SW Engineering CSC648/848

Section 01 Fall 2017

Project RECycle

Team 13

Jeremy Rodgers (jrodgers@mail.sfsu.edu)
Sabiha Barlaskar
Kachi Lau
Risha Shah
Parker Gray

Milestone 4

12/1/2017

Date	Revision
12/1/2017	V1.0 (initial draft)

1.) Product Summary

Our product is RECycle (a name that embodies the "**R**eal **E**state Cycle" of buying and selling properties). RECycle is as the name implies a real estate website in which users can put up property for sale as well as browse and purchase property through our communication channels.

Confirmed Features

Our product has all the basic features:

Search by city, zipcode and streetname
Feature listing on the home page
Location of the listing on maps
Sorting
Sign-in and Sign-up
Contacting the seller using messaging
Uploading of house details
Seller dashboard

Accessible URL
https://sfsuse.com/fa17g13/

2.) Usability Test Plan

Test objectives:

Test the degree of usability of the system according to the 3 usability metrics:

- Effectiveness
- Efficiency
- Satisfaction

System setup: Using Windows 10 open up the Firefox or Chrome browser and proceed to this URL: https://sfsuse.com/fa17g13/

Starting point: List URL that contains the test subject/any other particularly relevant things for the test like the user being logged in.

Task to be accomplished:

Focus group testing(the backend reviews the work of the front end and vice versa), by allowing the users to test all the paths of a task, without telling them how to do it.

Intended user:

A novice user

Completion criteria:

The focus groups decide whether a feature passes or fails the test of usability. If it fails, appropriate measures to be taken to correct the design and enhance its usability.

URL of System to be Tested:

https://sfsuse.com/fa17g13/

Likert Scale

	Strongly Agree	Agree	Disagree	Strongly Agree
The feature being tested performed as expected.	1	2	3	4
The website is pleasant to look at.	1	2	3	4

I didn’t encounter any major bugs unrelated to the feature being tested while testing the feature.	1	2	3	4
Easy navigation throughout the website.	1	2	3	4
Easy to use features.	1	2	3	4

3.) QA Test Plan

Test objectives:

1) Classical QA to check if the functional requirements are met
2) QA for bugs
3) QA for mobile platforms
4) QA for security
5) QA for cloud based services(In our case, checking if the website is correctly deployed on the amazon server)

Hardware and Software Setup:

Hardware:

Website served on amazon server with the browsers on desktops and mobile phones acting as clients

Software:

1) Mysql as the database management system
2) Node.js for the backend
3) Bootstrap incorporated with jade for the frontend design

Feature to be tested:

1) Search is functional and effective
2) Secure login with encrypted password
3) Sort feature functions well
4) Seller dashboard is up to date and and has good usability

Actual test cases:

Test #	**Test Title**	**Test Description**	**Test Input**	**Expected Output**	**Test Results**
1	Search for properties using zip code	User selects the zip code option from the dropdown list, and enters a 5-digit zip code in the search bar and hits the search button	94539, 95112,94501	**94539**: 3 results with the city name Fremont **95112**: 1 result with	Passed

				the city name San Jose **94501**: 1 result with the city name Alameda	
2	Zip code should only be a number and its length should be 5	User selects the zip code option from the dropdown list, and enters a 5-digit zip code in the search bar and hits the search button	**<u>945818</u>** **<u>a1</u>**	**945818** Alert box, asking user to input zipcode of length 5 **a1** Please input numeric characters only	Passed
3	Search bar shouldn't be left blank	User inputs nothing	" "	Value must be filled	Passed
4	If name of the city not present, should display an error message	User selects city option from the drop down list and enters a name and hits the search button	Austin, New York	"Sorry Not Found! Please try again"	Passed But should have an alert box
5	Sort results by Price, number of bedrooms and square feet	User enters name of the city the search bar by selecting the city option and hits search button. User selects one the option from the dropdown filter menu on the left	Fremont, Filter by Price	The results should show the property with the least price on the top and the highest price in the bottom	Passed
6	Input	User clicks on the	Leave the	Alert box	Passed

	validation on the login fields	“Agent Login” link on the homepage. This will take the user to the login page	username and password fields blank	asking user to fill out the field names	
7	Login and password match	User clicks on the “Agent Login” link on the homepage	Enter the user name “Admin” password “12345”	Username and password doesn’t match	Passed
8	Sending Message	A user who is interested in contacting the agent, will send message to the agent by clicking on the property.	Name: “John” Email: john.danny@xyz.com Phone: 122344555 Message: Hi, I’m interested in this property	Message sent successfully	Passed
9	Seller dashboard	Agents can log in and upload details of the house and find information of the house he has uploaded	Login page Enter Username: “admin” Password: admin	Takes the agent to his dashboard page	Passed
10	Viewing of messages	Agents can log in and upload details of the house and view the messages that the buyers have sent them along with their name and email	Login page Enter Username: “admin” Password: Admin	Clicking on the messages tab, the agent is taken to the page where messages have been displayed	Passed Indentation needs to be checked

4.)Code Review

Coding style

The team chose to follow snake casing in order to make the code consistent and readable. We further incorporated a secure style of coding for which emphasis was laid on checking unbound arrays, input validation and encrypted passwords. We also made sure we read the code written by the other team members, as done in white box testing, in order to make sure that it adheres to the conventions followed by the team.

Review of display of messages in the dashboard

```
body
  // Background
  #background
  // Header
  #header
  | &#x9;&#x9;
  .container-fluid
   .content
   #message
    ul.nav.nav-pills.nav-stacked.col-md-2
     li.active
      a(href='#tab_a', data-toggle='pill') New Mail
     | &#x9;&#x9;&#x9;
     li
      a(href='#tab_b', data-toggle='pill') Send a Message
    | &#x9;&#x9;&#x9;
    .tab-content.col-md-10
     // Send Mail Tab
     #tab_a.tab-pane.active
      .panel.panel-primary(style='width: 100%;')
     .panel-heading
         h3.panel-title New Mail
        | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
        .panel-body
```

```
          .form-group
           table#NewMails.dashboard
            tr.header
             th(style='width:40%;') From   //Increase the width of the header so that all the fields
are equally distributed
             th(style='width:50%;') Message
             |
             th(style='width:30%;') Subject
             |
             th(style='width:20%;') Date
             |
             th(style='width:5px;')
            | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
            tr
             - each item in rows

               td #{item.bname}
               | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
               td #{item.messages}

td #{item.vphone}
               | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
               td 11/21/17
               | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
               td
                button#deleteButton.btn.btn-default.btn-sm(type='button')
                 span.glyphicon.glyphicon-minus

      // New Mail Tab
      #tab_b.tab-pane
       #tab_a.tab-pane.active    // Make this tab active when the page is loaded
        .panel.panel-primary(style='width: 60%;')
         .panel-heading
          h3.panel-title Send a mail
         | &#x9;&#x9;&#x9;&#x9;&#x9;
         .panel-body
          .form-group
           label(for='subject') Subject:
           | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
```

```
input#subject.form-control(style='width: 80%;', type='text')
| &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
Form

 .form-group
  label(for='message') Message:
  | &#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;&#x9;
  textarea#message.form-control(style='width: 80%;', rows='5')
```

Comments

1. Increase the width of the header so that all the fields are equally distributed
2. Make the “New Mail” tab active instead of the “Send Mail”

Peer review of the project as a whole

Peer review to the backend team:

1) Excellent use of functions and scripts, variable names and indentation.
2) The backend meets all the required functional specifications. Slight modifications in the design can make it more modular . For example, instead of rendering all the views from the index.js file, there can be a separate ‘controllers’ folder that can contain the javascript files for the different functions. The db.js file can be in a models.js folder.

Peer review to the frontend team

1) Can have more comments to make the code self explanatory. For eg, comments can used to explicitly mark the start of a element. Especially because the frontend is rendered through jade files, it is difficult for novice readers to figure out the start and end of elements, making commenting imperative.

5.) Self-check on Best Practices for Security

Assets being protected: User emails, passwords, mobile numbers, First names, Last names, locations, IP Addresses.

Passwords **are** encrypted within the database.

User input for search queries **is** validated:

```
//code
function validateForm() {  //Input field shouldn't be left empty
                var x = document.getElementById("search_input").value;
                console.log(x);
                if (x == "") {
                  alert("Value must be filled out");
                  return false;
                }
               }

function allnumeric(inputtxt) //Checks the input in the zipcode field should be numbers only and length should be 5
               {
                var numbers = /^[0-9]*$/;
                if (inputtxt == "")
                {
                  return;
                }
                if(!inputtxt.match(numbers))
                {
                  alert('Please input numeric characters only');

                }
                if(inputtxt.length > 5)
                {
                alert('Please input a zipcode of 5 characters');
                }
                }
function alltext(inputtxt) //Checks the input in the city field should be lettters only
```

```
{
 var letters = /^[A-Za-z]+$/;
 inputtxt = inputtxt.replace(" ",""); // Strips off space in the input
if (inputtxt == "")
   {
  return;
}
if(!inputtxt.match(letters))

{
  alert('Please input letters only');


 }

}
```

6.) Self-check: Adherence to Original Functional Specs

1. The buyer shall be able to search the website for a property using zipcode, name of the city **DONE**
2. The buyer shall be able to filter the search details by price, the area of the property, the number of rooms and the year of construction **DONE**
3. The buyer shall be able to register and create an account for himself/herself **DONE**
4. The buyer shall be able to contact a customer service representative through interactive Messaging **DONE**
5. The buyer shall be able to send message or email to the seller/real estate agent **DONE**
6. The buyer shall be able to see a list of featured properties in the home page **DONE**
7. The buyer shall be able to rate a seller **NOT DONE (P3)**
8. The buyer shall be able to edit his/her profile **NOT DONE(P3)**
9. The seller shall be able to register and create an account **DONE**
10. The seller shall be able to add/delete/modify details of the property **DONE**
11. The seller shall be able to edit his/her profile **NOT DONE(P3)**
12. The seller shall be able to contact a customer service representative through interactive messaging **NOT DONE(P3)**
13. The admin shall be able to add/delete/modify the details of property **DONE**
14. The admin shall be able to suspend a seller/buyer if needed **NOT DONE(P3)**
15. The admin shall be able to respond a seller/buyer **NOT DONE(P3)**
16. The admin shall be able to contact a seller/buyer by message **NOT DONE (P3)**

7.) 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 Amazon Web Services as specified in the class.**DONE**
4. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.**ON TRACK**
5. 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**
6. Data shall be stored in the MySQL database on the class server in the team's account.**DONE**
7. Application shall provide real-estate images and optionally video.**DONE**
8. Maps showing real-estate location shall be required.**DONE**
9. Application shall be deployed from the team's account on AWS.**DONE**
10. No more than 50 concurrent users shall be accessing the application at any time.**DONE**
11. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.**ON TRACK**
12. The language used shall be English.**DONE**
13. Application shall be very easy to use and intuitive. No prior training shall be required to use the website.**DONE**
14. Google analytics shall be added.**ON TRACK**
15. 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**
16. Pay functionality (how to pay for goods and services) shall not be implemented.**DONE**
17. Site security: basic best practices shall be applied (as covered in the class)**DONE**
18. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.**DONE**

19. 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).**ON TRACK**