SW Engineering CSC648 / 848 Spring 2022

BusyGator

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 31, 2022

Milestone 01

Date	Version
03.05.22	First Version 01 submitted
03.31.22	Revision completed per instructor feedback Version 02

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 application, BusyGator, provides various materials to help SFSU students, staff, and faculty find everything they are looking for on campus all online. BusyGator is an e-commerce application that allows students, staff, and faculty to buy different items, from school materials to gym equipment, as well as list items for others to buy. This will motivate the SFSU community to use these products offered by the application to better their education and health.

The BusyGator will have numerous functionalities and services. The main service the application will provide is allowing SFSU students, staff and faculty to post and /or buy textbooks, sports gear, research devices, and much more all online. Each product that can be bought will show how much is in stock during the time of browsing, the price of the material, and the user selling the item. The BusyGator will provide a map pin-pointing various safe areas to pick up the product all around the SFSU campus. Other important functionalities the application 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. BusyGator 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 buy materials in person, the BusyGator will make this task a lot easier. It saves students, staff, 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 BusyGator is that all of the features and services in buying and selling 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 BusyGator to help us experience and prepare ourselves for our future careers.

2. Personae and Main Use Cases

Personas:

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. If he finds the book that he is looking for and the price is good, he can 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 sale. 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.

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. Search Item

John is looking for a textbook for one of his classes. He goes on to the site and enters what he is looking for in the search bar. The results for his search show up on the page. He scrolls through the search results and picks the items that he is looking for.

2. Create Account (buyer)

John is not able to log in because he doesn't have an account. To create an account, he goes into the signup page on the website. He is promoted to fill out the signup form and he gets his new account right after he fills out his information. Now when he tries to login, he is successful.

3. Add Item to Cart

John finds the book that he is looking for, and when he tries to add it to the cart, he is asked to login/ signup. John logs in to the website and tries adding the item to the cart for the second time, this time he is able to add the item to his cart because he is logged in.

4. Buy an Item

John wants to buy an item that he found on the website. He wants to contact the seller by sending them a message. When he tries to send the seller a message, he is prompted to log in. When he logs in, he is able to send the message.

5. List Item for Sale

Jennifer wants to list an item for sale. She goes to our website to list the item. When trying to list the item, she is asked to log in / sign up. After logging in, she is able to list the item and its details. However, the item wouldn't be visible on the website right away because it will take some time for the approval process.

6. Create Account (seller)

Jennifer is not able to list an item for sale on the website because she doesn't have an account. She wants to create an account on the website. Jennifer goes into the signup page on the website. She fills out the signup form and gets his new account. Now when she tries listing an item for sale, she is allowed to do so.

7. Update Price for Listed Item

Jennifer wants to update the price for an item that she has already listed. First, shs logs into her account. After that, she searches for the item that she wants to update. When Jennifer finds the item, she proceeds with the update process and updates the item.

8. Close Item Listing

Jennifer doesn't want to sell one of the items that she had listed previously. She wants to go on to her account to find the item, she is asked to login. After logging in, she searches for the item, and proceeds with the closing process.

9. Access Database

Mike wants to access the database system because he wants to review the data. To access the database, he needs to log into the database system application with his admin credentials. When he is logged in, he is able to see that database..

10. Disable User

Mike wants to disable a user by removing them from the database system. He logs into the database system, and searches the user. When he finds the user that he wants to remove, he proceeds to delete it from the database.

11. Remove Products

Mike wants to remove a listed item from the database. He logs into the database system and searches for the product that he wants to remove. When he finds the item, he proceeds with the closing process and removes it.

12. Update User Information

Mike wants to update a user's information in the database. He logs into the database system, and searches the user. When he finds the user, he proceeds to update it on the database.

13. Approve Items

Mike wants to approve items to be published on the website. To approve the items, he needs to be logged into the database system application. When he logs in, he can start approving the items, based on the terms and conditions.

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. Administrators are required to approve products before they go live.
    b. **Registered User:**
    This user is allowed to search, filter, and view existing items listed, message the sellers, and post, remove, and edit items of their own. They are able to buy items. In addition, this User can report items to Administrators for review. 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, filter, and view existing items. They are not required to login or register unless they want to post, buy, or message the seller.

2. **Data Items:**
   a. **Categories:**
   All items are categorized into five categories. These individual data items will have a key indicating which category they fall into.
      - Electronics
      - Pets
      - Home
      - Recreational
      - Books

   b. **Items:**
   This is a list of data items available for buy and sale by the User.
      - **Electronics:**
        - Batteries
        - Calculator
        - Headphones
        - Kindle
        - Laptop
        - Laptop Charger
        - Phone Charger
        - Television
      - **Pets:**
        - Pet Bowl
        - Collar
        - Pet Bed
        - Food
        - Leash
        - Toy
        - Bone
        - Potty Pads
      - **Home:**
        - Mattress
        - Shower Curtain
        - Pillow
        - Lamp
        - Sofa
        - Chair

- **Recreational:**
    - Tent
    - Sleeping Bag
    - Fishing Pole
    - Bike
    - Badminton Racket
    - Tennis Racket
    - Basketball
- **Books:**
    - Dictionary
    - Encyclopedia
    - Atlas
    - SFSU Textbook
    - Literature Book
    - E-Book Code

c. **Transaction:**
This item logs transaction details including, but not limited to, the time it was bought, the user(s) involved, the cost, etc.

d. **User Registration:**
This item holds all of the registration information of the users including their username, hashed password, and permission level.

4. List of Functional Requirements

1. Unregistered User

1.1 Unregistered Users will be able to search posts.

1.2 Unregistered Users shall be able to filter search results based on categories. This function will help to narrow down posts that fit what the user is looking for.

1.3 Unregistered Users shall be able to register for an account. The registration email can only be a SFSU email.

1.4 Unregistered Users shall not contact the seller. They will not be able message the seller unless they have a registered account.

2. Registered User

2.1 Registered Users can perform all functions that Unregistered User can along with some additional functionalities.

2.2 Registered Users can create posts. This function is required to help Users sell their products.

2.3 Registered Users can contact the seller of the post via in-site messaging after buy. This function is required for transactions between users.

2.4 Registered Users shall be able to edit the post after it has been submitted.

2.5 If the product is not available anymore then the User can delete the post.

2.6 The product can be reported to the Administrator if the product isn't accurate to the description provided / images displayed. In this case the seller shall be reported to the Administrator.

2.7 Registered Users can log in to their accounts to buy / sell items.

2.8 Registered Users can log out of their accounts. This is to enhance security in case they are accessing the account from a public place.

2.9 Registered Users can request to reset their password in case they forget their password.

2.10 Registered Users can propose the meeting location and time to complete the transaction.

2.11 Registered Users can look at their transaction history in case they want to revisit a post.

3. Administrator

3.1 Administrators can perform all functions that a Registered User can.

3.2 Administrators shall be able to delete or edit posts made by the User, if they do not follow the terms of service or they are offending in any capacity.

3.3 Administrators will be able to approve a post before it goes live, so that none of the posts go against the terms of service.

3.4 Administrators will be able to delete / ban a User if they consistently fail to comply with the terms of service.

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	BusyGator
Store Pickup	7 working days	15 working days	3 - 5 working days	Same day
Safe Locations	YES	NO	YES	YES
Marketing Strategies	Email / Website	Email	Website / Social Media	Website / Social Media / Email
Student Satisfaction	★	★★	★★★	★★★★
Map of Locations	NO	NO	YES	YES
Diverse Products	NO	NO	YES	YES

Our company BusyGator provides same day pickup services as compared to other companies, where users need to book in advance for the equipment. We also provide safe locations for our company to make it more convenient for them and ensure their safety. The application will provide a map of the safe locations for customers to know the pickup location of the items ahead of time. 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. In comparison to other companies, we have a diverse range of products. We have a variety of products available including, but not limited to, laptops, textbooks, calculators, lab coats, badminton rackets, basketballs, etc.

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

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