

# SW Engineering CSC648/848 Fall 2019

## “Gator Store”

### Section 02

### Milestone 5

### Team 12

Beibei Jiang - Team Lead / Github Master / Back-end Developer

Email: [bjiang1@mail.sfsu.edu](mailto:bjiang1@mail.sfsu.edu)

Jinying Ren - Front-end Lead

Jesse Smick - Back-end Lead

Angelo Solitario - Front-end Developer

Melissa Endang - Front-end Developer

Jinghan Cao - Back-end Developer

URL: <https://gatorstore.duckdns.org/>

Submitted Date: 12/17/2019

# 1. Product summary

Our product named “Gator Store” is a website that allows SFSU students to buy and sell items such as books, electronic products, furniture, and more. For those who have an economic burden and need to sell items in a short time, our platform is the best choice. What makes our product unique is in its ability to provide SFSU students with easy, fast, economic experience from one seamless shopping website.

## **All major committed functions:**

### Unregistered Users:

1. Users shall be able to create an account
2. Users shall be able to view listed items for sale
3. Users shall be able to browse through category
4. Users shall be able to see the remaining amount of items
5. Users shall be able to filter items by price

### Registered Users:

7. +Unregistered users’ functions
8. Users shall be able to log into the website
9. Users shall be able to post items attached with prices for sale on this website
10. Users’ posted items shall be reviewed by an administrator before it is officially posted on website
11. Users shall be able to have their own dashboard
  - a. View the in-site messages
  - b. View past posted items
  - c. Delete past posted items
12. Users shall be able to send in-site messages among each other
13. Users shall be able to determine a pick up spot on SFSU campus

### Admins:

14. Administrators shall be able to log in the website
15. Administrators shall be able to have administrator dashboard

16. Administrators shall be able to review the items before they are officially posted
17. Administrators shall be required to approve items before they are officially posted
18. Administrators shall be able to delete items

**URL:** <https://gatorstore.duckdns.org/>

## **2. Milestone documents – M1-M4**

# **SW Engineering CSC648/848 Fall 2019**

## **“Gator Store”**

### **Section 02**

### **Milestone 1**

### **Team 12**

Beibei Jiang - Team Lead / Github Master / Back-end Developer

Email: [bjiang1@mail.sfsu.edu](mailto:bjiang1@mail.sfsu.edu)

Jinying Ren - Front-end Lead

Jesse Smick - Back-end Lead

Angelo Solitario - Front-end Developer

Melissa Endang - Front-end Developer  
Jinghan Cao - Back-end Developer

Submitted Date	10/2/2019
Revised and Frozen Date	10/4/2019

## 1. Executive Summary

SFSU students have the need for buying and selling resources. Seniors about to graduate need to sell items such as books, electronic products and furniture in a short time. These things are exactly what current students and especially for those who have economic burden seek. “Gator Store” aims to offer easy, economic experience for and only for SFSU student from one seamless shopping website.

In our website, buyers can easily browse items by category, can search product using its name or class number, can sort products as well. After clicking on one particular item, they can see its picture, description and other information. Moreover, buyers can communicate with seller via in-site message box after logging in.

As for sellers, they are allowed to post information as well as edit posted items. Everything posted will show up only after it is approved by website admin who deals with improper post request regularly and also make sure in-site communication are healthy and smooth. In addition, lazy registration will be implemented which means anyone can browse and search easily without signing up or logging in until he or she wants to post, send messages or purchase.

One of the unique features of our website is allowing users to search listings not only according to the name but also its related course number. For example, a student is going to take csc510 and would like to buy a textbook. When referring to other shopping website, the student should provide the book name together with maybe other information like author name and book version. On our website, she/he can easily type csc510 in the search box and satisfied results show up immediately. On top of that, meetup spots are limited to a 0.5 mile radius near campus under security camera. Students get products easily, fast and safely compared with other websites.

Our team consists of six current SFSU students who are dedicated to solving student online trade. You can find more information about us in about page.

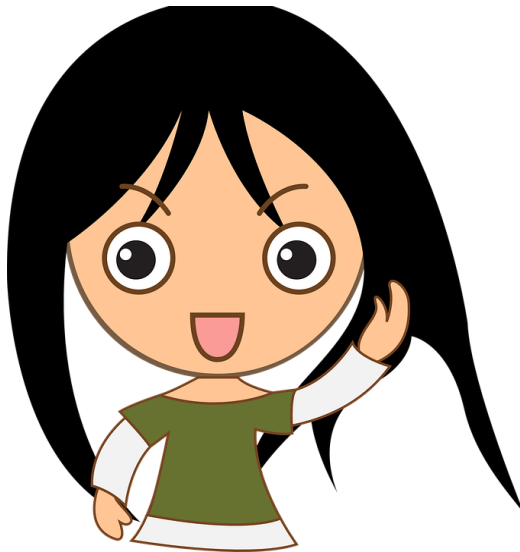
## 2. Personae and main Use Cases

### 2.1 Personae



**Ali – Admin**

Ali is an SFSU student and an administrator on the SFSU buy/sell website. He is very skilled and has a thorough understanding of the website, software, and browsers. He is very organized when it comes to events, files, and a system that makes sense for everyone.



**Bella- Buyer**

Bella is a fulltime student and who is also working full time. She is a hardworking person. Because she needs to help her parents with paying the bills, she does not have enough money to pay the full price for the book she needs for one of her classes.



**Silvester – Seller**

Silvester is a student who is graduating soon. He has no siblings or anybody that can use all his textbooks and things from his dorm. He needs to sell or get rid of his things very soon before he moves. Otherwise, he will need to pay extra to move them to his new place.

## **2.2 Use Cases**

### Managing the listings

Ali logs in on the website to check on the listings and approve all the listings. He can browse the listings starting from the most recent listing. While browsing, he realized that a user is trying to post very offensive and inappropriate stuff on the website. Ali proceeded to reject the posting and it will not be visible to the user.

After some time, the same user is trying again to post the same listing. Ali rejected the listing and sent a warning to the registered user that he/she may be banned for multiple and repeat attempts.

### Buying

Bella decided to visit the SFSU buy/sell website to see if the book is listed. There she was able to filter the listings by classes offered in SFSU. She found the book and proceeded to contact the seller and buy the book. Before completing the message, she is prompted to register. After contacting the seller, she was able to acquire the book and a new friend.

### Selling

Silvester decided to go to the SFSU buy/sell website because he knows that students will benefit from the low prices of textbooks and things. Silvester is browsing the website and he easily found the button for posting a listing on the website. He filled up the form needed to sell an item and then he was prompted to register. He then proceeded to register and sign up to post and sell. It was easy for him to post pictures of his items. Now, Silvester can go back and check if anybody is interested in buying his stuff.

### Editing listings

Silvester has posted many listings on the website most of them sell in a short amount of time. However, one item will not sell for some reason. He decided to check the listing using his dashboard and realized that he did not describe the item well enough. He decided to edit the listing's description to better describe the item. After changing it, he received offers for the item.

### Browsing

Bella was not looking for anything specific this time. She is using a different computer and forgot her password. Being lazy, she decided not to log in. She was able to browse the listings without the trouble of having to log in or register. She was able to compare the prices of items that have multiple listings. She visited the website a couple of times and was able to browse freely.

## **3. List of main data items and entities**

- **Unregistered Users:** Users who have not registered or log in to the account. They shall be able to search and browse items, but they cannot post new items or in-site messages to other users.
- **Registered Users:** Users who have accounts and are logged in (can be buyers or sellers). They can browse, search, post and remove items, and they can send in-site messages to interact with other users.

- Email
  - Username
  - Password
- **Admin:** The admin user has access to the website's administrator panel. The admin keeps the website running efficiently and securely. Their tasks are to manage user accounts, posts request and content.
  - Username
  - Password
  - Remove posts
  - Remove items
  - Issue warnings
  - Ban from posting
  - Contact sellers or buyers
- **In-site Messages:** Communication between registered users (buyer, seller, and admin). They can contact each other with text and pictures about items' problems or questions.
- **Items:** Items posted by registered users (sellers) for sale. Items include:
  - Name
  - Price
  - Description
  - Pictures
  - Condition
  - Distance from campus
  - Category
- **Category:** Users shall be able to look for specific kinds of items efficiently through category (books, electronics, school supplies, accessories, etc).
- **Search:** Users shall be able to find items that they want efficiently with items' names or keywords.
- **Login:** Users can login to their personal account page to access their information.
- **Registered users account:** It records the name, email, address, and purchase details of the registered users.



## **4. Initial list of functional requirements**

### **Unregistered Users**

1. Users shall be able to create an account
2. Users shall be able to view listed items for sale
3. Users shall be able to browse through category
4. Users shall be able to browse through tag
5. Users shall be able to see the remaining amount of items
6. Users shall be able to be redirected to register page if they want to post items for sale

### **Registered Users**

7. +Unregistered Users' functions
8. Users shall be able to log into the website
9. Users shall be able to have their own dashboard
10. Users shall be able to post items attached with descriptions for sale on this website.
11. Users shall be able to post items attached with images for sale on this website.
12. Users shall be able to post items attached with prices for sale on this website.
13. Users shall be able to post items attached with tags for sale on this website.
14. Users' dashboard shall be able to view the in-site messages
15. Users' dashboard shall be able to view past posted items
16. Users' dashboard shall be able to modify past posted items
17. Users' dashboard shall be able to delete past posted items
18. Users' posted items shall be reviewed by an administrator before it is officially posted on website
19. Users' posted items shall be reviewed again if they update the information about items.
20. Users shall be able to send in-site messages among each other.
21. Users shall be able to view approval results from an administrator regarding their proposed items for sale
22. Users shall be able to choose the location within campus to trade off-line
23. Users shall be able to receive confirmation message sent to their in-site message box

### **Administrators**

24. Administrators shall be able to log in the website
25. Administrators shall be able to have administrator dashboard

26. Administrators shall be able to review the items before they are officially posted
27. Administrators shall be required to approve items before they are officially posted
28. Administrators shall be able to send messages to sellers if the items are against the policies

## **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	Craigslist	Ebay	Facebook Market	SFSU Store
Browse by category	+	+	+	+
Communicate via in-site messaging	+	+	++	+
One-click access to SFSU books	-	-	-	++
Safe meetup spots on campus	-	-	-	++
	key:        - = don't have    + = has        ++ = does well			

We plan on making it easy to browse and submit posts before signing up is needed. A key feature we have over the competition is having predetermined safe meetup spots on campus to recommend to the users. Another feature we have over other sites is easy access to books for SFSU classes. Overall we've strategized to leverage the inefficiencies of existing platforms to bring a competitive advantage into the market.

## 7. High-level system architecture and technologies used

Server Host: Google Compute Engine 1vCPU 0.6GB RAM

Operating System: Ubuntu Server 18.04 LTS

Database: MySQL v5.7.x

Web Server: NGINX 1.17.x

Front-end Language: bootstrap, css, html, javaScript

Server-Side Language Engine: Node.js v10.x

Server-Side Language: JavaScript

Additional Technologies:

- Web Framework: Express.js v4.x

- Web Analytics: Google Analytics
- VirtualBox v6.x
- Vagrant v2.x
- Ansible v2.x

## 8. Team and roles

Beibei Jiang - Team Lead / Github Master / Back-end developer

Jinying Ren - Front-end Lead

Jesse Smick - Back-end Lead

Angelo Solitario - Front-end Developer

Melissa Endang - Front-end Developer

Jinghan Cao - Back-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 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

# SW Engineering CSC648/848 Fall 2019

## “Gator Store”

### Section 02

### Milestone 2

### Team 12

Beibei Jiang - Team Lead / Github Master / Back-end Developer

Email: [bjiang1@mail.sfsu.edu](mailto:bjiang1@mail.sfsu.edu)

Jinying Ren - Front-end Lead

Jesse Smick - Back-end Lead

Angelo Solitario - Front-end Developer

Melissa Endang - Front-end Developer

Jinghan Cao - Back-end Developer

Submitted Date	10/18/2019
Revised and Frozen Date	10/27/2019

# 1. Functional Requirements

## Priority 1

### Unregistered Users:

1. Users shall be able to create an account
2. Users shall be able to view listed items for sale
3. Users shall be able to browse through category
4. Users shall be able to browse through tag
5. Users shall be able to see the remaining amount of items
6. Users shall be able to filter items by price
7. Users shall be able to determine a pick up spot on SFSU campus
8. Users shall be able to have quick access to textbooks by searching by class or teacher

### Registered Users:

9. +Unregistered users' functions
10. Users shall be able to create an account
11. Users shall be able to view listed items for sale
12. Users shall be able to browse through category
13. Users shall be able to be redirected to register page if they want to post items for sale
14. Users shall be able to log into the website
15. Users shall be able to have their own dashboard
16. Users are required to post images with items for sale on this website
17. Users shall be able to post items attached with prices for sale on this website
18. Users' dashboard shall be able to:
  - a. View the in-site messages
  - b. View past posted items
  - c. Delete past posted items
19. Users' posted items shall be reviewed by an administrator before it is officially posted on website
20. Users shall be able to send in-site messages among each other

### Admins:

21. Administrators shall be able to log in the website
22. Administrators shall be able to have administrator dashboard
23. Administrators shall be able to review the items before they are officially posted
24. Administrators shall be required to approve items before they are officially posted
25. Administrators shall be able to delete items
26. Administrators shall be able to delete users

## **Priority 2**

### Registered Users:

1. Users shall be able to browse through tag
2. Users shall be able to post items attached with descriptions for sale on this website
3. Users' dashboard shall be able to modify past posted items
4. Users' posted items shall be reviewed again if they update the information about items

## **Priority 3**

### Registered Users:

1. Users shall be able to see the remaining amount of items
2. Users shall be able to post items attached with tags for sale on this website
3. Users shall be able to view approval results from an administrator regarding their proposed items for sale
4. Users shall be able to choose the location within campus to trade off-line

### Admins:

5. Administrators shall be able to send messages to sellers if the items are against the policies

## 2. UI Mockups and Storyboards(high level only)

2.1 Ali the admin goes through the posts to review them one by one to approve or reject them.

### Admin Dashboard page

CSC 648 Class Text			
Logo	Gator Store	adminName	▼
NAVBAR			
Showing 1 -10 of 20 results		All posts	All Users Pending
Title	Description	Seller	Date Posted
title1	description1	seller1	date1
title2	description2	seller2	date1
title3	description3	seller3	date1
title4	description4	seller4	date1
title5	description5	seller5	date1
	↓ ↓ ↓ ↓		
< First 1 2 Last >			
Footer			

### Admin views one post to review page

CSC 648 Class Text	
Logo	Gator Store adminName ▼
NAVBAR	
<div><div>Title</div><div><div>Images</div><div>◀ ▶</div><div>● ● ● ● ●</div></div><div>Full description of the item...</div><div>Seller Date Posted</div><div>Reject Approve</div></div>	
< Previous Post	Next Post >
Footer	



2.2 Bella went on the website to browse by category. Bella views one item.

### Browsing Page

CSC 648 Class Text			
Logo	Gator Store		Sign up   Login
NAVBAR	All ▼	Go	Sell
Showing 1-12 of 50 results		Sort by Price: low to high ▼	
<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>
<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>
<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>	<div>\$Price</div> <div>Image</div>
< First 1 2 3 4 5 Last >			
Footer			

### Item Page

CSC 648 Class Text	
Logo	Sign up   Login
Gator Store	
NAVBAR	
Title	Seller:
<div>◀</div> <div>Images</div> <div>▶</div> <div>● ● ● ● ●</div>	<div>Image</div> <div>userName</div> <div>Message</div>
\$Price	
Date Posted	
Full description of the item...	
Footer	

**2.3** Sylvester goes to the website to sell items and is then prompted to fill up the form and then sign up.

### Sell form

CSC 648 Class Text		
Logo	Gator Store	Sign up   Login
NAVBAR		
*Title	* - required fields	
<input type="text"/>		
*Price		
\$ <input type="text"/>		
*Category:	<input type="text" value="Furniture"/> ▼	
*Description		
<input type="text" value="Type description of item here..."/>		
Pictures: <input type="button" value="Add an image"/> 3 of 5		
<input type="text" value="picture1.jpg"/>		
<input type="text" value="picture2.jpg"/>		
<input type="text" value="picture3.jpg"/>		
<input type="button" value="SUBMIT"/>		
<input type="button" value="CANCEL"/>		
Footer		

### Sign up Page

CSC 648 Class Text	
Logo	Gator Store
SIGN UP	
First Name	
<input type="text"/>	
Last Name	
<input type="text"/>	
Email	
<input type="text"/>	
Password	
<input type="password"/>	
Confirm Password	
<input type="password"/>	
<input type="checkbox"/> I agree to terms and conditions	
<input type="text" value="CAPTCHA"/>	
<input type="button" value="JOIN"/>	
Already have an account? <a href="#">Login</a>	
Footer	

## 2.4 Sylvester logs in to view messages.

## Login page

Logo

CSC 648 Class Text

Gator Store

Login

Email

Password

CAPTCHA

Login

[Forgot Email?](#)

[Forgot Password?](#)

Don't have an account? [Sign up](#)

Footer

## User dashboard

CSC 648 Class Text

Logo

Gator Store

userName ▼

NAVBAR

All items

All messages

Sort by 

Date ▼

New Post

User	Message	Date
user1	message1	date1
user2	message2	date2
user3	message3	date3

Footer

### 3. High level Architecture, Database Organization

#### **DB Tables:**

1. **Entities:** Users

**Attributes:**

- User\_Id(primary key)
- Email
- Password
- Role

2. **Entities:** Items

**Attributes:**

- Item\_Id(primary key)
- Title
- Photo
- Price\_Link
- Category\_Id(foreign key)
- Description
- Tag
- User\_Id(foreign key)

3. **Entities:** Category

**Attributes:**

- Category\_Id(primary key)
- Title

4. **Entities:** Message

**Attributes:**

- Message\_Id(primary key)
- Content
- Time\_Stamp
- User\_Id(foreign key)
- Item\_Id(foreign key)

#### **Media storage:**

Images will be stored using the file path in the data field “photo” in table items.

### **Search/filter architecture and implementation:**

Our website’s search uses the simple MySQL filter query to access requested content on the database. The query is created by the user who fills out the searching bar then hits the submit button to search for specific items. Items can be filtered based on name, category and other related information(like course number).

In backend, data can be queried by using MySQL LIKE operation. For example:  
SELECT Price FROM Items WHERE Category\_Id LIKE ‘1%’;

### **APIs:**

No API will be used.

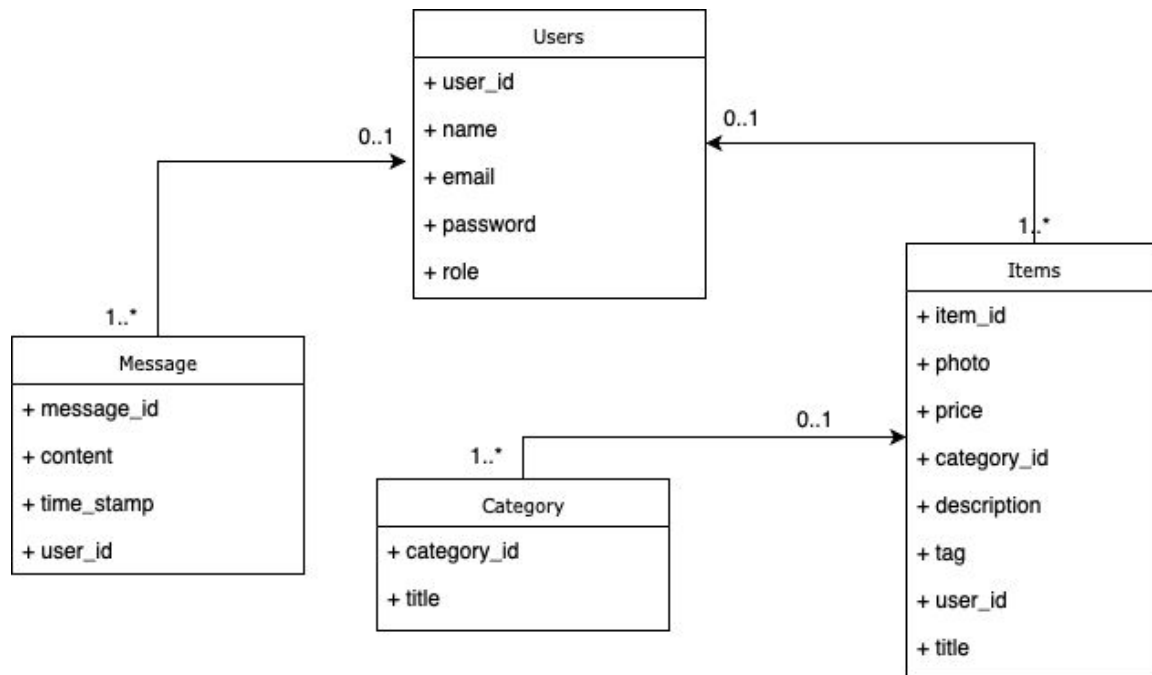
### **Significant non-trivial algorithm or process:**

No significant non-trivial algorithms or process will be used.

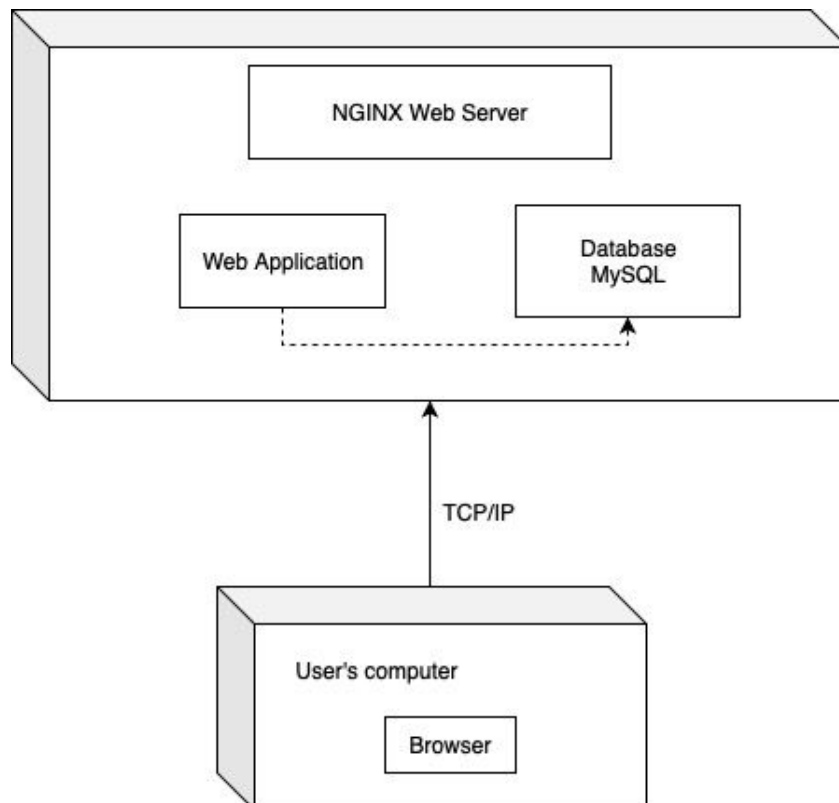
### **Any changes in Software:**

There is no significant change to the software stack at this time.

## 4. High Level UML Diagrams



### Deployment diagram:



## 5. Identify actual key risks for your project at this time

- Skills risks:  
Team members are all well informed with web development concepts and practices needed to complete the project.
- Schedule risks:  
Each team member is responsible to keep scope to a minimum viable for making sure they complete tasks on time before the deadline so there are no schedule risks.
- Technical risks:  
There are no technical risks that will affect the project because all team members will inform each other when there are any potential risks.
- Teamwork risks:  
All team members may be busy with other commitments such as school and work which makes finding a good time to meet outside of class more difficult.
- Legal/content risks:  
Content posted such as images shall be approved by Admin before appearing on the website. Those images must be cleared for any copyright issues and need proper licensing before posting.

## 6. Project management

In milestone 2, we chose two members from back-end to take charge of vertical prototype and the rest of us write documentation. We divided the work according to the workload of each subsection. Milestone 2 documentation has six subsections and among them, the second and fourth part have relatively heavier workload, we assigned two people for each of them and other members are responsible for two subsections respectively.

In future, after M2 is approved, we are planning to start working based on whether it is for frontend or backend. Each team member will be given a specific task to follow. We have started to use Github issues and projects to manage our tasks. After each milestone has been distributed, we will create specific issues and assign them to respective team members. From github project, every team member can easily know the process of every issue, whether it is just started, in processing or finished.

# **SW Engineering CSC648/848 Fall 2019**

## **Milestone 3 Feedbacks Summary**

**Team 12**



The following are things we need to do:

- homepage:
  - take out the bridge picture
  - take out the bridge picture
  - results page out of how many items
  - categories stay after reload, make it persistent
  - navbar links font color white
  - don't need home in the navbar
  - Gator Store logo should be moved to the left
  - link picture of item to item page
- details page:
  - use username for seller instead of first name and last name
  - user cannot edit the prefill message
- Dashboard
  - show how many items posted
  - sortable by subject
- posting page
  - make it fields shorter
  - fixed text your posting will show and approve after 24 hours
- login / signup

- remove password requirements if not implemented
  - all fields mandatory
  - shrink fields
  - add captcha
- Github
  - explain what the code is doing / put a header for every page
  - polish on commit comments

# SW Engineering CSC648/848 Fall 2019

## “Gator Store”

### Section 02

### Milestone 4

### Team 12

Beibei Jiang - Team Lead / Github Master / Back-end Developer

Email: [bjiang1@mail.sfsu.edu](mailto:bjiang1@mail.sfsu.edu)

Jinying Ren - Front-end Lead

Jesse Smick - Back-end Lead

Angelo Solitario - Front-end Developer

Melissa Endang - Front-end Developer

Jinghan Cao - Back-end Developer

Submitted Date	12/12/2019
Revised and Frozen Date	12/17/2019

## **1. Product summary**

Our product named “Gator Store” is a website that allows SFSU students to buy and sell items such as books, electronic products, furniture, and more. For those who have an economic burden and need to sell items in a short time, our platform is the best choice. What makes our product unique is in its ability to provide SFSU students with easy, fast, economic experience from one seamless shopping website.

### **All major committed functions:**

#### Unregistered Users:

1. Users shall be able to create an account
2. Users shall be able to view listed items for sale
3. Users shall be able to browse through category
4. Users shall be able to see the remaining amount of items
5. Users shall be able to filter items by price

#### Registered Users:

7. +Unregistered users’ functions
8. Users shall be able to log into the website
9. Users shall be able to post items attached with prices for sale on this website
10. Users’ posted items shall be reviewed by an administrator before it is officially posted on website
11. Users shall be able to have their own dashboard
  - a. View the in-site messages
  - b. View past posted items
  - c. Delete past posted items
12. Users shall be able to send in-site messages among each other
13. Users shall be able to determine a pick up spot on SFSU campus

#### Admins:

14. Administrators shall be able to log in the website

15. Administrators shall be able to have administrator dashboard
16. Administrators shall be able to review the items before they are officially posted
17. Administrators shall be required to approve items before they are officially posted
18. Administrators shall be able to delete items

**URL:** <https://gatorstore.duckdns.org/>

## **2. Usability test plan**

**Test Objective:** This test is designed to test the effectiveness, subjectivity, and efficiency of the search functionality. We decided to test the search functionality since it is a buy and sell website. Users are most likely going to be looking for specific items therefore the search functionality is very important. The intended goal of our website is to be able to provide users the efficiency and effectiveness of searching items.

### **Test background and setup:**

**System Setup:** Using the latest 3 updates of the browsers listed below

Chrome

Firefox

Safari

### **Intended Testers:**

Any SFSU student or faculty member.

### **Usability Task Description**

#### **Tasks:**

**Starting point for the test:** <http://gatorstore.duckdns.org>

**Each user will be asked to find an item that they are interested in.**

**Each user will be asked to find a book.**

**Each user will be asked to find items by category**

**Each user will be asked to browse all the items**

**Each user will be asked to complete a satisfactory survey**

**Effectiveness:** The effectiveness of the functionality is based on how many users are able to complete the task without any errors within the given time. For this test, we will be looking for if the user can complete the task in a minute. For example, if a user types in “book”, the results should show all the books that are available for sale. When using categories, the user should be able to see items related to that category plus any keywords that a user enters in. In simple terms, we will measure the percentage of successful tasks completed.

Tasks	Success / Fail %	Errors
Task1		
Task4		

*Success/Fail refers to whether a user is able to find the item*

*Tasks refer to the tasks listed above*

**Efficiency:** To measure the efficiency of the function, we will be watching closely on how fast they can find the item. Therefore, we will be watching the number of clicks and the amount of time they spent looking for the item.

Tasks	Number of Clicks / Pages	Amount of time task took
Task1		
Task4		

*Tasks refer to the tasks listed above*

To ensure consistency, we plan that all users will be completing the same tasks as outlined above.

**(Lickert) Satisfactory survey ( circle one ).**

**1. The pull down category was easy to use to find items in that category.**

Strongly Agree      Agree      Neither agree or disagree      Disagree      Strongly Disagree

**2. It was easy to find the book that I was looking for.**

Strongly Agree      Agree      Neither agree or disagree      Disagree      Strongly Disagree

**3. I am satisfied with the search function.**

Strongly Agree

Agree

Neither agree or disagree

Disagree

Strongly Disagree

**4. Comments (if any):**

**3. QA test plan - max 2 pages**

**Test Objectives:** Test the search feature for correctness under multiple scenarios. Ensure it adheres to the specification.

**HW and SW setup:** Server running on Google Cloud. Testing done with Firefox 71 and Chrome 78 on Fedora Linux. Reset database to a consistent and known set of posts. Website URL is <https://gatorstore.duckdns.org/>.

**Feature to be tested:** Searching for posts.

**QA Test plan:**

#	Title	Description	Input	Expected Output	Results
1	Search by Category	Perform search with empty search text and the "Furniture" Category selected	"Furniture" Category.  Empty search text	Output contains one item titled "IKEA Sofa"	Firefox: ✓  Chrome: ✓
2	Search text ignores non alpha-numeric	Perform test with only alpha-numeric and then add non alphanumeric to make sure they are the same	"All" Category.  First search text: "textbook"  Second search text: "~>textbook@"	Both outputs contain one item titled "MATH 324 textbook".	Firefox: ✓  Chrome: ✓

3	Search text validation	Ensure search text is limited to 40 characters	“All” Category.  Search text: 1. 39 characters 2. 40 characters 3. 41 characters	Search text contains: 1. 39 characters 2. 40 characters 3. 40 (NOT 41) characters	Firefox: ✓  Chrome: ✓
4	Search for item	Perform search for specific items	“All” Category.  Search text “longboard”	Output contains one result titled “Sector 9 longboard”	Firefox: ✓  Chrome: ✓

## 4. Code Review

### 1. Coding Style:

We focus on two major aspects in terms of the coding style. Firstly, code indentions should be controlled. For example, we applied the default indentation provided by the IDE (Visual Code Studio). When individual contributor pulls request to the official development branch, one or two other team members will check the codes to ensure no unnecessary empty lines and redundant comments. Secondly, we utilize the third-party npm tool Prettier to control the code formats of all the .js files. This tool helps us to break long one-line function down to several lines to make the code easier to maintain. To be more specific, we have the following rules to manage the uniformity of every member's coding style:

- Ensure the indention is two-space long
- Functions with multiple input parameters should break down to multiple lines
- Between two functions, there should be an empty line to seperate
- Object as input should break into multiple lines.

### 2. Code Review:

All the committed code should pass code review before they are officially merged into the main repo. Before everyone starts to work, team leader assigns part of the project to each member through github so that everyone knows their own job. Each member is encouraged to set up their own branch and develop on the local. Once the codes are implemented, pull requests will be initiated to the official development branch. Every pull request should provide detailed description about new updates and functions added to the project. Then other team members will review and provide suggestions to ensure the code quality.



## In-code review comments

```
133 +     photo: "/static/photos/breadmaker.jpg",
134 +     approval: "approved",
135 +     UserId: 1,
136 +     CategoryId: 3
137 +   },
138 +   {
139 +     title: "Water Boiler",
140 +     price: 60,
141 +     description:
142 +       "It is still functional after using for one year. Contact me if you want it",
143 +     photo: "/static/photos/waterboil.jpg",
144 +     approval: "approved",
145 +     UserId: 1,
146 +     CategoryId: 3
147 +   }
148 + },
149 + {}
150 }
```

1 comment on commit 4ea2ba9

Lock conversation



bessiejiang commented on 4ea2ba9 16 hours ago

+ @ ...

This looks good! Now we have more items in database.

## Advice on project structure

### adding views folder #4

**Closed** angelosolitario wants to merge 1 commit into [About · page](#) from [views-branch](#)

Conversation 1 Commits 1 Checks 0 Files changed 3



angelosolitario commented on Sep 22

+ @ ...

No description provided.

adding views folder

04eede3



janka102 commented on Sep 22

+ @ ...

This looks good. One question is where do we put our own profile/about page?

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

## Past Commits are available

### Posting Item Function Finished #49

Edit

**Open** KamyC wants to merge 5 commits into [OfficialDev](#) from [jinghan](#)

Conversation 0 Commits 5 Checks 0 Files changed 25

+241 -224

Commits on Dec 13, 2019

add more items to the database

KamyC committed yesterday

1



4ea2ba9



add more items in tutorial

KamyC committed yesterday



03c2096



Commits on Dec 14, 2019

## New pull request waiting for code review

Posting Item Function Finished #49 Edit

[Open](#) KamyC wants to merge 5 commits into [officialDev](#) from [j1ngnan](#)

Conversation 0 Commits 5 Checks 0 Files changed 25 +241 -224

**KamyC** commented 17 minutes ago

1. add some more items into the seeder in diff categories
2. revised the item model by adding two other attributes: userID and categoryID
3. in controller, add one more create function so that the item can be inserted into database with proper information provided
4. provide comparator in the router for sorting function.
5. add one more router "postItem" before directing to the posting page. This router is used for : user authentication, upload image to the server, create item, rename image path and its information stored in database
6. delete a folder "posting" which is not necessary in this project

**KamyC** added 3 commits yesterday

- add more items to the database 4ea2ba9
- add more items in tutorial 03c2096
- link posting to server, but cannot insert into database. add more if ... 8cef7b8

**bessiejiang** commented on 4ea2ba9 16 hours ago

This looks good! Now we have more items in database.

**Reviewers**

- janka102**
- angelosolitario**
- bessiejiang**

At least 2 approving reviews are required to merge this pull request.

**Assignees**

No one—assign yourself

**Labels**

None yet

**Projects**

None yet

**Milestone**

No milestone

**Notifications** [Customize](#)

[Unsubscribe](#)

You're receiving notifications because

## 5. Self-check on best practices for security – ½ page

### Major assets we are protecting:

1. Information system
2. User database
3. Individual user record

We protect the information system by using security procedures to prevent unauthorized users from accessing any information and protect sales item data and images. User database and individual user record will be protected by requiring users to authenticate themselves when logging into their account. Once users log into their account we can track system usage to increase security and prevent users from gaining access to confidential data.

### Password encryption in database:

Confirmed password is encrypted in database.

### Input data validation:

The search bar input validation has been confirmed to have up to 40 alphanumeric characters.

## 6. Self-check: Adherence to original Non-functional specs

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).

DONE

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers

DONE

3. Selected application functions must render well on mobile devices

DONE

4. Data shall be stored in the team's chosen database technology on the team's deployment server.

DONE

5. No more than 50 concurrent users shall be accessing the application at any time

DONE

6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.

DONE

7. The language used shall be English.

DONE

8. Application shall be very easy to use and intuitive.

DONE

9. Google analytics shall be added

DONE

10. No email clients shall be allowed

DONE

11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

DONE

12. Site security: basic best practices shall be applied (as covered in the class)

DONE

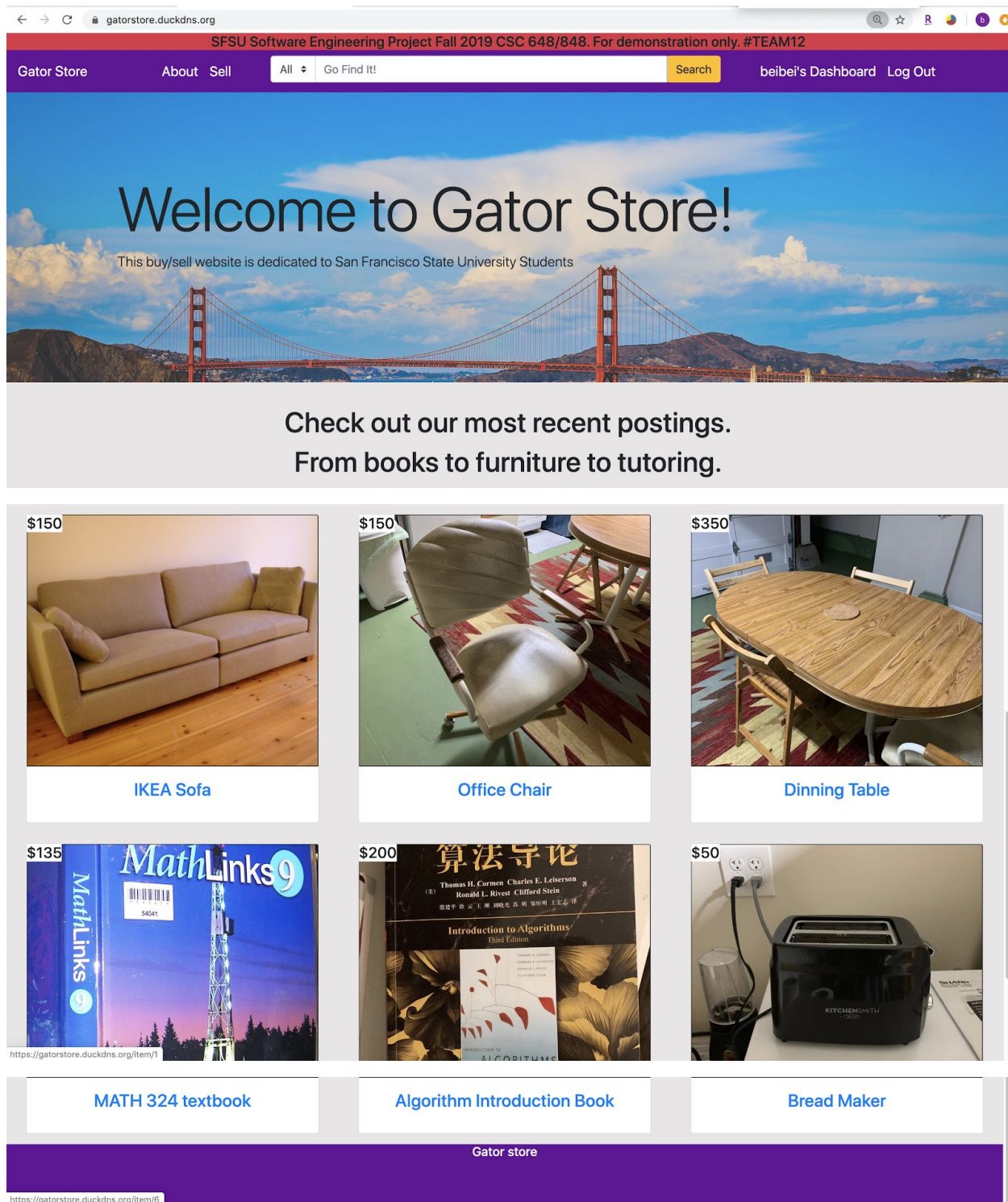
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

DONE

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).

DONE

### 3. Screenshots of actual final product as shown in the demo



Showing 3 items out of 3

Sort by: Most Recent

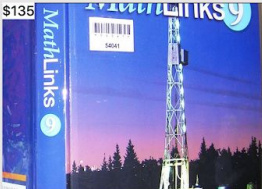
Filter price by: 101-150



IKEA Sofa



Office Chair



MATH 324 textbook

## IKEA Sofa - \$150



Seller:



Silvester Sam

Contact Seller

### Item description:

Selling my sofa I had for 3 semesters. Color is light gray. Clean, no stains.

Last updated: Mon Dec 16 2019 03:41:44 GMT+0000 (Coordinated Universal Time)

SFSU Software Engineering Project Fall 2019 CSC 648/648. For demonstration only. #TEAM12

Gator Store

About

Sell

All

Go

Search

beibel's Dashboard

Log Out

IKEA Sofa

Buyer beibel

Seller Silvester


Library type in time to contact

type in details

Cancel

Send

IKEA Sofa - \$150



Item description:

Selling my sofa I had for 3 semesters. Color is light gray. Clean, no stains.

Last updated: Mon Dec 16 2019 03:41:44 GMT+0000 (Coordinated Universal Time)

Silvester Sam

Contact Seller

Gator store

## Gator Store

### Post Items

Title

Price

Category

Description 

Type the description of item here...

Pictures 

Choose Files No file chosen

\* Your posting will show and approve after 24 hours.




Cancel

Submit



## Gator Store Admin Dashboard

### Items

Title	Category	Price	Description	Photo	Review
CSC 510 textbook	Book	70	Foundations of Algorithms Using C++ Pseudocode (Third Edition). ISBN: 0763723878. Good condition.		<a href="#">Review</a>
Japanese Dictionary	Book	10	New Japanese dictionary at a low price!		<a href="#">Review</a>
ipad	Electronics	500	Used ipad at a low price!!		<a href="#">Review</a>

Dashboard

Recently posted items:



Messages Your Items(2)

From	To	Item	Deal Location	Deal time	Message	Created Time	Reply
Ali	beibel	Japanese Dictionary			approved Good!	Tue Dec 17 2019 20:35:51 GMT+0000 (Coordinated Universal Time)	Reply

## Dashboard

### Recently posted items:



Messages

Your Items(2)

Title	Category	Price	Description	Photo	Approval	View	Delete
Japanese Dictionary	Book	\$10.00	New Japanese dictionary at a low price!			<a href="#">View</a>	<a href="#">Delete</a>
Ipad	Electronics	\$500.00	Used Ipad at a low price!!			<a href="#">View</a>	<a href="#">Delete</a>

## TEAM 12



BEIBEI JIANG



JESSE SMICK



JINGYIN REN



ANGELO SOLITARIO



JINGHAN CAO



MELISSA ENDANG

## 4. Screenshots of key DB tables

1 • **SELECT \* FROM csc648.Users;**

#	id	firstName	lastName	email	password	role	createdAt	updatedAt
1	1	Silvester	Sam	silvester-seller@ma...	\$argon2i\$V=...	user	2019-12-16 11:19:20	2019-12-16 11:19:20
2	2	Bella	Bob	bella-buyer@mail.s...	\$argon2i\$V=...	user	2019-12-16 11:19:20	2019-12-16 11:19:20
3	3	Ali	Admin	ali-admin@mail.sfs...	\$argon2i\$V=...	user,admin	2019-12-16 11:19:20	2019-12-16 11:19:20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

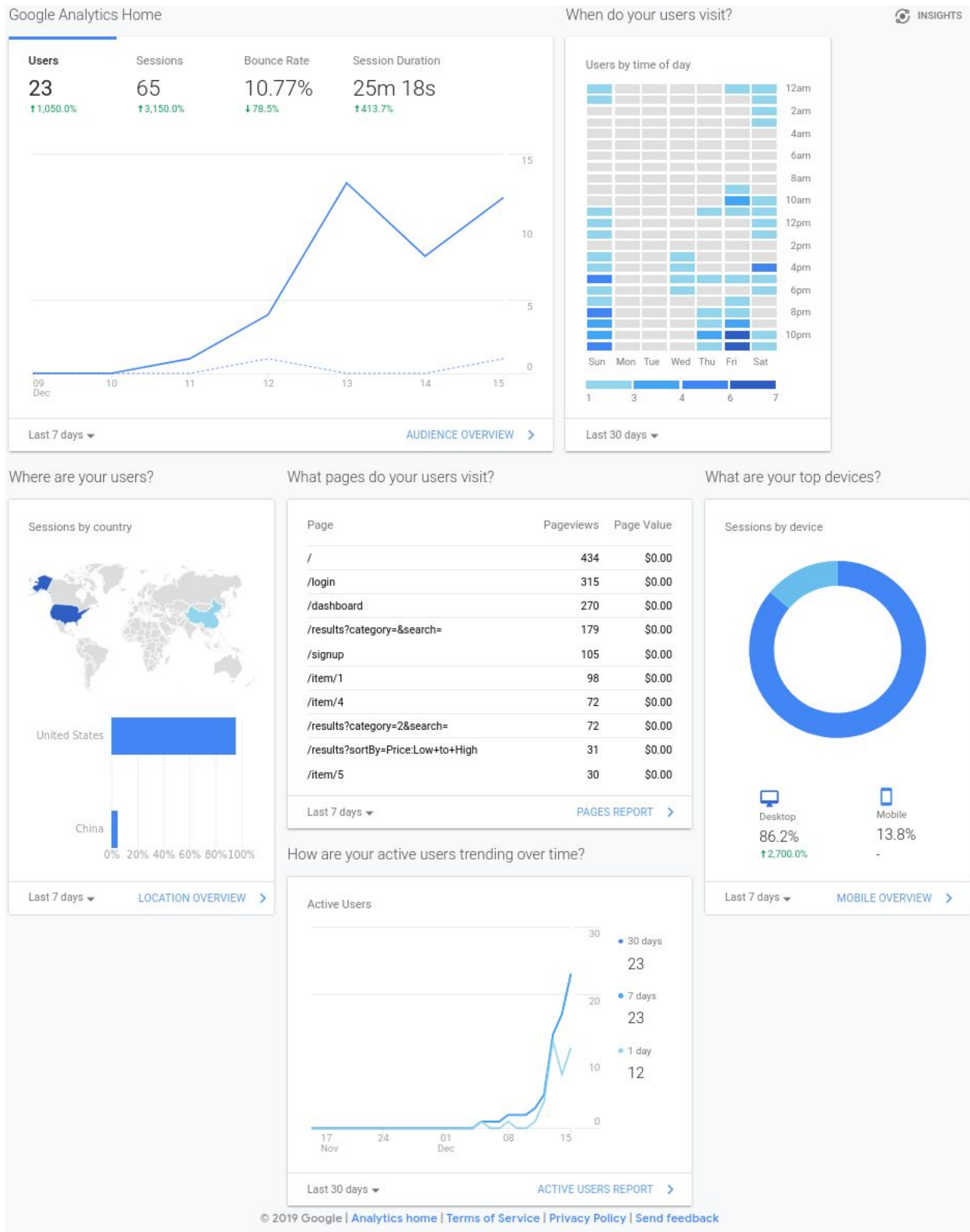
1 • **SELECT \* FROM csc648.Items;**

#	id	title	price	description	photo	tag	approval	UserId	CategoryId	createdAt	updatedAt
1	1	IKEA Sofa	150	Selling my sofa I had for 3 semesters. Color is light gray. Clean, no stains.	/static/photos/456776348_0d3644f...	NULL	approved	1	1	2019-12-16 11:19:20	2019-12-16 11:19:20
2	2	CSC 510 textbook	70	Foundations of Algorithms Using C... ISBN: 0763723878. Good condition.	/static/photos/23170998252_1c52e...	NULL	rejected	1	2	2019-12-16 11:19:20	2019-12-16 12:13:27
3	3	My book	1	Book for sale	NULL	NULL	rejected	1	2	2019-12-16 11:19:20	2019-12-16 11:19:20
4	4	Sector 9 longboard	35	Bearings are only one month old. It... Selling this because I got a boosted...	/static/photos/5697784828_41e0a5...	NULL	approved	1	4	2019-12-16 11:19:20	2019-12-16 11:19:20
5	5	MATH 324 textbook	135	Math book for MATH 324. No writin...	/static/photos/4764741238_6d041f...	NULL	approved	1	2	2019-12-16 11:19:20	2019-12-16 11:19:20
6	6	Algorithm Introdu...	200	Introduction to Algorithms. Chinese...	/static/photos/algorithmbook.jpg	NULL	approved	1	2	2019-12-16 11:19:20	2019-12-16 11:19:20
7	7	Office Chair	150	Office chair in good condition. Pick ...	/static/photos/chair.jpg	NULL	approved	1	1	2019-12-16 11:19:20	2019-12-16 11:19:20
8	8	Dinning Table	350	Very large table with three other ch...	/static/photos/dinningtable.jpg	NULL	approved	1	1	2019-12-16 11:19:20	2019-12-16 11:19:20
9	9	Bread Maker	50	I already have one bread maker, so...	/static/photos/breadmaker.jpg	NULL	approved	1	3	2019-12-16 11:19:20	2019-12-16 11:19:20
10	10	Water Boiler	60	It is still functional after using for on...	/static/photos/waterboil.jpg	NULL	approved	1	3	2019-12-16 11:19:20	2019-12-16 11:19:20
11	11	Artist Teacher	35	I can teach you water color painting...	/static/photos/artist_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
12	12	Astronomy Teacher	35	Do you want to explore the mystery...	/static/photos/astronomy_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
13	13	Math Teacher	35	Are you still bothered with the diffic...	/static/photos/math_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
14	14	Physics Teacher	35	Interested in the nature of this worl...	/static/photos/physics_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
15	15	Coding Teacher	50	I am an experienced coder and enr...	/static/photos/coding_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
16	16	Zoologist Teacher	10	I am NOT a body artist. I am simply...	/static/photos/zoologist_teacher.jpg	NULL	approved	1	5	2019-12-16 11:19:20	2019-12-16 11:19:20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

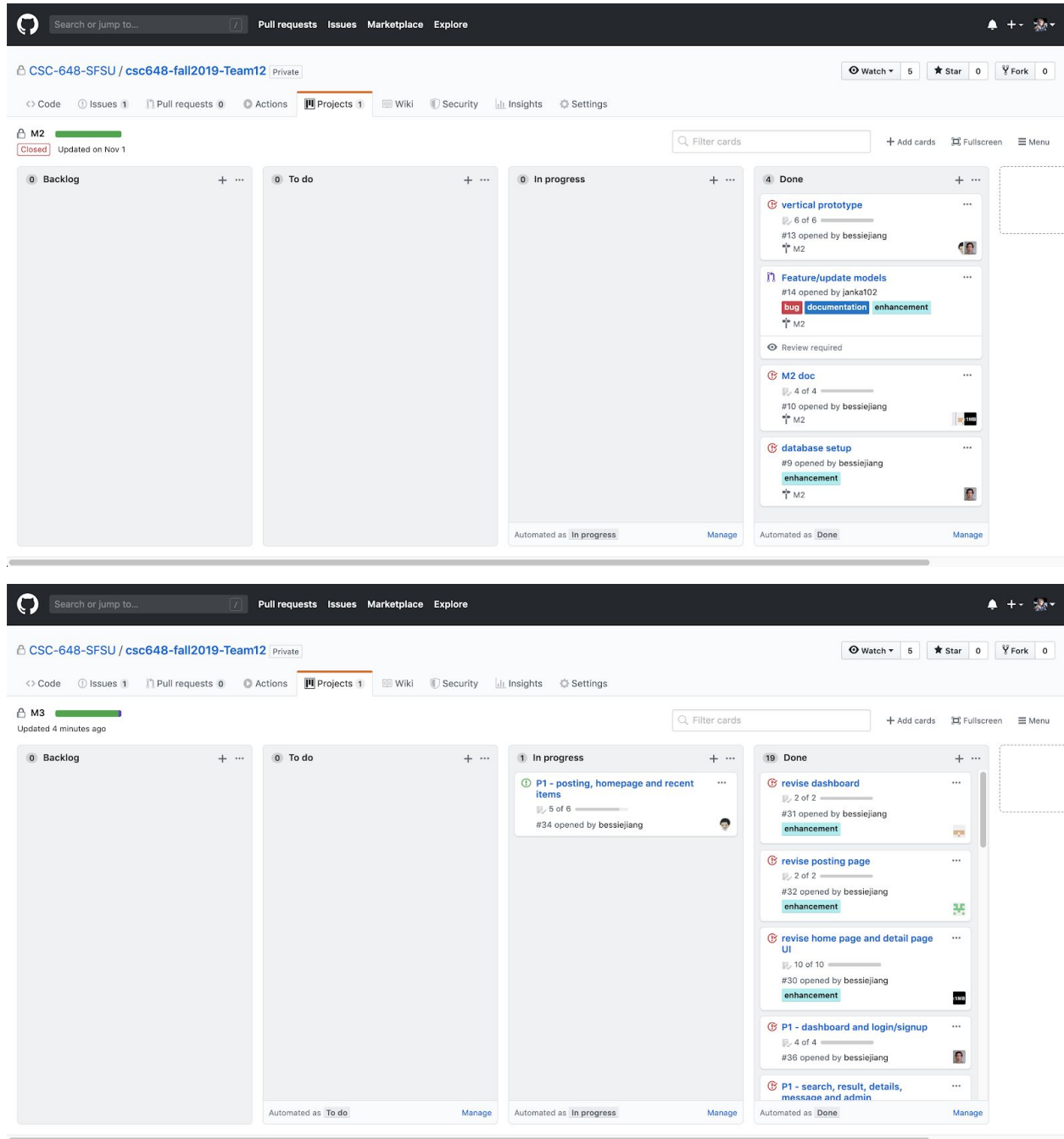
4 • **SELECT \* FROM csc648.Categories;**

#	id	name	createdAt	updatedAt
1	1	Furniture	2019-12-16 11:19:20	2019-12-16 11:19:20
2	2	Book	2019-12-16 11:19:20	2019-12-16 11:19:20
3	3	Electronics	2019-12-16 11:19:20	2019-12-16 11:19:20
4	4	Equipment	2019-12-16 11:19:20	2019-12-16 11:19:20
5	5	Tutor	2019-12-16 11:19:20	2019-12-16 11:19:20
*	NULL	NULL	NULL	NULL

## 5. Google analytics plot for your WWW site



## 6. Screenshots of your task management system (like Trello) showing a snapshot of your project management





## 7. Team member contributions



Jinghan Cao

Mon 12/16/2019 6:32 PM

Beibei Jiang; Jesse Lee Smick; Angelo Garingan Solitario; JinYing Ren; Melissa Endang ✕



Hi Team,

My contributions for "Gator Store" application are :

1. Initialize a sample file structure for prototype
2. Add a basic front-end page and basic bootstrap style to make it responsive for prototype
3. Fetch data from mysql database and display in front-end for prototype
4. add some more items into the seeder in diff categories
5. revised the item model by adding two other attributes: userID and categoryID
6. in controller, add one more create function so that the item can be inserted into database with proper information provided
7. add one more router "postItem" before directing to the posting page. This router is used for : user authentication, upload image to the server, create item, rename image path and its information stored in database

Total Pull Requests: 7

Total Commits: 16

Good luck for the finals!!!!

Jinghan



Angelo Garingan Solitario

Mon 12/16/2019 7:35 PM

Beibei Jiang; Jinghan Cao; JinYing Ren; Melissa Endang; Jesse Lee Smick ✕



Hi Teammates!!!

Here are my contributions:

As a Frontend Member,

- Helped design mockup pages
- In charge of the landing page, item details page, login and signup pages
- Helped with sorting and displaying the results based on price and date
- Implemented the navbar with the search bar
- Implemented captcha in login and signup

Total Commits: 39

Total Pull Request: 12

All the best with finals!!!

-Angelo





Beibei Jiang

Tue 12/17/2019 12:21 AM

Jesse Lee Smick; Angelo Garingan Solitario; Jinghan Cao; Melissa Endang; JinYing Ren



Hi team,

The following are my contributions:

- Team lead: organized discussions, created and arranged tasks, kept all members in the same pace, coordinated between frontend and backend.
- Github master: managed different branches, merged features and resolved conflicts, made sure the master branch works well.
- Documentation master: assigned documentation tasks to each team member and fixed format issues.
- In charge of message feature: Created Message schema and table, router and controller, and frontend template. The message feature allows buyer to contact seller.
- Worked on dashboard feature: Allow login user to view all the incoming and outgoing messages grouped by product, user can also reply the incoming messages.
- In charge of admin feature: Allow admin user to review all posted pending products, make approval or denial decisions.
- In charge of about page. Display the information of all team members.

Total Commits: 47

Total Pull Requests: 51

Best wishes with your finals!

Beibei



Jesse Lee Smick

Tue 12/17/2019 1:03 AM

Beibei Jiang; Angelo Garingan Solitario; Jinghan Cao; Melissa Endang; JinYing Ren ✨



Hi all,

Here are my team contributions as back-end lead:

- Setup development environment with virtual machines
- Setup Google Cloud to host our project
- Setup initial database items
- Get initial authentication for login/signup working
- Small fixes here and there

Total commits: 65

Total Pull Requests: 13

Happy Holidays,

Jesse Smick



Melissa Endang

Tue 12/17/2019 1:42 AM

Jesse Lee Smick; Beibei Jiang; JinYing Ren; Angelo Garingan Solitario; Jinghan Cao ✨



Hi Team,

My contributions as front-end are:

- In charge of the Gator Store dashboard
- Implemented the recently posted items from user
- Implemented messaging between buyer and seller
- Helped with sorting messages by items posted
- Added feature to reply to messages for responding to buyers on front end

Total Commits: 11

Total Pull Requests: 1

Good luck on finals everyone!

Best,  
Melissa



JinYing Ren

Tue 12/17/2019 2:00 AM

Jesse Lee Smick; Beibei Jiang; Angelo Garingan Solitario; Jinghan Cao ✉



Hi Team,

My contribution as front-end lead for the team project include:

- Created about page
- Worked on parts of each milestone document
- Designed post a listing page for our website
- Used feedback from CTO to change documentation of project
- Helped with the UI design

Total commits: 5

Total Pull Requests: 2

Good luck on your final exams!

Thanks,  
Jinying

## 8. Post analysis – lessons learned

### Team Lead Summary

After working with a team of bright minds, We have come to understand our Team 12 was self aware, confident in our ability and capable to achieve the feats. We as a team found CSC648-848 to be a great experience overall. We're proud of the work all members did as we all witnessed each other invest long hours in order to create "Gator Store". There were never any team quarrels and we completed the features we envisioned for our website. Walking away at the end of this project, we feel thankful for the experience and for the great individuals that we got to work with.

Issues we faced were as follows:

- - Managing time constraints
- - Technical issues
- - Issues with time management
- - Group ability to gather for code reviews
- - Maintaining git structure of branches

We believed the process of solving a problem can be divided into four steps:


- -Identify the problem

- -Gather information
- -Iterate potential solutions
- -Test solution

We discussed project work on “Discord” besides face to face meetings. Everyone was actively discussing potential solutions whenever one faced an issue. After gathering potential solutions, we kept trying things until we’ve made substantial progress on the problem. Often we did testing by hand on local: run our server and check if it contains all the elements we expect, then make the change.

Some issues seemed to be at times hard to overcome, we also used the techniques learned in the class - code reviews and massive amounts of debugging, we were successful to solve these issues and create our website with the abilities to help students buy and sell items.

Overall, we enjoyed working with our team and will look to improve upon the skills that we’ve learned in this class to create more great applications like “Gator Store” in the future.



Beibei Jiang

Tue 12/17/2019 2:41 AM

Jesse Lee Smick; Jinghan Cao; Angelo Garingan Solitario; Melissa Endang; JinYing Ren; Angelo Garingan Solitario

After working with a team of bright minds, I have come to understand my Team 12 was self aware, confident in their ability and capable to achieve the feats. We as a team found CSC648-848 to be a great experience overall. We're proud of the work all members did as we all witnessed each other invest long hours in order to create "Gator Store". There were never any team quarrels and we completed the features we envisioned for our website. Walking away at the end of this project, we feel thankful for the experience and for the great individuals that we got to work with.

Issues we faced were as follows:

- - Managing time constraints
- - Technical issues
- - Issues with time management
- - Group ability to gather for code reviews
- - Maintaining git structure of branches

We believed the process of solving a problem can be divided into four steps:

- -Identify the problem
- -Gather information
- -Iterate potential solutions
- -Test solution

We discussed project work on "Discord" besides face to face meetings. Everyone was actively discussing potential solutions whenever one faced an issue. After gathering potential solutions, we kept trying things until we've made substantial progress on the problem. Often we did testing by hand on local: run our server and check if it contains all the elements we expect, then make the change.

Some issues seemed to be at times hard to overcome, we also used the techniques learned in the class - code reviews and massive amounts of debugging, we were successful to solve these issues and create our website with the abilities to help students buy and sell items.

Overall, we enjoyed working with our team and will look to improve upon the skills that we've learned in this class to create more great applications like "Gator Store" in the future.

### **Team Member Reflection:**

- **Beibei Jiang:** I took CSC867 prior to taking this course which prepared me well for the challenges that group based projects provide. I feel that the biggest factor as to whether a group will succeed or not is team cohesion. Luckily, we had just that within my team as it was comprised of individuals who not only carried great computer science ability, but also didn't

hesitate to contribute or communicate with the rest of the team members. Our team was interactive from the first day so that we were easily able to divide our work efficiently. In this project, I also got an opportunity to learn about new technologies such as Node.js, ejs, Express, vagrant, which I had no prior experience with. Classes such as this one made me feel like we've gained real world software engineering experience. In conclusion, I am happy that I took this course.

- **Jesse Smick:** I learned a lot about team development in this course. It was a very good real-world experience especially since there were clearly defined roles and tasks. We did well making the plans and tasking out milestones. This is invaluable knowledge that would only otherwise be learned on the job and having this in college is a big advantage. Communication was a big success where we expressed concerns and recognition. All in all, I am proud to be a part of this team and project and think we all helped each other to learn a lot.
- **Jinying Ren:** This project is a great experience to see how a team works together and delivery an awesome product at the end. I learned a lot about how to build a website, create webpages and design them with UI. Also, I learned how the cooperation works between the front-end and back-end, and more knowledge of software stack. Throughout the project, I realized that communication is the main key to success, understanding of each member's ideas and helping each other to fix their problems. Team lead played an important role in planning and assigning tasks, so we all know what exactly the tasks we needed to complete and when to finish it by. Overall, I am so proud of my teammates and our final product.
- **Angelo Solitario:** As a front end developer, I really enjoyed being able to design the pages and see them through deployment. Working on a project with an environment like this and with a team was something new to me. It has made me realize that teamwork is very important. Working together not only makes the work fun and easier but also you can learn things that you didn't know before. One of the challenges that recently became a problem for is using the EJS template to manage the DOM which is impossible. One thing that I could have done better is use more technologies. In this case, I would try to explore ajax calls. Overall, I really enjoyed working with my team and I am thankful to have experienced this.
- **Melissa Endang:** Working as a front end developer has helped me learn a lot about how much details go in to building web pages. The beginning was the most difficult for me because this was the first time I worked together in a team for one project and building it from the very start. This project has made me realize that transitioning from working individually to a group of people can be easy if your teammates are willing to work with each other. My team members were all actively helping one another whenever any of us faces a problems, which helped things go more smoothly. Although, one of our biggest concerns was scheduling a time to meet up, we mostly discussed our process on Discord. Along the way, my main challenges were adjusting to working in a team and the documentation. As a front end, I would mainly be responsible for just the user interface and there were some things that could have been made easier to fix from the back end. For example, while creating the part of the dashboard where it holds the messages, being able to

sort the messages from user and buyer would make it easier to read. Overall, with the knowledge I have gained from this experience, I would definitely work on scheduling more time to meet up with with team members.

- **Jinghan Cao:** I have finally realized the importance of front-end and back-end cooperation. For some function, it requires back-end team to sort out a way to process data and reflect the data to front-end. In this case, people need to work closely and know the whole project thoroughly. Moreover, as a back end developer, I have noticed the power of MVC pattern in application development. MVC plus encapsulation can make the whole project development cycle faster. For example, I could only invoke the given API from the “model” so achieve data manipulation. I can also add more functions to the model which makes the development more flexible. In conclusion, I really appreciate having such a good team for this semester.