SW Engineering CSC 648/848 Fall 2020
SFSU Trade Mart
Team 7

Team Members:

Team Lead:	Alicia Ramirez (Email: aramirez23@mail.sfsu.edu)
Front-End Lead:	Jonathan Pak
Back-End Lead:	Valeria Vallejo
Github Master:	Ricardo Carretero
Member:	Chandler Cruz
Member:	Chris Manaoat

Milestone 1
September 20, 2020

History Table

Date Submitted: September 22, 2020
Date Revised:

1. Executive Summary:

Students are always looking for the best book prices, but finding them becomes a very tedious task. Our website will relieve students of the anxiety that comes with trying to find all of the textbooks they need and ensures they will get the best deal whenever possible.

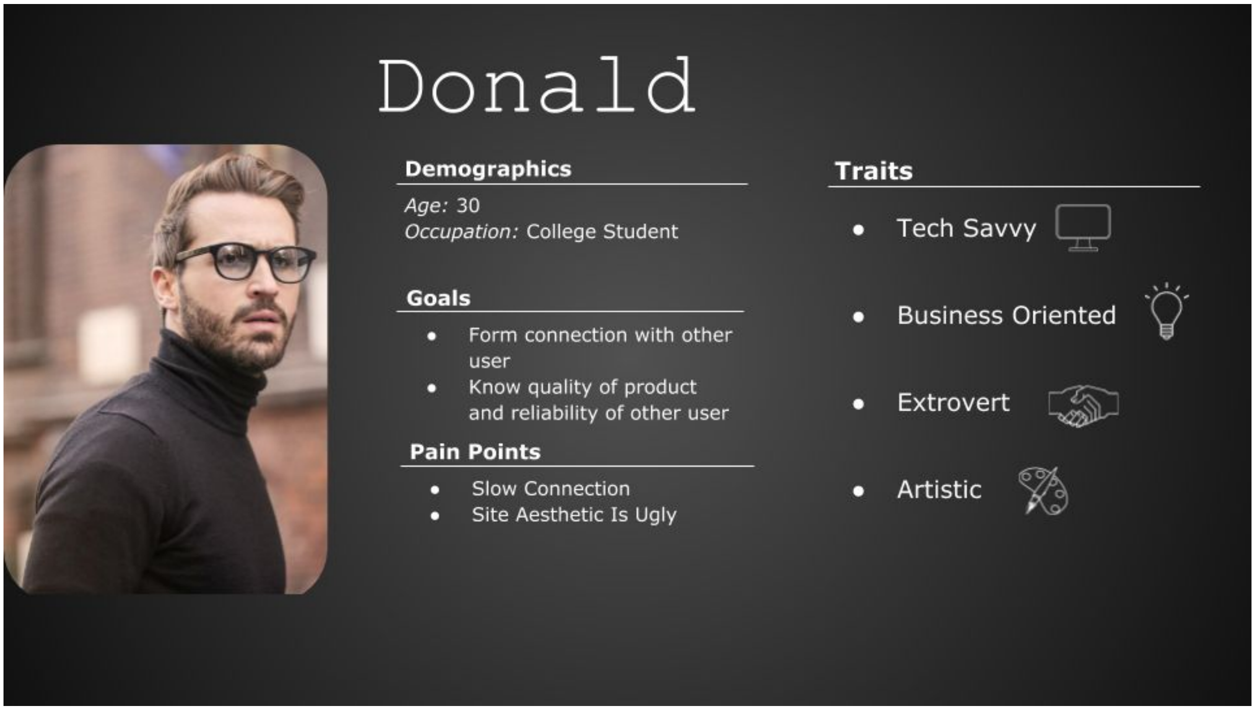
On our site, staff and students alike can sell and buy textbooks to other people at SFSU safely and efficiently. Being able to search for books by class, allows students to easily find the materials they'll need before classes even start. To ensure they're getting the best price, they'll be able to compare the price of their selection to the prices of similar items. If they see a listing that they want to track, they can put it on their watchlist and wait for the best price. We know that item quality is a major factor for a lot of students and our site requires that all sellers provide detailed descriptions of the product condition. Concerns can be addressed through our system and will keep the buyer and seller in contact.

For those who want an easy way to sell their old items, this site provides a safe and easy platform for sellers to advertise, sell, and distribute their goods. Sellers may choose from any bids and offers that come their way. Our sellers can be sure that they'll receive what they believe their product is worth.

Our unique system also gives buyers and sellers reviews in order to give people peace of mind that they are buying from or selling to a trustworthy person. For those who prefer not to meet with their buyer, or seller, in person, a secure drop-off location on campus may be agreed upon while they are in contact. And due to our unique OneCard integration, buyers now have a new place to use up their SFSU currency to save even more on their required materials for their semester.

2. Personae and Main Use Cases:

Donald (Student 1): Donald is a **student** and has a pretty large group of friends in his major. While he was able to order all his books and class materials ahead of time, some of his friends taking similar classes weren't able to before they sold out at the bookstore. Since he's always eager to help his friends out, Donald looks at the **items on sale** and **new arrivals** in hopes to help find their books. Donald utilizes the search function and due to the clean and easy to read **object listings** he was able to find what he was looking for. He takes note of the **contacts** listed and notifies his friends about this application.



Donald

Demographics

Age: 30
Occupation: College Student

Goals

- Form connection with other user
- Know quality of product and reliability of other user

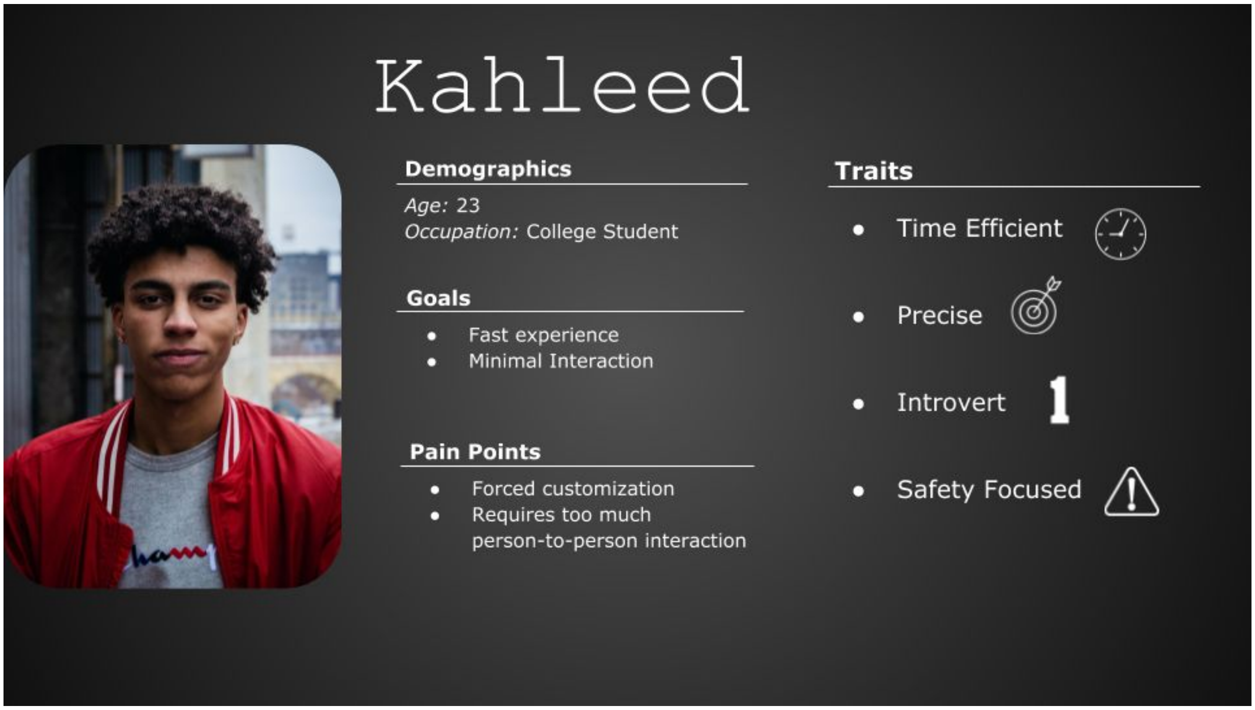
Pain Points

- Slow Connection
- Site Aesthetic Is Ugly

Traits

- Tech Savvy
- Business Oriented
- Extrovert
- Artistic

Kahleed (Student 2): Kahleed wants to get his book before the midterm. He thought he wouldn’t need it since he has google, but it was not enough. The bookstore would take a few days to stock, so he is looking for a used copy. After talking with classmates he learns about **SFSU Trade Mart**. On the app he can quickly find what he needs and make his order without much hassle on his part. Knowing about the multiple versions available made him hesitant, but he found his book after looking at **images** on the **items for sale** section. The app has default options that allow for quick checkout. He also wants as little person to person contact as possible so was relieved to find out there is an option for **locker drop off**.



Kahleed

Demographics

Age: 23
Occupation: College Student

Goals

- Fast experience
- Minimal Interaction

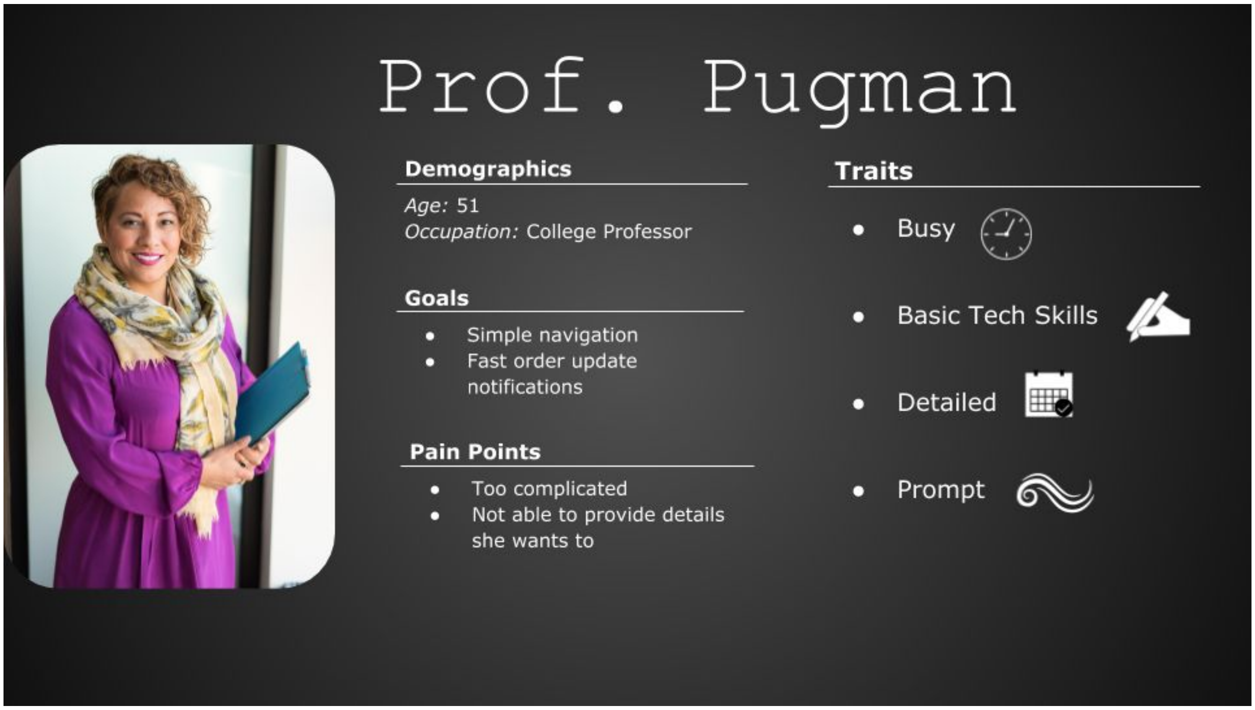
Pain Points

- Forced customization
- Requires too much person-to-person interaction

Traits

- Time Efficient
- Precise
- Introvert
- Safety Focused

Professor Pugman (Teacher): Professor Pugman just finished teaching her class. She has some additional resources for her class, so she listed them as optional materials in the **class resource form**, so her future students could see them when they **search by class**. On hand, she also had extra copies left over from previous years so she **posted some listings** on the site. The site was easy to navigate and she completed this process in a couple of minutes with the details she wanted to include. Afterwards the site provided **updates** that kept her up to date on the selling process. The simple **messaging feature** allowed her to easily manage and finalize all of her transactions.



Prof. Pugman

Demographics

Age: 51
Occupation: College Professor

Goals

- Simple navigation
- Fast order update notifications

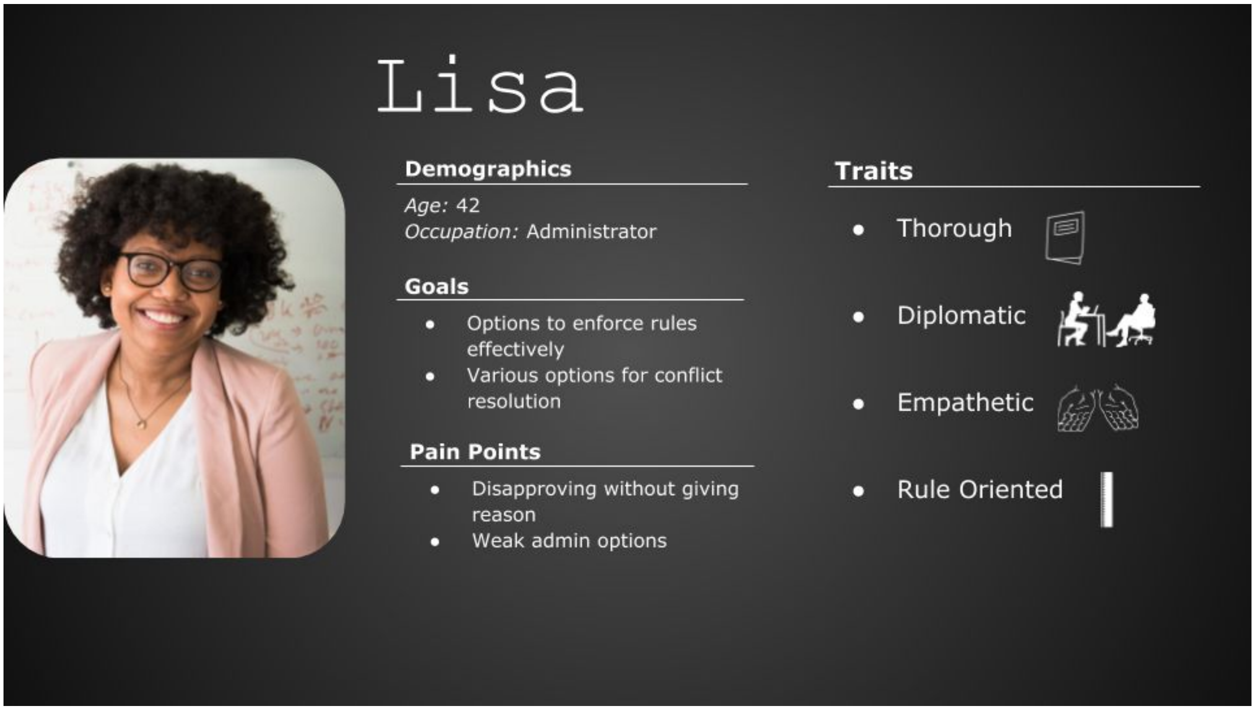
Pain Points

- Too complicated
- Not able to provide details she wants to

Traits

- Busy
- Basic Tech Skills
- Detailed
- Prompt

Lisa (Admin): Lisa has just finished sending out a batch of emails to her colleagues and has a free moment to **approve listings**. Since she has such scattered moments of reprieve throughout her workday, she can't devote long stretches of time to navigating through the website to approve all incoming student and teacher listings. However, since the UI was easy and quick to navigate she was able to **approve** a healthy amount of posts and **disapprove posts** with reasons while reading over the necessary details of each one during her small break.



Lisa

Demographics

Age: 42
Occupation: Administrator

Goals

- Options to enforce rules effectively
- Various options for conflict resolution

Pain Points

- Disapproving without giving reason
- Weak admin options

Traits

- Thorough
- Diplomatic
- Empathetic
- Rule Oriented

3. List of main data items and entities – data glossary/description:

- Image database: Houses the collections of images used to visually describe items.
- Object listing: The data structure of the item for sale.
- Vendor contacts: Data structure containing the vendors information, chat history, items for sale, etc...
- Client contacts: Data structure containing registered users transactions and chat history, and possibly items browsed.
- Guest contacts: Data structure containing all information related to a guest purchase.
- List of previous orders: Sub class that is stored in one of the contact data structures containing all orders by the subject
- Items on sale: Items that are currently discounted (might not be implemented)
- New arrivals (items): Recently listed items that have come up for sale or items from the starting semester (might not be implemented)
- Message history: Data structure stored within the contacts struct that stores all the conversations between seller and client
- Notification history: Data structure containing all the notifications the user has received.
- Class resource form: Using this form, staff can easily input all required and optional resources that students will need for their class.

4. Initial List of Functional Requirements:

Regular features:

1. The user shall have the ability to search for specific listings based on words in its title or description.
2. Users shall be able to compare the price of the listing they're viewing to prices of other similar listings.
3. Users shall be able to compare the listings of textbooks they're searching for based on its version.
4. The website shall load the appropriate pages and content to the user based on the user's navigation through the site.
5. The website's database shall store and provide the website with the items users post for sale.
6. Registered vendors shall be able to create listings for their items and put them on sale through the website.
7. Users shall be able to filter their search results based on the prices of the listings.
8. Users shall be able to filter their search results based on the book versions of any textbook listings.
9. Users shall be able to filter their search results based on which materials are required for specific courses.
10. Users shall be able to filter their search results based on which semesters the classes they're required for have been scheduled for.
11. Users shall be able to filter their search results based on which class materials are required for classes taught by specific professors.

12. Users shall be able to filter their search results based on which class materials are required for classes taken for specific majors.
13. Users shall be able to register accounts with the website.
14. Users looking to purchase items on the website shall be able to contact vendors for listings they're interested in for further clarification.
15. Vendors who are approached by users interested in their listings shall be able to respond to users and address their questions and concerns.
16. Descriptions of the items for sale shall be visible to the user on the search results and browsing pages of the website.
17. Admins shall be allowed to approve or disapprove listings as they see fit.
18. Users shall receive notifications when their listing is accepted/rejected, someone purchases their listing, they receive a message, or they receive a rating.

Special features:

19. Users purchasing products on the website shall be able to choose the meeting or drop off location on the website.
20. Users interested in specific listings shall be able to make bids and offers to the vendors of those listings.
21. Vendors who receive bids and offers from users about their listings shall be able either to accept their offers or to respond to those users with a counteroffer.
22. Users shall be able to purchase listings on the website using funds from their Gator Dollars or One Card.
23. Users shall be able to view the bookstore's inventory via listings on the website.
24. Users shall be able to rate and review vendors they've had transactions with.
25. Vendors shall be able to rate and review users they've had transactions with.
26. Users shall be able to watch specific searches to keep up-to-date with any new listings for those specific searches.

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. 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 and all privacy policies will be appropriately communicated to the users.
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 e-mail 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, Fall 2020. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive analysis:

	Craigslist	Amazon	Ebay	Trade mart
Book name	y	y	y	y
Book by class	n	n	n	y
Locker drop off	n	y/partial	n	y
Message seller	y	y	y	y
Barter	y	n	n	y
Version compare	n	n	n	y
Search by class	n	n	n	y
One card purchase	n	n	n	y
Rate/review seller	y	y	y	y
Item price watch	n	y/partial	n	y

Our product in comparison to the other brands listed, has a clear advantage with the ability to use the OneCard as a payment option. Other sites also do not have the ability to produce listings of goods for sale that are relevant to that semester only. Furthermore another feature that is being explored is the ability to suggest a previous semester's version of the book, but that may not be implemented due to it depending on the amount of changes made to the version. The ability to search by class has to be one of the most useful features compared to our competitors, this allows any incoming freshmen to purchase their books for the upcoming semester easily.

7. High-level system architecture and technologies used:

- Server Host: AWS, 1 3.3 Ghz Intel Scalable Processor, 1 GB Ram
- Operating System: Linux 20.04
- Database: mySQL 8.0.21
- Web Server: Nginx 1.14.0
- Server-Side Language: Python 3.8.2
- Additional Technologies:
    - Web Framework: Bootstrap, Flask
    - IDE: VS Code, Vim

8. Team and roles:

Team Lead/Document Master:	Alicia Ramirez
Front-End Lead:	Jonathan Pak
Back-End Lead:	Valeria Vallejo
Github Master:	Ricardo Carretero
Member:	Chandler Cruz
Member:	Chris Manaoat

9. Checklist:

- So far all team members are engaged and attending ZOOM sessions when required - **DONE/OK**
- 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 decided and agreed together on using the listed SW tools and deployment server - **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 organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) - **DONE/OK**