Software Engineering CSC 648/848 Section 02 Spring 2019

Milestone 1

Team 08

Cheryl Fong
::
cfong8@mail.sfsu.edu

Jerry Wong

Chirag Agarwal

Harold Pedroso

Johanna Hadgu

Sharan Lao

Ermias Haile

REVISION TABLE		
Date	**Version Number**	**Description**
2/28/19	1.0	First draft
3/6/19	1.1	Second draft
3/11/19	1.2	Third Draft

FROZEN STATE

1. EXECUTIVE SUMMARY

The launch of GatorRent, an accommodation-search application is intended for the use of the students of San Francisco State University (SFSU) to find housing and for landlords that are looking for renters alike. This application is important as on-campus housing at SFSU is extremely limited and expensive thus making it difficult for both domestic students and international students to find accommodation within their budget and preference.

The GatorRent application will allow a user (users of the GatorRent application includes landlords looking for renters and students looking for housing etc.) to either post houses, apartments, rooms that they are renting out, or for the user to filter through the currently available listings posted on the application. Our application will allow users to find listings based on e.g. distance from the university, rent price. The design of this application will take into account of the technical experience/familiarity of the user by implementing a simple and straightforward design with large buttons and appropriately sized writing along with the use of images and colours to create a more visual-oriented design for ease of use and accessibility. Thus, making GatorRent uniquely tailored for students and landlords as compared to other applications currently on the market.

Our student start-up team is comprised of current computer science students of SFSU, including domestic and international students of different backgrounds. Our personal experiences with finding off-campus and on-campus housing have contributed largely to the ideation of the intended purpose and design of the GatorRent application with the potential range of users in mind.

2. PERSONAE AND USE CASES

Personae

2.1 Kelly

Kelly is a recently admitted college student from Maine, who's attending SFSU in the fall as a freshman. She needs to find a place to live before classes begin. However, she's finding it hard to find an apartment that's close to campus and accommodating for her as a student. Luckily, she stumbles upon GatorRent**.**

2.2 Andrew

Andrew is an ICU nurse at Zuckerberg San Francisco General. After working 6 years in the field, and experienced the successes and struggles of his career, he decides to pursue his Master of Science in Nursing at San Francisco State University. Recalling the earlier struggles of finding housing in his undergraduate years, Andrew decides to use GatorRent to find an apartment.

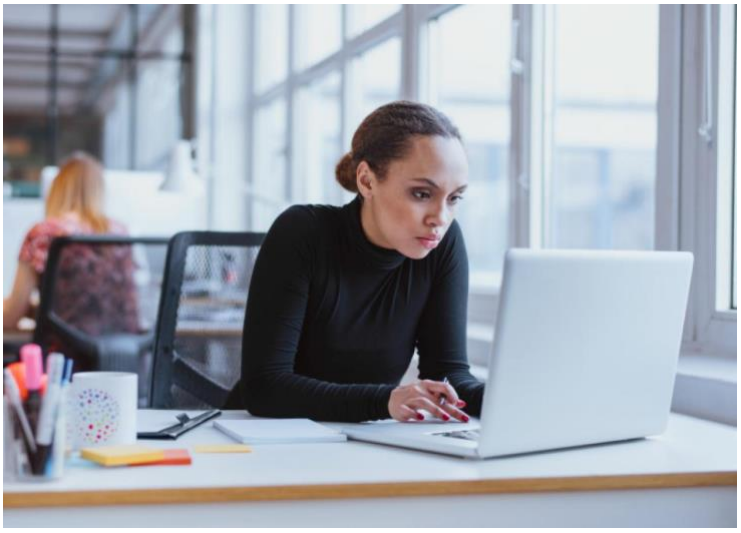
2.3 Max

Max is a junior at San Francisco State University, who recently got accepted to study abroad in Stockholm, Sweden. However, since he is still in the middle of his long-term lease, he realizes that he cannot fulfill his current lease's terms and needs to break it. Max was recently informed by his **landlord** that he needs to find a sufficient replacement. Since Max's

place is relatively close to SFSU, he considers using GatorRent, a site he had used in his sophomore year of university, as a way to find his replacement.

2.4 Jordan

Jordan is a part of the **Admin** team at GatorRent. She recently transferred from a local startup and has spent a sufficient amount of time familiarizing herself with the site.



Use Cases

2.1 Non-registered User (Kelly)

As a **non-registered user**, Kelly is presented with a variety of apartments **listings** available for rent and is able to **browse** through them. Kelly wants to search for apartments that are in close proximity to public transportation, campus and also within her price range. She **filters** her results by **zip code**, **distance,** and **price** and is presented with several apartment listings that are according to her specifications. Now that she found an acceptable apartment, she has solved her problem with GatorRent.

2.2 Registered User (Andrew)

Upon opening the website, Andrew notices that several **listings** are far beyond his price range and are not convenient enough to support his commute to work and school. Therefore, he filters his results according to the **distance** from local transit and **price.** Additionally, GatorRents allows him to accommodate for his cat allergies by **filtering** out renters that allow **pets** to reside in the premises. After finding one apartment that satisfies his conditions, Andrew creates an account and becomes a **registered user,** where he is required to accept the **terms and conditions** of the site. Once this is done, he is able to message the **landlord** and secure an apartment with the help of GatorRent.

2.3 Landlord (Max)

Having logged into GatorRent, Max is able to find options that allow him to post his apartment as a potential place to rent. Once he provides a description of the place, pricing, and images of the **property**, Max is prompted to register or log in before completing the post. Then, Max submits his post for **admin** approval.

2.4 Admin (Jordan)

After logging in to the GatorRent dashboard, Jordan is greeted with a well-organized data list to include **listings** in need of approval and user-abuse reports. As part of her job, during a session of listing evaluation, Jordan comes across a **listing** that did not include photos of the **property**. Jordan denies the listing, contacting the lister and specifying that he'll need to modify his **listing** before it becomes live on the site. After approving several **listings**, she also notices that another **landlord** has been reported by 3 registered users. Having recalled that this specific **landlord** had already received a warning several months prior, Jordan immediately terminates the account.

3. DATA ITEMS AND ENTITIES

Database Name:

Gatorrent-db - This database will contain the following tables below:

The Database Tables:

Table Name	Fields/Columns										
Users	id	username (unique)	password	email							timestamp
Admins	id	username (unique)	password	email							timestamp
Listings	id	user-id (foreign-key)	description		rent-type (house, apartment, room)	rent-type-size	price (per month)	distance from campus	postcode	isPet (bolean - are pets allowed?)	isShared (boolean - is the rent-type shared?)	timestamp
Markers	id	address	lat (lattitude)	long (longitute)	listing-id (foreign-key)						timestamp
Messages	id	subject	body	user-id-sender (foreign-key)	user-id-recipient (foreign-key)	isRead (boolean - is the message read/opened?)					timestamp

Click here to View Image 1.1v Above

Website Meta-Data:

user-session — To keep track of the user's state when using the website.
user favourites — To keep a list of URLs of the user's saved listings.
user-preference — To keep track of user's preferences.
activity-log — A log of entries added or removed from databases.

Glossary:

User/Users	A user of the site. Could be registered or non-registered. If registered, will be stated as “Registered user(s)” otherwise “Non-registered user(s)”.
Registered User	A user who has signed up on the website, i.e. the user’s username, password, and email address will be contained in the Users Table. This user has full access to all user functionalities, such as sending messages to other users.
Non-registered User	A user of the website who did not sign up, i.e. this user does not have an entry on the Users Table. This user is only allowed to browse listings on the website.
Admin/Admins	Website staff who are site usage maintainers, i.e. responsible for handling the appropriateness of user activities by disapproving or deactivating users. This user has full access to all user functionalities.
Student	A user of the site, specifically looking/browsing for listings.
Landlord	A user of the site, specifically looking to rent out their property**.** Has the intention of creating listings.
Filter	The ability/functionality to filter results from the Listings Table.
Listing	An entry on the Listings Table, which includes information on a posted property such as rent-type, its description, rent-type size in square feet, the rent price per month, distance from campus, if pets are allowed, if the rent-type is a shared or not shared occupancy and postcode.
rent-type	Refers to the type of space that is intended to be rented out (e.g. house, room, apartment, etc.).

Registration	An entry added to the Users Table, containing user data during the registration process, such as their username, password, and email address.
Drop-pins	Refers to an entry on the Markers Table, as a visual marker  on Google Maps
Terms and Conditions	The website staff is not responsible for interactions between users outside the realms of the website.

4. FUNCTIONAL REQUIREMENTS

4.1 Non-registered Users:

4.1.1 **S**hall be able to see/browse all **listing** results on the website, excluding the description of **listing** results.

4.1.2 Shall have the option to register themselves as a **user** of the website.

4.1.3 Shall have the option to use the message function to inform **admins** about site related issues, such as inappropriate images or untrustworthy listings.

4.1.4 Shall be able to **filter listing** results by lowest or highest price, price range, postcode, distance from campus, rent-type, rent-type price, if pets are allowed, and if the rent-type is shared.

4.2 Registered Users:

4.2.1 Shall have access to all the functionalities as a **non-registered user**.

4.2.2 Shall be able to see/browse all **listing** results on the website, excluding the description of **listing** results.

4.2.3 Shall accept all **terms and conditions** during the registration process.

4.2.4 Shall be able to publish a property for rent, i.e. this process creates a **listing** entry with the required data fields as stated in the **Listings Table**.

4.2.5 Shall be able to communicate with other **users** via the website message function.

4.2.6 Shall have the option to view their **username**.

4.2.7 Shall have the option to view and change their **email address**.

4.2.8 Shall have the option to view and change their **user** preferences.

4.2.9 Shall receive notifications and the option to turn it off for new messages sent to them.

4.2.10 Shall have the option to receive notifications for newly added **listing** entries that match their price range.

4.2.11 Shall have the option to provide their price range preference on their **user** profile.

4.2.12 Shall have the option to save and unsave desired **listing** results.

4.2.13 Shall have the option to view all saved **listing** results.

4.2.14 Shall have the option to share a **listing** on social media i.e. on **Facebook** and **Twitter** via their respective **API's**.

4.2.15 Shall have the option to view nearby transit methods via the **Google Transit API**.

4.2.16 Shall have the option to view directions to any location they prefer, with SFSU as a quick selection option, from a **listing** result's location address as a starting point.

4.2.17 Shall have the option to rate any other user, if and only if the two **registered users** (the one rating and the one being rated) have at least sent a message to each other via the message function.

4.2.18 Shall have the option to view the number of **registered users** who have their preferred price preference equal to the **listing** price that this registered user has created.

4.2.19 Shall be able to see available **listing** results as **drop-pins** of locations on the Google Maps, via the **Google Maps API**.

4.3 Administration:

4.3.1 Shall have access to all the functionalities of a **registered user**.

4.3.2 Shall approve or disprove **listing** results.

4.3.3 Shall view website **activity log** and act accordingly to unusual website behaviour.

4.3.4 Shall deactivate **users** for their misuse of the website.

5. 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 have to 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 email 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 <u>prominently</u> display the following <u>exact</u> text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2019. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

6. COMPETITIVE ANALYSIS:

	Trulia	Zillow	Apartments	Nestpick	GatorRent
Listings	+	+	+	+	+
Expanded Apartment Info Page	+	++	+	+	+
User Navigation Bar	++	+	+	-	+
Filter	++	+	+	-	+
Search	+	+	+	+	+
Distance from SFSU filter	-	-	-	-	+
Distance from Transportation options	++	-	-	-	+
Map	++	++	+	-	+
- Not Implemented		+ Implemented		++ Implemented Well	

Our website shall comprise of listings, expanded apartment info pages, user navigation bar, and filters as our basic feature set because they are prevalent features amongst real estate listings and rental properties. By polishing within this feature set, it will allow us to gain a market share over our smaller competition and make strides in becoming a larger company.

By incorporating SFSU specific filters to include distance from campus as well as number of nearby SFSU recommended transportation options, GatorRent is focusing on providing resourceful features that will benefit our niche market: SFSU students. Additionally, these features will allow us to dominate within our niche market, therefore potentially increasing our user-base considerably every semester.

7. HIGH-LEVEL SYSTEM ARCHITECTURE AND TECHNOLOGIES

Framework	Python Flask Bootstrap MySQL-Connector-Python
APIs	Google Maps Facebook Twitter Google Transit
Tools	Docker AWS S3 Bucket
System	Ubuntu 18.04 LTS MySQL 5.7 Server Container Python Flask Web Server Container
Deployment Platform	Amazon Web Services (AWS)
Supported Browsers	Desktop and mobile browsers i.e. Chrome versions: 72.0 and 68.0 Firefox versions: 65.0 to 61.0 Safari versions: 12.0 to 11.0

8. TEAM

Position	Names
Team Leader	Cheryl Fong
Front-End Lead	Jerry Wong
Back-End Lead	Chirag Agarwal
Document Master	Johanna Hadgu | Sharan Lao
Team Member Front End	Ermias Haile | Sharan Lao
Team Member Back End	Harold Pedroso | Johanna Hadgu
Github Master	Harold Pedroso

9. CHECKLIST

Status	To Do Item
DONE	Team found a time slot to meet outside of the class. - Every Tues & Thurs at 10am-12pm
DONE	Github master chosen. - Master Harold Pedroso
DONE	Team decided and agreed together on using the listed SW tools and deployment server.
ON TRACK	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.)