SW ENGINEERING CSC648/848 SECTION 02 FALL 2017

Landmark

Milestone 4

Team 03

Jesse Gabriel (jgabrie2@mail.sfsu.edu),
Avneesh Setia, Huiliang Huang, Yingjing Chen, Jenny Ngo,
Matthew Berkman, David Lau

12/7/2017

History Table	12/7/17	Rough Draft #1 Submitted for Review
	12/17/17	Addressed the following comments from the CEO: • In Product Summary, the dashboard and URL has been listed. • In Usability Test Plan, a few text explanations have been added to address the objective of the test plan and changed the wording of the Likert Scale statements. • In QA Test Plan, the task description and a few test plans has been omitted, and more text explanations have been added.
	12/17/17	FROZEN

1) Product Summary

Name of product: Landmark

The available features that Landmark will have are:

1. Searching for Homes: Potential buyers will be able to look-up available listings with the search bar by entering the city, zip code, or specific address they plan to live at.
2. Filtering: Potential buyers can filter the search results that they find from lowest to highest price along with specifying the number of bedrooms or bathrooms.
3. Viewing listings: Potential buyers will see multiple listings with images which overlay a brief description of the listing's price and number of bedrooms and bathrooms.
4. Registration and Login: Anyone will be able to sign up for an account which allows them to access their personal dashboard.
5. Upload: Any agents who plan to use our website must sign up for an account which allows them to post real estate information on our website for potential buyers to view.
6. Dashboard: Agents will have access to see the listings they have uploaded, along with the ability to edit information of the selected listing
7. URL: https://sfsuse.com/fa17g03/

2) Usability Test Plan

Testing Functionality for Search

This usability test plan will test the usability of the search function. The objectives listed will test specific tasks someone would like to perform on the website. There will also be certain setup that a typical person would have when they are using the website. And lastly, a Likert Scale is provided to gauge the usability of the search function.

Test Objectives

The objective is to evaluate the usability of the search function. The table below shows each task will address a different type of usability and have a description of what the task will test.

Task	**Description**
Find a listing under a specific city	Someone should be able to find a listing by specifying the city a listing may be at.
Find a listing under a specific zip code	Someone should be able to find a listing by specifying the zip code.
Find a listing under a specific address	Someone should be able to find a listing by specifying the address.
Filter by Price	Someone should be able to filter the results from lowest to highest or highest to lowest.
Filter by Number of Bedrooms	Someone should be able to filter the results by how many bedrooms they would like in their new home.
Filter by Number of Bathrooms	Someone should be able to filter the results by how many bathrooms they would like in their new home.
Find more info for a listing	Someone should be able to get more information from a listing.

Test Plan

- System setup
  - Test environment: Google Chrome, Mozilla Firefox using internet connection
  - Equipment: desktop, laptop, tablet, and mobile devices.
- Starting point: Home page
- Intended users are prospective buyers and sellers.

URL of the system to be tested
https://sfsuse.com/fa17g03/

Completion Criteria

User has successfully done the following:
Found all listing properties according to city name, zip code, or address.

Likert Scale

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The GUI was user friendly and easy to navigate					
The search functionality was easy to use					
The listings displayed the necessary details cleanly					
The website layout resized well to different screen sizes (mobile, desktop etc.)					
Search filters were easy to use					

Optional Comments:

____

____

____

____

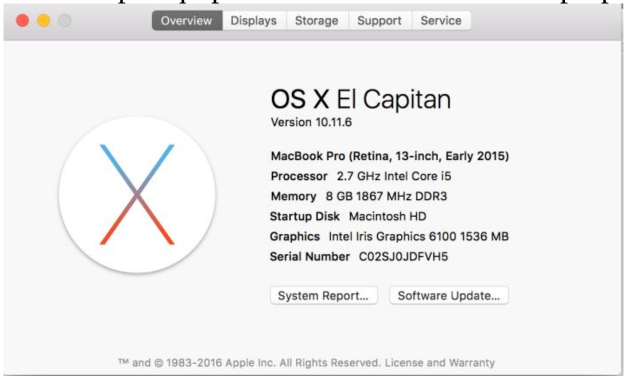
3) QA Test Plan

Test Objective

The objective of the QA will be to evaluate the functionality of the search function by providing specific inputs and observing if the expected output appears.

HW/SW Setup

- HW Setup: Equipment is a MacBook Pro laptop



OS X El Capitan
Version 10.11.6

MacBook Pro (Retina, 13-inch, Early 2015)
Processor 2.7 GHz Intel Core i5
Memory 8 GB 1867 MHz DDR3
Startup Disk Macintosh HD
Graphics Intel Iris Graphics 6100 1536 MB
Serial Number C02SJ0JDFVH5

™ and © 1983-2016 Apple Inc. All Rights Reserved. License and Warranty

- SW setup:
  - Google Chrome browser (version 61.0.3163.100) using a Wi-Fi connection.
  - Firefox browser (version 57.0.1) using a Wi-Fi connection.
  - URL of the software to be tested: https://sfsuse.com/fa17g03/

Test Chart

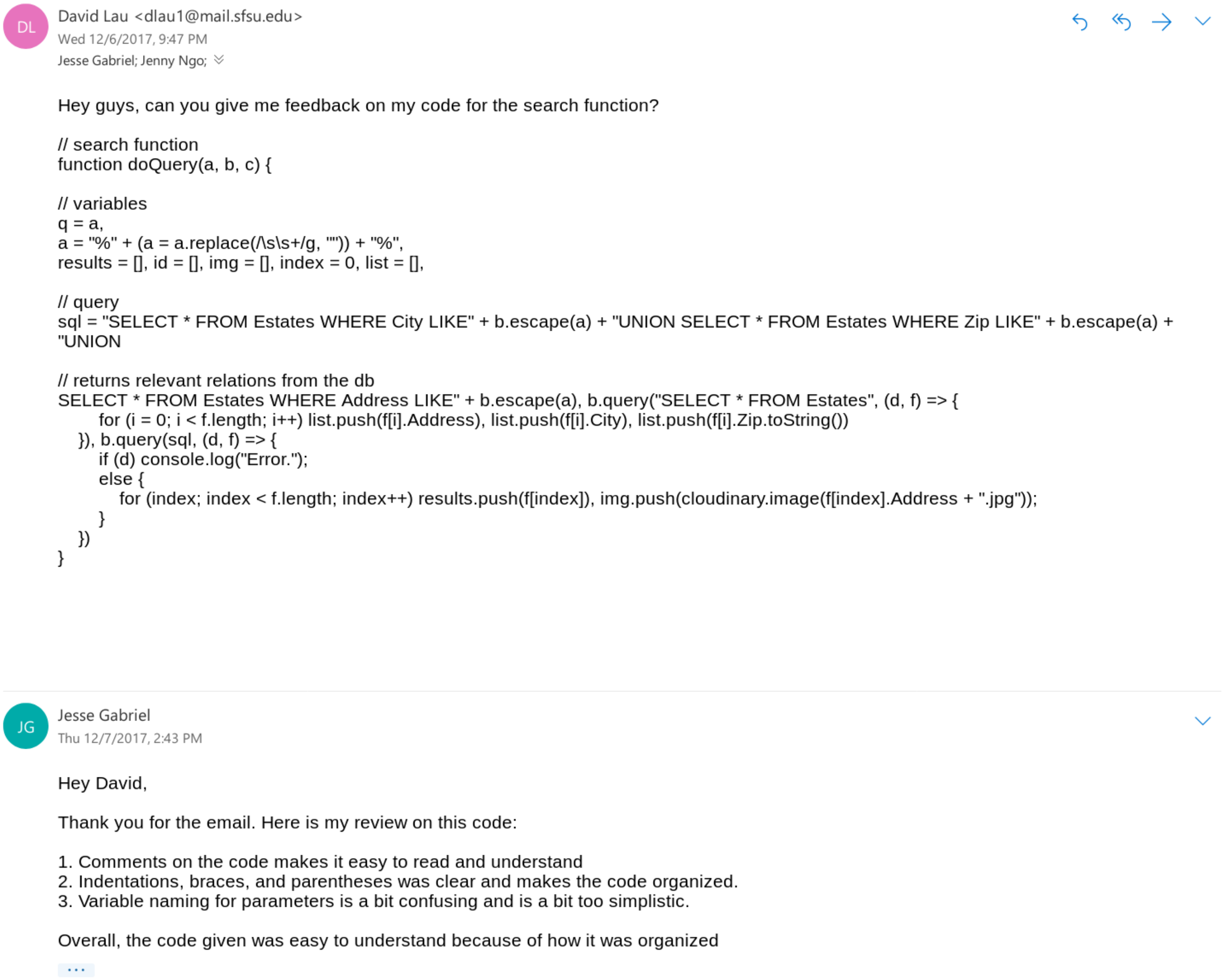
The chart below shows the following testing to be done for the search function.

- Number of Open Bugs at the Time: Informs how many bugs were found during this test.
- Description: Informs what and how exactly to test the search function.
- Test Input: Informs what to input for that test.
- Expected Output: For each test input, there is an expected output.
- Pass/Fail: The test is a pass if the input given results in the expected output, otherwise it is a fail.

Number of Open Bugs at the Time	Description	Test Input	Expected Output	Pass/Fail
	Setup: Open the following link in browser: https://www.sfsuse.com/fa17g03/ Description:	Search Category: All Search box: San Diego Click button: Magnifying Glass Or press Enter	2 Results for "San Diego"	Pass / Fail

	Users shall be able to search for listing by using “All” filter category.			
	Setup: Open the following link in browser: https://www.sfsuse.com/fa17g03/ Description: Users shall be able to search for listing by using “All” filter category.	Search Category: All Search box: 94116 Click button: Magnifying Glass Or press Enter	1 Results for "94116"	Pass / Fail
	Setup: Open the following link in browser: https://www.sfsuse.com/fa17g03/ Description: User shall be able filter by lowest to highest price	Enter Search box: San Francisco Filter: Price Select: Low to High	Changes order of listings from lowest to highest price.	Pass / Fail
	Setup: Open the following link in browser: https://www.sfsuse.com/fa17g03/ Description: User shall be able filter by bedrooms	Enter Search box: San Francisco Filter: Bedrooms Select: 2+	All 5 Listings from San Francisco	Pass / Fail
	Setup: Open the following link in browser: https://www.sfsuse.com/fa17g03/ Description: User shall be able filter by bathrooms	Enter Search box: San Francisco Filter: Bathrooms Select: 2+	2 Listings	Pass / Fail

4) Code Review



David Lau <dlau1@mail.sfsu.edu>
Wed 12/6/2017, 9:47 PM
Jesse Gabriel; Jenny Ngo;

Hey guys, can you give me feedback on my code for the search function?

```
// search function
function doQuery(a, b, c) {

// variables
q = a,
a = "%" + (a = a.replace(/\s\s+/g, "")) + "%",
results = [], id = [], img = [], index = 0, list = [],

// query
sql = "SELECT * FROM Estates WHERE City LIKE" + b.escape(a) + "UNION SELECT * FROM Estates WHERE Zip LIKE" + b.escape(a) +
"UNION

// returns relevant relations from the db
SELECT * FROM Estates WHERE Address LIKE" + b.escape(a), b.query("SELECT * FROM Estates", (d, f) => {
        for (i = 0; i < f.length; i++) list.push(f[i].Address), list.push(f[i].City), list.push(f[i].Zip.toString())
    }), b.query(sql, (d, f) => {
        if (d) console.log("Error.");
        else {
            for (index; index < f.length; index++) results.push(f[index]), img.push(cloudinary.image(f[index].Address + ".jpg"));
        }
    })
}
```

Jesse Gabriel
Thu 12/7/2017, 2:43 PM

Hey David,

Thank you for the email. Here is my review on this code:

1. Comments on the code makes it easy to read and understand
2. Indentations, braces, and parentheses was clear and makes the code organized.
3. Variable naming for parameters is a bit confusing and is a bit too simplistic.

Overall, the code given was easy to understand because of how it was organized

5) Self-check on best practices for security

- Major assets we are protecting:
    - User information: Only an administrator has access to Buyer/Seller account info though anyone can see an Agent’s email address on the site. Registration requires an email, a password, first and last name along with agreeing to the Terms of Service.
- Password protection
    - The passwords are stored in the database after they are encrypted. Once stored in the database, passwords cannot be seen by anyone, even by the administrator.
    - When a user logs in the password characters are replaced by dots.
- Input Form Validation
    - Register Validation: User must pass Google captcha to register after filling out all the correct fields and agreeing to the Terms of Service.
    - Search Bar Input validation: maximum of 20 characters
    - Search Results Validation: Provides valid search results for the correct input if data is in the database.
    - Empty search bar will display all results
    - SQL Form injection prevention

6) Self-check: Adherence to original Non-functional specs

1. Application shall be developed and deployed using class provided deployment stack. DONE
2. Application shall be developed using pre-approved set of SW development and collaborative tools provided in the class. Any other tools or frameworks must be explicitly approved by Anthony Souza on a case by case basis. DONE
3. Application shall be hosted and deployed on Web Services by Google. DONE
4. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed. DONE
5. Data shall be stored in the MySQL database on the class server in the team's account. DONE
6. Application shall provide real-estate images and optionally video. DONE
7. Maps showing real-estate location shall be required. DONE
8. Application shall be deployed from the team's account on the Google Web Service. DONE
9. No more than 50 concurrent users shall be accessing the application at any time. DONE
10. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. DONE
11. The language used shall be English. DONE
12. Application shall be very easy to use and intuitive. No prior training shall be required to use the website. DONE
13. Google analytics shall be added. DONE
14. Messaging between users shall be done only by class approved methods and not via e-mail clients in order to avoid issues of security with e-mail services. DONE
15. Pay functionality (how to pay for goods and services) shall not be implemented. DONE
16. Site security: basic best practices shall be applied (as covered in the class). DONE
17. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. DONE
18. The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Fall 2017. For Demonstration Only”. (Important so as to not confuse this with a real application). DONE