

SW Engineering CSC648/848
Section 01
Fall 2018

GatorTrade

Team 04

Abigail Chin | chinabigail08@gmail.com

Alyssa Malunao

Dhruv Shah

Jarek Rettinghouse

Jed Ahmadia

Nikita Bajracharya

Robert Quiñones

Milestone 1

Submission Date

Feedback Given

(First Draft) October 4, 2018	
--------------------------------------	--

1. Executive Summary

GatorTrade is the ultimate buying, selling, listing site for San Francisco State University (SFSU) students. Students are often in search of items and services specific to their needs. Whether it be roommates, concert tickets, bikes, computers, books, tutors, or other necessities, GatorTrade aims to supply an easy way for SFSU students to find what they are looking for. Amazon, Ebay, and other listing sites such as these may provide some amenities, but GatorTrade delivers a well-rounded, one-stop site for all a student may be searching for. We believe in students helping students. This site is designed for students to help each other—in finding housing, tutoring, or merely offering a ride to a similar destination. With San Francisco's rent rapidly increasing in price, students are often looking to find cheaper housing and frequently have to move several times throughout their college career. Not only this, but the price of living in San Francisco is also steadily rising. We aim to help students find cheaper housing easily, quickly, and sell items for profit, funds they can use for surviving the harsh living costs of San Francisco.

On GatorTrade, users can peruse the site listings by choosing categories or searching key terms for the item they desire. Should they find something they are interested in, they can then choose to create an account and message the seller. A user with an account can also post listings for sale or services they wish to offer. For each item offered, a preferred meetup spot will be assigned (by the seller) and a detailed description with photos can be provided. The poster of the listing can select this meetup spot and the buyer can determine if this is an acceptable meeting place or choose to message the seller for a different option. Since the site is designed for SFSU students, the listings will be specific for this demographic: the events, tutoring services, rideshare opportunities, books, jobs, lost and found, and other items will be in San Francisco or solely for SFSU students. Users can also view their dashboard: a place where they can see past messages and a watchlist—a list of items they can flag for future reference.

GatorTrade will be made by a group of students who are passionate about coding and invariably learning more. We are all computer science majors hoping to discover new things and this project will be a challenge for us all. With a positive outlook and an insightful, hard-working, incredibly talented team, we hope to make this project great.

2. Personas and Use Cases

1. Josh (Registered User)

Josh is a graduating student from San Francisco State University (SFSU) who wants to get rid of all his books. Josh accesses our buy and sell website, GatorTrade. He goes to our option for posting an ad and is then prompted to create an account in order to post an ad on our website. After creating an account he posts an ad on the website. For the purpose of his privacy he only provides his email address to the buyers. A buyer decides to contact Josh via the email and sets up a time and place to purchase the books. For safety and convenience reasons they decide to meet up at the given designated safe place on SFSU's campus.

2. Dylan (Unregistered User)

Dylan is a freshman student at SFSU; he is still undecided about his major. Therefore, this semester he decides to take all General Education courses. The bookstore is out of stock for the books on couple of his courses and Amazon will take at least a couple of days to get his books delivered. Dylan hops on our site and searches by course number to look up books for the different classes he is taking this semester. After finding the books he needs, he decides to contact the seller for one. He is then prompted to create an account.

3. John (Registered User)

John finds out that his microwave is not working and he is not looking to spend too much money on a new microwave. John's friend recommends GatorTrade to help him find a used microwave for sale. While John is browsing, he also notices that a few other household items and furnitures are on sale for reasonable prices. He decides to buy the microwave and arranges a meet up with the seller on SFSU campus (provided by the seller for their safety and convenience). The seller's name is Layla and she receives a message on her dashboard that John wants to buy her microwave. He would like to meet with her for the exchange at one of the designated areas on SFSU campus provided by GatorTrade. After a pleasant meeting with Layla, he also gains a new friend.

4. Andrea (Administrator)

Andrea is an employee at GatorTrade's office and she is assigned the duty to work with an administrative team in managing its web services. She is granted all of the proper permissions on her account and starts familiarizing herself with the website and its functionalities. While browsing the seller lists, she notices that there are a couple of spammers posting on the site, so she bans them and deletes the posts that are considered

to be spam. Later, she encounters another issue regarding someone posting an inappropriate item so, she swiftly removes the post and warns the user.

3. Data Definitions

1. Registered Users

A user who has an account on the site and is logged in. They shall be able to post new items for sale, message others, and can be either a buyer or a seller.

2. Unregistered Users

A user who has no account, or has yet to log in to account. This type of user shall still be able to view content on the site, but cannot make a new posting or message a seller. This user shall be able to create an account if he/she so chooses.

3. Admin

Special credentials gives the admin access to the Admin Console. The Admin Console allows the admin to approve pending ads and upgrade other users to admins as well.

4. Items

At item posted by a Registered User for sale. Items include a title, description, price, category, related images (zero or more), and a preferred meeting place.

5. Messages

A form of communication between Registered Users relating an item for sale. Messages are sent from a buyer to a seller, with contact information attached so that the seller will be able to contact the buyer based on the given contact information.

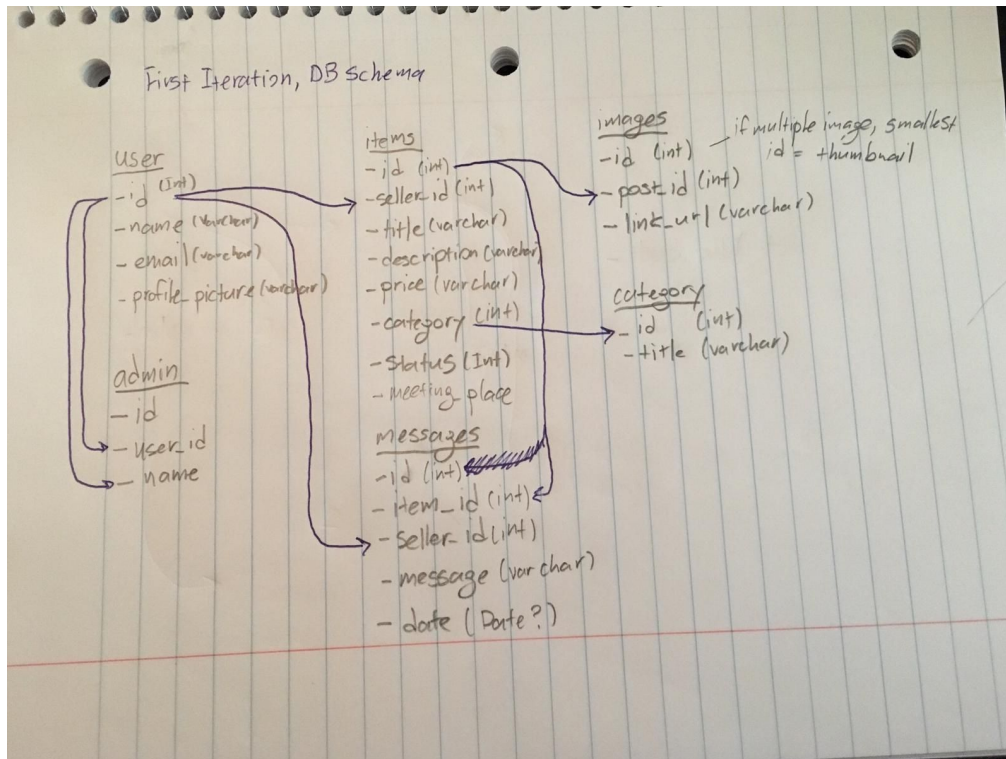
6. Images

Images can either be a user's profile picture, or images relating to the item being sold.

7. Category

Subcategories for items posted on the site (ie. books, electronics, home goods, etc.)

First Draft - Database Entity Diagrams



DB Table								
1) item	id	seller_id	title	description	price	category	status	meeting_place
2) user	id	name	email	profile_picture				
3) category	id	title						
4) ad_images	id	post_id	link_url					
5) admin	id	user_id	name					
6) messages	id	item_id	seller_id	message	date			

4. Initial List of Functional Requirements

Unregistered users

1. Unregistered Users shall be able to browse site and view listings without creating an account.
2. Unregistered Users shall be able to view 50 items per page.
3. Unregistered Users shall be able to view item details.
4. Unregistered Users shall be able to view seller's information.
5. Unregistered Users shall be prompted to create new account for buying and selling purposes.

Registered Users

6. Registered Users shall log in and log out of the system.
7. Registered Users shall be able to view dashboard.
 - 7.1 Dashboard: profile page for user. Contains messages from buyer/seller for interested items. Contains user settings and watchlist.
8. Registered Users shall be able to post items/services details for sale.
9. Registered Users shall be able to communicate with buyer or seller.
 - 9.1. Registered Users shall be able to message the seller for an item of interest.
 - 9.2. Registered Users shall be able to view messages from interested buyers.
10. Registered Users shall be able to mark items they have sold as sold.
11. Registered Users shall be able to leave a feedback for website.

Admin

12. Admin shall be able to login and logout.
13. Admin shall be able to grant and deny access to users.
14. Admin shall be able to manage contents (CRUD).
15. Admin shall be able to create and delete account.
16. Admin shall be able to manage items and postings.
 - 16.1. Admin shall be able to approve items for sale. Items shall not be posted until admin's approval.
 - 16.2. Admin shall be able to remove inappropriate or unfit items and message the user.

Items

17. Item shall have all the available configurations.
 - 17.1 Item shall have one or more images (optional) , title, description, preferred meeting place, and price
18. Item shall have product image in product detail page.
19. Only approved items shall be posted.

System

- 20. System shall provide full functionality without using mouse.
- 21. System shall provide user access to summary as well as raw data.

Privacy

- 22. Messages shall be private.
- 23. User information shall be private.

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 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).

6. Competitive Analysis

Features	GatorTrade	Amazon	Ebay	CraigsList	Buy near.me
Buy	+	+	+	+	+
Sell	+	+	+	+	+
Categories	+	+	+	+	+
Search	+	+	+	+	+
Login/Accounts	+	+	+	+	+
No Shipping	+	-	-	-	+
Preferred Meet Up Locations	+	-	-	-	-

Summary of Features:

Here are some of the early features we are implementing on our site. Users shall be able to buy various items and services by creating an account and browsing through the many things that are available. Users shall be able to sell various items and services by creating an account and posting what it is they are selling. Website shall give users access to searching for items and or services through either a search or by categories. Buyers shall meet up with nearby sellers and exchange cash (or other method of payment) for item.

Advantages Against Competition:

Whenever a company goes into business it is natural that the company will have competition. Here are a few advantages GatorTrade has against our competitors. One advantage we have is that we don't have shipping, where all exchanges happen either on or near the SFSU campus. Another advantage we have is that sellers shall be able to choose a preferred meet up location (on or near SFSU campus) with the buyer. One of the last advantages we have so far is that the user shall be able to find specific categories based on location (i.e. events, tickets, jobs, tutoring services, etc.).

7. High-Level System Architecture

Software Components

- **Server Host / Deployment Platform:** Heroku-18
- **Server Operating System:** Ubuntu 18.04 Server
- **Web Server:** Node.js 8.12.0
- **Database Management System:** PostgreSQL 10.5
- **Server-Side Programming Language:** JavaScript (ES6)

Frameworks

- **Front End**
 - **Bootstrap**
 - Free and Open-Source Front End framework for designing websites and web applications.
- **Back End**
 - **Node.js**
 - An asynchronous, event driven, JavaScript runtime. Designed to build scalable network applications.
 - **Express.js**
 - Express.js is a Node.js web application server framework, designed for building dynamic web applications.
 - **EJS - Embedded JavaScript Templates**
 - EJS is a simple templating language that lets you generate HTML markup with plain JavaScript.
 - **Cloudinary (API)**
 - Manages web and mobile media assets. Heroku addon.

IDE: Microsoft Visual Studio Code - A streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle.

Browser Version Support

- **Google Chrome** (69.0.3497.100, 69.0.3497.105, or higher)
- **Mozilla Firefox** (62.0.2 or higher)
- **Safari** (12.0, or higher)

Version Control System: GitHub

Analytics: Google Analytics

8. Team

- **Abigail Chin** - Team Lead
- **Robert Quiñones** - Back End Lead
- **Dhruv Shah** - Front End Lead
- **Jarek Rettinghouse** - Back End Developer
- **Alyssa Malunao** - Front End Developer
- **Jed Ahmadia** - Front End Developer
- **Nikita Bajracharya** - Front End Developer

9. 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 who needs to learn are working on it - **DONE**
- Team lead ensured that all team members read the final M1 and agree / understand it before submission - **DONE**