Software Engineering CSC 648/848
Section 01

SFSU Congre-Gators Market
Team 04

Ihsan Taha (ihsan@mail.sfsu.edu)
Prateek Gupta
Darel Ogbonna
Shane Cota
Kevin Chu
Mark Tompong

Milestone 1
28th, February 2017

History table

Initial draft Milestone 1	**V1.0**

1) Executive Summary

“SFSU Congre-Gators Market” is a buy/sell web application exclusively for SFSU students that is beneficial, safe, and of great value to the SFSU community. Users can create item listings on the website to sell various types of items including books, electronics, furniture, and more that are of interest to students. The application consists of two general types of users: non-registered and registered. Both of these users can browse the website for interesting items, and the latter can purchase items posted from other registered users or sell items of their own. Registered users may contact each other through messaging provided by our web application while transactions take place outside of the application after an agreement. Moreover, the website shall have at least one admin whose job shall be to monitor postings and messages to ensure a safe and legitimate environment.

Our web application provides multiple advantages such as items directly relevant to SFSU students, a common meeting ground at the SFSU campus or nearby, and the security that comes with each buyer or seller being a verified SFSU student. Students can find products relevant to their needs by finding items posted from other students who took similar courses. There is no hassle waiting on a shipment since all members are also students. Hence, they are local to the area and can meet on campus. Furthermore, being on campus naturally discourages members from engaging in misconduct. Even if a transaction were to take place off campus, students are less likely to sell a false advertisement when they have to attend to their classes.

Our product is unique compared to competitors. Registered users may post multiple images instead of a single image to better advertise their item. A single image may be insufficient for buyers or sellers if the product is not visible from a different angle or if comparison images are unavailable. Without multiple images, the consumer might need to contact the seller for details, but they may not have the time to go through the process because they are busy with assignments and tests. Our multi-image upload feature eliminates this problem by giving the buyer a better view of the product and building confidence in a possible purchase.

We are Team 4 of CSC 648-848 Section 1 - Software Engineering at SFSU, Spring 2017. Our mission is to experience the real world and what it means to build a startup company by developing a fully functional buy/sell web application that is also promotable as a legitimate product. Not only do we desire to create a great application, but we do so as a team that appreciates the open-mindedness of its members and uses real world techniques like Agile and Continuous Software Development.

2) Use Cases

Non-registered User: Jane is in the market for a new laptop after her old laptop, which she owned for several years, recently crashed mid-semester. As a student, a new laptop costs more than Jane can afford, so she is looking for a used laptop. Jane has heard about the student web-service that links students who want to sell items, including electronics. She visits the website and navigates to the electronic section and, with the use of filters, quickly finds an **item listing** with the item she is looking for at an affordable price. Jane decides to contact the seller, but since she is a **non-registered user,** she is only allowed to browse and cannot purchase (or post) items for sale. Jane is therefore prompted to register and verify her status as an SFSU student. She happily complied with the registration request and became a **registered user**. Her status as a registered user allows her to view the seller's contact information. Jane then contacted the seller and agreed to purchase the laptop.

Registered User (Buyer/Seller): Joe is a senior student at SFSU who lives on campus. He is planning on graduating this semester and wants to sell his dorm room furniture. However, Joe is still quite busy finishing his school work. Therefore, he needs an ideal way to offload his furniture to a fellow student while also allowing Joe to coordinate payments and move his items efficiently. Joe is already a **registered user** (since he bought his dorm furniture through the website when he was an incoming freshman) and thus has the freedom to log onto the website and post his furniture for sale. Joe creates a new **item listing** on the website for each of the furniture items he wishes to sell. He then waits for another **registered user**, who is interested in his items, to contact him to arrange a purchase.

Administrator: Jack, the IT guy, has been receiving complaints from several sellers about misconduct from a particular **registered user.** Apparently, the user has been sending **messages** which bluff interest in an item but result in non-payment or perpetual negotiations between the two parties. This user has accrued enough complaints from other registered users that Jack the IT guy confirms the issue by investigating the history of messages. Jack flags the offending user and warns him/her that further misconduct will result in a ban from the web service.

3) Data Definitions

1. **Non-registered User (nonreg_user):**
    1.1. Can browse on the website
    1.2. Can filter items on the website
    1.3. Can register on the website
    1.4. Does not need to be logged-in/registered
    1.5. Cannot post items for sale.
    1.6. Cannot access contact information of registered users.

2. **Registered User (reg_user):**
    2.1. Can browse on the website
    2.2. Can filter items on the website
    2.3. Can login on the website
    2.4. Can purchase items for sale
    2.5. Can post items for sale
    2.6. Can contact other registered users

3. **Administrator (admin):**
    3.1. Needs to login/register
    3.2. Can monitor messages
    3.3. Can remove items posted for sale
    3.4. Can contact registered users
    3.5. Can ban registered users.

4. **Item Listing (item_listing):**
    4.1. Price
    4.2. Description
    4.3. Keyword
    4.4. Photos (up to five)

5. **Message (message):**
    5.1. Title
    5.2. Text
    5.3. To (Recipient)
    5.4. From (Sender)
    5.5. Date/Time

4) Initial List of Functional Specs

1. The non-registered/registered user shall be able to browse the website for available items for sale.
2. The non-registered/registered user shall be able to filter the items for sale by ‘Category’.
3. The non-registered/registered user shall be able to filter the items for sale by ‘Price'.
4. The non-registered/registered user shall be able to filter the items for sale by ‘Date of posting’.
5. The non-registered user shall be able to register on the website to buy or sell.
6. The registered user shall be able to login to the website.
7. The registered user shall be able to post an item for sale.
8. The registered user shall be able to update his/her item for sale.
9. The registered user shall be able to remove his/her item for sale.
10. The registered user shall be able to view the items that were posted by him/her for sale.
11. The registered user shall be able to view contact information of other registered users.
12. The admin shall be able to remove an item listing.
13. Registration form: Allows users to register an account on the website. Registration requires a unique username, a password, and an SFSU email.

5) List of Non-Functional Specs

1. The application shall be developed using the LAMP stack provided by the class CTO.
2. The application shall be developed using an approved PHP Framework from the class CTO.
3. The application shall be developed using GUI technologies.
4. The application shall be hosted and deployed on Amazon Web Services provided by the Class CTO.
5. The application shall be optimized for standard desktop/laptop browsers.
6. The application shall render correctly on the two latest versions of at least three major browsers such as Mozilla, Safari, and Chrome.
7. The application shall have responsive UI code so it can be adequately rendered on mobile devices, but no mobile native app is to be deployed.
8. Data shall be stored in the MySQL database on the class server in the team's account.
9. The application shall be served from the team's account.
10. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
11. No more than 50 concurrent users shall be accessing the application at any time.
12. The language used shall be English.
13. The application shall be very easy to use and intuitive. No prior training shall be required to use the website.
14. Google Analytics shall be added.
15. Messaging between users shall be done only by class approved methods to avoid issues of security with e-mail services.
16. Pay functionality shall not be implemented on the website.
17. Site security shall be applied according to best practices.
18. Modern SE processes and practices shall be used including collaborative and continuous SW development.
19. The website shall prominently display the following text on all pages "SFSU Software Engineering Project, Spring 2017. For Demonstration Only."
20. The team shall use Github to maintain the website codebase.

6) Competitive Analysis

Registered users can post multiple images for each product to enhance the advertising quality of their product. Messaging is also available directly through the website for faster communication, and the admin can monitor any misconduct for a safe environment.

Competitive Analysis Graph of Features

	Our Application	Competitor A	Competitor B	Competitor C
Search	+	+	+	+
Filters	+	-	-	+
Uploading Photos	++	+	+	+
Messaging	++	-	-	+

7) High Level Architecture

1. **01-LAMP Stack:** (SF-SUSE Server Information)
    - 1.1. Host: Amazon Cloud
    - 1.2. OS: Ubuntu Server, Version: 16.04
    - 1.3. MySQL Version: 5.7
    - 1.4. PHP Version: 7.0.13
    - 1.5. OpenSSH Version: 7.2
    - 1.6. Git Version: 2.7.4
    - 1.7. Python: 2.7
    - 1.8. Ruby: 2.3.1
    - 1.9. nodejs: 4.2.6
    - 1.10. npm: 3.5.2
    - 1.11. Less: 481
    - 1.12. Sass: 3.4.23

2. **PHP Framework:**
    - 2.1. CodeIgniter: Version 3.1.3

3. **GUI Technologies:**
    - 3.1. Javascript: Version 1.7
    - 3.2. jQuery: Version 3.1.1
    - 3.3. Bootstrap: Version 4.0.0-alpha.6
    - 3.4. CSS 3

4. **Major Browsers:**
    - 4.1. Mozilla Firefox: Version 51.0.1 (64-bit)
    - 4.2. Safari: Version 9.0.2
    - 4.3. Google Chrome: Version 56.0.2924.87 (64-bit)

5. **Team's account where website will be served:**
    - 5.1. http://www.sfsuse.com/~sp17g04

6. **Team's application of choice where website contributions will be made:**
    - 6.1. Github

8) Team

CEO: Ihsan Taha
CTO: Prateek Gupta
Back-end Developer: Darel Ogbonna
Back-end Developer: Shane Cota
Front-end Developer: Kevin Chu
Front-end Developer: Mark Tompong

9) Checklist

- Team decided on basic means of communications: DONE
- Team found a time slot to meet outside of the class: DONE
- CTO chosen and working out well so far: DONE
- Github master chosen: DONE
- Team ready and able to use the chosen framework: DONE
- Skills of each team member defined and known to all: DONE
- Team lead ensured that all team members read the final M1 and agree/understand it before submission: DONE