SW Engineering CSC648/848 Spring 2019

GatorHouse

Team 14 (Local)

Davis Hoang (Davis.hoang2015@gmail.com) (Team Lead / Github Master)

Front-End

Jonathan Fox (Front-End Lead)
Kevin Mitsuda
Jarrett Lee

Back-End

Hashim Jacobs (Back-End Lead)
Eric Chen
Shalaka Aigal

Milestone 4
05/08/2019

Revision Table		
05/07/2019	1.0	Initial Document
05/20/2019	2.0	Revised Document

Product summary

Gatorhouse is a service that anyone can use for hassle-free house-hunting or house-selling. If you are a student, faculty member, or just someone searching for a new living situation near San Francisco State University, Gatorhouse is your new best friend.

1. **Simple Search**

   A simple search and a couple of clicks will get you that much closer to acquiring your ideal living space. With Gatorhouse, you're able to connect with multiple landlords with a high degree of simplicity in just minutes.

2. **Upload Property**

   If you're a seller, post your property with brief details, sit back, and watch the buyers/renters flood in. Gatorhouse is here to do the hard work so that everyone can live their best life.

3. **Admin control**

   Gatorhouse enforces reliable house listings. Our admin will be able to approve or reject uploaded house listing before they go live within 24 hours.

You can reach us at: http://ec2-52-53-199-247.us-west-1.compute.amazonaws.com

Usability Test Plan

Test objectives:

Purpose: Verify that GatorHouse is usable to average users and that users would use GatorHouse for renting housing online.

Problem Statement: Is the main function of GatorHouse, (housing rental), easy for any average user to use?

Usability Task description:

Users will be able to locate and search for housing. Users will be able to view specifics of the housing such as number of rooms, cost and type(studio,apartment,or house).
If a user is interested in housing they do not need to register and can just use the contact information listed on the page to contact the landlord to rent the housing.

-User Profile: College student of the bay area, conversational level of English, ages 20 to 30. Tests will be presented to voluntary participants on the test monitor's personal computer

-Method and test design: Users will be asked to perform a simple search for a type of housing (studio,apartment,or house) that meets the criteria (lowest cost,number of rooms) and view the contact details of said housing.

-Test environment and equipment:

Hardware Setup- Website on Amazon Web Services running on a MAC Machine

Software Setup- GatorHouse default homepage on Chrome browser of a MAC Machine. 12 different types of housing are present in the postgres database.

-URL: http://ec2-52-53-199-247.us-west-1.compute.amazonaws.com
-Test monitor role: The test monitor will simply silently observe and time the amount of time it takes for the User to reach the details page of a house that meets the criteria of the user.

-Legal issues: Test is up to the participants and it totally voluntary. Users who test will not be asked their name or any other criteria. The only information we are keeping from the users will be their experience with the website itself. No personal information will be kept because Users will not be entering any information themselves, but if users do enter personal information accidentally, all of that information will be deleted.

-Report: The final report will contain information based on how quickly users found it to navigate through GatorHouse to find an apartment that meets their criteria.

Questionnaire:

Task 1: Locate the type of housing that is on 526 Font Blvd

Task	Find the type of the housing on the address 526 Font Blvd
Machine State	Home page of GatorHouse: http://ec2-52-53-199-247.us-west-1.compute.amazonaws.com
Success Criteria	Seeing the type of the house on 526 Font Blvd
Benchmark	13 seconds

Circle the statement that best describes how much you agree with each statement

-It was clear where to type the address on the page

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

-The search was quick and displayed the correct result

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

-After locating the house the type of housing at the address was clearly indicated

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

Task 2: Locate the price of any house on “Bark Madly Court”(an address)

Task	Find the price of a house on 100 Font Blvd
Machine State	Home page of GatorHouse: http://ec2-13-52-77-234.us-west-1.compute.amazonaws.com:8080/
Success Criteria	User can see the price of houses on Bark Madly Court
Benchmark	30 seconds

Circle the statement that best describes how much you agree with each statement

-It was clear where to type the address on the page

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

-The search was quick

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

-Displayed the correct result

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

-The prices of the houses on Bark Madly Court are clearly displayed and obvious

Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree

Comments:

QA test plan

Test objectives: QA testing is required in software in order to confirm that the software meets requirements.Further testing is then required to ensure that all features of priority one are completed and tested to meet specifications previously mentioned.

HW setup: Using Amazon Web Services, running on a Linux Machine
SW setup: Home page of GatorHouse on a Chrome browser of Windows 10.
URL: http://ec2-52-53-199-247.us-west-1.compute.amazonaws.com
Feature to be tested: Search function
Housing can be located by entering the housing’s address by name, price, housing number, and type.

QA TEST PLAN

Number	Description	Test input	Expected output	PASS/FAIL
1	Test % like in search for name field	Type in “Townhouse” in the search bar	Check that results and see 1 listing is displayed	PASS
2	Test if search returns all listings when given no input	“ “	Get all available listings	PASS
3	Test for searching by filtering type	Change dropdown value to “room”	Get all listings that are labeled as room	PASS
4	Test for searching by price	“3400”	Get all listings that are listed at $1,600	PASS
5	Test for search by housing number	“526”	Get all listings that have a housing number of 777	PASS

Code Review:

The code format that we use for Gatorhouse includes the following:

- Include header comments for all files
- Function usage comments are not necessary if name is adequately intuitive
- Two-space indentation for nested blocks
- Function and variable naming will be in camel case
- Functions and variables must be named with emphasis on purpose. For example, if the function searches for and returns rows in a table called “listings” that includes keywords from user input, it should be written in the following manner:

```
searchPropertiesByKeywords(searchInput){
  …
  res.send(properties);
}
```

- We use the ‘prettier’ javascript package, so formatting is done automatically to guarantee consistency.

Self-check on best practices for security

- Major assets being protected include email address and password.
- emails stored in secure DB
- Password in the database have been encrypted using the bcrypt hashing function
- Input validation is presented in login and registration.
- validate search text up to 40 alphanumeric characters
- Store images as files

Self-check: Adherence to original Non-functional specs

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) (DONE)
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers (Done)
3. Selected application functions must render well on mobile devices (ISSUE - Images are not properly aligned, Cannot click on Navbar dropdown menu)
4. Data shall be stored in the team's chosen database technology on the team's deployment server. (Done)
5. No more than 50 concurrent users shall be accessing the application at any time (Done)
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. (Done)
7. The language used shall be English. (Done)
8. Application shall be very easy to use and intuitive. (Done)
9. Google analytics shall be added (ISSUE - URL contains a fragment)
10. No e-mail clients shall be allowed (On Track)
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated. (Done)
12. Site security: basic best practices shall be applied (as covered in the class) (On Track)
13. Before posted live, all content (e.g. apartment listings and images) must be approved by site administrator (Done)
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development (Done)
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). (Done)