

SW Engineering CSC648/848 Fall 2019

“Buy and Sell”

Section 02 Team 11

Milestone 1 v.2

Oct. 12 2019

1. **Daniel Janes** - Team Lead djanes@mail.sfsu.edu
2. **Gem Lagman** - Github master / Back-End
3. **Monique Marti** - Front-End Lead
4. **Raya Farshad** - Back-End Lead
5. **Pramish Dhakal** - Front-End

Revision	Date Submitted
Milestone 1 v.1	Oct. 2 2019
Milestone 1 v.2	Oct. 12 2019

1. Executive summary

“Buy and Sell” will be the go-to web marketplace for SFSU students to buy, sell, and discover great items locally. Buy and Sell is a unique platform that combines the best features of market leading competitors while offering the personalization for each unique college market, such as browse textbooks based off the class. Only verified students will be able to register, ensuring a safe community where students will flock to. Students will spend less time looking for their required textbooks and save money in the process.

Students will be able to list as many items as they want for free at the same time. Buyers can browse available items by category or by classes offered at SFSU. With our team’s local market expertise, Buy and Sell can provide a level of customization to the buying process that no other online marketplace has ever offered, such as suggesting safe meeting places on or near campus. Buy and Sell can offer prescreen safe locations in the SFSU area where buyers and sellers can meet up at in order to safely exchange their items.

Buy and Sell provides a safe online messaging service between the buyers and the sellers to help facilitate communication between the two parties. Admins will approve each potential post ensuring that they comply with the sites Terms of Service. Admins will also be available to help resolve any disputes that might arise. Buy and Sell has plans to expand to other college markets as the initial SFSU market rises in popularity.

Our team is comprised of distinguished SFSU computer science students with years of combined experience designing and building websites. This team of highly motivated individuals are ready to prove themselves by designing and implementing a functioning marketplace for SFSU students.

2. Personae and Main Use Cases

a. Personae



Joe is a first year student at SFSU. He is very new to San Francisco. He doesn't have many friends and doesn't know many people. He loves to browse in his free time. He is majoring in Sociology. He wants to buy second hand furniture to save money and time.



Alice is a second year student at SFSU. She likes to make money extra money in her spare time. She plays basketball during her leisure time. She participates in different student organizations workshops. She loves new technology and is always an early adopter.



Jack is a third year student at SFSU. He is fairly lazy and prefers to stay to himself. He loves coffee and energy drinks. He prefers not to talk to people in real life and would rather type than talk.



Amy is a senior student at SFSU. She loves to be organized. Amy is looking for a new job in New York. She doesn't like old materials. She loves to watch movies. She is financially struggling. She writes poems in her free time.



Mary is an SFSU alumni. She loves to help people. She is on time everyday and listens to every student's problems. She plays a good team lead role and knows how to resolve the conflict.

Main Use Cases

1. Unregistered User

Joe is a new student at SFSU looking for a desk and doesn't want to go far to find one because he doesn't have a car to transport it. They find the site and searches for a desk in the furniture category. They find what they want and clicks contact seller. Joe is prompted to register to continue with the purchasing process. Joe inputs their email and password and is now a registered user and can proceed to messaging the seller. They click then proceed to communicate with the seller about price and pickup availability.

2. Registered User

Alice is a registered user on our site and wants to sell her books for computer science. They don't have time to wait in line to sell them back to the bookstore at school. So, they decide to list the books online instead. They log into their profile on the website, enters her email and password to verify their login and then it directs them to their profile. They click on "sell item" option and lists their books for sale on our site under the correct class category. They submit their listing to be approved by an administrator before it is published on the website.

3. Buyer (Actively looking for something to buy)

Jack is a student at SFSU and is looking to buy some books for their class, which are not very expensive. They don't have a car nor do they use public transportation. They find our site and browse the books category to see if there is anything they are looking. He finds that someone is selling the books for the class they are currently taking and clicks to view item. When they attempt to contact the seller, they are prompted to register. Once they register, they can continue to contact the seller and negotiate price as well as a pick up location and date.

4. Seller (Actively selling)

Amy is a senior at sfsu and is moving out of their apartment on campus. They need to get rid of as much as they can because they cannot take it back home on the airplane. They find our site and clicks on sell items option to sell their furniture and books. They create a listing in the furniture category and books category, but before they can publish the listing they are prompted to register. They submit their listing to be approved by an administrator before it is published on the website.

5. Admin (Approve/deny posts kind of admin. Not the coder/programmer of the site)

Mary is the admin for our site and approves or rejects user published listings. Their job is to make sure that nothing obscene is posted on the website. They log into their account, and it directs them to their admin profile. They enter their dashboard to view new published listings waiting for approval. They approve or deny the listings after verifying that none of the content violates the site's terms of usage policy. They are visible immediately after their approval.

3. List of Main Data Items and Entities

- **Unregistered User:** User that is not registered. An Unregistered User can browse and search for listings but is not able to post, buy, or message other users.
- **Registered User:** User that is registered. A Registered User can browse, search, post, and purchase listings as well as message other users.
 - o ID
 - o Username
 - o Email
 - o Password
 - o isBanned
- **Admin:** Employee that oversees the website. An Admin approves and moderates the listings to make sure that they are not inappropriate.
 - o ID
 - o Email
 - o Password
- **Listings:** Items/services put up for sale by Registered Users for other Registered Users to purchase. Listings can be items such as books and electronics or they can be services offered such as tutoring.
 - o ID
 - o Title
 - o Category
 - o Price
 - o Pictures
 - o SellerID
 - o isSold

4. Initial List of Functional Requirements

For guests/Unregistered Users:

1. Shall be able to browse items and visit the website simply.
2. Shall be able to sort the items.
3. Shall be able to search for posts by keyword.
4. Shall be able to filter the items by cost.
5. Shall be able to register for an account by completing an online form.

For Registered Users (Shall perform all tasks of Unregistered User):

6. Shall be able to contact the Post creator through the platform.
7. Shall be able to login to their account.
8. Shall be able to logout of their session.
9. Shall be able to change their password.
10. Shall be able to use forgot password to recover login info.
11. Shall be able to buy listings.
12. Shall be able to post listings to sell.
13. Shall be able to message seller.
14. Shall be able to view messages from a potential buyer.
15. Shall be able to delete their own postings.

For Administer (Shall be able to perform all tasks of the seller):

16. Shall be able to delete any postings.
17. Shall be able to block accounts.
18. Shall be able to approve the pending posting of seller to let it show on the website.
19. Shall be able to disapprove posting of seller.

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	Buy and Sell	Amazon	Cregslis	Ebay
Browse and Filter Listings by Category	+	+	+	+
Quick Listing Options	++	-	+	+
Shopping Cart	-	++	-	+
Filter by SFSU Class Textbooks	++	-	-	-
Admin Approved Listings	++	-	-	-

Buy and Sell will allow the user to find any available textbooks by the specific SFSU class. The listing process will also be smooth and fast which can not be said for some of our competitors. While we do not have a shopping cart planned for the initial release of our website, easy browsing and sorting options makes finding what you are looking for a breeze. Buy and Sell will have only admin approved posts to ensure the quality of the posts meet the terms of service of the website. This ensures that our users are never subjected to obscene content.

7. High-level System Architecture and Technology Used

- **Server Host:** Application shall be hosted on a Pebble Host.
- **Operating System:** Application shall use Ubuntu 16.04 LTS.
- **Database:** Applications' data shall be stored using MySQL (Ver 8.0.17).
- **Web Server:** VPS.
- **Server-Side Language:** Server-side language shall be JavaScript.
- **Front-end Language:** Application's front-end shall be developed using HTML, CSS and Bootstrap
- **Web Framework:** Application's back-end shall be developed with Node.js (Ver 10.16.3)
- **Supported browsers:** Application shall be able to work on either mobile, laptop or website browsers including Google Chrome, Mozilla Firefox, and Safari.
- **Additional Technologies:**
 - o IDE's: Application shall be developed using IntelliJ, Visual Studio Code, WebStorm and Sublime Text.
 - o Express.js (Ver 4.16.1)
 - o jQuery

8. Team and Roles

6. **Daniel Janes** - Team Lead / Document Master
7. **Gem Lagman** - Github master / Back-End
8. **Monique Marti** - Front-End Lead
9. **Raya Farshad** - Back-End Lead
10. **Pramish Dhakal** - Front-End

9. Checklist

1. Team found a time slot to meet outside of class: **DONE**
2. Github master chosen: **DONE**
3. Team decided and agreed together on using the listed SW tools and deployment server:
DONE
4. 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**
5. Team lead ensured that all team members read the final M1 and agree/understand it before submission: **DONE**
6. Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.): **DONE**