

SW Engineering CSC 648/848 Fall 2020
Team 4

Milestone 1
GatorTrader

Niall Healy (Team Lead, nhealy@mail.sfsu.edu)

Aaron Lander

Dale Armstrong

Lukas Pettersson

Joseph Babel

Vern Saeteurn

| Date submitted | Date(s) Revised |
|--------------------|-----------------|
| September 24, 2020 | |

Table of Contents:

| | |
|--|-----------|
| 1. Executive Summary: | 3 |
| 2. Personae and main Use Cases: | 4 |
| 3. List of main data items and entities – data glossary/description: | 7 |
| 4. Initial list of functional requirements: | 9 |
| 5. List of non-functional requirements: | 10 |
| 6. Competitive analysis: | 11 |
| 7. High-level system architecture and technologies used: | 12 |
| 8. Team and roles: | 13 |
| 9. Checklist: | 14 |

1. Executive Summary:

As new students who are proficient in using technology enroll at San Francisco State University, a need for a platform that provides the ability to buy, sell, and trade essential items increases. Buying textbooks straight from a vendor is expensive, and finding good deals can be challenging. Our company would like to develop a free-to-use website that gives students the ability to find textbooks and other goods, at an affordable price.

Since many students attend the same classes at San Francisco State University, and classes usually use the same textbooks and supplies every semester, these goods should be able to be passed down from student to student each semester. This does not happen, however, because there is no communication medium in place to facilitate these transactions. Our company would like to provide this communication medium in the form of an intuitive website named **GatorTrader**. **GatorTrader** will allow the students, professors, and faculty of San Francisco State to easily buy, sell, and trade items. Some of **GatorTrader**'s features include: search by category, search by class, a messaging system, and a clean user interface.

Our team of six students is excited to provide an excellent product for our fellow Gators to use. Our ultimate goal is for **GatorTrader** to be an easy to use website that allows buyers and sellers to communicate with each other over a secure and reliable interface. In a time where the price of attending university is skyrocketing, we believe **GatorTrader** will make college life more affordable.

2. Personae and main Use Cases:

David Miller

- Characteristics
 - Sophomore SFSU Student
 - History Major
 - Holds Part-time Job
- Goals
 - To become a professor.
- Skills
 - Competent with office applications and general computing.
 - Unfamiliar with programming.
- Pain Points
 - It is the end of the semester and he wishes to sell textbooks to someone in need of them.



Main Use Case

Finals week is over for the Spring semester and David passed all of his courses without issue despite having a part-time job. He took GE courses for Introduction to Philosophy and World Literature and is looking to sell the textbooks for these courses to someone else to help pay for next semester's textbooks. Unfortunately, he does not have the time to go around asking if anyone will be taking these courses, so he goes to our website **GatorTrader** on his smartphone. From there he is immediately introduced to a few options including one to list his items for sale. Upon pressing, he is introduced with a form that allows him to describe his item and include a picture. He tags the first textbook with the course type and number (PHIL 101, Section 1), as well as ISBN and other identifying information to help others find his **Listing**. Once he presses the option to submit his **Listing**, he is prompted to register for an account. After making an account, he is brought to an overview page that lists his item for sale and he notices that it is tagged to be reviewed by an administrator and to check back later to verify his **Listing** was successful. He proceeds to follow the same process for his other textbook, except now owning an account he no longer needs to register again. David's items are now for sale and he saved time allowing him to move on to other matters.

Sarah Nelson

- Characteristics
 - Senior SFSU Student
 - Electrical Engineering Major
 - Full-time Student
- Goals
 - To pass courses with an A and maintain a high GPA.
- Skills
 - Competent in general computing.
 - Has experience with Python for general scripting purposes.
- Pain Points
 - Is in need of tutoring for Calculus III to help her prepare for an upcoming exam.



Main Use Case

There is an exam coming up for Calculus III, and Sarah cannot make time in her schedule to visit office hours at the specified times given by her professor. Since she is free on Wednesday, she decides to look towards other methods to help her prepare; she preferably wants to find a tutor that can help her work through the problems step-by-step. Upon arrival at our website **GatorTrader**, she finds an option that allows her to refine her search to show tutors for Calculus III. The **Listings** that the tutors put up show available dates/times and possible locations, such as the library or the nearby Peet's Coffee. After finding a tutor to schedule time with, she clicks an option to **Message** them and is prompted to register for an account. After making an account, she receives a disclaimer that warns the user about meeting with people at odd times and unfamiliar locations. After dismissing the disclaimer, she is prompted with a **Message** box addressed to the tutor with some pre-filled information based on her search criteria. She successfully sends her initial **Message** and she and the tutor go on to discuss her preferred payment method, as well as time and location. Sarah can now rest easy knowing she will get the help that she needs.

John Anec

- Characteristics
 - Senior SFSU Student
 - Undergrad Kinesiology Major
 - Part of the wrestling team
- Goals
 - Find somewhere to sell his workout equipment that is currently in the dorms before he graduates.
- Skills
 - Bodybuilder and knows about all the muscles in the body.
 - Loves to entertain guests
- Pain Points
 - Very little experience with computers.
 - Most internet usage has been on his cellphone.
Finds some websites difficult to navigate.



Main Use Case

John Anec is graduating in a few months and is moving across the country for a job he has lined up. Recently he has been working out in his dorm room as the gym is often filled with students using all the equipment, and in that time he has accumulated a lot of equipment. Rather than throwing it all away, he would like to find somewhere to sell it, preferably to someone that is close by and/or ideally someone else in the dorms. He isn't able to find any buyers from direct contact, but some friends recommend using the site **GatorTrader**. Even though he has limited experience with computers and websites, he is able to create an account on **GatorTrader** on his phone. Then finds the **List Items** category, selects the type of item he is selling, lists information about the item along with his contact information. With this he is able to quickly and efficient list his workout equipment for other students to purchase.

Mike Jones

- Characteristics
 - Sophomore SFSU Student
 - Biomechanical Engineering Major
 - Works part time to pay for school
- Goals
 - Lowering the costs of school
- Skills
 - Proficient with computers and programming
 - Loves reading spy novels
- Pain Points
 - Not a lot of friends because of commuting and busy from studying and working.
 - Works to pay for school, struggles to keep up with the cost.



Main Use Case

Mike has been working to pay for school and is always low on funding. He needs a way to save money anywhere he can. Some of his biomechanical engineering books cost upwards of \$200. He searches for a site where he can buy discounted books for his SFSU classes and finds **GatorTrader**, a site specifically for SFSU students and alumni. He is able to quickly find the search bar and enters in the books that he needs being sold at far lower prices by other students. He can see the **Listing** with the condition they are in, pictures, and the contact information. Mike is able to send a **Message** to the seller to get into contact with them.

3. List of main data items and entities – data glossary/description:

Data Items:

1. User

- 1.1. Users of the website have a login and password that are stored on the server. Stores all current and previous listings that the user has made. Stores listings that the user has saved for later. Stores what type of privileges this account has, this could be admin, student, or faculty.

2. Messages threads

- 2.1. Message threads are stored in tables with the two parties usernames along with a key to a message table where individual messages will be stored. This will allow us to easily get all the message threads for a particular user.

3. Listing

- 3.1. This data item is used to store sellers listings, depending on the type of listing (textbook, furniture, housing, etc.) it has different data items that need to be stored. For example, selling a textbook would require entering the ISBN, and name of the class that it is used for. If selling furniture, the color, the type(sofa, bed, table, etc.) will be stored.

4. Categories

- 4.1. Our Category table will group all the listings together based on the category. This will allow us to easily have all of the listings in one sorted place when we need to retrieve them.

4. Initial list of functional requirements:

1. Users

1.1. Registered Users

- 1.1.1. Shall be able to delete their account
- 1.1.2. Shall be able to edit account information
- 1.1.3. Shall be able to post item/service **listings**
- 1.1.4. Shall be able to delete item/service **listings**
- 1.1.5. Shall be able to edit item/service **listings**
- 1.1.6. Shall be able to post **listings** with a multiple photos
- 1.1.7. Shall be able to tag **listings** with common attributes
- 1.1.8. Shall be able to send messages to sellers/respond to buyers
- 1.1.9. Shall login with a username and password
- 1.1.10. Shall be able to report users/**listings** for suspicious/illegal activity
- 1.1.11. Shall be able to view past messages for specific **listings**

1.2. Any Users

- 1.2.1. Shall be able to use their SFSU email to create an account
- 1.2.2. Shall be able to search for items by item name
- 1.2.3. Shall be able to search for items by **category**
- 1.2.4. Shall be able to search for items by class name

1.3. Admins

- 1.3.1. Shall be able to review **listings**/edits/reports
- 1.3.2. Shall be able to remove **listings** at any time
- 1.3.3. Shall be able to provide feedback if a **listing** is rejected
- 1.3.4. Shall be able to message registered users
- 1.3.5. Shall be able to ban user account
- 1.3.6. Shall be able to report illegal activities to the authorities

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. All or selected application functions must render well on mobile devices
4. Data shall be stored in the database 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 (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application should follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
12. No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive analysis:

| <u>Features</u> | GatorTrader | Amazon | eBay | Craigslist |
|---------------------------|--------------------|---------------|-------------|-------------------|
| Search By Category | ++ | ++ | ++ | ++ |
| User-friendly UI | ++ | ++ | ++ | + |
| Returns | + | ++ | + | - |
| Price Negotiation | + | - | + | + |
| Reselling | + | + | + | + |
| SFSU Class Search | ++ | - | - | - |

++ = excellent

+ = good

- = unavailable

GatorTrader will combine the best aspects of other e-commerce websites (Amazon, eBay, Craigslist) while offering exclusivity to the SFSU campus. **GatorTrader** will be different from other websites in that it allows users to search for items pertaining specifically to a SFSU class. This will allow users to quickly find needed textbooks and supplies to prepare for the upcoming semester. Additionally, since **GatorTrader** is specific to SFSU, users will be ensured that pickup spots will always be within a reasonable distance. All of this will all be available on a clean and intuitive user-interface free from advertisements and clutter that are often found on other websites.

7. High-level system architecture and technologies used:

Software stack:

Server Host: Amazon AWS 1vCPU 1 GB RAM

Operating System: Ubuntu 18.04.5 LTS

Database: MySQL 14.14 Distrib 5.7.31

Web Server: Uvicorn 0.11.8

Server-Side Language: Python 3.6.9

Additional Technologies:

Frontend Framework: Bootstrap, React

Web Framework: FastAPI (by proxy, Starlette)

IDE: JetBrains PyCharm, IntelliJ

Web Analytics: Amazon Kinesis Data Analytics

SSL Certificate: Let's Encrypt (Cert Bot)

SASS: 3.5.6

Supported Browsers:

Firefox 81.0

Google Chrome 85.0.4183.121

8. Team and roles:

- Niall Healy - Team Lead, Document Master
- Aaron Lander - Github Master
- Dale Armstrong - Front End Lead
- Lukas Pettersson - Back End Lead
- Joseph Babel - Team Member Back End
- Vern Saeteurn - Team Member Front End

9. Checklist:

DONE So far all team members are engaged and attending ZOOM sessions when required

DONE Team found a time slot to meet outside of the class - 2 PM Fridays

DONE Back end, Front end Leads, and 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.)