Final Project for SW Engineering Class CSC 648-848

Section 01 Fall 2016

Team 05

GatorRent

Local

Soumithri Chilakamarri (Team Lead) schilaka@mail.sfsu.edu

Guanming Pan (C.T.O.)

Matthew Wishoff

Kevin Fang

Jeffrey Ilar

Emil Santos

URL: http://sfsuswe.com/~f16g05/Group5/

19/12/2016

Index

I. Product Summary

GatorRent

GatorRent is a rental website which allows SF state students to rent apartments near the SFSU campus. The UI is designed to be user-friendly and comes with a powerful search. The home page displays the pictures of apartments that are recently added to the website. Users can search for the apartments entering either pincode, area or the city. The search results show a list of apartments that can be rented. For example if you enter a city name in search bar and press the enter button, it will show the list of apartments (images) as per the city name. The user can then click on the view button to find more information about the apartment. After selecting the view button, it displays the apartment details that has the apartment picture along with the google map location of the apartment. Like this, the user can view the location of the apartment using the google maps.

In order to contact the leaseholder, the user must login to the GatorRent website. In case of a new user, he/she should register to the website by entering the details in the registration form present on the home page. After the user is successfully logged in to the website, he/she can contact the poster by clicking the contact poster button available on apartment details page. A message will be sent to the leaseholder that helps to contact the user.

If you have an apartment to rent, GatorRent provides you to post the apartment on the website so that SFSU students can rent them. To post an apartment at GatorRent, the poster must login to the website and if case of a new user, he/she should register first before posting. Once the poster is logged in, he/she can post the apartment by clicking Post an Apartment button present on top in the GatorRent. The poster can then fill a form regarding the apartment details to be posted. The poster must upload a picture of the apartment in the form if he/she wishes to post on the website.

Product URL :- http://sfsuswe.com/~f16g05/Group5/

S.No.	Functional Specification	Priority
1	The **Student** shall be able to search the website for rental listings	1
2	The **Student** shall be able to create an account	1
3	The **Student** shall be able to login to an account	1
4	The **Student** shall be able to contact a Leaseholder or other Students	1
5	The **Student** shall be able to view pictures	1
6	The **Leaseholder** shall be able to create an account	1
7	The **Leaseholder** shall be able to update their posting	1
8	The **Leaseholder** shall be able to login to their account	1
9	The **Leaseholder** shall be able to list posting(rentals)	1
10	The **Leaseholder** shall be able to post a picture of their rental	1
11	The **Administrator**s shall be able to log into an administrative account through MySQL Workbench	1
12	The **Administrator**s shall be able to delete posts	1
13	The **Administrator**s shall be able to remove/ban users	1
14	The **Leaseholder** shall be able to mention special details	1
15	The **Student/Leaseholder** shall be able to logout to their account after he/she login to their account	1
16	The Student shall be able to contact the Leaseholder	1

17	The Leaseholders shall able to see their listings after they login to their account	1
18	The **Students** shall be able to view the location of an apartment using Google Maps	1
19	The **Leaseholder** shall be able to view the contact information of the **Student** when he contacts the **Leaseholder**	1

Note: Leaseholder refers to the poster and a Student can be a user.

II. Milestone Documents

SW Engineering CSC648/848 Section 01 Fall2016

On-line apartment rental site tailored to SFSU students

GatorRent

Team 05

Soumithri Chilakamarri (Team Lead) schilaka@mail.sfsu.edu

Guanming Pan (C.T.O.)

Matthew Wishoff

Kevin Fang

Jeffrey Ilar

Emil Santos

"Milestone 1"

10-05-2016

History Table

S.No.	Version	Revision Date	Revision Description
1.	1.0.	10-05-2016	First Draft
2.	2.0.	10-12-2016	Final Draft 1. Added Roommate finder feature to Executive Summary 2. Added 2 new Data Definitions - 3.4-Roommate_Info, 3.5-Apartment_for_Rent 3. Changed Functional Specification-4.1.3 4. Added Roommate finder in Competitive Analysis table

1. EXECUTIVE SUMMARY

We are a group of students at San Francisco State University creating a website to help SFSU students find and rent apartments more easily. **On December 15th**, we will be releasing Gator Rent a website that allows San Francisco State University students to rent apartments, or find roommate(s) to lease too. **Users will also be able to use our website from their mobile device.** We are going to make sure we launch on time on December 15th by using an agile development structure and methodology. Our product will allow San Francisco State University students to search and rent apartments they're interested in. By being able to filter by things such as price, allowance of pets, distance from the school, smoking allowed or not. A Leaseholder will be forced to create an account before posting so content on the website can be better managed, and people posting against our terms of service can be removed by an admin. A user will also be able to see the location of the house or apartment with the use of google maps. This allows users to have a better sense on where they will be living instead of just giving them an address. One of our special features is called **"Roommate Finder"**. This will allow SFSU students to search for other students at SFSU to be roommates with. Overall our team seeks to make the renting process easier for students attending San Francisco State University. We accomplish this by giving them a website to seek out a place to live for a semester or longer while attending San Francisco State University, and the ability to customize their search for their individual housing needs.

2. USE CASES

2.1. Student:

Tony is a 20 year old **student** at San Francisco State looking to find an **apartment**.With **GatorRent** he is able to narrow his search to apartments with two bedrooms. Tony also **filters** apartments by **price** and **pets allowed** aswell. In just one hour of using GatorRent Tony finds two suitable aprtments and sends a **message** to both posters.

Sandra uses **GatorRent** to search for leaseholders specifically subletting a single **room.** By using the map feature of **GatorRent** Sandra is able to get a nice idea of the relative distance between the listings and school. Within a day Sandra is able to find several apartments in Park Merced subletting rooms.

2.2. Leaseholder:

Dorothy is a 57 year old **leaseholder** to a **house** near San Francisco State and she would like to advertise it for rent. She does not like long **registration** processes on websites. Dorothy wants to post **pictures** of the apartment and also set the **price** she is renting for. She also wants renters to know the house includes its own washing machine and dryer. Lastly she wants to tell renters she is willing to negotiate the price as a **special detail**. **GatorRent** is able to meet all of her needs. After going through the quick registration process; she uploads her pictures, sets the desired price, and indicates the house has a washer and dryer. Before finalizing the posts she writes that the **price** is negotiable. Within three weeks she meets five potential rentees and is able to sublet the house the next month.

2.3. Roommate Finder:

Henry is a **student** at San Francisco State and is looking to replace his roommate. With **GatorRent** he is able to indicate he is looking for a non-smoker who does not drink. Henry also lists specific details of the apartment such as **size**, **number of roommates**, the total rental **price**

that is paid by the roommates each month. Within a week, Henry receives messages from 14 interested **students**.

2.4. Administrator:

Miranda is one of the hired **administrators** of GatorRent. Her job is to review **flagged** postings and **deletes** them if they violate the site's guidelines. She is also in charge of support and as such she can change user passwords and respond to any user messages. On rare occasions she also **deletes** user accounts responsible for repeatedly posting offensive content.

3. DATA DEFINITION

3.1. User : Any person who uses GatorRent is a User. In GatorRent, a user can be Student, Leaseholder or an Admin.

3.1.1. Student : A User who is studying in SFSU is a Student. A Student uses GatorRent for finding the desired room.

3.1.2. Leaseholder : A User who would like to advertise the room for renting in GatorRent is a Leaseholder. A Leaseholder should have an account in GatorRent and a Student can also be a Leaseholder.

3.1.3. Admin : A User who monitors GatorRent is an Admin.

3.2. User_Type : It denotes the type of user - Student, Leaseholder, Admin. This information is used to authorize privileges around the website.

3.3. Account : An Account is the User Information in GatorRent. An Account is created when a user signs up in GatorRent. A user can login into GatorRent using his Account credentials. Every user need not have an account in GatorRent.

3.4. Roommate_Info : Shares info about potential roommate.

3.4.1 Grad_Year : Specifies roommates graduate year

3.4.2 Major : Specifies roommates Major

3.4.3 Minor : Specifies roommates Minor

3.5. Apartment_For_Rent : the rental housing that is advertised on GatorRent.

3.5.1 Num_Of_Bedrooms : Specifies the number of bedrooms.

3.5.2 Num_Of_Bathrooms : Specifies the number of bathrooms.

3.5.3 Floor_Size : Specifies the floor size in square feet.

3.5.4 Rent : Specifies the amount the renter will be paying per month.

3.5.5 Lease **:** Specifies the amount of time the renter will convey the land by contract

3.5.6 Address **:** Specifies the address of the room.

3.5.7 Relative Distance **:** Specifies distance between the room and SFSU, in miles.

4. INITIAL LIST OF FUNCTIONAL SPECIFICATIONS

4.1. Student

4.1.1 The **Student** shall be able to search the website for rental listings

4.1.2 The **Student** shall be able to filter listings by marking box settings that fit their needs

4.1.3 The **Student** shall be able to find the approximate location of renting apartments using Google Maps while preserving privacy of landlords

4.1.4 The **Student** shall be able to create an account

4.1.5 The **Student** shall be able to login to an account

4.1.6 The **Student** shall be able to contact a Leaseholder or other Students

4.1.7 The **Student** shall be able to report false claims

4.1.8 The **Student** shall be able to favorite certain rentals

4.1.9 The **Student** shall be able to view pictures and videos

4.2.0 The **Student** shall be able to create postings looking for roommates in roommate finder

4.2. Leaseholder

4.2.1 The **Leaseholder** shall be able to create an account

4.2.2 The **Leaseholder** shall be able to update their posting

4.2.3 The **Leaseholder** shall be able to login to their account

4.2.4 The **Leaseholder** shall be able to list posting(rentals)

4.2.5 The **Leaseholder** shall be able to post pictures/videos of their rental

4.2.6 The **Leaseholder** shall be able to mention special details

4.2.7 The **Leaseholder** shall be able to specifically lease the room to SFSU students

4.3. Administrator (Admin)

4.3.1 The **Administrator**s shall be able to log into an administrative account

4.3.2 The **Administrator**s shall be able to delete posts

4.3.3 The **Administrator**s shall be able to remove/ban users

4.3.4 The **Administrator**s shall be able to change passwords

4.3.5 The **Administrator**s shall be able to communicate with Students/Leaseholders

4.3.6 The **Administrator**s shall be able to view an admin panel

5. LIST OF NON FUNCTIONAL SPECIFICATIONS

5.1 Security

5.1.1. Application shall be hosted and deployed on Amazon Web Services as specified in the class

5.1.2. Data shall be stored in the MySQL database on the class server in the team's account

5.1.3. Application shall be served from the team's account

5.1.4. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

5.1.5. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.

5.2. Performance

5.2.1. No more than 50 concurrent users shall be accessing the application at any time

5.2.2.Google analytics shall be added for major site functions.

5.2.3. Site security: basic best practices shall be applied (as covered in the class)

5.3. Optimization

5.3.1. Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. It shall degrade nicely for different sized windows using class approved programming technology and frameworks so it can be adequately rendered on mobile devices

5.3.2. The language used shall be English.

5.3.3. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

5.3.4. Pay functionality (how to pay for goods and services) shall be simulated with proper UI, no backend.

5.4 Best Practices

5.4.1. Application shall be developed using class provided LAMP stack

5.4.2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class.

5.4.3. The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Fall 2016. For Demonstration Only". (Important so as to not confuse this with a real application)

6. COMPETITIVE ANALYSIS

Website	Search	Map	Photo	Message	Roommate Finder	Feedback
craigslist.org	Y	Y	Y	Y	N	N
zillow.com	Y	Y	Y	Y	N	Y
apartments.com	Y	Y	Y	Y	N	Y
GatorRent	Y	Y	Y	Y	Y	?

Y: Feature is available

N: Feature unavailable

?: Potential feature

The unique feature of GatorRent, when compared to other rental websites is the Roommate Finder. Roommate Finder allows an SF State Student to rent a room with other SF State Students. By using Roommate Finder, a Student can even share the rooms with another Student pursuing the same major in SFSU.

7. HIGH-LEVEL SYSTEM ARCHITECTURE

The System architecture for **GatorRent** uses two servers - ***sfsuswe.com*** and ***sweng.education***. GatorRent is developed and hosted on ***sfsuswe.com*** sever, which consists of LAMP stack, MySQL database, LINUX shell accounts and MINI php frameworks. The other server, ***sweng.education*** has GitLab accounts used as a cross-communication platform for code deployment and QA testing.

The front-end development uses LAMP stack tool and Bootstrap framework. So, what is LAMP stack? **LAMP stack** tool is a popular open source web platform commonly used to run dynamic web sites and servers. It is very flexible choice for developing web applications which need high performance and reliability. It is Linux, Apache, MySQL, PHP, Python, Perl compatible platform.

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Every website we find on internet needs to be developed has a foundation of skeleton structure. Likewise, the skeleton structure for our website is MINI. **MINI** is an extremely simple and easy to understand skeleton PHP application. Why MINI? Well, the straight-forward answer would be MINI is easy to install, runs nearly everywhere and doesn't make things more complicated than necessary.

In this project, the back-end development uses PHP as the scripting language. What is PHP? Why do we use it? Firstly **PHP**, *Hypertext Preprocessor*, is a server-side scripting language basically designed for web development. PHP code is interpreted by a web server via a PHP processor module, which generates the resulting web page. For the later question, the answer is

that PHP is free and easy to code. PHP, though GNU General Public License (GPL) incompatible, can be embedded into HTML code and can be used in combination with various web template systems, web content management systems and web frameworks..

The data for the website is stored in ***sfsuswe.com*** server. **MySQL**, an Open Source Relational SQL database management system, is hosted on this server. It is one of the best RDBMS being used for developing web-based software applications in terms of usage and practice.

jQuery is another tool which will be used to design GatorRent. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

For the other server, ***sweng.education***, has **GitLab** which is an application to code, test, and deploy code together. It provides Git repository management with fine grained access controls, code reviews, issue tracking, activity feeds, wikis, and continuous integration. GitLab consists of Repository, a file storage manager, and Master which is a branch that contains the entire code of the website.

During the last 2 weeks of semester, the website will be hosted on **Amazon Cloud**, a cloud storage application managed by Amazon. The service offers secure cloud storage, file backup, file sharing, and Photo printing. Using an Amazon account, the files and folders can be transferred and managed from multiple devices including web browsers, desktop applications, mobiles, and tablets.

8. TEAM

1. Soumithri Chilakamarri : Team Lead and front-end developer for GatorRent
2. Matthew Wishoff : Tech Lead and Back-end developer for GatorRent
3. Kevin Fang : Front-end developer of GatorRent
4. Guanming Pan : Front-end developer of GatorRent
5. Jeffrey Ilar : Back-end developer of GatorRent
6. Emil Santos : Back-end developer of GatorRent

SW Engineering CSC648/848 Section 01 Fall2016

Online Apartment Rental Site Tailored to SFSU Students

GatorRent

Team 05

Local

Soumithri Chilakamarri (Team Lead) schilaka@mail.sfsu.edu

Guanming Pan (C.T.O.)

Matthew Wishoff

Kevin Fang

Jeffrey Ilar

Emil Santos

"Milestone 2"

10-26-2016

History Table

S.No.	Version	Revision Date	Revision Description
1.	1.0.	10-26-2016	First Draft
2.	2.0.	11-03-2016	1) Changed UI mockups as per feedback 2) Added AWS component in UML class diagram 3) Modified Technical skills as per feedback

1. USE CASES

1.1. Student:

Tony is a 20 year old **student** at San Francisco State looking to find an **apartment**.With **GatorRent** he is able to narrow his search to apartments with two bedrooms. Tony also filters apartments by price and pets allowed as well. In just one hour of using GatorRent Tony finds two suitable apartments. Tony then proceeds to register on the site in order to **message** both posters.

Sandra uses **GatorRent** to search for leaseholders specifically subletting a single **room.** By using the map feature of **GatorRent** Sandra is able to get a nice idea of the relative distance between the listings and school. Within a day Sandra is able to find several apartments in Park Merced subletting rooms.

1.2. Leaseholder:

Dorothy is a 57 year old **leaseholder** to a **house** near San Francisco State and she would like to advertise it for rent. She does not like long registration processes on websites. Dorothy wants to post **pictures** of the apartment and also set the price she is renting for. She also wants renters to know the house includes its own washing machine and dryer. Lastly she wants to tell renters she is willing to negotiate the price as a **special detail**. **GatorRent** is able to meet all of her needs.Before making a post she is prompted through the quick registration process, she then uploads her pictures, sets the desired price, and indicates the house has a washer and dryer. Before finalizing the posts she writes that the **price** is negotiable. Within three weeks she meets five potential rentees and is able to sublet the house the next month.

1.3. Roommate Finder:

Henry is a **student** at San Francisco State and is looking to replace his roommate. With **GatorRent** he is able to indicate he is looking for a non-smoker who does not drink. Henry also lists specific details of the apartment such as size, number of roommates, the total rental price

that is paid by the roommates each month. Within a week, Henry receives messages from 14 interested **students**.

1.4. Administrator:

Miranda is one of the hired **administrators** of GatorRent. Her job is to review **flagged** postings and deletes them if they violate the site's guidelines. She is also in charge of support and as such she can change user passwords and respond to any user messages. On rare occasions she also deletes user accounts responsible for repeatedly posting offensive content.

2. DATA DEFINITION

2.1. User : Any person who uses GatorRent is a User.

2.1.1. Student : A User who is studying in SFSU is a Student. A Student uses GatorRent for finding the desired room. They may also use GatorRent to find a roommate.

2.1.2. Leaseholder : A User who would like to advertise a space for renting in GatorRent is a Leaseholder.

2.1.3. Administrator : A User who has control over all GatorRent data.

2.1.4. Guest : A User who has not signed into an account. Guests are able to view Listings but can not create or respond to them.

2.2. Account : An Account is the user information in GatorRent. An Account is created when a user signs up in GatorRent. Only users with an account may create or respond to listings. Accounts are created when Users Sign Up.

2.2.1. First Name

2.2.2. Last Name

2.2.3. Email

2.2.4. Password

2..4: Address: The location of the Listing. For Roommate Finder, the address is the location of where the poster lives.

2.4.1. Address

2.4.2. District

2.4.3. City

2.4.4. State

2.4.4. Country

2.4.4. Zipcode

2.5. Listing: Homes open for Students to rent. There are three types of housing available;

single room, apartment, and an entire house.
2.5.1 Number of Bedrooms **2.5.2 Number of Bathrooms** **2.5.3 Floor Size** **2.5.4 Rent** . **2.5.5 Lease Time** **2.5.6 Number of Tenants** **2.5.7 Relative Distance** **2.5.8 Date** **2.5.9 Post Date** **2.5.10 Relative Distance** **2.5.11 Leaseholder ID** **2.5.12 Picture ID** **2.5.13 Adress ID**
2.6. Roommate: A person who will occupy the same housing as another.
2.4.1 Graduation Year **2.4.2 Major** **2.4.3 Minor**
2.7 Picture: A Picture is an image of the housing for rent.
2.4.1 Picture ID **2.4.2 Image File**
2.8 Special Detail: A detail about a Listing written by the Listing creator.
2.9 Flag: A listing that is marked for Administrative review.
2.10 Roommate Finder: GatorRent feature that allows students to make postings to specifically look for roomates

3. LIST OF FUNCTIONAL SPECIFICATIONS

Section	Functional Specification	Priority
3.1.0	The **Student** shall be able to search the website for rental listings	1
3.1.1	The **Student** shall be able to filter listings by marking box settings that fit their needs	1
3.1.1	The **Student** shall be able to create an account	1
3.1.2	The **Student** shall be able to login to an account	1
3.1.3	The **Student** shall be able to contact a Leaseholder or other Students	1
3.1.4	The **Student** shall be able to view pictures	1
3.1.5	The **Leaseholder** shall be able to create an account	1
3.1.6	The **Leaseholder** shall be able to update their posting	1
3.1.7	The **Leaseholder** shall be able to login to their account	1
3.1.8	The **Leaseholder** shall be able to list posting(rentals)	1
3.1.9	The **Leaseholder** shall be able to post pictures of their rental	1
3.1.10	The **Leaseholder** shall be able to specifically lease the room to SFSU students	1
3.1.11	The **Administrator**s shall be able to log into an administrative account through MySQL Workbench	1
3.1.12	The **Administrator**s shall be able to delete posts	1
3.1.13	The **Administrator**s shall be able to view an admin panel	1
3.1.14	The **Administrator**s shall be able to remove/ban users	1

3.1.15	All **Users** shall be able to retrieve their password if it is forgotten	1
3.1.16	The **Leaseholder** shall be able to mention special details	1
3.2.0	The **Student** shall be able to find the approximate location of renting apartments using Google Maps while preserving privacy of landlords	2
3.2.1	The **Student** shall be able to create postings looking for roommates with Roommate Finder	2
3.3.0	The **Student** shall be able to report false claims	3
3.3.1	The **Student** shall be able to favorite certain rentals	3
3.3.2	The **Leaseholder** shall be able to post videos of their rental	3
3.3.3	The **Administrator**s shall be able to change passwords	3
3.3.4	The **Administrator**s shall be able to communicate with Students/Leaseholders	3

4. LIST OF NON FUNCTIONAL SPECIFICATIONS

4.1 Security

4.1.1. Application shall be hosted and deployed on Amazon Web Services as specified in the class

4.1.2. Data shall be stored in the MySQL database on the class server in the team's account

4.1.3. Application shall be served from the team's account

4.1.4. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

4.1.5. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.

4.2. Performance

4.2.1. No more than 50 concurrent users shall be accessing the application at any time

4.2.2.Google analytics shall be added for major site functions.

4.2.3. Site security: basic best practices shall be applied (as covered in the class)

4.3. Optimization

4.3.1. Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. It shall degrade nicely for different sized windows using class approved programming technology and frameworks so it can be adequately rendered on mobile devices

4.3.2. The language used shall be English.

4.3.3. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.

4.3.4. Pay functionality (how to pay for goods and services) shall be simulated with proper UI, no backend.

4.4 Best Practices

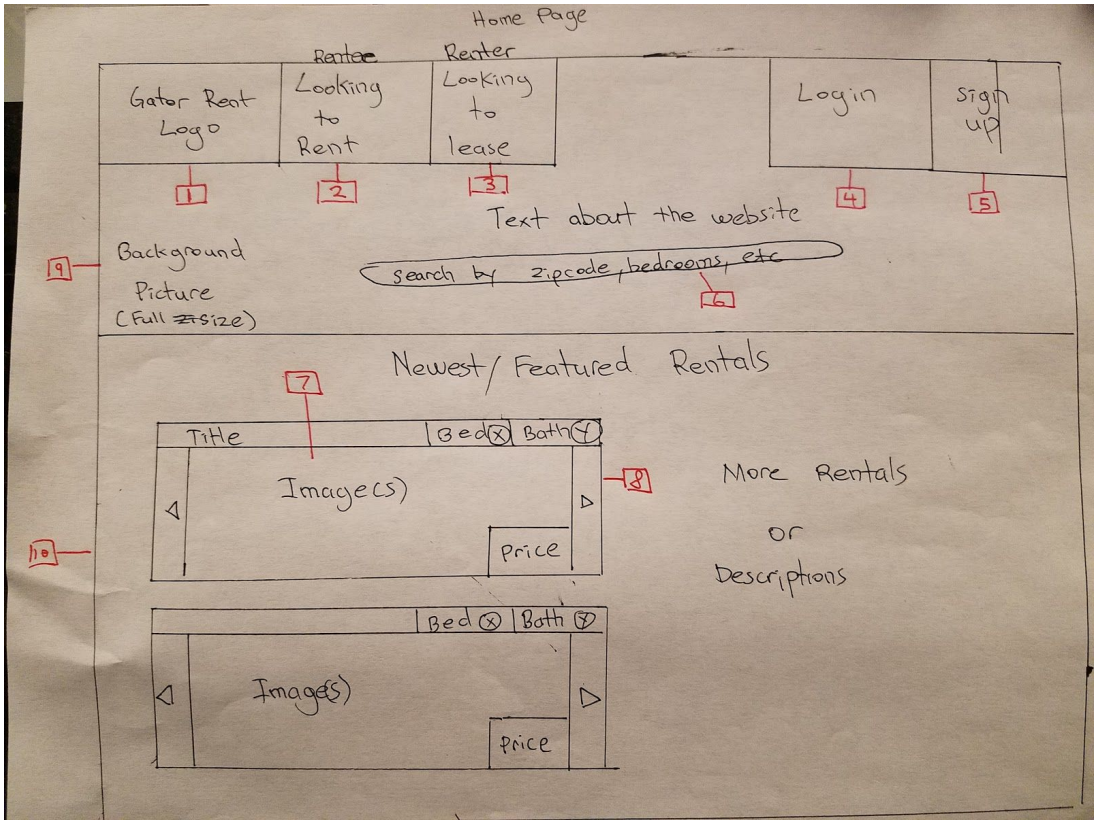
4.4.1. Application shall be developed using class provided LAMP stack

4.4.2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class.

4.4.3. The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Fall 2016. For Demonstration Only". (Important so as to not confuse this with a real application)

5. UI Mockups and Storyboards

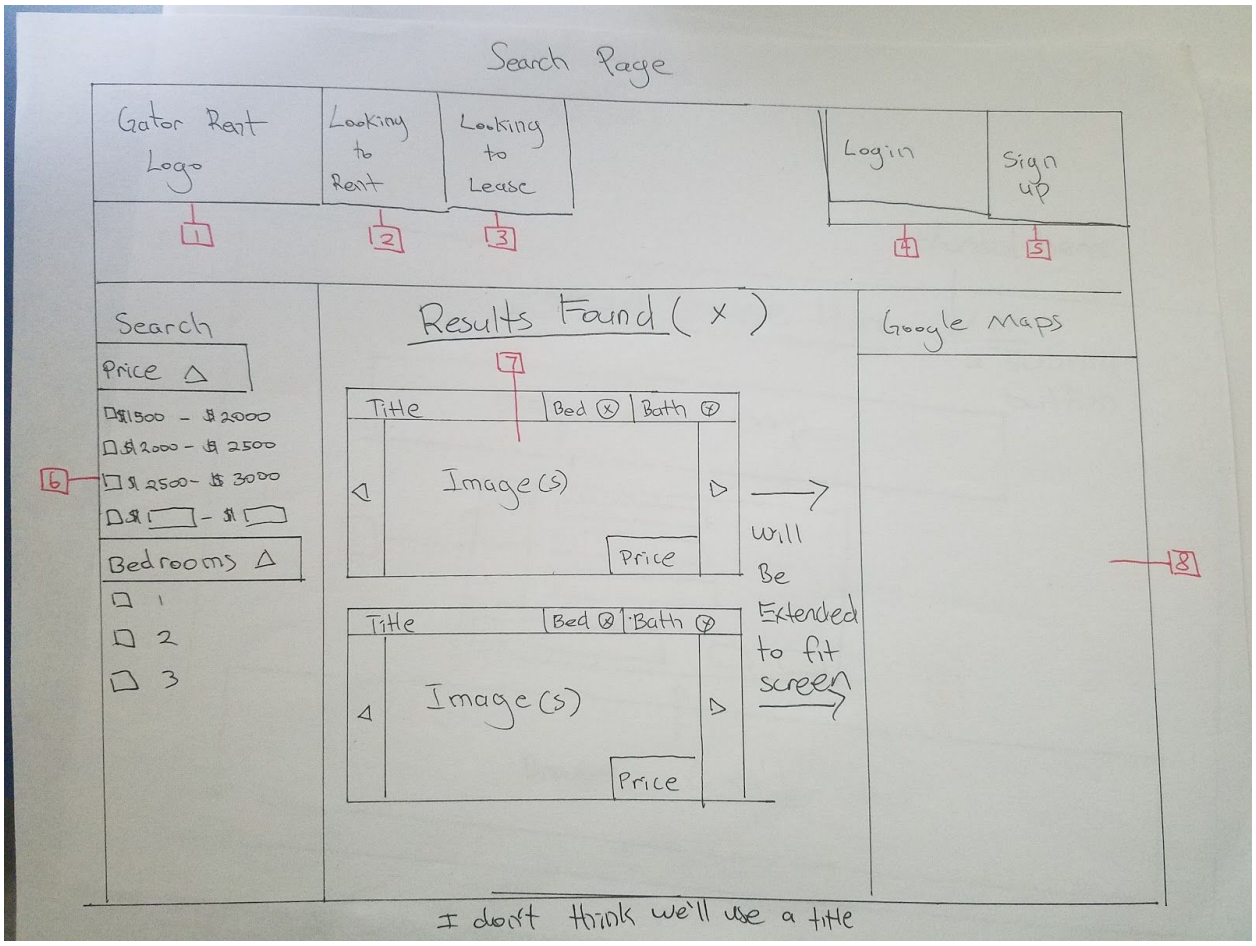
Home Page



Note: Tabs are part of a Navigation Bar

1. Home button titled Gator Rent (with a logo)
2. Links to a page where you can search for rentals
3. Links to a page where you can create a listing
4. Opens a login window
5. Opens a sign up window
6. Search bar
7. Clickable image of a posting
8. Navigate through various photos
9. Initial landing page
10. List of newest rentals

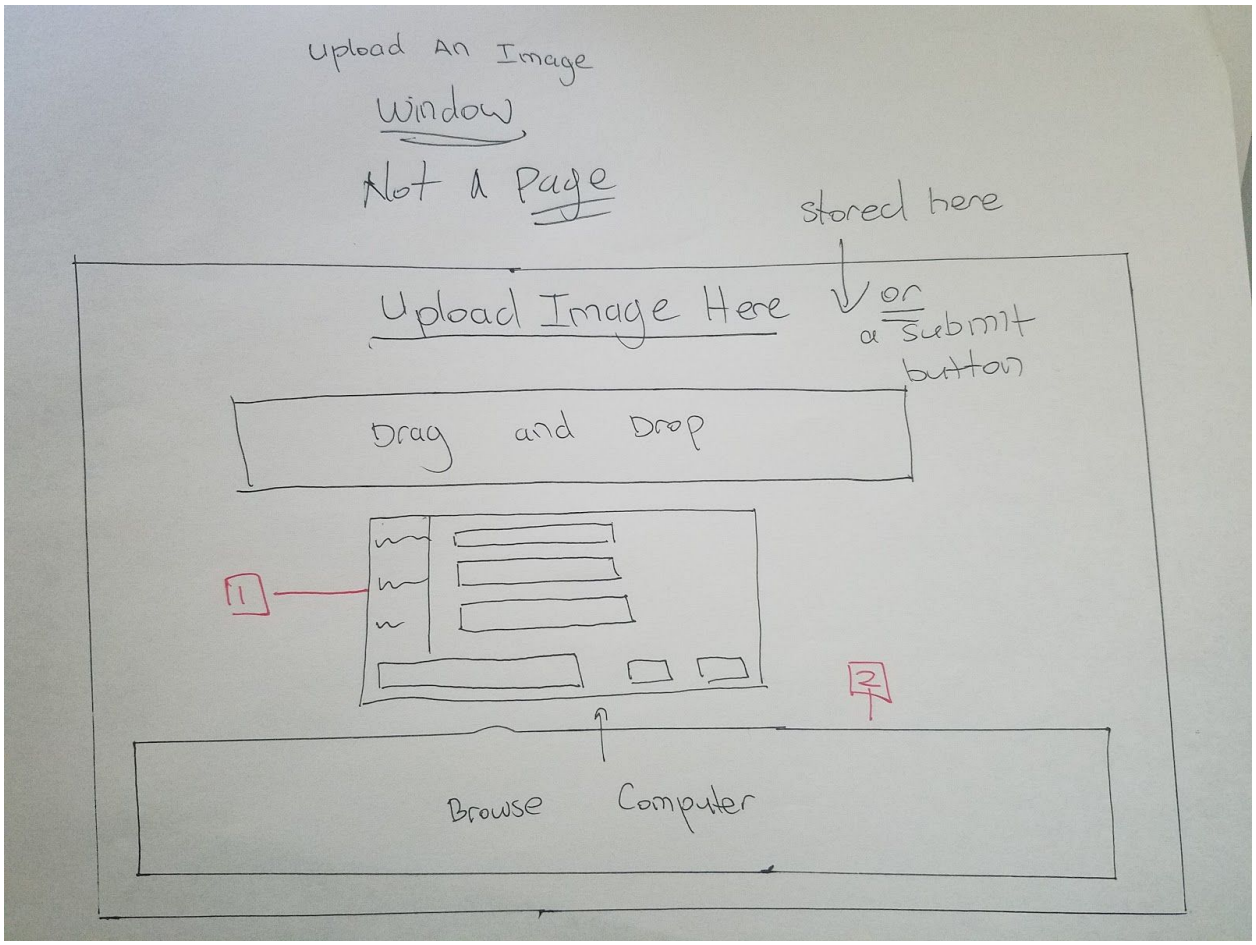
Search Results Page



Note: Tabs are part of a Navigation Bar

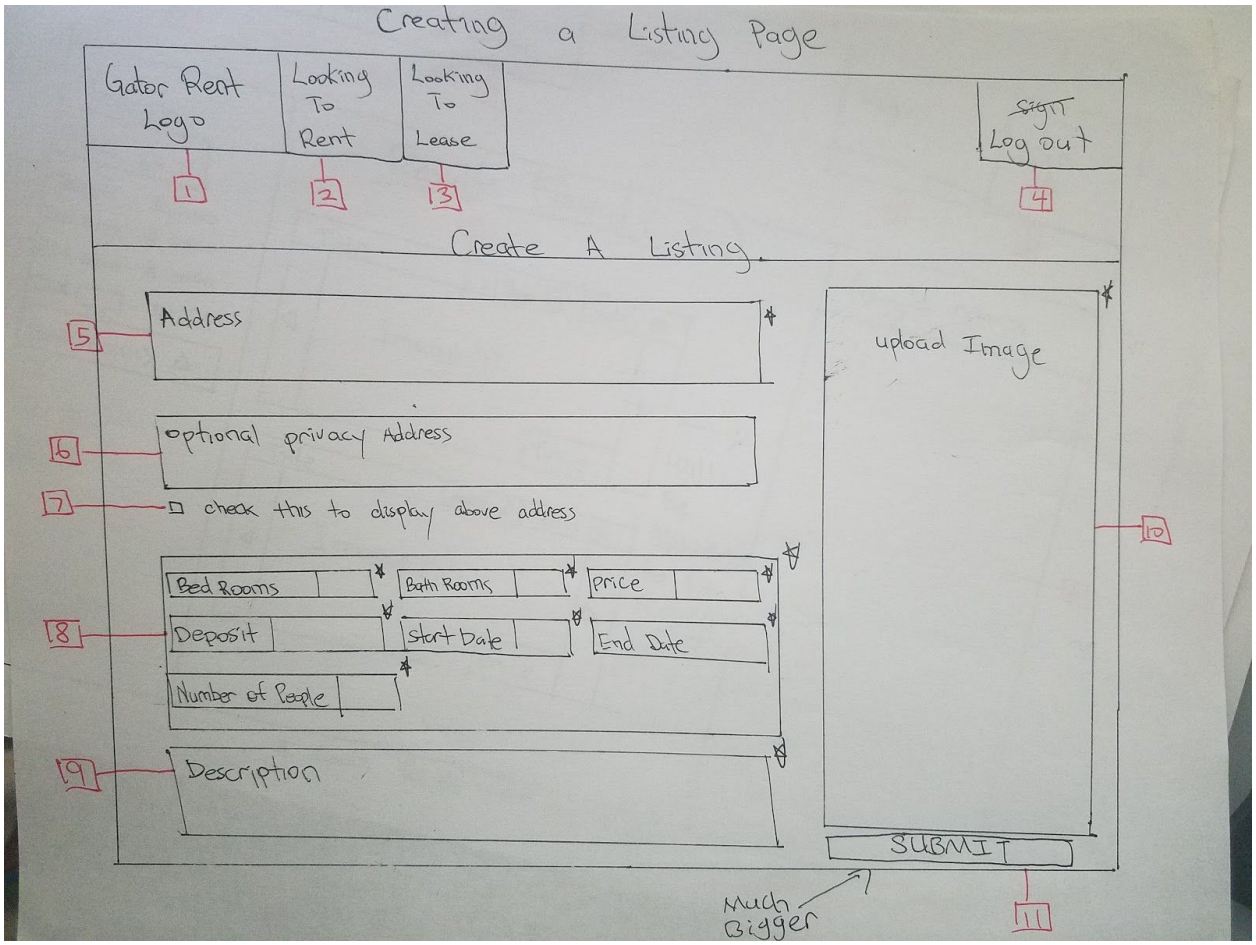
1. Home button titled Gator Rent (with a logo)
2. Links to a page where you can search for rentals
3. Links to a page where you can create a listing
4. Opens a login window
5. Opens a sign up window
6. Filters that allow the user to adjust their search
7. Sample information of a rental
8. Location of where the rental is on google maps.

Upload Page



1. Upload image by searching through computer folder system.
2. Upload image by dragging it into the box.

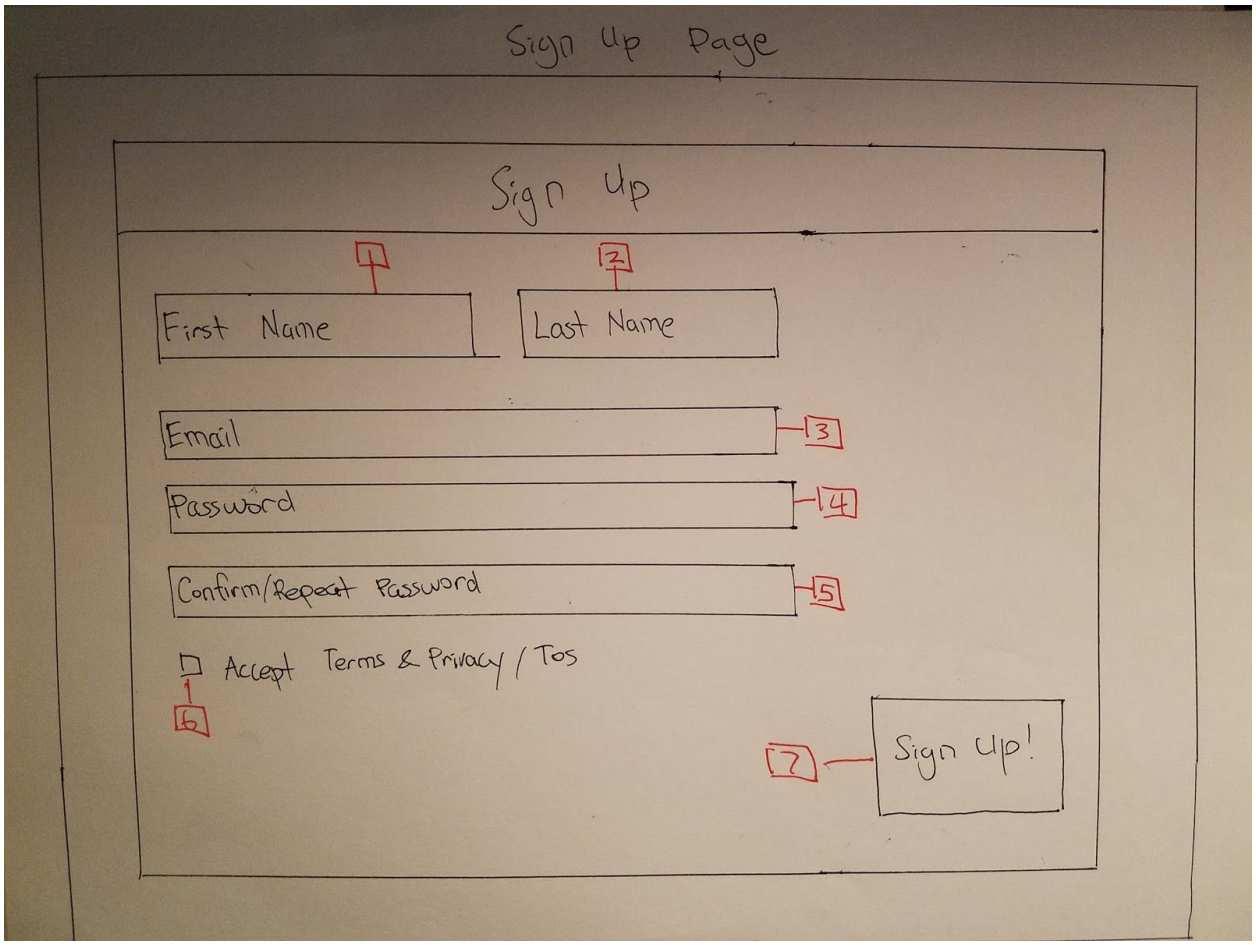
Create a Listing



Note: Tabs are part of a Navigation Bar

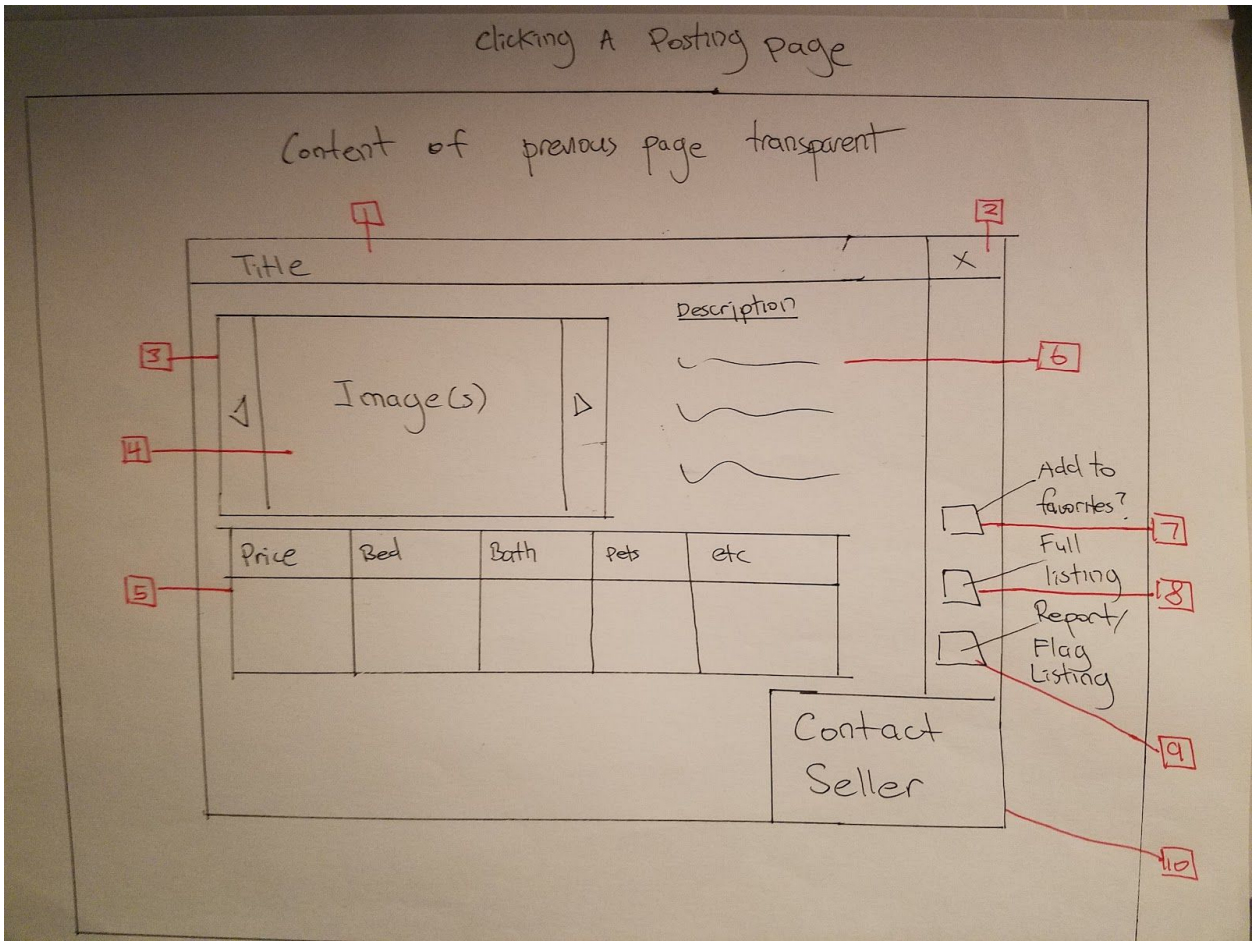
1. Home button titled Gator Rent (with a logo).
2. Links to a page where you can search for rentals.
3. Links to a page where you can create a listing.
4. Logs user out of their account.
5. Prompts user for address.
6. Prompts user for an alternative address they'd like to use..
7. Button that allows user to use their alternative address information.
8. Prompts user for basic house information.
9. Prompts user for description of the house.
10. Upload an image or images of the house.
11. Button to submit the listing.

Sign up Page



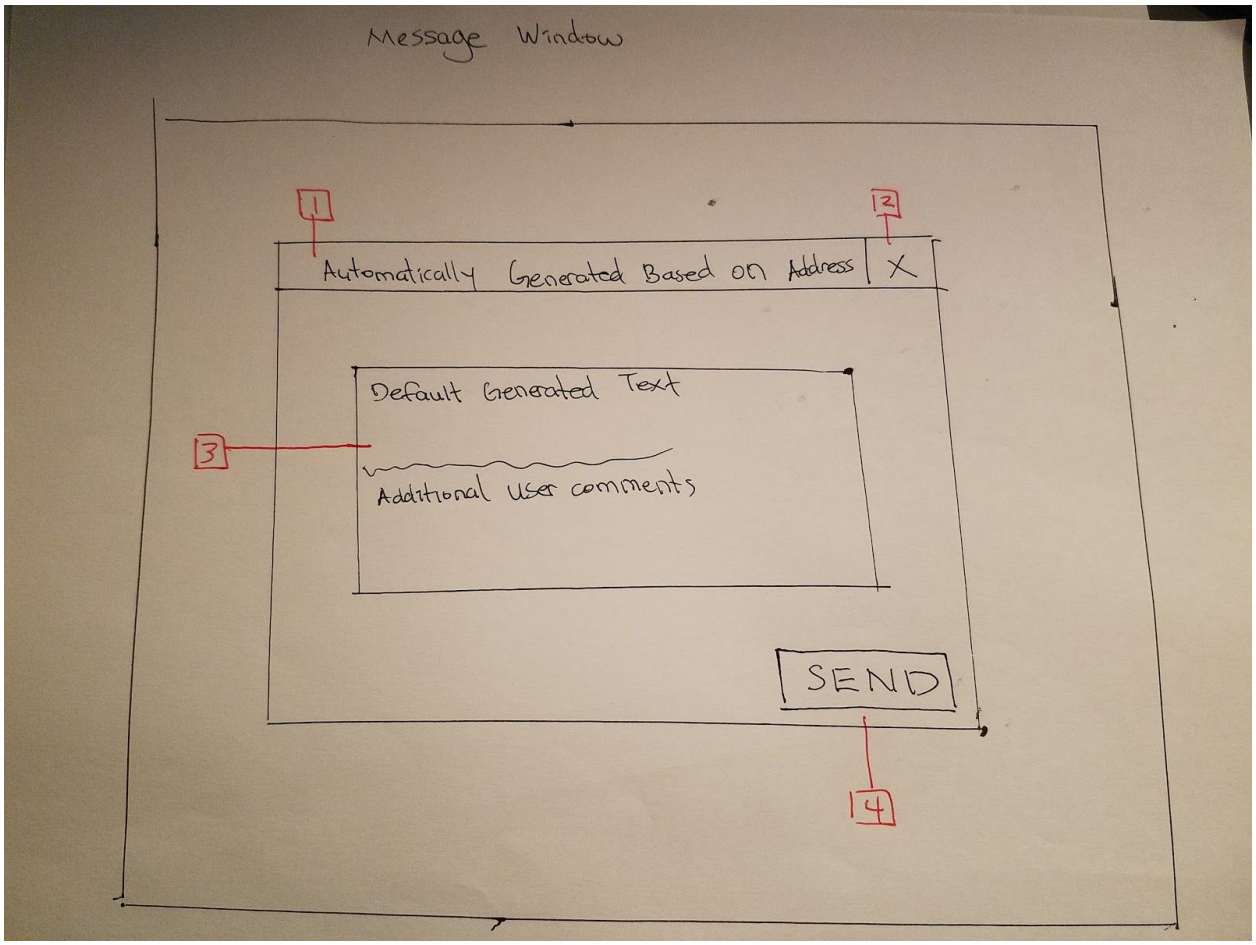
1. Prompts user for first name.
2. Prompts user for last name.
3. Prompts user for email.
4. Prompts user for password
5. Prompts user to confirm password
6. Forces users to accept TOS and Privacy agreements
7. Sign up button/Complete.

Clicking a Posting window



1. Title of posting.
2. Exit window and return to your original page.
3. Arrows to move slider.
4. Clickable images.
5. Specific details about the rental.
6. Description of the rental.
7. Button to add listing to favorites/remembered
8. Button to see the full listing
9. Button to flag/report listing
10. Contact the seller

Message window



1. Automatically generated title based on address
2. Exit the window
3. Automatically generated initial text, user may enter more text.
4. Button to send message.

6. HIGH-LEVEL SYSTEM ARCHITECTURE, DATABASE ORGANIZATION

The System architecture for **GatorRent** uses two servers - ***sfsuswe.com*** and ***sweng.education***. GatorRent is developed and hosted on ***sfsuswe.com*** sever, which consists of LAMP stack, MySQL database, LINUX shell accounts and MINI php frameworks. The other server, ***sweng.education*** has GitLab accounts used as a cross-communication platform for code deployment and QA testing.

The front-end development uses LAMP stack tool and Bootstrap framework. So, what is LAMP stack? **LAMP stack** tool is a popular open source web platform commonly used to run dynamic web sites and servers. It is very flexible choice for developing web applications which need high performance and reliability. It is Linux, Apache, MySQL, PHP, Python, Perl compatible platform.

Bootstrap is a free and open-source front-end web framework for designing websites and web applications. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many web frameworks, it concerns itself with front-end development only.

Every website we find on internet needs to be developed has a foundation of skeleton structure. Likewise, the skeleton structure for our website is MINI. **MINI** is an extremely simple and easy to understand skeleton PHP application. Why MINI? Well, the straight-forward answer would be MINI is easy to install, runs nearly everywhere and doesn't make things more complicated than necessary.

In this project, the back-end development uses PHP as the scripting language. What is PHP? Why do we use it? Firstly **PHP**, *Hypertext Preprocessor*, is a server-side scripting language basically designed for web development. PHP code is interpreted by a web server via a PHP

processor module, which generates the resulting web page. For the later question, the answer is that PHP is free and easy to code. PHP, though GNU General Public License (GPL) incompatible, can be embedded into HTML code and can be used in combination with various web template systems, web content management systems and web frameworks..

The data for the website is stored in ***sfsuswe.com*** server. **MySQL**, an Open Source Relational SQL database management system, is hosted on this server. It is one of the best RDBMS being used for developing web-based software applications in terms of usage and practice.

HTML, **CSS** and **jQuery** are the scripting languages used at the front-end design of GatorRent. **HTML** (HyperText Markup Language) is the basic building block of most web pages. **CSS** (Cascading Style Sheet) is combined with HTML to allow the developers to style each element individually, or group elements together for a uniform look. **jQuery** is another tool which will be used to design GatorRent. jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.

The compatible browsers for GatorRent are:

- Mozilla Version: 44.0.2 and 45.0
- Chrome Version: 49.0.2623 and 48.0.2564

For the other server, ***sweng.education***, has **GitLab** which is an application to code, test, and deploy code together. It provides Git repository management with fine grained access controls, code reviews, issue tracking, activity feeds, wikis, and continuous integration. GitLab consists of Repository, a file storage manager, and Master which is a branch that contains the entire code of the website.

During the last 2 weeks of semester, the website will be hosted on **Amazon Cloud**, a cloud storage application managed by Amazon. The service offers secure cloud storage, file backup, file sharing, and Photo printing. Using an Amazon account, the files and folders can be

transferred and managed from multiple devices including web browsers, desktop applications, mobiles, and tablets. The Amazon server specifications is listed below:
Hardware 64 bit Intel(R) Xeon(R) CPU E52651 v2 @ 1.80GHz, 4GB RAM.

APIs

Google Maps API : The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface.

Google Analytics API : The Google Analytics Embed API is a JavaScript library that allows you to easily create and embed a dashboard on a third-party website in a matter of minutes. It gives you a set of pluggable components that can work together to build complex tools, making it both simple and powerful at the same time.

Data Organization:

Image Format/Size: Format GIF, JPG and PNG will be the accepted image formats for our website. The resolution of the image file is 1600 x 900 ppi and the size is 317.42 KB.

Thumbnail Format: Thumbnails will be saved as images of the same format as the original in a maximum of 144 pixels in the longest direction.

Metadata:

Image information regarding the filename, the size, the resolution, the upload date, the thumbnail path, the filename of the thumbnail will be stored in the file system. The images are stored in file systems not in blobs. An advantage of file system approach is the efficiency and performance of data retrieval from the database.

Related Data:

The data of the users will be stored in database tables. Whenever a user signs up, new user data record is generated and stored. The User_id of a user is unique like the email address and password. For the user who are students, we store additional information such as grad_year, minor and minor.

Search Architecture:

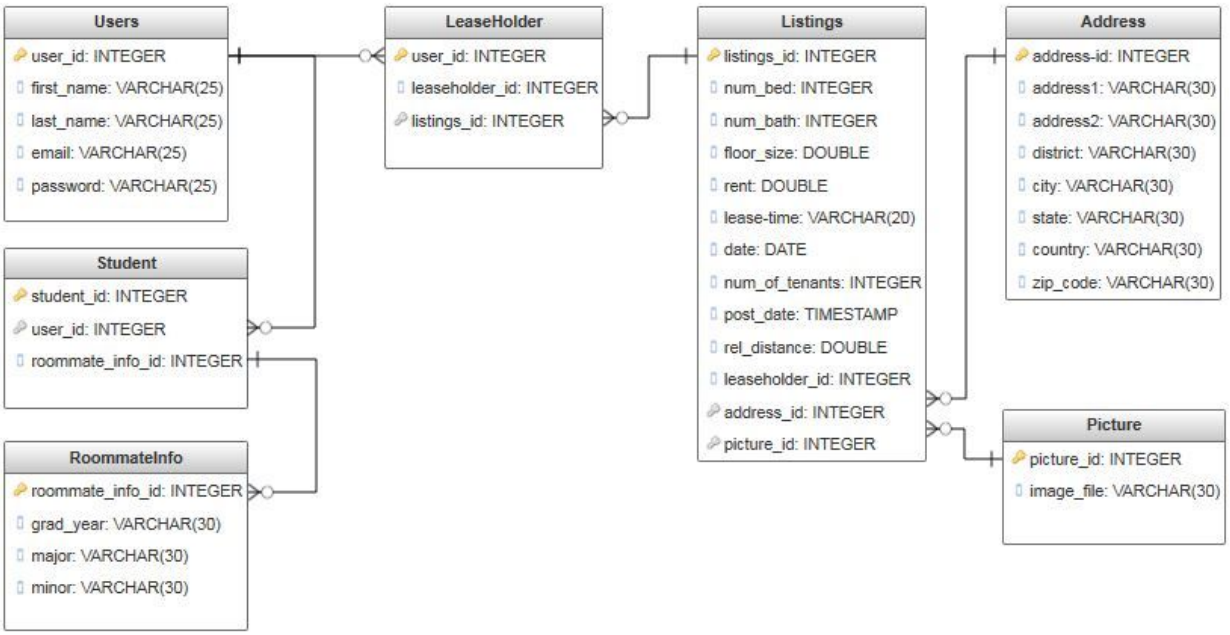
GatorRent uses %LIKE search architecture. Using the LIKE keyword, we pass in the percentage sign (%), which is a wildcard character that returns zero or more characters and our name variable from the search field. As a result, the LIKE keyword (in conjunction with our wildcard character) will find any name that matches in our database table.

For example,

```
$sql="SELECT user_id, first_name, last_name FROM Users WHERE  first_name LIKE '%" . $first_name. "%' OR last_name LIKE '%" . $last_name  ."%'";
```

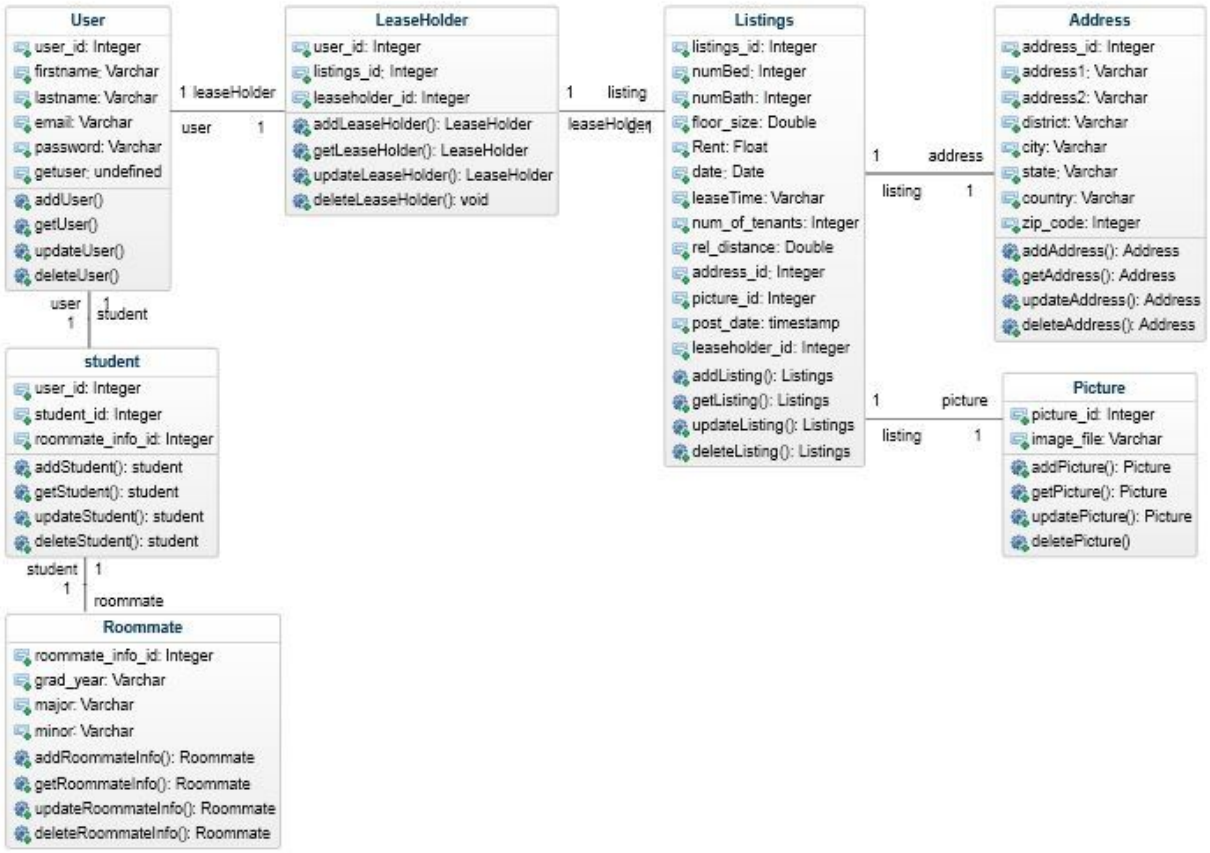
The above query retrieves the list of all the user data from database when either first name or last name of the user. Similarly, GatorRent does a %LIKE search when a user enters the area (apartments to be found in that area) in the search bar. Say that the user enters ‘park’ in the search bar, GatorRent displays all the listings present in ‘park ’ area.

Database Organization

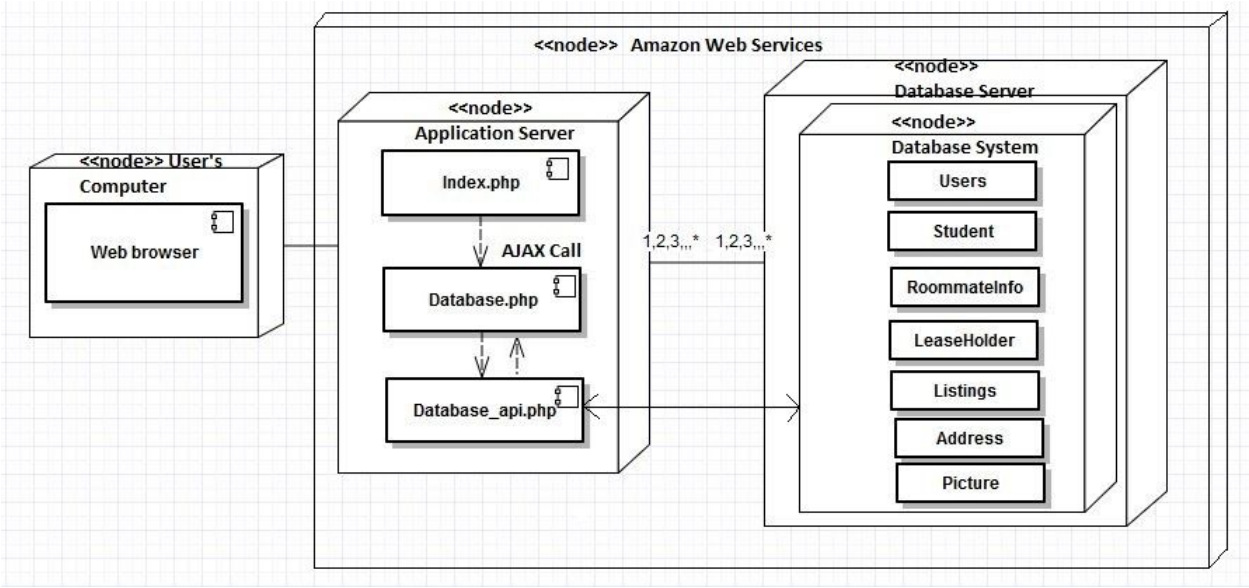


7. UML Class Diagram

7.1. High Level UML Class Diagram:



7.2 UML Component and Deployment Diagram



8. High Level APIs

User

addUser() : adds a new user account to the database.

getUser() : fetches the details of user from the database

updateUser() : edits the attributes of user to the database

deleteUser() : deletes the user account from the database

Student

addStudent() : adds a new student account to the database

getStudent() : retrieves the student account from the database

updateStudent() : updates/edits the student account in the database

deleteStudent() : deletes the student account from the database

Roommate

addRoommateInfo() : adds a new roommate information (like grad year) to the database

getRoommateInfo() : retrieves the roommate information account from the database

updateRoommateInfo() : updates/edits the roommate information account in the database

deleteRoommateInfo() : deletes the roommate information account from the database

LeaseHolder

addLeaseHolder() : adds a new LeaseHolder account to the database

getLeaseHolderr() : retrieves the LeaseHolder account from the database

updateLeaseHolder() : updates/edits the LeaseHolder account in the database

deleteLeaseHolder() : deletes the LeaseHolder account from the database

Listings

addListing() : adds new Listings (like Rent) to the database

getListing() : retrieves the Listings from the database

updateListing() : updates/edits the Listings in the database

deleteListing() : deletes the Listings from the database

Address

addAddress() : adds a new address information (like city) to the database

getAddress() : retrieves the address information from the database

updateAddress() : updates/edits the address information in the database

deleteAddress() : deletes the address information from the database

Picture

addPicture() : adds the new picture information (like name) to the database

getPicture() : retrieves the picture information from the database

updatePicture() : updates/edits the picture information in the database

deletePicture() : deletes the picture information from the database

9. Key Risks

Skill Risks:

PHP

PHP is new to majority of our team since no one worked on it before. Learning PHP is one of the vital tasks for both, the frontend and the backend, since many website developers today use the language to increase their site's performance. In order to bridge this gap, our team worked together to complete PHP course in codeacademy.com. In addition, we also used the online resources such as w3schools.com, stackoverflow.com to clarify few questions we could not understand.

MySQL Workbench 6.3

All of our team members have worked with MySQL using the Command shell or the UNIX shell but none of them have worked MySQl using the Workbench. Designing the database schema is a major task and few of our members took this task of using Workbench to design databases while guiding the inexperienced members.

GitLab

Only few of our team members have prior experience using the GitLab software. Even though the git commands are similar to GitHub, a lot of care must be taken while pushing the code to the repository. Extreme care should be taken while pushing and committing to a branch since any small code conflict in master will cause the whole website to crash. We have organized a small session to learn GitLab and its functionality so that our team members can communicate within ourselves while pushing the code into development branch to ensure that the website does not crash.

Schedule Risks:

The time and resources are the two major factors involved in creating the website. Taking time into consideration, our team members have work and other classes and is a potential risk in completing the work. So, we decided to meet every Thursday outside of class for an hour. This meeting is a scrum where we would discuss on the progress of each member and the future tasks we have to complete. If any member needs assistance, all the team members sit together and help to solve the issue. Like this, we are dedicated in finishing all the milestones set forth in the project.

Technical Risks:

Cross-platform compatibility

A technical issue is the Cross-platform compatibility. It's hard to get websites to look and function perfectly on the browsers taking considerations of the orientation and the screen size. When we try to optimize for a browser, it may de-optimize for another. In order to ensure that GatorRent works on the specified browser, our team would make the website to be standard-compliant by making the better use of Bootstrap and keeping the code simple for better traceability.

Teamwork Risks:

An important teamwork risk would be the mutual agreement between the front-end and the back-end members. Although the front-end and back-end work independently, there might be situations where we should work together. Some of the decisions made by the front-end team may not agree with the back-end team. To resolve this, our team split into frontend and backend teams. At first, the backend team designs and organizes the APIs using MVC approach. Then the frontend team can display the data as needed without interfering with the backend. This process is a cyclic process and prevents any mutual risk between the two teams.

Legal/Content Risks:

GatorRent allows the Leaseholders to post their pictures of rental apartments. However, there may be spam users who upload the images which that violate our terms and conditions. To resolve this issue, we provide our users to flag their posts which they find inappropriate. The Admin will check the flagged posts and removes these spam posts and block users that violate our terms and conditions.

10. TEAM

1. Soumithri Chilakamarri : Team Lead and front-end developer for GatorRent
2. Guanming Pan : Tech Lead and front-end developer of GatorRent
3. Matthew Wishoff : Back-end developer for GatorRent
4. Kevin Fang : Front-end developer of GatorRent
5. Jeffrey Ilar : Back-end developer of GatorRent
6. Emil Santos : Back-end developer of GatorRent

CSC 648/848 SFSU Fall 2016 Milestone 3

Checklist for instructors and teams

10/31/16

This checklist is to be used by instructors and by student teams to record feedback from M3 reviewed meetings. Instructors will fill this out during M3 review. Students should take notes then record the summary in this form and save it for final document submission in M5

Date	**Team**	**Number of team members present**
11-28-2016	5	4

Meeting started on time? (yes)

4:30 pm

Required documentation brought to the meeting?

-UI screen shots, one per page hard copy (yes)

-M2 revised document) (yes)

<u>UI prototype (assessment, comments)</u>

-Test main use cases

-Observe layout, flow, clarity, functionality vs. use cases

-Mark comments on hard copy, team to record comments

Home Page:

1) In the navigation bar, call listing as “My Listing”
2) Make the GatorRent in the navigation bar clickable to make it go to home page.
3) Clickable rent button and contact button on apartments
4) No filter for home page
5) After successful search, say the number of results found for the search result.
6) Prompt a message for invalid search entry in the search bar

Post Apartment Page:

1) Add contact button on top as well and repeat the same information there as well
2) Make the picture and the map smaller

Edit Profile page:

1) As of priority 1, remove this page and make it as priority 2

Listing Page:

1) Add view message button in the Listing Page

Vertical product prototype (assessment, comments) – architecture, PHP OO style, coding style, SVN usage and comments, DB schema, adherence to secure best practices (PW encrypted, inputs validated, HTTPS used) etc.

- Runs from deployment server
- Code documented (header, in code) with good coding style
- MVC//MINI pattern followed up

Comments:

1) Do code reviews
2) Be specific while documenting the code

***Teamwork*: is the team working out, any related issues**

Yes. The team is working hard to manage the skill gaps required to develop the website.

***Risks*: all <u>actual</u>**

– list them and develop strategy to mitigate them

<u>Skill Risks:</u>

Since our team do not have experience in designing the website, we had to learn a lot of tools and languages to bridge this gap. Our tech lead is very instrumental in managing and explaining on how to implement the functions or features required for our website.

***Coding practices and style*:**

-Coding style adopted, code documented (header, in code)

We have adapted the basic MVC approach that came up with the MINI PHP framework which reduces the task of strenuous coding.

Adherence to best practices of security (PW encrypted, inputs verified, use of HTTPS etc.)

1) No HTTPS certificate required since the license has to purchased
2) PW should be encrypted in database

Usage of proper SE code management practices including using only approved tools:

1) Only tools that are approved in class are used – Bootstrap, MySQL Workbench

Usage of gitlab

-Comments on posting

-number of posting

-Appr. even distribution of submissions among team members (check github post stats for all members

Be specific on the messages that we are committing to our code to gitlab repository

Digital content **(e.g. images, video) please provide information on status and availability, chosen formats, resolution etc.**

Copyright images should not be used in GatorRent. The image format should be only jpg, png

<u>Comments and checkpoints (if any)</u>

1) Comments while pushing the code to gitlab repository should be improved.
2) Make sure the password of the users is protected in the Data Base.
3) Move the search bar to the navigation bar (header file).

SW Engineering CSC 648/848 Section 01 Fall 2016

Online Apartment Rental Site Tailored to SFSU Students

GatorRent

Team 05

Local

Soumithri Chilakamarri (Team Lead) schilaka@mail.sfsu.edu

Guanming Pan (C.T.O.)

Matthew Wishoff

Kevin Fang

Jeffrey Ilar

Emil Santos

"Milestone 4"

12-12-2016

History Table

S.No.	Version	Revision Date	Revision Description
1.	1.0.	12-05-2016	First Draft
2.	2.0.	12-10-2016	1) Changed the usability test plan 2) Update the QA test case for content in each apartment page

1. Product Summary

GatorRent

GatorRent is a rental website which allows SF state students to rent apartments near the SFSU campus. The UI is designed to be user-friendly and comes with a powerful search. The home page displays the pictures of apartments that are recently added to the website. Users can search for the apartments entering either pincode, area or the city. The search results show a list of apartments that can be rented. For example if you enter a city name in search bar and press the enter button, it will show the list of apartments (images) as per the city name. The user can then click on the view button to find more information about the apartment. After selecting the view button, it displays the apartment details that has the apartment picture along with the google map location of the apartment. Like this, the user can view the location of the apartment using the google maps.

In order to contact the leaseholder, the user must login to the GatorRent website. In case of a new user, he/she should register to the website by entering the details in the registration form present on the home page. After the user is successfully logged in to the website, he/she can contact the poster by clicking the contact poster button available on apartment details page. A message will be sent to the leaseholder that helps to contact the user.

If you have an apartment to rent, GatorRent provides you to post the apartment on the website so that SFSU students can rent them. To post an apartment at GatorRent, the poster must login to the website and if case of a new user, he/she should register first before posting. Once the poster is logged in, he/she can post the apartment by clicking Post an Apartment button present on top in the GatorRent. The poster can then fill a form regarding the apartment details to be posted. The poster must upload a picture of the apartment in the form if he/she wishes to post on the website.

Product URL :- http://sfsuswe.com/~f16g05/Group5/

S.No.	Functional Specification	Priority
1	The **Student** shall be able to search the website for rental listings	1
2	The **Student** shall be able to create an account	1
3	The **Student** shall be able to login to an account	1
4	The **Student** shall be able to contact a Leaseholder or other Students	1
5	The **Student** shall be able to view pictures	1
6	The **Leaseholder** shall be able to create an account	1
7	The **Leaseholder** shall be able to update their posting	1
8	The **Leaseholder** shall be able to login to their account	1
9	The **Leaseholder** shall be able to list posting(rentals)	1
10	The **Leaseholder** shall be able to post a picture of their rental	1
11	The **Administrator**s shall be able to log into an administrative account through MySQL Workbench	1
12	The **Administrator**s shall be able to delete posts	1
13	The **Administrator**s shall be able to remove/ban users	1
14	The **Leaseholder** shall be able to mention special details	1
15	The **Student/Leaseholder** shall be able to logout to their account after he/she login to their account	1
16	The Student shall be able to contact the Leaseholder	1

17	The Leaseholders shall able to see their listings after they login to their account	1
18	The **Students** shall be able to view the location of an apartment using Google Maps	1
19	The **Leaseholder** shall be able to view the contact information of the **Student** when he contacts the **Leaseholder**	1

Note: Leaseholder refers to the poster and a Student can be a user.

2. Usability Test Plan

2.1. Test Objective

The following test is carried out to check the usability of the GatorRent website's search function.

To test the search feature, the user can shall enter the search term - San Francisco . If the usability test is successful, the user shall be able to view the image of an apartment or images of apartments (if many) posted on the website as per the search criteria. The response time for the search function should be within 5 seconds.

Task	Description
Task	Find an apartment in "San Francisco"
Machine State	The user has accessed the link for GatorRent
Successful completion criteria	Displays the images of apartments in San Francisco
Benchmark	Whole task is completed in 1 minute

2.2. Test Plan

Purpose: To test the usability of GatorRent's search function

Indented User: The user can be anyone who has basic knowledge of how to use a computer/laptop.

System Setup: To test the search function, the user should have a laptop with an internet connection. The laptop should have a browser Mozilla version 44.0.2 /45.0 or Chrome 49.0.2623 /48.0.2564 installed in it.

Starting Point: The starting point of the test plan is the GatorRent's home page. When the user enters the website's URL, it should load the homepage of the GatorRent.

Problem Statement: To do a search for an apartment in the GatorRent homepage. The search term is the city name 'San Francisco'.

Test plan and Objectives: The user shall find an apartment in San Francisco in the GatorRent website. The images of apartments satisfying the search criteria should be displayed on the page.

Completion Criteria: The search results page will be displayed which has the images of the apartments in San Francisco.

URL to be tested: http://sfsuswe.com/~f16g05/Group5/

2.3. Questionnaire form

Please rate the following statements on a scale between 1 and 5

1. I found the GatorRent website easy to use.

(1)Strongly Agree (2)Agree (3)Neither Agree nor disagree (4)Disagree (5)Strongly Disagree

2. I was able to search for my apartment in the GatorRent

(1)Strongly Agree (2)Agree (3)Neither Agree nor disagree (4)Disagree (5)Strongly Disagree

3. The aspects that you liked in GatorRent

_____

_____

_____

4. Any suggestions for improving the website

_____

_____

_____

5. Would you like for more search options?

(1) Yes (2) No

3. QA Test Plan

Test Objectives: The objective of the QA test plan is to check the search functionality of the GatorRent website. The expected result is that the user can view the images of the apartments posted on the GatoRent that matches the given search criteria in the search bar. The security feature and the response time are not tested in this test plan

3.1. Hardware Setup:

The list of required hardware components are:

- Laptop
- internet connection - wired/wireless

Optional components:

- Mouse
- Keyboard
- Monitor

3.2. Software Setup:

Required Browsers:

- Google Chrome - Version: 49.0.2623 or any lower version
- Mozilla Firefox - Version: 45.0 or any lower version

Set the browser's window size to full screen.

3.3. Feature to be tested:

- The search feature is the feature that will be tested.

- The search bar can be found on top of the GatorRent's homepage navigation bar.

Test No.	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS / FAIL)
1.1.	Test for the cursor in the search bar	Check for cursor in the search bar	Move the cursor to the search bar and click the left mouse button on it.	(1) The cursor should be present at the left-most section of search bar. (2) The cursor should be blinking	
1.2.	Test for a single input character in the search bar	Enter a character inside the search bar	Type the character 'S'	(1) The placeholder 'Please Enter a Zip Code or City' should disappear from the search bar. (2) The character 'S' should appear inside the search bar.	
1.3.	Test for deletion of keyword in the text box	Pressing the backspace button	Now, press the backspace button on the keyboard in the search bar	(1) The character 'S' should be disappear (2) The placeholder 'Please Enter a Zip Code or City' should reappear in the search bar.	
1.4.	Test for multiple input characters	Enter multiple characters in the search bar	Type the keyword 'San Francisco' in the search bar	(1)The placeholder 'Please Enter a Zip Code or City' should disappear from the search bar. (2) The keyword 'San Francisco' should appear in the search bar.	
1.5.	Test for Search button in the search bar	Check for cursor in the search bar	After entering the keyword 'San Francisco', move the cursor	(1) The mouse cursor changes to indicate that it is clickable.	
1.6.	Test for Apartments in the Search Results page	Left click the mouse button on the Search button.	Now, left click on the Search button in the search bar. Check for the number of images of apartments in the page.	(1) The page redirectors to the Search Results page. (2) The text - "3 results of San Francisco" is displayed on the screen. (3) A list containing 3 apartment images from	

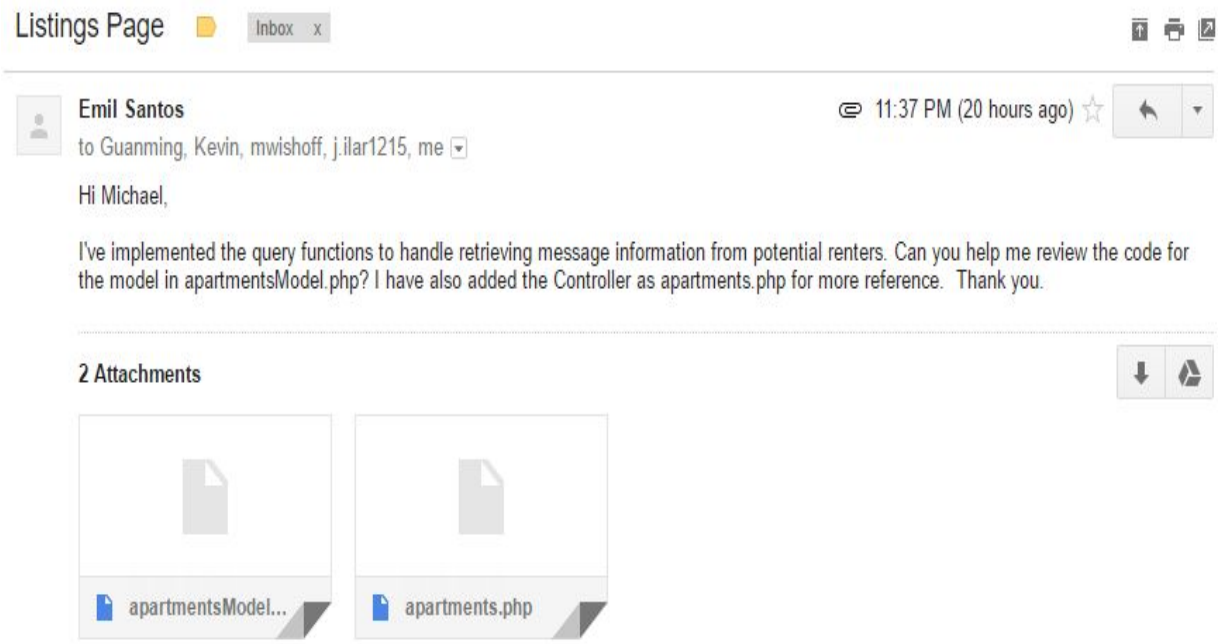
				San Francisco will appear on the search results page.	
1.7.	Test for content in each apartment image	Check the contents for each image in the Search Results Page.		(1) The content for three images should have following descriptions: Title Address City, Zipcode Price Bed Rooms Bath Rooms	
1.8.	Test for Enter button	Check for the functionality of the Enter button	Go back to test number 1.4. After entering the keyword 'San Francisco' in the search bar, press the Enter button on the keyboard.	(1) The page redirectors to the Search Results page. (2) The text - "3 results of San Francisco" is displayed on the screen. (3) A list containing 3 apartment images from San Francisco will appear on the search results page.	
2.1.	Test for the Search Results page	Check the 'Contact button' on the Search Results page	After the search results are displayed, choose an image and click the Contact button present below the image.	(1) The cursor changes to indicate it is clickable before you click it. (2) A window pops at the top and has a message included for contacting the poster.	
2.2.	Test for the Close button	Check the 'Close' button for Contact Poster window	Left click the Close button on the Contact Poster window.	(1) The window should close (2) The search results page should be resumed as before.	
2.3.	Test for the Details button	Check for the 'Details' button on the Search Results page	Move the cursor to the Details button on any image and left click the Details button.	(1) The cursor changes to indicate it is clickable before you click it. (2) The page redirects to the apartment details page. This page should contain the image, details and google map location of the apartment.	

3.1.	Test for home button	Return to the home page from the apartment details image.	On the apartment details image, click the ‘Home’ link on the top left of the screen.	(1) The page is redirected back to the home page.	

4. Code Review

The objective of the code review is to develop a good coding pattern so that every team member can understand and adapt similar styles to make the website look symmetric. In GatorRent, we have used the Model View Controller (MVC) pattern. Our team have defined the controllers say Apartment_Detail_Controller. The controller then calls the Apartment_Detail_Model. Inside our model, we have have defined the functions for Apartment_Detail_Page. The View contains the PHP , HTML and JAVAScript code and the stylesheets for each page are written in the stylesheet folder. We adapted the same coding pattern to all our pages for better functionality of the GatorRent.

The search functionality is written by our tech lead. Our tech lead has put a lot of effort in reviewing all the code part. We practiced the code reviews for all the pages.
The following pictures show the code review for the Listings Page.



Guanming Pan 7:27 PM (23 minutes ago)

to Emil, Kevin, mwishoff, j.ilar1215, me

Hi Emil,

Thank you for hard working. I have reviewed your code and found some issues which I list below.

1. In apartmentsModel, the getUserListings function has used a hard code "1" as a user id.
   THE FUNCTION NEED USER ID AS A PARAMETER TO PASS IN. I ALREADY SENT A WAY HOW TO GET CURRENT USER ID FROM SESSION TO OUR GROUP CHANNEL ON THIS THURSDAY.

2. In apartmentsModel, the getReservationsList function has used $apt_id variable in $sql, another string variable.
   THE $apt_id SHOULD NOT WORK. YOU NEED TO USE :apt_id TO REPLACE IT AND USE $parameters = array(':apt_id' => $apt_id); TO PASS $apt_id VARIABLE.

3. In apartments controller, what is the displayReservationList function?
   I THINK IT IS A USELESS FUNCTION AND WAS PLACED IN WRONG POSITION. REMOVE IT PLEASE.

4. In apartments controller, the listings function missed a parameter.
   THE FUNCTION NEED USER ID AS A PARAMETER TO IDENTIFY WHAT APTS SHOULD BE DISPLAYED.

Please fix above issues and test it. Before you merge code, please let me review it again. Any questions, Let me know. Thanks.

Best Regards,
Guanming

5. Security Practices

In GatorRent, we have implemented the security features for the user to protect their data. The following assets are the assets being protected:

- Login Password
- Poster's Address
- Database password credentials of the User
- Input validation of the user

GatorRent uses the Bootstrap validator to check the input validation when the user registers or logins to the website. For example, the email address format for the user is checked during registration. In case of invalid format, it reports an error and displays the error message asking the user to enter a valid format of his/her email address.

Also when the user logins, GatorRent checks for the valid username and the password. The password are encrypted in the UI as well as the backend to ensure the best security practice for the user. The Administrator functions are implemented in the MySQl WorkBench. Using this tool, the Admin has the ability to delete the postings of the user if the images are found inappropriate or do not agree to the terms of use policy of the website.

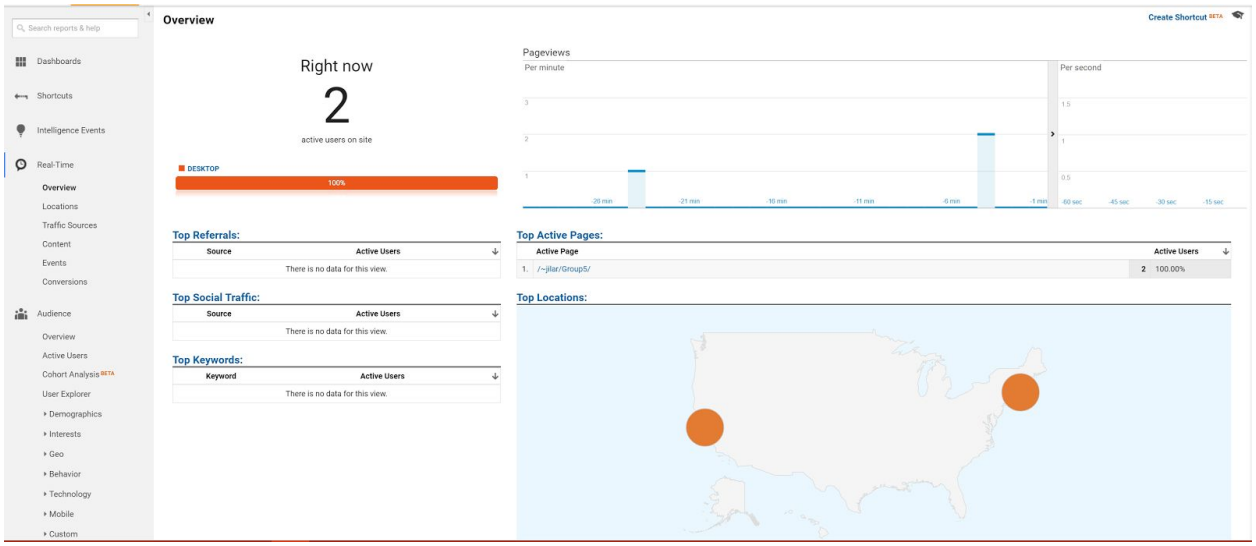
6. Non-Functional Specifications

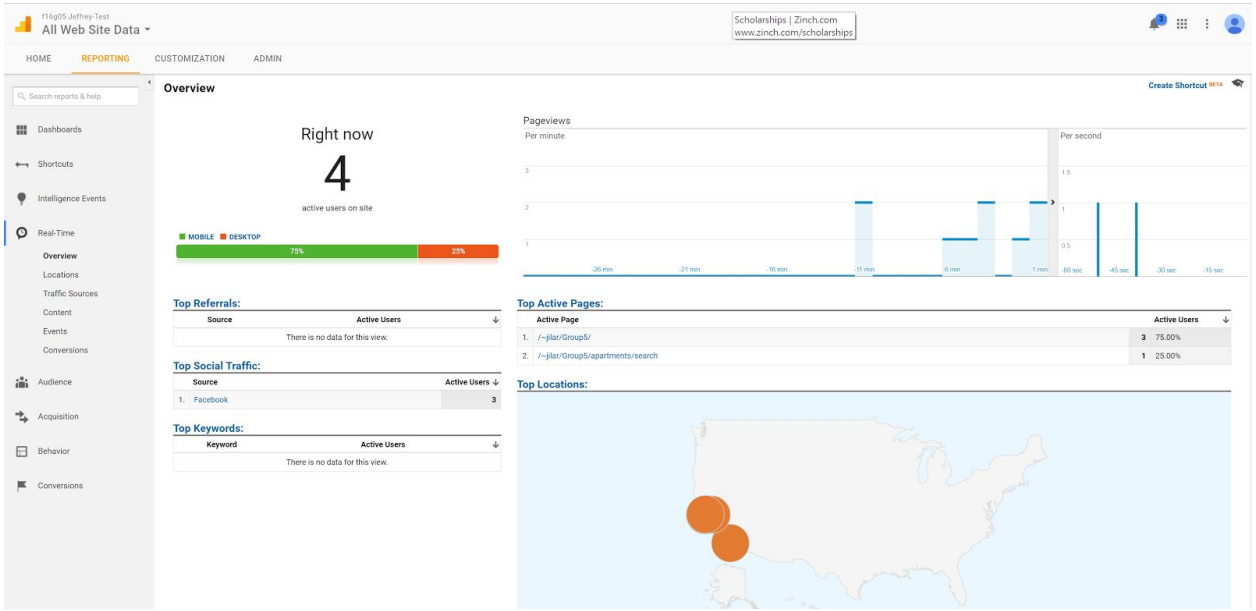
Specification	Status
Application shall be hosted and deployed on Amazon Web Services as specified in the class	Done
Data shall be stored in the MySQL database on the class server in the team's account	Done
Application shall be served from the team's account	Done
Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users	Done
Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.	In progress
No more than 50 concurrent users shall be accessing the application at any time	Done
Google analytics shall be added for major site functions.	Done
Site security: basic best practices shall be applied (as covered in the class)	Issue : No https certificate for the website
Application shall be optimized for standard desktop/laptop browsers, and shall render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome. It shall degrade nicely for different sized windows using class approved programming technology and frameworks so it can be adequately rendered on mobile devices	Done

The language used shall be English.	Done
Application shall be very easy to use and intuitive. No prior training shall be required to use the website.	Done
Application shall be developed using class provided LAMP stack	Done
Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class.	Done
The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Fall 2016. For Demonstration Only". (Important so as to not confuse this with a real application)	In Progress

7. Google Analytics

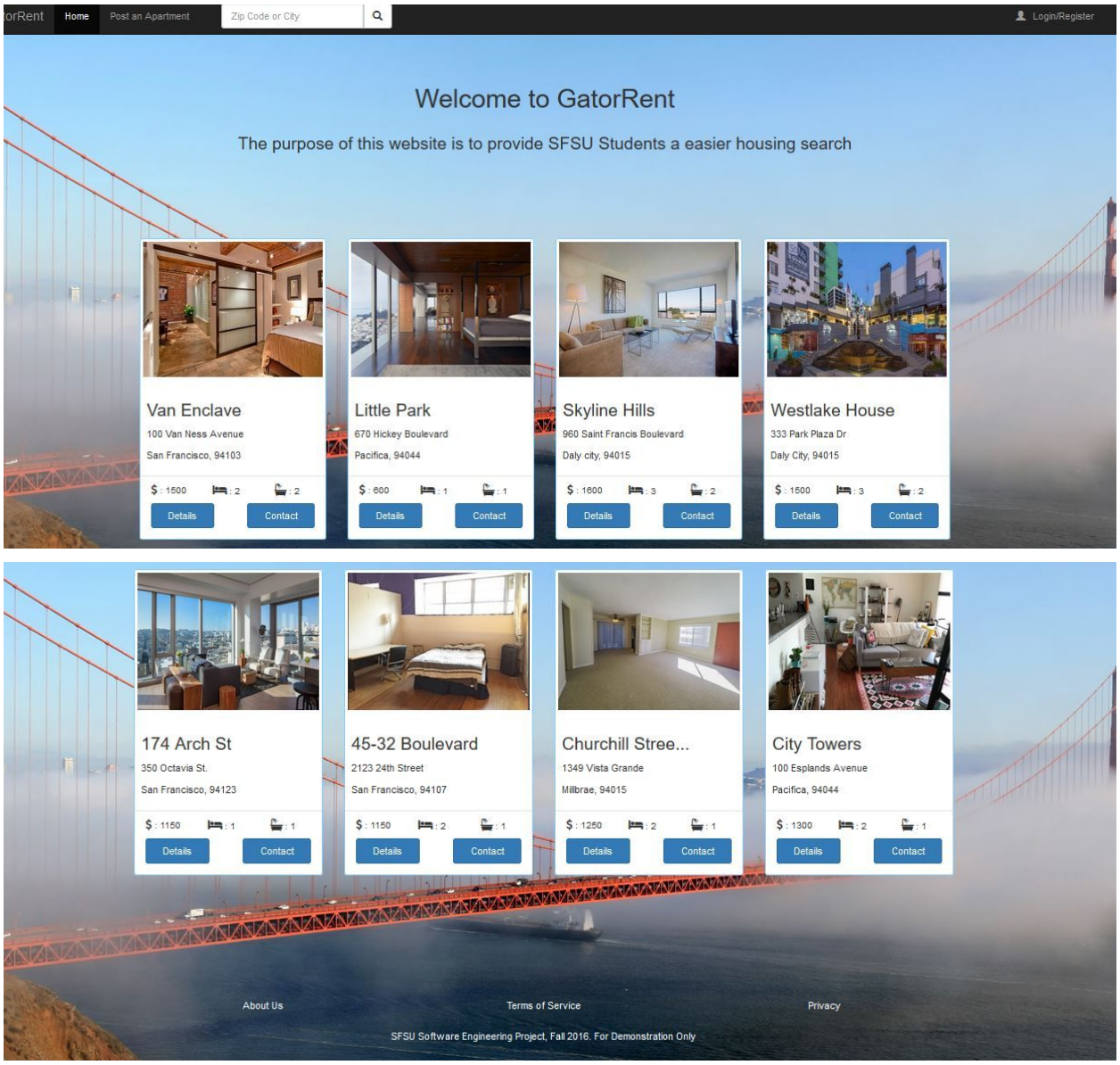
GatorRent uses the google analytics API to get the status of the website. We have implemented this API in our home page to get the status of the home page. Like this, it helps us to improve the website to provide more satisfactory customer experience as we value the users feedback given to us for the GatorRent.



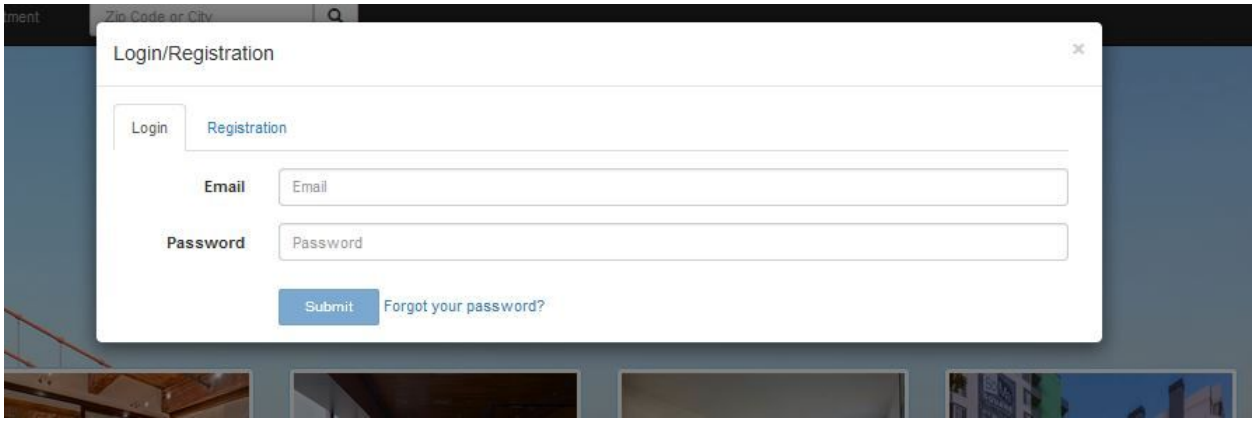


III. Final UI Screen Shots

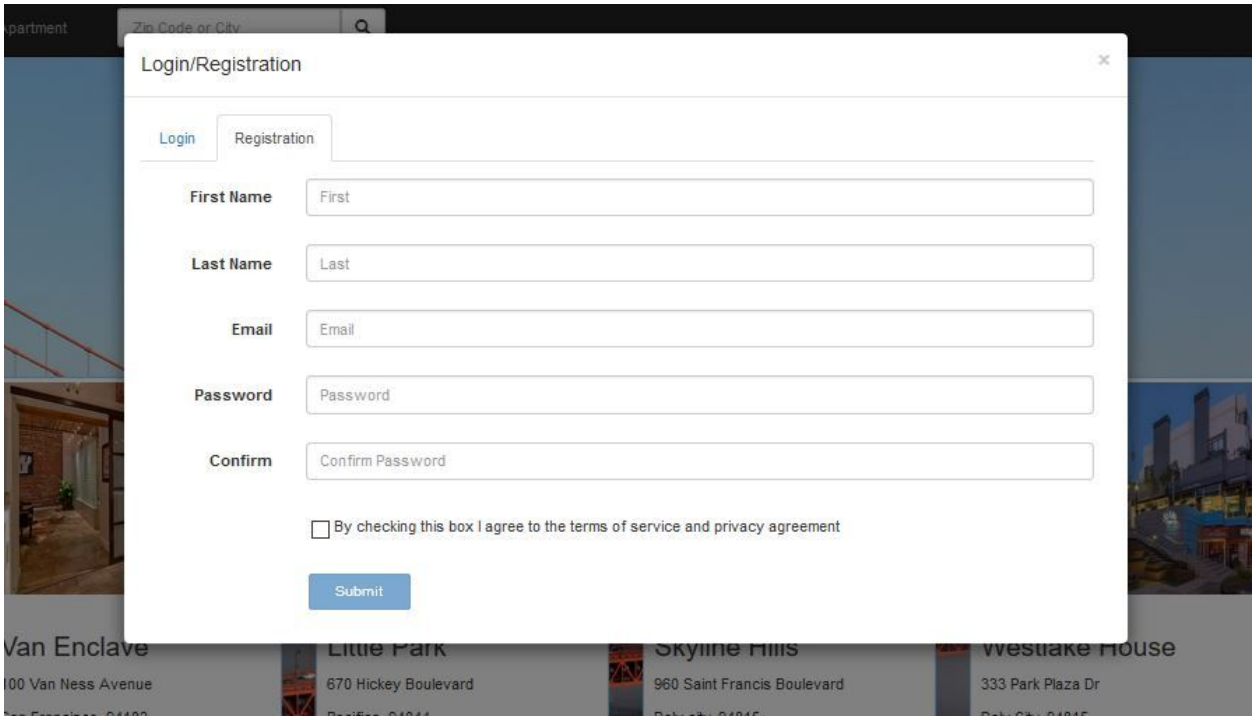
1. Home page



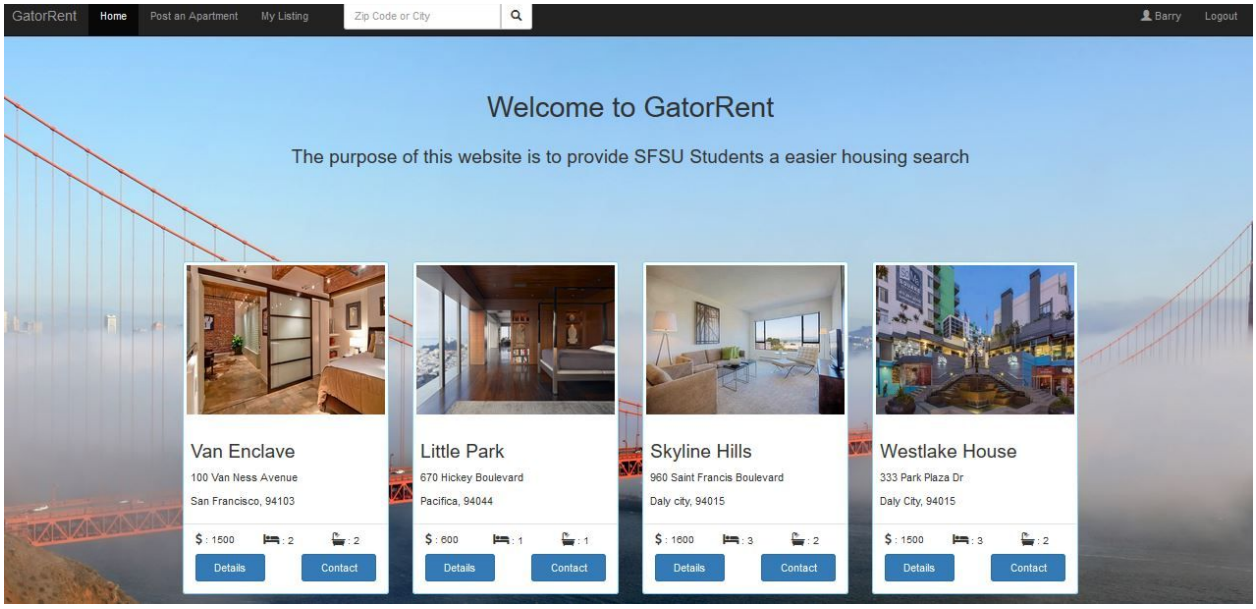
2. Login Modal:



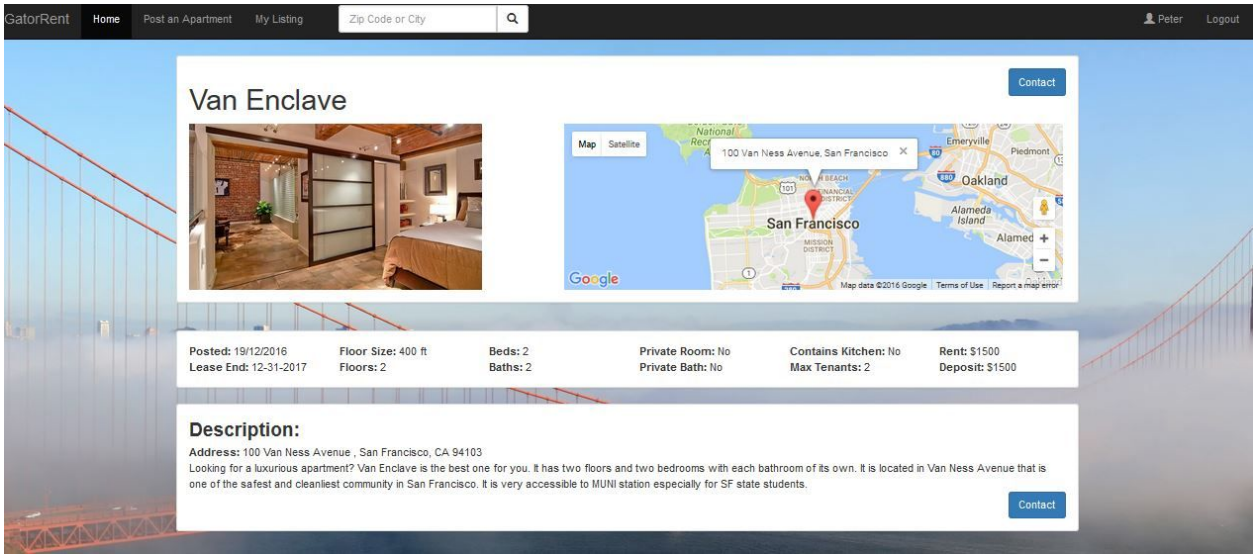
3. Registration Modal:



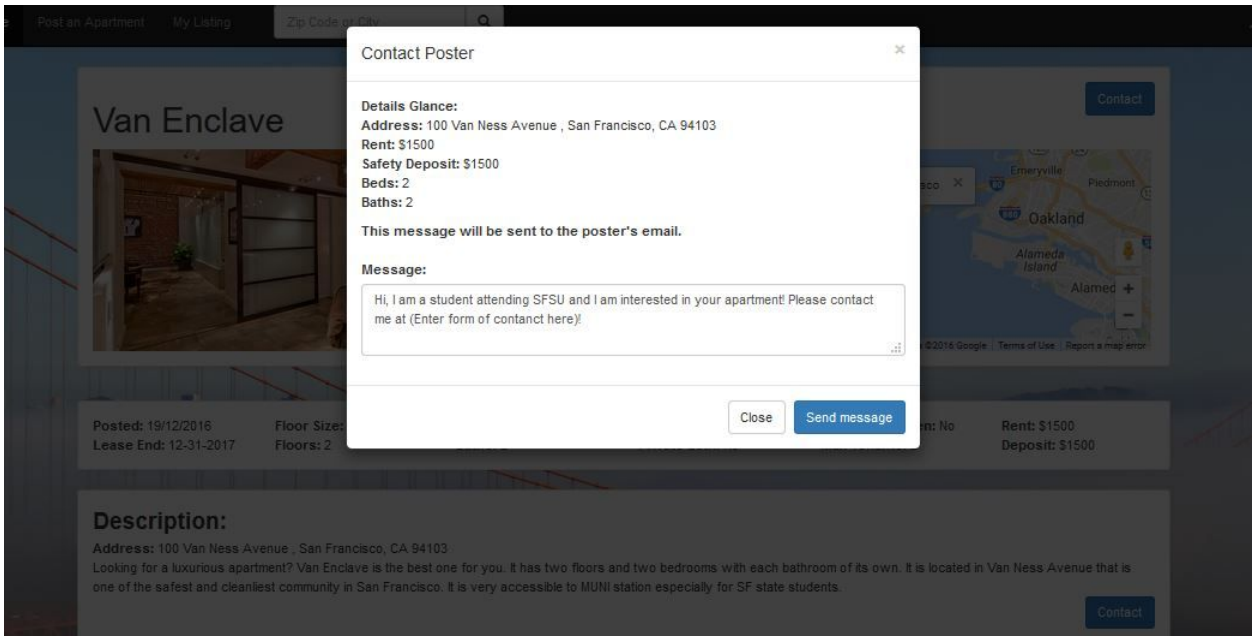
5. Home page after login



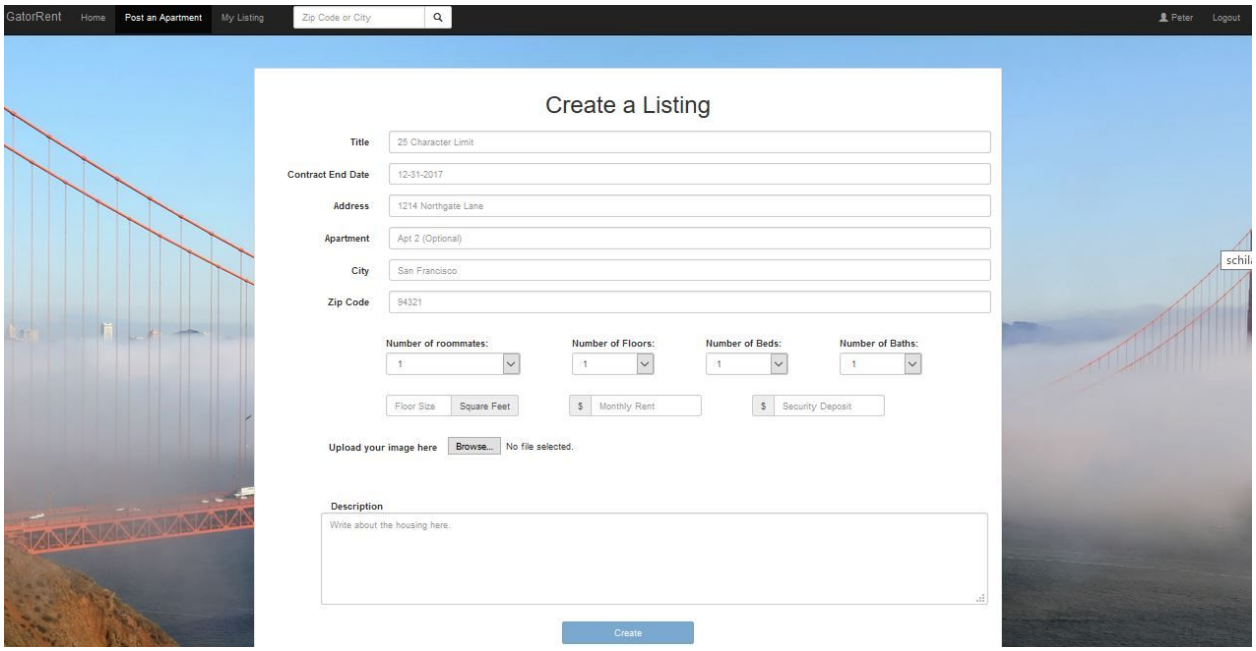
6. Apartment details page



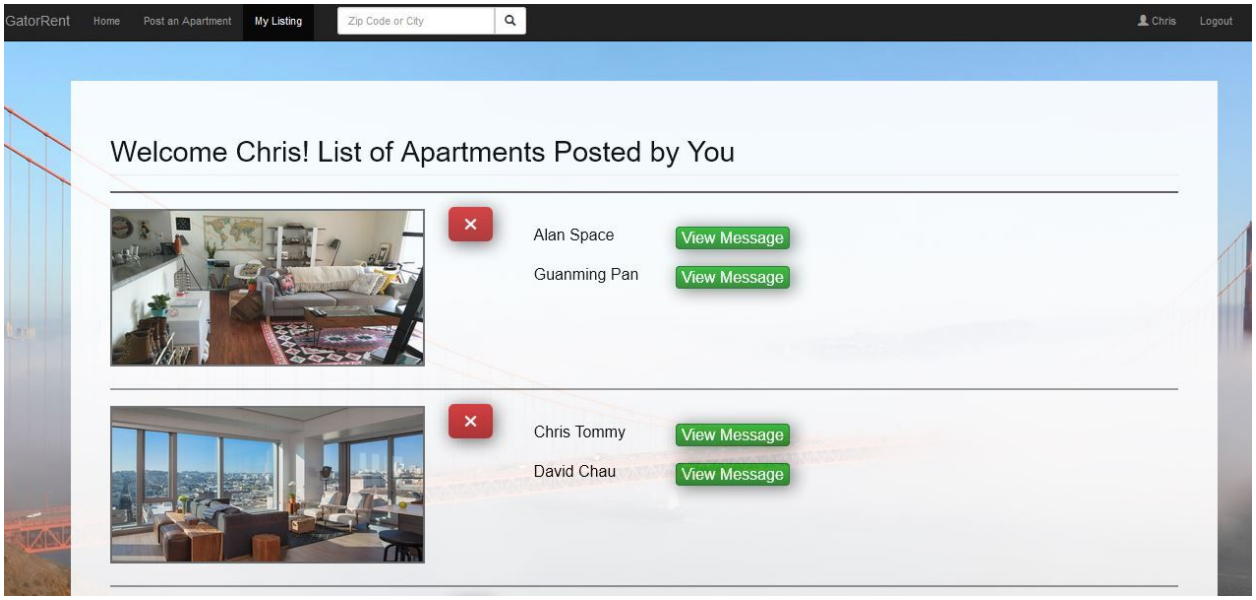
7. Contact modal



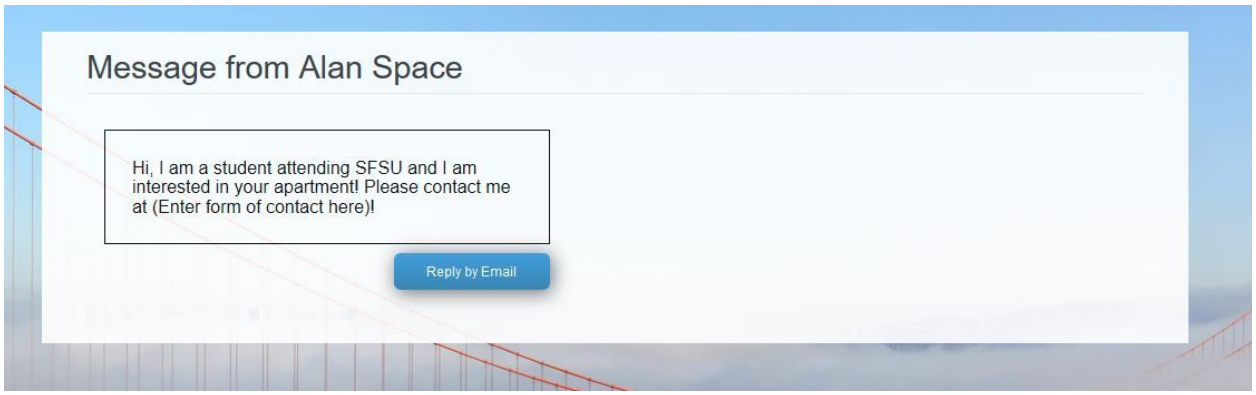
8. Post Apartment page



9. Listings Page

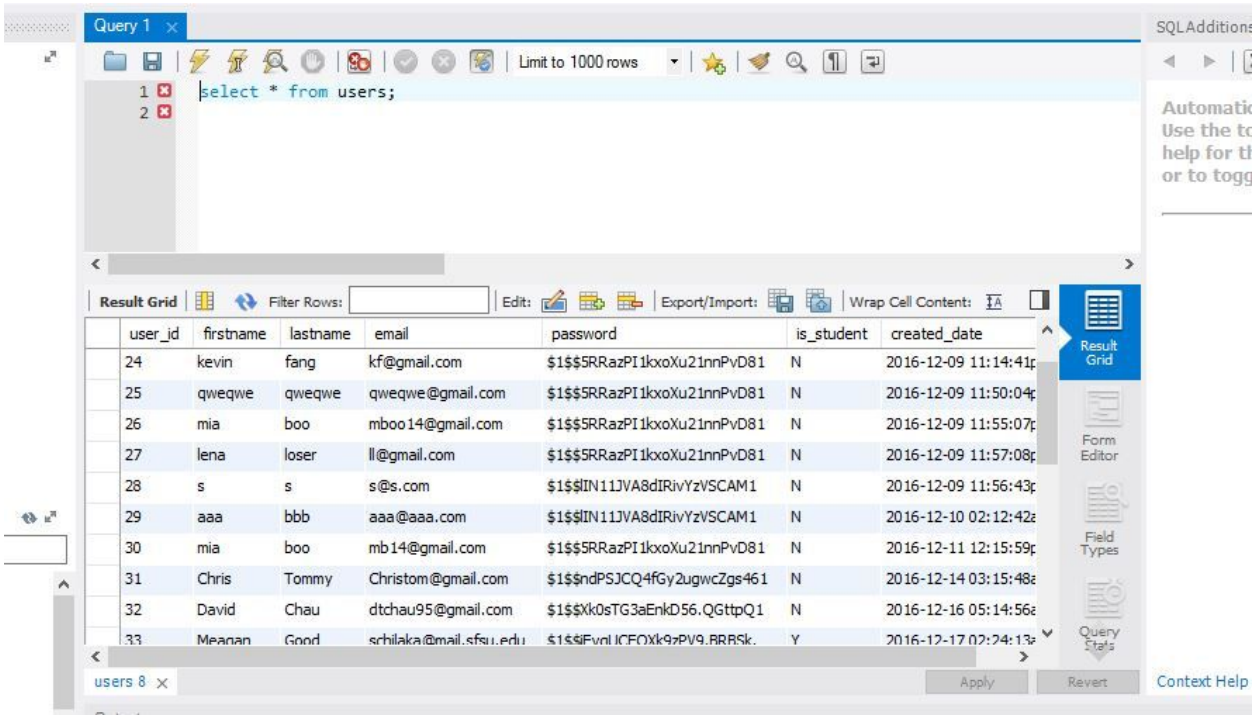


10. Message modal

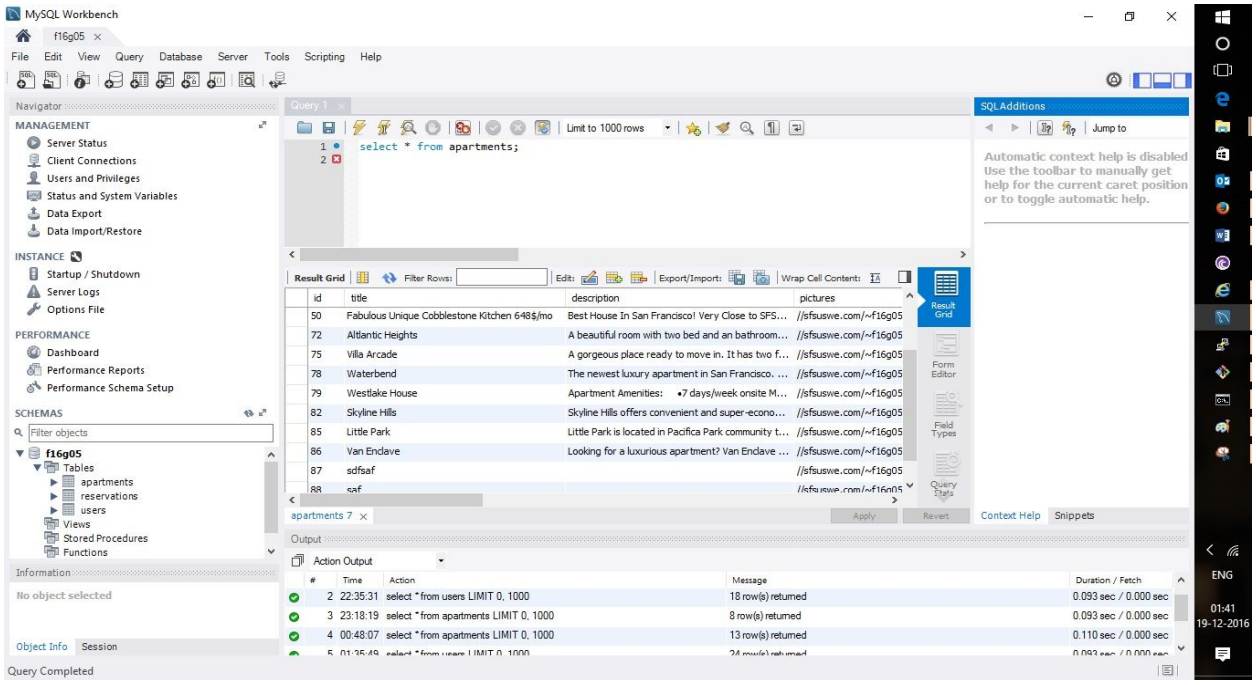


V. Workbench Screen Shots

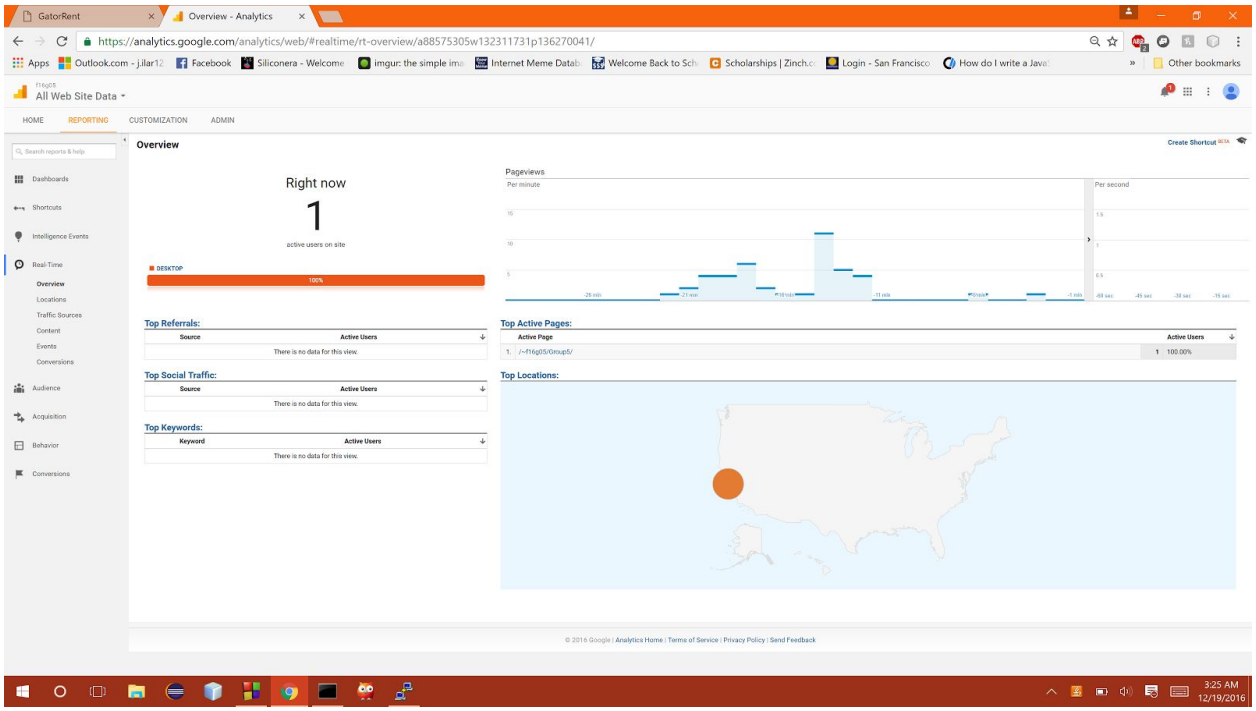
1. Users table

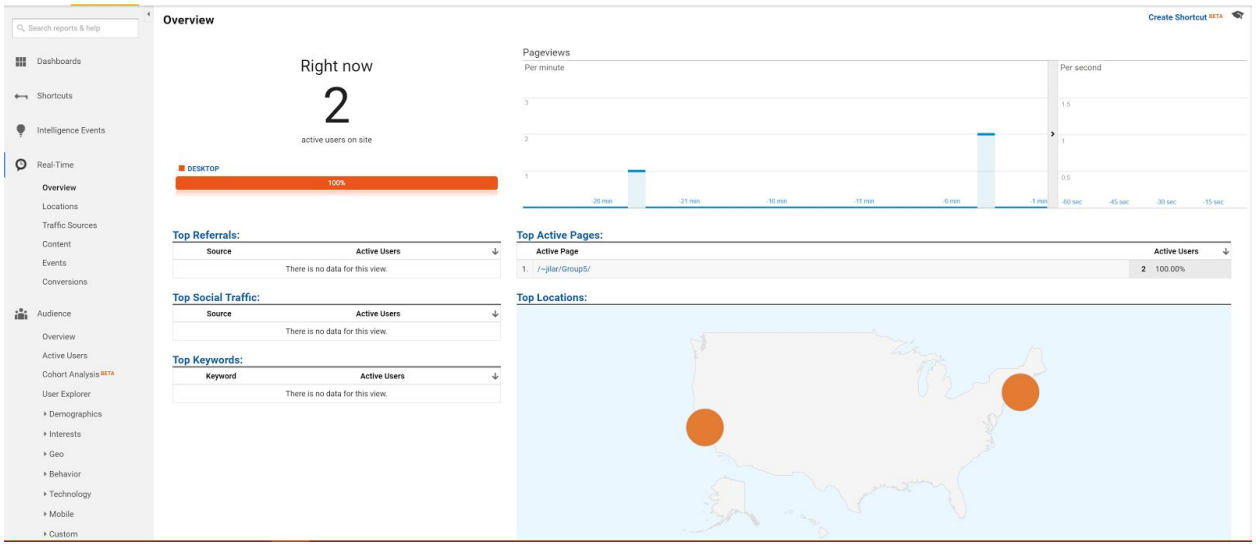


2. Apartments table



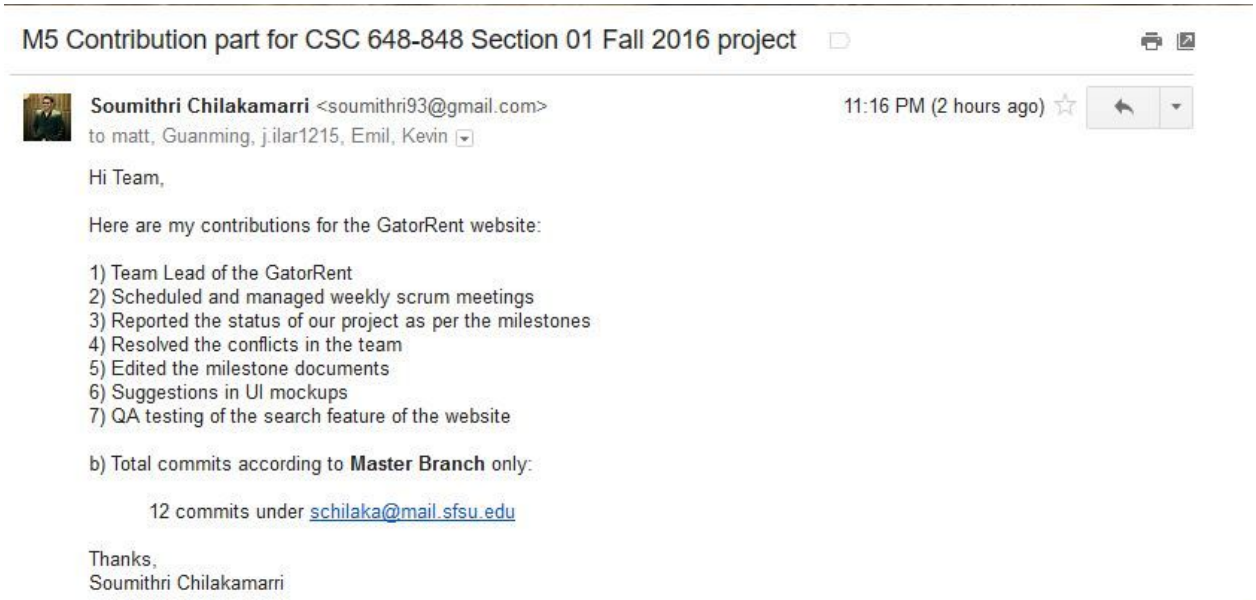
VI. Google Analytics Plot





VII. Team Member Contributions

1. Soumithri Chilakamarri (Team Lead)



M5 Contribution part for CSC 648-848 Section 01 Fall 2016 project

Soumithri Chilakamarri <soumithri93@gmail.com> 11:16 PM (2 hours ago)
to matt, Guanming, j.ilar1215, Emil, Kevin

Hi Team,

Here are my contributions for the GatorRent website:

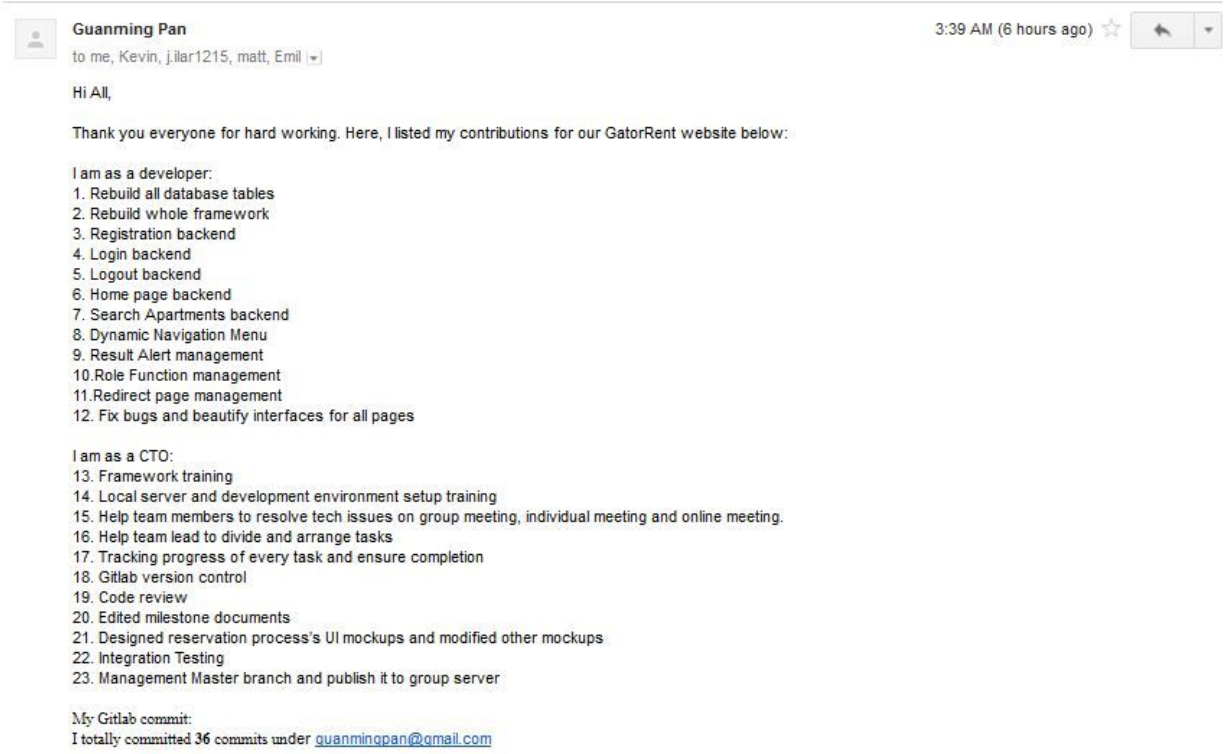
1) Team Lead of the GatorRent
2) Scheduled and managed weekly scrum meetings
3) Reported the status of our project as per the milestones
4) Resolved the conflicts in the team
5) Edited the milestone documents
6) Suggestions in UI mockups
7) QA testing of the search feature of the website

b) Total commits according to **Master Branch** only:

12 commits under schilaka@mail.sfsu.edu

Thanks,
Soumithri Chilakamarri

2. Guanming Pan (C.T.O.)



Guanming Pan 3:39 AM (6 hours ago)
to me, Kevin, j.ilar1215, matt, Emil

Hi All,

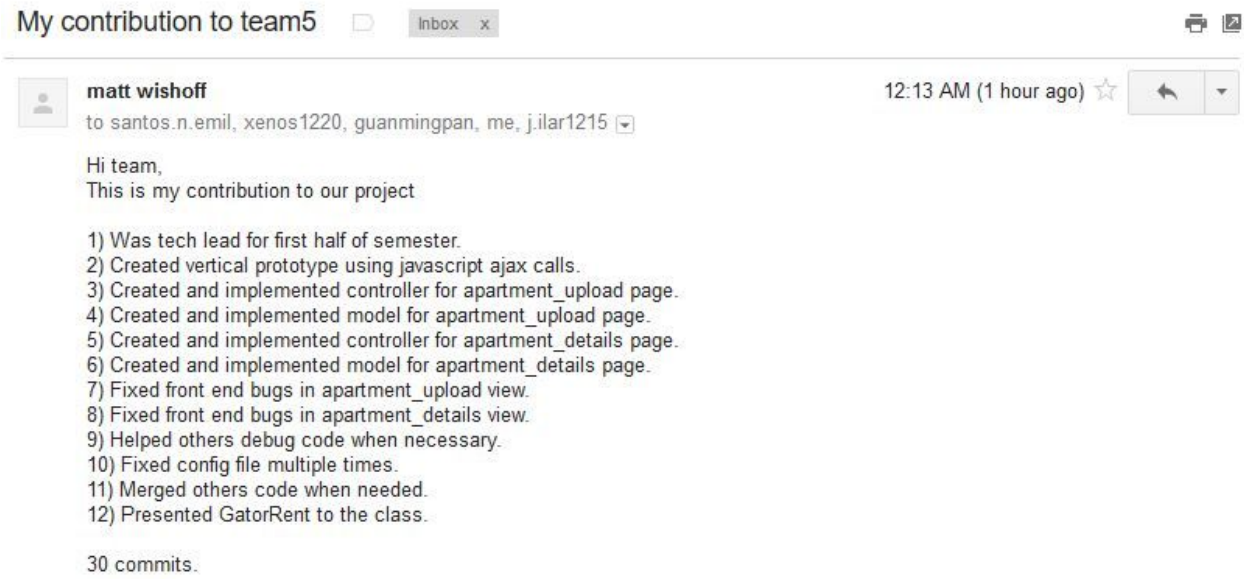
Thank you everyone for hard working. Here, I listed my contributions for our GatorRent website below:

I am as a developer:
1. Rebuild all database tables
2. Rebuild whole framework
3. Registration backend
4. Login backend
5. Logout backend
6. Home page backend
7. Search Apartments backend
8. Dynamic Navigation Menu
9. Result Alert management
10.Role Function management
11.Redirect page management
12. Fix bugs and beautify interfaces for all pages

I am as a CTO:
13. Framework training
14. Local server and development environment setup training
15. Help team members to resolve tech issues on group meeting, individual meeting and online meeting.
16. Help team lead to divide and arrange tasks
17. Tracking progress of every task and ensure completion
18. Gitlab version control
19. Code review
20. Edited milestone documents
21. Designed reservation process's UI mockups and modified other mockups
22. Integration Testing
23. Management Master branch and publish it to group server

My Gitlab commit:
I totally committed **36** commits under guanmingpan@gmail.com

3. Matthew Wishoff



My contribution to team5 Inbox x

matt wishoff 12:13 AM (1 hour ago)

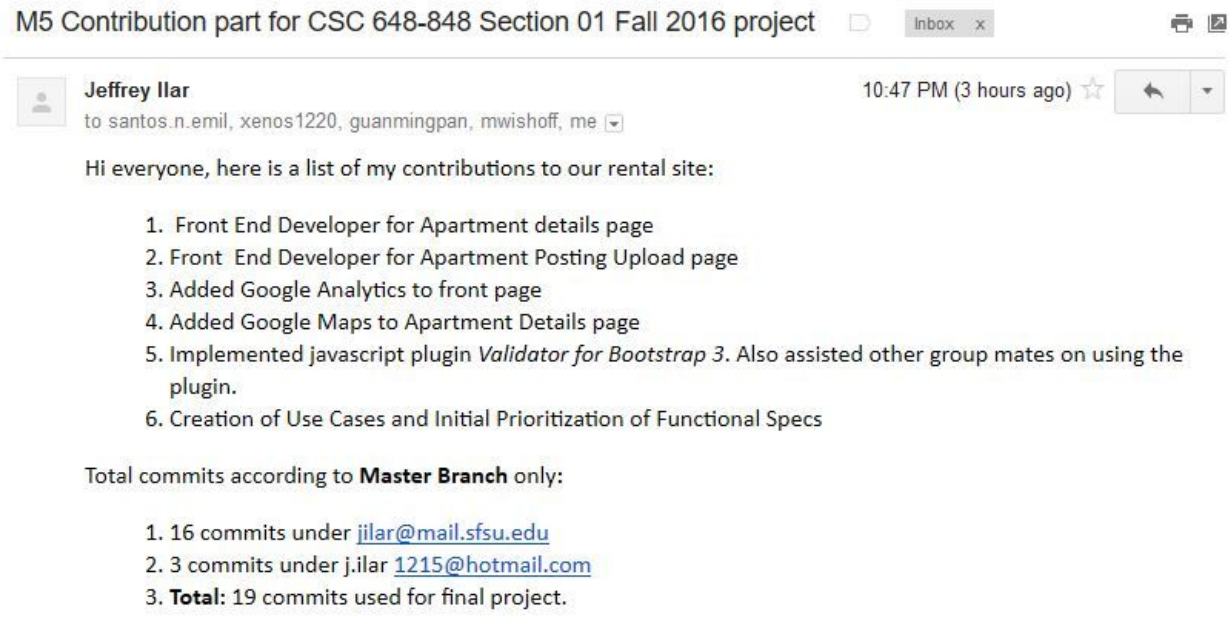
to santos.n.emil, xenos1220, guanmingpan, me, j.ilar1215

Hi team,
This is my contribution to our project

1) Was tech lead for first half of semester.
2) Created vertical prototype using javascript ajax calls.
3) Created and implemented controller for apartment_upload page.
4) Created and implemented model for apartment_upload page.
5) Created and implemented controller for apartment_details page.
6) Created and implemented model for apartment_details page.
7) Fixed front end bugs in apartment_upload view.
8) Fixed front end bugs in apartment_details view.
9) Helped others debug code when necessary.
10) Fixed config file multiple times.
11) Merged others code when needed.
12) Presented GatorRent to the class.

30 commits.

4. Jeffrey Ilar



M5 Contribution part for CSC 648-848 Section 01 Fall 2016 project Inbox x

Jeffrey Ilar 10:47 PM (3 hours ago)

to santos.n.emil, xenos1220, guanmingpan, mwishoff, me

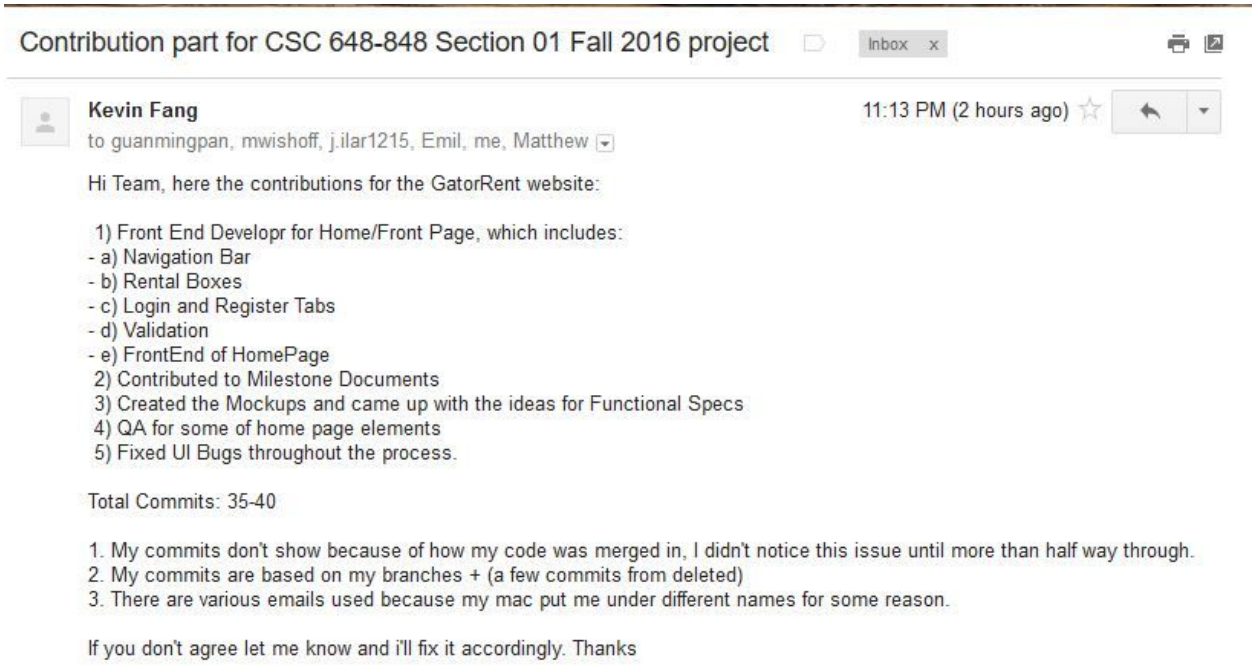
Hi everyone, here is a list of my contributions to our rental site:

1. Front End Developer for Apartment details page
2. Front End Developer for Apartment Posting Upload page
3. Added Google Analytics to front page
4. Added Google Maps to Apartment Details page
5. Implemented javascript plugin *Validator for Bootstrap 3*. Also assisted other group mates on using the plugin.
6. Creation of Use Cases and Initial Prioritization of Functional Specs

Total commits according to **Master Branch** only:

1. 16 commits under jilar@mail.sfsu.edu
2. 3 commits under j.ilar 1215@hotmail.com
3. **Total:** 19 commits used for final project.

5. Kevin Fang



Contribution part for CSC 648-848 Section 01 Fall 2016 project Inbox x

Kevin Fang 11:13 PM (2 hours ago)
to guanmingpan, mwishoff, j.ilar1215, Emil, me, Matthew

Hi Team, here the contributions for the GatorRent website:

1) Front End Developr for Home/Front Page, which includes:
- a) Navigation Bar
- b) Rental Boxes
- c) Login and Register Tabs
- d) Validation
- e) FrontEnd of HomePage

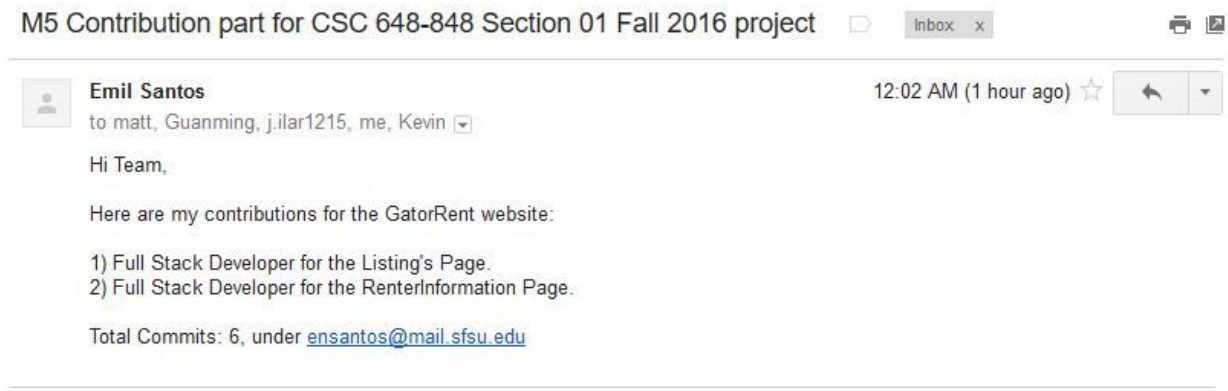
2) Contributed to Milestone Documents
3) Created the Mockups and came up with the ideas for Functional Specs
4) QA for some of home page elements
5) Fixed UI Bugs throughout the process.

Total Commits: 35-40

1. My commits don't show because of how my code was merged in, I didn't notice this issue until more than half way through.
2. My commits are based on my branches + (a few commits from deleted)
3. There are various emails used because my mac put me under different names for some reason.

If you don't agree let me know and i'll fix it accordingly. Thanks

6. Emil Santos



M5 Contribution part for CSC 648-848 Section 01 Fall 2016 project Inbox x

Emil Santos 12:02 AM (1 hour ago)
to matt, Guanming, j.ilar1215, me, Kevin

Hi Team,

Here are my contributions for the GatorRent website:

1) Full Stack Developer for the Listing's Page.
2) Full Stack Developer for the RenterInformation Page.

Total Commits: 6, under ensantos@mail.sfsu.edu