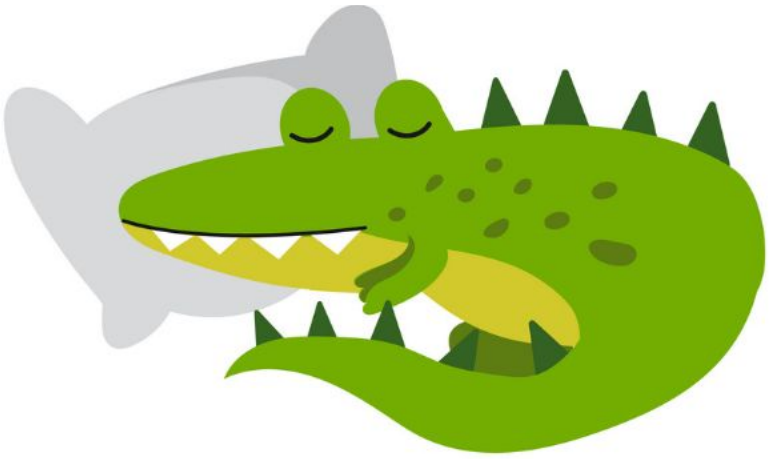
SW Engineering CSC 648/848 Spring 2019



LiveGator

(temp)

Team 13
Amari Bolmer - Team Lead - amarirules@gmail.com
Brian Ho, Simon Tan, Adeel Bhatti,
Kim Wang, Kurtis Hoang, Sushil Kumar Plassar

Milestone 1

March 5, 2019

Table of Contents

1. Executive Summary

Do you think finding somewhere to live around SFSU is unnecessarily difficult? Are you tired of commuting to school from the East Bay? Have you ever wanted to know what your commute options are before moving to a new home? If you said yes to any of these questions, you need LiveGator in your life.

LiveGator is for SFSU students that are looking for somewhere to live near campus. We believe that students should not have to worry about having a roof over their heads while they're struggling with the challenges of school. Currently there is no application on the market targeted at helping SFSU students find housing near campus. We think this is a huge problem because there are over 1000 people waiting for university housing. Hopefully with LiveGator, we can reduce that number, so students can focus entirely on their education.

LiveGator will be a platform that connects students to landlords. Landlords will be able to post photos, descriptions, and a price for their residences. What makes us different from other rental sites is that we automatically include the best ways to get to SFSU on each listing. We will have directions for taking public transit, walking, and driving to campus. Students will be able to browse through the site to find a place to rent. Once they've decided, they will have to make an account and message the landlord. When the connection is made, it is up to the landlord and the tenant to decide how to proceed.

The LiveGator team is made of seven SFSU computer science students, who are frustrated with how difficult it is to rent in San Francisco. We hope to be able to launch this site so we can focus on making software instead of worrying about shelter.

2. Personae and Main Use Cases

<table>
<tr>
<td>Sam
  </td>
<td>About Sam:
Entering Junior year at SFSU
Cannot stay in dormitory for Junior year
Likes to browse websites without signing up
English major
Poetry club</td>
</tr>
<tr>
<td>Behaviors:
Sometimes wakes up late for class
Sometimes indecisive
Willing to move off campus for the right apartment
Reads a lot</td>
<td>Needs and wants:
Wants to find an apartment near campus
Needs pictures to help make final decision
Needs to compare prices</td>
</tr>
</table>

Loren	About Loren
	• Senior at SFSU • Is a music major • Plays the piano and drums • Loves music • Have used LiveGator before
Behaviors: • Pracaticies music throughout the day • Plays drums and instruments loudly • Practicing disrupts people studying • Does not like disrupting people studying	Needs and wants: • Needs to practice playing his instruments • Wants to practice music with other people • Wants to meet other musicians

A. Unregistered Users:

Sam is entering her Junior year at San Francisco State University. No longer able to stay in the dorms, she has to find a place to live before the semester starts. She is unsure of whether she wants to rent an on-campus apartment or find one off-campus. She loves the idea of being close to school, so she browses LiveGator for apartments to rent and uses the pictures, price compare, and distance filter features to find the right apartment for her near campus. She finds her perfect home and is prompted to register for LiveGator in order to contact the poster.

B. Registered Users:

Loren is a Senior and a music major at San Francisco State University. He has received a few complaints from his current housemates about his instrument practicing being too disruptive throughout the semester. Having used LiveGator before, he remembers there is a feature to filter rentals by majors. He uses LiveGator to find a new house full of other music majors, hoping not to disrupt other people studying because everyone else would be practicing their instruments as well. As he browses through the listings on LiveGator, he keeps an eye on the “about housemate” section where there are icons that indicate the major’s of the other students living there (e.g. a book for english majors, a flask for science majors, a music note for music majors, ect).

C. Landlord:

Shannon has a decided to rent out her second home to students. She has heard about the website LiveGator from her niece and has decided to use it instead of hiring a real estate agent. Although Shannon does not generally use computers, the easy to follow instructions prompts her to input all relevant information (e.g. pricing, photos, contact info, etc). She especially likes the “more info” section where she is able to post an introduction about herself and a set of house rules for prospective renters. The UI informs Shannon that her post must be approved, then she fills out all the information, and she must agrees before submitting that her post must be approved first before going live. She agrees and receives an email letting her know her post has been received and is being reviewed.

D. Administrator:

Kerry is a LiveGator team member. She was hired because she has a background in working with spreadsheets. Because of this experience, she has some decent technical skills. When she was hired, she was given training on what is acceptable to post on the website. Kerry would log in like a normal user, except she would have an admin account. Then she will see a dashboard with pending posts and approved posts. She would then

look through the pending posts and approve the ones that meet the website's guidelines. If the post do not meet guidelines, she rejects the post and ask the submitter to make the necessary changes accordingly.

3. List of Main Data Items and Entities

A. Unregistered Users:

Users who have not signed up or logged in. They can use the website to browse the listings and house details. They cannot send renting requests or contact landlords.

B. Registered User:

Registered users can be both customers and landlords at the same time, but they shall have different dashboard to view specific activities. They do not have to be SFSU students. They can browse the website and listing details. Also, they can contact landlords, send renting requests, and post their properties for renting.

C. Administrator:

Users who have access to the database and perform administrative tasks.

D. Listing

Listed properties on the website. Listings can be houses, apartments, condos or other types.

E. User record

User record includes user messages and user orders. Registered users can view their customer dashboard or landlord dashboard to view their completed orders and messages.

4. Initial List of Functional Requirements

A. Unregistered User

1. Shall be able to register an account or log in
2. Shall be able to browse the available houses
3. Shall be able to view listing details
4. Shall be able to filter listings by price, size and distance range
5. Shall be able to sort listings by price, size and distance

B. Registered User

Shall be able to do what unregistered users can do plus:

6. Shall be able to contact landlords
7. Shall be able to make a request to rent properties
8. Shall be able to contact registered customer who has sent renting request
9. Shall be able to make add, delete, and edit renting properties

C. Administrators:

Shall be able to do what regular registered users can do plus:

10. Shall be able to access the database
11. Shall be able to delete listings
12. Shall be able to block users
13. Shall be able to approve listings for posting

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 e-mail clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class)
13. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
15. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2019. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive Analysis

Currently in the market there are three primary ways students find housing around our campus. All three of them are free to use and easy to access: Craigslist, Zillow and Facebook.

On Craigslist and Zillow- master tenants, realtors or landlords can post listings of housing situations. After listings have been made- students can browse and search to find an appropriate fit. They can sort and filter by price and location. After they click on a listing there are photos and there is text information displayed. Despite having good browsing and search, Zillow and Craigslist both lack any form of messaging.

On Facebook, students can post in the SFSU housing group any vacancies they are trying to fill. This is different than the anonymous nature of Craigslist and Zillow as people making the listings can see the students responding and vice versa. However, Facebook is weak in it's browsing capabilities and search functionality. The search functionality is not detailed and it consistently shows listings that are outdated.

Coupled with knowing that all the users on LiveGator will be SFSU students- messaging will help renters build trust with landlords. In addition to all of the features these 3 competitors have, we are including a "Sort by distance to SFSU" feature to sort listings according to how far they are from the SFSU campus.

Features:

	User-Registration	Messaging	Browse	Search	Distance to SFSU
LiveGator	x	x	x	x	x
Craigslist	x		x	x	
Facebook	x	x	x		
Zillow	x		x	x	

As you can see, LiveGator is due to have all of the same features as current market leaders in addition to our "Distance to SFSU" feature. Each of the competition has their own weak functionalities that we hope to create better versions of.

7. High-level System Arch & Technologies Used

A. Platform

1. Ubuntu 16.04 - Linux operating system
2. Amazon Web Service - Cloud service platform
3. Apache 2.4 - Web server

B. Server-Side Language

1. Python 3.5 - high-level programming language

C. Frameworks

1. Bootstrap v4.3.1 - CSS framework for responsive mobile-first websites
2. MySQL 8.0 - Database management system
3. Flask 1.0 - Web framework

D. IDE

1. PyCharm IntelliJ - Integrated development environment specifically for the language Python

E. Tools

1. GitHub - Git repository hosting service
2. Git - Distributed version control tool
3. MySql Workbench - Visual database design tool

F. API

1. Google Analytics - Web service that tracks and reports website traffic
2. Google Maps - Web-based service that provides detailed geographical information around the world

G. Supported Browsers

1. Chrome - version: 72.0.3626
2. Firefox - version: 65.0.1 & 60.5.1
3. Safari - version: 10.14 Mojave & 10.13 High Sierra

8. Team

Amari Bolmer	Team Lead
Brian Ho	Front End Lead
Kim Wang	Back End Lead
Sushil Kumar Plassar	Front End/Github Master
Simon Tan	Front End
Kurtis Hoang	Back End
Adeel Bhatti	Back End/Document Master

9. Checklist

1. Team found a time slot to meet outside of class -
   DONE

2. Github master chosen -
   DONE

3. Team decided and agreed together on using the listed SW tools and deployment server -
   DONE

4. 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

5. Team lead ensured that all team members read the final M1 and agree/understand it before submission -
   DONE

6. Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) -
   DONE