listings and be able to put their properties up to the public. This will allow for users/registered users to view listings and rent as they please. Without listings, there are no users to be able to rent freely.

- Test description per function
- The system setup for this function resides in landlord profile page.
- The starting point of this will be at the profile page of a landlord. Once logged in, a landlord will have the ability to click on their public profile page, In viewing said page, there will be a button at the left hand side of the screen that would be labeled “post a listing.” Once clicked, the user will then be able to fill in all necessary information to regarding the listing in which hey desire to post. For example, the amount of beds/baths, the address as well as any relevant photos of the property.
- The intended usage for this feature is only the registered landlords. This comes with the assumption that the purpose for hte user to register as a landlord i because they posses properties in which they are trying to advertise.
- The url for this feature is http://localhost:8080/postListingPage

3.Filters

- Test objectives per function
- Our filters will be tested for the sole purpose of having the freedom to search for their needs. We want it to be easier for the consumer to search for listings/landlords to their personal preference. There might be some people who have certain preferences and will be required to search for different types of listings. They need to be able to search for listings that fit the description they are looking for. We feel it is an important factor to make sure the user searches for listings more easily.
- Test description per function
- The system set up for applying the filters resides in the listing results page. When a user uses the search bar to search for a listing, the website would refresh. In doing so, the results of the search should then be apparent on the right hand side of the screen. At this point as well, the options to apply features will be present on the left hand side of the screen. A user can then click on the buttons given in the filter section indicating which type of properties you want to view.
- The starting point of this function resides in the searchpage of the website. When a user wants to search for a specific property, the result page will then showcase the options to apply filters to the search results. When the filters are pressed, there will be a button on the left hand side which will communicate to the user to apply all the indicated specifications. Once done, the search results should alter based on available properties adhering within those indicated specifications.
- This feature should be available for use to all general users. This is because this feature doesn’t require a user to be registered. We want the general public to be able to search through all provided properties first when observing the website. They will be able to view all given homes that fall under their indicated specifications.
- The Url is http://localhost:8080/listings/search

4.Landlord registration

- Test objectives per function

- Landlord registration is a way for the user to indicate their role while using our site. This allows for special privileges that registered user won't be able to access unless they fall under the role of landlord. Landlords are able to post listing that other users will have access to seeing and observing further. The landlords will also have the ability to manage their listing. They will be able to see all their current listings that are posted as well as view the reviews left by registered users on the property. This also allows for landlords to become a public image. They have the ability to be rated by other users freely. This is to ensure the objective of the website. We want to encourage a good proper relationship with landlords and their renters and by creating a public image for each registered landlord, we will be able to give access to registered users to do just that.

- Test description per function

- The system set up of this function relies in in the registration page on the site. A user will be prompted to enter their information, as if they are signing up for our services. However, there is a button that will prompt the user to differentiate themselves from being a regular registered user or a landlord registered user. When registered, they will then login to the site using the information that they have provided to us. Upon logging in, there is a button that will cause the landlord registered user to be able to see their public profile as well as any reviews that were left for them from other registered users who they have had contact with.
- The starting point for this function begins at the point of registration. As previously stated, a user will be prompted to provide the site with certain information, in the regards to creating their profile. However, an interested user will have the ability to differentiate themselves by clicking on a button that will ask if they wish to be considered a landlord on this site. Upon registration, a landlord will be able to view their profile by clicking on the profile button placed on the top right hand side of the screen. By pressing it, the public profile of the landlord will be accessed where thre options to view and manage listing will be available. Alongside so, the reviews that were made as a result of the landlord will also be visible in its designated section. The main difference from signing up as a landlord instead of a regular user is the ability to post listings. The post button will be available for use on the left hand screen with the label "post a listing."
- The intended usage for this is anyone who desires to be registered as a landlord. Those who have properties to list will be able to post said homes or apartments to the site in hopesof getting tenants to occupy the spaces.
- The URl is http://localhost:8080/register

5.Leave comments on reviews

- Test objectives per function
- The objective of this function is to allow the registered user to communicate with other users. There will be instances where someone would like to comment or ask a question to someone who left a review. Sometimes the people want them to ellaborate or be more specific with their reviews. It helps understand the experience that a renter had with a Landlord. We believe it is a great feature to include because it gives the renters a chance to comprehend and connect with one another to help everyone find the Landlord they are looking for. It is a function that will allow the users to have a little more in depth knowledge about the Landlord/Listing.

- Test description per function
- The system set up for this function resides in the listing page. The place designated for viewing reviews on will also have the option to leave comments on said reviews.
- The starting point for this feature will reside in the listing page. A user can use the search bar provided at the top of the screen to search for a property of their choosing. Once all properties are viewed regarding the users indicated search result, once a specific listing is clicked, the user will then be transported to the actual listing page. On said listing page, a user can scroll down to and there they will be able to view all reviews left by other users based off the specific listing. If a user wanted to leave a comment, there is a button that will indicate to the user of their ability to apply this feature.
- The intended usage of this feature is all general users as well as registered users and registered landlords.
- The url for this feature is http://localhost:8080/listingpage

Usability Metrics: Effectiveness

Test/ Use Case	**% Completed**	**Errors**	**Comments**	**% In time completed**
Post Review	80%	-doesn't post	-I can't see my review	80%

Post Listing	80%	-doesn't post	-can't see personal listing	80%
Filters	60%	-buttons doesn't filter	-previous listings still show	70%
Landlord Registration	70%	-implemented but doesn't differentiate	-still the same as regular user	70%
Leave comments on reviews	30%	-not yet implemented -hasn't been made public -only available in backend and database	There is no function for comment	0%

Questionnaire

Strongly Disagree Strongly Agree

1 **2** **3** **4** **5**

Post Review

- I easily figured out how to post a review

1 2 3 4 5

-The post review function works properly and is ready for public use.

1 2 3 4 5

-I think the Post review function can be easily used by many

1 2 3 4 5

Post Listing

-I found the post listing form difficult to fill out

1 2 3 4 5

-A landlord would have no trouble posting a listing

1 2 3 4 5

-I think the Post Listing form is set up well for the user to understand

1 2 3 4 5

Filters

-I understood how to use the filters fairly quickly

1 2 3 4 5

-I think that other people will be able to search with filters easily

1 2 3 4 5

-I feel like the search filters were well implemented

1 2 3 4 5

Landlord Registration

-I understood the purpose for the landlord registration

1 2 3 4 5

-I think I would enjoy using this website as a Landlord

1 2 3 4 5

-I believe the check box placement to confirm you are a landlord is well placed and visible.

1 2 3 4 5

Leave comments on reviews

-I found this feature complicated and hard to find

1 2 3 4 5

-I believe leaving comments on reviews can be improved

1 2 3 4 5

-I think this feature would be loved by many

1 2 3 4 5

3. QA test plan

1. Registration with password

- Test objectives: Passwords should be validated and encrypted properly upon registration
- HW and SW setup (including URL): http://localhost:8080/register
- Feature to be tested

#	Description	Test Input	Expected Output	Pass Fail
1	Test a password with no special characters	abc12345	No registration	Pass
2	Test password encryption	Randell123@	$2b$10$6oe7ZqkouHmoNX3JtRbXvOxT1CFMEswqQEOAAVeWw3z7TSlYUhKVq	Pass
3	Test a password that fulfills all requirements	Randell123@	Registration	Pass

2. Mysql Query

- Test objectives: - what is being tested
- HW and SW setup (including URL): http://localhost:8080/
- Feature to be tested

#	Description	Test Input	Expected Output	Pass Fail
1	Test a query with a condition in registeredUser table	“SELECT reg_user_id FROM registeredUser WHERE email = ‘randell@gmail.com’”	1 user	Pass

2	Test a query with joining tables and a condition.	“SELECT rating FROM review LEFT OUTER JOIN registeredUser ON review.referLandlordId = registeredUser.reg_user_id WHERE review.referLandlordId = 17 AND registeredUser.role = 'landlord';”	1 rating	Pass
3	Test a query getting 3 landlords with the highest rating	SELECT reg_user_id, firstName, lastName, email, bio, user_rating FROM registeredUser WHERE role = 'landlord' ORDER BY user_rating DESC LIMIT 3;	3 users that have a rating greater than 4	Pass

3. Rating

- Test objectives: - rating is displayed, calculated, and posted properly
- HW and SW setup (including URL): http://localhost:8080/
- Feature to be tested

#	Description	Test Input	Expected Output	Pass Fail
1	Calculate user rating	4 and 3	3.5	Pass
2	Display rating of 3.4	3.4	3 stars and 40% of a star	Pass
3	No rating in db will display N/A	null	N/A	Fail

4. Code review

Internal Review

```
// Execute Landlord Search
function executeLandlordSearch() {
    // Get input value that is searched for
    let searchTerm = document.getElementById("search-text").value;
    // If there are no value
    if (!searchTerm) {
        // reload to home page
        location.replace('/');
        return;
    }
    // if there is a value go to search results page and output the results of the value
    let searchURL = `/users/search?search=${searchTerm}`;
    location.replace(searchURL);
}

// Get the input field
var input = document.getElementById("search-text");

// Execute a function when the user presses a key on the keyboard
input.addEventListener("keypress", function(event) {
  // If the user presses the "Enter" key on the keyboard
  if (event.key === "Enter") {
    // Cancel the default action, if needed
    event.preventDefault();
    // Trigger the button element with a click
    document.getElementById("search-button").click();
  }
});
// const sb = document.getElementById('search-select');

searchButton.onclick = (event) => {
    event.preventDefault;
    if (currentOption == "listings") {
        executeSearch();
    } else if (currentOption == "landlords") {
        executeLandlordSearch();
    } else {
        location.replace('/');
    }
}
```

Why we picked this code:

We chose this piece of code because it is very crucial to our website. This piece of code is for executing user input in the search bar. What this code means is that when the user clicks on the enter key or pressed the search button, one of two things will happen; If the dropdown menu next to the search bar is selecting “Listings”, the executeSearch() function will be called. This function will receive user input from the search bar and search for and output the results from the listings table in the database. However, if the “Landlords” option is selected in the dropdown menu, the executeLandlordSearch() will be called. This function will receive user input from the search bar and search for and output the results from the users table in the

database. The search feature in our website is one of the most important features, thus why we picked this code for review, so we can be more cautious of the correct functionality and implementation of it.

Review from team member:

1. Comments help understand the code
2. Variable names are clear
3. Some comments missing
4. Some comments are redundant
5. Functions have a distinct goal and are separated appropriately

External Reviews

```
const db = require('../database/db');
// var express = require ('express');
// var router = express.Router();
// var sharp = require ('sharp');
// var multer = require ('multer');



//mounting data according to our database
class Register {
    constructor(landlord_id, street_num, street_name, city, state, zipcode,
description, bed, bath, price, file_name, rating) {
        this.landlord_id = landlord_id;
        this.street_num = street_num;
        this.street_name = street_name;
        this.city = city;
        this.state = state;
        this.zipcode = zipcode;
        this.description = description;
        this.bed = bed;
        this.bath = bath;
        this.price = price;
        this.file_name = file_name;
        this.rating = rating;
    }
    //keeping track of time of listing creation
    save() {
        let d = new Date();
        let yyyy = d.getFullYear();
        let mm = d.getMonth() + 1 ;
        let dd = d.getDay();



        let createdAtDate = `${yyyy}-${mm}-${dd }`;
```

```
        //simple sql code to update listing details
        let sql = `
        INSERT INTO listing(
            landlord_id,
            street_num,
            street_name,
            city,
            state,
            zipcode,
            description,
            bed,
            bath,
            price,
            created_at,
            file_name,
            rating

        )
        VALUES (
            '${this.landlord_id}',
            '${this.street_num}',
            '${this.street_name}',
            '${this.city}',
            '${this.state}',
            '${this.zipcode}',
            '${this.description}',
            '${this.bed}',
            '${this.bath}',
            '${this.price}',
            '${createdAtDate}',
            '${this.file_name}',
            '${this.rating}'
        )
        `;


        return db.execute(sql);

    }
        //default to display listings
    static search(search, filters, sorting) {
        let sql = `SELECT * FROM listing WHERE listing_id BETWEEN '0' AND '7';`;


        return db.execute(sql);
    }
    //display listings according to zipcode
    static getListByZipcode(zipcode) {
        let sql = `SELECT * FROM listing WHERE zipcode LIKE '${zipcode}' OR
street_number LIKE '${zipcode}';`;
```

```
        return db.execute(sql);
    }
    //display listings according to city
    static getListByCity(city) {
        let sql = `SELECT * FROM listing WHERE city LIKE '%${city}%' OR street LIKE
'%${city}%';`;

        return db.execute(sql);
    }
    //display listings according to address
    static getListByAddress(city) {
        let sql = `SELECT * FROM listing WHERE address LIKE '%${city}%';`;

        return db.execute(sql);
    }
    //if user doesn't search for anything we just diplay listings anyways
    static findAll() {
        let sql = `SELECT * FROM listing WHERE listing_id BETWEEN '0' AND '7';`;



        return db.execute(sql);
    }



    static getListingById(id) {
        let sql = `SELECT * FROM posts WHERE listing_id=${id}`;
        return db.execute(sql);
    }



}


module.exports = Register;
```

Why we picked this code:

We picked this piece of code because we have trouble with the routing of the post listing and we spent more than 5 days trying to solve this problem. This piece of code is important for this matter since it tells us what the user should input so that the data can be transferred to the database.

Review from other team:

- backend sql sanitation: even if your front end also validates the data, if there's a point where that is forgotten that can be a huge vulnerability

- for search(), look into default parameters. There's a way you can specify default values for parameters in the function definition `function search(search="*", filters=None, sorting=order) {` since that can help with code reusability.
- This is a personal choice, but I would name your variables more explicitly with what they contain since that helps with readability. (This mainly applies to sql in this snip

5. Self-check on best practices for security

Major assets:

- User login information
- User contact info
- Listing information
- Login sessions
- Reviews

Encrypted password

- Using bcrypt library to change the password into hash password for protection and security purposes.

reg_user_id	firstName	lastName	password	email
13	Pat	Thettick	$2b$10$xbVCQ.JhqhqzbY8mH/zXT.XvLn5dwf1FlHj3tq3X9VMNRv/YSeW6i	patthettick@gmail.com
14	Abby	Boren	$2b$10$CIg49J/IuAzNL94BqSsMneV3zHkf8h0dHCZk9YosII1qVVu/Hr9NW	abbyboren@gmail.com
15	Daniel	DeGuzman	$2b$10$DAJL7IBPqyTLCuZU/P6Z5umwRW2I73LH2Qc.9Pptz.rJNq.04HlQ.	daniel@gmail.com
16	Daniel	Smith	$2b$10$aV5qSVm.dutbQ7E7liUsqOJo6e2Ul502v5USo7kzX6U2ZlOF0YgCe	daniel@gmm
17	Bethy	Lam	64f274d7339bb87570954c876d19da61d93e02a91c15c07ce8f2655612770d95892f58c49e95ae060fe0b4f7b74…	bethy@gmail.com
18	Lynda	Lum	65abc6c65c60f73918a99354510ed9bd351c04ffcdd85de4615bac89fd532bced1591968ba488265f1afde052edfd…	lynda123@gmail.com
19	Randell	Park	$2b$10$6oe7ZqkouHmoNX3JtRbXvOxT1CFMEswqQEOAAVeWw3z7TSlYUhKVq	randell@gmail.com
20	Lula	Herby	$2b$10$YSGh3higOfEnC/w8fv4LfuL2FRp0zyoAC8ADOwYdOj8rABLvfIrA.	Lula@gmail.com

When a user creates the account, the backend will do the password, email validation such as:

- Passwords: password input requires an upper case, a number, and a special symbol. After that, the password will be checked with a confirmed password.

- Email: The system will check the database if there is already an email in the db or not. Then it will check the format of an email.

```
exports.createUser = async (req, res, next) => {
    try {
        let { firstName, lastName, password, email, confirmPassword, role } = req.body;
        const hashpassword = bcrypt.hashSync(password, 10);
        if(role != "landlord"){
            role = 'renter';
        }
        let register = new Register(firstName, lastName, hashpassword, email, role);
        let count = await Register.checkEmail(email);

        if (count[0] != 0) {
            res.status(409).json({ message: "Email already exists! ", count });
        }
        else if (!validator.validate(email)) {
            res.status(409).json({ message: "Incorrect email format" })
        }

        else if (confirmPassword != password || !checkPassword(password)) {
            res.status(409).json({ message: "Incorrect password" });
        }
        else {
            register = await register.save();
            res.status(201).json({ message: "User created " });
        }
    }
    catch (error) {
        console.log(error);
        next(error);
    }
}
```

6. Self-check: Adherence to original Non-functional specs

Criterion	Requirements
1. Capability	2.f. When a User that is not Registered tries to use a feature that requires an account, the system shall prompt the user to register or login -DONE 3.a. The website header must include a search input field as the search feature -DONE 3.c. The website header must have a login button that navigates users to login to their registered account if not already logged in -DONE 3.d. The website header must have a button to to navigate users to their profile if they are logged in -DONE 7.b. Users shall filter Listing searches by address by typing an address or zip code in the search bar -DONE 7.c. Users shall filter Listing searches by price by typing a lower bound and an upper bound-DONE 7.d. Users shall filter Listing searches by rating by selecting a star value -ON TRACK 8.b. A “No Listings found” will pop up if a specific search is unavailable. -ON TRACK 8.c. The system will give suggested Listings if no listings were found -ON TRACK 7.a. Users must choose a specific search filter to be able to filter listings -DONE 8.a. Search Results will not pop up if the User does not use the Search bar. -DONE 10.a. User must be registered to be able to post rating -DONE 10.b. Registered Users must be able to rate a property by selecting how many stars out of 5 -DONE 10.c. Users must have rented out a Listing to be able to post a rating. -ON TRACK 10.d. Landlords can reply to a comment once it is posted -ON TRACK 10.e. Rating will not be posted without choosing a star rating beforehand -ON TRACK 11.a User must write a minimum of one character to Post a Review. -ON TRACK 11.b. User will be able to post reviews by clicking on “Post

	Review” -DONE 17.b. Registered Users shall be provided contact information to the Customer Service -DONE 17.c. Registered Users shall give feedback about the site by rating several aspects of the site out of 10 -ON TRACK 17.d. Registered Users shall rate Customer Service -ON TRACK 17.e. Registered Users shall rate how easy the site is to navigate -ON TRACK 12.h. The landlord should be able to upload pictures and description to EZrent by clicking on “POST listing” - DONE 12.a. Users must fill out a property information form to post a listing -DONE 12.b. User must insert images only from their computer -ON TRACK 12.c. User cannot submit form unless required input it filled -DONE 12.d. User cannot submit form unless TOS is agreed -DONE 12.e. User must agree to TOS by checking the TOS checkbox -DONE 12.f. User must be a registered user to post a listing -ON TRACK 12.g. User must be logged in to post a listing -ON TRACK 13.b. System must show landlord info to user only if user clicks on the listing post - DONE 13.f. The landlord should be able to edit whether a listing is currently available or not by clicking on “manage listings” - ON TRACK 13.g. Landlords shall be able to upload pictures from their file system as their Profile picture -ON TRACK
3. Storage	3.a. Registered User Account information must be stored in the database -DONE 3.b. Registered User Profile information must be stored in the database -DONE 3.c. Property Listing information must be stored in the database -DONE 3.d. Reviews must be stored in the database -ON TRACK 3.e. Comments must be stored in the database -ON TRACK 3.g. Saved Searches must be stored in cached memory -ON TRACK 3.k. Login sessions must be stored in cookies -DONE
6. Security	6.a. Web server shall have a not easily guessed password -DONE 8.e. User needs to type a password to be able to Register -DONE 8.f. System must notify User if password meets requirements once clicked out of the password field -DONE 8.g. Passwords must be at least 8 characters long -DONE

	8.h. Passwords must contain at least 1 uppercase letter -DONE 8.i. Passwords must contain at least 1 lowercase letter -DONE 8.j. Passwords must contain at least 1 special character -DONE 8.k. Passwords must contain at least 1 number-DONE 8.l. Emails must be unique to one Registered User -DONE 8.m. Users must confirm password by typing it again -DONE 8.n. The password field must be hidden 2.a. The system shall allow users to reset their password by clicking “forgot password?”- ON TRACK 2.b. The system shall allow users to request their username by clicking “forgot username?” -ON TRACK 2.c. User must type in their password to be able to login -DONE 2.d. User must type in their email to login -DONE 2.e. The password field must be hidden -DONE 16.a. User must comply with regulations to be able to comment -DONE 16.b. User will get banned if not following regulations -ON TRACK 16.c. Users can be blocked by Registered Users -ON TRACK
7. Look and feel	7.a. Website shall use sans serif fonts for headers-DONE 7.b. Website shall use sans serif fonts for buttons-DONE 7.c. Website shall use serif fonts for paragraphs-DONE 7.d. Website shall have a green and white color scheme-DONE 7.e. Website shall have a set of consistent button shapes-DONE 7.f. Website shall have a set of consistent input fields and forms-DONE 7.g. Website shall be able to present information clearly with any browser size-DONE 3.e. The website header position must be fixed on the top of every page-DONE 4.a. Home Page must feel inviting to new and returning users-DONE 4.b. Home Page must allow Users to get started right away-DONE 4.c. Users will be able to maneuver through the Home Page by clicking on different features.-DONE 5.a. The website footer position must be fixed on the bottom of every page-DONE 7. Users will click one of the 5 stars to select a rating value-DONE 17.a. Registered Users shall be introduced to the Help Page with a description of the website and an FAQ section -DONE 13.a. System must show property details to user only if user clicks on the listing post -DONE 13.c. Property details must show property size only in square feet -DONE

	13.d. Property details must show property price only in dollars -DONE
8. Privacy	8.a. User needs to type first name to be able to Register-DONE 8.b. User needs to type last name to be able to Register-DONE 8.c. User needs to type an email to be able to Register- DONE 8.d. System must notify User if email is not in email format once clicked out of the email field -DONE
9. Network	9. Connect the database to aws -DONE
12. Testing	12.a. Testing must be done on localhost
13. Coding Standards	13.b. Styling must be specified by element classes -DONE 13.c. There must be a script file for each page that submits information -DONE 13.d. Handlebars file names must match the express routes that their associated with -DONE 13.e. Controllers are there to manage how the user interacts with listings and how general guest becomes a user -DONE

7. List of Team Contributions

Devon Dy-Liacco	Documentation: ● Assign tasks ● QA tests ● Self check on best practices for security ● Internal review
Score: 7/10	Beta Prototype: ● Assign tasks ● Reconcile frontend and backend ● Get feature landlord
Praise O Eubany	Documentation: ● Wireframe revision ● Storyboard revision ● Mockup revision ● Website styling ● Made document 4 ● Usability testing ● Self-check: Adherence to original Non-functional specs
Score: 9/10	Beta Prototype: ● Github management
Youssef Hammoud	Documentation: ● Provided code for code review ● Explained why we chose internal code
Score: 8/10	Beta Prototype: ● Redesigned home page ● Styled home page ● Styled listing page ● Created home page animation ● Restyled login and registration pages ● Styled listing page ● Changed most hard coded data to dynamic data
Issa Shihadeh	Documentation: ● Commented a piece of code and sent it to team 5 ● Gave code review to team 5
Score: 8/10	Beta Prototype: ● Started sessions ● Post listing

	● Set up flash in the backend ● Set up uploads ● Set up comments table in the database
Tung Nguyen	Documentation: ● QA tests ● Self check on best practices for security
Score: 9/10	Beta Prototype: ● Create Post Review ● Get User Profile with review ● Fix Post Listing ● Update Landlord rating ● Add look up ip address ● Modify tables in db ● Fix frontend input to backend
Ricardo Lopez	Documentation: ● Helped edit older Milestone Documents ● Edited Storyboard Mockup ● Helped with Wireframe revision ● Edited non-functional requirements ● Helped create the Usability Test Plan ● Usability Test Plan ● UI/UX Testing
Score: 9/10	Beta Prototype: ●