SW Engineering CSC648/848 Fall 2018

Project Title: Aquirium

Team Number: 08

Names of Students:

Jianfei Zhao (jzhao11@mail.sfsu.edu) (Team Lead)
Feras Alazzeh (Back-end Lead)
Tommy Lik (Front-end Lead)
Lileana Wright
Edward Barajas
Alex Li
Jiawei Xu

Milestone: 1

Date: 10/13/2018

History Table:

Date	Revision
10/02/2018	Created as the 1st version
10/13/2018	Revised according to CEO's feedback

I. Executive Summary

Ever wanted a more convenient marketplace at your institution? Well, our web application ***Aquirium*** allows for on-campus buying and selling of items at anyone's convenience. Typically, college students struggle financially on their day to day lives. However, ***Aquirium*** will give them the opportunity to sell whatever they want for some extra cash. This web application will allow students to effortlessly sell products online or buy products, with the majority of customers being either nearby residents or students dorming at San Francisco State. Our web application serves as a primary marketplace for students at any institution, in which the marketplace can be utilized by different institutions across the world and thus, allowing all students this opportunity to buy and sell.

The ***Aquirium*** web application will primarily focus on students and nearby residents of San Francisco State University in which they can buy and sell products online easily. This application is important because not only is it convenient, but it also creates opportunities for college students who are extremely tight on budgeting and just need a little extra cash, or it gives students the opportunity to buy items without the need to wait for shipping and processing. Rather, students will be able to buy products and have the option of picking it up from the seller, or having it shipped to them at their own convenience.

As team 08, we are hardworking and dedicated programmers interested in helping our community by providing this interactive buying/selling web application to the general public. With 4 front-end and 3 back-end engineers, we seek to create a friendly user environment by providing all users with a simplistic yet efficient user interface in which they'll be able to easily utilize multiple functionalities created by the back-end engineers.

II. Personas and Use Cases

i.) Personas

1). Unregistered User
Jason is a transfer student and is beginning his studies at San Francisco State University. He is a computer science major and is very nervous to start the school year because he doesn't know what how college life is at San Francisco State. Currently, he is planning on moving into the dorms and is trying to find items for his dorm room. However, he does not feel ready to decide on what to purchase. He has not used too many websites to purchase items and has never had to buy items for a dorm room before. As a result, he simply wants to see what different sites have to offer college students.

2). Registered User

Adam is a senior at San Francisco State University. He is an art major. It is the middle of the semester and Adam needs some new supplies that he can use for school. He needs to find these items quickly since his art classes have begun. Since he is a senior, he does not have that many classes left and will graduate at the end of the semester and leave San Francisco. He has many items other college students may want, that he will no longer need. He wants to get rid of these items quickly. He uses multiple different web applications so he has a good idea as to how to navigate and buy items online. However, he already has many different accounts and does not like the idea of having to make another account and keep track of it if it is not absolutely needed.

3). Admin User
Susan is a software engineer at ***Aquirium***. She is an admin user that has the permission to approve or remove requests. Susan ensures that there is no inappropriate data so that the users have a good experience on ***Aquirium***. Susan gets notified when posts need to be approved and she is busy every other weekday maintaining the site.

ii). Use Cases

1). Browsing and Purchasing Items

On our site, Jason looks through the diverse list of categories. By choosing the "furniture" category, he narrows down the results and finds the dorm items that he is looking for. After browsing furniture, he realizes that he still needs to get his school supplies such as books. He has never taken courses at SFSU so he is not comfortable with the course numbers for the corresponding course topics. He wants to search for books relating to his courses, so he switches into the "book" category, and types "Computer Science" in the text search bar. Through the review of search results, he finds the book that he eagerly needs. When he decides to contact the seller for more details, Jason is prompted to register or log in. Since Jason is an unregistered user, he goes to registration.

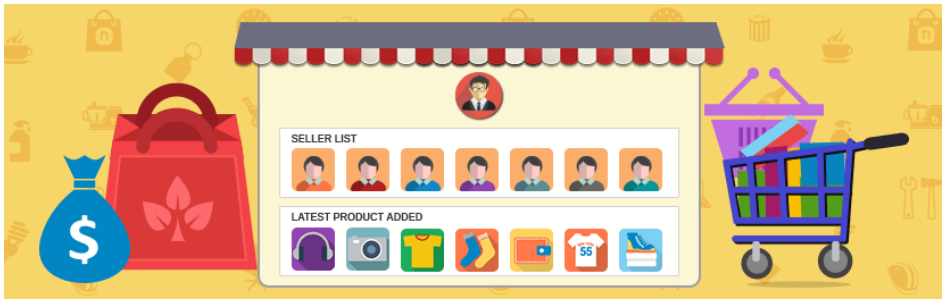
2). Posting Items

Adam is almost finished with school and will graduate at the end of the semester. However, he has several pieces of furniture that are too large to bring with, including a table, a queen-size mattress, and two chairs. Moreover, he has books that are no longer needed. Thus, he decides to sell all these items to new incoming SFSU students. Then he chooses ***Aquirium***, the website specifically designed for SFSU. After filling in the details of his mattress and clicking "post", he is prompted to either register or log in. Since Adam has already registered before, he directly logs in his account and finishes posting those items.

3). Checking Dashboard

Several items have been posted by Adam. 3 days later, he logs back in his ***Aquirium***'s account. This is the first time he posts things online, so he is curious about the status of the items he tries to sell. When he goes to the seller dashboard to check the status of his posts, he finds that all of them have been approved and displayed on the website. Besides, in his dashboard, he also receives in-site messages from some potential buyers who contact him for purchasing his table and chairs. He happily replies these messages through in-site messaging. Having browsing other mattresses on ***Aquirium***, Adam feels that the price he set for his mattress is too high for students. Therefore, he goes back to the dashboard and edits his mattress by lowering the price.

4). Monitoring Posts and Users



Susan is an admin user at ***Aquirium*** that has the permission to approve requests. Susan ensures that there is no inappropriate data so that the users have a good experience on ***Aquirium***. She is busy every other weekday maintaining the site. One day Susan receives notifications of two new posts from sellers that need to be approved. She goes on to the admin dashboard to review the posts. Susan feels that one of them meets our site's guideline and then approve the content to go live with a simple click. However, she finds that the other post contains some illegal information, and therefore she removes (rejects) that post request and deletes that user.

III. Data Definitions

1). **User**

- The users will utilize our web application to purchase items they need, or post items they want to sell to others. Basically, there are 3 types of users, including unregistered, registered, and admin users. Below are some basic components for each user.
- *username* (unique username in string format)
- *password* (encrypted for safety)
- *email* (email address ended with "@mail.sfsu.edu" to be validated as an sfsu user)

2). **Priority**

- It describes the permission level for users. Unregistered users can only browse and search items on our website. Registered users have all the privileges that unregistered users have, plus the functions of contacting sellers, posting items, etc. Admin users can also delete illegal items and users with inappropriate behavior.

3). **Item**

- Items refer to the goods or products which users will post or purchase on our website. Each item has a title, together with a short paragraph of description to introduce the item. Price will be used for sorting the items in both ascending and descending order. Below are some basic components for each item.

- *category* (category to which the item belongs)
- *title* (title of the item posted on website)
- *description* (short paragraph of introduction and related details for the item)
- *price* (numeric value, also used for sorting)
- *title_img* (title picture/image to show the item)

4). **Category**

- Multiple categories are used to classify all the items into specific types. Users may browse the list of all items at first and then narrow down the results by using different categories. Below are some basic components for each category.
- *id* (unique numeric value for each category)
- *title* (title/tag of the category)

5). **Img**

- To provide users with intuitive impression of the items, a title image will be shown for each item. If a user clicks the title image, s/he will be redirected to another web page, with several other images to display the details of that item.

6). **Status**

- It describes whether the item is approved by admin. For each post request, it will be marked as "pending" automatically and will not be shown on the website immediately. It will then become "approved" and go live after it is approved by an admin user.

7). **Message**

- In-site messaging system is designed for users to communicate with each other.If a user is interested in an item, s/he may ask the seller for more details. Registered users can send messages to the sellers via in-site web form, and the sellers can also reply to them. Below are some basic components for each message.
- *from_user* (user who sends the message)
- *to_user* (user who receives the message)
- *content* (content of the message)

IV. Initial List of Functional Requirements

The requirements for unregistered users are 1) ~ 6).

1). Unregistered users shall be able to browse the items on the website without being registered.

2). Unregistered users shall be able to search the items according to their categories, such as furniture, clothes, electronic device, etc.
3). Unregistered users shall be able to search the items by using text search bar. If there is matching, the number of results shall also be shown together.
4). Unregistered users shall be able to sort the list of items by price, in both ascending and descending orders.
5). Unregistered users shall be asked for registration only when they decide to contact the seller or post an item.
6). During registration, the checkboxes for terms and conditions shall be unchecked by default. The registration form shall be successfully submitted only when all these boxes are checked and all the required fields (username, password, etc.) are filled in.

The requirements for registered users are 7) ~ 11), plus 1) ~ 6).
7). Registered users shall be able to contact sellers as well as post items.
8). Registered users shall be able to access the dashboard with the list of items they have posted.
9). Registered users shall have the privilege to edit the items they have posted, but Admin users shall not.
10). Registered users shall be able to send messages to sellers via web form (in-site messaging), but not via email.
11). Registered users shall be able to receive messages regarding any postings from inquiring buyers.

The requirements for admin users are 12) ~ 15), plus 1) ~ 6).
12). Admin users shall have the privilege to censor and delete inappropriate items.
13). Admin users shall have the privilege to delete users who post inappropriate items.
14). Admin users shall have the privilege to approve legal items before they are shown on the website.
15). Admin users shall have the privilege to add/edit search categories.

V. 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 all major browsers: Mozilla, Safari, Chrome.
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). Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
14). The website shall prominently display the following exact text on all pages "SFSU-Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

VI. Competitive Analysis

Below is a table for the comparison of several websites, with our web application ***Aquirium*** marked as the shaded column.

	Etsy	**eBay**	**Amazon**	**Bonanza**	**Aquirium**
Seller Services	+	++	+	+	+
Categories for Browsing & Filtering	++	++	++	+	+
Contact Seller	+	+	+	+	+
Online Payment & Mailing Service	+	+	++	+	-
Delivery Speed	+	+	+	+	++

Although lacking in online payment and mailing service, ***Aquirium***'s main focus is on a buy/sell service for SFSU students and their needs, which means two things:

- Transactions will more than likely be completed on campus and therefore the items are ready for pick-up on an agreed upon and convenient time.
- In-site messaging will be provided for buyers and sellers to communicate. For example, registered users will contact sellers for item details through in-site messaging, and sellers will then reply to them accordingly.

VII. High-Level System Architecture

1). Server Host:
   - Amazon Web Services (AWS) 1vCPU 1GB RAM

2). Operating System:
   - Microsoft Windows Server 2012 R2 Base

3). Database:
   - MySQL 5.7.21

4). Web Server:
   - Apache 2.4.33

5). Server-Side Language:
   - PHP

6). Additional Technologies:
   - Web Framework: Laravel
   - Web Analytics: Google Analytics
   - Front-end: Bootstrap
   - Back-end IDE: NetBeans

7). Supported Browser:
   - Google Chrome and Mozilla Firefox (latest 2 versions for each)

VIII. Team

1). Jianfei Zhao:
   - Team Lead & Back-end Engineer

2). Feras Alazzeh:
   - Back-end Lead

3). Tommy Lik:

- Front-end Lead

4). Lileana Wright:
- Front-end Engineer

5). Edward Barajas:
- Front-end Engineer

6). Alex Li:
- Front-end Engineer

7). Jiawei Xu:
- Back-end Engineer

IX. Checklist

1). Team found a time slot to meet:
- **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 and working on it:
- **DONE**

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