Software Engineering CSC 648 Section 4

Milestone 2

FraGrant

Team 3 - Debuggers

Ruqaiyah Angeles - Team Lead

Aisha Asif - Scrum Master

Chris Solo - Back End Lead

Shahriz Malek - Front End Lead

Mandy Noto - GitHub Master

Ahmed Ballal - Universal Helper

10.17.22

Revision ID	Revision Date	Revised By

Table of Contents

Data Definition V2

Primary Data Name	**Sub Data**
Users	➔ Student ID ➔ Password ➔ Email ➔ Age ➔ Grade Level ➔ Name ➔ Username
Orders	➔ Order Number ➔ Order date and time ➔ Number of items ordered ➔ User email address ➔ User Phone number ➔ User physical address. ➔ User special instructions. ➔ User Name ➔ Payment-Method
Products	➔ Product ID ➔ Product Description ➔ Product type ➔ Product Date ➔ Product Time ➔ Product Transaction ID ➔ Product Image max size ➔ Product image list formats ➔ Product max

Functional Requirements V2

GREEN = Priority 3 (Low Priority)
YELLOW = PRIORITY 2 (Medium Priority)
RED = PRIORITY 1 (High Priority)

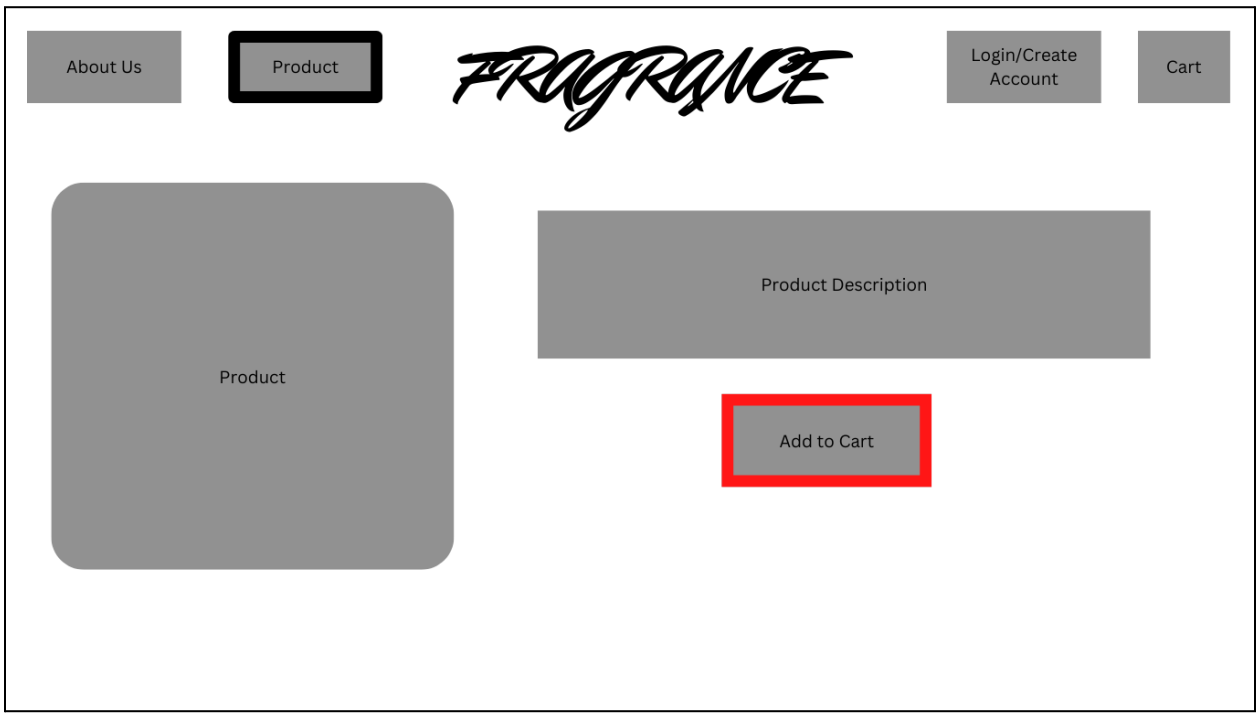
1. Users should be allowed to add to cart as a guest. (Priority 3)
2. Users should be allowed to browse the site as a guest. (Priority 1)
3. Users should not be allowed to checkout without signin/signout. (Priority 1)
4. Paypal or Stripe shall be supported as a payment method. (Priority 2)
5. Customers are allowed to customize their item. (Priority 3)
6. Users should be able to view the product. (Priority 1)
7. Users should be able to add products to the cart.(Priority 1)
8. Users should be able to remove products from the cart. (Priority 1)
9. Users should be able to view their order. (Priority 1)
10. Users should be able to delete or cancel their order. (Priority 1)
11. Users should be able to edit their profile. (Priority 3)
12. Users should be allowed to purchase products. (Priority 1)
13. Users should be able to check the history of the products they were viewing? (Priority 3)
14. Edit profile - Users should access and add to their profile. (Priority 2)
15. Users should be able to terminate their account. (Priority 3)
16. Users should be allowed a wish list. (Priority 2)
17. Users should be allowed to access past orders. (Priority 2)
18. Users should be allowed to access and manage their payment options. (Priority 1)
19. Users should be allowed to view all transactions. (Priority 1)
20. Users should be allowed to manage their address. (Priority 1)
21. Users should be allowed to increase the quantity of a product in their carts. (Priority 2)
22. Users should be allowed to compare different products. (Priority 1)
23. Users should be allowed to save a product for later from the cart. (Priority 2)
24. Users should be allowed to sign out. (Priority 1)
25. Users should be allowed to browse best sellers. (Priority 3)
26. Users should be allowed to browse new releases. (Priority 3)
27. Users should be allowed to buy again. (Priority 1)
28. Users should be able to track their order. (Priority 1)
29. Purchase - Every purchase should have an order number. (Priority 1)
30. Users should be able to create their own product. (Priority 3)
31. Users should be allowed to receive a random product. (Priority 3)
32. Website should work on other devices. (Priority 1)

UI Mockups and StoryBoards (High level only)

Mockups and Storyboards

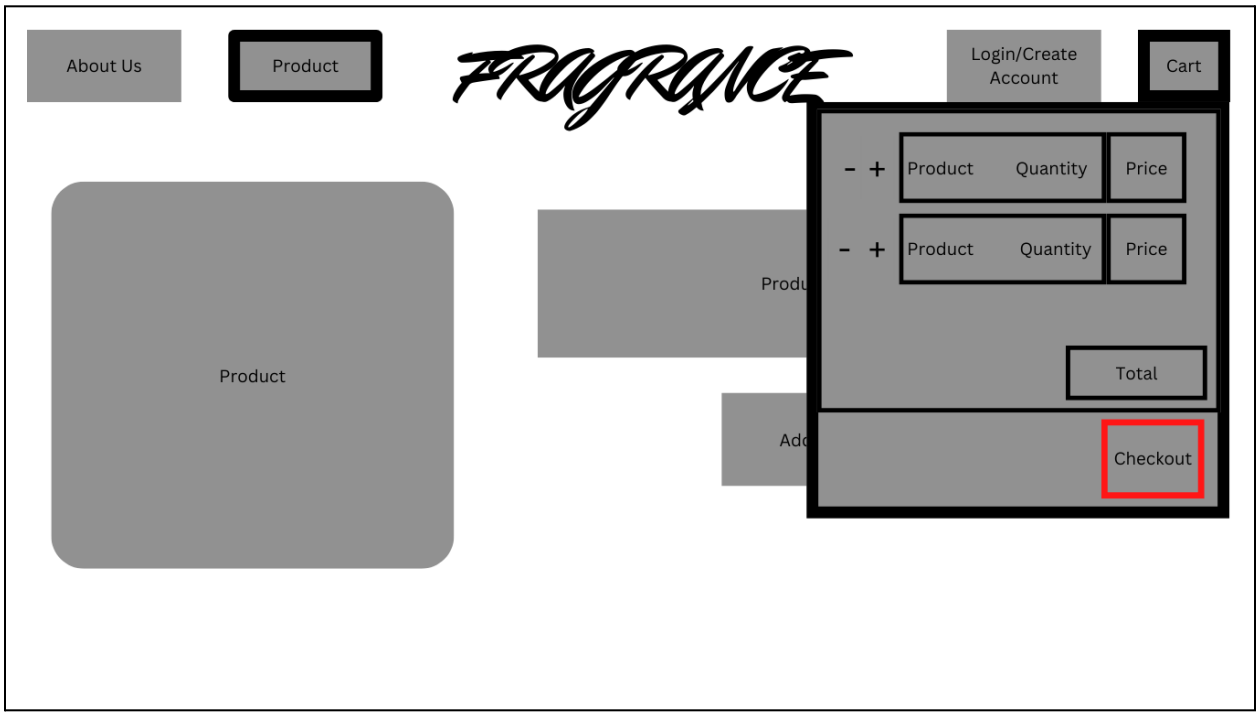
Users should be allowed to browse the site as a guest. Notice that the user is not logged in and can still view the products page and have access to other pages on the website.

Users should be able to add products to the cart: Notice that the user is not logged in, yet can still add items to the cart which will show up in the cart.
Users should be able to view the product: The user of this website is able to view the product.

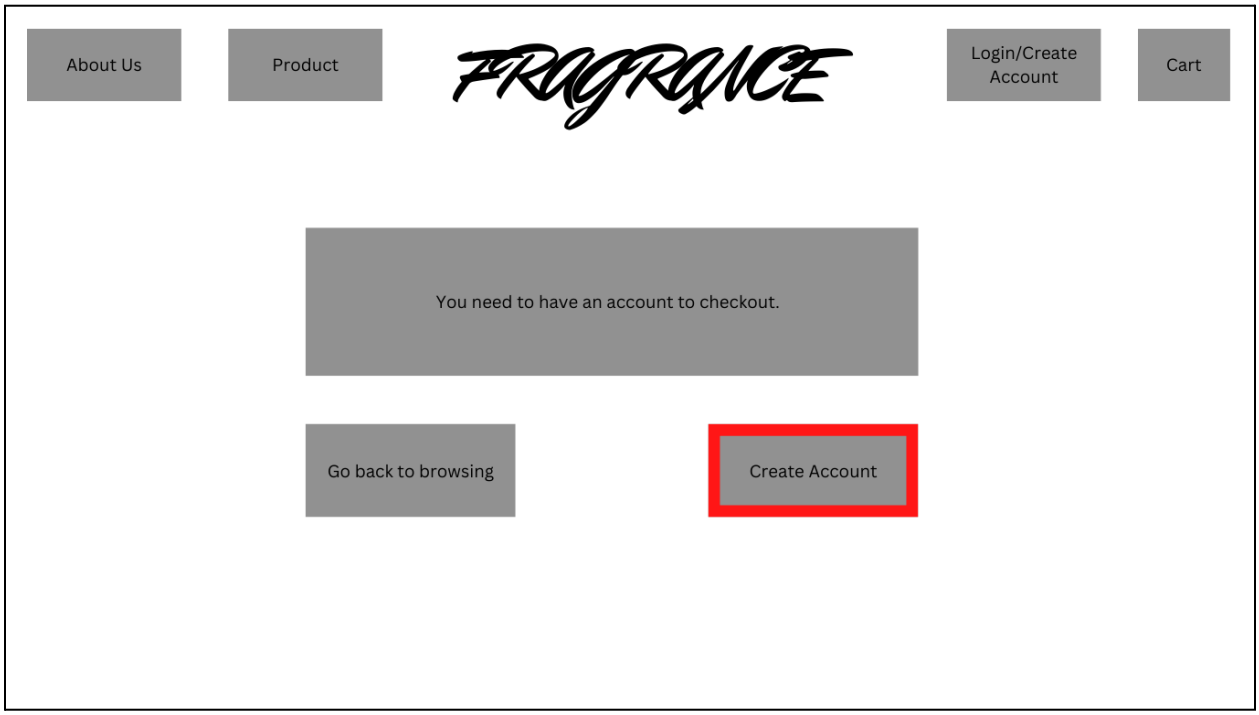


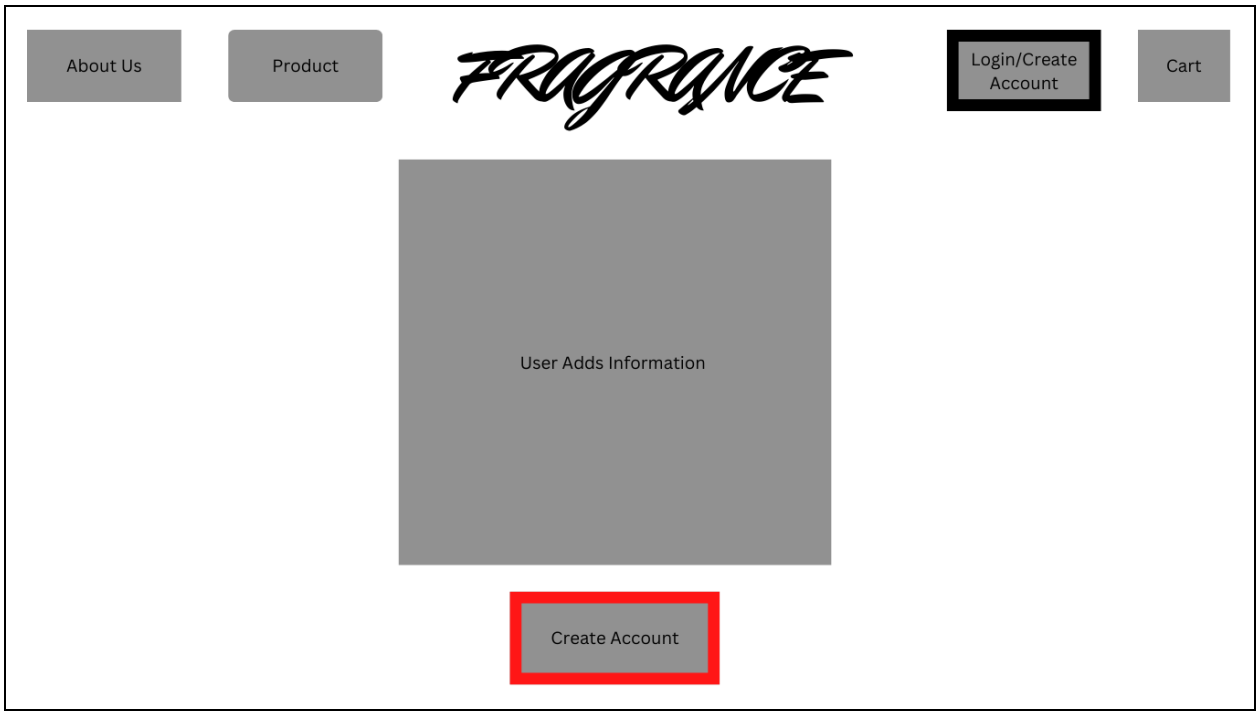
Users should be able to remove products from the cart: Users can add or remove products

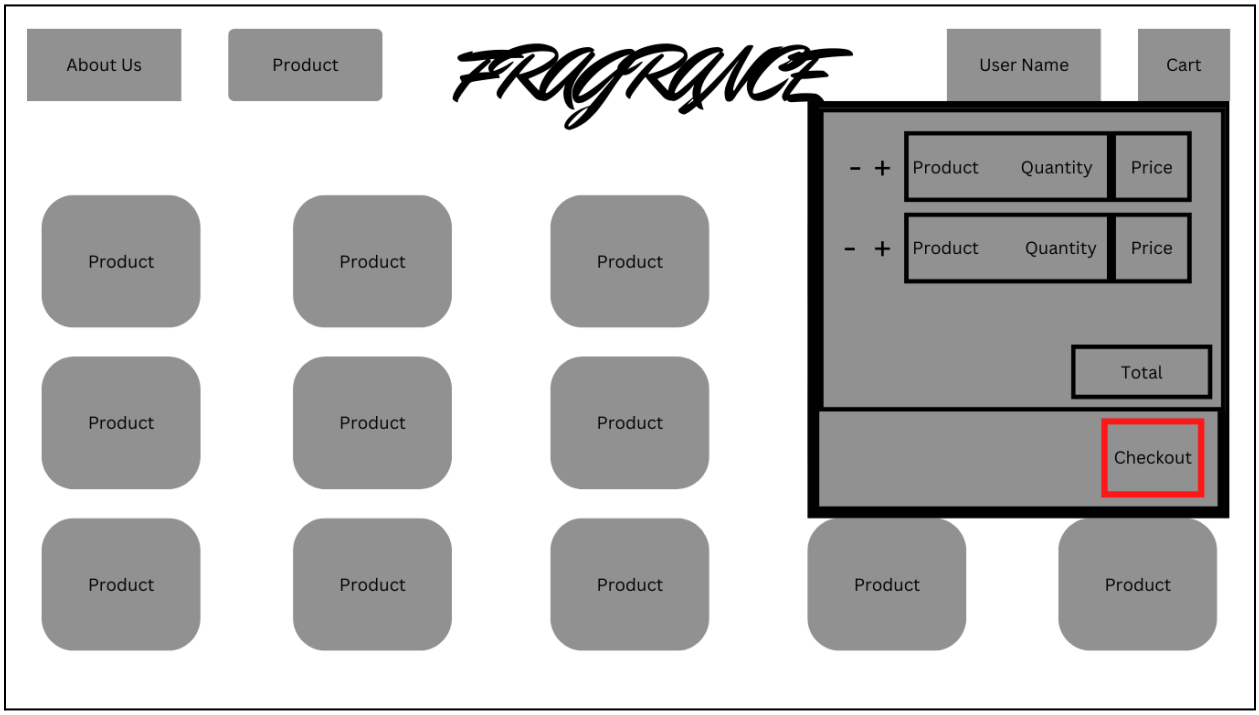
from the cart by using the + and - buttons in the cart.



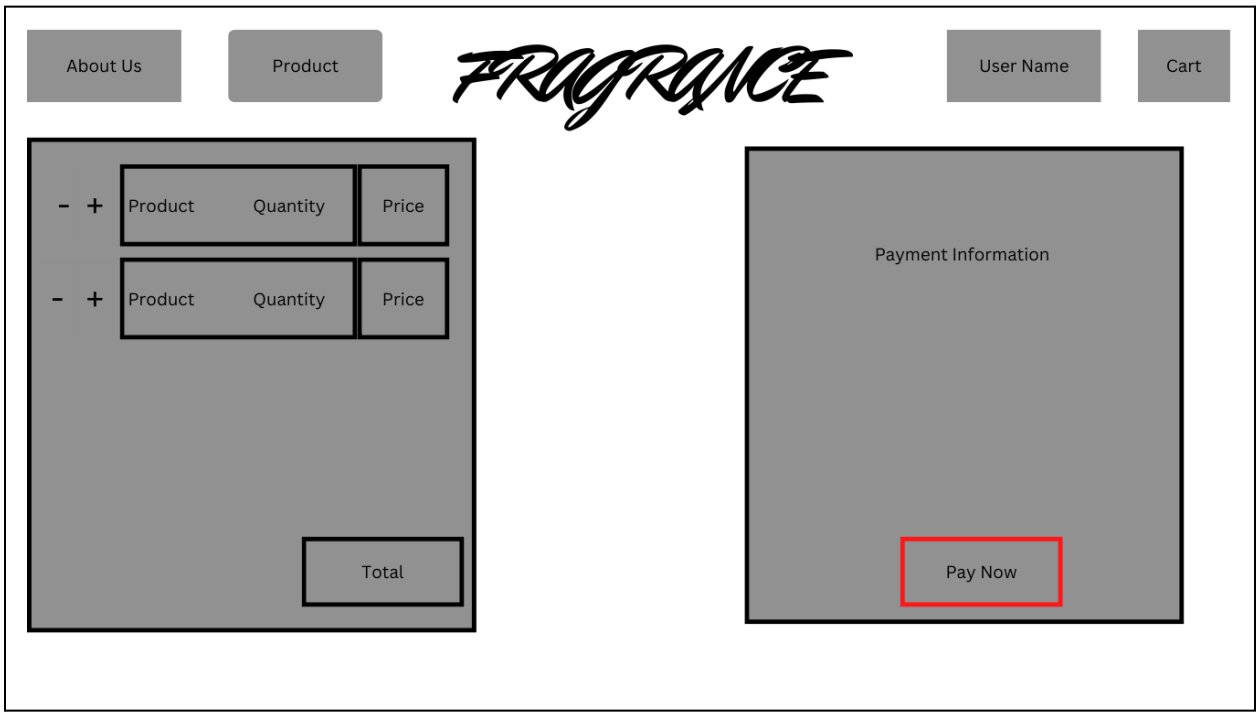
Users should not be allowed to checkout without signing in: User does not have an account and wants to checkout items. They are prompted to create an account in order to checkout or they can go back to browsing products.




About Us
Product
FRAGRANCE
Login/Create Account
Cart
User Adds Information
Create Account


About Us
Product
FRAGRANCE
User Name
Cart
- +
Product
Quantity
Price
- +
Product
Quantity
Price
Total
Checkout
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product
Product

Users should be able to view their order: Users are allowed to view their order before they checkout and make any edits to it that are necessary.



UX Validation Meeting:

During our UX validation meeting a lot was discussed. One of the main points was what we wanted the feel of our website to be. We decided on a very simple and elegant design that would add to the high-class persona we want for our product. Since our product is a perfume it was decided that a short description of each perfume be added. We decided to make the buttons very simple. The mockups on the storyboard are just bare bones of what we are actually going for. One major thing we changed was that we took out a lot of useless buttons. It was quite excessive to have so many buttons- so we just took all of them out. Another thing we discussed during the UX validation meeting was that we wanted our user to be able to add and delete things directly from the cart drop-down menu rather than being redirected to an entire page to edit their cart. We also wanted users to be able to edit the cart when they were about to checkout. In this meeting we also decided that a home page might not be necessary, because oftentimes people skip past that anyway and they truly care about the product.

High Level Architecture, Database Organization

Users

Users
student_ID int PK
student_age tinyint
student_grade_level varchar(50)
password_hash varchar(255)
student_email varchar(50)
student_username varchar(20)

Orders

Orders
order_number int PK
order_date_time DATETIME(0)
number_of_items int
student_ID FK -> user.student_ID
user_phone varchar(14)
physical_address varchar(100)
special_instructions varchar(255)
user_name varchar(50)

Products

Products
product_ID int PK
product_description varchar(255)
product_type varchar(50)

product_date DATE
transaction_ID int
products_available int

Add/Delete/Search Architecture	Functional Requirement
Add/Delete/Search for Users	For users that want to register or terminate their account -also for signing in/out
Add/Delete/Display orders	For registered users who want to cancel/modify orders
Add/Delete/Display user profile	For registered users editing their profiles
Add/Delete/Display for products	For registered users who want to edit their shopping cart, edit a product, or check the history of products they're viewing -guests have the same functionality
Search/Display products	For registered users or guests viewing our store -all users should be able to compare our products

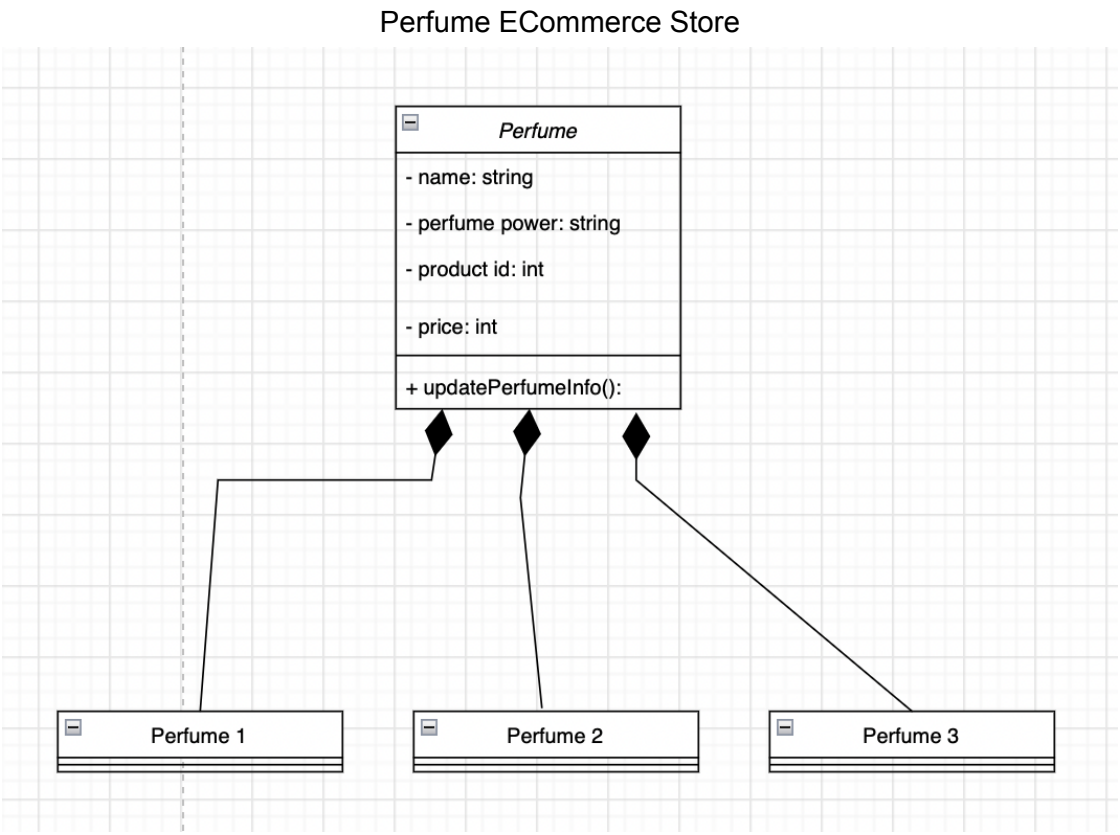
Technical feasibility of DB operations:

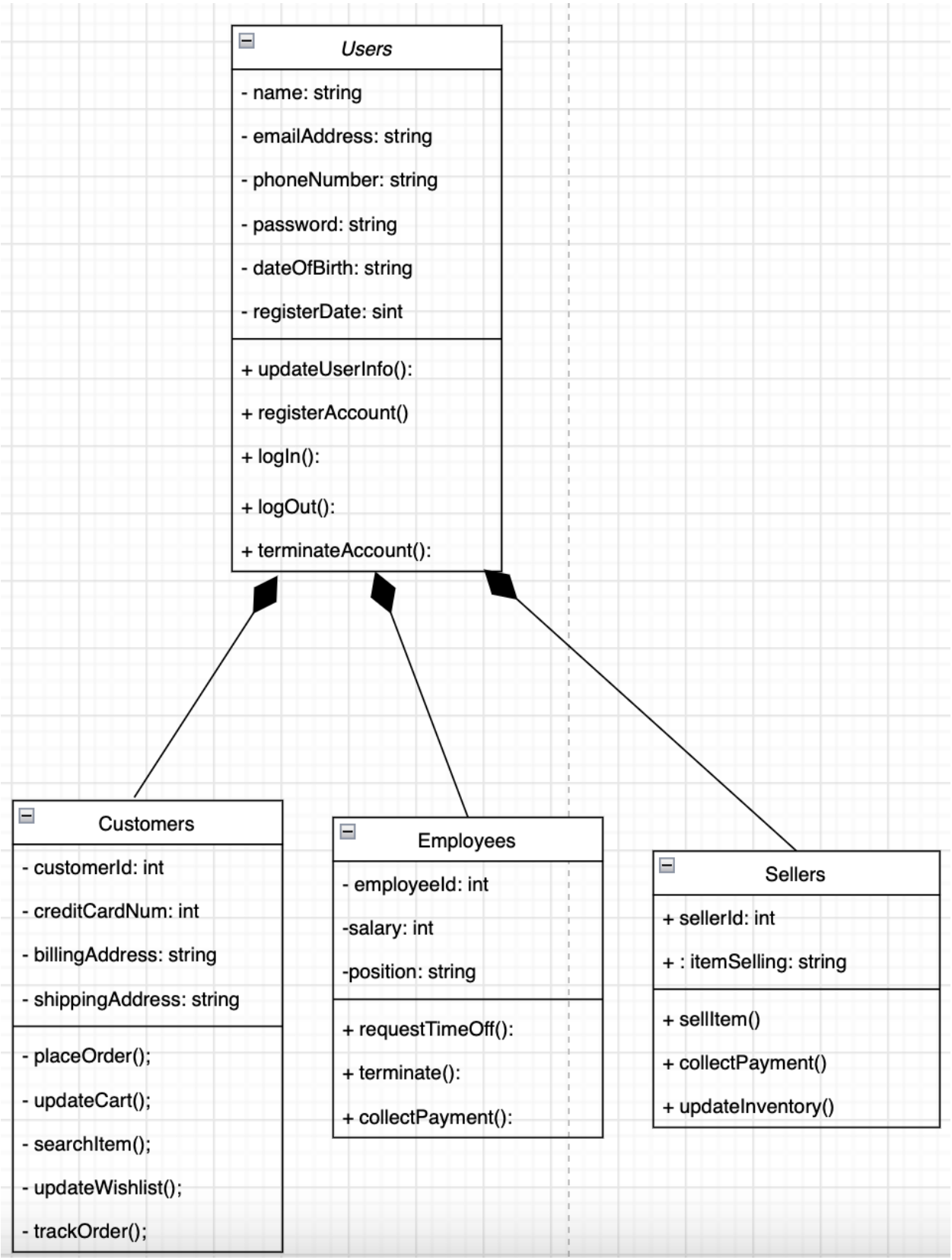
Considering our data definitions and functional requirements, users will be allowed to search our storefront website as a guest, or become registered users. The more important functionality is having secure, modifiable account information, as well as order information so that we can keep track of our products, for example, when a user/guest purchases a product, we will modify the number of **products_available** for each type of product the user purchased.

Our data definitions have been simplified from M1 based on highest-priority first and fit into condensed tables that will make it easier for us to keep track information in our database and display that information back to the user through techniques such as relating the **order_number** back to the **student_ID** of the buyer.

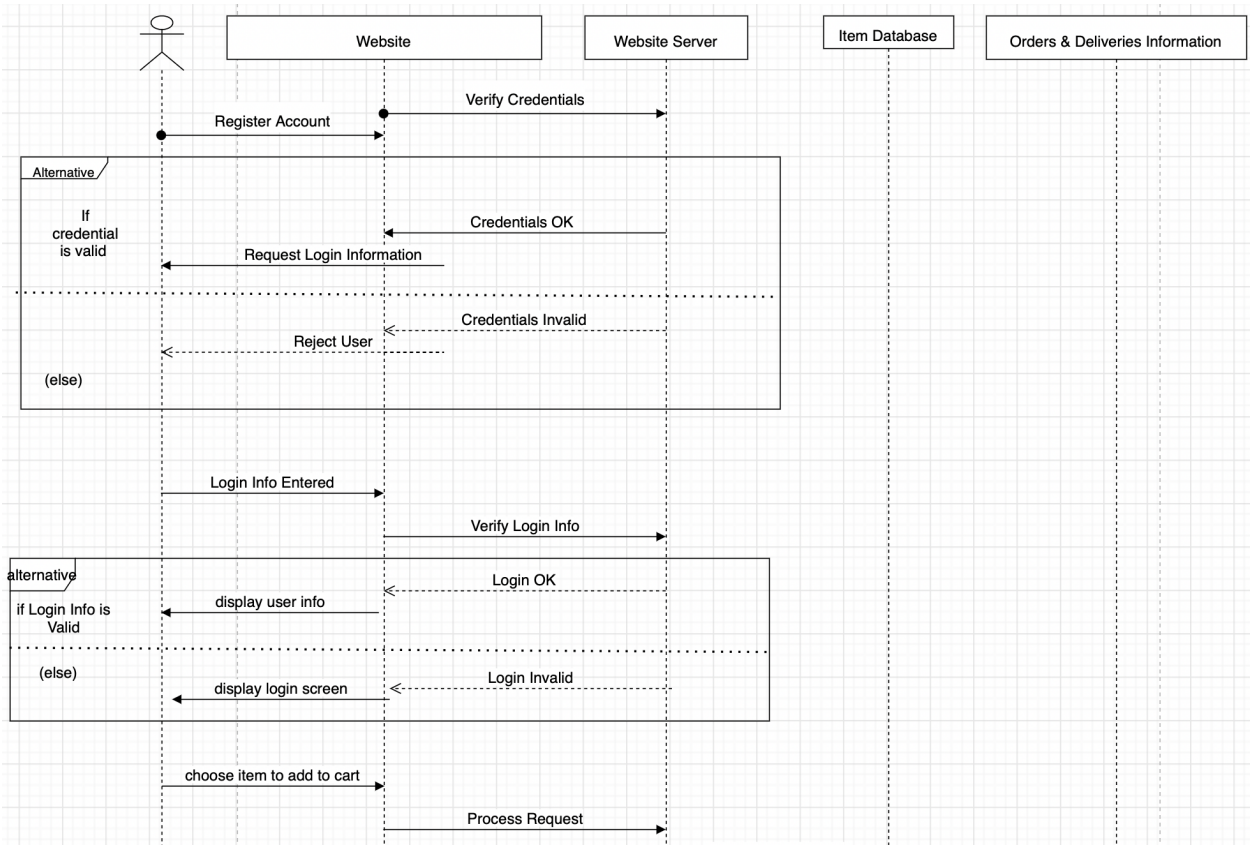
Additional Information:
Our API is created using HTTP, which allows users on the front end to communicate by sending requests for various information to our back end server. HTTP will be responsible for most of our Add/Delete/Search DB functionality. We will not be using any API

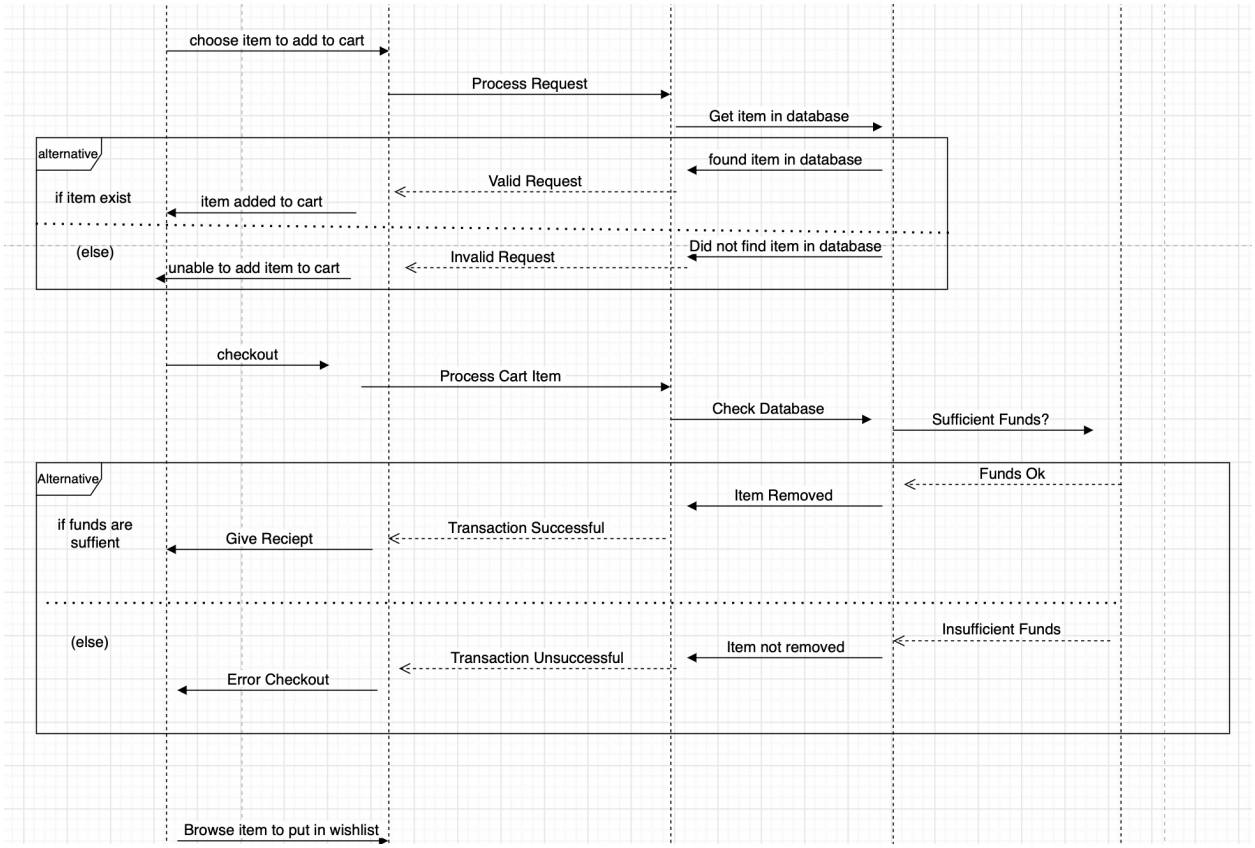
High Level UML Diagrams

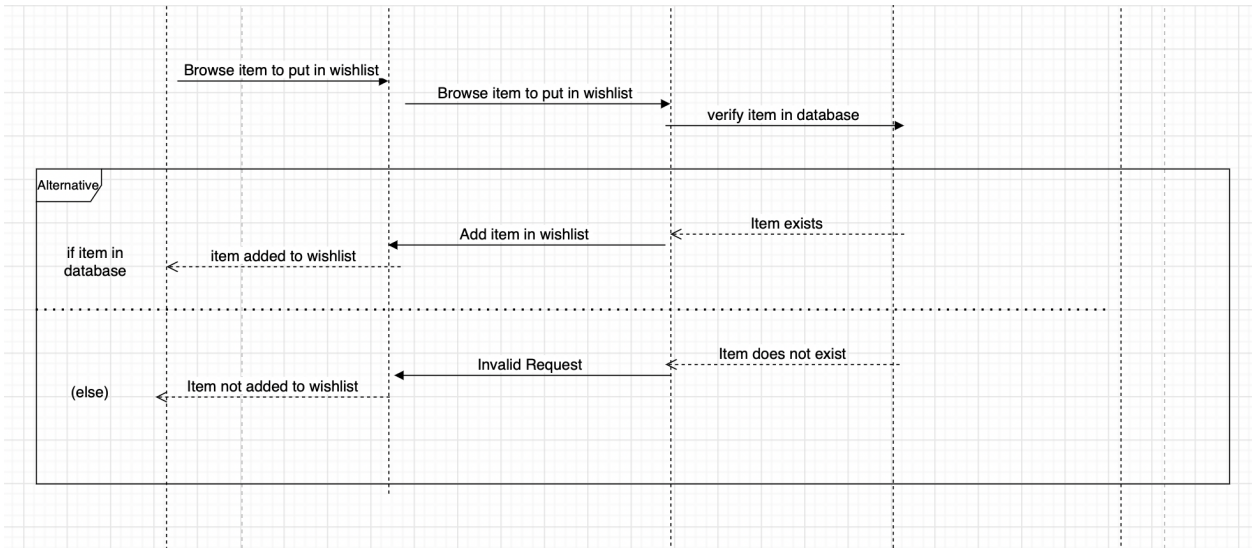



Users
- name: string
- emailAddress: string
- phoneNumber: string
- password: string
- dateOfBirth: string
- registerDate: sint
+ updateUserInfo():
+ registerAccount()
+ logIn():
+ logOut():
+ terminateAccount():
Customers
- customerId: int
- creditCardNum: int
- billingAddress: string
- shippingAddress: string
- placeOrder();
- updateCart();
- searchItem();
- updateWishlist();
- trackOrder();
Employees
- employeeId: int
-salary: int
-position: string
+ requestTimeOff():
+ terminate():
+ collectPayment():
Sellers
+ sellerId: int
+ : itemSelling: string
+ sellItem()
+ collectPayment()
+ updateInventory()

High Level Sequence Diagram




choose item to add to cart
Process Request
Get item in database
alternative
found item in database
Valid Request
if item exist
item added to cart
(else)
Did not find item in database
Invalid Request
unable to add item to cart
checkout
Process Cart Item
Check Database
Sufficient Funds?
Alternative
Funds Ok
Item Removed
if funds are suffient
Transaction Successful
Give Reciept
(else)
Insufficient Funds
Item not removed
Transaction Unsuccessful
Error Checkout
Browse item to put in wishlist


Browse item to put in wishlist
Browse item to put in wishlist
verify item in database
Alternative
Item exists
Add item in wishlist
if item in database
item added to wishlist
Item does not exist
Invalid Request
(else)
Item not added to wishlist

Key Risks for Project

Skills risks and mitigation plan

Do you have a proper study plan to cover all the necessary technologies?

- Yes. The front end team, along with help from the universal helper, will be assigned days to study WWW pages to assist the back-end team and research UI Mockup and Storyboards.

Risk summary

- If the front-end team or back-end team do not tend to their assigned study plan to research the technologies to perform his role, we will rely on our universal helper to have done it. That is, the universal helper will reserve as backup if the front end does not fulfill his duties. The risk may come from unwelcome emergencies, such as another team member requiring to attend exams for other classes. If the universal helper also does not perform, GitHub master will work with both of them to make up for lost progress.

Schedule Risks

Does your team have a team schedule for every member including their detailed task?

- Our team lead will create a checklist on a Google document which the rest of the team members will periodically check to know the details and deadlines of their tasks.

if change happens, does it update transparently? Does your team use project management tool (e.g. Jira, Trello)

- When changes happen, such as that to meeting schedules, our team will await instructions from the team lead via iMessage and Discord to announce a new meeting schedule.
- The team lead will also use a group-scheduling website called LettuceMeet to ensure the meetings will be successful, i.e., to know optimal times when the team can attend them. These will be transparent because everybody has access to LettuceMeet, and therefore to everybody else's availability.
- In addition, everybody will also be synchronously informed via iMessages or Discord, which we have been instructed to constantly be alert for since the beginning of our group's formation.

Risk Summary

- A risk in scheduling is that one or more team members may not be available to meet on our designated meeting day. If this happens, said team members will inform the team lead who will provide instructions to create new availability on LettuceMeet. If in-person meeting is not possible, Zoom or Discord will be utilized.

Teamwork Risks

Is everybody meeting regularly?

- Yes. Mondays, Wednesdays and Fridays either in-person or online.

Is everybody keeping his/her pace? If not, what is your plan to mitigate the risks?

- Yes, so far everybody has kept his or her pace. To mitigate the risk of this not happening in the future, Team lead assigns deadlines to tasks assigned to the rest of the team members. Team lead will have known and if so, team members will be encouraged to ask other team members who are done for help.

Risk summary

- If team members are not working well together, a solution is adjustment. That is, Scrum master and Team lead, in private, will periodically check in with the rest of the team members to audit their collaborative effectiveness and to know if they are meeting their task deadlines, respectively. For example, Scrum Master may speak to the front-end and universal helper individually to investigate if they are working well together; if not, she will make appropriate adjustments to build a solution, such as switching roles or teams.

Legal and Content Risks

(can we obtain content/SW we need legally with proper licensing, copyright?).

Risk Summary

- The package manager for Javascript programming language follows the Arctic License 2.0, which permits copying and distribution of the program without changing it. We are not changing it, therefore there is no risk in continuing to use it. Moreover, we are not using another API besides the one that we are creating with our website; hence, this will pose no legal risks with proper licensing copyright since we are not using other APIs

Project Management

During our team meetings, the team lead goes over the entire M2 document with the team to ensure that everyone understands the expectations of the milestone and the timeline we are given. After that, the tasks are assigned based upon our defined roles in the team. With Shahriz being the front end lead, he was assigned to the data definitions and functional requirements. Because Aisha is more creative, she took the lead on the ui mockups and storyboards. Chris took the task of working on the APIs and high level UML diagram because he is mostly working on the backend section of the project. Ahmed being the universal helper, he chose to work more on the high level architecture and database organization. Mandy took charge of the key risks and being the team lead, I chose to work on project management, as I would know how things are being separated into tasks and given to each member of the team.

After each person was assigned their tasks and sections, we were supposed to take a few days to do further research into what our portion of the project was and how to complete it. Those that felt that they did enough research started on their parts, while others came to our check in meeting with questions about their respective sections. What we typically like to do is go around and everyone has the floor to discuss their parts of the project. From there, we have our Scrum Master taking down notes of the meeting and noting down what was done and what needed to be worked on. When everyone has gone through their sections, we break up into groups to continue to work on our parts and because half of our team is usually on zoom, we set up breakout rooms as well.

For those in person, we take the white boards and write out what we want to accomplish and draw out what we want the website to look like. As a team, we try to be as transparent and realistic as possible. We have fostered a trust between one another to be upfront about our progress on our tasks, rather than fluffing up our work to appear completed. Right before we leave our meeting, we go over what was done and by whom, what needs to be done by when, when check in times will be and when the entire assignment is due so that everyone is on the same page and understands their portion. We do not use a tool to manage everyone's tasks; however, we do utilize our group messages and post-its to keep organized.