SW Engineering CSC648 / 848 Spring 2022

GATORent

Team 04		
Role	**Name**	**Email**
Team Lead	Samantha Saxton-Getty	ssaxtongetty@mail.sfsu.edu
Github Lead	Vishal Ramanand Sharma	vsharma5@mail.sfsu.edu
Front End Lead	Elyssa Mari Tapawan	etapawan1@mail.sfsu.edu
Back End Lead	Aaron Carlson	acarlson8@mail.sfsu.edu
Front End	Abdullah Sharaf	fabdullah1@mail.sfsu.edu
Front End	Siqi Guo	sguo4@mail.sfsu.edu
Back End	Janvi Patel	jpatel6@mail.sfsu.edu

March 05, 2022

Milestone 01

Milestone / Version	Date
Milestone 01 / Version 01	03.05.22

Table of Contents

1. Executive Summary

There are many students who cannot find the resources they need to properly study, work out, or research. Our app, GATORrent, provides various materials to help SFSU students find everything they are looking for on campus all online. GATORrent is an e-commerce app that allows students to rent different items, from school materials to gym equipment, as well as list items for others to rent. This will motivate students to use these products offered by the application to better their education and health.

The GATORent will have numerous functionalities and services. The main service the application will provide is allowing SFSU students and faculty to post or rent textbooks, sports gear, research devices all online. Each product that can be rented will show how much is in stock during the time of browsing, the time slot for the material, and the location on where to pick up the product. The time slot and location of the product will vary depending on which resources students want to rent out. As for the location, materials will all be on campus. All of this information will be provided on the application by the users selling the material and the school. Other important functionalities the app will include are a search option and categories to look for more specific items, and a sign up / sign in option to confirm users are SFSU students or faculty. These services will provide users with an easy and accessible way to find various resources all in one place. GATORent gives SFSU an opportunity to help their community by managing their resources within the application. Instead of going to different locations on campus to see if they can borrow or rent out materials in person, the GATORent will make this task a lot easier. It saves students and faculties the time and hassle in their already busy schedules. Also, this allows alumnis to give textbooks they no longer need to newer students who will utilize them more for classes. The unique aspect of GATORent is that all of the features and services in renting out material will be in one place and in one press of a button, where anyone can easily follow.

We are Team 4 from the Software Engineering class of Spring 2022 consisting of Aaron Carlson, Siqi Guo, Janvi Patel, Samantha Saxton-Getty, Abdullah Sharaf, Vishal Ramanand Sharma, and Elyssa Mari Tapawan, with Samantha being the team lead. Other roles consist of Aaron being the back-end lead, Elyssa being the front-end lead, and Vishal being the GitHub lead. The back-end operates the parts of the application that aren't accessed by a user such as data organization. The front-end works on the parts of the websites that allow users to interact with the web features, such as the application design. The GitHub lead ensures that there are no errors and that the application is functional. We are a group of aspiring students learning the aspects of team software development by making an application ourselves. Our goal is to work together to create the GATORent to help us experience and prepare ourselves for our future careers.

2. Personae and Main Use Cases

John is a SFSU student and a part time worker (buyer).

About John:

- On a budget
- Doesn't have a car
- Busy
- Basic WWW skills
- Wants to cut down on spending too much money shopping online
- Doesn't want to spend a lot of money on school materials

Goals and Scenario:

John needs to buy a textbook for a class. He wants to check if he can afford to buy the book or rent it. If he finds the book that he is looking for and the price is good, he can contact the seller.

Use Cases for John:

1. Search Item

John is looking for a textbook for one of his classes. He goes on to the site and searches the book by the name e.g. "software engineering 2020".

2. Create Account

John wants to create an account on the website. He goes on the website and clicks on the sign-up icon. He continues the sign up process and creates an account.

3. Add Item to Cart

John wants to add an item to his cart for future reference. He clicks the "add" button and is prompted to log-in before he can add the item to his cart. Once logged-in, he adds the item to his cart.

4. Purchase an item

John wants to purchase an item that he found on the website. He adds the item to his cart, proceeds the purchase process and gets to contact the seller.

Jennifer is a full time SFSU student (seller).

About Jennifer:

- Needs money
- Is a senior
- Has a lot of textbooks from previous semesters
- Very busy
- Basic WWW skills
- Doesn’t have a car

Goals and Scenario:

Jennifer decides to list a textbook for rent. She wants to quickly list the book on the website. If she finds a buyer for the book, she can close the textbook listing from the website.

Use Cases:

1. List Item for Sale / Rent

Jennifer wants to list an item for sale / rent. She goes to our website and fills out the item's description under “list item”, then publishes the item once the process is complete.

2. Create Account

Jennifer wants to create an account on the website. She goes on the website and clicks on the sign-up icon. She continues the sign-up process and creates an account.

3. Update Price for the Listed Item

Jennifer wants to update the price for an item that she has already listed. She goes to our site and searches the item by name, once the item is found she proceeds with the update process and updates the item.

4. Close Item Listing

Jennifer wants to close a listed item. She goes to our site and searches the item by name, once the item is found, she proceeds to the closing process and removes the item.

Mike is an administrator.

About Mike:

- Likes problem solving
- Has basic database management skills
- Doesn’t live on campus
- Doesn’t like working on multiple problems at once

Goals and Scenario:

Mike needs to disable a user from the website. He wants to check if he can find the user on the database. If he finds the user on the database, he would like to remove the user but he also wants the items they have listed deleted when the user is deleted.

Use Cases:

1. Access Database

Mike wants to access the database system. He goes to the database system application and enters the system using his credentials.

2. Disable User

Mike wants to disable a user. He goes to the database system, and searches the user by user name e.g. “user1”, and proceeds to remove it from the database.

3. Remove Products

Mike wants to remove a listed item. He goes to the database system and searches the item by name, once the item is found, he proceeds with the closing process and removes the item.

4. Update User Information

Mike wants to update a user’s information in the database. He goes to the database system, and searches the user by user name e.g. “user1”, and proceeds to update it on the database.

3. Main Data Items and Entities - Glossary / Definitions

1. **User:** Individuals utilizing the product and application.
    a. **Administrator:**
    This user has access to all privileges given to registered and unregistered users as listed below. In addition, they are able to modify permissions for all users as well as edit existing posts or remove them regardless of it being their own post. This user also has access to modify the database for the application.
    b. **Registered User:**
    This user is allowed to search and view existing items listed as well as post, remove, and edit items of their own. They are able to rent / purchase items. This user will be required to have a username and password to login or register. This user is required to have an SFSU email (faculty or student).
    c. **Unregistered User:**
    This user is allowed to search and view existing items. They are not required to login or register unless they want to post or rent / purchase.
2. **Data Items:**
    a. **Items:**
    This is the list of products provided by users that are utilized by the users to rent or purchase. All items have their specific item details, price, condition, who listed the item and when, and photo(s). This list of items can be found in functional requirements.
    b. **Categories:**
    All items are categorized into three categories, general, art & science, and athletic equipment. These items will have tags indicating which category they fall into.
    c. **Transaction:**
    This item logs transaction details such as, but not limited to, the time it was rented / purchased, the length the item has been rented for, the user(s) involved, and the cost, etc.

4. List of Functional Requirements

1. **Home Page**
    a. This will feature all available products in cards with their price.
    b. There will be a search bar at the very top to search between products.
    c. You can filter by different categories:
    General Equipment:
    Laptops
    Textbooks / Books
    Calculators

    Art & Science Equipment:
    Lab Coats
    Beakers
    Art Essentials

    Athletic Equipment:
    Badminton Racket
    Basketball
    Bike
    d. Sorting Functionality between products.
    e. The nav bar will feature a Sign In / Sign Out button and will hold the User Information.
    f. We can maintain a cart to group checkout multiple items.
2. **Product Page**
    a. This will showcase additional technical information regarding the product.
    b. It can showcase different sellers / renters to choose from.
3. **Checkout Page**
    a. This shall be used for entering the contact information.
4. **Login / Sign Up Page**
    a. Allows the user to login or register for an account.
    b. If the user has any items for sale / rent, they see it in another tab.
5. **User Access Levels:**
    a. **Unregistered User:**
    Shall only browse through the available listings.
    Shall see the individual product page but will not be able to rent out items.

    b. **Registered User:**
    Shall browse through the available listings.
    Shall be able to view individual product pages and will be able to contact

the owner to rent it out.
The registered user shall be able to post listings to rent.
They shall be able to edit their existing or even delete them if required.

c. **Admin:**
Shall be able to see all listings.
Shall be able to edit / delete listings if they see it fit to be done so.
Shall be able to delete a User account if the user was reported.
Shall be able to access and modify the application database.

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.
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.
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 email 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, Spring 2022. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

6. Competitive Analysis

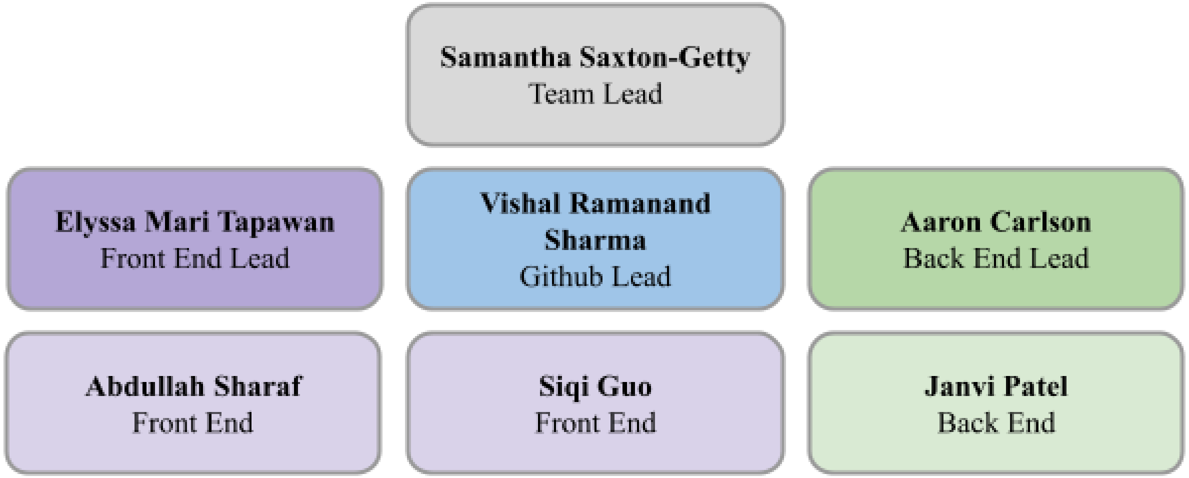
Feature	SFSU Library	Public Library	Amazon	GATORent
Store Pickup	7 working days	15 working days	3 - 5 working days	Same day
Marketing Strategies	Email / Website	Email	Website / Social Media	Website / Social Media / Email
Student Satisfaction	★	★★	★★★	★★★★
Product Recalls	NONE	YES	YES	YES
Strength	Cheap	Huge Audience	Long Term	Diverse Products
Weakness	Limited Resources	Unresponsive Customer Service	Expensive	Dependent on SFSU Community

Our company GATORent provides same day pickup services as compared to other companies, where users need to book in advance for the equipment. We have many ways to show our products, such as our website, social media, posters, email, etc. In addition, we predict a 4 star student satisfaction as the services we provide will be very convenient for students. We have a variety of products available including, but not limited to, laptops, textbooks, calculators, lab coats, badminton rackets, basketball, etc. In comparison to other companies, we have a diverse range of products. Along with our diverse range of products, we also offer a short and long term rental service with same day pick up unlike any company out there. The only disadvantage here is that we are completely dependent on the SFSU community as it is only available for SFSU students and faculty for now.

7. High-level System Architecture and Technologies Used

- **Server Host:** Amazon AWS 1vCPU 2 GB RAM
- **Operating System:** Ubuntu 20.04.3 LTS Server
- **Database:** MySQL v 8.0.28
- **Web Server:** NGINX 1.21.6
- **Server-Side Language:** Javascript
- **Additional Technologies:**
- **Web Framework:** Express
- **IDE:** VS Code
- **Web Analytics:** Google Analytics
- **SSL Cert:** Lets Encrypt (Cert Bot)
- **SASS:** 1.49.7

8. Team and Roles



Samantha Saxton-Getty
Team Lead

Elyssa Mari Tapawan
Front End Lead

Vishal Ramanand Sharma
Github Lead

Aaron Carlson
Back End Lead

Abdullah Sharaf
Front End

Siqi Guo
Front End

Janvi Patel
Back End

9. Checklist

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

  ISSUE
  The current issue is team member(s) not attending meetings or being engaged. This is being handled with the Professor through the Team Lead.

- Team found a time slot to meet outside of the class.

  DONE / OK

- Back end, Front end leads and Github master chosen.

  DONE / OK

- 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 / OK

- Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.).

  DONE / OK