

# SW Engineering CSC648/848 Summer 2022

## College Essentials

### Team 4

| Student Name   | Student Email          | Github Username | Student ID | Student Roles              |
|----------------|------------------------|-----------------|------------|----------------------------|
| Ryan Murphy    | rmurphy4@mail.sfsu.edu | deadlepricon    | 921366560  | Team Lead,<br>Backend Lead |
| Duncan Forniss | Dforniss@mail.sfsu.edu | DunkeyFunkey    | 917678507  | Frontend Lead              |
| Wee Yang Leong | wleong@mail.sfsu.edu   | kleong2000      | 918614936  | Backend Assistant          |
| Darren Wong    | dwong19@mail.sfsu.edu  | dawong025       | 918565679  | Database Master            |
| Justin Wong    | jwong66@mail.sfsu.edu  | jwong66         | 918516578  | Github Master              |

**Milestone:** Milestone 4

**Date:** 07/28/22

## Version History

| Milestone | Date       |
|-----------|------------|
| M4V1      | 07/28/2022 |
| M3V2      | 07/28/2022 |
| M3V1      | 07/19/2022 |
| M2V2      | 07/19/2022 |
| M2V1      | 07/06/2022 |
| M1V2      | 07/01/2022 |
| M1V1      | 06/21/2022 |

## **Table of Contents**

|  |           |
|--|-----------|
| <b>[Section I] Product Summary</b>   | <b>4</b>  |
| <b>[Section II] Usability Test Plan</b>                                    | <b>9</b>  |
| <b>[Section III] QA Test Plan</b>  | <b>29</b> |
| <b>[Section IV] Code Review</b>  | <b>33</b> |
| <b>[Section V] Self-check: Best Practices for Security</b>                 | <b>38</b> |
| <b>[Section VI] Self-check: Adherence to Original Non-Functional Specs</b> | <b>41</b> |
| <b>[Section VII] List of Contributions</b>                                 | <b>54</b> |

## **[Section I] Product Summary**

### **Product Name:** College Essentials

College Essentials is an online marketplace that is catered towards students, professors or anyone who has products or services to sell related to college.

College students currently have a difficult time trying to buy textbooks at fair and reasonable prices. Students have to either buy their books in new condition at full price, or find it used for close to original price. It is also difficult for students to be able to sell their old textbooks that they no longer need. College Essentials aims to fix these problems by implementing a marketplace catered to users who want to buy new and used textbooks at reasonable prices, and be able to sell their old textbooks for prices that they choose. The three main sections that College Essentials aims to focus on is an online marketplace, forum posts, and service listings.

### **What is unique in our product:**

To set College Essentials apart from its competition, we aim to implement a user based online marketplace that allows registered users to be able to buy and sell new and used items from people similar to them at reasonable prices. The marketplace will have grown to include not only textbooks, but other college essential items as well. College Essentials will aim to stand out from their competition by allowing users to sell their used products and let them set their own prices. We plan to have a ratings system to have users know that they are buying from a trusted source. Our goal is to make College Essentials an all in one place for College students and professors alike. Forum postings and service listings will help connect these user groups by giving users an area for communication, and allowing them to buy and sell services related to college that they might have. Finally, we

want to be accessible from any university, by allowing users to search the marketplace based off of location.

### **Itemized List of All Major Committed Functions :**

1. General User
  - 1.1. A general user shall be able to browse forum posts.
  - 1.2. A general user shall be able to browse items on the marketplace.
  - 1.3. A general user shall be able to browse service listings.
  - 1.4. A general user shall be able to search for their needed service.
  - 1.5. A general user shall accept our terms of conditions.
  - 1.6. A general user shall be able to search by item condition in the marketplace.
2. Registered User
  - 2.1. A registered user shall be able to log in with their username.
  - 2.2. A registered user shall be able to review another user.
  - 2.3. A registered user shall be able to create their own profile with text.
  - 2.4. A registered user shall be able to create their own profile with pictures.
  - 2.5. A registered user shall be able to edit their own profile with text.
  - 2.6. A registered user shall be able to edit their own profile with pictures.
  - 2.7. A registered user shall be able to post a picture as their profile picture.
  - 2.8. A registered user shall be able to post a service listing.
  - 2.9. A registered user shall be able to list items for sale in the marketplace.
  - 2.10. A registered user shall be able to post on the forums.
  - 2.11. A registered user shall be able to log out.
  - 2.12. A registered user shall be able to rate other general users.

### 3. Service Listing

- 3.1. A service listing shall have a title.
- 3.2. A service listing shall have the location of where the service will be needed.
- 3.3. A service listing post shall have characters in its content body.
- 3.4. A service listing post shall be able to have its content body edited.
- 3.5. A service listing post shall be able to be unlisted.

### 4. Forum Listing

- 4.1. A Forum listing shall have a title.
- 4.2. A Forum listing post shall have characters in its content body.
- 4.3. A Forum listing post shall have basic information like time posted.
- 4.4. A Forum listing shall be visible to all general users.
- 4.5. A Forum listing post shall be able to be unlisted.

### 5. Shopping Cart

- 5.1. A shopping cart shall remember what has been stored within it.
- 5.2. A shopping cart shall remember how many of one entity have been stored within it.
- 5.3. A shopping cart shall allow users to edit products within it.
- 5.4. A shopping cart shall allow for check out.
- 5.5. A shopping cart shall give descriptions of products stored within it

### 6. Checkout

- 6.1. Checkout shall allow a user to input their payment information.

- 6.2. Checkout shall allow a user to input their shipping address.
  - 6.3. Checkout shall allow a user to input their billing address.
  - 6.4. Checkout shall allow users to review their purchase on last time.
7. Navigation Bar
- 7.1. A navigation bar shall have a section for students selling their products.
  - 7.2. A navigation bar shall have a section for teachers to look for student assistants.
  - 7.3. A navigation bar shall have a section for the student forums.
  - 7.4. A navigation bar shall have a section for the about us page.
8. Marketplace Listing (Buy/Sell Listing)
- 8.1. A marketplace listing shall have a title.
  - 8.2. A marketplace listing shall have a description.
  - 8.3. A marketplace listing shall have a photo (or many).
  - 8.4. A marketplace listing shall have a price.
  - 8.5. A marketplace listing shall be able to be unlisted.
  - 8.6. A marketplace listing shall have filters to categorize.
9. Ratings
- 9.1. User ratings shall rate a profile on a scale of 1 to 5.
  - 9.2. User ratings shall be visible on a registered user's profile.
  - 9.3. User ratings shall be visible on a registered user's marketplace listing.
  - 9.4. User ratings shall be visible on a registered user's service listing.
  - 9.5. User ratings shall be able to be posted by registered users.

**URL to our product:** <http://my-app-dev.us-west-1.elasticbeanstalk.com/>



## **[Section II] Usability Test Plan**

### **Search**

- Test Objectives
  - The main objective for this test is to make sure that our search function works in relation to our database. The search bar should be able to search through the database based off of the title of the object searched. It should also be able to search for items with specific categories, and return only items in those categories. These functions are important to be tested because the search bar is the main way to be able to search for items on College Essentials. Without the search bar working, users will not be able to aggregate the page with the specific items that they want, but will instead be flooded with every single item in the database.
- Test Description
  - The system that will be used for the test will be a Windows 10 operating system. Multiple web browsers will be used, from Google Chrome, Firefox, and Microsoft Edge. The tests should be able to run on all the browsers with the same end results.
  - The starting point for the test will be a user who has not logged in. The search bar should work the same regardless if the user is logged in or not, so having a user not be logged in yet should emulate the ability of having majority functionality on the website without needing to be registered.
  - The intended users for this test are anyone who visits this website. Because our starting point was having the user not be logged in, this will be the same as having the average person being able to access

College Essentials and search for what they need first, before deciding if they want to create an account with our services.

- The URL that is going to be tested is:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/home>.

We are aiming to measure the aggregation of the website based on the users inputs or lack of inputs. This means that regardless of what the user searches for, the website should be able to interact with the database in order to populate the page with what the user aims to find.

- Usability Test Table

| <b>Number of pages needed</b> | <b>Test/Use case</b> | <b>% completed</b> | <b>Errors</b> | <b>Comments</b> | <b>% of time to complete task (efficiency)</b> |
|-------------------------------|----------------------|--------------------|---------------|-----------------|--|
| 1                             | Search               | 100%               | N/A           | N/A             | 100%   |
| 1                             | View Items           | 100%               | N/A           | N/A             | 100%   |
| 2                             | Select Item          | 100%               | N/A           | N/A             | 100%   |
| 2                             | Add to Cart          | 100%               | N/A           | N/A             | 100%   |
| 2                             | View Seller          | 100%               | N/A           | N/A             | 100%   |

- User Satisfaction

I was able to search for items with ease. \*

|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

i think there should be a marketplace tab on the left section too.

The information for each search item was easily visible. \*

|                   | 1                     | 2                     | 3                     | 4                                | 5                     |                |
|-------------------|-----------------------|-----------------------|-----------------------|----------------------------------|-----------------------|----------------|
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | Strongly Agree |

Comments:

one pic didnt load

The items displayed after searching is in a good format. \*

|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

pictures could be bigger

## Item Post

- Test Objectives
  - Posting has the objective of giving a user who is unfamiliar with technology the ability to post on College Essentials without the prior knowledge of how to. Even without prior experience with other sites, posts should have easy and straightforward labels on all the input fields so that they are clear to the user in what they do. There shouldn't be a question as to how our posting should work, and should allow for users to be able to post whatever items they want on our site (within guidelines) as long as they are logged in. The importance of testing these functions is that College Essential relies on user interaction with buying and selling items, so users must be able to sell items on the website.
- Test Description
  - The system that will be used for the test will be a Windows 10 operating system. Multiple web browsers will be used, from Google Chrome, Firefox, and Microsoft Edge. The tests should be able to run on all the browsers with the same end results.
  - The starting point for the test will be a user who has logged in. Only users who have logged in should be able to make posts. However, any user can find posts. This helps the buying and selling system work on College Essentials, where only registered users are allowed to post, but anyone is allowed to buy.
  - The intended users for this test are any registered users on College Essentials. Because our starting point is having the user be logged in,

only registered users should be able to sell. Registered users are thus our intended test group for this section.

- The URL that is going to be tested is:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/postItem>.

We are aiming to measure the ability of the user to add an item to the database to be able to sell, as well as add any information related to said item. The user must be able to find their item, and edit the information if needed.

- Usability Test Table

| <b>Number of pages needed</b> | <b>Test/Use case</b>  | <b>% completed</b> | <b>Errors</b>                                     | <b>Comments</b>   | <b>% of time to complete task (efficiency)</b> |
|-------------------------------|-----------------------|--------------------|---|---|--|
| 1                             | Access Sell Item Form | 100%               | N/A   | N/A   | 100%   |
| 1                             | Sell Item             | 100%               | N/A   | N/A   | 100%   |
| 2                             | Search for Item       | 90%                | Item appears in item view as new instead of used. | This can be implemented by changing the item information. | 80%  |

|   |                     |     |                                      |  |     |
|---|---------------------|-----|--------------------------------------|--|-----|
|   |                     |     |                                      |  |     |
| 3 | See Item on Account | 20% | Selling history not implemented yet. | Selling history needs to pull from user id and item id.                    | 20% |
| 3 | Edit Item           | 0%  | Edit item is not yet implemented.    | We can implement an edit function based off of selling history on account. | 0%  |

- User Satisfaction

Item Post Survey

I was able to post a product to sell with ease. \*

12345

Strongly DisagreeStrongly Agree

Comments:

straightforward

I was able to easily find what items I am selling in the marketplace. \*

12345

Strongly DisagreeStrongly Agree

Comments:

if it was there i found it

I am easily able to track what items I am selling. \*

|                   |                       |                       |                                  |                       |                       |                |
|-------------------|-----------------------|-----------------------|----------------------------------|-----------------------|-----------------------|----------------|
|                   | 1                     | 2                     | 3                                | 4                     | 5                     |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly Agree |

Comments:

it wasn't working



## Forum Posts

- Test Objectives
  - The main objective of this test is to verify that all the functions regarding forum postings are functional. Forums are another feature that College Essentials should have that sets them apart from the competition. Forum posts allow for multiple users to interact with one another just through the use of forum posts. Forum posts consist of a topic that the main user is posting, and a section for comments from other users to reply with. The importance of testing forum posts is to allow another way for users to interact with one another for different topics, without the need to purchase items.
- Test Description
  - The system that will be used for the test will be a Windows 10 operating system. Multiple web browsers will be used, from Google Chrome, Firefox, and Microsoft Edge. The tests should be able to run on all the browsers with the same end results.
  - The starting point for the test will be for users who have logged in. Similar to item posting, only registered users should be able to post forum posts, however, all users should be able to view forum posts regardless of logged in or not. Forum posts have a section for the user who posted them, and thus they need to be logged in.
  - The intended users for this test are anyone registered with the website. In this test, the user must have already been registered and logged in with College Essentials, however the test shall be accessible for users who are not adept with technology. This means that being able to post a forum post shall be easy for any user to post it.

- The URL that is going to be tested is:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/forumList>.

We are aiming to measure the ability of the forum list to be populated with forum posts, as well as allowing any user to be able to search for a specific forum post. Users should also have an easy time making posts of their own.

- Usability Test Table

| <b>Number of pages needed</b> | <b>Test/Use case</b> | <b>% completed</b> | <b>Errors</b>      | <b>Comments</b>  | <b>% of time to complete task (efficiency)</b> |
|-------------------------------|----------------------|--------------------|--------------------|--|--|
| 1                             | View Forum Lists     | 100%               | N/A                | N/A  | 100%   |
| 1                             | Make a New Post      | 80%                | Username is wrong. | Username is hard coded and does not change based on login. | 70%  |
| 2                             | View Forum Post      | 80%                | Username is wrong. | Username is hard coded and does not change based on login. | 70%  |

|   |                       |     |                                      |   |     |
|---|-----------------------|-----|--------------------------------------|---|-----|
|   |                       |     |                                      |   |     |
| 2 | Comment on Forum Post | 60% | Comment section not yet implemented. | There is no button to submit a comment.                           | 50% |
| 1 | Search Forum Posts    | 0%  | Search bar not implemented.          | Search bar does not work, search button does not return anything. | 20% |

- User Satisfaction

I was able to post a forum post with ease. \*

|                   |                       |                       |                       |                       |                                  |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

.....

I was able to easily find forum posts. \*

|                   |                                  |                       |                       |                       |                       |                |
|-------------------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------|
|                   | 1                                | 2                     | 3                     | 4                     | 5                     |                |
| Strongly Disagree | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Strongly Agree |

Comments:

search option didn't work

.....

I am easily able to navigate through forum posts. \*

|                   |                       |                       |                       |                       |                                  |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

it was organized by title but user was wrong

.....

## ShoppingCart

- Test Objectives
  - The main objective of this test is to verify that all the functions in regards to the shopping cart are working properly. The main functions that we want to test are adding items to the shopping cart, being able to access and manipulate the shopping cart itself, and being able to exit the shopping cart for checkout. We need to make sure that the shopping cart is working as intended since commerce is the central part of College Essentials. Without a working shopping cart, the main functionality of College Essentials is lost.
- Test Description
  - The system that will be used for the test will be a Windows 10 operating system. Multiple web browsers will be used, from Google Chrome, Firefox, and Microsoft Edge. The tests should be able to run on all the browsers with the same end products.
  - The starting point for the test will be a user who has not logged in. In our website design, an unregistered user shall be able to access the main functions of the website, including the ability to make purchases without being logged in. This means that a user who is not logged in should be able to use the shopping cart and be able to checkout.
  - The intended users for this test are anyone who visits this website. Because our starting point was having the user not be logged in, this will be the same as having the average person being able to access College Essentials and be able to purchase products without having the need to sign up.

- The URL that is going to be tested is:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/ShoppingCart>.

We are aiming to measure the ability of the shopping cart by allowing the user to add items to the shopping cart, being able to show the details of the items in the shopping cart, and being able to interact with said items as well as checkout.

- Usability Test Table

| <b>Number of pages needed</b> | <b>Test/Use case</b> | <b>% completed</b> | <b>Errors</b>                   | <b>Comments</b>                             | <b>% of time to complete task (efficiency)</b> |
|-------------------------------|----------------------|--------------------|---------------------------------|---|--|
| 2                             | Add to Cart          | 100%               | N/A                             | N/A   | 80%  |
| 1                             | Access Shopping Cart | 100%               | N/A                             | N/A   | 100%   |
| 1                             | Edit Cart            | 100%               | N/A                             | N/A   | 90%  |
| 2                             | Checkout Cart        | 90%                | Checkout not fully implemented. | Checkout does not create a new transaction. | 90%  |
| 3                             | Check Purchase       | 30%                | Purchase                        | Purchase history                            | 20%  |

|  |         |  |                        |  |  |
|--|---------|--|------------------------|--|--|
|  | History |  | history not appearing. | page is on account, but the purchase history is not implemented. |  |
|--|---------|--|------------------------|--|--|

- User Satisfaction

I was able to access my shopping cart with ease. \*

1

2

3

4

5

Strongly Disagree

☐

☐

☐

☐

☒

Strongly Agree

Comments:

.....

I was able to interact with my shopping cart with ease. \*

1

2

3

4

5

Strongly Disagree

☐

☐

☐

☐

☒

Strongly Agree

Comments:

i can add and remove items easily .....

I am easily able to checkout my shopping cart. \*

|                   |                       |                       |                       |                       |                                  |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

completed my order



## Account

- Test Objectives

- The main objective of this test is to verify that different functions of the user account are working. These functions include being able to view account information, and edit said account information.

Accounts shall have an about me that is editable. There should also be a section for recent orders, user listings, and user ratings. The importance for this is because everything that a user does while logged in should be linked to their user account. Users who are also making purchases should be able to view a user's account before making said purchase in order to make sure that they are not getting scammed.

- Test Description

- The system that will be used for the test will be a Windows 10 operating system. Multiple web browsers will be used, from Google Chrome, Firefox, and Microsoft Edge. The tests should be able to run on all the browsers with the same end products.
- The starting point for the test will be a user who is logged in. A user must be logged in in order to access the account page of the website. Without being logged in, a user wouldn't have an account, and therefore not have any information that needs to be tracked.
- The intended users for this test are anyone registered users on the website. Because our starting point is a user who is logged in, any registered user on the website should be able to do the actions needed for the test. The test should be easy enough for any user with any experience in technology to be able to perform.

- The URL that is going to be tested is:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/Account>.

We are aiming to measure the usability of the accounts page, which would include populating the page with user information, and allowing them to edit their own user information. Users should also be able to look at any history regarding their account and any other information.

- Usability Test Table

| <b>Number of pages needed</b> | <b>Test/Use case</b> | <b>% completed</b> | <b>Errors</b>                             | <b>Comments</b>  | <b>% of time to complete task (efficiency)</b> |
|-------------------------------|----------------------|--------------------|---|--|--|
| 1                             | Access Account       | 100%               | N/A                                       | N/A  | 100%   |
| 2                             | Edit Account         | 50%                | Edit account does not change information. | Text fields do not change user information on the main page. | 50%  |
| 2                             | View Recent Orders   | 30%                | Recent orders not yet implemented.        | Recent orders do not yet appear.                             | 10%  |
| 2                             | View User Items      | 30%                | User items not                            | User items being   | 10%  |

|   |              |     |                              |   |     |
|---|--------------|-----|------------------------------|---|-----|
|   |              |     | yet implemented.             | sold does not yet appear.                           |     |
| 1 | Post Ratings | 40% | Ratings not yet implemented. | Drop down has a number section for posting ratings. | 10% |

- User Satisfaction

I was able to access my account information with ease. \*

1
2
3
4
5

Strongly Disagree
☐
☐
☐
☐
☒
Strongly Agree

Comments:

I was able to edit my account information with ease. \*

1
2
3
4
5

Strongly Disagree
☐
☐
☒
☐
☐
Strongly Agree

Comments:
  
about me and contact me section didn't work

I am able to access my account history with ease. \*

|                   |                       |                       |                       |                       |                                  |                |
|-------------------|-----------------------|-----------------------|-----------------------|-----------------------|----------------------------------|----------------|
|                   | 1                     | 2                     | 3                     | 4                     | 5                                |                |
| Strongly Disagree | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input checked="" type="radio"/> | Strongly Agree |

Comments:

.....

## **[Section III] QA Test Plan**

### **Test objectives:**

- The purpose for these QA tests are to make sure our superior functions are working properly for College Essentials. Since college essentials is designed to replace ebay with its superior usability, UI and features. We want to make sure our Search and Upload functions are working as intended. Without these features being properly implemented, the entire website will shut down. To ensure this does not happen we are designing tests to test our multiple search features including marketplace listings, form listings and service listings. These are what allows the user to communicate and to profit within the website. The next features are the upload functions for marketplace, forms and service listings. These allow registered users to post their products or services so other users can purchase or communicate with the poster.

### **HW and SW setup (including URL):**

- Hardware requirements need to be any laptop. As for software there needs to be an OS of any type such that the device is setup properly ie. Windows or Mac. They will also need to have any browser in order to access our website, such as Google, Opera GX, Microsoft Edge, etc. As well as a proper connection such that they can interact with our website, and the other users involved. Once a user all the prerequisites in order to access our website, they are able to visit it here at:

<http://my-app-dev.us-west-1.elasticbeanstalk.com/>

### **Feature to be tested:**

- Search
  - High level searching such as individual item or category search

- Item Posting
  - Testing input fields such as URL, Title, Price, Description, Quantity, and Quality. Also if logged in
- Form Posting
  - Test Input fields such as title and description. Also tests if user is logged in
- Shopping Cart
  - This testing field is for adding times, removing or adding more quantity, and completing a checkout
- Account
  - Check if editing account is working, ability to use all features within, and see others accounts

### QA Test plan:

| Number | Test Title | Description   | Test Input  | Expected Output                                     | Pass/Fail                      |
|--------|------------|---|---|---|--------------------------------|
| 1      | Search     | Make sure the search function for their respective page | marketplace/<br>home page<br>(Book)<br>-----<br>Forum page/<br>Service<br>Listing<br>(Any Post) | A post for a<br>book<br>-----<br>Respective<br>Post | Pass<br>-----<br>-----<br>Fail |
| 2      | Item       | Make sure the item is                                   | Test by   | (Respective   | Pass                           |

|   |               |  |  |  |      |
|---|---------------|--|--|--|------|
|   | Posting       | posted properly on the marketplace, and properly stored in database with its proper values (URL, Title, Price, Description, Quantity, and Quality) | adding image url, tittle, description, quantity, quality and price | item)<br>is posted on the marketplace, and appears on our database with proper URL, Title, Price, Description, Quantity, and Quality |      |
| 3 | Form Post     | Makes sure that the new form post is properly added to the form listing and is stored in the database()  | Enter title and description  | There should be a pop up saying that the form was properly posted along with being able to see it on formlistings                    | Pass |
| 4 | Shopping Cart | Adding items to the cart with the ability to   | Adding selected item   | (Respective item) is   | Pass |

|   |         |  |                                     |   |      |
|---|---------|--|-------------------------------------|---|------|
|   |         | add or reduce quantity then complete the checkout or remove an item. | in full item view                   | present in cart with ability to add or reduce quantity, and the ability to remove item or remove checkout |      |
| 5 | Account | See if account pull up is accurate for same user that clicked        | Click on account button after login | Account page should open up with your personal info   | pass |



## **[Section IV] Code Review**

### **I. eXtreme Programming**

A. Our team's coding style has solidified itself to be an eXtreme Programming style. We demonstrated many if not all components of XP such that we had come to the conclusion that this coding style is what fit our situation the best when compared to the other styles. These components include communication, simplicity, feedback, courage, and respect. First being communication which our team has an abundance of. We had plenty of meetings, which albeit had no webcam still had plenty of communications. The communication did not stop there as we had plenty outside of voice chats as well, through things like chatting by text. Overall, it helped keep us on track as to what tasks were at hand. Second feature we practiced was simplicity. We all thought it best that we stuck with what was best for our skills. It worked off our specialties such that we could finish our work in an efficient manner. We also tend to not work towards future prospects, rather we worked with what we were comfortable with to make a simple working project which later we can build upon. Third feature we practiced was feedback, which we were definitely not lacking in. If we found ourselves dissatisfied with a certain feature we would not be afraid to voice our opinions on the subject, and provide feedback. It helped refine our product to what we thought it should be. Which leads me to the fourth we practiced, courage. None of us were shy of courageousness, and were not afraid to voice any of our opinions. Through those opinions we were not afraid of change either, and were willing to change any of our code in a heartbeat if it meant a better project. Last but not least is the respect aspect. No one on the team

lacked a sense of respect for the other person. This builds upon courage and feedback because we never put each other down. We always respected each other's time, skill, and availability which enabled courage, and feedback. Overall, all of these components helped build up to how we got our coding style, which in the end we concluded fit the most into extreme programming.

## II. (Code portion review which needs the QA test plan done)

We are doing a code review for our post item checking algorithm. For this we are using flags and null checking to make sure all the fields are filled out so users can't post blank items. Ryan is asking for this to be reviewed because he wanted the others to see if they could understand the code since he felt like it made no sense to the outside viewer.

Ryan Michael Murphy



To: Darren Lok-Lu... +1 other

Wed 7/27/2022 6:27 PM

Here is our checking method for posting an item to the website, How does it look?

```

129  if(LoginView.logStatus()){
130      float p = Float.parseFloat(price.getValue());
131      int q = Integer.parseInt(quant.getValue());
132      if(price.getValue().contains("$") || quant.getValue().contains("$")){
133      }{
134          showMsg(userBean);
135          flag++;
136      }
137
138
139      if(p == 0 || q == 0){
140          showZero();
141          flag++;
142      }
143
144      }else{
145          flag++;
146          showFail(userBean);
147      }
148      if(flag == 0 && url.getValue() != "" && title.getValue() != "" && description.getValue() != ""
149      && category.getValue() != null && condition.getValue() != null && price.getValue() != ""){
150          db.StorePostItem(title.getValue(), url.getValue(), condition.getValue(), category.getValue(),
151          price.getValue(), description.getValue(), quant.getValue());
152          showSuccess(userBean);
153          userBean = new PostItemDetail();
154          this.getUI().ifPresent(ui ->
155          ui.navigate(location: "/home"));
156      }
157
158  });
159

```

Brendan sent a response of :

Wee Brendan Leong



To: Ryan Michael Murphy

Thu 7/28/2022 10:50 AM

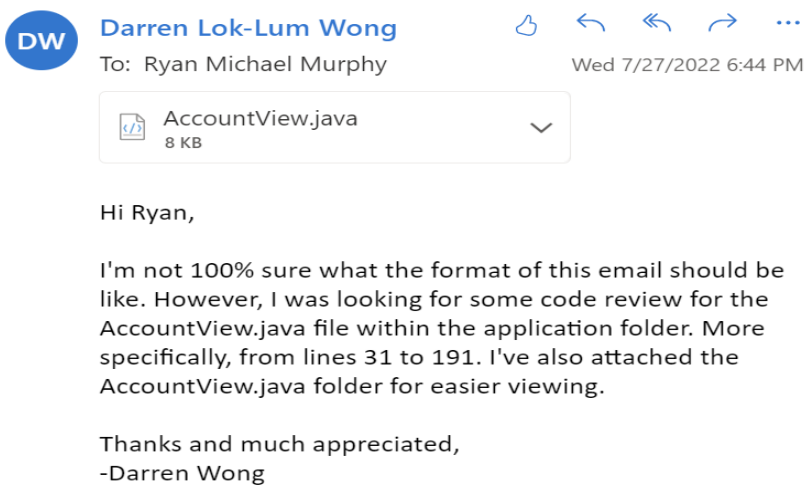
Good morning Ryan. I think whatever you have here is really good. I can see that this function will be used to populate a site with a new item. I can also see that a user will not be able to populate the price field with a negative pricing. However, there seems to be a lack of comments inside this function. I think without proper comments it will be hard for people without proper context to grasp what this function does in a timely manner. Otherwise, I think this function looks amazing!

...

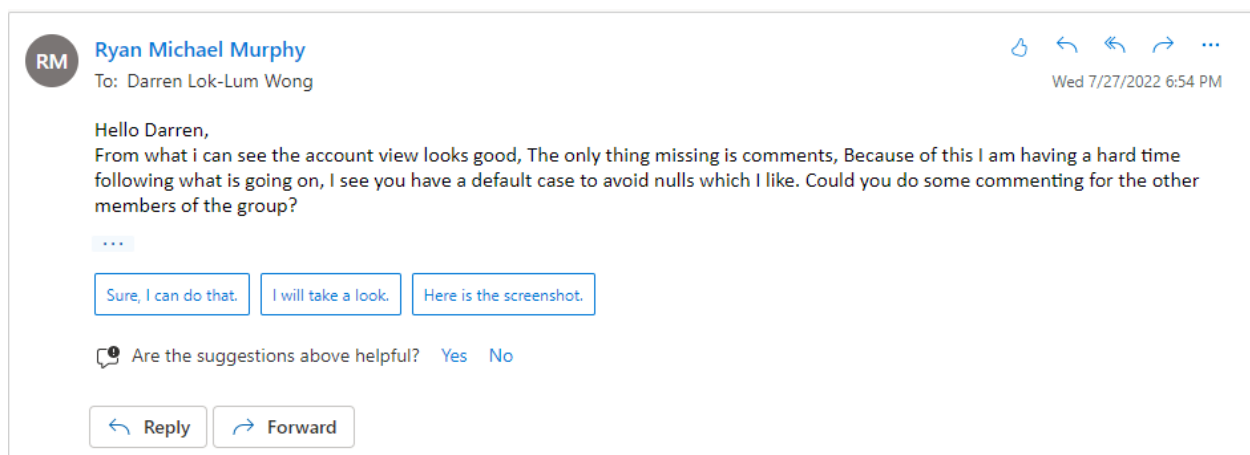
Because of this response Ryan edited the variable names and added comments to our postItemView.java class. He made the code more readable and spaced things

out. The code is now working fully as expected and the other members of the team can now understand his thought process.

The next code review we as a group checked was Darren's AccountView.java. Darren chose this segment of code because he worked on it a while ago and wanted us to understand what his thought process was and to see if we could see any logic errors.



Ryan responded with this:



Darren went back to accountView and added comments and made his code legible and spaced out. There were also no major logic errors any of the team mates could see at the moment.

## **[Section V] Self-check: Best Practices for Security**

- I. Account Information
  - A. A general user can view other user profiles
  - B. Only a registered user can edit their own user profile
- II. Forum Posts / Replies
  - A. A general user can view forum posts and replies
  - B. Only a registered user can post forum posts and replies
- III. Service Listings / Replies
  - A. A general user can view service listings and replies
  - B. Only a registered user can post service listings and replies
- IV. Ratings
  - A. A registered user can rate another registered user if they have a transaction with that user.
- V. Payment
  - A. Payment methods will be hashed upon entry, and stored inside the database as hashed values instead of plain text (depending on the payment API)
- VI. Password
  - A. Passwords are hashed upon registration, and stored inside the database as hashed passwords instead of plain text. The following screenshots illustrate how a password from the registration form is stored as a hashed password within the database.

**Signup for College Essentials**

First name • Kevin

Last name • K

Username • tig777

University • SFSU

Email • kevink@sfsu.edu

Password • ..... Confirm password • team4\_2022

☒ Agree To Terms & Services? [Terms & Services](#)

**Create Account**

B.

C.

|    |       |                                  |                 |   |        |
|----|-------|----------------------------------|-----------------|---|--------|
| 23 | Kevin | 44873c4ef315d2a4884cd3a29eabfcb5 | kevink@sfsu.edu | K | tig777 |
|----|-------|----------------------------------|-----------------|---|--------|

## VII. Data Input Validation

- Registration form ensures that all the fields must be filled out, including having an 8+ character password
- Login form ensures that the email/username exists, and if so, the password entered will be hashed and matched with the hashed password in the database.
- Sell item ensures that all the fields must be filled out, and numerical values such as the quantity of the item listed and its price are logical values (above 0 for quantity, and above \$0).

D. Posting a forum post or service listing ensures that all the fields must be filled out

## VIII. SQL Injection

A. Using preparedStatements and VALUES(?, ?, ?..), the user input given is handled as a parameter instead of pure user input



## [Section VI] Self-check: Adherence to Original Non-Functional Specs

### Functionality

| Non-Functional Spec  | Status      | Comment |
|--|-------------|---------|
| The website shall use the tools and frameworks that the group has agreed on and approved by the CTO. | <u>Done</u> | N/A     |
| The website shall use Amazon Web Services for deployment.  | <u>Done</u> | N/A     |
| The website shall be user friendly and simple.   | <u>Done</u> | N/A     |
| The website shall be easy to navigate for everyone   | <u>Done</u> | N/A     |
| The website shall be available for students and professors at any school.                            | <u>Done</u> | N/A     |

### **Security**

| Non-Functional Spec   | Status             | Comment |
|---|--------------------|---------|
| The user shall need a username/email and password to be able to post and respond to posts on the website. | <b><u>Done</u></b> | N/A     |
| The user shall need an account in order to interact with posts  | <b><u>Done</u></b> | N/A     |
| User registration shall only be available to those with emails.   | <b><u>Done</u></b> | N/A     |
| The user shall need a registered account in order to write reviews for other users.                       | <b><u>Done</u></b> | N/A     |
| Admins shall be able to access all aspects of the websites.   | <b><u>Done</u></b> | N/A     |
| Admins shall be able to moderate the website.   | <b><u>Done</u></b> | N/A     |

### **Privacy**

| Non-Functional Spec   | Status              | Comment  |
|---|---------------------|--|
| The user shall agree to a terms of service agreement in order to access our services. | <b><u>Done</u></b>  | N/A  |
| Passwords and other personal information shall be kept hidden.                        | <b><u>Done</u></b>  | Passwords are hashed and stored into a database                  |
| Users shall be authenticated through username and password.                           | <b><u>Done</u></b>  | N/A  |
| Users shall only be able to sign up if they have an email.                            | <b><u>Done</u></b>  | N/A  |
| Payments shall be collected through an API.   | <b><u>Issue</u></b> | Further testing and research must be done regarding payment APIs |

### **Performance**

| Non-Functional Spec                   | Status             | Comment |
|---------------------------------------|--------------------|---------|
| The website shall be up at all times. | <b><u>Done</u></b> | N/A     |

|   |                    |     |
|---|--------------------|-----|
| The website shall make it easy to post listings.                              | <b><u>Done</u></b> | N/A |
| The website shall make it easy to see other users ratings.                    | <b><u>Done</u></b> | N/A |
| The website shall make it easy to delete old listings.                        | <b><u>Done</u></b> | N/A |
| The website shall make it easy to allow the user to go through payment system | <b><u>Done</u></b> | N/A |

### **System Requirements**

| Non-Functional Spec   | Status             | Comment |
|---|--------------------|---------|
| The website shall work on version 1 to 98 of Mozilla Firefox. | <b><u>Done</u></b> | N/A     |
| The website shall work on version 1 to 99 of Google Chrome.   | <b><u>Done</u></b> | N/A     |

|  |                    |     |
|--|--------------------|-----|
| The website shall work on version 1 to 99 of Microsoft Edge. | <b><u>Done</u></b> | N/A |
| The website shall work on version 1.0 to 15.2 of Safari.     | <b><u>Done</u></b> | N/A |
| The website shall work on version 2.0 83 of Opera.           | <b><u>Done</u></b> | N/A |

### **Marketing**

| Non-Functional Spec                                      | Status             | Comment |
|--|--------------------|---------|
| The website shall be viewable for both guests and users. | <b><u>Done</u></b> | N/A     |
| The website shall have its own logo.                     | <b><u>Done</u></b> | N/A     |
| The website shall be easy to navigate for any user.      | <b><u>Done</u></b> | N/A     |
| The website shall be seen as a familiar alternative      | <b><u>Done</u></b> | N/A     |

|   |                    |     |
|---|--------------------|-----|
| for other online marketplaces.  |                    |     |
| The website shall be seen as friendly for students and faculty of any school. | <b><u>Done</u></b> | N/A |
| The website shall be not directly linked with any specific school             | <b><u>Done</u></b> | N/A |

### **Content**

| Non-Functional Spec  | Status             | Comment |
|--|--------------------|---------|
| The website shall allow users to search posts by categories. | <b><u>Done</u></b> | N/A     |
| The website shall allow users to search by object name       | <b><u>Done</u></b> | N/A     |

|  |                    |     |
|--|--------------------|-----|
| The website shall allow users to search by listing name.         | <b><u>Done</u></b> | N/A |
| The website shall allow users to leave reviews on other users.   | <b><u>Done</u></b> | N/A |
| The website shall allow users to see the reviews of other users. | <b><u>Done</u></b> | N/A |
| The website shall have a section for miscellaneous postings.     | <b><u>Done</u></b> | N/A |

### **Scalability**

| Non-Functional Spec   | Status             | Comment |
|---|--------------------|---------|
| The website shall be able to handle a large number of users from any school.          | <b><u>Done</u></b> | N/A     |
| The website shall have enough categories to sort through the growing number of posts. | <b><u>Done</u></b> | N/A     |

|   |                    |     |
|---|--------------------|-----|
|   |                    |     |
| The website shall be able to accommodate larger amounts of listings.    | <b><u>Done</u></b> | N/A |
| The website shall be able to accommodate large amounts of user reviews. | <b><u>Done</u></b> | N/A |
| The website shall be able to accommodate large amounts of forum posts.  | <b><u>Done</u></b> | N/A |

### **Capability**

| Non-Functional Spec   | Status             | Comment |
|---|--------------------|---------|
| The website shall be able to provide data requested by the user.      | <b><u>Done</u></b> | N/A     |
| The website shall allow posts to be uploaded in a timely manner.      | <b><u>Done</u></b> | N/A     |
| The website shall be able to moderate content within a timely manner. | <b><u>Done</u></b> | N/A     |



|   |                    |     |
|---|--------------------|-----|
| The website shall be able to recover from failures in a timely manner.              | <u><b>Done</b></u> | N/A |
| The website shall be able to communicate directly and effectively with their users. | <u><b>Done</b></u> | N/A |
| The website shall clean up old content within timely manners.                       | <u><b>Done</b></u> | N/A |

### **Look and Feel**

| Non-Functional Spec                     | Status             | Comment |
|---|--------------------|---------|
| The website shall have readable fonts.  | <u><b>Done</b></u> | N/A     |
| The website shall have a simple layout. | <u><b>Done</b></u> | N/A     |
| The website shall be easy to navigate   | <u><b>Done</b></u> | N/A     |
| The website shall load quickly.         | <u><b>Done</b></u> | N/A     |

|  |                    |     |
|--|--------------------|-----|
| The website shall differentiate between different types of listings through service listings and marketplace listings. | <u><b>Done</b></u> | N/A |
| The website shall differentiate between listings and miscellaneous forum postings.                                     | <u><b>Done</b></u> | N/A |
| The website shall visibly show the ratings of other users on their profile   | <u><b>Done</b></u> | N/A |

### **Coding Standards**

| Non-Functional Spec                             | Status             | Comment |
|---|--------------------|---------|
| The code shall be organized and understandable. | <u><b>Done</b></u> | N/A     |

|  |                    |     |
|--|--------------------|-----|
| The code shall have proper working functions.        | <b><u>Done</u></b> | N/A |
| The code shall use git to be pushed and pulled from. | <b><u>Done</u></b> | N/A |
| The code shall use a framework.                      | <b><u>Done</u></b> | N/A |

### **Availability**

| Non-Functional Spec  | Status              | Comment  |
|--|---------------------|--|
| The website shall always be available for users.             | <b><u>Done</u></b>  | N/A  |
| The website shall generate error messages when errors occur. | <b><u>Done</u></b>  | N/A  |
| The website shall refresh when it fails to load.             | <b><u>ISSUE</u></b> | Error checking includes prevention of as many null exceptions and SQL exceptions, but does not refresh when it fails to load |

|  |                    |     |
|--|--------------------|-----|
| The website shall allow postings to be available whenever for users. | <b><u>Done</u></b> | N/A |
| The website shall be available to view for unregistered users.       | <b><u>Done</u></b> | N/A |

### **Fault Tolerance**

| Non-Functional Spec   | Status              | Comment  |
|---|---------------------|--|
| The website shall refresh requests for products if request for purchasing fails.    | <b><u>Issue</u></b> | Error checking includes prevention of as many null exceptions and SQL exceptions, but does not refresh when it fails to load. Based on many tests, our website properly loads product listings |
| The website shall re-attempt to access the database for listings if querying fails. | <b><u>Issue</u></b> | Difficult to requery if the input fields are the same  |

|  |                     |   |
|--|---------------------|---|
| The website shall refresh access requests if users' internet fails.        | <u><b>Done</b></u>  | N/A                                     |
| The website shall notify users and allow user's to retry if payment fails. | <u><b>Issue</b></u> | Could not implement or test payment API |

### **Storage**

| Non-Functional Spec                                | Status                       | Comment  |
|--|------------------------------|--|
| The website shall store usernames in the database. | <u><b>Done</b></u>           | N/A  |
| The website shall store passwords in the database. | <u><b>Done</b></u>           | N/A  |
| The website shall store schools in the database.   | <u><b>On Track/Issue</b></u> | We have some California universities stored in our database, but not a comprehensive list of all the universities in the country or the world. At this point of our project, we didn't have the time to implement colleges throughout our website, |

|  |                    |   |
|--|--------------------|---|
|  |                    | e.g. service listings.<br>Colleges in terms of<br>functional requirements<br>would've been done in a<br>future update |
| The website shall store<br>the user's listings in the<br>database. | <b><u>Done</u></b> | N/A   |
| The website shall store<br>user's ratings in the<br>database.      | <b><u>Done</u></b> | N/A   |

## **[Section VII] List of Contributions**

### **I. Ryan**

- A. Worked on the QA testing plan and Code review
- B. Worked on form validation for login and registration (e.g. making sure fields were filled out, responsive feedback for passwords)
- C. Worked on a post item for more validation and error checking for user input.
- D. Worked on account for user specific clicks
- E. Worked on Adherence to original Non-functional specs

### **II. Duncan**

- A. Worked on the Product Summary
- B. Worked on hashing passwords, including hashing the password upon registration and ensuring login worked with the stored hashed password
- C. Debugged the hashing function for hashing passwords.

### **III. Justin**

- A. Worked on creating and documenting Usability Test Plan.
- B. Worked on product summary and itemized list of committed functions.
- C. Worked on adding a Sell History tab to display the items sold by a user.
- D. Worked on login in verification on posting forums and service listings.

### **IV. Brendan**

- A. Worked on QA Test Plan and Code Review sections
- B. Worked on responsive forum listings, such displaying the forums from the database
- C. Made dynamic user pages that conform to a clicked on user from the forum listing page.

### **V. Darren**

- A. Worked on Self-check: Adherence on Best Practices for Security
- B. Worked on posting forums and service listings (e.g. posting in the database)
- C. Worked on making individualized account pages based on user login

D. Worked on general styling of the website, including to Registration and Sell Item