SW Engineering CSC 648 Fall 2019

Section 2

Team 8

October 3, 2019

Milestone 1

Team:

- Oleksii Butakov -- Team Lead / Front-end Lead (obutakov@mail.sfsu.edu)
- Andrew Copas ------ GitHub Master / Front-end
- Alex Lee ---------- Front-end
- Emanuel Saunders ---------- Back-end Lead
- Kevin Truong ---------- Back-end
- Tim Lei ---------- Back-end

History Table:
10/03/19 - Initial submission
10/04/19 - Professor's feedback received
10/07/19 - Document revised: more use cases has been added, live-chat feature removed from the executive summary, formatting.

Content

I. Executive Summary

Our service (SFSU Quad) is a marketplace exclusively for SFSU students, where every registered SFSU student can buy and sell goods.

SFSU Quad is a unique place that benefits every student who wants to buy/sell items quickly and locally, as well as help build a sense of community among SFSU students. SFSU Quad would have a basic functionality that you would normally expect from a typical classified advertisement website, such as: search, post, and filter functions., plus social-media tool such as SFSU's Twitter timefeed that would attract more users, make users spend more time on our website and make a website more entertaining.

Back in the day, people used to gather in their local squares or next to the city halls, where they could share their thoughts and ideas with their community. Today, all of the communications are happening in a private chats or groups and we start to lose that sense of community and belongingness that we used to have during social gatherings. SFSU Quad is here to put this missing puzzle back in our lives. It wouldn't only be a place where students can effectively buy and sell goods, but rather a virtual SFSU quad where students can hang out while learning what's for sale locally. The mission of SFSU Quad is to break the walls between students and provide them with an effective tool for buying and selling goods.

Our engineering team of SFSU computer science students use a solid, well-supported development framework for SFSUQuad.com that is proven to be extremely reliable and scalable to virtually any size. Simple, yet sophisticated structure of application makes it possible to deploy our service not only on SFSU campus, but on any university campus in the US.

II. Personas and Main Use Cases

For the application, we want to create a buy and sell website for SFSU students. Students will post items on their account on the site, and students will buy respective items on the site from other students of SFSU. This will use delayed registration so the student can view the items or consult the SFSU student seller.

Jason is an SFSU student and part time worker.

About Jason:

- Busy
- Works and studies during the daytime from 9-5.
- Wants to utilize the purchasing of most pristine conditioned items from other students mainly due to the low budget costs compared to other shopping methods.

Jason's goals:

- Wants to buy a tablet which another student is selling.
- His schedule permits him to only pick up the item or meet the seller during the nighttime
- To ensure the safest interaction and transaction, he would like to meet up in public, and lighted locations such as the library.
- Wants to see a demo of the product before deciding on purchase to ensure satisfaction.

User skills: Is skilled at using WWW apps and using WWW platforms.

Usage: Online

Procedure: go to site or application, browse site, use the search menu to find items, browse items, select items of what the user will find suitable, buy, register, and set meeting time with seller.

Difficulties: Finding proper times to meet regarding online item at certain times outside of studying and work. Might need to learn how to use the application if new

to interface. Application is online-based and would require an internet connection, which can be difficult depending on where the user is.

User satisfaction: slightly confused at first, and might tire at times. Is skilled at registering credentials in order to use WWW apps. Is skilled at entering the specific credentials to register as a user for the site, and sending messages to seller regarding the specific item.

Admin use cases:

- Admin approves posted items before they are officially online.
- Admin confirms the registered users.
- Admin moderates comments of registered users regarding items online at times.
- Takes action against unusual user activity which can include the deletion of certain user accounts.
- Approves certain locations on campus that are deemed safe before locations are posted for users.

User cases:

- Decides to consider purchasing a certain item from an SFSU seller. Checks either from home or work for a safe location on campus.
- Should the user have doubts about the security of a location, user would want to report it but doesn't have time.
- To ensure satisfaction, user might contact seller for more information on item, and possibly requesting seller for a live demo of item.

III. List of main data items and entities

Unregistered Users:

Unregistered users are viewers of the website. They can browse the website, but if they want to purchase an item, sell an item or comment on an item they have to become registered users.

Registered users:

- Name
- SFSU email
- Password
- Nickname

Registered users are people who created an account on the website. Registered users have to provide their name and SFSU e-mail in order to register. Once they are registered they can buy or sell items on the website. The buyers of registered users can comment on the items they bought.

Admin:

- Name
- E-mail
- Password

Admin login with their email and password. Admin has the highest privilege. Admin approve/reject selling posts and manages account of which admin can delete accounts.

Posting:

- Name of item
- Picture(s) of item
- Description of item
- Price of item
- Seller's contact information
- Category of item
- Pickup locations

A post of an item for sale. The post has name, picture(s), description, category and price of item, pickup locations and seller's contact information.

IV. Initial list of functional requirements

Unregistered Users:

1. Unregistered users shall be able to search.
2. Unregistered users shall be able to browse different categories.
3. Unregistered users shall be able to filter items.
4. Unregistered users shall be able to access posts.
5. Unregistered users shall be able to register.
6. Unregistered users shall be able to sort items.

Registered Users:

7. +Functions of unregistered users.
8. Registered users shall be able to login.
9. Registered users shall be able to post.
10. Registered users shall be able to make modifications on their selling items.
11. Registered users shall be able to request take down of their selling items.

12. Registered users shall be able to buy items.
13. Registered users shall be able to submit their buy requests.
14. Registered users shall be able to comment.
15. Registered users shall be able to buy many items at once.
16. Registered users shall be able to contact the seller.
17. Registered users shall be able to choose pickup locations.

Admin:

18. +Functions of registered users.
19. Admin shall be able to reject selling request.
20. Admin shall be able to accept selling request.
21. Admin shall be able to approve posts.
22. Admin shall be able to delete accounts.
23. Admin shall be able to take down selling posts.

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 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 email clients shall be allowed
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
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 Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

VI. Competitive analysis

	Amazon	eBay	Craigslist	SFSU Quad
Localized	-	-	+	++
Customer Security	+	-	-	++
Messaging	-	-	+	+
Buyer/Seller fee	-	-	+	+

Our website offers localization, unlike both Amazon and eBay. All of our Users shall register using an SFSU email ensuring that they are a part of the SFSU community. While you can search by region on Craigslist, we offer a product that specifically services those at SFSU. As for customer security, Amazon now offers lockers for their users which removes the possibility of packages being stolen unlike eBay. This offers their users some security, while craigslist offers no security or protection for their users which has led to many acts of criminal activity in the past. SFSU Quad offers the most security for our users by providing designated meetup locations for our users that are safe and reliable. When it comes to messaging back and forth between buyers and sellers, only Craigslist offers a pre-purchase way to contact them. They do this by implementing an internal

emailing client. We also offer a pre-purchase messaging feature that allows the buyer and seller to negotiate price and decide on a time to meet. Finally, our Web App is completely free for our users, both buyer and seller. Amazon and eBay push a fee upon both the buyer and seller.

VII. High-level system architecture and technologies used

Server Host: Google Compute Engine 1vCPU 1.7 GB RAM
Operating System: Ubuntu 18.04 LTS
Database: MySQL Community Server 8.0.2
Web Server: Apache HTTP Server 2.4.41
Server-Side Language: Python
Additional Technologies:
Web Framework: Flask, Bootstrap v4.3.1
IDE: PyCharm
Web Analytics: Google Analytics

VIII. Team and roles

- Oleksii Butakov - Team Lead / Front-end Lead (obutakov@mail.sfsu.edu)
- Andrew Copas - GitHub Master / Front-end
- Alex Lee - Front-end
- Emanuel Saunders - Back-end Lead
- Kevin Truong - Back-end
- Tim Lei - Back-end

XI. Checklist

· Team found a time slot to meet outside of the class - **DONE**

· Github master chosen - **DONE**

· Team decided and agreed together on using the listed SW tools and deployment server - **DONE**

· 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**

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

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