

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, 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 3

Date: 07/19/2022

Version History

Milestone	Date
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

[Section I] Data Definitions	4
[Section II] Functional Requirements	10
[Section III] Wireframes Based on Mockups/Storyboards	16
[Section IV] High Level Database Architecture and Organization	23
[Section V] High Level Diagrams	36
[Section VI] List of Contributions	37

[Section I] Data Definitions

1. **General User:** A user who is not signed in or does not have an account with College Essentials. A general user has the minimum permissions on the website and can **register** for more permissions.

- 1.1. **Registration:** A general user shall be able to create an account to get more permissions.

- 1.2. **Account Contains:**

- 1.2.1. **Username:** Needs a username to create an account

- 1.2.2. **Password:** Needs a password that is encrypted to create an account.

- 1.2.3. **Email:** Needs a unique edu email to create an account.

- 1.2.4. **School:** Can choose to register a school to the account.

- 1.2.5. **Accept Terms of Service:** Must accept the terms of service to create an account.

2. **Registered User:** A user who has signed up to use College Essentials. This user gains the extra permissions needed to access the other features on the website. A general user shall be able to **login** to gain access to all these features.

- 2.1. Can login to the system.

- 2.1.1. Username: Needs a unique username to login.

- 2.1.2. Password: Needs a unique password to login.

- 2.2. Has the ability to post Marketplace Posts.

- 2.2.1. **Buyers:** All registered users are buyers by default. They can see and interact with Marketplace posts.

- 2.2.2. **Sellers:** All registered users can put up their own Marketplace posts to become a seller. Sellers will be differentiated on the Marketplace post that they have posted.
 - 2.3. Has the ability to post forum Posts.
 - 2.4. **Profile:** Contains username, school, ratings, private messaging section and marketplace posts.
 - 2.4.1. **Username:** Shows the unique username of the user.
 - 2.4.2. **School:** Shows the listed schools linked with the user.
 - 2.4.3. **Ratings:** Shows the ratings on the user posted by other users.
 - 2.4.4. **Private Message:** Has an area that allows users to private message other users.
 - 2.4.5. **Marketplace Posts:** Has a list of marketplace posts made by the user.
 - 2.4.6. **Service Listings:** Has a list of service listings made by the user.
- 3. **Admin:** A user who moderates content on the website. A moderator still has the same permissions that registered users have, but have additional **admin tools** on our website.
 - 3.1. **Admin Tools:** Contains the ability to delete posts, and temporarily restrict users.
 - 3.1.1. **Delete Posts:** Can choose to delete posts by other users.
 - 3.1.2. **Restrict Users:** Can restrict user accounts who break the terms of service.
- 4. **Guest:** This is a user who is unregistered or not logged in who is able to view all posts but not reply. Guests will be ghosts within the site, they are able to search and look but not interact.

- 4.1. **Username:** Will show as a guest with numbers ie. Guest####
 - 4.2. **Browsing:** A guest will be able to browse listings, and add them to cart but not complete the transaction
 - 4.3. **Ads:** A guest shall be able to contact, and pay us to have their ads on the site.
5. **Login / User Registration:** This is the section that would allow users to login or register accounts with College Essentials.
- 5.1. **Login Section:** Users have the ability to login to their accounts here.
 - 5.1.1. **Username:** Users will have to enter a unique username to login.
 - 5.1.2. **Password:** Users will have to enter a unique password to login.
 - 5.2. **Registration Section:** Users have the ability to register new accounts here.
 - 5.2.1. **Username:** Needs a username to create an account
 - 5.2.2. **Password:** Needs a password that is encrypted to create an account.
 - 5.2.3. **Email:** Needs a unique edu email to create an account.
 - 5.2.4. **School:** Can choose to register a school to the account.
 - 5.2.5. **Accept Terms of Service:** Must accept the terms of service to create an account.
6. **Private Message:** This is the section that allows users to private message each other through their profiles.
- 6.1. **Contacting a Person:** Will allow the contacting of a user, client, seller, guest, etc.
 - 6.2. **Timestamp:** Shows when a message was sent.

6.3. **Read Status:** Will show if a message was seen

7. **Forum Post Section:** This is where users are able to post miscellaneous forum posts that don't fit within the other categories. Users must be registered users to make a forum post.

7.1. **Forum Posts:** Contains a post body and post replies.

7.1.1. **Post Body:** A registered user will be able to post a forum post that contains username, post body, and timestamps.

7.1.2. **Post Replies:** A registered user will be able to reply underneath the forum posts that contains a body, the replier's username, and a timestamp.

8. **Marketplace Listings:** A place for registered users to post if they want to buy or sell marketplace listings.

8.1. **Marketplace Listing:** Contains title, location, photo, price, description, expiration date, bidding(If allowed by seller), and tags.

8.1.1. **Title:** Gives a small description of the item at hand.

8.1.2. **Location:** Gives the location of where the deal is to be held.

8.1.3. **Photo:** If provided there will be a picture of the product.

8.1.4. **Price:** Displays the price of the item being sold.

8.1.5. **Description:** A body section that describes the listing.

8.1.6. **Expiration Date:** Displays how long the listing is valid.

8.1.7. **Bidding:** Shows the current minimum bid.

8.1.8. **Tags:** Key words appended to a listing to categorize for searching.

9. **Service Listings:** A place for registered users to post if they need or provide

services.

9.1. **Service Listings:** Contains, a Title, Location, Photo, Price, Description, an expiration date, and tags.

9.1.1. **Title:** Gives a small description of the item at hand.

9.1.2. **Location:** Gives the location of where the deal is to be held.

9.1.3. **Photo:** If provided there will be a picture of the product.

9.1.4. **Price:** Displays the price of the item being sold.

9.1.5. **Description:** A body section that describes the listing.

9.1.6. **Expiration Date:** Displays how long the listing is valid.

9.1.7. **Tags:** Key words appended to a listing to categorize for searching.

10. **User Rating:** This is a section after completing a listing that will allow registered users to leave ratings for the sellers after a purchase. We will provide the section in order to fight against fraud.

10.1. **User Ratings:** Will contain the sellers username, the buyers username, a star rating on the seller, and a description body.

10.1.1. **Seller Username:** This is the username of the seller who provided the listing.

10.1.2. **Buy Username:** This is the username of the buyer who purchased the listing.

10.1.3. **Star Ratings:** This is the section where buyers are allowed to rate a seller out of 5.

10.1.4. **Description:** This is a description box with a body for a short description of the rating.

11. **Payment System:** The place where you choose your payment method for

the services or items involved.

11.1. **Type:** type of the payment.

11.2. **Amount:** The shown amount needed to be paid.

[Section II] Functional Requirements

Priority 1:

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 category 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 email other registered users.
 - 2.13. 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 have basic information like time posted.
- 3.5. A service listing post shall be able to be unlisted.
- 3.6. A navigation bar shall have a section for teachers to look for student assistants.

4. Shopping Cart

- 4.1. A shopping cart shall remember what has been stored within it.
- 4.2. A shopping cart shall remember how many of one entity have been stored within it.
- 4.3. A shopping cart shall allow users to edit products within it.
- 4.4. A shopping cart shall allow for check out.
- 4.5. A shopping cart shall give descriptions of products stored within it

5. Receipt

- 5.1. A receipt shall have a way to contact admin service support.
- 5.2. A receipt shall have the product bought.
- 5.3. A receipt shall have the price of the product bought.

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 the student forums.

7.3. 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.

10. Forum Listing

10.1. A Forum listing shall have a title.

10.2. A Forum listing post shall have characters in its content body.

10.3. A Forum listing post shall have basic information like time posted.

- 10.4. A Forum listing shall be visible to all general users.
- 10.5. A Forum listing post shall be able to be unlisted.

Priority 2:

1. General User

- a. A general user shall be able to leave comments on products/services.
- b. A general user can search by college institution in the marketplace.
- c. A general user can search by college institution in the service listing section.
- d. A general user shall be able to submit their own ads to be promoted on our website.
- e. A registered user shall be able to edit a forum post.
- f. A registered user shall be able to edit a service listing.
- g. A registered user shall be able to edit a sales listing.
- h. A registered user shall be able to pay with different methods, ie. Cash, debit, etc.
- i. A registered user shall be able to pay to be on the top of the forums.
- j. A registered user shall be able to report another user.
- k. A registered user shall be able to block another user.

2. Registered User

- a. A registered user shall do anything an unregistered user can.
- b. A registered user shall rate products they have used.
- c. A registered user shall be allowed to save payment information.
- d. A registered user shall be able to save products they like.
- e. A registered user shall be able to log in using their user id.
- f. A registered user shall be able to log in using their email.

- g. A registered user shall be able to message other registered users.

3. Service Listings

- a. A service listing post shall be able to have filters.
- b. A service listing post shall have optional tags.
- c. A service listing post shall be able to have images in its content body.
- d. A service listing post shall be able to have its content body edited.
- e. A service listing post shall be able to have its title edited.

4. Admin

- a. An admin shall be able to review reports.
- b. An admin shall be able to ban registered users.
- c. An admin shall be able to unban registered users.
- d. An admin shall be able to delete forum posts of other users.
- e. An admin shall be able to remove service listings from the website.
- f. An admin shall be able to remove marketplace listings from the website.
- g. An admin shall be able to view a chat log between two users.
- h. An admin shall be able to do all that a registered user can.
- i. An admin shall be able to see personal information of users.
- j. An admin shall be able to close group posts.
- k. An admin shall be able to re make posts.
- l. An admin shall be able to see IP's.

5. Receipt

- a. A receipt shall have a way to contact admin service support.
- b. A receipt shall have the method of contacting the seller via email..

- c. A receipt shall have the method of contacting the seller via phone number.
- d. A receipt shall have the time of the transaction.

6. Shopping Cart

- a. The shopping cart shall have a button to relocate you to the marketplace

7. Navigation Bar

- a. The Navigation Bar shall have zip code search.

8. Messages

- a. A message shall have characters in its content body.
- b. A message shall have a timestamp.
- c. A message shall have a status (sent/received/read).
- d. A message shall allow a user to edit it.
- e. A message shall allow a user to delete it.

9. Marketplace Listing (Buy/Sell Listing)

- a. A marketplace listing shall have the location of where the deal is going to be held.
- b. A marketplace listing shall have an expiration date.
- c. A marketplace listing shall have a bidding minimum.
- d. A marketplace listing shall have optional tags.
- e. A marketplace listing shall have the option to be returned or not.

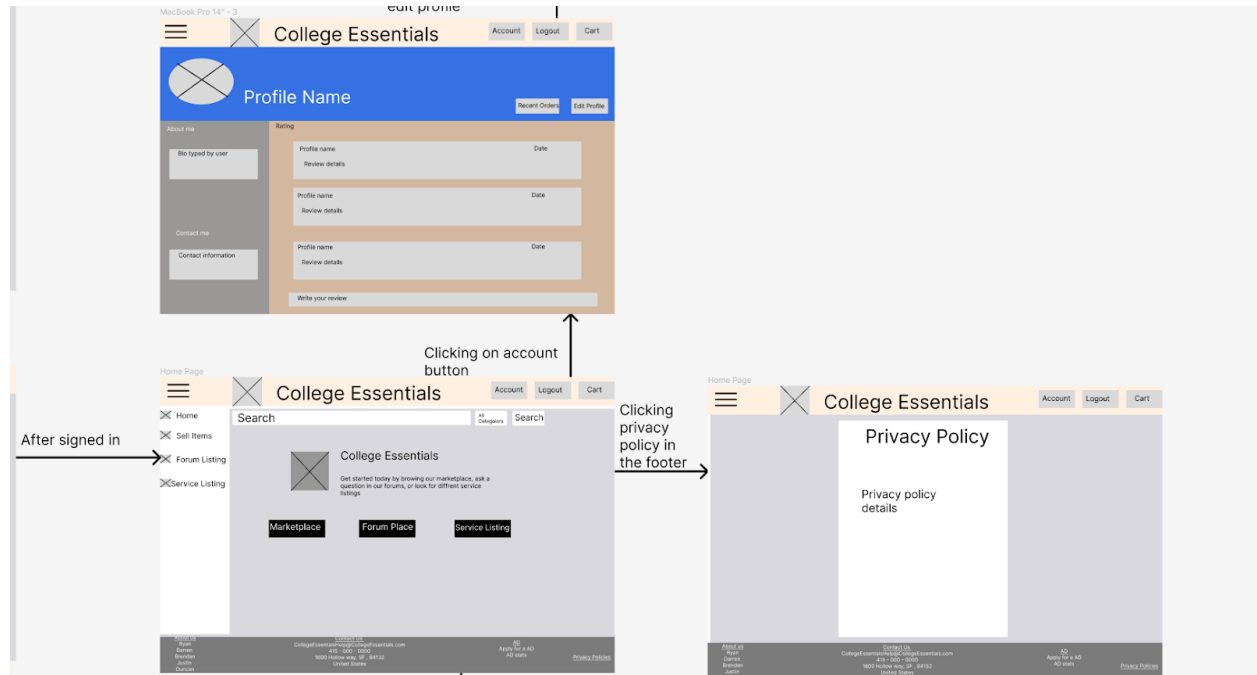
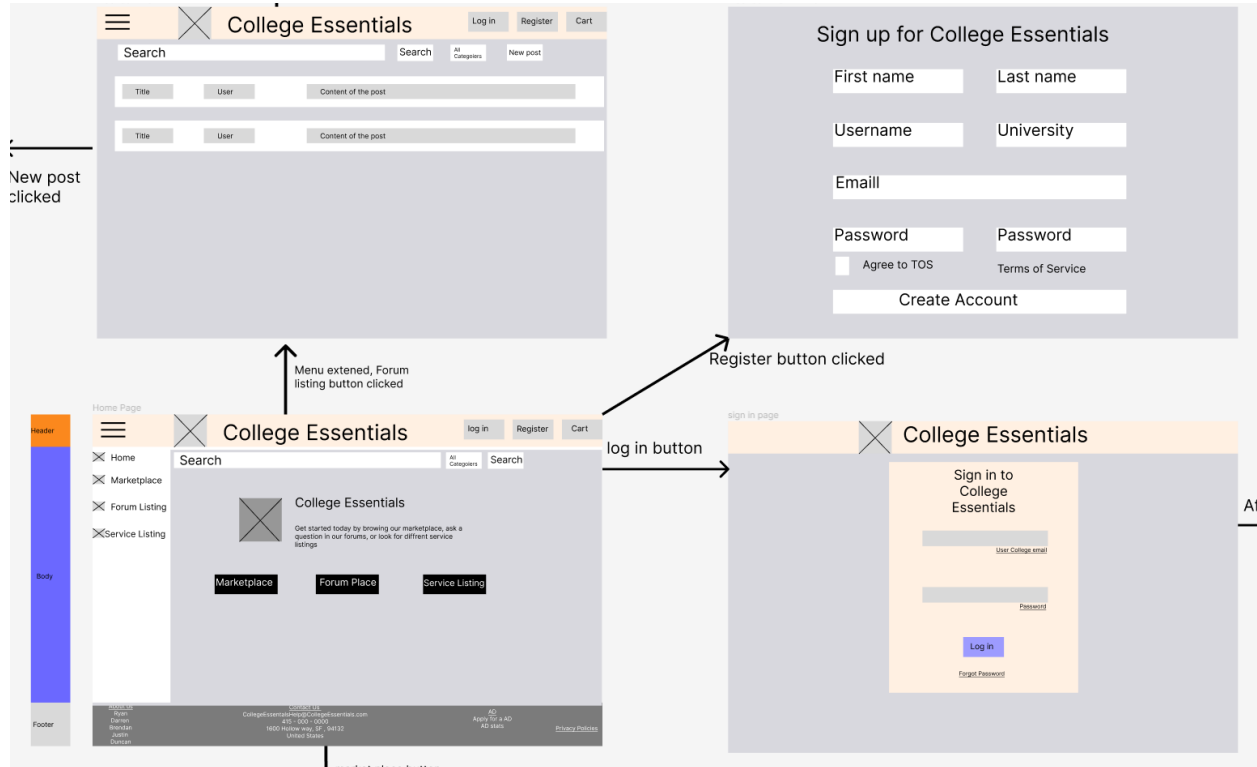
Priority 3:**1. Shopping Cart**

- a. A shopping cart will have a little number showing how many items are within it, while you are not on the checkout screen.

2. Checkout

- a. Checkout shall prompt users to create an account if not yet registered.

[Section III] Wireframes Based on Mockups/Storyboards



MacBook Pro 14" - 6



College Essentials

Account

Logout

Cart



Profile Name

About me

Bio typed by user

Contact me

Contact information

Change college

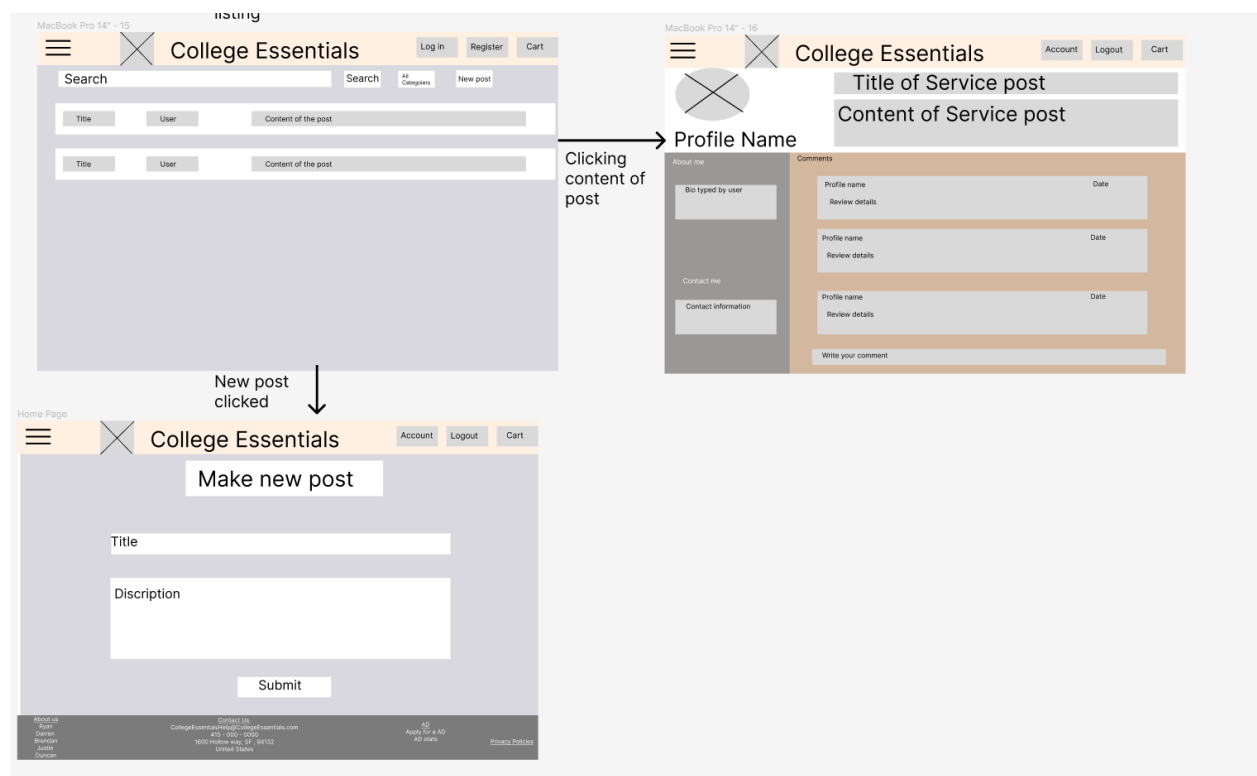
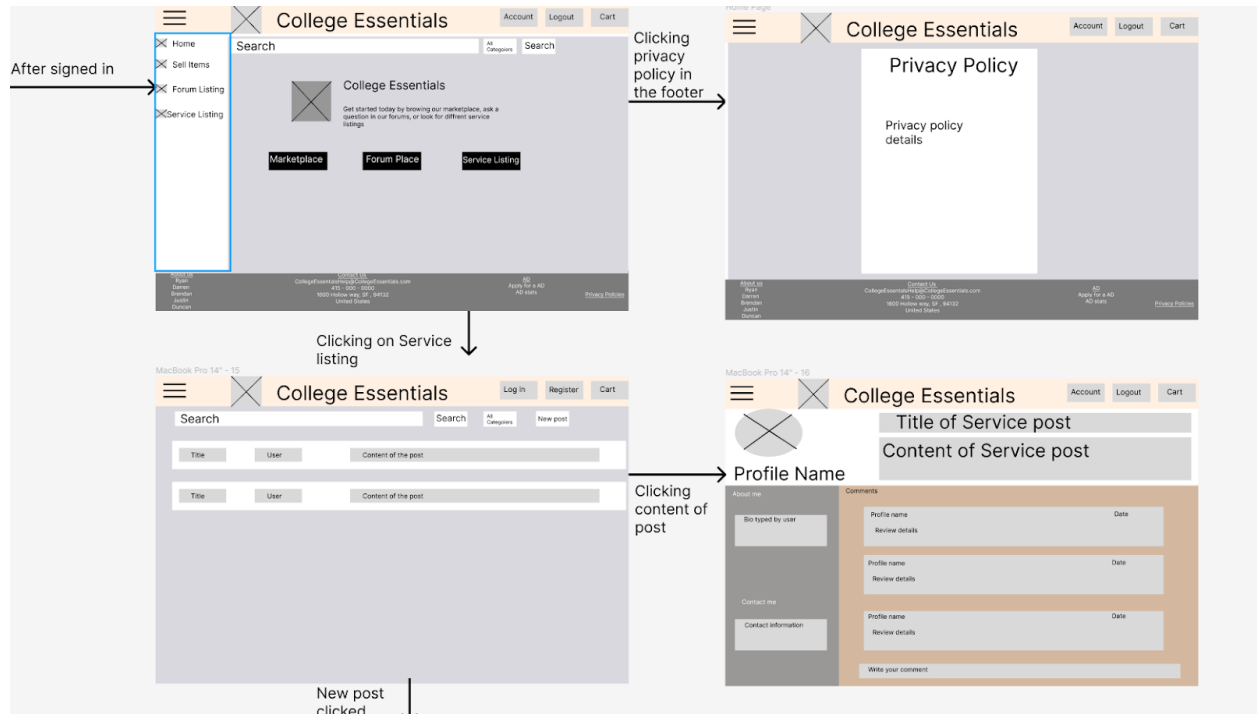
College email

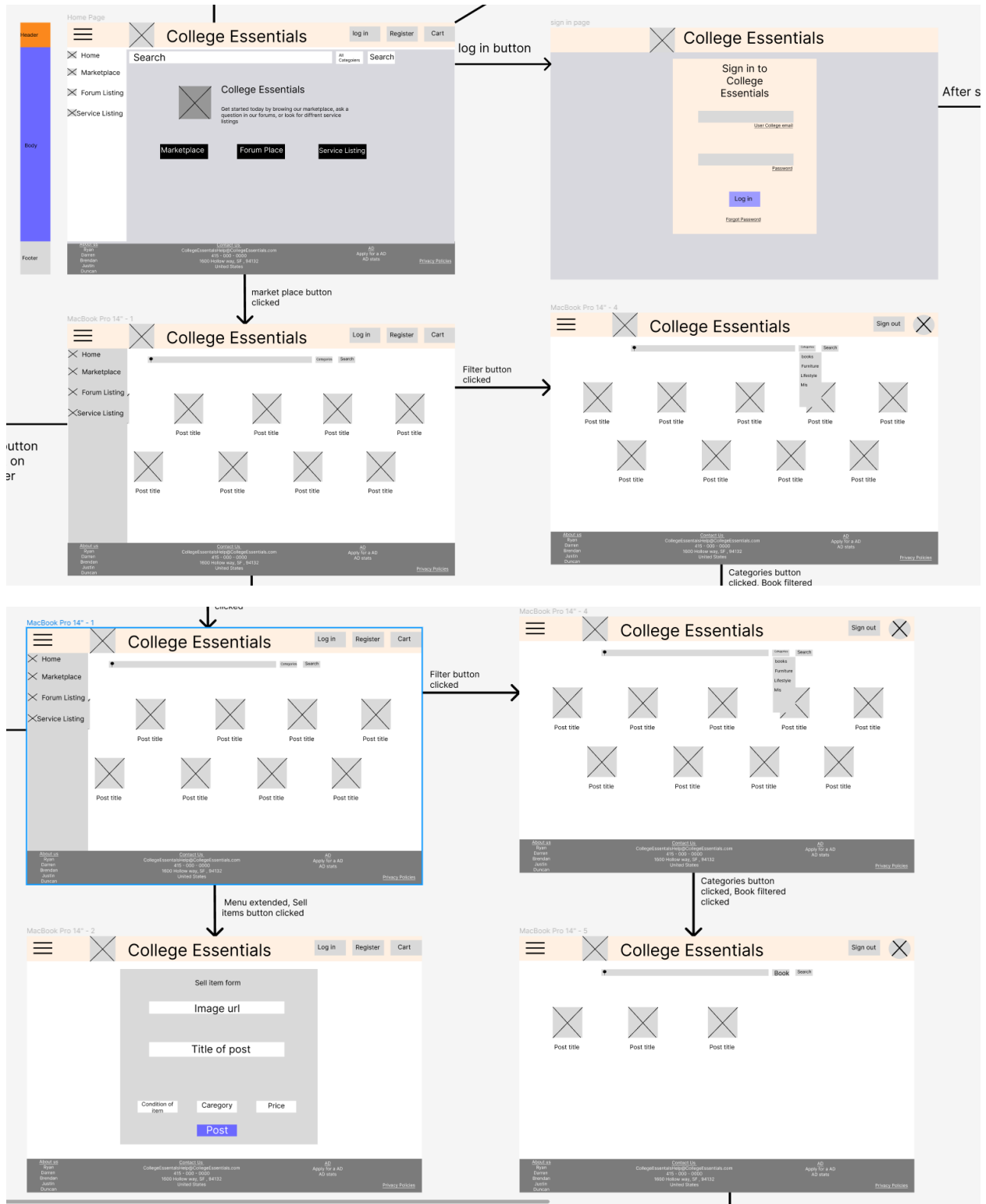
Submit

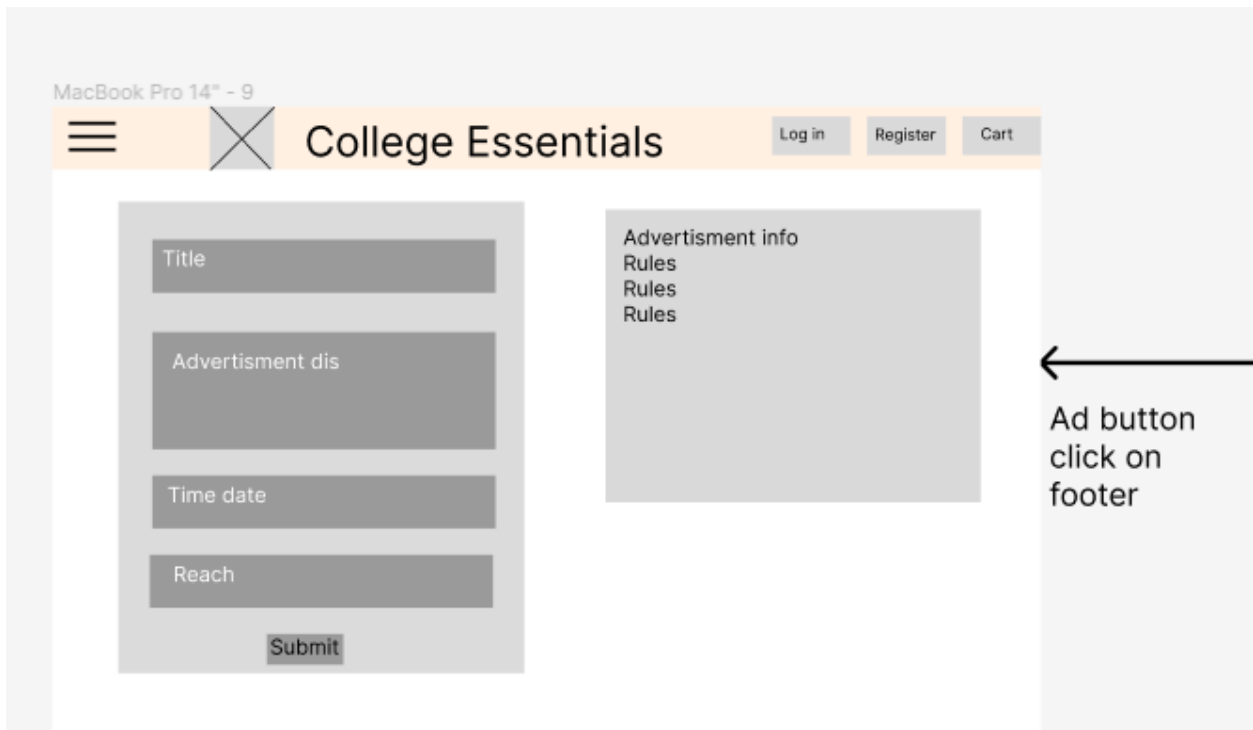
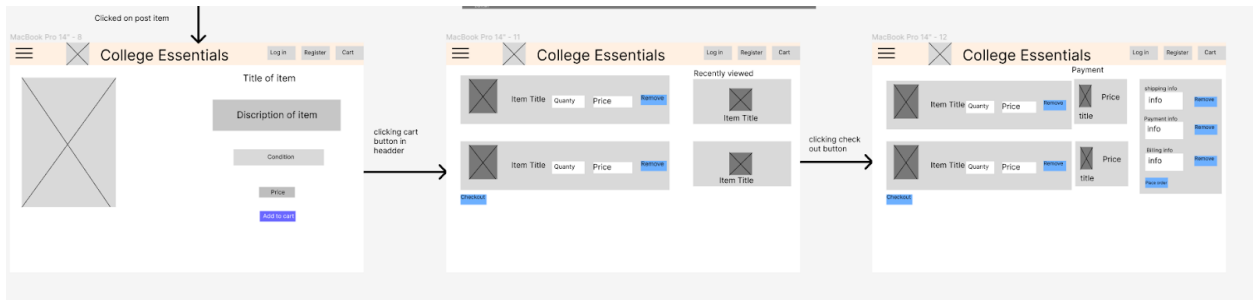
Clicking on
edit profile

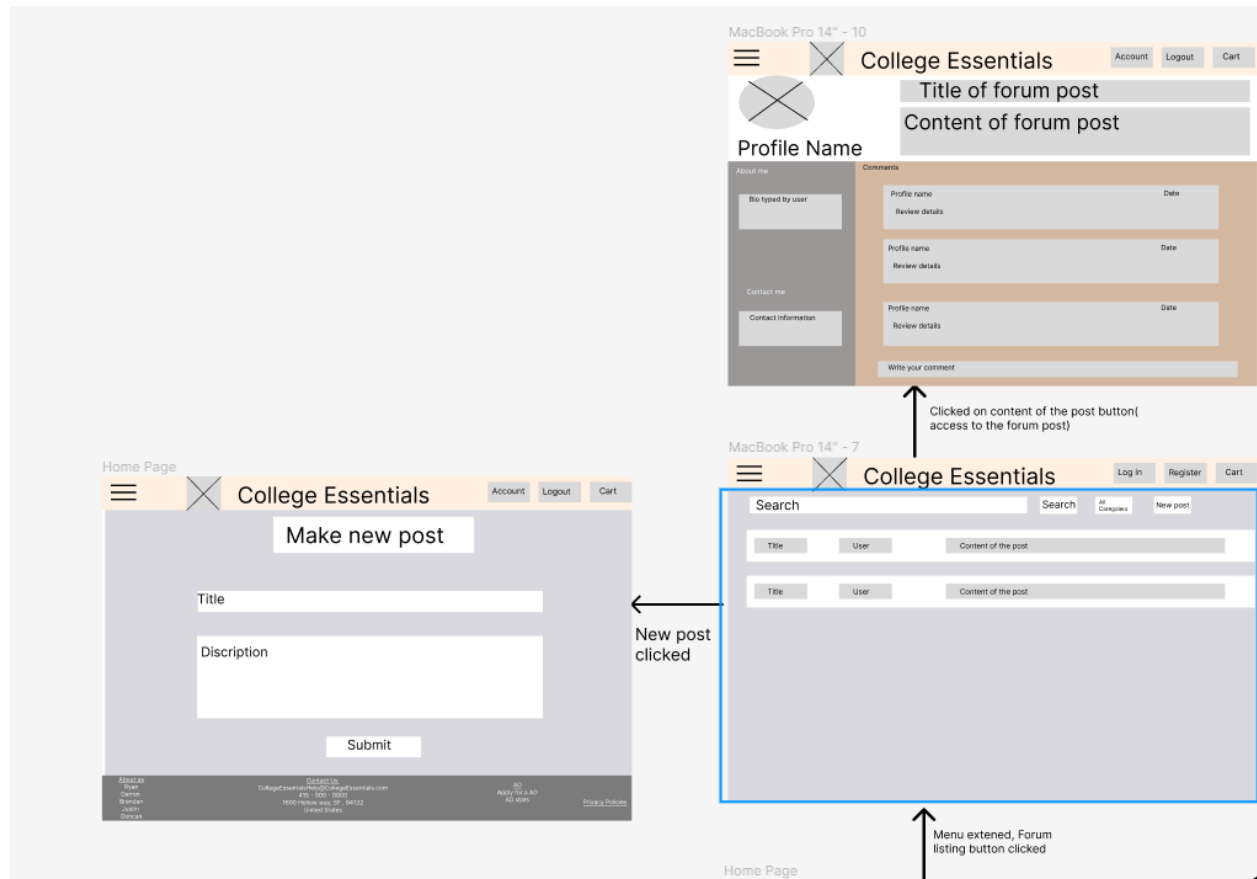


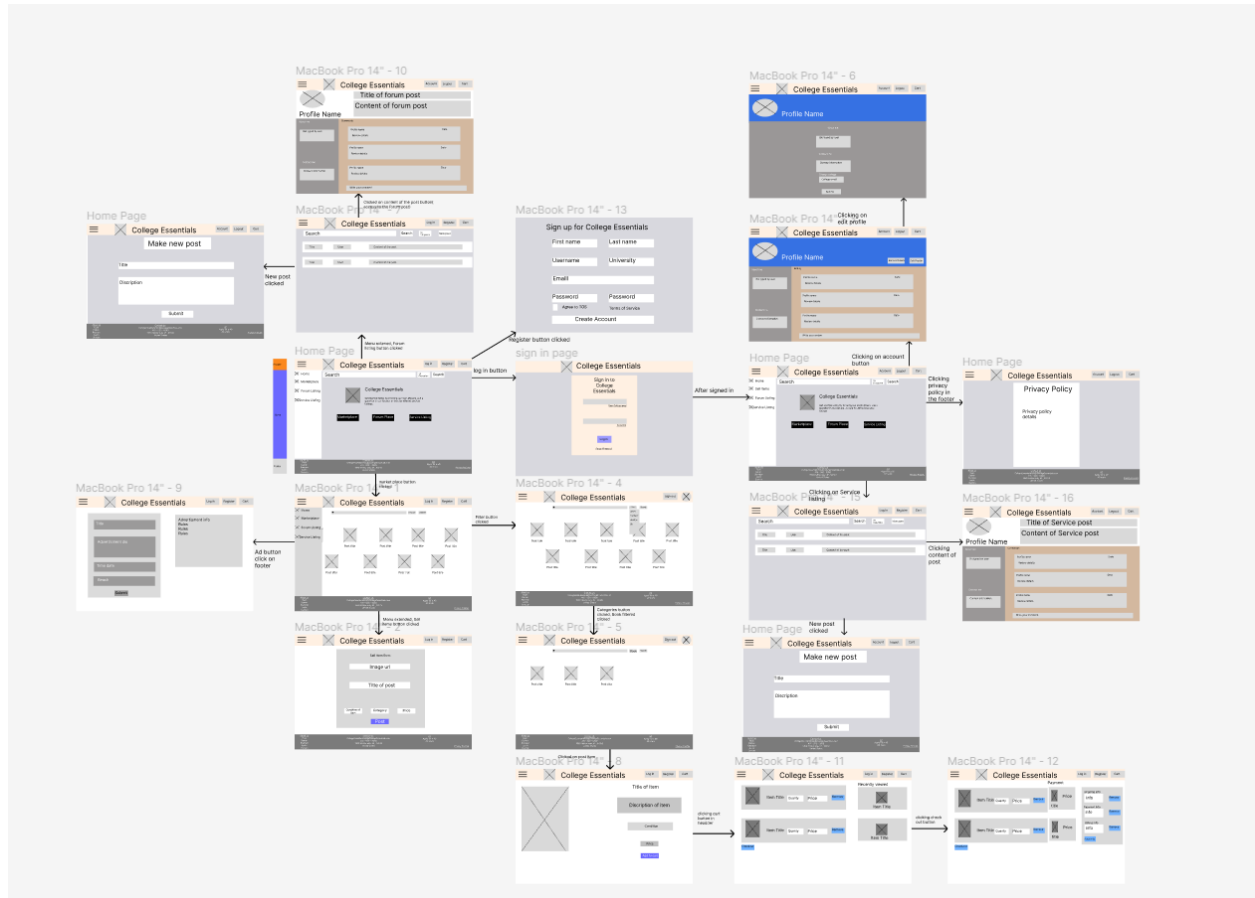
MacBook Pro 14" - 3











Link to all the diagrams:

<https://www.figma.com/file/nq07WHQeXEGWxPcPPLDjl7/Untitled?node-id=0%3A1>

[Section IV] High Level Database Architecture and Organization

1. Business Rules

a. GeneralUser

- i. A GeneralUser shall be able to log into a registered account from many devices. (RegisteredUser)
- ii. A GeneralUser shall have a ShoppingCart.
- iii. A GeneralUser shall be able to checkout their purchases.
- iv. A GeneralUser shall be able to view the marketplace.
- v. A GeneralUser shall be able to view the service listings.
- vi. A GeneralUser shall be able to view the forum listings.
- vii. A GeneralUser shall receive many receipts for their transactions.

b. RegisteredUser

- i. A RegisteredUser shall be able to be an Admin
- ii. A RegisteredUser shall have one or many Colleges associated with their accounts.
- iii. A RegisteredUser shall be able to post zero or many ServiceListings.
- iv. A RegisteredUser shall be able to post zero or many ServiceReplies
- v. A RegisteredUser shall be able to post zero or many ForumPosts
- vi. A RegisteredUser shall be able to post zero or many ForumReplies
- vii. A RegisteredUser shall have zero or many Ratings.
- viii. A RegisteredUser shall be able to post zero or many Ratings.

ix. A RegisteredUser shall have many forms of Payment

c. Rating

- i. A Rating shall belong to one RegisteredUser.
- ii. A Rating shall be posted by zero or many RegisteredUsers.
- iii. A Rating shall be tied to zero or one Transaction

d. Admin

- i. An Admin shall be considered a RegisteredUser.

e. College

- i. A College shall be associated by many RegisteredUsers (e.g. college student or college faculty)

f. MarketplaceListing

- i. A Marketplace Listing shall be listed by one RegisteredUser
- ii. A Marketplace Listing shall list at least one Product

g. ServiceListing

- i. A ServiceListing shall be listed by one RegisteredUser.
- ii. A ServiceListing shall be followed by many ServiceReplies

h. ServiceReply

- i. A ServiceReply will be tied to many ServiceListings
- ii. A ServiceReply is posted by a RegisteredUser

i. Transaction

- i. A Transaction shall have many Receipts.
- ii. A Transaction shall be linked to one Payment method.
- iii. A Transaction shall be linked to a Buyer
- iv. A Transaction shall be linked to a Seller
- v. A Transaction shall be linked to a Rating
- vi. A Transaction shall be linked to many TransactionItems

j. TransactionItems

- i. A TransactionItems is tied to a Transaction.
- ii. A TransactionItems is tied to a Product

k. ShoppingCart

- i. A ShoppingCart is tied to one GeneralUser.
- ii. A ShoppingCart is linked to one Transaction.

l. Payment

- i. A Payment shall belong to one RegisteredUser.
- ii. A Payment shall be linked to a Transaction.

m. Receipt

- i. A Receipt shall be given to one RegisteredUser (buyer).
- ii. A Receipt shall be given to one RegisteredUser (seller).
- iii. A Receipt shall belong to one Transaction.

n. Product

- i. A Product is ordered by zero or many GeneralUsers
- ii. A Product is added to many ShoppingCarts

- iii. A Product is listed on a MarketplaceListing
- iv. A Product is tied to many TransactionItems
-
- o. ForumPost
 - i. A ForumPost is posted by one RegisteredUser
 - ii. A ForumPost shall be followed by many ForumReplies
-
- p. ForumReply
 - i. A ForumReply will be tied to many ForumPost
 - ii. A ForumReply is posted by a RegisteredUser

2. Entities

- a. GeneralUser (Strong)
 - i. General_user_id: key, numeric; This is the unique id registered to each GeneralUser
 - ii. ip_address: numeric; This is the IP address associated with a GeneralUser
-
- b. RegisteredUser (Weak)
 - i. registered_user_id: key, numeric; This is the unique id registered to each RegisteredUser.
 - ii. general_user_id: key, numeric; This is the id associated to the generalUser tied to RegisteredUser
 - iii. first_name: alphanumeric; This is the first name registered to a RegisteredUser.
 - iv. last_name: alphanumeric; This is the last name registered to a RegisteredUser

- v. password: alphanumeric; This is the unique password registered to each RegisteredUser.
- vi. email: alphanumeric; This is the unique username registered to each RegisteredUser.
- vii. about: alphanumeric; This is the about info tied to a RegisteredUser
- viii. contact: alphanumeric; This is the contact info tied to a RegisteredUser
- ix. is_banned: numeric; This is a 0 or 1 value that denotes if the RegisteredUser is banned or not
- x. pfp: alphanumeric; This is the image link tied to a RegisteredUser's profile picture

c. Ratings (Weak)

- i. rating_id: key, numeric; This is the unique id given to the specific rating.
- ii. rating: numeric: This is a value between 0 and 5 given to the rating.
- iii. description: alphanumeric: This is a short description of the given rating.
- iv. transaction_id: key, numeric; This is the transaction linked to the rating.
- v. registered_user_sender_id: key, numeric; This is the ID of the user giving/sending a rating
- vi. registered_user_receiver_id: This is the ID of the user receiving the rating
- vii. posted_at: timestamp; This is the timestamp of when the rating

was posted

d. Admin (Weak)

- i. admin_id: key, numeric; This is the unique id linked to an Admin account.
- ii. registered_user_id: key, numeric; This is the RegisteredUser id associated with an Admin

e. College (Strong)

- i. college_id: key, numeric; This is the unique college id associated with each college
- ii. name: alphanumeric; This is the name for a college

f. MarketplaceListing (Strong)

- i. marketplace_listing_id: key, numeric; This is the unique id linked with the MarketplaceListing.
- ii. title: alphanumeric; This is the title given to the MarketplaceListing.
- iii. body: alphanumeric; This is the body given to the MarketplaceListing.
- iv. created_at: timestamp; This is the timestamp given for when a marketplace listed was posted
- v. seller_id: key, numeric; This is the unique seller id used in a transaction.
- vi. price: numeric; This is the price associated with a MarketplaceListing
- vii. quantity: alphanumeric; This is the quantity of the product

listed

g. ServiceListing (Strong)

- i. service_listing_id: key, numeric; This is the unique id linked with the ServiceListing.
- ii. title: alphanumeric; This is the title given to the Service Listing.
- iii. description: alphanumeric; This is the description given to the ServiceListing.
- iv. created_at: timestamp; This is the timestamp given to the ServiceListing.

h. ServiceReply (Weak)

- i. service_reply_id: key, numeric; This is the unique id linked with the ServiceReply.
- ii. description: alphanumeric; This is the description given to the ServiceReply
- iii. posted_at: timestamp; This is when the ServiceReply was posted

i. Transaction (Strong)

- i. transaction_id: key, numeric; This is the unique id of the transaction.
- ii. completed_at: timestamp; This is the timestamp of when a transaction is completed
- iii. payment_id: key, numeric; This is the unique id of a payment
- iv. seller_id: key, numeric; This is the unique id associated with the seller (RegisteredUser) in the transaction

- v. buyer_id: This is the unique id associated with the buyer (RegisteredUser) in the transaction

j. TransactionItems (Weak)

- i. Transaction_id: numeric; This is the transaction id associated with a Transaction
- ii. product_id: numeric; This is the product id associated with a TransactionItem
- iii. title: alphanumeric; This is the title associated with a product
- iv. image: alphanumeric; This is the image associated with a product
- v. Price: alphanumeric; This is the product associated with a product
- vi. quantity: alphanumeric; This is the quantity associated with the product

k. ShoppingCart (Strong)

- i. shopping_cart_id: key, numeric; This is the unique id for a ShoppingCart
- ii. buyer_id: key, numeric; This is the unique id for a Buyer (GeneralUser)

l. Receipt (Strong)

- i. receipt_id: key, numeric, This is the unique id of the receipt given in the transaction.
- ii. seller_id: key, numeric; This is the unique id of the seller in the transaction.

- iii. buyer_id: key, numeric; This is the unique id buyer in the transaction.
- iv. transaction_id: key, numeric; This is the unique id of the transaction.

m. Payment (Strong)

- i. Payment_id: key, numeric; This is the unique id given to the payment.
- ii. payment_type: alphanumeric; This is the type of payment used by the buyer.
- iii. card_number: numeric; This is the number associated with a card

n. Product (Strong)

- i. product_id: key, numeric; This is the unique id of the product
- ii. general_type: alphanumeric; This is the general type of the product (e.g. a textbook, bicycle, furniture, etc.)
- iii. picture: alphanumeric; This the picture link associated with the product
- iv. Condition: alphanumeric; This is the condition associated with a product

o. ForumPost (Strong)

- i. forum_id: key, numeric; This is the unique id of a ForumPost
- ii. title: alphanumeric; This is the title of a ForumPost
- iii. description: alphanumeric; This is the description of a ForumPost

iv. created_at: timestamp; This is when a ForumPost was created

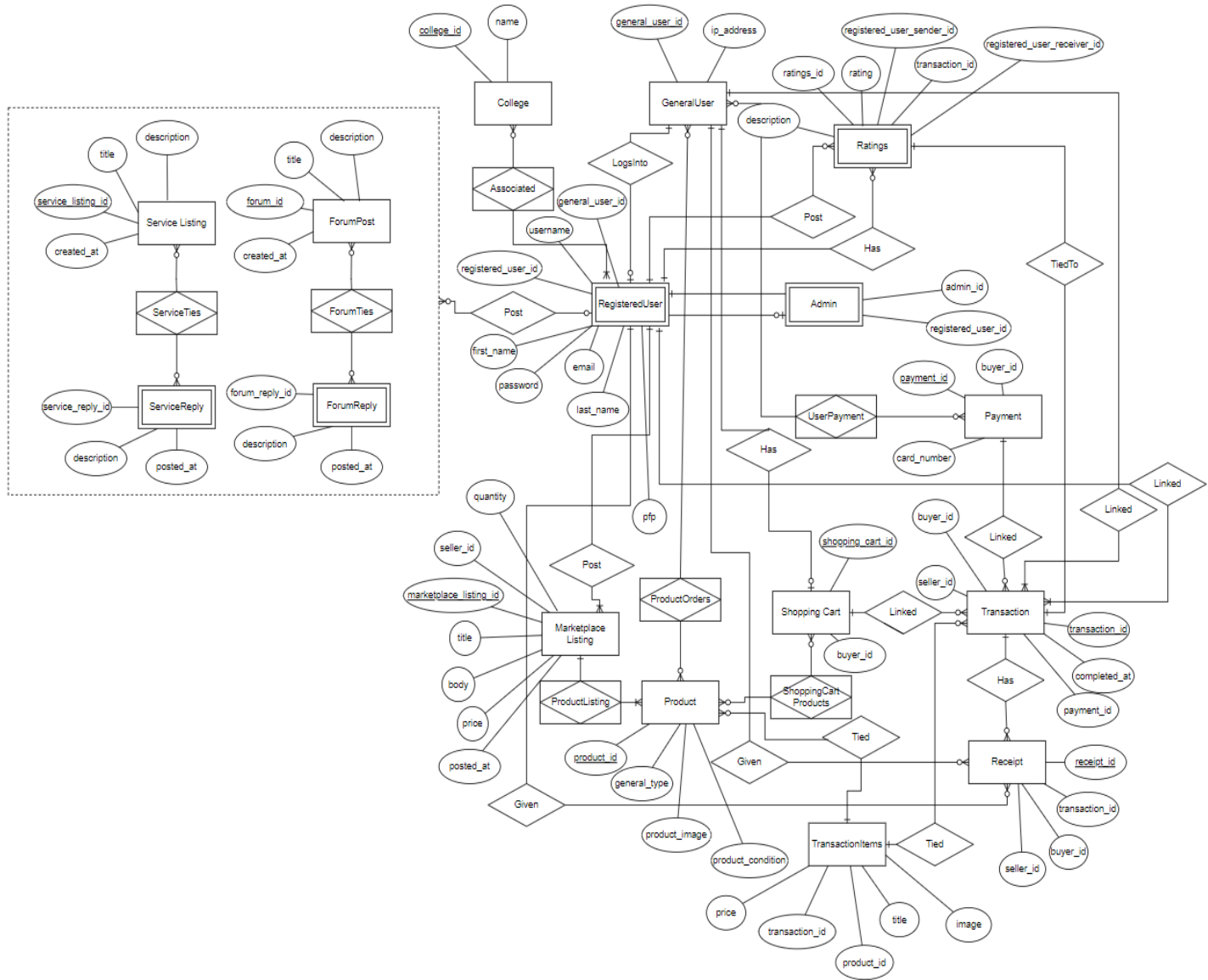
p. ForumReply (Weak)

i. forum_reply_id: key, numeric; This is the unique id of a ForumReply

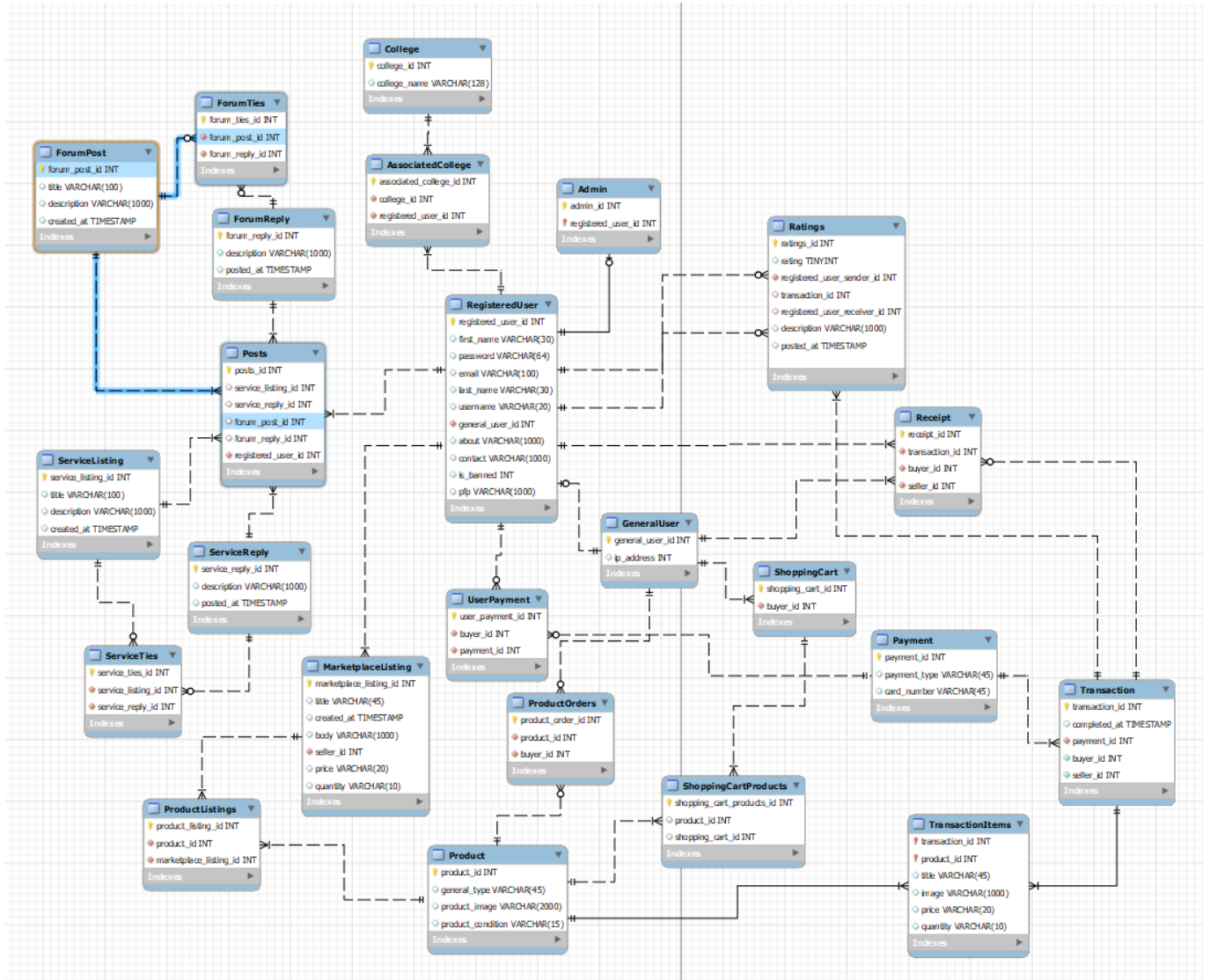
ii. description: alphanumeric; This is the description of a ForumReply

iii. posted_at: timestamp; This is when a ForumReply was posted

3. Entity Relationship Diagram (ERD)



4. Database Model (EER)



5. DBMS

College Essentials will use MySQL as its DBMS, given its widely compatible and easy to use nature. MySQL has many tools available, such as MySQL Workbench, and continues to grow alongside cloud data platforms.

6. Media Storage

The College Essentials database will store images in links within the Product table, but the image and files themselves will be stored within the project's file system. This will allow us to keep the files in a consistent location, and allow us to easily be able to locate and reference these files. Because all files will be in the file system in a possible "uploads" folder, we would be able to just implement a path in the SQL database to be able to point to the file.

7. Search/Filter Architecture and Implementation

- a. Our search algorithm will allow the user to use a search bar for terms within a marketplace listing. There will also be a drop down menu for specifying the condition of a product that will work in tandem with the search bar. At this time, we have filters that allow the user to search based on condition, but we plan to allow users to search based on college as well.
- b. In order to implement the search filters, we will have a table in the database that would show the specific keywords linked to each marketplace listing. Here is an example of how the search would be implemented based on no filters, using SQL LIKE to search for the keyword within the title of the marketplace listings:

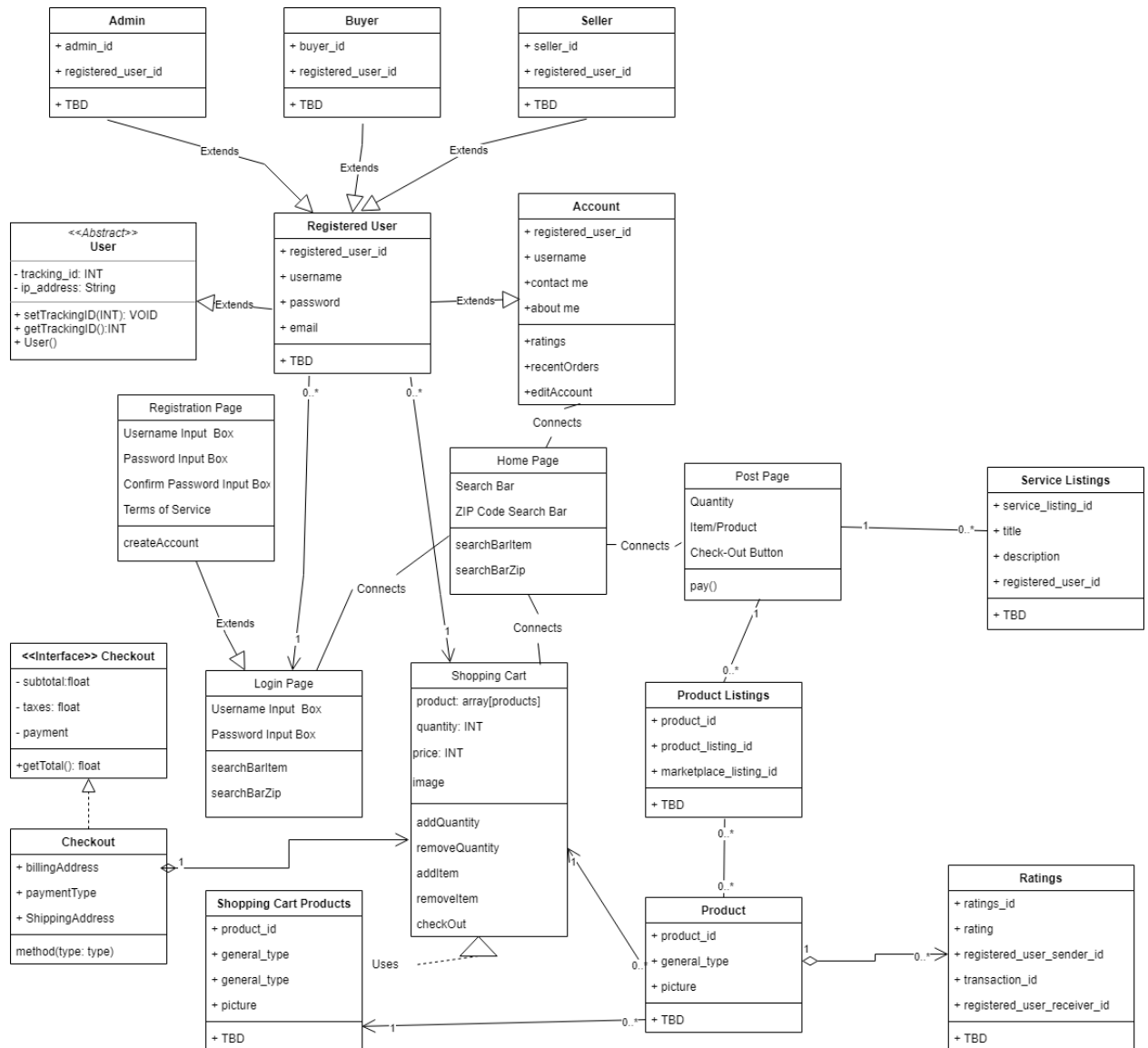
```
SELECT MarketplaceListing.title,  
Product.product_image FROM ProductListings  
JOIN MarketplaceListing ON  
ProductListings.marketplace_listing_id =  
MarketplaceListing.marketplace_listing_id  
JOIN Product ON ProductListings.product_id =  
Product.product_id
```

```
WHERE MarketplaceListing.title LIKE '%tes%';
```

- c. When searching for products using a condition filter, we will use a similar SQL LIKE to search for the specific keyword given, however, the product condition will be added using the drop down menu with the SQL AND function. A Query example would be:

```
SELECT MarketplaceListing.title,  
Product.product_image, Product.condition FROM  
ProductListings  
JOIN MarketplaceListing ON  
ProductListings.marketplace_listing_id =  
MarketplaceListing.marketplace_listing_id  
JOIN Product ON ProductListings.product_id =  
Product.product_id  
WHERE MarketplaceListing.title LIKE '%searchterm%'  
AND Product.condition = "NEW";
```

[Section V] High Level Diagrams



[Section VI] List of Contributions

I. Ryan

- A. Facilitated group meetings and plans
- B. Worked on categorizing the priorities with the group
- C. Worked on frontend for the horizontal prototype
 - 1. Worked on the CSS for the login, register page, PostItem, SellItem, Home Page, and item View
 - 2. Worked on the CSS for the footer along with the java
 - 3. Worked on fixing Home backend, itemView, login, registration
- D. Worked on backend for the vertical/prototype
 - 1. Worked to refactor putting marketplace listing products into the remote database
 - 2. Worked to fix the search features for Marketplace listings (back-end)

II. Duncan

- A. Created all the wireframes drawings for Milestone 3.
- B. Revised all High level UI Mock up drawings in Milestone 2.
- C. Revised use cases that were given in Milestone 1.
- D. Created the UI design of the Front end of each page (not the coding).

III. Justin

- A. Worked on categorizing the priorities with the group.
- B. Revised the functional requirements on Milestone 3.
- C. Revised the data definitions for Milestone 3.
- D. Revised the UML diagram for Milestone 3.
- E. Worked on the frontend for the horizontal prototype.

1. Designed and wrote the code for the UI for the Shopping Cart and Checkout pages.
2. Worked on the routing for the Advertisements, Privacy Policies, and Term of Service pages.
3. Created the UI for the Advertisement page, designing an easy forum for users to fill out.
4. Added CSS for the Shopping Cart and Checkout pages.

IV. Brendan

- A. Assisted in altering CSS for various part of the site
 1. Login CSS assistance
 2. Home page search bar CSS assistance
- B. Worked on routing related to the forum and service listing
 1. Routing to proper external links
 2. Routing to proper dummy user pages mentioned in service and forum listing
- C. Created dummy user account pages for proper routing.
- D. Worked on UI for the pages of forum and service listing
 1. Implemented grid design to forum pages for a for fulfilling design
 2. A smoother design for forum and service listing for easy reading, and proper design to fit with background

V. Darren

- A. Worked on categorizing the priorities with the group and reworked the documentation on the database (section IV)
- B. Worked on revising the database (on MySQL and AWS) based on the feedback from M2V1
- C. Worked on frontend for the horizontal prototype
 - 1. Made the home page more user friendly
 - 2. Made the account the page
 - 3. Added css to various details in the website, including the footer, buttons, and menu
- D. Worked on backend for the vertical/prototype
 - 1. Worked to refactor putting marketplace listing products into the remote database
 - 2. Worked to fix the search features for Marketplace listings (back-end)

[Appendix I] Milestone 3 Review: Project Status and UI Review

1. UI and Functionality Feedback

- a. Home screen has a good hook
- b. Website is relatively user friendly and intuitive
 - i. e.g. Sell Item leads to a form to post items to sell
 - ii. e.g. Forum Listings lead to a centralized hub for forum posts
 - iii. e.g. Service Listings lead to a centralized hub for service listings
 - iv. Most buttons and entry fields are responsive
- c. Input fields should be validated
- d. Too much blank space for ForumPost description and ServicePost listing
 - i. Needs more dynamic sizing
- e. Home screen doesn't utilize space well, needs better alignment and spacing
 - i. Needs images or something to display what each button represents (something for marketplace, another for forum listing, and another for service listing)
- f. Website needs a logo instead of plain text "College Essentials"

2. Brief Review of Coding, Github, Database, etc.

- a. NOTE: CTO did not review code, Github, Database during the meeting. This is based on feedback from the meeting about our practices in general along with feedback from the milestone doc in regards to the database
- b. Input fields must be validated to ensure they are not empty or pass in illogical values (e.g. negative values for quantity)

- c. Database is done relatively well, “Relationships are beautifully implemented as per database requirements “ (per M2 feedback too)

3. Project Status

- a. Teamwork:

- i. The team dynamic has been overall positive so far. Some of the problems that have arisen in regards to team dynamic is the amount of work that some teammates have compared to others. As a team overall, we should try to abide by timelines better, and communicate with the group the progress on our individual parts. Overall, we are working well together as a team. We are able to keep each other accountable, and make sure that even if we are late on our deadlines, we are still getting the work done in the end.

- b. Risks:

- i. As a team, our main risk is that we are attempting to have more functional and nonfunctional requirements than what we should have attempted. Because our group is smaller, each person has to contribute more work than a group with more members would have to. That means that we will inevitably fail to meet some functional and nonfunctional requirements. To mitigate these issues we have been pushing back some functional requirements to priority 2, and focusing on being able to deliver all of our priority 1 requirements.

- c. Coding Practices:

- i. 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.

d. Usage of Proper SE Code Management Practices:

- i. For code submissions, we work on individual parts that need to be done, and test them with the rest of the application to make sure that everything still works. The workflow is done by assigning each individual member a section to work on, then having all the other team members available to review the code before pushing it to the development branch on github. This process repeats for all team members and any work that they have to do. After all the work is done for a section, we review the working prototype, and finally push to master and deploy into AWS.

e. How did you address Site Security and Safe Coding Practices

- i. For site submissions, passwords have the ability to be shown or hidden on registration and login. Passwords are also hashed and stored into the database instead of plain text. Insert statements to the remote DB and protected from SQL injections using `VALUES(?, ?, ..)` etc.

f. Digital Content:

- i. All images that we use for the application use external urls that are saved in the database. That means that the database only saves the url of the image, and does not have to store the image itself. Any page that needs the images pulls the url from the database to populate the pages. Any formats or resolutions are allowed, but the resolutions are chosen by the website.

4. List of P1 Features Committed for Delivery

a. General User

- i. A general user shall be able to browse forum posts.
- ii. A general user shall be able to browse items on the marketplace.
- iii. A general user shall be able to browse service listings.
- iv. A general user shall be able to search for their needed service.
- v. A general user shall accept our terms of conditions.
- vi. A general user shall be able to search by item condition in the marketplace.

b. Registered User

- i. A registered user shall be able to log in with their username.
- ii. A registered user shall be able to review another user.
- iii. A registered user shall be able to create their own profile with text.
- iv. A registered user shall be able to create their own profile with pictures.
- v. A registered user shall be able to edit their own profile with text.
- vi. A registered user shall be able to edit their own profile with pictures.
- vii. A registered user shall be able to post a picture as their profile picture.
- viii. A registered user shall be able to post a service listing.

- ix. A registered user shall be able to list items for sale in the marketplace.
- x. A registered user shall be able to post on the forums.
- xi. A registered user shall be able to log out.
- xii. A registered user shall be able to rate other general users.

c. Service Listing

- i. A service listing shall have a title.
- ii. A service listing post shall have characters in its content body.
- iii. A service listing post shall have basic information like time posted.
- iv. A service listing post shall be able to be unlisted.

d. Forum Listing

- i. A Forum listing shall have a title.
- ii. A Forum listing post shall have characters in its content body.
- iii. A Forum listing post shall have basic information like time posted.
- iv. A Forum listing shall be visible to all general users.
- v. A Forum listing post shall be able to be unlisted.

e. Shopping Cart

- i. A shopping cart shall remember what has been stored within it.
- ii. A shopping cart shall remember how many of one entity have been stored within it.
- iii. A shopping cart shall allow users to edit products within it.
- iv. A shopping cart shall allow for check out.

- v. A shopping cart shall give descriptions of products stored within it

f. Checkout

- i. Checkout shall allow a user to input their payment information.
- ii. Checkout shall allow a user to input their shipping address.
- iii. Checkout shall allow a user to input their billing address.
- iv. Checkout shall allow users to review their purchase on last time.

g. Navigation Bar

- i. A navigation bar shall have a section for students selling their products.
- ii. A navigation bar shall have a section for teachers to look for student assistants.
- iii. A navigation bar shall have a section for the student forums.
- iv. A navigation bar shall have a section for the about us page.

h. Marketplace Listing (Buy/Sell Listing)

- i. A marketplace listing shall have a title.
- ii. A marketplace listing shall have a description.
- iii. A marketplace listing shall have a photo (or many).
- iv. A marketplace listing shall have a price.
- v. A marketplace listing shall be able to be unlisted.
- vi. A marketplace listing shall have filters to categorize.

i. Ratings

- i. User ratings shall rate a profile on a scale of 1 to 5.
- ii. User ratings shall be visible on a registered user's profile.
- iii. User ratings shall be visible on a registered user's marketplace listing.
- iv. User ratings shall be visible on a registered user's service listing.
- v. User ratings shall be able to be posted by registered users.

[Appendix II] Milestone 3 Review: SW Review

- Git/Github organization (e.g. organization of branches)
 - Our group has only been pushing to development, and then pushing to master only at the end of each milestone. We can keep working in the development branch, but we need to push to master more often, such as once a week.
- Git/Github usage
 - Our github usage is in general good. We have an even number of submissions between the team members, and we post to github very often.
- Code documented (header, in code) with good coding style
 - Our code follows a consistent coding style that all of us have adapted since milestone 2. We based our coding style off of one another, where the comments are in line with the code.
- MVC/OO patterns followed up
 - In terms of MVC, all pages use a MVC pattern where the user sees the original page, and then updates the page based on their inputs. After that, the new updated page is generated using the information from the database, and then is sent back to the user on the new page.
- Frameworks (back end front end) deployed correctly
 - The back end and front end are both deployed correctly to AWS and the application. They both work together when getting a page, and testing all functionality of the page.
- Database organization (tables, naming...)
 - Our database is well organized, where all tables that are needed are named similarly to their respective pages.
- Blobs being used? If so, is it working?

- Our project does not use any blobs.
- Adherence to best practices of security (PW encrypted, search inputs verified etc.)
 - Passwords are hashed and stored in the database instead of plain text
 - Input fields, such as fields in Registration and postItem are validated to ensure that the fields are not empty, and contain logical values if applicable (e.g. quantity value must be above one, a price shall not be negative, etc.)
- Efficiency (proper use of image thumbnails, efficient search etc.)
 - Images are stored as links within our database, and are posted on the website using the image link
 - Search throughout our project utilizes the SQL keyword LIKE 'keyword'
 - Searching for marketplace listings has an optional categories tag based on the product being listed
 - Searching for forum and service listings has an optional dropdown menu for searching with all (title and description), just title, or just description