

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	ktleong2000	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 2

Date: 07/06/2022

Version History

Milestone	Date
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] Prioritized Functional Requirements	10
[Section III] UI Mockups and Storyboards (High Level Only)	17
[Section IV] High Level Database Architecture Organization	27
[Section V] High Level APIs and Main Algorithms	41
[Section VI] High Level UML Diagrams	44
[Section VII] High Level Application Network and Deployment Diagrams	45
[Section VIII] Identify Actual Key Risks For Your Project At This Time	47
[Section IX] Project Management	50
[Section X] Detailed List of Contributions	51

[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] Prioritized 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 can search by college institution in the marketplace.
- 1.6. A general user can search by college institution in the service listing section.
- 1.7. A general user shall accept our terms of conditions.
- 1.8. A general user shall be able to submit their own ads to be promoted on our website.
- 1.9. 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 edit a forum post.
- 2.12. A registered user shall be able to edit a service listing.
- 2.13. A registered user shall be able to edit a sales listing.
- 2.14. A registered user shall be able to log out.
- 2.15. A registered user shall be able to message other registered users.
- 2.16. A registered user shall be able to pay with different methods, ie. Cash, debit, etc.
- 2.17. A registered user shall have setting options
- 2.18. A registered user shall be able to pay to be on the top of the forums.
- 2.19. A registered user shall be able to report another user.
- 2.20. A registered user shall be able to block another user.
- 2.21. A registered user shall be able to rate other general users.

3. Admin

- 3.1. An admin shall be able to review reports.
- 3.2. An admin shall be able to ban registered users.
- 3.3. An admin shall be able to unban registered users.
- 3.4. An admin shall be able to delete forum posts of other users.
- 3.5. An admin shall be able to remove service listings from the website.
- 3.6. An admin shall be able to remove marketplace listings from the website.

4. Service Listing

- 4.1. A service listing shall have a title.
- 4.2. A service listing shall have the location of where the service will be needed.
- 4.3. A service listing post shall have characters in its content body.
- 4.4. A service listing post shall be able to have images in its content body.
- 4.5. A service listing post shall have basic information like time posted.
- 4.6. A service listing post shall be able to have its content body edited.
- 4.7. A service listing post shall be able to have its title edited.
- 4.8. A service listing post shall be able to be unlisted.
- 4.9. A service listing post shall be able to have filters.
- 4.10. A service listing post shall have optional tags.

5. Receipt

- 5.1. A receipt shall have the product bought.
- 5.2. A receipt shall have the price of the product bought.
- 5.3. A receipt shall have the time of the transaction.
- 5.4. A receipt shall have the email of the user involved.
- 5.5. A receipt shall have the method of contacting the seller via email.
- 5.6. A receipt shall have the method of contacting the seller via phone number.

6. Shopping Cart

- 6.1. A shopping cart shall remember what has been stored within it.
- 6.2. A shopping cart shall remember how many of one entity have been stored within it.

- 6.3. A shopping cart shall allow users to edit products within it.
- 6.4. A shopping cart shall allow for check out.
- 6.5. A shopping cart shall give descriptions of products stored within it
- 7. Checkout
 - 7.1. Checkout shall allow a user to input their payment information.
 - 7.2. Checkout shall allow a user to input their shipping address.
 - 7.3. Checkout shall allow a user to input their billing address.
 - 7.4. Checkout shall allow users to review their purchase on last time.
- 8. Navigation Bar
 - 8.1. A navigation bar shall have a section for students selling their products.
 - 8.2. A navigation bar shall have a section for teachers to look for student assistants.
 - 8.3. A navigation bar shall have a section for the student forums.
 - 8.4. A navigation bar shall have a section for the about us page.
- 9. Marketplace Listing (Buy/Sell Listing)
 - 9.1. A marketplace listing shall have a title.
 - 9.2. A marketplace listing shall have a description.
 - 9.3. A marketplace listing shall have a photo (or many).
 - 9.4. A marketplace listing shall have a price.
 - 9.5. A marketplace listing shall be able to be unlisted.
 - 9.6. A marketplace listing shall have filters to categorize.
 - 9.7. A service listing shall have optional tags.

9.8. A service listing shall have the option to be returned or not.

10. Ratings

10.1. User ratings shall rate a profile on a scale of 1 to 5.

10.2. User ratings shall be visible on a registered user's profile.

10.3. User ratings shall be visible on a registered user's marketplace listing.

10.4. User ratings shall be visible on a registered user's service listing.

10.5. User ratings shall be able to be posted by registered users.

Priority 2:

1. General User

a. A general user shall be able to leave comments on products/services.

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.

3. Admin

a. An admin shall be able to view a chat log between two users.

b. An admin shall be able to do all that a registered user can.

c. An admin shall be able to see personal information of users.

- d. An admin shall be able to close group posts.
- e. An admin shall be able to re make posts.
- f. An admin shall be able to see IP's.

4. Receipt

- a. A receipt shall have a way to contact admin service support.

5. Shopping Cart

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

6. Navigation Bar

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

7. 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.

8. 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.

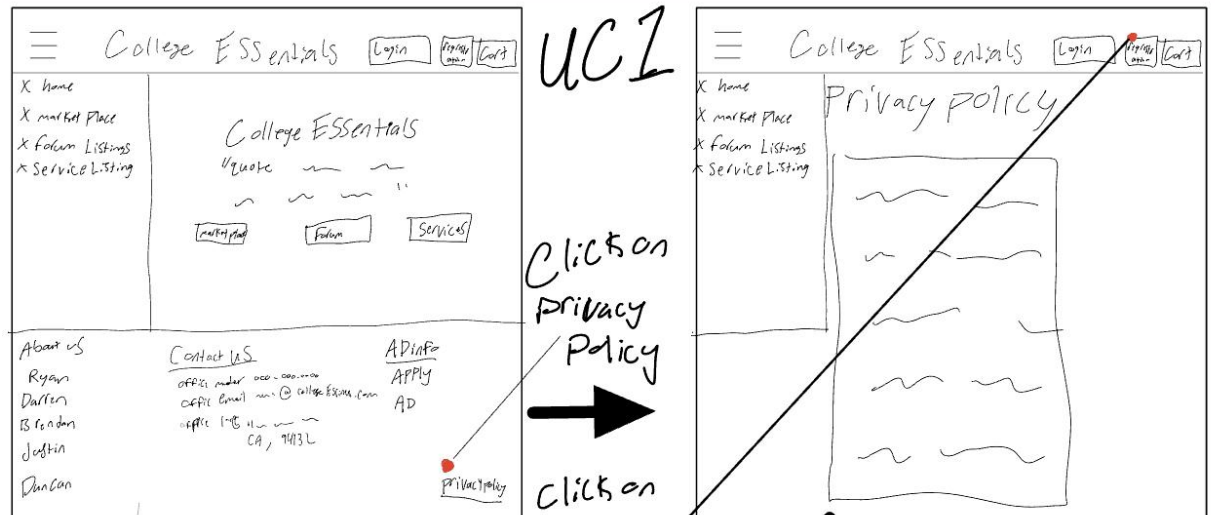
Priority 3:**1. Shopping Cart**

- 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.

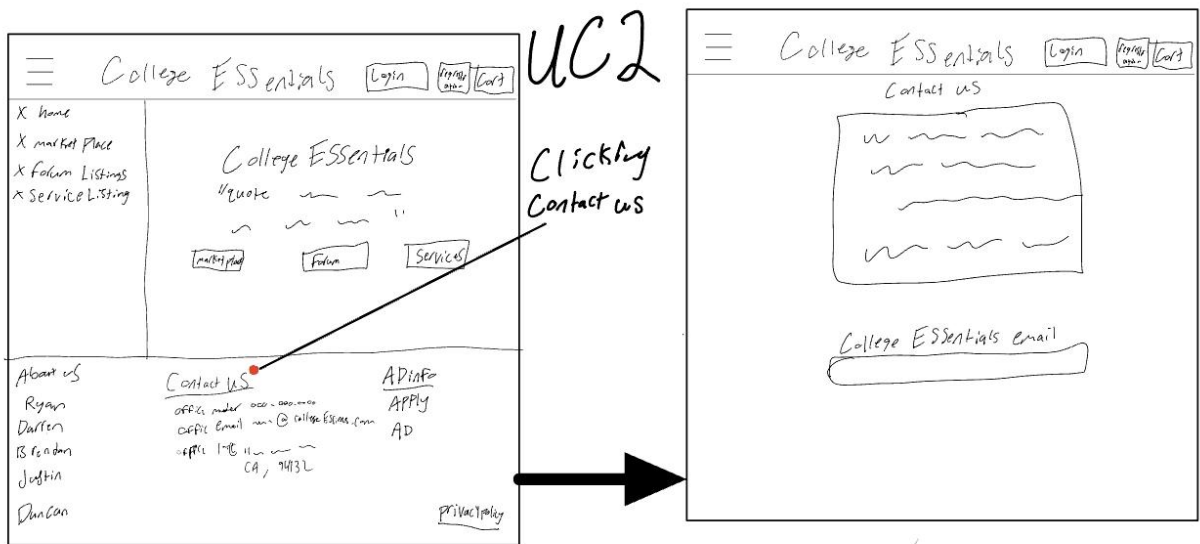
[Section III] UI Mockups and Storyboards (High Level Only)



Billy is a freshman in college. He is currently about to sign up for College Essentials But he would like to read the privacy policy

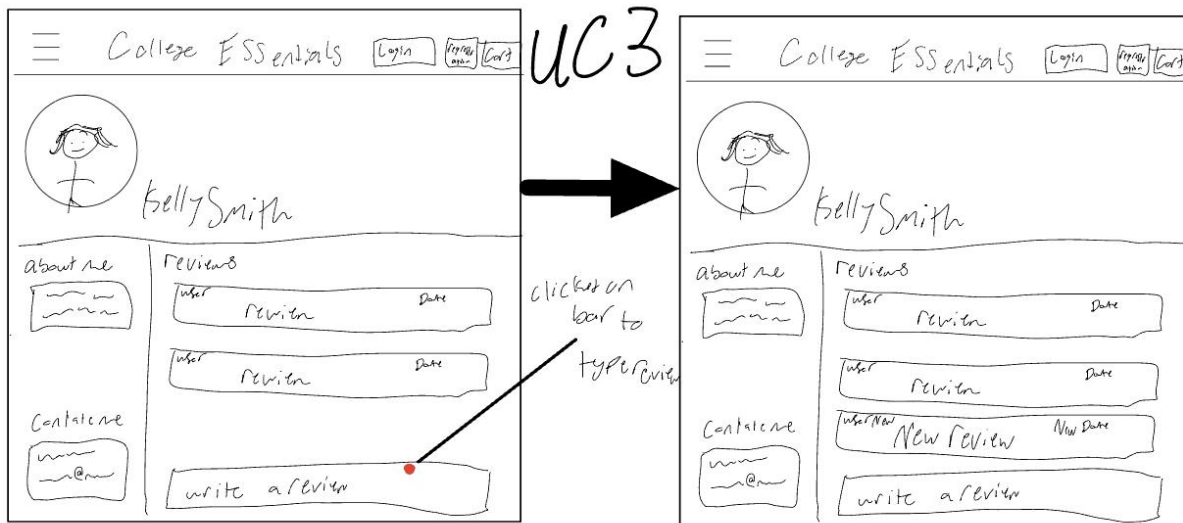
Sign up For College Essentials

First	Last
Username	Uni
email	
pass	pass
<input type="checkbox"/> Agree to TOS TOS	
create ACC	



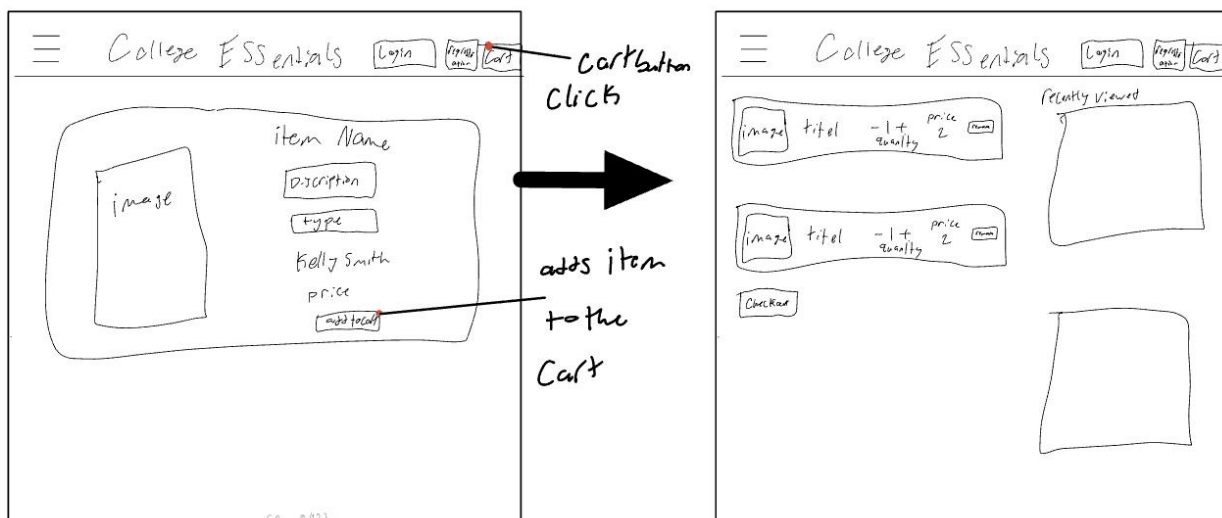
John is a freshman in college. He had been very satisfied with the website. And would love to contact the team at College Essentials he clicks on Contact Us

John can use this page to contact the team



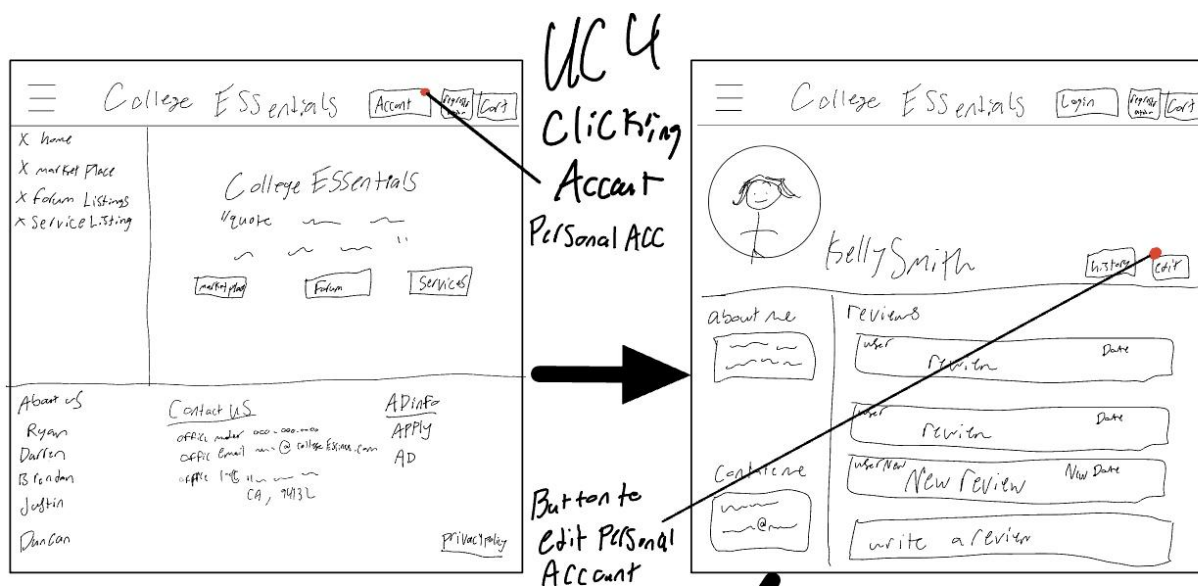
John is a freshman in college. He had been very satisfied with his textbook that he had just bought from another student, Kelly, from College Essentials. The textbook looked new inside, and out. There also didn't seem to be any graffiti or damage found within the book as well.

Since John was so happy with the condition of the textbook he had purchased, he decided to leave a review on Kelly's profile detailing the textbook's condition and how smoothly the transaction went.



A few days later, Jake sees another one of Kelly's textbooks up for sale in the marketplace. He is skeptical of Kelly at first, so he goes to Kelly's profile to see if there are any ratings. Much to Jake's delight, he finds that John loved his textbook that he purchased from Kelly a few days ago.

This review is enough for Jake to think that Kelly is credible, and he buys that textbook moments later.



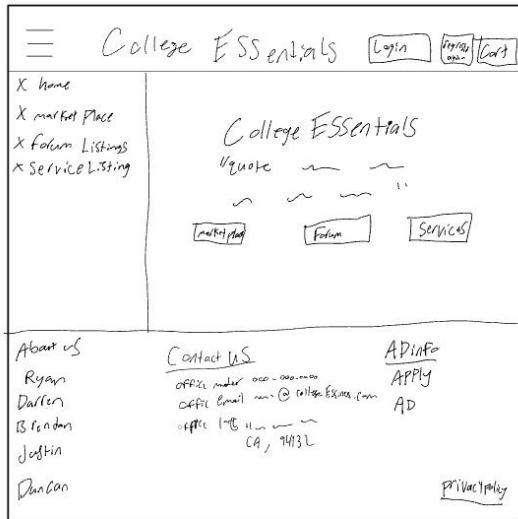
College Essentials

About me

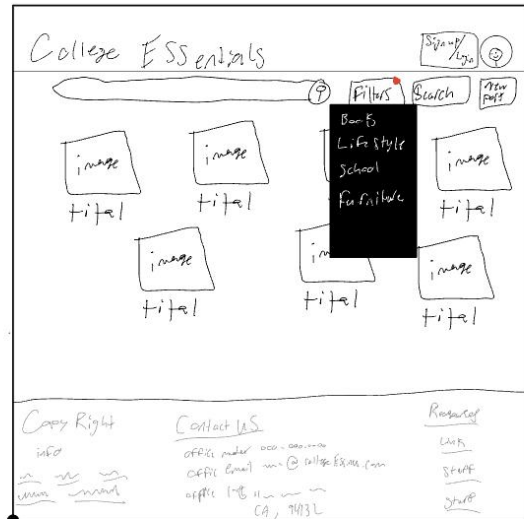
Contact info

College email

Submit

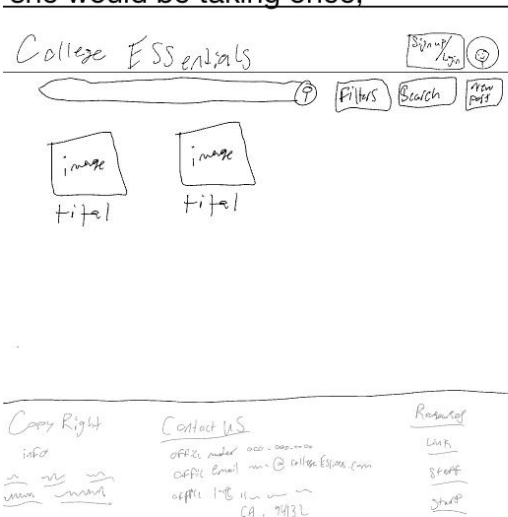


UC 5

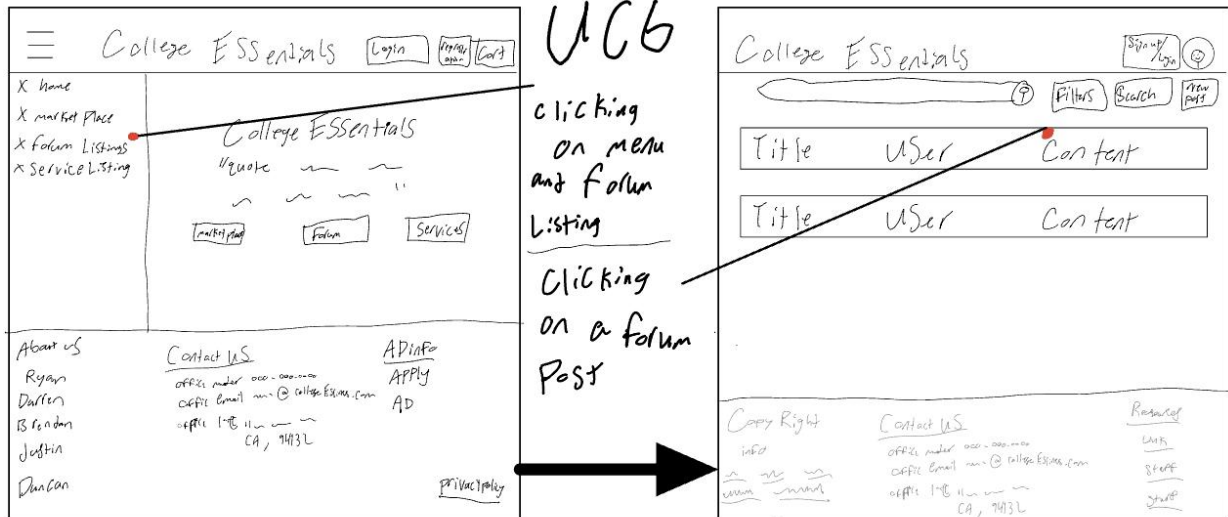


Jennifer needs to buy textbooks for her new classes, so she goes to her community college bookstore to look for the textbooks that she needs. Unfortunately, her community college doesn't offer any courses that use those textbooks. Jennifer doesn't want to buy a new condition textbook for a class that she would be taking once,

Jennifer decides to go to College Essentials's marketplace, where she is able to filter her search based on various universities as well as the condition of the textbook she needs.

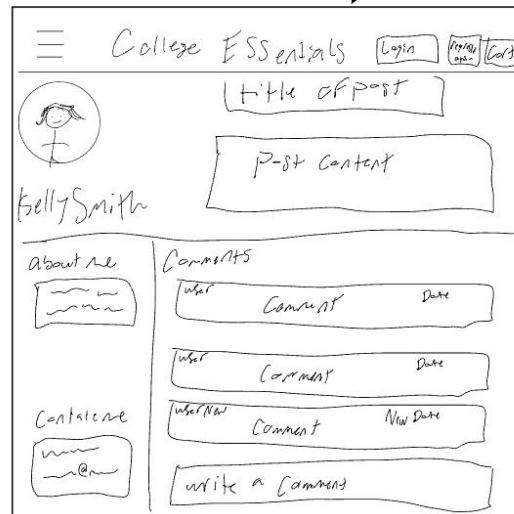


Because her local community college doesn't offer the textbooks she needs, our website would allow her to search different colleges in different cities to find someone who is selling those textbooks. If she wants to find the textbook from a closer city, she can search from schools that are closer to her. If she wants to find the textbook from her new school, she would be able to search from that school directly.

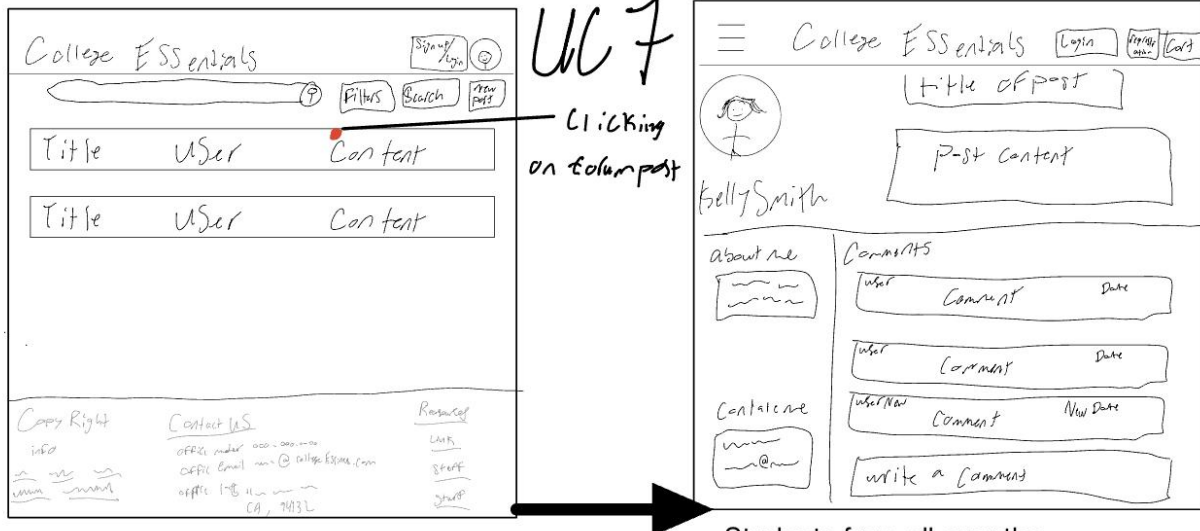


Ryan is looking for a possible opportunity to work under a professor at his college. Ryan has a poor experience with other websites, struggling to find opportunities for his university amongst the thousands of results that show for many other universities. However with College Essentials, he notices a menu that can easily return results from a specific institution.

He therefore enables the filter and gets new results that are all from his college that he is attending. He was able to find a post from one of his favorite professors and applied.



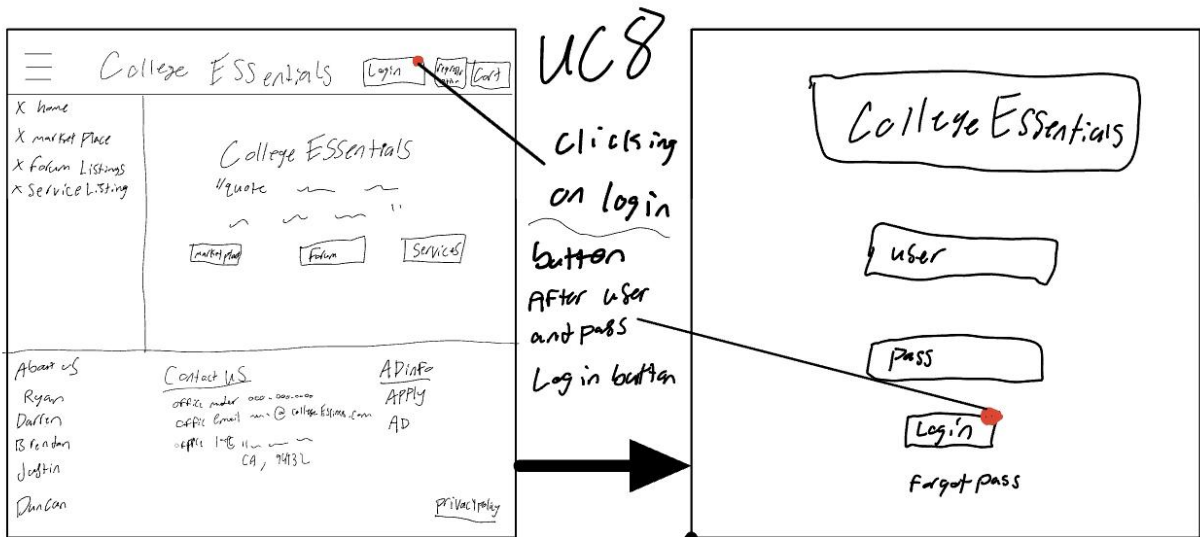
College Essentials will enable everyday users like Ryan to easily filter among many search results for services that are related to his university.



Dr. Ethan is a professor at San Francisco State University who teaches biology. He comes up with a brilliant idea and needs a few students to help him bring his idea to life. He decides to go on College Essentials and posts a job offer for students who are willing to help him with his experiments.

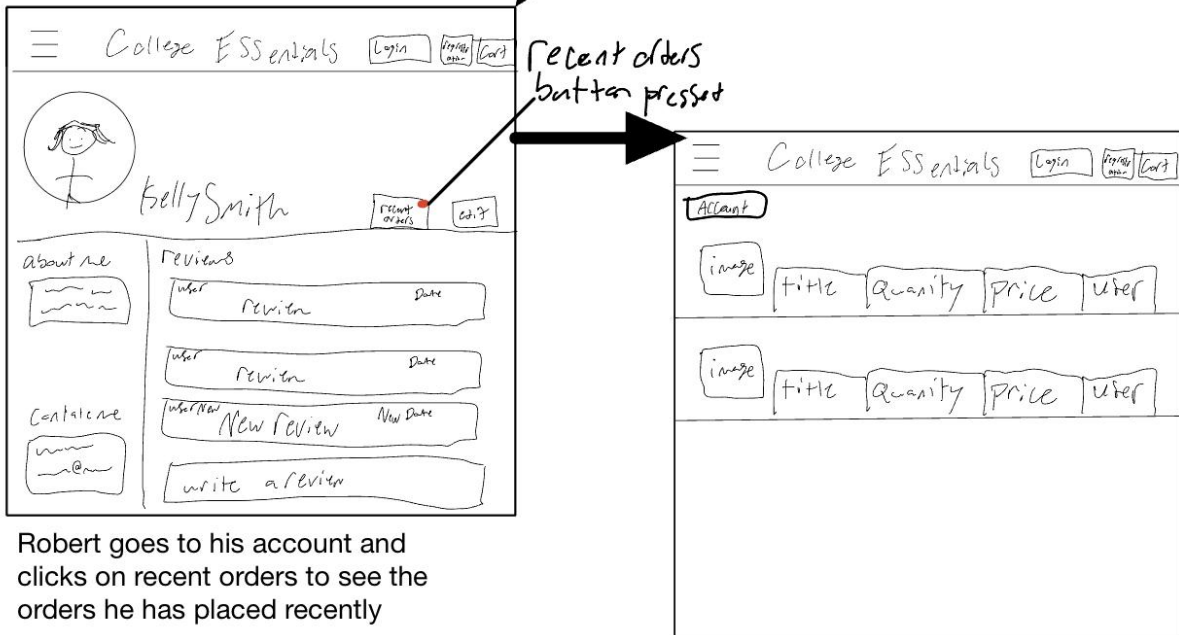
Students from all over the college reply to his advertisement for seeking help and he is able to choose the best qualified student to hire.

College Essentials will make it easier for Dr. Ethan to seek help for his upcoming research project. For any professor like Dr. Ethan, finding time out of his day away from holding lectures and grading work is already difficult as is.

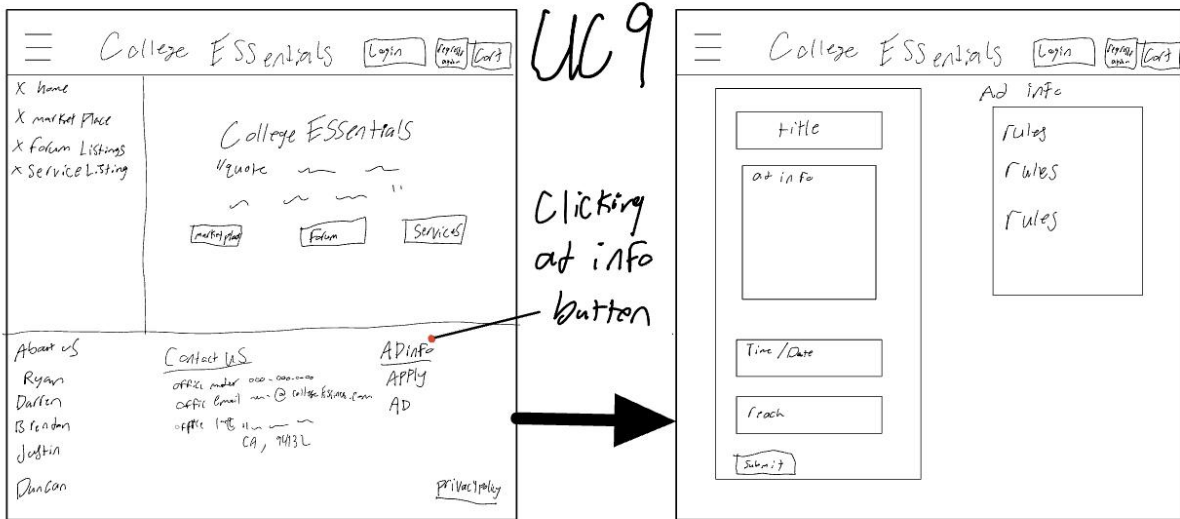


Robert is loving College Essentials and has bought a couple of items from the website,

Robert is logging in to his account to check on his recent orders

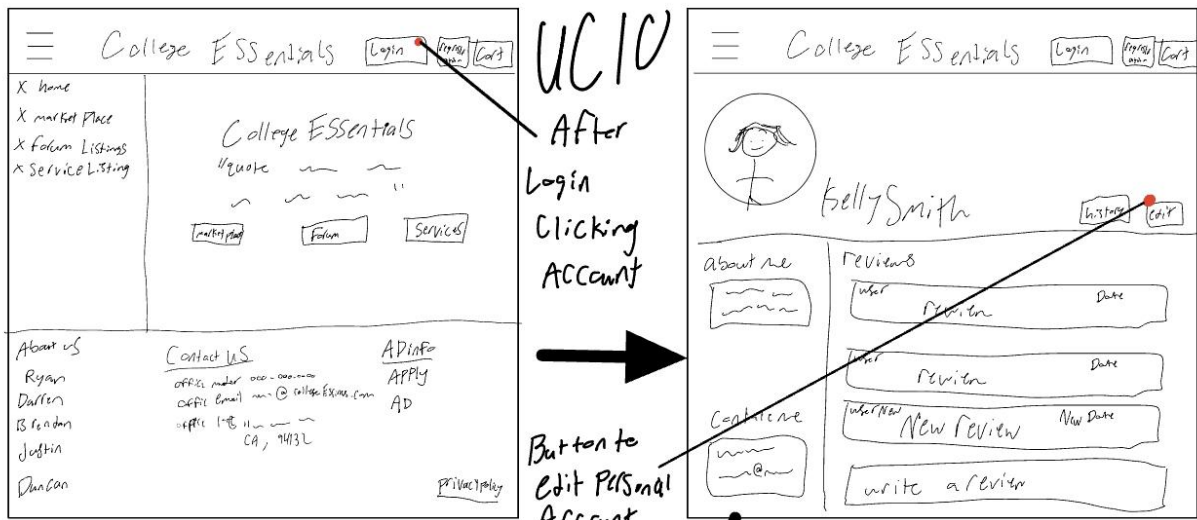


Robert goes to his account and clicks on recent orders to see the orders he has placed recently



The user bill wants to purchase an Student Ad on the College Essentials website in order to promote his new YouTube video he made on his college

Bill clicks on our sib-link in order to purchase an Ad, adds all his ad information



The user Greg is transferring colleges and wants to change his account from SFSU to UCLA he clicks on log in after that he would click account

On his account setting page he clicks on the button edit account after that takes him to the edit page

The College Email is the way we verify the College

[Section IV] High Level Database Architecture Organization

1. Business Rules

a. GeneralUser

- i. A GeneralUser shall be able to log into a registered account (RegisteredUser)
- ii. A GeneralUser shall be a buyer.

b. RegisteredUser

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

c. Rating

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

d. Buyer

- i. A Buyer shall be considered a GeneralUser or RegisteredUser.
- ii. A Buyer shall have at most one ShoppingCart at a time
- iii. A Buyer shall have ordered zero or many Products.
- iv. A Buyer shall have zero or many Payment methods.
- v. A Buyer shall receive a Receipt for their transaction.
- vi. A Buyer is linked to many Transactions

e. Seller

- i. A seller shall be considered a RegisteredUser
- ii. A seller shall have posted at least one MarketplaceListing
- iii. A seller shall receive many Receipts for their transactions.
- iv. A seller is linked to many Transactions

f. Admin

- i. An Admin shall be considered a RegisteredUser.

g. College

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

h. Marketplace Listing

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

i. ServiceListing

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

j. ServiceReply

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

k. 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

l. ShoppingCart

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

m. Payment

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

n. Receipt

- i. A Receipt shall be given to one Buyer.
- ii. A Receipt shall be given to one Seller.
- iii. A Receipt shall belong to one Transaction.

o. Product

- i. A Product is ordered by zero or many Buyers
- ii. A Product is added to many ShoppingCarts
- iii. A Product is listed on a MarketplaceListing

p. ForumPost

- i. A ForumPost is posted by one RegisteredUser
- ii. A ForumPost shall be followed by many ForumReplies

q. 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.

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

d. Buyer (Weak)

- i. buyer_id: key, numeric; This is the unique buyer id used in a transaction.
- ii. general_user_id: key, numeric; This is the unique GeneralUser id associated with a buyer
- iii. registered_user_id: key, numeric; This is the optional unique RegisteredUser id associated with a buyer

e. Seller (Weak)

- i. seller_id: key, numeric; This is the unique seller id used in a transaction.
- ii. registered_user_id: key, numeric; This is the unique RegisteredUser id associated with a buyer

f. 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

g. 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

h. 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

i. 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.

j. 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

k. 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 in the transaction
- v. buyer_id: This is the unique id associated with the buyer in the transaction

l. 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

m. 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.

n. 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

o. Product (Strong)

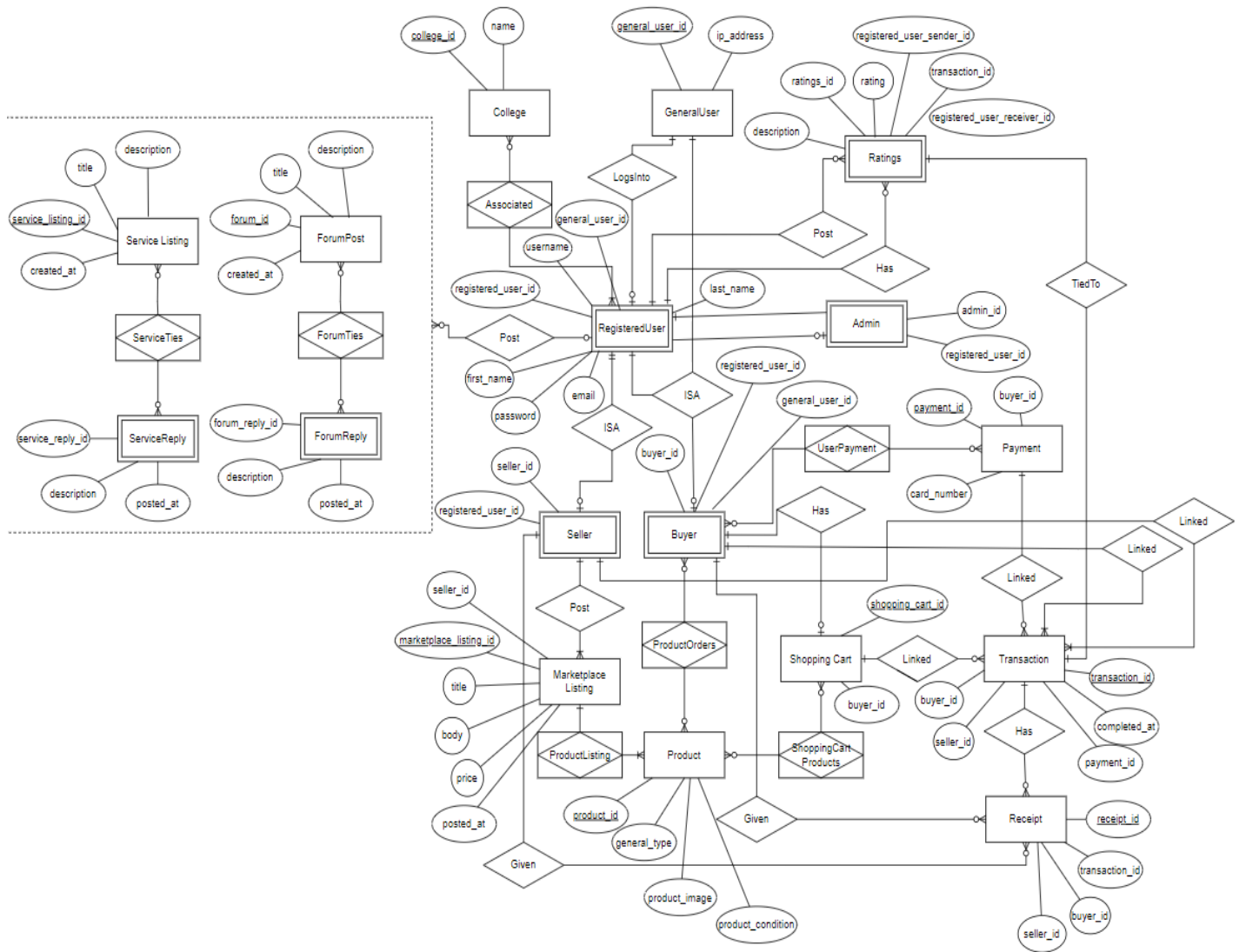
- i. product_id: key, numeric; This is the unique id of the product
- ii. general_type: alphanumeric; This is the general type of the product (e.g. a textbook, bicycle, furniture, etc.)
- iii. picture: alphanumeric; This the picture link associated with the product
- iv. Condition: alphanumeric; This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

p. ForumPost (Strong)

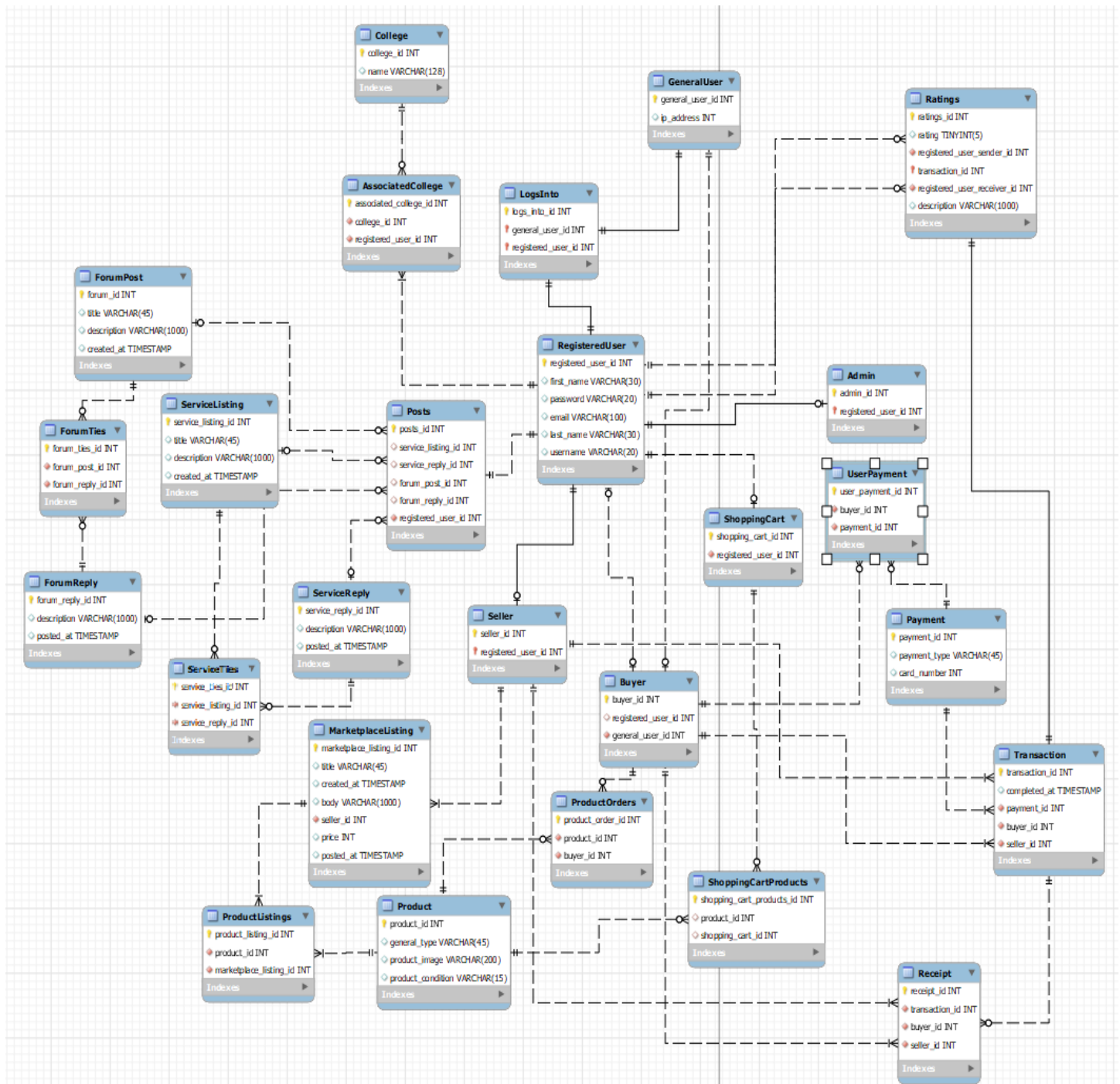
- i. forum_id: key, numeric; This is the unique id of a ForumPost

- 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
- q. 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 APIs and Main Algorithms

1. Authentication API

a. Registration

- Post Request: When the user wants to register for the website, the backend will receive a post request that will contain user information such as username, password, and email. The backend will check if the user information is valid, and then send a success message. If the information is not valid, the backend will send a failure message.

b. Login

- Post Request: When the user wants to login, the backend will receive a post request that will contain the user's information such as username, password, and email. The backend will check the database if the user information is valid and then send a success message. If the information is not valid, the backend will send a failure message and request that user information is incorrect or does not exist.

2. Buying / Selling API (Marketplace/Service Listings):

a. Selling Post Request

- When the user wants to sell a product, the backend will receive a post request with all the required information for the product. The backend will save that information in the database and return a success.

b. Buy Get Request

- When the user wants to buy a product, the backend will receive a get request with all the required information for the product. The backend will then return that information in a hashmap to be saved in the shopping cart.

3. **Forum API:**

a. Forum Post Request

- When the user wants to post a forum post, the backend will receive a post request with the necessary information for the forum post. The backend will save that information to the database and return a success message.

4. **Search Bar API:**

a. Search Get Request

- When the user wants to search for a marketplace item or service listing, the backend will receive a get request that contains all of the search filters requested by the user. The backend will then check the database for the information that the user is requesting, and then return that information and a success to the user if possible, otherwise will return a failure.

5. **User Ratings API**

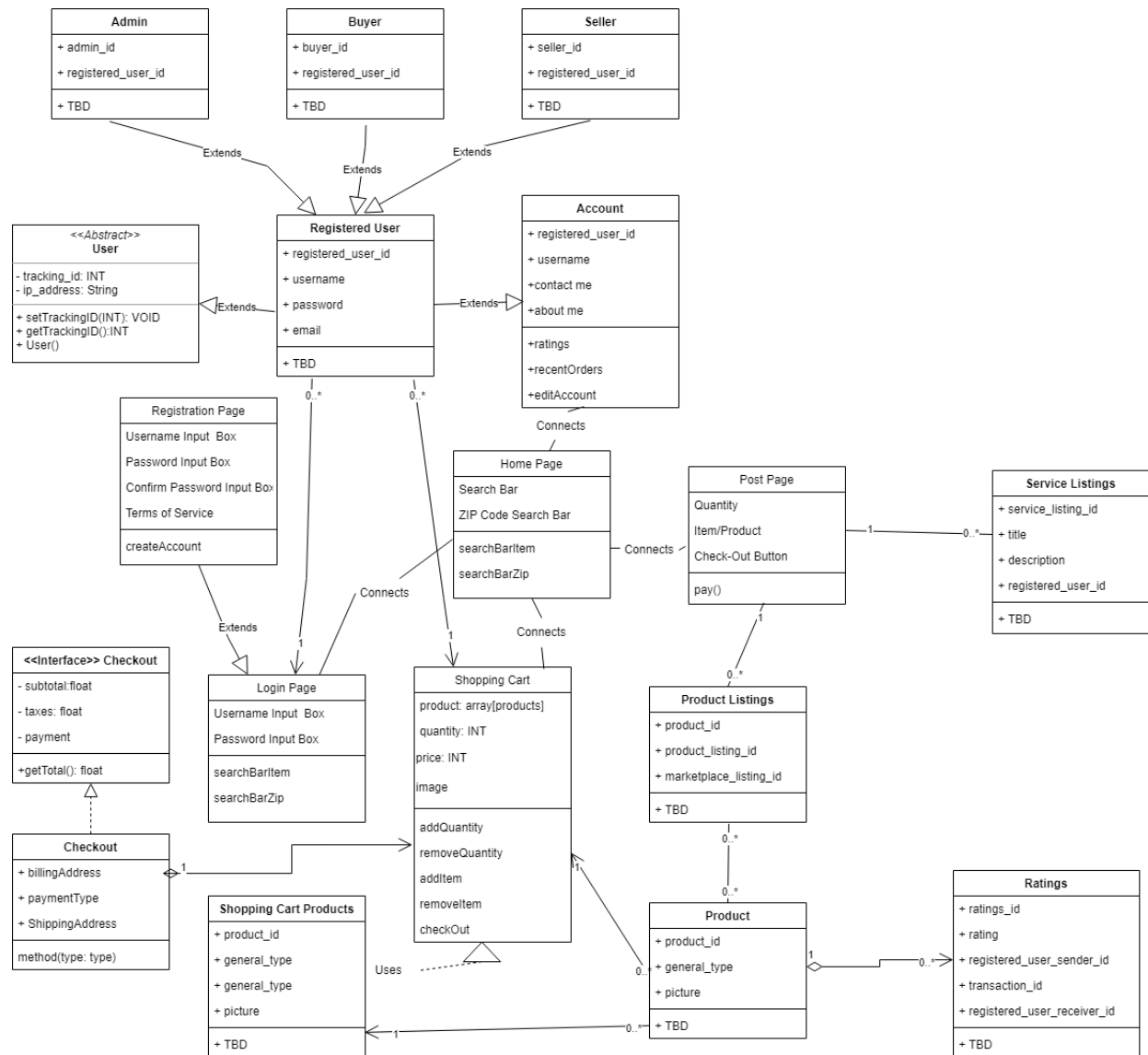
a. Post Request

- When the user wants to post a user rating on another user's profile, the backend will receive a post request that will receive the rating and other details of the post. The backend will then save in the database and send a success message.

b. Get Request

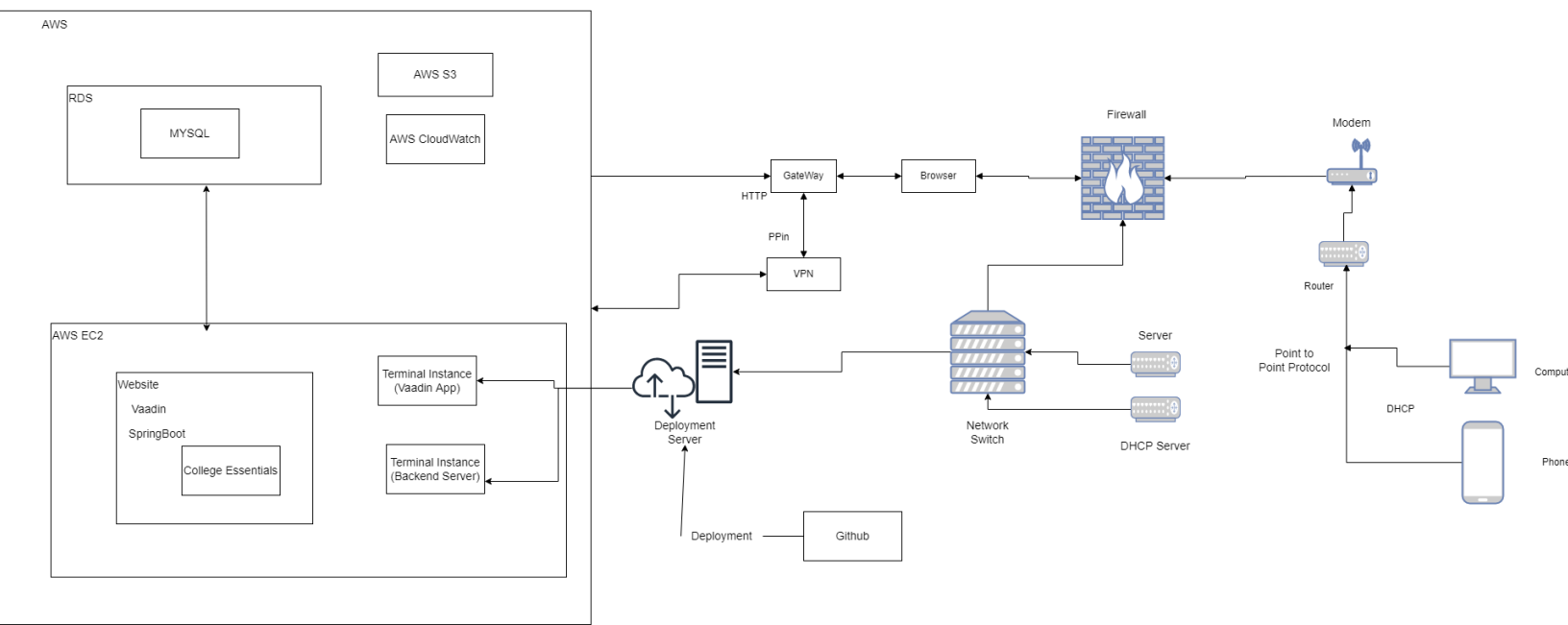
- When the user wants to view a user's profile, the backend will automatically receive a get request for the user's ratings, and then display on the user's profile page.

[Section VI] High Level UML Diagrams

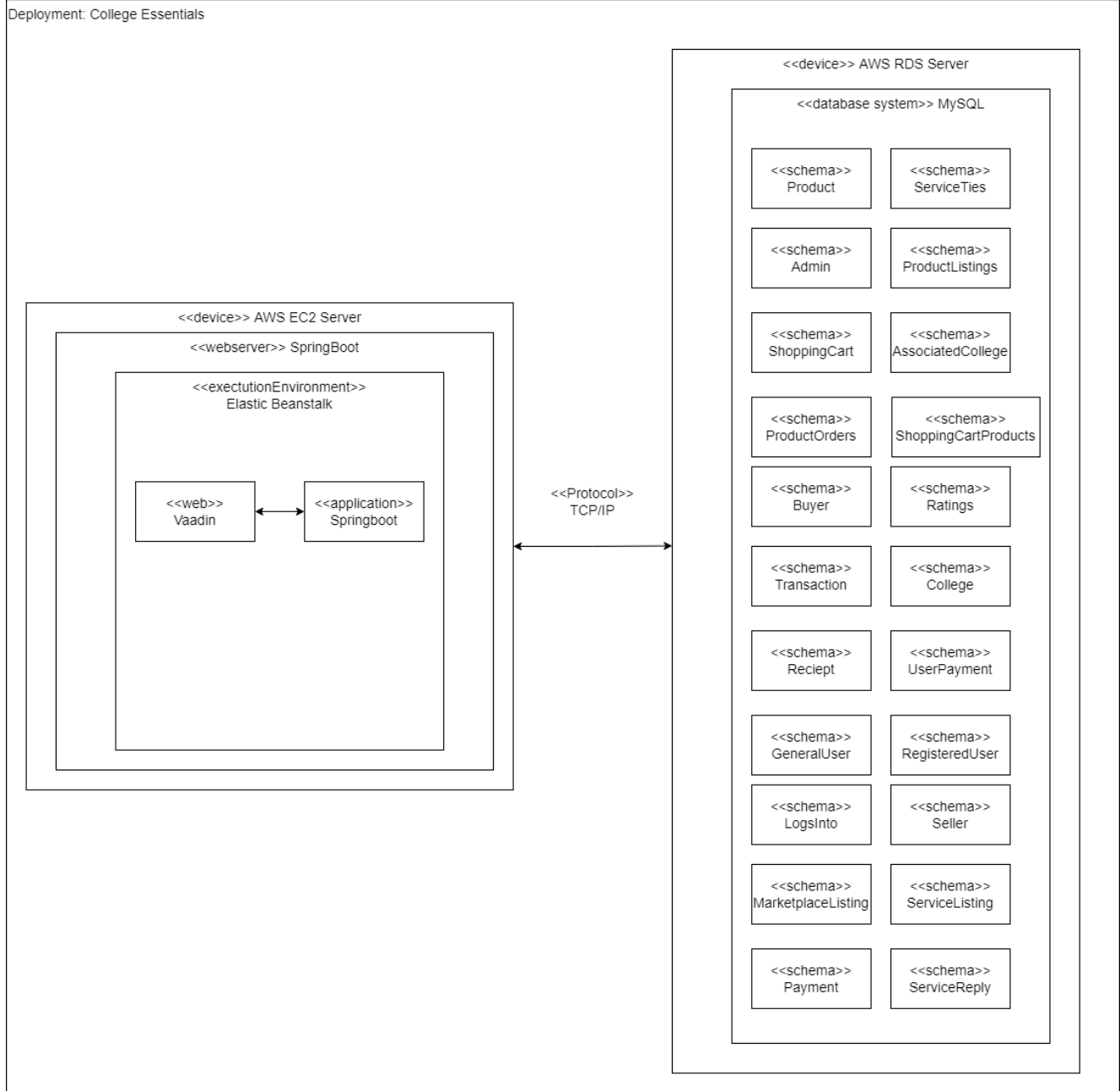


[Section VII] High Level Application Network and Deployment
Diagrams

Network Diagram



Deployment Diagram



[Section VIII] Identify Actual Key Risks For Your Project At This Time

1. Skills Risks

The main areas that we are whether or not our team are all on the same skill level when it comes to working on the code. We have two teams; one for front-end and one for back-end, and the risks that come with it are whether each member is on the same page.

To mitigate these risks, we are making sure that we all learn how to work with Vaadin and that all our team members are working with the same tech stack. We are making sure that we learn from each other to help the teammates that might have more skill risks than others.

2. Schedule Risks

There are two main problems that we face with schedule risks. The first main problem is whether the time schedules of each individual member lines up with one another. The second problem is whether or not we are able to allocate enough time to do the project.

To address these problems, we have shared our schedules with each other in order to determine the best times to meet as a group. We are working together to keep each other responsible for the work that needs to be done. We are also using Trello to keep track of which tasks need to be done, and when we should have it done by.

3. Technical Risks

The main areas that we are technical skill risks are related to the deployment of our app. The majority of our team has no experience with Vaadin,

which is the development platform we are using to build our web app. Our other points of risks are in regards to setting up the database, and our understanding of networking.

To address these issues, our team is putting in the effort to research each area on our own, while working together to teach each other how each section should be done. We continue to work as a group to learn how to use Vaadin properly, and will continue our research on how to set up the database and broaden our knowledge of networking.

4. Teamwork Risks

The only major risks that can be considered teamwork risks are making sure that each member of the team is putting in enough work. We all have our strengths and weaknesses, the only risk is whether each member is held accountable for the work that they should be doing.

Our solution to this is to have all the members hold each other accountable for the work that they should be doing. Our team shouldn't be afraid to make sure that we are all doing the work, and that we are comfortable telling each other if they have a task that they need to do.

5. Legal/Content Risks

Some of the Legal Risks that we face would be the risks of users abusing our website. The two major problems related to this would be suspicious service listings, and scamming other users.

To mitigate these risks, we have a moderation system where users would be able to report listings as suspicious, and then moderators would be able to delete

these suspicious listings. We also have a rating system that would allow people to make sure that the people they are buying from are safe.

[Section IX] Project Management

As a team, we started using Trello in order to determine which tasks we needed to get done, and how we wanted to manage our time when doing so. We have trello split our tasks into three sections: TODO, in progress, and done. Along with this, we consistently met on Discord to discuss which sections we needed to work on, as well as how we wanted to split the work. Initially, we would all take a look at each task, attempt working on it as a group, and then split individually to finish up every task. After that, we would regroup and add our completed tasks Discord and Github, and then as a group we would go over final checks before fully submitting the task as completed. Then we would update Trello and continue on working with the other tasks. We used Discord to maintain communication, giving continuous updates on our own tasks and helping each other in areas that we needed help in. We shall continue to use Discord as our main way of communicating along with consistent meetings, and using Trello as our main checklist for which tasks we still need to complete.

[Section X] Detailed List of Contributions

I. Ryan

- A. Facilitated group meetings and plans
- B. Contributed to the initial network diagram.
- C. Researched and documented the High Level APIs
- D. Documented the list of contributions for each team member.
- E. Worked on frontend for the vertical prototype.
 - 1. Worked on the CSS for the login, register page, and home view.
 - 2. Designed the file system that the project will be using throughout the web application.
 - 3. Troubleshooted the remaining problems that occurred in the backend code.

II. Duncan

- A. Research and created all the UI Mockups needed for the use cases.
- B. Created and organized some prioritized functions needed.
- C. Worked on the frontend for the vertical prototype.
 - 1. Worked on some CSS for the home view page.

III. Justin

- A. Wrote the data definitions needed for each data group.
- B. Created and organized some prioritized functions needed.
- C. Worked on the business rules, search/filter function and media storage for the web application.
- D. Helped edit the high level UML diagrams.
- E. Created and edited the network and deployment diagrams.

- F. Discussed and documented the key risks and project management sections of Milestone 2.
- G. Edited the Milestone 2 document to be accurate and consistent.
- H. Worked on the deployment of the webapp to AWS and some backend code needed for the deployment.

IV. Brendan

- A. Wrote the data definitions needed for each data group.
- B. Created and organized some prioritized functions needed.
- C. Created the first draft for the high level UML diagram.
- D. Created and documented the guidelines for the UML diagram.

V. Darren

- A. Created and organized some prioritized functions needed.
- B. Worked on the business rules needed for the database.
- C. Created the ERD and EER diagrams for the database.
- D. Worked on creating the database and connecting the database to the website.
- E. Edited the Milestone 2 document to be accurate and consistent.
- F. Worked on the backend for the prototype.
 - 1. Connected the database to the website, including the backend needed for the diagram.
 - 2. Troubleshoot the remaining problems that occurred in the backend code.