

SW Engineering CSC648/848 Fall 2019
Gator Trader - SFSU Buy and Sell Website
Section 1
Team 3

Steve Rentschler - Team Lead (srentsch@mail.sfsu.edu)
Johnson Wong - Back-end Lead / GitHub Master
Sergei Katakhov - Front-end Lead
Ho Yin Mak (Tevis)
Tsun Ming Lee (Matthew)
Tim Chan
Osbaldo Martinez

Milestone 1
9/27/19

History Table

Date Submitted	9/30/19
Date Revised	10/8/19

Content and structure for Milestone 1 document for review

1. Executive Summary

For our project, we will be making a buy and sell website dedicated to students at San Francisco State University, called, "Gator Trader". There is a need for a user friendly website where students can safely and easily find what they are looking for and be able to sell things as well. This site will be a great way for students to look for a book for their class and sell the one they don't need anymore from the previous semester to their fellow students.

We plan on implementing all of the standard features one would find on a buy and sell site like Amazon or Craig's List like the ability to post an item for sale, search for items, having categories of items for sale, and the ability to message the seller. Some of the unique features that are not offered on other sites that we are planning on implementing are: The ability to safely find a place to engage with a seller for pick up of the item. A smart search that could help students easily navigate and find things they are looking for without using a nav bar. For example, you can set up a profile with your current classes which will be used by a smart search suggesting the items to consider. A spending/tracking analysis where students can track their spending of items. User background preferences where a user can change the theme or background of the website. Featured reviews on the homepage where a random review of an item will be featured on the main page of the website. We will have a quick and easy way to purchase items that you don't necessarily need to read all about and be able to stay on the search page to look at other items. We will have an automatic checklist with reminders that will check off an item once it is bought and notify the buyer what they will need for a class. In addition to these features, we plan on making the website as user friendly as possible, where things will be laid out to utilize space and where users will be able to instantly be able to do what they want to do without having to hunt for things.

Our team of students developers are well suited for this project. Three of us have experience of being Team leads, Back-end Leads, and Front-end leads. Our team lead has a background in art and design. Our Back-end Lead has a lot of knowledge of the software we will be using, with excellent problem solving abilities. And our Front-end lead was Team Lead, Front-end Lead, Back-end Lead on another project. We are dedicated to having tasks done on time with a polished look and feel.

2. Personae and Main Use Cases



About Diana:

- Uses her smartphone daily
- Hates slow websites
- Rarely uses laptop/desktop to browse the internet
- Lives with parents

Goals and scenario:

- She is trying to buy school supplies for the new term.
- She is trying to spend as little money as possible.



About Robert:

- He does most of his web surfing through his laptop
- He is a busy man
- Loves to read books

Goals and scenario:

- He is trying to buy books that he hasn't read before.
- He wants to save money by buying used books.

**About Mark:**

- Lives in a school dorm
- Takes great notes in class
- Buys things he might not be able to carry back home
- He loves technology

Goals and scenario:

- He wants a new video game, but lacks funds.
- He has stuff that he doesn't need anymore.

**About Alexander:**

- Follows guidelines to the letter
- He is strict
- Loves to help others

Goals and scenario:

- He wants everyone to follow guidelines
- He wants to help students sell their stuff.

Use Cases:

Use Case 1: Diana Buying

Diana uses her phone to navigate to the website. Upon loading the website she starts to browse through the items to see if something catches her eye. Remembering she needs a calculator, she searches for a calculator. After finding the right calculator for a reasonable price, she tries to buy the calculator and she is prompted to sign up/sign in. Before she buys the calculator, she messages the buyer where to meet up to before finishing the transaction.

Use case 2: Mark Selling

Mark is trying to make a quick buck to buy a new video game. He hears about our website and remembers he has some stuff that he doesn't need anymore, so he heads to our website to check how others have priced their items. He starts to browse through the categories and notices that there is a category for services and knowing that he is a good note taker plus he has beautiful handwriting he decides to make a listing for the digital version of his class notes of Calculus 3. He fills out the submission form for a new listing and after finishing filling it out and trying to submitted he is prompted to sign up/sign in.

Use Case 3: Robert Buying

Robert hears about our website. He knows he can use the website to buy used books from students, so he goes to the books category and browses the books . He finds a book he wants to buy and decides to buy it. He tries to buy the book and then he is asked to sign in/sign up.

Use Case 4: Admin Alexander

Alexander logs into his admin account for the website. He sees a bunch of items that still needs approval or rejection. He looks at the items to make sure they are appropriate. He doesn't see any problems with the first few items, so he approves them. He stumbles upon an item with blurry images, so he can't make out the item. He rejects item and lets user know why his item was denied. After some time, he sees a listing with an inappropriate image and he rejects the item.

3. List of Main Data Items and Entities

Unregistered Users:

Anybody who uses the site but did not log in/register

Registered Users:

- school email
- school id number
- password
- rating
- course1 ... n

The Registered Users entity will store information needed for registered users to log in, have their rating displayed and what courses they are taking. The courses they are taking will be used for the class material checklist and smart search.

Administrator(s):

- username
- password

Administrator entity will hold all the login information needed for administrators to log in.

Sales items:

- product name
- category
- seller
- delivery method
- active or sold?
- timestamp listed
- purchaser
- timestamp sold
- approved or unapproved?

Sales items entity will hold all product listings with details consisting of the product name, the category it belongs in, seller's school ID number, delivery method (shipping or pickup), active or sold listing, when the product was listed, purchaser's username (if applicable), date and time the product was sold (if applicable), and if the listing was approved or unapproved by an administrator.

4. Initial List of Functional Requirements

Unregistered user:

1. Shall be able to register for an account
2. Shall be able to browse home page
3. Shall be able to search for a product by name and/or category
4. Shall be able to browse a product search results page
5. Shall be able to browse a product listing page
6. Shall be able to view the form for listing a product

Registered user (In addition to unregistered user):

7. Shall be able to do 1-6
8. Shall be able to login and logout
9. Shall be able to add item(s) to cart
10. Shall be able to checkout item(s) in cart
11. Shall be able to contact the seller of a listing
12. Shall be able to submit a listing for approval
13. Shall be able to remove their listing(s)
14. Shall be able to respond back to a message from a potential buyer
15. Shall be able to access and view account profile page
16. Shall be able to access and view purchase history page

Administrator:

17. Shall be able to see listing request(s)
18. Shall be able to approve and disapprove listing request(s)
19. Shall be able to remove any active listing
20. Shall be able to ban any account

5. 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).

6. Competitive Analysis

Features	Gator Trader	Amazon	eBay	Craigslist
Delivery method (ship or pickup)	+	-	+	+
Smart search to suggest class materials based on the classes being taken by a student	+	-	-	-
Spending analysis	+	-	-	-
Ability to change site theme/ background	+	-	-	-
Featured products based on popularity (number of views)	+	-	-	-
Add to cart button for each product in product results page	+	-	-	-
Checklist of class materials visible in the student's profile page	+	-	-	-

Gator Trader will be very competitive compared to Amazon, eBay, and Craigslist. Since this is a buy and sell platform exclusive to SFSU students, delivery is very fast and efficient instead of Amazon's two-day shipping where you need to pay extra for even after one-year trial if you're a student. Compared to eBay, our pickup feature is better since meeting up with the seller won't cost you extra for shipping where as eBay would probably need to pay extra depending where you are or if the seller provides that. Next, we have a smart search that is designed for student's needs based on their profile while Amazon, eBay, and Craigslist doesn't have that. Being a college student, it's important to know how much you can spend without going over your budget and that's why we have a spending analysis. We also have a user background preference where users can personally customize their own background/theme, which is great since you can't do that with the other 3 competitors. On the home page, there will be a section that will list all the featured products based on the number of times that product has been viewed. More so, there will be a quick and easy way to add an item to cart that you don't necessarily need to read all about and be able to stay on the search page to look at other items. Finally, we will have a class material checklist that will help a student keep track of what they need to purchase for their classes.

7. High-level System Architecture and Technologies

Below is a list of the technologies used in our software stack:

Server Host: AWS, vCPUs: 1, Memory: 1GB
Operating System: Ubuntu 18.04
Database: MySQL 8.0.16
Web Server: NGINX 1.14.0
Server-Side Language: JavaScript ECMAScript 2018

Additional Technologies:

Web Framework: Express 4.17.1
IDE: Visual Studio Code 1.38.1
Web Analytics: Google Analytics
SSL Cert: N/A
CSS: Version 4
CSS Framework: Bootstrap 4

8. Team and Roles:

Steve Rentschler - Team Lead/ Document Master
Johnson Wong - Back-end Lead/ Github Master
Sergei Katukhov - Front-end Lead
Ho Yin Mak (Tevis) - Front-end
Tsun Ming Lee (Mathew) - Back-end
Timothy Chan - Front-end
Osbaldo Martinez - Back-end

9. Checklist

- Team found a time slot to meet outside of 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: **On track**
- 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**