San Francisco State University

Software Engineering

CSC 648-848 SUMMER 2020

GatorHub

Milestone 01

Team 03

Megha Babariya (mbabariya@mail.sfsu.edu) (Team Lead)

Abraham Zepeda (Github Master)

Harsh Saxena (Back-end Lead)

Nathalia Sainez (Front-end Lead)

Raymond Kuang

Yaxin Deng

Tania Nemeth

History Table:

Version No.	Date	Comments
01	06/25/2020	Initial Document
02	06/29/2020	Comments (by Professor) implemented

1. Executive Summary

Our team's vision is to develop a web application exclusive for San Francisco State students and faculty called GatorHub to help them find places to rent . We aim to help set our users up for success by designing robust services specifically with their needs in mind. We understand how much an unsound living environment can affect an individual's mental health or ability to focus. Our intent is to streamline the renting process in hopes to alleviate the stresses that burden students and faculty. Our mission is to present practical information about the rooming arrangements to help them find housing that best accommodates their needs.

To make our application competitive and attractive for SFSU faculty and students we will showcase each listings' distance to the San Francisco State campus using the GoogleAPI. We want to enable our users with a solid understanding of the locations that they're considering with respect to the school and other resources. By limiting our tenants to only students and faculty members of San Francisco State, we are able to offer custom features to assist users in finding compatible roommates. Users can message landlords to ask questions and get more information and even leave reviews. Our goal is to ensure that users on our website are equipped with all the tools to feel empowered to find the best location.

Our team possesses useful insight necessary to craft an effective solution as we are composed of current students at San Francisco State University. Unlike our competitors, our application is designed with our users in mind from the beginning to the end. We know firsthand what the pain points are to get to and from the San Francisco State campus and we've learned how to overcome them. Our desire is to develop a simple, easy to use application with an exceptional user experience. GatorHub will not only impact individuals; it will also foster a greater sense of community at San Francisco State.

2. Personae and main Use Cases

2.1 Key Personas:

2.1.1 Tenants (Buyer) :

Eg: Eric

- Shifting to San Francisco from Texas
- Looking for affordable homes
- Concerned with price
- Willing to get a user-friendly website to quickly check for some good places to rent.

Most of the users for renting homes are SFSU students (tenants) or some professors. They have low to intermediate knowledge of web browsing due to advanced and fast developing technologies used by the websites. They are in search of homes at a good affordable range due to lack of income and high prices. So, the main task of the tenants is to find home at an inexpensive price and close to their workplace.

2.1.2 Landlord (Agent/Seller) :

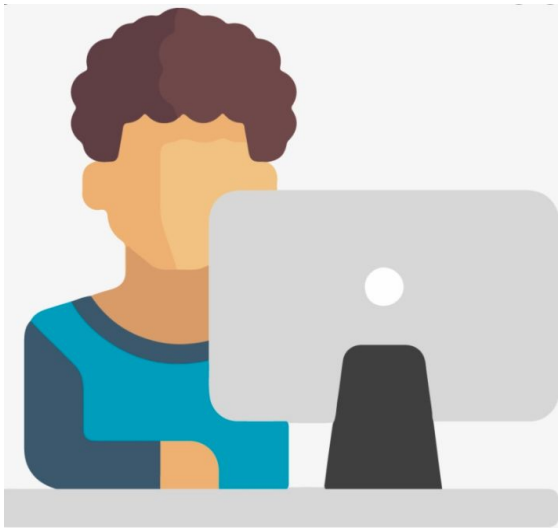
Eg : Sarah

- Sells a home
- Willing to rent the place
- Wants a reliable site to get good tenants
- Very Busy
- Doesn't want to spend much time looking for tenants herself

As Landlords, they want to rent their houses asap. They may or may not be tech-savvy. Don't want to go about physically finding tenants or paying high fees to real estate management firms to get the tenants.

2.1.3 Admin :

Eg: John



- Needs to approve postings from the landlords
- Manages the database
- Handle issues
- Maintain updated information

An admin is responsible for secure maintenance and updating of all data. They have relevant skills to handle complex UI issues or database maintenance. They are also responsible for troubleshooting issues that the tenants or landlords face. All the supervising responsibilities are managed by the Admin department.

2.2 Use Cases:

2.2.1

Eric (Tenant)(Searching for home):

- Desires an affordable home in a desirable neighborhood, ideally within a 25 minute walk or commute to SFSU, with other students in the house. Eric wants to minimize use of his car, so he is seeking a rental that would allow him to walk or bike to the grocery store and some bars or restaurants. He is forced to look from his home in Texas rather than in

person so he needs a tool that can give him information about location and the rental itself. He utilizes the search tool, filtering for his desired price range and move-in date. Eric finds the Google Maps API extremely helpful, as it allows him to explore the areas around each property without being there. He can explore bus routes and MUNI stops near the home, as well as map to the nearest grocery store. He also quickly checks the questions and answers section of each rental, to see if there's any information he would find helpful, before messaging the landlord. He favorites each listing he likes so he can easily look back and keep track of potential homes. Whenever Eric favorites a home, if other SFSU students have expressed interest, he will be notified and have the option to view their profiles. If he thinks someone would make a good roommate/housemate, he has the option to reach out via direct messaging. Eric loves that he can quickly check for new listings daily and eliminate the ones he doesn't like.

2.2.2

Sarah (Landlord) (Renting out Properties):

- Just bought a new home and she's looking for potential tenants to lease the space. Sarah needs one integrated platform where she can easily list her home, filter out tenants, communicate with tenants, keep track of interest in her property and answer questions. She doesn't have a lot of free time and after doing some research, finds our site is reliable and easy to use. Sarah creates a rental listing for her property with all the information (availability, price range, location on Google Maps, floor space, parking, etc.) and waits for a message. She loves that she can easily edit the listing and answer questions asked by potential renters on the Q&A section. Sarah also loves that our platform provides a lot of transparency and helps her avoid going through a company to lease her home. She depends on the direct messaging aspect of the platform, as she hates getting Craigslist emails and text messages and wants to keep all communications with potential tenants in one area. An aggregated, easy to navigate platform is crucial, as it assists her with

keeping track of who is interested, what they've offered and if their profile indicates they would be a good fit. Sarah can also easily track her communication between prospective tenants, without unnecessary stress or switching between multiple websites.

2.2.3

John (Admin) (Managing content, giving permissions and handling data) :

- John is a skilled worker who handles site maintenance for GatorHub and updates for the website. The admin functions built into the application allow him to monitor all content and postings on the site and take action if anything needs to be modified or removed. John frequently manages users and property listings to make sure all content has a place on the site and follows community guidelines. When John sees a user posting inappropriate comments or violating any rules, he decides whether to ban them/their ip address temporarily or permanently. John also takes down scam listings and suspect user profiles, as well a direct line of communication with registered users.

1. Administrator can manage Us
2. Administrator can manage properties
3. Agent (Seller) create property listing
4. Buyer send message to agent
5. User (Buyer/Agent/Anonymous User) can search and list properties (Available for Sell and Rent)
6. User (Buyer/Agent/Anonymous User) can Join/Register
7. User (Buyer/Agent/Anonymous User) can sign-in

3. List of Main Data Items and Entities :

The GatorHub Application is supported by MySQL and its database. MySQL Server provides a good response time of the data being stored making the search effective, convenient way for storing the photographs of the properties and storing the entire description and features of the Property Listings.

1. Registered User :
    1.1. Buyer : A Tenant or Purchaser
        - Will be able to view the property listings, but need to register for being able to see images of the apartment. This user can contact the Seller Agent to buy or rent a property.
    1.2. Agent : A person representing from Landlord to Rent or Sale properties
        - Will be posting property details and pricing, will need approval from Admin to post the properties on Rent or Sale.
    1.3. Admin : A person managing website, users, and its content
        - Responsible for providing permissions to Buyer, Agent, and Registered User. Would be handling all the accounts and listings of the property.

2. Unregistered User :
    2.1. It is an anonymous user who is visiting the website and didn't register and login.
    2.2. An unregistered user will not be able to contact the landlord.
    2.3. Unregistered users will again have the option to create an account as a Landlord, an Agent or a tenant.

3. Property:
    3.1. A house or office or apartment available for Rent or Sale. The property will have its features with image, prices, address, and posted by an Agent.
    3.2. The property will also show the distance from SFSU.
    3.3. The property listing will also have the landlord ID for each Landlord.

4. Property Image:
    4.1. Each property will have zero or many images for its features such as Bed room, Kitchen, Bathroom etc.
    4.2. Each property image will be associated with the listing.

5. Message:
    5.1. The messages communication between buyer and agent regarding property sale or rent.
    5.2. It consists of a message id, body, timestamp value.

6. Favourite Listing :
    6.1. Tenants will be able to list the properties as favourite for future reference.

4. Initial List of Functional Requirements

Unregistered User Functions:

1. Search – Users shall be able to search for real estate.
2. Sort – Sort search results based on parameters such as distance and price.
3. Google Maps API – Users shall be able to view the listing on Google Maps within the webpage.
4. Social Media – Display buttons that users can click to post on their social media (Reddit).
5. Registration and Profiles – Users shall create an account to store data. Data includes personal information such as age, ethnicity, address, contact, and favorites..

Registered User Functions:

1. Login/Logout –Users shall be able to to login and logout of their account.
2. Posting – Users shall be able to post a listing.
3. Edit Posting – Users shall be able to edit or remove their listing.
4. Questions/Answers – Users shall have a section on each listing to post questions. Listing owners shall be able to answer user questions.
5. Favorites – Users shall be able to save listings for later access.
6. Logging – Display how many visits a listing has.
7. Report/Flag – Users shall have an option to report a listing on the page. Reasons include price gouging and false advertising.
8. Internal Direct Messaging – Users shall communicate with each other via direct (private) messaging.

9. Roommate Finder – Users shall be able to search for roommates based on multiple parameters.
10. Audit History – Changes to a listing shall be documented and accessible to users.
11. Ratings for listing and owner – Users shall be able to rate a listing and the owner

Admin Functions:

1. Has permissions to remove listings, content on its page, and user accounts.
2. Administrator approval is required for listings before they can be visible to users.

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 (no localization needed)
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be used

10. No email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
12. Site security: basic best practices shall be applied (as covered in the class) for main data items
13. Media formats shall be standard as used in the market today
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
15. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Summer 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive Analysis

Features	Zillow	Rent.com	Realtor	**GatorHub**
Search bar & Filters (Price, Location)	+	+	+	+
Map (& distance to SFSU)	+	-	+	++
Roommate Finder	-	-	-	+
Contact agent/owner information	++	+	+	+
Can be added to favorites	-	-	-	+
Reviews from users	+	+	+	+
UI design and ease of use	++	-	+	++

+ feature exists ++ superior feature - feature does not exist

Our product will provide SFSU students with an actual place to rent apartments seamlessly that will incorporate both **Distance** from the rental place to SFSU, as well as an option to search for other potential SFSU roommates through **Roommate Finder**. This is in contrast to other online rental places such as Zillo or Realtor, which both focus on the general public and do not provide distances to/from a specific place. The unique feature of our site, and what gives us competitive advantage over the others, is being able to tap into the SFSU niche market through 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. Also, another feature giving us a good advantage vs our competitors involves being able to **Internally Direct Message** the renter for further piece of mind when doing business. This messaging system is internal and no email server is necessary.

Instead of focusing on providing rental services for the general public like the rest of our competition, we only focus on SFSU students with @sfsu.edu verified emails. This is our niche market and what gives us competitive advantage and differentiation amongst the others. All the competitors referenced above provide countless types of filters for home searching such as number of bedrooms, bathrooms, by neighborhood, and many other criteria. But in our plan, the search result will be displayed on both map and in a list side by side by default. This will give our users a better overview of the area they are searching in. Like all of our competitors, we provide our registered users with the option to favorite the property they are interested in to narrow down their potential choices. Our biggest advantage is in the ease of use that will be achieved by keeping a clean and minimalistic look that focuses on the guest.

7. High-level system Architecture and Technologies used

The Real Estate Web Application is built using a layered architecture where the total functionality can be divided into layers having different functionalities. The main layers include the database access layer, business logic layer and the presentation layer. Thus, this application follows the 3-Tier Architecture.

High-level system Architecture Design:

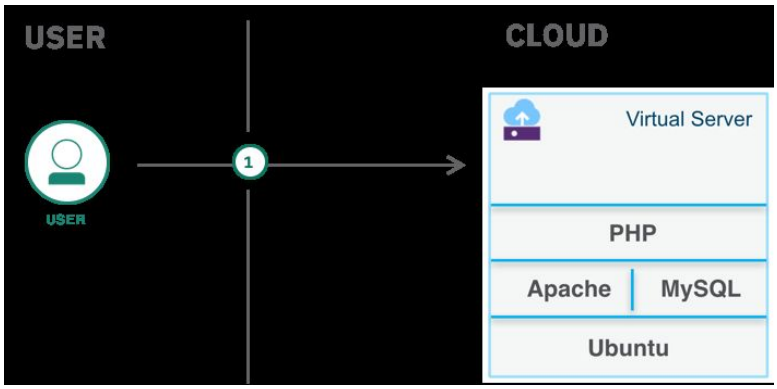


Image 1. High-level design

Architectural Design for the Real Estate Web Application which represents a three tier architecture comprising the Presentation Tier: list of web pages planned to be implemented in the application, the Middle Tier: class diagrams and description of each class and the Data Tier: comprising the ER diagram and database schema.

Tools and Technologies:

1. Server Host : Amazon Web Services (AWS-Free Tier EC2 Instance)
2. Operating System: Linux : Ubuntu 16.04 Server
3. Database : MySQL 5.6
4. Web Server: Apache HTTP WS
5. Server-Side Language: PHP 7.4
6. Frontend : HTML5, BootStrap, JQuery, CSS, JavaScript
7. IDE: PHPStorm (Community Edition),
8. SQL Editor: MySQL Workbench (Community Edition)
9. SSH : Putty/Terminal
10. FTP : FileZilla (Free Version)
11. Google Docs: Functional Requirement, Technical Design
12. Repository : Github

8. Team and Roles

1	Team Lead	Megha Babariya
2	Front End Lead	Nathalia
3	Back End Lead	Harsh
4	Github Master	Abraham
5	Front End Member	Yaxin
6	Back End Member	Tania
7	Back End Member	Raymond

9. Checklist

9.1	So far all team members are engaged and attending zoom sessions when required	On Track
9.2	Team found a time slot to meet outside of the class	Done
9.3	Github master chosen	Done
9.4	Team decided and agreed together on using the listed SW tools and deployment server	Done
9.5	Team ready and able to use the chosen back and frontend frameworks and those who need to learn are working on learning and practicing	On Track
9.6	Team Lead ensured that all team members read the final M1 and agree/understand it before submission	OK
9.7	Github organized as discussed in class(eg: master branch, development branch, folder for milestone documents etc)	Done