

Software Engineering
CSC648/848 Section 02
Fall 2019

Gator Joe's

Milestone 1

Team 13

Piero Calenzani (Team Lead): pcalenzani@gmail.com

Grant Kennedy (Front-end Lead)

Mark Hammond (Back-end Lead)

Chuhao 'William' Yang (Github Master)

Varun Sura (Document Master, Front-end)

Zhuoxin Tan (Back-end)

Date Submitted	Date Revised	Description
10/02/2019	10/04/2019	First draft
10/11/2019		M5 draft

1. Executive Summary:

Gator Joe's is the online marketplace for SFSU students by SFSU students. Current online solutions profit from students by doing the bare minimum to meet the needs of students. Students who need to buy books for their classes are required to do unnecessary research before even beginning to search for the items they need. Students attempting to sell their no longer needed items or provide services such as tutoring have to worry about scams and unsafe situations. Gator Joe's is explicitly for SFSU students. All listings will be limited to the SFSU community, without having to worry about unseemly actors from outside.

In Gator Joe's, users will be able to buy and sell textbooks from other students. Students will not only be able to search using the traditional metrics of ISBN, title, author, etc. Students will also be able to search for textbooks by class number and section (BIO 100-2, for example). All content on the site will come from members of the SFSU community. A SFSU email address will be required to buy and sell. Services will not be limited to buying and selling books. Students will also be able to list items for their living spaces. Because the living spaces of students tends to be limited, students will be able to list and search for the dimensions of the items listed. Finally, meet-ups will be able to be arranged on the site, taking advantage of SFSU's safe exchange sites.

Gator Joe's is a team of undergrad (and one graduate) students coming together to make lives easier for our gator community. We are an international group coming together for local solutions.

2. Personae and main Use Cases:

Undergrad Student – Lily (Unregistered Guest)

- Bored and impatient
- Careless, skips classes frequently
- Drawn towards instant gratification
- Tech-savvy, very active on the internet and social phone apps.
- Low attention span, expects quick results
- Not wealthy, very mindful of prices over other aspects

Graduate Student – Noah (Unregistered Seller)

- Studious and responsible
- Trying to make extra money with minimal effort
- Good internet skills and awareness
- Working part time while studying
- Needs to complete tasks within few minutes

Returning Student – Martha (Registered Buyer)

- Easily frustrated with technology
- Non-native English speaker
- Very poor web skills, relies on her children's help often
- Balancing work, school and children
- Slow reader, does not understand common internet terms

Part-time worker – Steven (Hired Administrator)

- Middle aged and short-tempered
- Has full-time job on top of this one
- Minimal internet browsing skills
- Job must be finished within two hours on each workday
- Quality of work dependent on ease of use

Use Case #1:

Lily is an undergrad student currently taking her general education courses. Due to her carelessness, she misplaced her backpack and needs a new one. She heard of this application from a classmate and decided to look through the listings for a new backpack and other miscellaneous school supplies she must replace. Lily opens the application and searches for backpacks. She sorts by price and quickly browses through the picture of each search result until finding one she wants. After checking the listing and moving to purchase, she is prompted to register. The registration is quick and asks for a small amount of info before prompting her payment. After paying, she is given the seller's email and contacts them to agree on an exchange spot.

Use Case #2:

Noah is a graduate student currently trying to complete his Master's in Biology. He often looks for good deals on other internet sites and resells them locally on craigslist. He is looking to try this application where he hopes to continue reselling items for a small profit to local students. He wants a quality user experience where he can advertise buyers towards his listings. Noah opens the application and navigates to the seller page where he fills out the all the information for a nightstand he is selling. He is able to upload multiple pictures showcasing his item and also lists the furniture dimensions. After finalizing his listing, he is prompted to register and promptly fills it out. He is met with a confirmation that tells him his listing is waiting for approval.

Use Case #3:

Helen is a middle-aged woman returning to school to follow a new career in accounting. She has two children and works full-time to support them while studying. Her required texts are expensive and have inconvenient shipping schedules for her. She has previously used, and registered to, this application when purchasing a graphing calculator. Helen opens the application and browses under the textbook category. She fills in her class' id and section into the sorting options and quickly finds the textbook needs. She opens the listing and is prompted to register or log. She fills in her login credentials and is taken to the payment page. After paying, she is given the seller's email to decide on a meeting spot.

Use Case #4:

Steven is an adult hired through a third-party to moderate and approve listings on the application. He has taken this job for supplemental income on top of his office job. His duty is to log into the application daily and approve or deny all pending listings. After opening the application, Steven logs in from the home page and is met with the first pending listing of the day. He checks the title, description and all uploaded images to ensure they are consistent and appropriate. He approves the listing and moves on to the next, which has an inappropriate picture attached. This time, Steven denies the listing and writes a quick sentence to explain why. When he finishes going through all listings, he notified that his work is done for the day.

3. List of main data items and entities:

1. Users:

- a) **Guest users:** Can browse through all listings. Does not need to login/register.
- b) **Registered User – Buyer/Seller:** User with access to existing account. Can be buyer, seller, or both.
 - **Username:** Chosen name or alias by account owner.
 - **Password:** Key or phrase providing access to account.
 - **Email:** University email used to register account.
 - **Student ID:** University ID number input by user at registration.
 - **Phone:** Phone number associated with account.
- c) **Registered User - Administrator:** Inherits all privileges from registered user. Administrator users can approve or deny listings created by sellers and provide optional feedback.

2. Website Content:

- a) **Listings:** A for-sale post for various items created by registered users.
 - **Title:** Selected title for listing.
 - **Description:** Description of item to be sold.
 - **Class:** Specified class, if applicable.
 - **Section:** Specified section, if applicable.
 - **Dimensions:** Specified dimensions, if applicable.
 - **Price:** Asking price of listing chosen by seller.
 - **Category:** Designated category of listing.
 - **Photo(s):** Image of listing, can be multiple.
- b) **Categories:** Classification grouping to facilitate browsing of listings.
 - **Name:** Category description/title.
 - **Amount:** Number of items currently listed in category.

4. Initial list of functional requirements:

General:

1. Website shall display listings on home page.
2. Website shall show the images of listings.
3. Website shall have a category bar for browsing.
4. Website shall show the prices of listings.
5. Website shall display any description for listings.

Unregistered user:

6. Unregistered users shall be able to search.
7. Unregistered users shall be able to view listing info.
8. Unregistered users shall be able to register.
9. Unregistered users shall be able to search and browse products.

Registered user:

10. +All Unregistered User functions.
11. Registered user shall be able to buy and sell.
12. Registered user shall be able to log in and log out.
13. Registered user shall be able to view order status.
14. Registered user shall be able to view order history.
15. Registered user shall be able to add and remove items from their shopping cart.
16. Registered user shall be able to enter the information of their listings.

Admin:

17. +All Registered User functions.
18. Admin shall be able to monitor all user listings.
19. Admin shall be able to deny and approve listings.
20. Admin shall be able to cancel orders and transactions.

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 e-mail 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:

Feature	Amazon	Craigslist	eBay	Our Product
Textbook search by Class	-	-	-	+
Furniture search by dimensions	-	-	-	++
Arrange meet-ups	-	+	-	+
Browse	++	-	++	+

Key: “-” = Not present, “+” = Present, “++” = Upgrade

From the above competitive-analysis we can see the advantages our product will offer over its competitors such as Amazon, Craigslist, and eBay. First, the textbook search functionality will allow students to search for their textbooks by department and class number. For example, it will take only two selections to find textbooks for BIO 100 by selecting “BIO” from a menu, and “100” from a secondary menu. This eliminates the need for students to manually search books by author, title, ISBN, and the correct edition. In addition to text search, users can do a refined Boolean search by total dimensions. This allows users to search for furniture such as a carpet or cabinet by length and width. This feature is not present in any of the above competitors’ refined search. Our service is also exclusive in offering a meet-up feature to facilitate students to meet each other at a place and time for same-day transactions. Our front page will feature images and listings to greet first time users unlike Craigslist.

7. High-level System Architecture and Technology:

1. Web Server
 - 1.1. Amazon AWS EC2
 - 1.2. Ubuntu 16.04
 - 1.3. Apache 2.4
2. Framework
 - 2.1. HTML, CSS, Python 3.6.8
 - 2.2. Python Flask 1.1.1
 - 2.3. Bootstrap
 - 2.4. MySQL
3. APIs
 - 3.1. Google Analytics
4. Supported Browsers
 - 4.1. Chrome: version 76, 77
 - 4.2. Mozilla Firefox: version 68, 69
 - 4.3. Safari: version 12.1
5. Other tools
 - 5.1. PyCharm
 - 5.2. Workbench
 - 5.3. GitHub
 - 5.4. Visual Studio Code

8. Team members:

- ❖ Piero Calenzani (Team Lead)
- ❖ Mark Hammond (Back-end Lead)
- ❖ Grant Kennedy (Front-end Lead)
- ❖ Chuhao 'William' Yang (Github Master)
- ❖ Varun Sura (Front-end Programmer, Document Master)
- ❖ Zhuoxin Tan (Back-end Programmer)

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