GatorList

SW Engineering CSC648/848 Spring 2019

Team 4

Karuna Nayak(knayak@mail.sfsu.edu), Victor Muñoz, Huawei Gao, Dylan Shwan,
Daniel Mossaband,
Gabriel Alfaro & Aditya Sheoran

Milestone 1

Date Created: March 4, 2019

Date Submitted: March 6, 2019

Date Revised: March 21, 2019

I. Executive Summary:

Finding a place to live in a popular city like San Francisco can be daunting, especially when you are a student just looking to graduate college. To address this, we introduce GatorList, a web application that offers an easy and simple way to connect with a renter or rentee that is tailored specifically to San Francisco State University students as well. We believe in a no nonsense and getting straight to the point approach so you can find rentals on your own terms. The application will be able to help many incoming freshmen who were unable to get on-campus housing. This web application is crucial for SFSU students, especially for those that will not graduate in 4 years or cannot afford staying on-campus housing for those 4 years. No other web application can search specifically by distance to San Francisco State, making GatorList the single best tool for finding a suitable place to live while attending SFSU.

GatorList in general is an online website that creates an intermediary between the renter and the rentee, removing the pain of having to look through countless websites and comparing each of them. For students, GatorList will offer listings of apartments, rooms, houses, and the ability to contact the landlord or owner for that listing. We are offering SFSU students the opportunity to be able to find housing, either before or during the academic school year, in a seamless fashion. Apart from the standard information such as pricing and room layout, GatorList offers the ability to see distance and time it takes to get to campus. With funding, not only can this application have future updates and more exciting features, the concept can be applied to other schools around the United States. We wish for all students to experience the luxury of having access to renting websites tailored to their colleges.

We are a team of seven programmers working on this application and we are preparing to launch in mid-May. We are very eager to delve into the world of website programming and use languages such as MySQL. With fully dedicated front-end and back-end teams, we will not only be getting the User Interface right, but also the back-end database architecture as well. We will have a fixed budget for now and plan to use the necessary API's to keep the basic functions simple. This will allow us to debug our functions while improving the basic flow of the website.

II. Personae & Main Use Cases:

Personae:

1. John is a graduating senior from Berkeley High School who excels in baseball. Like most high school seniors, he has applied to many different colleges and universities across the country. Unfortunately, John was not the best student and has no definite plan regarding what he wants to major in. After several rejection letters from some of the top universities, John received his first acceptance letter from San Francisco State University. Excited, John browses the internet for a place to stay near school because he does not want to live at home with his mom while in college. As a high school graduation present, his parents bought him a new car. John does not mind driving a little bit to get to school; however, he does not want to experience the heavy traffic that comes with driving along the Bay Bridge. Being a young adult, John has great computer skills.

2. Mary is a 70 years old lady who owns 5 apartments in San Francisco and Daly City. Mary has experienced her fair share of bad tenants and wishes to help her fellow alma mater by solely offering SFSU students a place to live. Mary has very basic knowledge of the world wide web; however, she can navigate her way through a website if it is simple

enough. Being handicapped, she can no longer post flyers around the school and wants to find a website where she can easily post her listings. All her apartments have been rented to San Francisco State University students. One of her tenants, George, is about to graduate and has informed Mary that he will be moving out the day he graduates. George thanked Mary for her hospitality and kind heartedness, but now it's time for Mary to find another SFSU student to fill her soon to be vacancy.

3. Sandra is a junior at SFSU majoring in Biology. She has lived on-campus for the first two years of her college degree. She has made some very good friends while living on campus; however, she is tired of the outrageous housing fees and being forced to move back home during the summer. Sandra and a few friends want to move out of their small on-campus housing to a bigger apartment. Having been living on-campus for two years now and exploring San Francisco with her friends, she is quite capable of navigating her way through the city using public transportation. Since Sandra does not own a car, finding an apartment that is near school is of the utmost importance.

4. Richard recently graduated from SFSU with a B.S. in Computer Science. He along with a few friends have created a startup company called 'GatorList' that allows SFSU students can search for housing. Richard has both great technical and communication skills, which have been extremely helpful when creating the company.

Use Cases:

Use Case #1 (Unregistered User Tenant): John visits the ‘GatorList’ website and searches for apartments available for rent using price, number of bedrooms and bathrooms. John sorts the apartments by price and distance. John manages to find an apartment near campus. Liking what he saw, John attempts to contact the landlord. A registration form pops up, John creates an account on the website and gets in touch with the landlord.

Use Case #2 (Unregistered User Landlord): Mary visits the ‘GatorList’ website to advertise her apartment. A page prompts to register. Mary register herself. Mary successfully uploads the photos of the apartment, provides the address, number of bedrooms, number of bathrooms, monthly rent of the apartment and description. Mary accepts the terms and conditions to post ethical content and submit the request. Upon submitting the listing, Mary is notified that her listing is waiting for admin approval.

Use Case #3 (Registered User Tenant): Sandra is a registered user of ‘GatorList’ and checks the apartment listing regularly. She sorts her search by distance from the university to find a place she and her friends would like. One day Sandra logs into the website and check out the photos and details of a new listing. She then contacts the landlord to schedule an appointment to visit the complex.

Use Case #4 (Administrator): Richard is an admin for ‘GatorList’ and has the right to block/unblock users and approve/disapprove listings on the website before they go live. Richard logs in and checks for any new listings. He makes sure all the uploads are appropriate regarding the guidelines and approves the listing. The new listing can be seen by everyone upon visiting the website.

III. List of Main Data Items and Entities:

Unregistered User: An Unregistered User will be a user who has not logged in to the website. There will be no information stored about this user. This user is one who wants to browse available rooms for rent. This user will not have access to any feature other than being able to search and sort a list of available rooms for rent.

Registered User: A Registered User will be a user who has made an account, is currently logged in and is viewing the browsing section of the website. The information stored about this user will be their first name, last name, username, email address, account creation date, as well as their salted and hashed password. This user is one who can message tenants or post listings.

Administrator: An Administrator will be those who will manage this application. There is only a password stored for this user that will be salted and hashed. Administrator can approve or delete listings as well as delete existing users. The Administrator will not be able to post.

Registered User Dashboard: This is a screen that only Registered User can access. This will display all the rooms that the Registered User has starred.

Login: Login is a page that allows Registered User to login via their email address and password

Logout: Logout is a button that allows Registered User to logout from their account

Register: Register is a page that allows Unregistered User to create an account by supplying their first name, last name, username, email address, account creation date, and password.

Message: Registered Users can send messages to the Landlord

Listing: The items that will be under listing are, listing_unique_id, creation_date, last_day_for_rent, number_of_rooms, address, number_of_bathrooms, furnished_status, unique_landlord_id.
User shall be able to browse by the categories which are housing types, zip code, city name, area name, number of bedrooms/ bathrooms, price range. The listings can be sorted based on posted date, price and distance to the campus.

Database Tables: There will a single database table for Registered Users as well as Tenants for they are both technically the same user. This table will hold the user's unique ID, first name, last name, username, email address, account creation date, as well as their salted and hashed password.
There will be one table that stores conversations between Registered Users and Tenants. This table will hold the message sent, the timestamp of the message sent, sender unique ID, and receiver unique ID.
There will another table that stores the rooms that the Tenant is for sale. This table will have the Landlord's Unique ID, creation date, last day for rent date, number of rooms, address, number of bathrooms, and furnished status.

Listing Form: Required for registered users (landlord) to upload information about the house/apartment they want to advertise. Contains photos, address, blank text field. This form will then prompt users the completion of uploading and status shows whether this
form is approved by an administrator.

Registration Form: Required for users to register. Contains name, email, phone, job, address(optional). All the information will be then stored in the database.

Privacy: Data is used only by 'GatorList' for tracking customer usage and shall not be shared with anybody else. Registered users may be contacted with questions, promotions, redirected website message.

IV. Initial List of Functional Requirements:

Unregistered Users:

1.1 Unregistered users shall be able to browse through available listing information.
1.2 Unregistered users shall be able to sort by posted date, price and distance.
1.3 Unregistered users shall be able to search by keyword defined by location, room# available, type of apartment/house, etc..
1.4 Unregistered users shall be able to see listing information related to their search.
1.5 Unregistered users shall be able to register.
1.6 Unregistered users shall be prompted to register or sign-in upon confirmation or contact landlord.

Registered Users:

2.1 Registered users shall have all the function of unregistered user has.
2.2 Registered users should accept the terms and conditions upon registration.
2.3 Registered users shall be able to login.
2.4 Registered users shall be able to view their own dashboard and edit information.
2.5 User dashboards shall show all personal information.
2.6 User dashboards shall show all viewed and tagged listing information.
2.7 Registered users shall be able to message landlords.
2.9 Registered users shall be able to post new listing.
2.10 Registered users shall be prompted to confirm to post the new listing.
2.11 Registered users shall be prompted a status of the newly posted listing whether being approved or disapproved by an admin.
2.12 Registered users shall be able to edit/delete their listing.
2.13 Registered users shall be able to change their personal information.
2.14 Registered users shall be prompted to sign the declaration to post ethical content while posting new listing.

Admin:

3.1 Admin shall have an admin panel.
3.2 Admin shall be able to approve/disapprove all listing through admin panel before they go live.
3.3 Admin shall have all the function of unregistered/registered user has.
3.4 Admin shall be able to review all listing through the admin panel.
3.5 Admin shall be able to block/unblock users through the admin panel.

V. List of Non-functional Requirements:

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team, but all tools and servers must be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices

4. Data shall be stored in the team's chosen database technology on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be added
10. No e-mail clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
15. 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 to not confuse this with a real application).

VI. Competitive Analysis:

	Zillow.com	Redfin.com	Trulia	GatorList
Search Bar	++	++	++	+
Browse	++	++	++	+
Categories	++	++	+	+
Login	++	++	++	+
Message	+	+	+	+
Map	+	+	+	+
Students Only	-	-	-	+
Commute Time to SFSU	-	-	-	+
Distance from SFSU	-	-	-	+

++ Superior, + feature exist, - feature does not exist

The market is constantly changing and keeping up with it is becoming hard for customers, especially for those who looking for a place to move into within a short amount of time. While companies like Zillow, Redfin, and Trulia offer good opportunities for people to find a place in their price range, it does not help students of SFSU to easily find a place. Since the students are our target audience, we decided to specialize in matters that would interest them, such as information regarding commute time to SFSU and overall distance from campus. By tailoring specifically to the needs of the student, we can provide a service that outclasses our competitors.

VII. High Level System Architecture & Technologies Used:

- Server Host: Amazon Web Services 1vCPU 3.3GHz/ 1GB RAM
- Operating System: Ubuntu 18.04.1 LTS
- Database: mySQL 14.14
- Web Server: Apache 2.4.29
- Server-Side Language: PHP 7.2.15
- **Additional Technologies:**
    - Back end Framework: Laravel
    - Front end Framework: Bootstrap 4 / Vue.js
    - Web Analytics: Google Analytics
    - SSL Cert: Lets Encrypt (Cert Bot)

VIII. Team:

Team Member	**Front End/Back End**	**Special Role**
Karuna Nayak	Front End & Back End	Team Lead, Github Master
Victor Muñoz	Front End	Front End Lead
Huawei Gao	Front End	-
Dylan Shwan	Front End	Document Master
Daniel Mossaband	Back End	Back End Lead
Gabriel Alfaro	Back End	-
Aditya Sheoran	Back End	-

IX. Checklist:

Task	**Completion Status**
Team found a time slot to meet outside of the class	DONE
Github master chosen	DONE
Team decided and agreed together on using the listed SW tools and deployment server	DONE
Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing	DONE
Team lead ensured that all team members read the final M1 and agree/understand it before submission	DONE
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	DONE