SW Engineering CSC648/848

GatorRoom

Team 103

Milestone 4

5/8/2019

Nico Graves - theredhead242@gmail.com
Feona Garcia
David Hernandez
Michael Tran
Jahon Khusanov
Carlos Velasco
Michael Nelson

Turned In: 5/8/2019
Revised: 5/17/2019

GatorRoom

http://gatorroom.xyz/

- SFSU students may sign up using their school email, and make a custom account with their own search filters.
- SFSU students may upload their apartments/houses that have vacant rooms looking for other students to fill them.
- Private landlords may sign up for landlord accounts.
- Landlord can upload private listings for students to rent.
- Users can view house listings based on a variety of search features (Zip Code, City, Etc…)
- Students may set search filters to only see houses that fit their needs.
- Students can browse houses listed and see a description of the listings specifications as well as find a contact email for the renting landlord.

QA test plan

I. Unit Test

- Features to be tested : Apartment Search and Account Sign Up
- HW and SW setup :
  - Chrome Version 74.0.3729.131 (Official Build) (64-bit)
  - Mozilla Firefox(for Linux Mint) Version 61.0.1 (64-bit)
  - Linux Mint 18.3 Mint 18.3.3.6.7 (64-bit)
  - MacOS Mojave 10.14.3 (18D109)
- Actual test cases:

Test #	1	2	Result
Operating System	Mac	Linux Mint	n/a
Browser Type	Chrome	Mozilla Firefox	n/a
Feature #1: Search	Free Text Search & Additional Features	Free Text Search & Additional Features	PASS
Feature #2: Account Sign Up	Database Updates & Successful Portal Creation	Database Updates & Successful Portal Creation	FAIL

- Test Coverage Analysis:
  The search bar can successfully search free text like “Daly City” and “San Francisco” along with zip codes. User sign up has not been fully implemented yet, it is only minor touches away from finished. The form for user sign up/ account creation populates the screen, but the submission to our database is not fully completed yet. We have both the landlord and the student portals completed, filters and image carousel done.

II. Integration Test

- User story to be tested:

Story 1 : A 1st year student Harry has never rented an apartment. He would like to live close to campus and meet new people. His hometown is far from SFSU and he probably will have to rent a place without actually seeing it. He wishes he could view listings on a “map view” near the school with important student features like transit, grocery stores, gyms, laundry and other amenities highlighted to help him choose his housing.

- HW and SW setup:
  - Chrome Version 74.0.3729.131 (Official Build) (64-bit)
  - Mozilla Firefox(for Linux Mint) Version 61.0.1 (64-bit)
  - Linux Mint 18.3 Mint 18.3.3.6.7 (64-bit)
  - MacOS Mojave 10.14.3 (18D109)

- Test Values:

***TABLE 1.0 : Using Chrome on MacOS**

Feature	Case Type	Case Desc.	Dates	Scenario	Prereqs.	Data	Results
Login	Smoke & Unit	Build verification ready for user portal	05/05, 05/06	Validation responsive; Portal successfully populates in a timely manner.	A valid account pre-registered in the database.	Harry's (login info): email & password	FAIL
Home Page	Smoke & Unit	Build verification for general visitors/guest users.	05/05, 05/06	No dead links; Responsive. Harry wants to live close to SFSU.	n/a	n/a	PASS
Search	Smoke & Unit	Build verification for further pages, as well for filters (ie; details page). Build verification for maps.	05/05, 05/06	Shows search results close to campus, scalable, no dead links; Map and listings populate providing visual distance.	Desirable areas for residency, SFSU relative zip codes.	San Francisco, San Fran, SF, San Francisco State University, Daly City, 94132, 94114	PASS
Details Page	Smoke & Unit	Build verification for contacts, filters and routing.	05/05, 05/06	Corresponds with map feature. Details page with active contact link.	Desired filters/amenities in mind.	Laundry, gym, close to public transit system, etc.	FAIL

***TABLE 2.0 : Using Mozilla Firefox on Linux Mint**

Feature	Case Type	Case Desc.	Date	Scenario	Prereqs.	Data	Results
Login	Regression	Test latest log in communication with database.	05/07	Same as TBL 1.0	Same as TBL 1.0	Same as TBL 1.0	FAIL
Home Page	Regression	Test minor listing changes.	05/07	Same as TBL 1.0	Same as TBL 1.0	Same as TBL 1.0	PASS
Search	Regression	Test for filters implementation, map updates and image thumbnails.	05/07	Same as TBL 1.0	Same as TBL 1.0	Same as TBL 1.0	PASS

Details Page	Regression	Test for contact functionality.	05/07	Same as TBL 1.0	Same as TBL 1.0	Same as TBL 1.0	FAIL

- Test Coverage Analysis:
  Although we have been using CircleCI as our primary source for integration testing, these manual unit/smoke/regression tests have as well been used to monitor the progression of our site/keep bugs to a minimum. Moreover, the analysis over the past 3 days prior to M4's submission have proven to be promising. The features that did not successfully pass have minor bugs that will be resolved in the coming days prior to final submission. As mentioned in unit testing, Log in/Sign up forms populate correctly but can not push through database yet. Details Page populates and is scalable. Maps is scaled correctly and with pins displayed, zoom in/out and full screen/escape works perfectly. Maps however still needs to be connected to listing requests when the pins are clicked to generate a listing pop up. Most if not all links worked, and the site is overall responsive.

Beta Test Plan

Website URL: gatorroom.xyz

I. Objectives

The objective for this test is to allow the user to easily navigate through our website, and fully interact with its features. By the end of this test, student should be able to access their profile by their point of interest, whether they may be a tenant or a student.

- **Objectives for a first-time user who is a student:**

a.) Upon arriving at the landing page, students should be able to easily use the search feature as well as clicking on the profiles of the site creators to check out their "About" Pages

b.) As a first-time user, students should be able to sign up for an account on *Gator Room.*

c.) The database should store the student's information and will allow the user to log in at any moment, if they choose to do so.

d.) At the student portal page, students should be able to edit their initial listing of their current property, while also editing their profile, with a short description of themselves along with a Profile Picture and their name displayed. Students should also be able to edit amenities filters along with a budget filter and distance to campus.

- **Objectives for a first-time user who is a tenant:**

a.) Upon arriving at the landing page, just as students, tenants should also be able to easily use the search feature as well as clicking on the profiles of the site creators to check out their "About" Pages

b.) As a first-time user, tenants follow the same steps as students who start their initial sign up for an account on *Gator Room*.

c.) The database should store the tenant's information and will allow the user to log in at any moment, if they choose to do so.

d.) At the tenant portal page, tenants have one listing that they should be able to upload onto the site so students could view it. Just like the students, tenants can edit their profile care with a profile picture, a short description about themselves and an email. This email can be used for students to tenant contact.

- **General Objectives:**

a.) Upon arrival to the landing Home Page, the user should be able to log in using their registered email and password. The registered email should bring the user to their destined portal page; either student or tenant.

b.) In the home page, the user should be able to type in a surrounding zip code and be able to use the search bar and then be taken to the listing page where the user my browse for current listings.

c.) In the Listings page, the user should fully utilize the map API and the listings should match with its given pinpoint.

d.) For the Sign-Up component, the user should be able to input his or her information in to the text edits, along with a password which would then be saved in the database.

e.) In every page of the site, the search feature, login, sign up, and home buttons should be easily navigable and take the user to its given feature.

II. Test Plan

- **System Setup**: User types in the site's URL address: *gatorroom.xyz* in their browser of their choice. The user may enter this site using either mobile or through their PC.
- **Starting Point**: The user is greeted with the homepage of gator room. To begin their sign up for the website, they would have to navigate to the sign-up button on the top right of their screen. This would take them into a new page which they would then input their information. After they are finished, the user is then taken back into the homepage which they are now allowed to login using the log-in button beside the sign-up button.

  Once logged in, the user has options:

  1). Students can enter their information and upload their current listing after reaching their student portal page.

  a. Generally, the student may go back to the home page by clicking the Home Page button on the top right of the screen. From there, they may use the search bar and its functions to browse local listings.

  2.) Tenants can also enter their information and upload their current listing after reaching their tenant portal page.

  a. Just like if the user is a student, tenants may return to the homepage and browse local listings.

- **Intended Users**:

1.) Students looking for local listings near San Francisco State University, who desire to have other housing options aside from living inside campus with the University's dorm option.

2.) Tenants that can upload their living space as a listing in which students can view and contact the Tenants.

- **Feedback Process**:

When the student is interested in their desired listings that fits their personal tastes, the student may contact the listings tenant via email. The two may communicate with one another through e-mail and may exchange phone numbers, information, etc.

- **Functionality**:

The function that any given user will use the most would be the search bar. The search bar is for general use, not just for account users. Typical online browsers may not want to sign up for Gator Room but may be interested to view its features and possible listings. Users are most likely to visit the listing page more than their portals. The portals are mostly used for users with interest in a given listing. Editing and uploading their information into their portals may vary by time depending on how detailed the user chooses their listings to be.

Code Review

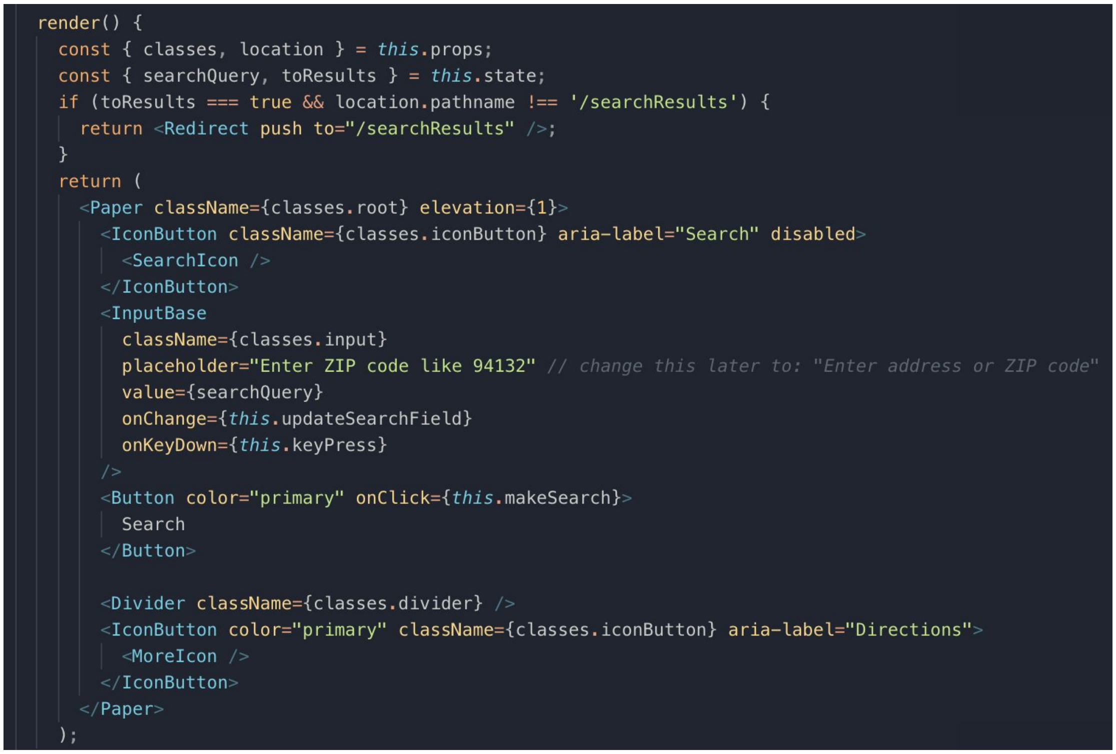
Coding Style

To maintain a consistent coding style for this project, we installed a library called esLint. Essentially what the library does is add an additional set of compiler rules that will raise an error indicator when code deviates from our standard chosen coding style. We installed AirBnB's esLint ruleset, so our coding practices are identical to a highly successful development companies (https://github.com/airbnb/javascript). We also use CircleCI to enforce coding styles along with the Prettier extension on VSCode.

Process of Code Review

1. One team member should submit code to other team member(s) for peer review. When someone is done coding their bit of the project, one must create their own branch to push to. The individual will be prompted to submit a pull request through GitHub.
2. Once the pull request is made through GitHub an admin/member has to review it, approve the request and merge. The admin or member who approves this can review the feedback from CircleCI which automatically generates on its own.
3. Comments in code are always recommended but we also push for descriptive commit messages, and are big on communicating changes/complications via our primary communication platform;Discord. Below is an example of a code segment that could be submitted for review with minimal comments. The styling of the code is very clear and concise. With every member following the same coding styles we develop the consistency and ease that many group projects lack and usually struggle with.

***Example 1.0 : Render() for Search Bar**



```
render() {
  const { classes, location } = this.props;
  const { searchQuery, toResults } = this.state;
  if (toResults === true && location.pathname !== '/searchResults') {
    return <Redirect push to="/searchResults" />;
  }
  return (
    <Paper className={classes.root} elevation={1}>
      <IconButton className={classes.iconButton} aria-label="Search" disabled>
        <SearchIcon />
      </IconButton>
      <InputBase
        className={classes.input}
        placeholder="Enter ZIP code like 94132" // change this later to: "Enter address or ZIP code"
        value={searchQuery}
        onChange={this.updateSearchField}
        onKeyDown={this.keyPress}
      />
      <Button color="primary" onClick={this.makeSearch}>
        Search
      </Button>

      <Divider className={classes.divider} />
      <IconButton color="primary" className={classes.iconButton} aria-label="Directions">
        <MoreIcon />
      </IconButton>
    </Paper>
  );
```

Self-check

Adherence to original Non-functional specs

1. Students can browse the student portal and view listed houses.
   DONE
2. Landlords can upload houses through the landlord portal to the house database.
   IN PROGRESS
3. Users can set filters to their profile to affect the houses listed for them.
   a. Smoking Habits
   b. Pets
   c. Budget
   d. Age of roomates
   e. Gender of roomates
   f. Location of houses
   g. Disabilities
   h. Amenities
      i. Gym
      ii. Office
      iii. Study/Public area
      iv. Onsite Laundry
   i. Parking
   DONE
4. Landlords can set specific rules for different houses they list.
   a. Smoking
   b. Pets
   c. Visitors Allowed
   IN PROGRESS
5. Students can view a map of the location of the listed houses.
   DONE
6. Students can ask their landlords for permission to become master tenants.
   N/A
7. Landlords can approve master tenants.
   N/A
8. Master tenants can list the house they live in with vacancies just like landlords.
   DONE