Team X

Online Computer Store
Design Report
Data Structure and Logic

Version 1.0

Team Members:

Chelsea Lantigua
Shazad Outar
Randy Castillo
Yasiris Ortiz
Shiva Shrestha

Table of Contents

Introduction **2**

Use Cases Scenarios **3**

Registered Customer Collaboration Diagram 3

Visitor User Collaboration Diagram 4

Manager Collaboration Diagram 5

Store Clerk Collaboration Diagram 6

Delivery Company Collaboration Diagram 7

Manager Petri-net Diagram 9

Registered Customer Petri-net Diagram 10

Delivery Company Petri-net Diagram 10

E-R Diagram **12**

Detailed Design **13**

Visitor Detailed Design 13

Customer Detailed Design 13

Store Clerk Detailed Design 17

Delivery Detailed Design 18

Manager Detailed Design 20

System screens **24**

Login/Register GUI 24

Homepage GUI 25

Discussion Forum GUI 26

Shopping Cart GUI 27

Checkout GUI 28

Store Clerk Dashboard GUI 29

Delivery Company Dashboard GUI 35

Manager Dashboard GUI 38

Product **39**

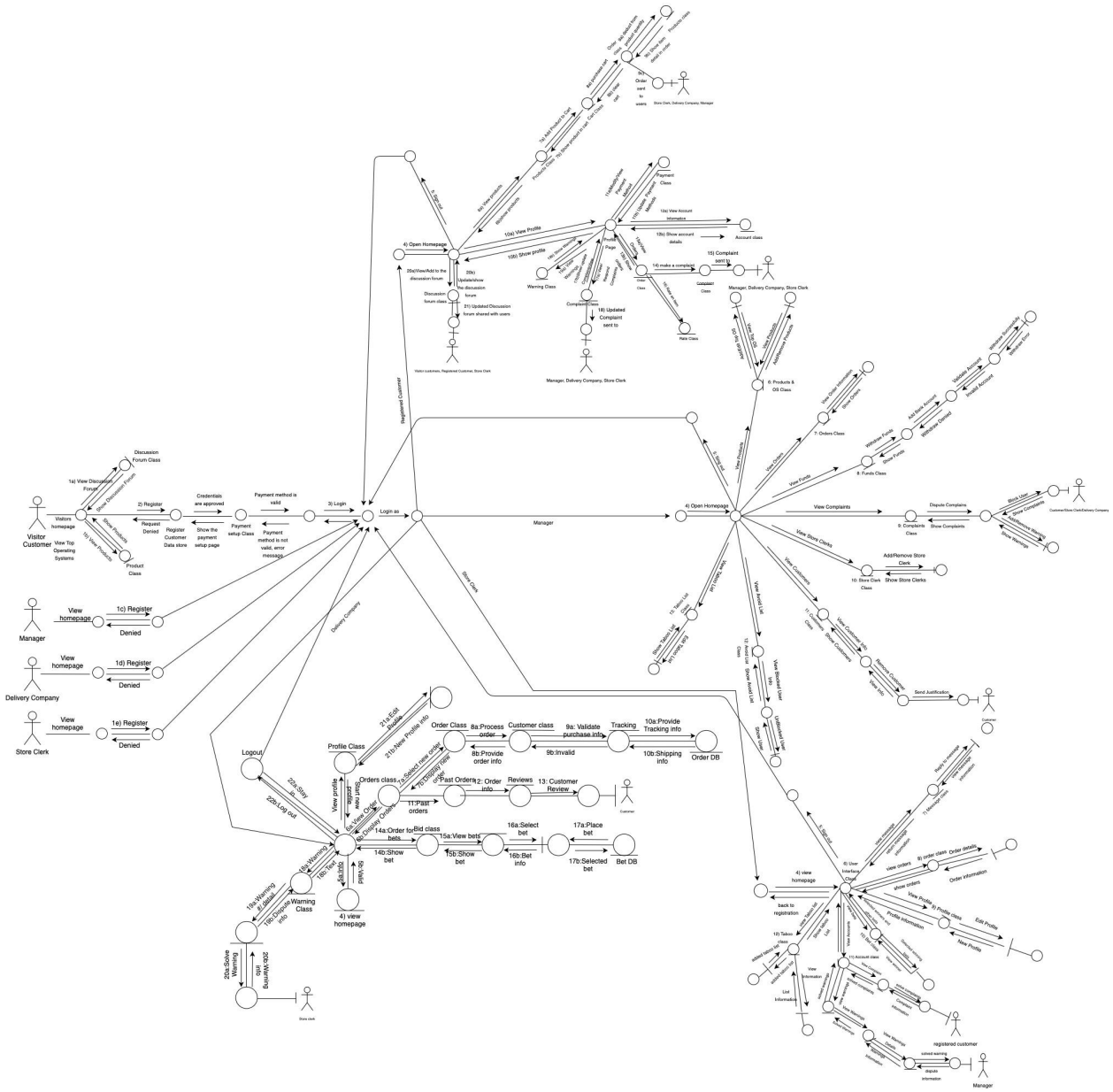
Placing Bet Prototype **40**

Group meetings **43**

Github (github): **44**

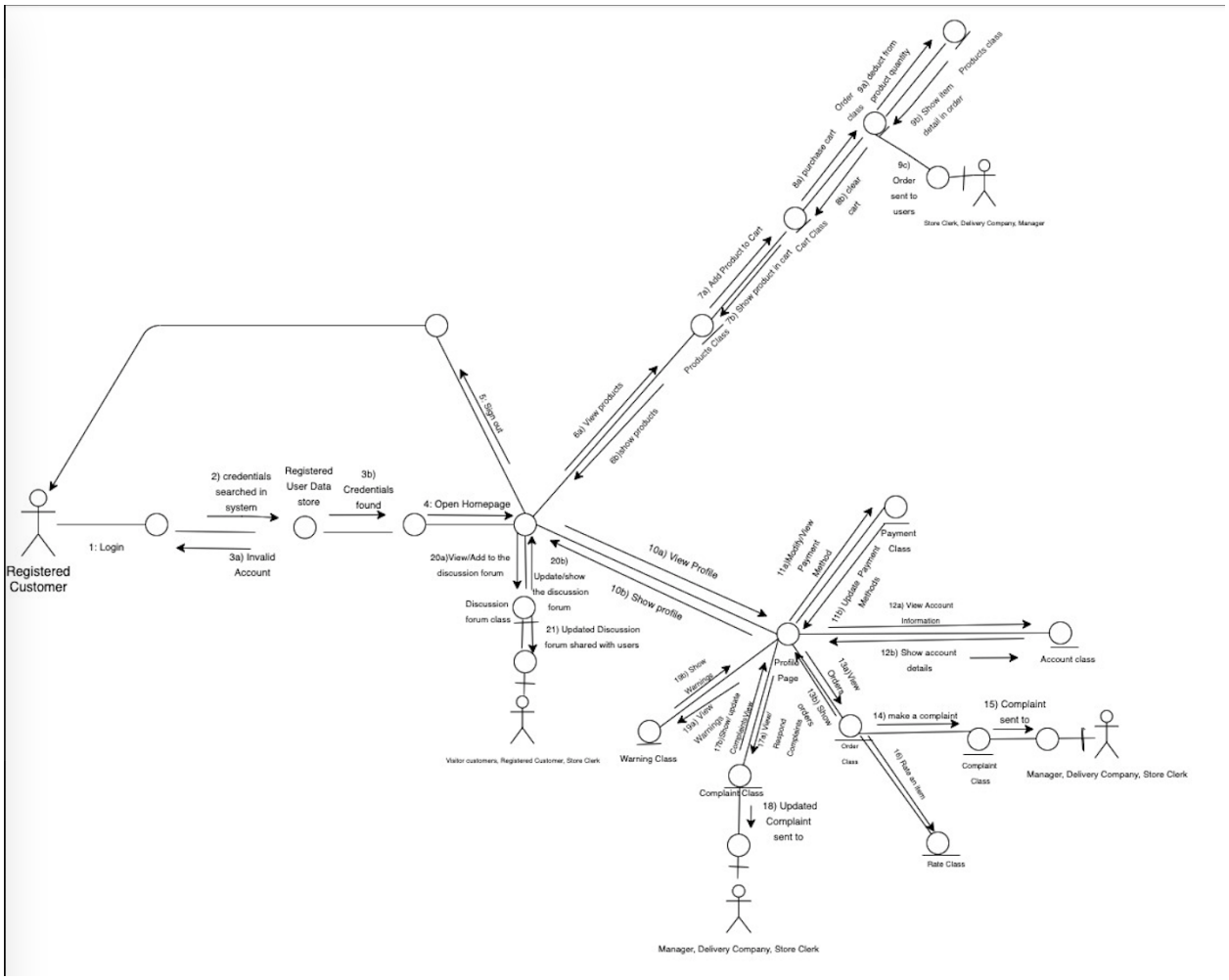
Introduction

Overall picture of the system using collaboration class diagram

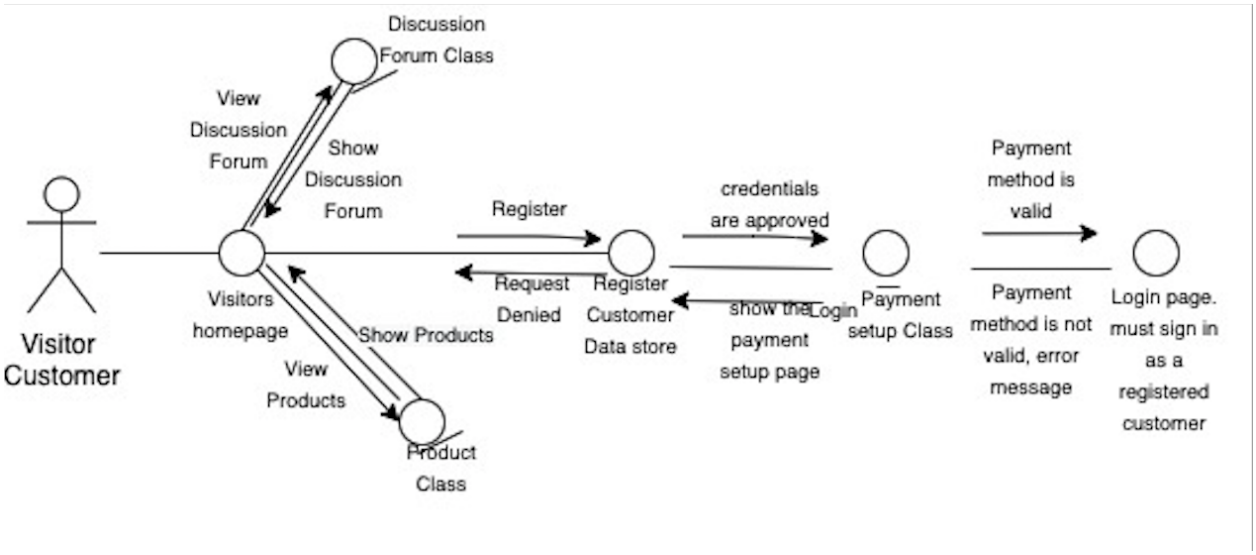


Use Cases Scenarios

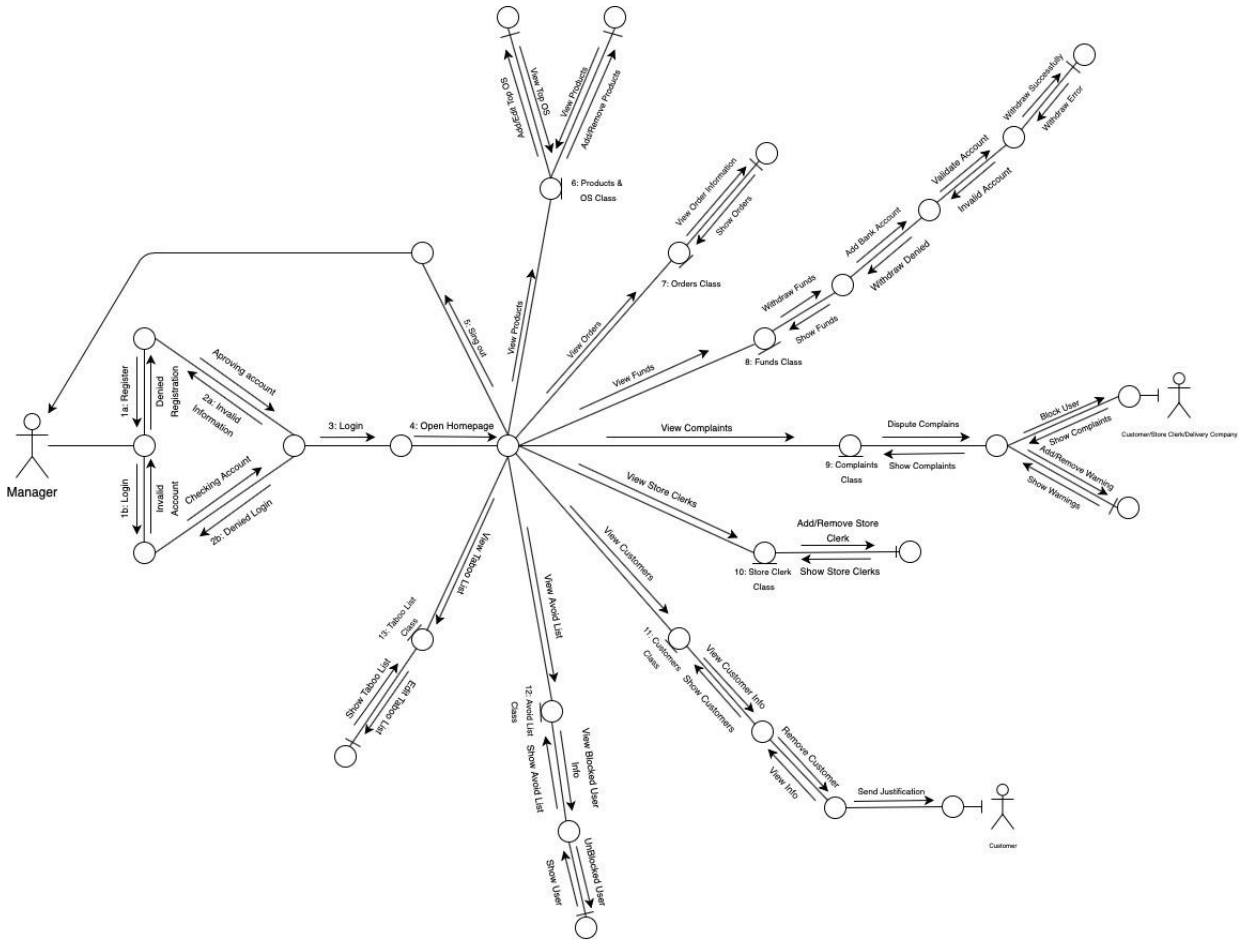
Registered Customer Collaboration Diagram



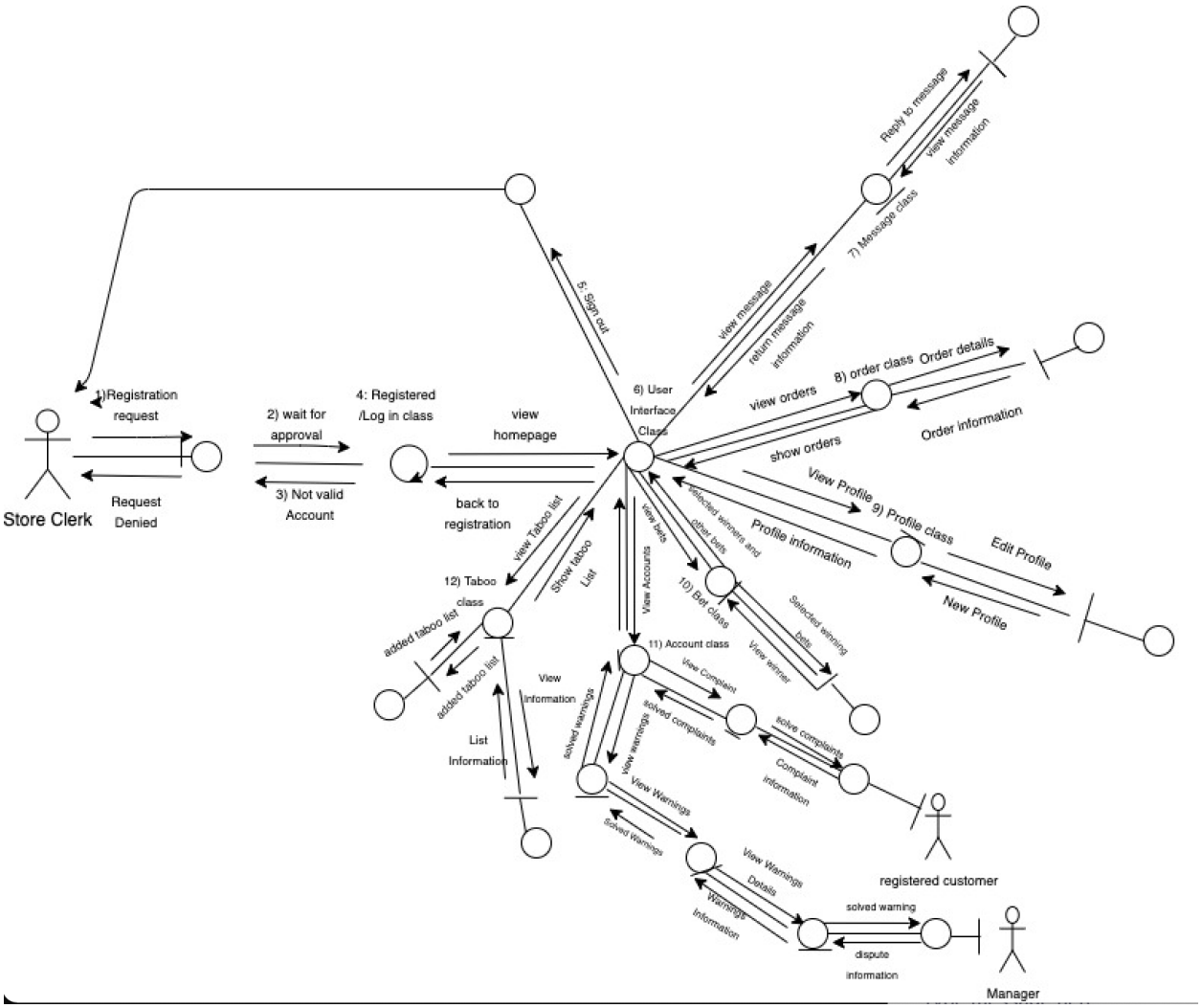
Visitor User Collaboration Diagram


Discussion
Forum Class
View
Discussion
Forum
Show
Discussion
Forum
Register
credentials
are approved
Payment
method is
valid
Visitor
Customer
Visitors
homepage
Request
Denied
Register
Customer
Data store
show the
payment
setup page
Login
Payment
setup Class
Payment
method is not
valid, error
message
Login page.
must sign in
as a
registered
customer
Show Products
View
Products
Product
Class

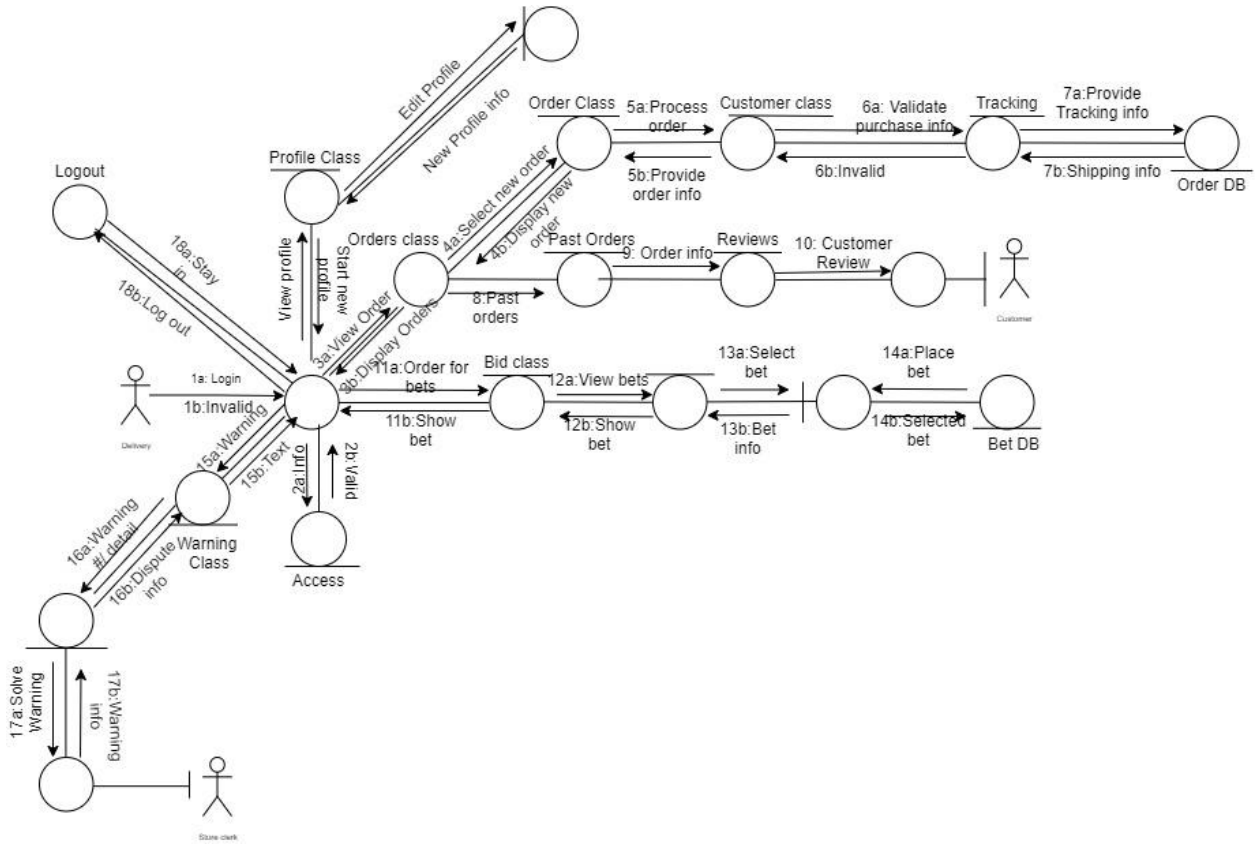
Manager Collaboration Diagram



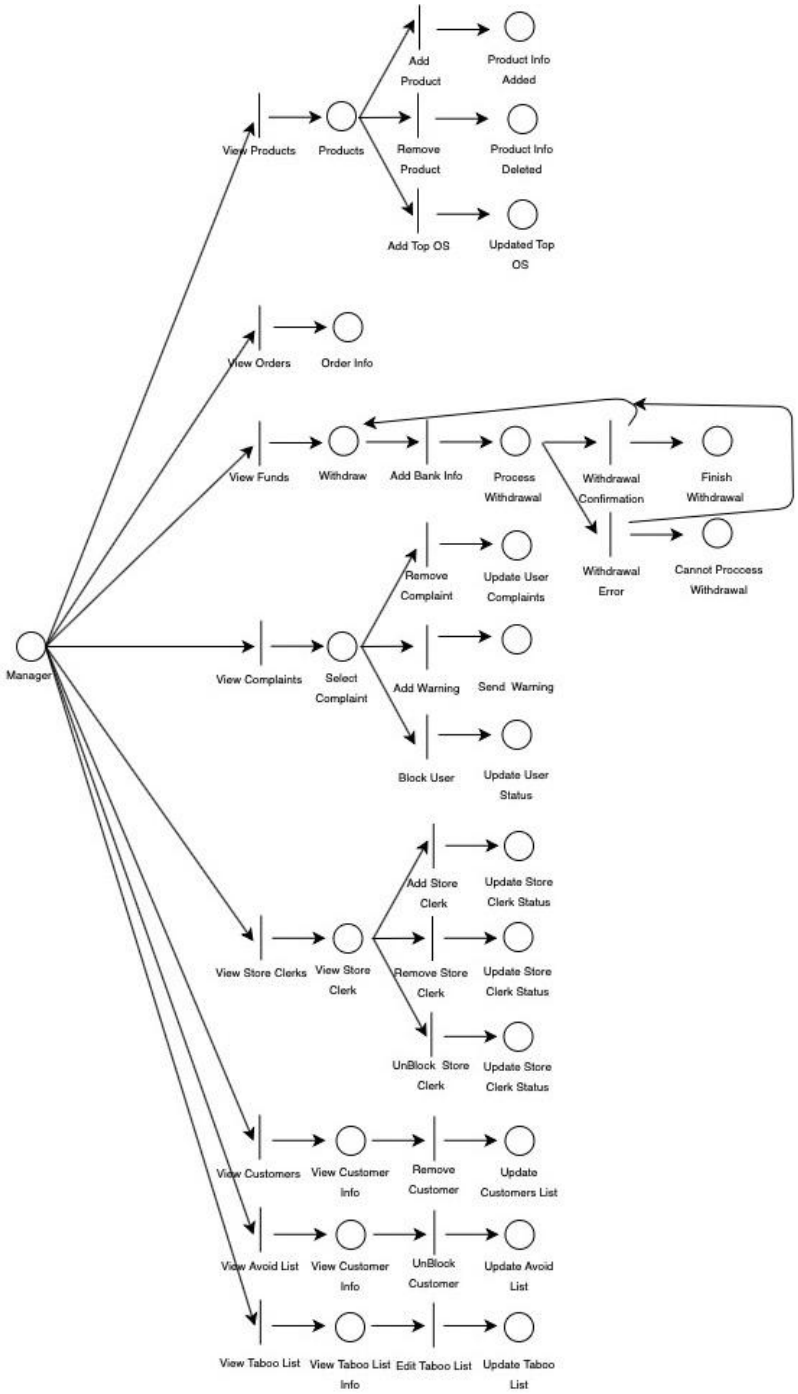
Store Clerk Collaboration Diagram



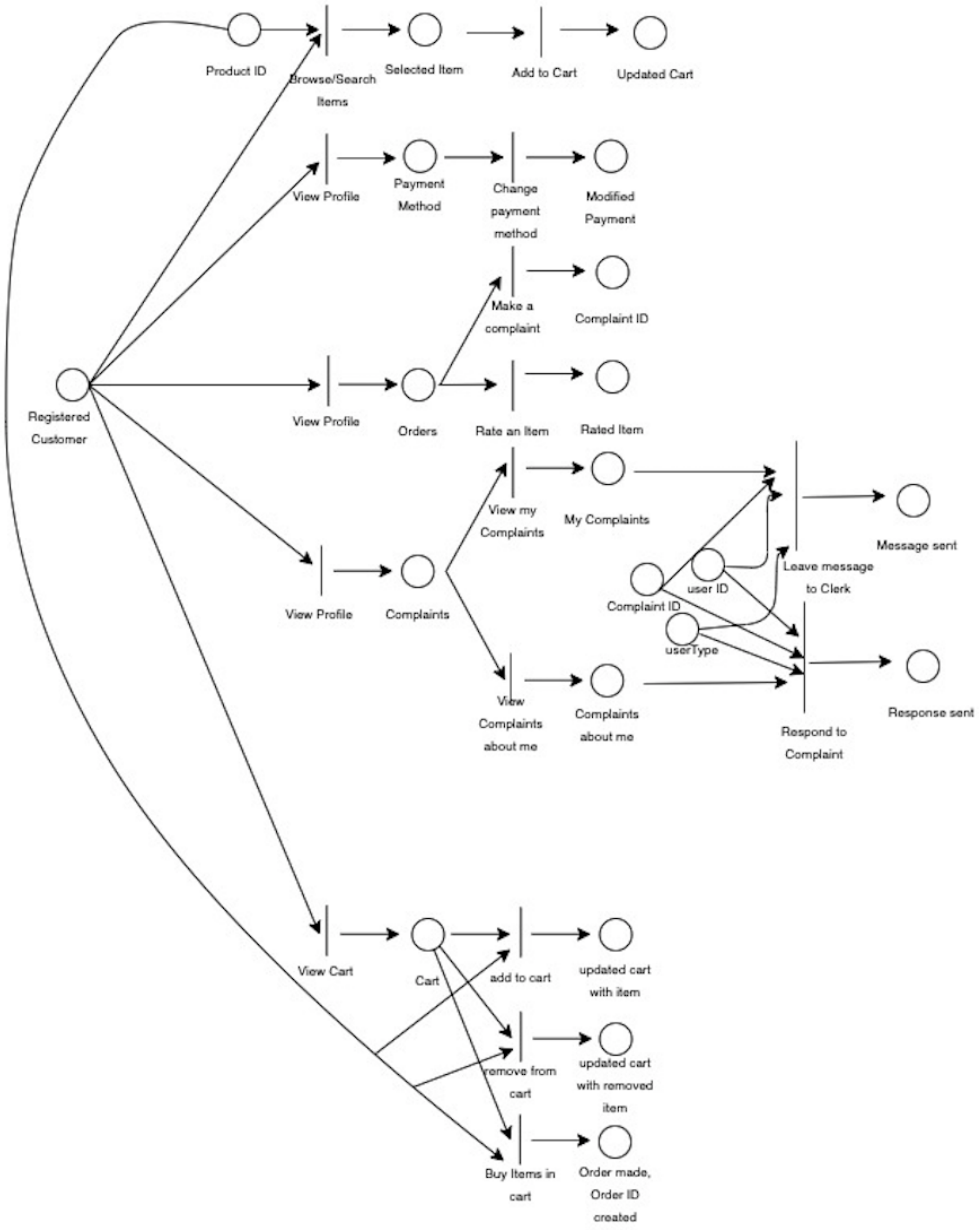
Delivery Company Collaboration Diagram



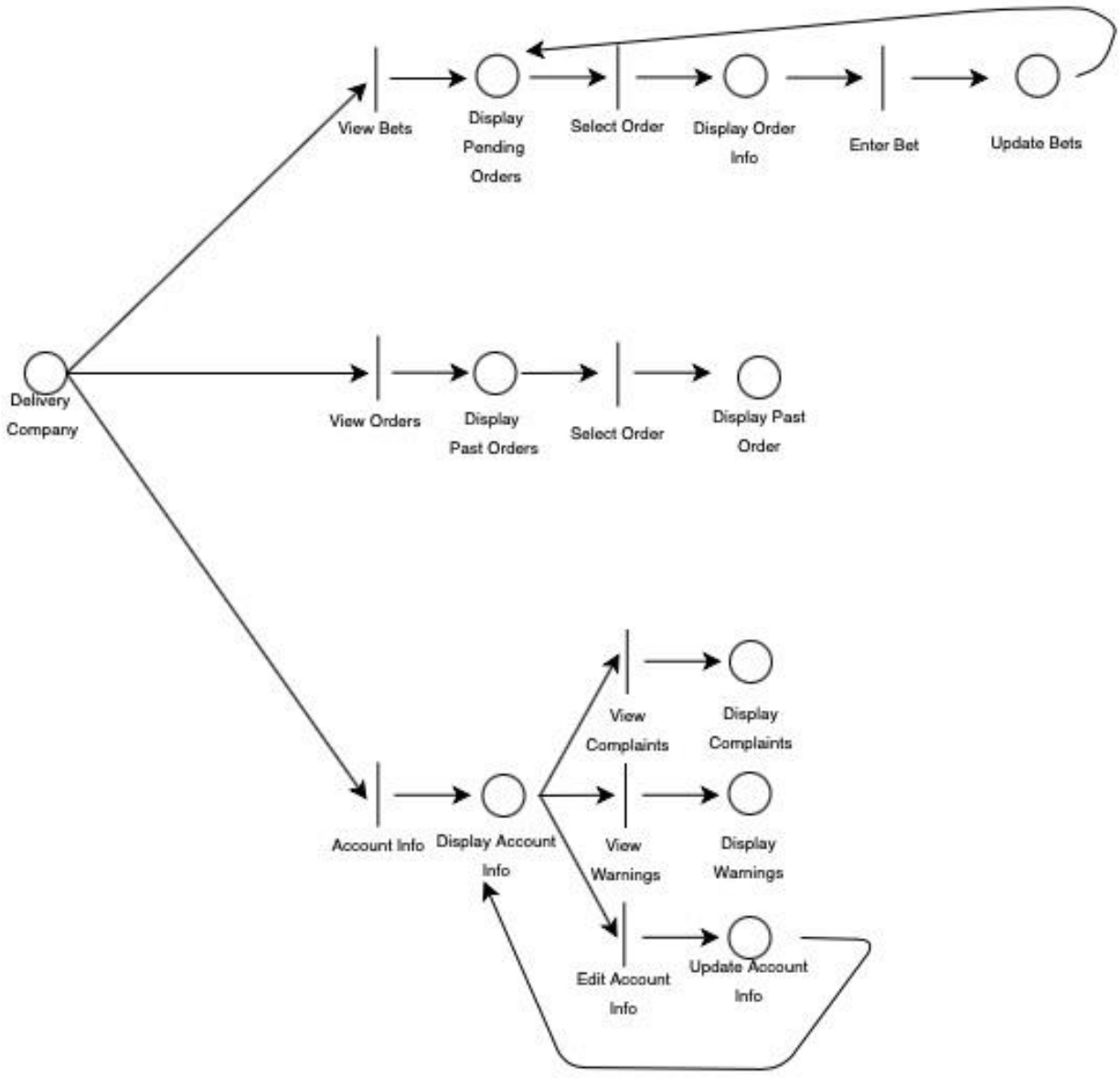
Manager Petri-net Diagram



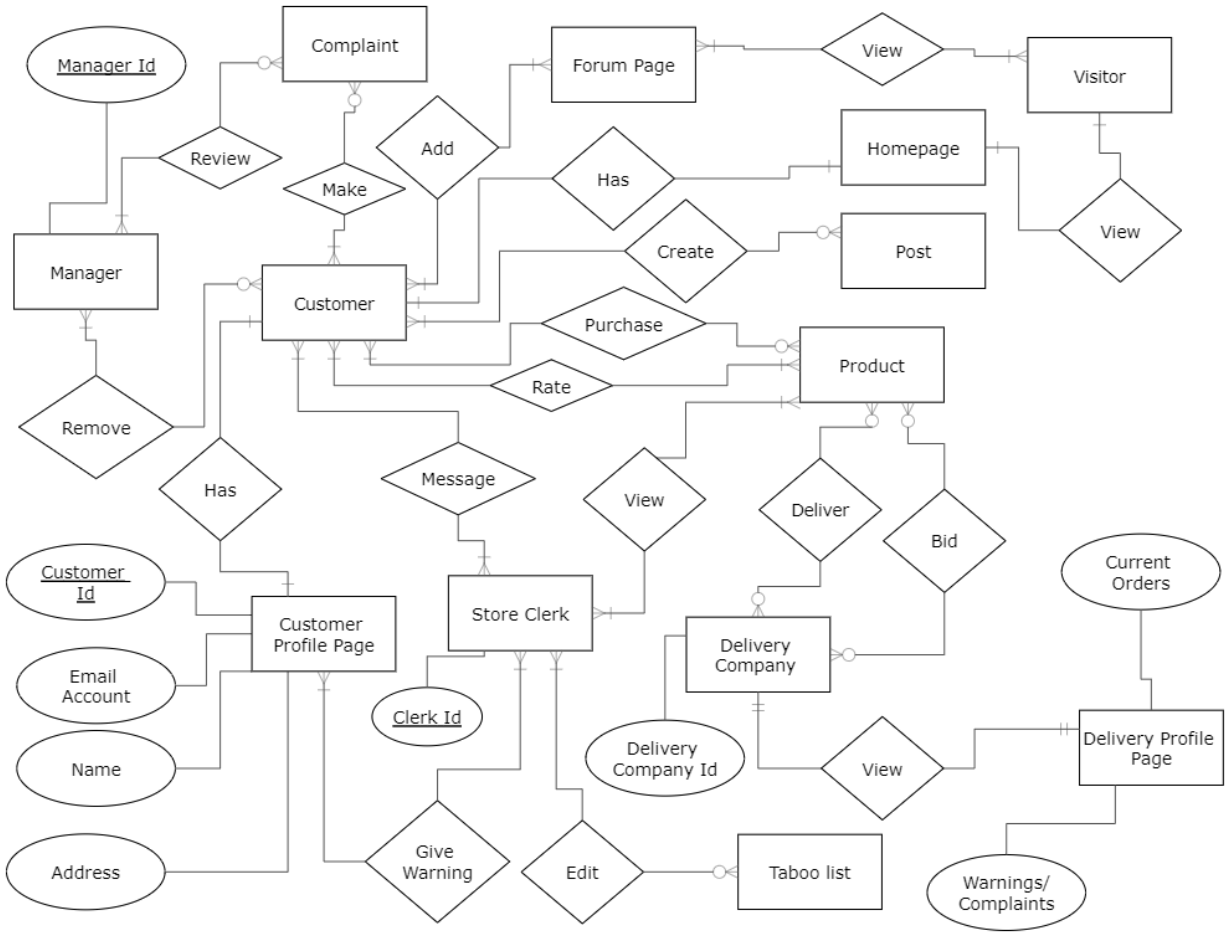
Registered Customer Petri-net Diagram



Delivery Company Petri-net Diagram



E-R Diagram


Manager Id
Complaint
Forum Page
View
Visitor
Review
Add
Make
Has
Homepage
View
Manager
Create
Post
Customer
Purchase
Product
Rate
Remove
Has
Message
View
Deliver
Bid
Customer Id
Customer Profile Page
Store Clerk
Delivery Company
Current Orders
Email Account
Clerk Id
Delivery Company Id
View
Delivery Profile Page
Name
Address
Give Warning
Edit
Taboo list
Warnings/ Complaints

Detailed Design

Visitor Detailed Design

checkEmail function will take the email the visitor user wants to use to become a registered customer. The *registeredCustomer.txt* will be read to see if any other registered user already has it. If so, then the visitor will get an error message. If not then the user will be directed to the sign in page.

checkPayment function will verify the payment method the visitor inputs when they complete the second step of registration. The length of the card number is read to determine if it is 16 digits long. If the card is 16 digits then the card is valid, if not then they will get an error message.

Customer Detailed Design

checkUserType function will record what user type is trying to sign in and then generate the correct homepage for them if their passwords and username have been stored in the following textfiles:

- Registered Customer - *registeredCustomers.txt*
- Store Clerk - *storeClerks.txt*
- Delivery Company - *deliveryCompanies.txt*
- Manager - *managers.txt*

generateProfilePage function reads the *registeredCustomers.txt* and then displays the information on the users profile page

generateItems reads the *inventory.txt* generated when the manager adds items to the inventory and writes the information to the homepage. This file should contain the following information about items:

- Product name
- Product ID
- Product price
- Product description
- Product Image ---yasiris
- Type of Operating System

- Number of Times Sold
- In stock count

generateTopOS reads the *topOpSystems.txt* from the manager which states the top operating systems they have selected. The information read will be displayed on the homepage. The topOpSystems.txt will include the following information:

- Name of the OS system
- An image representing the OS system

generateTopSellers reads the *topSellers.txt* from the manager which states the items most sold. This file will contain the ID of the item and will use these unique IDs to traverse the *inventory.txt* to find additional information to display - Name of the product, product price, and image of the product. The function will display the following on the homepage:

- The Product Id

addToCart is called when a user adds a product to their cart. The function will look at the labels corresponding to that product which will be the name of the product, the price, and its description and will search the *inventory.txt* to get additional information like it's product ID. All of this information will then be written into *cart.txt*. Every registered customer will have their own *cart.txt*. As this text file is modified the user's cart page will be updated. This function will also be able to read the quantity the user wants of a product. The cart.txt will include the following:

- Product ID
- Product Name
- Quantity
- Price
- Total (Quantity by Price. These values will be added to determine the final total of the entire cart)

generateTopOS1 will be called when a user presses to see all the buttons associated with a top Operating system. The function will look at the associated top OS system and will traverse through the *inventory.txt file to* display those items on a page. The items will be organize with their names, images, and price.

generateItemPage will be called when a user presses the image of a product, this will lead them to an item page where they will see more information about the item. The function will look at the labels corresponding to that product which will be the name of the product, the price, and

image and will search the *inventory.txt* to get additional information like it's product description. The function will also search the ratings.txt to find the rating of the product based on it's product ID. This information will be displayed on the item page. The item page will include the following:

- Product Name
- Price
- Description
- Ratings
- Image

generateRatings will compute the ratings of a product based on the response of users on a scale (1-5) and the number of customers, an average will be generated. These two quantities will be stored in a separate text file called rating.txt that will store all products that have been rated by someone. The rating of the product will be displayed on the item page. The rating.txt will contain the following

- Product ID
- Rating sum
- Number of Customer
- Average of the ratings

generateOrder function will refer to the cart.txt file to write the order into another text file, *order.txt*. New Orders will be written to the top of the file and old ones to the bottom. The orders.txt will also be unique to a customer and will contain all the information in the cart.txt and additional information including the following:

Order Number

- Order Status
- Tracking Information
- Shipping Address
- Store Clerks that handled the order
- Delivery Companies that are delivering the items

generateOrderPage will generate a user's orders page based on the information in the *order.txt* file.

generateComplaint will store the data from a user when they fill out our complaint form.The message should be written into a complaint.txt file which will contain the following:

- Customer ID who is making the complaint
- Order ID
- user ID to who the user Directed the message
- The description of the message
- Status - updated by manager
- Resolution - updated by manager

generateComplaintPage will read the complaint.txt and write the information into the complaints page.

generatePaymentPage will refer to a payment.txt file unique to every customer to write out the customers different payment methods they have approved. payment.txt will contain the following:

- customer ID
- card number
- name on card
- billing address

addNewPayment will record the user's input when they completed the form to add a new payment method. The information associated with this new payment method will be recorded into the payment.txt.

generateWarningPage will read the warning.txt file which will be unique to every customer and maintained by the manager. The information contained in this file includes the following:

- the number of warning
- the reason for the warning
- The date.

chatStoreClerk will take as input the message the user inputs into a chatbox and their Customer ID and stores it into the *chat.txt*. These one-to-one chats will be stored in unique textfiles. A store clerk chosen at random will be able to view this message and respond directly to the customer. This will then be written into the text file for the customer to read.

generateChatHistory will generate a view of the messages that are being sent between a store clerk and a customer. The function will refer to the *chat.txt* file to create the chat history view.

Store Clerk Detailed Design

profileRequest will check the registration request from the store clerk and return an approved or denied application depending if all the information is correct.

viewProfile will read the user personal information and allow any edits or new information. It will return the new profile information.
getMessage will read the chat.txt file and generate a list with each message as an item in the list. Each item in the list will contain

- Username
- Message

replyMessage will add the text input in the generated list from getMessage, and then will use the list to add the message to the chat.txt

displayMessage will display the items from the generated list from the getMessage and display each item in the new line

addTabooList will add a taboo to the *tabooList.txt* file

displayTabooList will display the taboo list from the *tabooList.txt* file

getOrders will read the unique #ordernumber.txt file and generate a list with all the info of the Order.

displayOrderInfo Will retrieve all the order information and display it in the View Order/View Bets Page. Each order in the list will contain

- Order ID
- Date Created
- Customer Id
- Delivery Status
- Total Order Cost
- Delivered Date

getAccount will contain the user sensitive information and provide access to the methods below getComplaint and getBets.

getComplaint will read the unique complaints. file of each user and generate a list with all the info of the Complaints.

getWarnings will retrieve all the Warning information and display it in the Warnings frame with the following details:

- Warning Header
- Warning description
- Warning Count

getBets will read the bet_#ordernumber.txt file, where every bet_#ordernumber.txt would be unique to the order number and generate a list of the bets of the specific #ordernumber.

selectBet will confirm the delivery company for the selected order.

displayBets
will retrieve all the bets information and display it in the Bets table. Each order in the list will contain

- Price
- Delivery Company Name

Delivery Detailed Design

getOrderInfo will read the unique #ordernumber.txt file and generate a list with all the info of the Order.

getProductInfo will read the unique productId.txt file and generate a list with all the info of the Product.

getBets will read the bet_#ordernumber.txt file, where every bet_#ordernumber.txt would be unique to the order number and generate a list of the bets of the specific #ordernumber.

getComplaints will read the unique complaints. file and generate a list with all the info of the Complaints.

getWarnings will read the unique complaints. file and generate a list with all the info of the Complaints.

displayOrderInfo Will retrieve all the order information and display it in the View Order/View Bets Page. Each order in the list will contain

- Order ID
- Date Created
- Customer Id
- Delivery Status
- Total Order Cost
- Delivered Date

displayProductInfo Will retrieve all the product information and display it in the View Bets Page. Each order in the list will contain

- Product ID
- Product Name
- Price
- Weight

displayBets will retrieve all the bets information and display it in the Bets table. Each order in the list will contain

- Price
- Delivery Company Name

displayComplaints will retrieve all the Complaints information and display it in the Complaints frame. Each complaint in the list will contain

- Complaint Header
- Complaint Text
- Customer Name

displayWarning will retrieve all the Warning information and display it in the Warnings frame. Each Warning in the list will contain

- Warning Header
- Warning description
- Warning Count

addBets will get the amount input and then add it to the bets_ordernumber.txt

Manager Detailed Design

displayStoreProducts will retrieve the store products information from *products.txt* and display them on the Products page. Each product on the list will contain the following attributes:

- Product Name
- Price
- Operating System (OS)
- Product ID

displayProductInfo will retrieve the full product information from *products.txt* and display it on the page where the manager can edit/view the product information. The page will contain the following attributes:

- Product Images
- Product Name
- Product Description
- Price
- Operating System (OS)

saveProduct will save any changes to a product to the *products.txt*. All the following attributes should be filled in order to save a product:

- Product Images
- Product Name
- Product Description
- Price
- Operating System (OS)

selectTopOS will save the top Operating Systems to be displayed on the homepage to the *operatingSytems.txt*. The manager can select any of the following Operating Systems:

- Windows
- macOS
- Linux

addStoreProduct will create an empty product information page where the manager can add the product information. The new product will be stored to the *products.txt* file and added to the store products.

removeStoreProduct will delete all the product information from the *products.txt* file.

diplayOrders will display all customers orders from the *customersOrders.txt*. Each order in the list will contain the following attributes:

- Order ID

- Date Created
- Customer Name
- Order Status (Fulfilled/Unfulfilled)
- Total Order Cost
- Order Profit
- Delivery Status
- Updated Time

diplayFunds will display the available funds that can be withdrawn from *funds.txt*. The manager needs to add a bank account in order to withdraw any funds.

addBankAccount will add the bank account information to the manager account and save it to *funds.txt*. The Bank account is composed of the following attributes:

- Bank Name
- Routing Number
- Account Number

saveBankAccount will save the bank account information to *funds.txt*. All the following attributes need to be filled:

- Bank Name
- Routing Number
- Account Number

withdrawFunds will begin the process to withdraw the funds to the bank account. If there are no funds available the function will return an error, else the withdrawal will be confirmed and funds will be updated on the *funds.txt* file.

displayComplaints will display all the users complaints from *complaints.txt*. Each complain in the list will contain the following attributes:

- User Name
- User Type
- Number of Complaints
- Warnings
- User Status (Blocked/Active)

displayComplainInfo will display the complaint information for a user from *complaints.txt*. The page will contain the following attributes:

- User Name
- User Email Address

- User Justification message
- Message to user from manager

removeComplaint will discard the complaint from a user and remove it from the *complaints.txt* file.

addComplaintWarning will add a warning to the user and automatically block the user if it has more than 3 warnings.

blockUser will change the user status to blocked and prevent it from login into the application.

unBlockUser will change the user status to active and give it back access to the application.

displayStoreClerks will display the store clerk list from *storeClerks.txt*. Each store clerk in the list contain the following attributes:

- Clerk ID
- Clerk Name
- Email
- Status (Waiting for approval/Active/Removed)

addStoreClerk will approve a store clerk to the store and give it access to the application. The store clerk status will change to active on the *storeClerks.txt file*.

removeStoreClerk will remove a store clerk to the store and remove access to the application. The store clerk status will change to removed on the *storeClerks.txt file.*

activateStoreClerk will enable store clerk access to the application. The store clerk status will change to active on the *storeClerks.txt file.*

displayStoreCustomers will display the customers from *customers.txt.* Each customer on the list contains the following attributes:

- ID
- Name
- Email
- Status
- Number of Orders

displayCustomerInfo will display the full customer information from *customers.txt*. The page contain the following attributes:

- Name
- Email

- Home Address
- Number of Orders
- Number of Complaints
- Number of warnings

removeCustomer will change the customer status to blocked on the *customers.txt* file and restrict its access to the application. The customer information will be added to the avoid list.

displayAvoidList will display all the blocked customers from *customers.txt*. Each Customer in the list contains the following attributes:

- ID
- Name
- Email
- Number of warnings
- Number of Orders

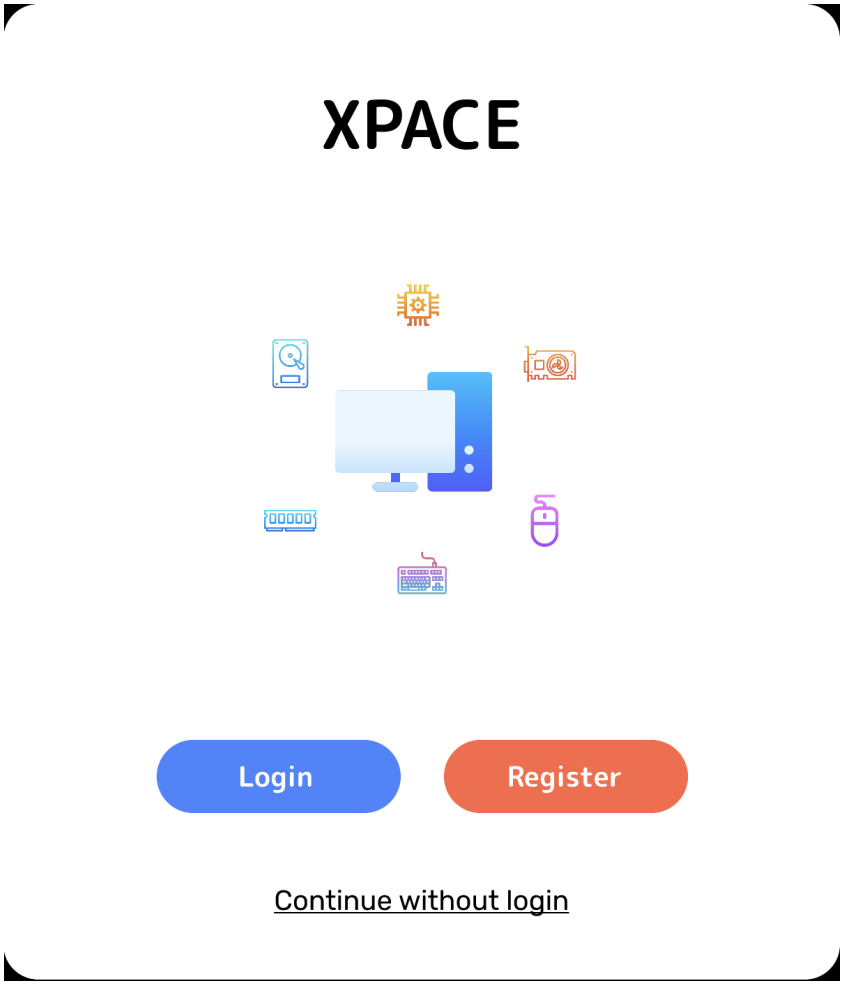
unBlockCustomer will change the customer status to Active on the *customers.txt* file.

displayTabooList will display the taboo list from the *tabooList.txt* file.

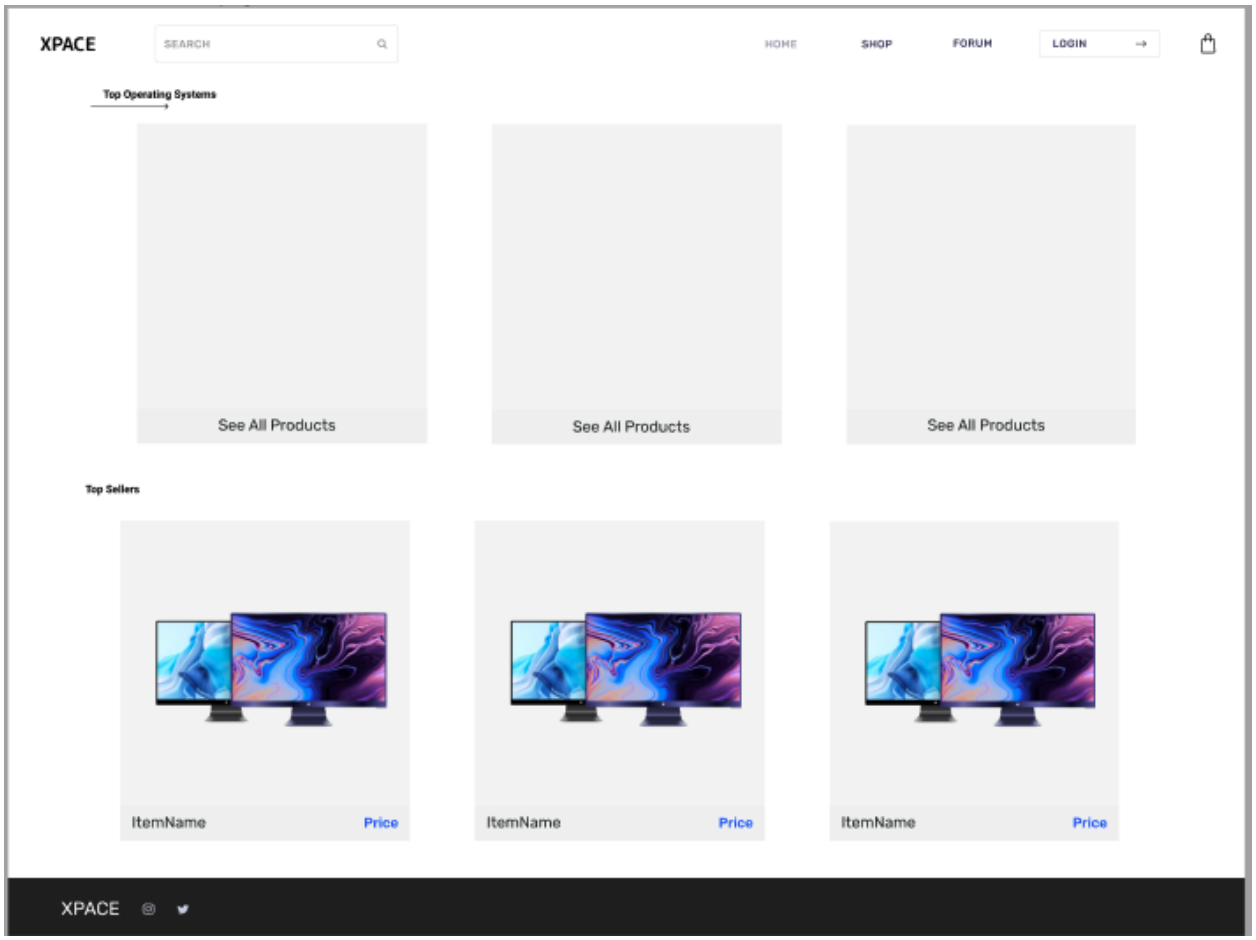
saveTabooList will save any changes made to the taboo list to the *tabooList.txt* file.

System screens

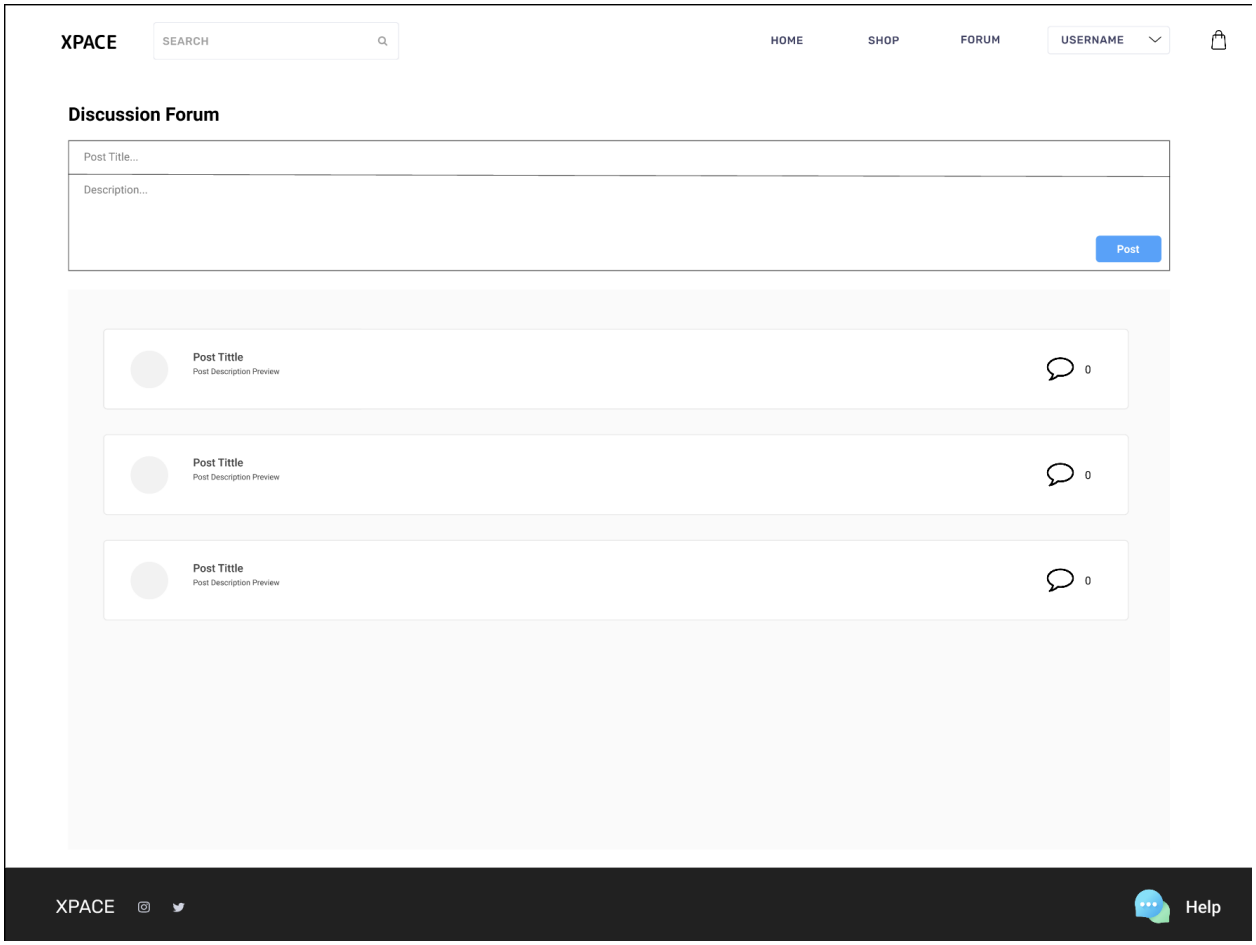
Login/Register GUI



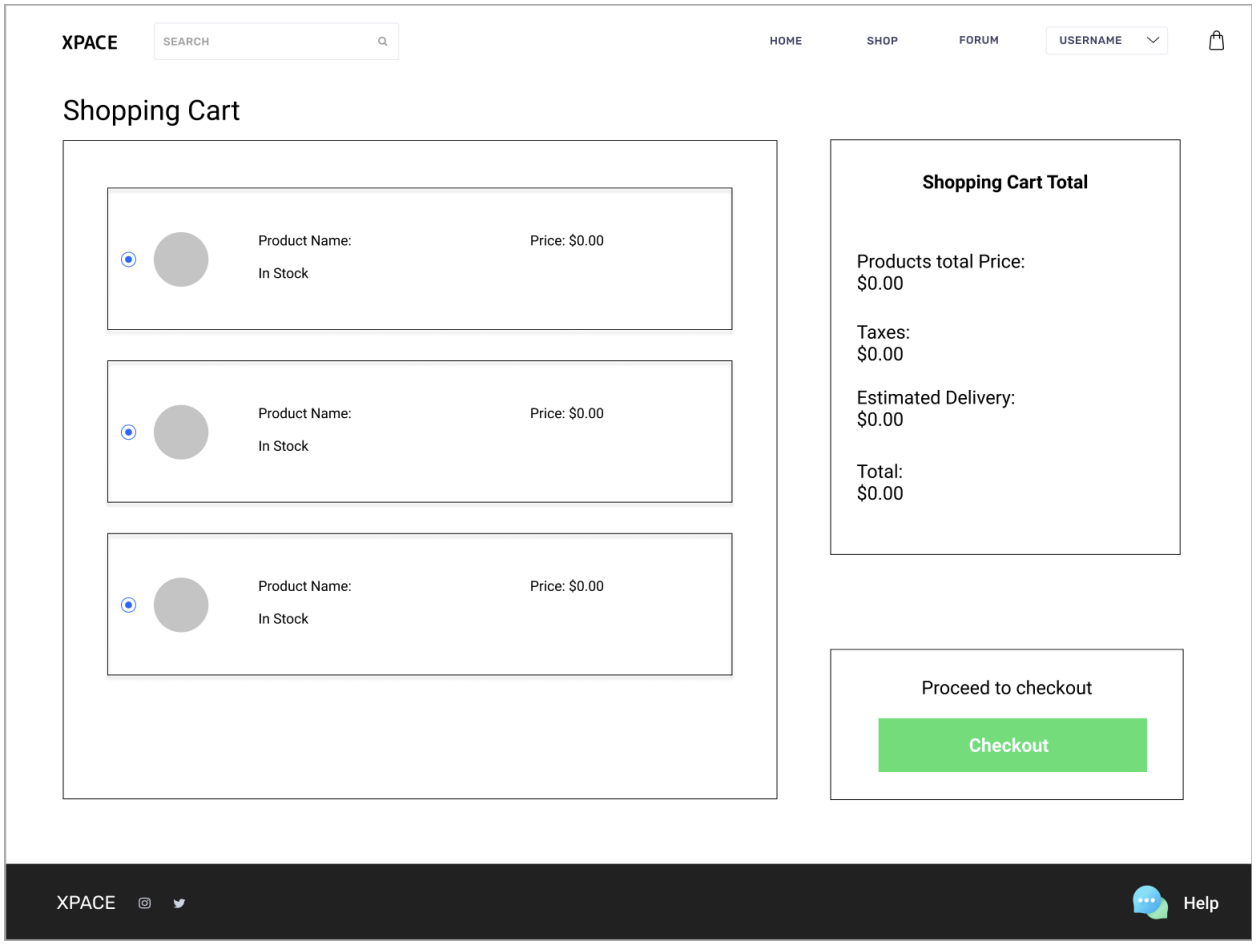
Homepage GUI



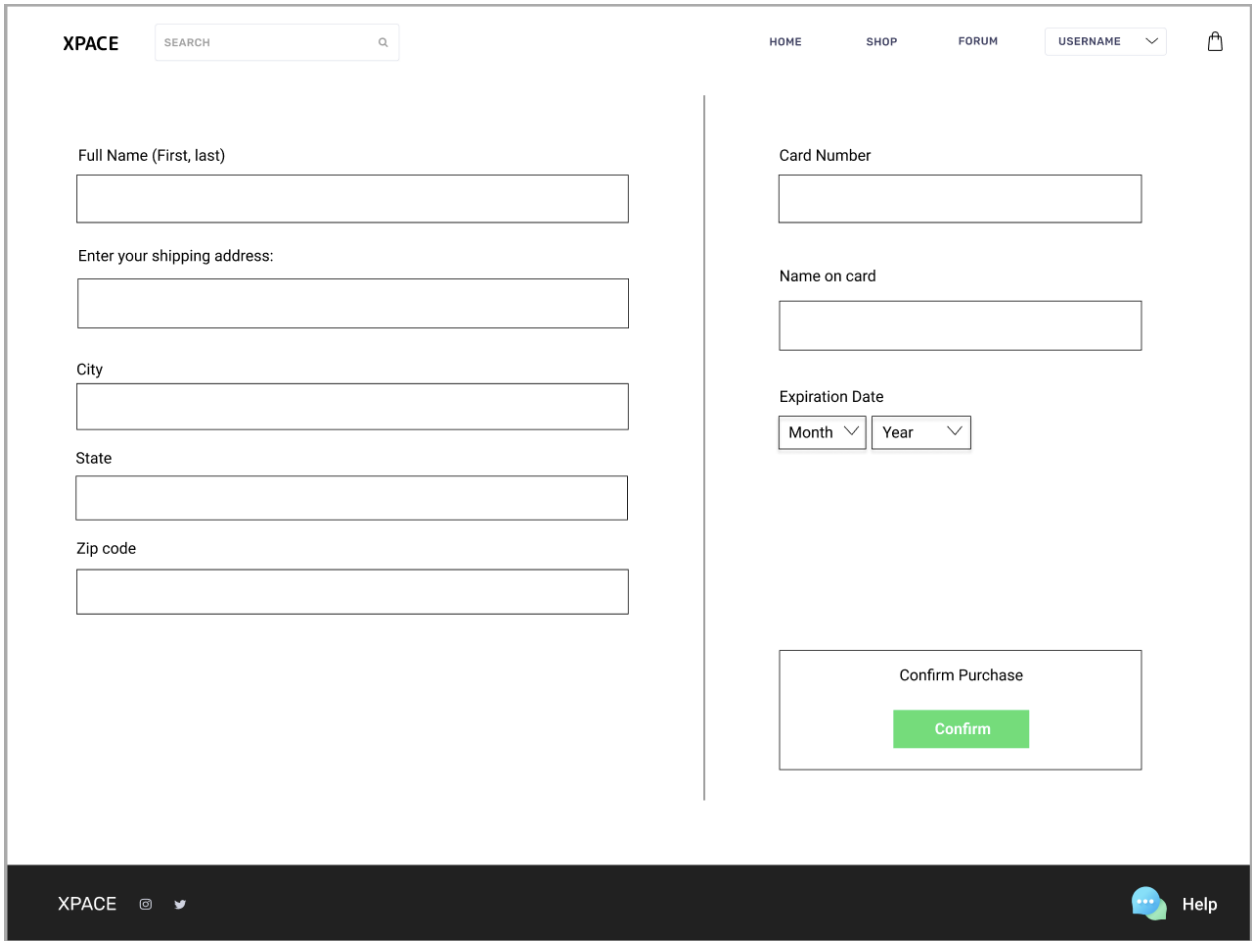
Discussion Forum GUI



Shopping Cart GUI

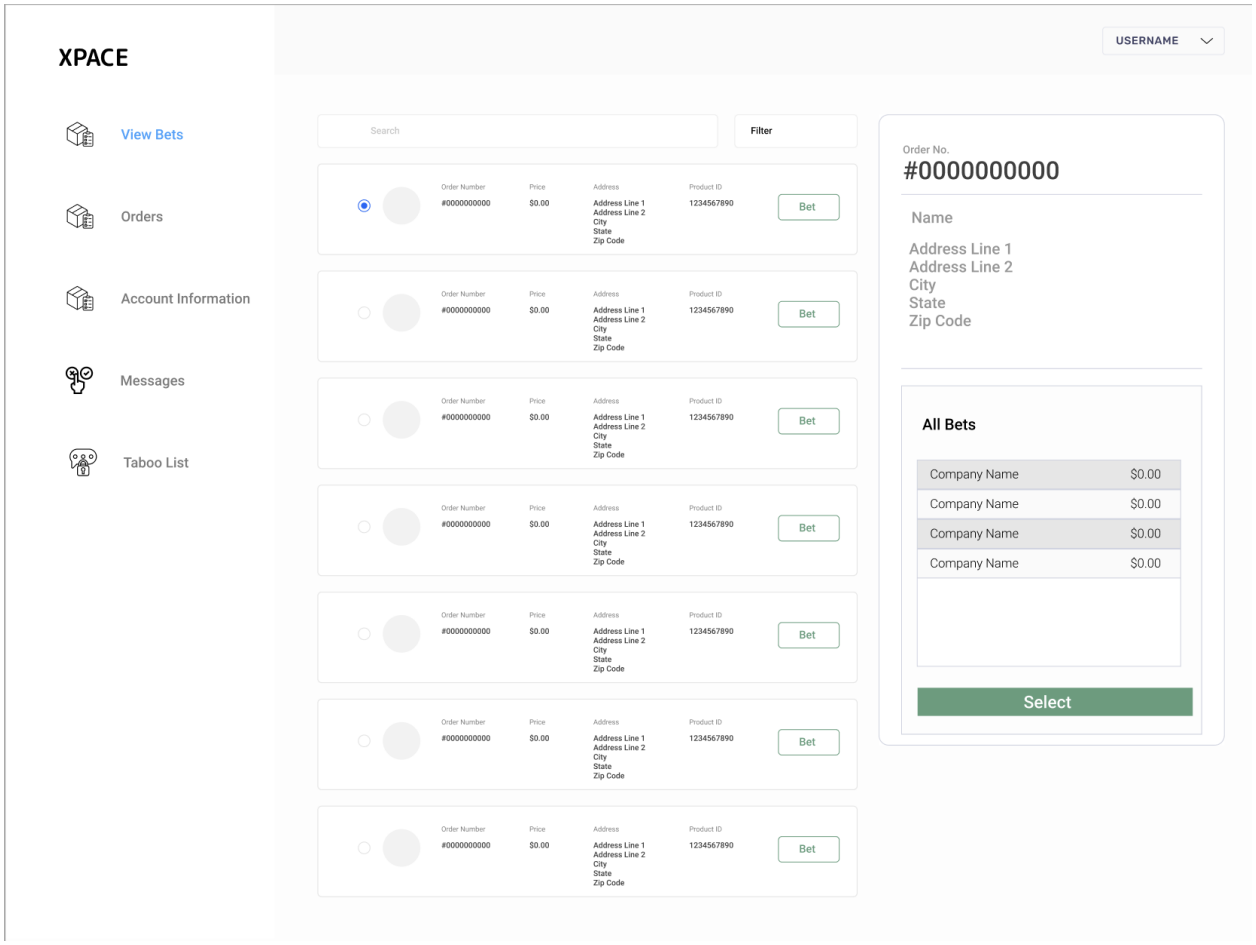


Checkout GUI

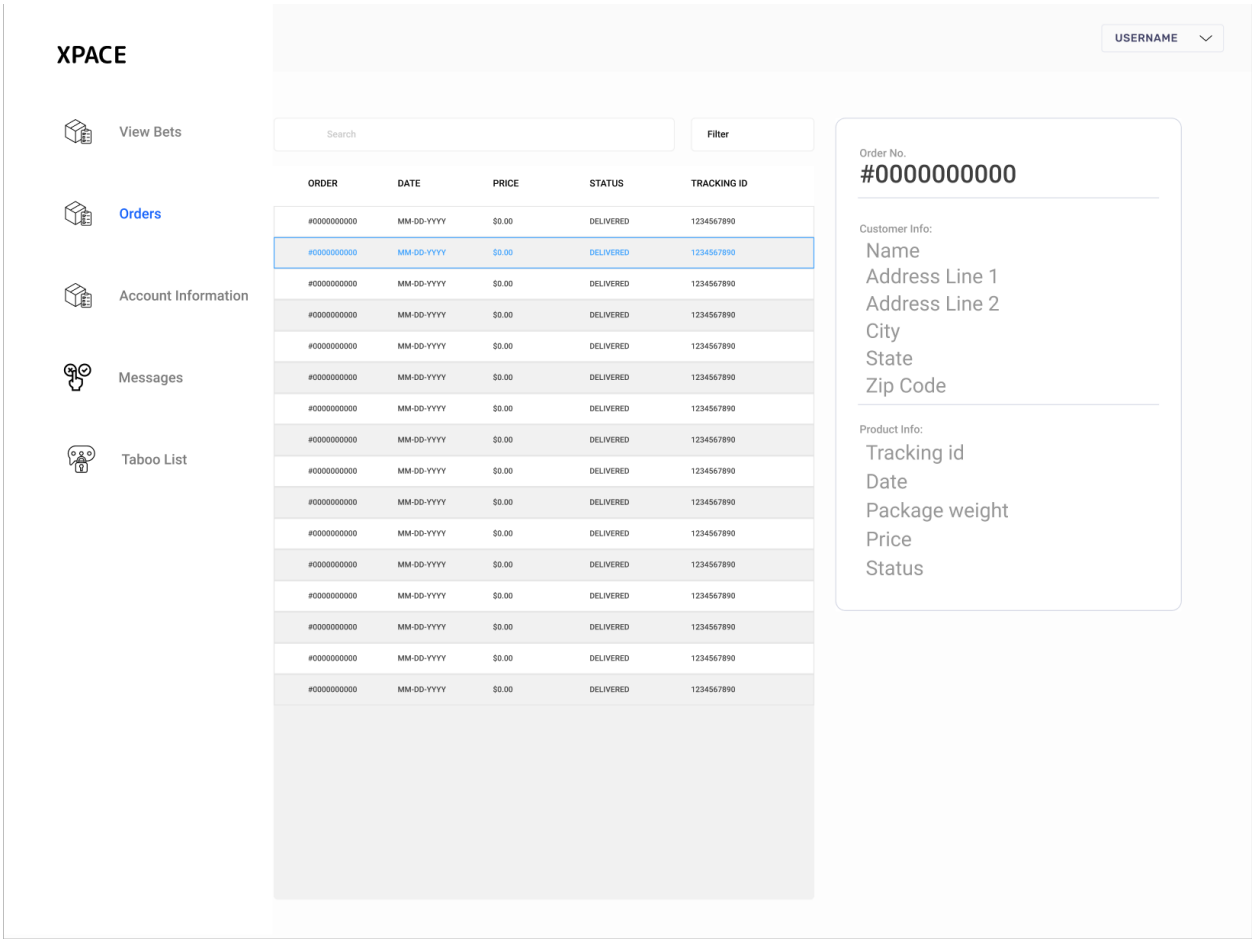


Store Clerk Dashboard GUI

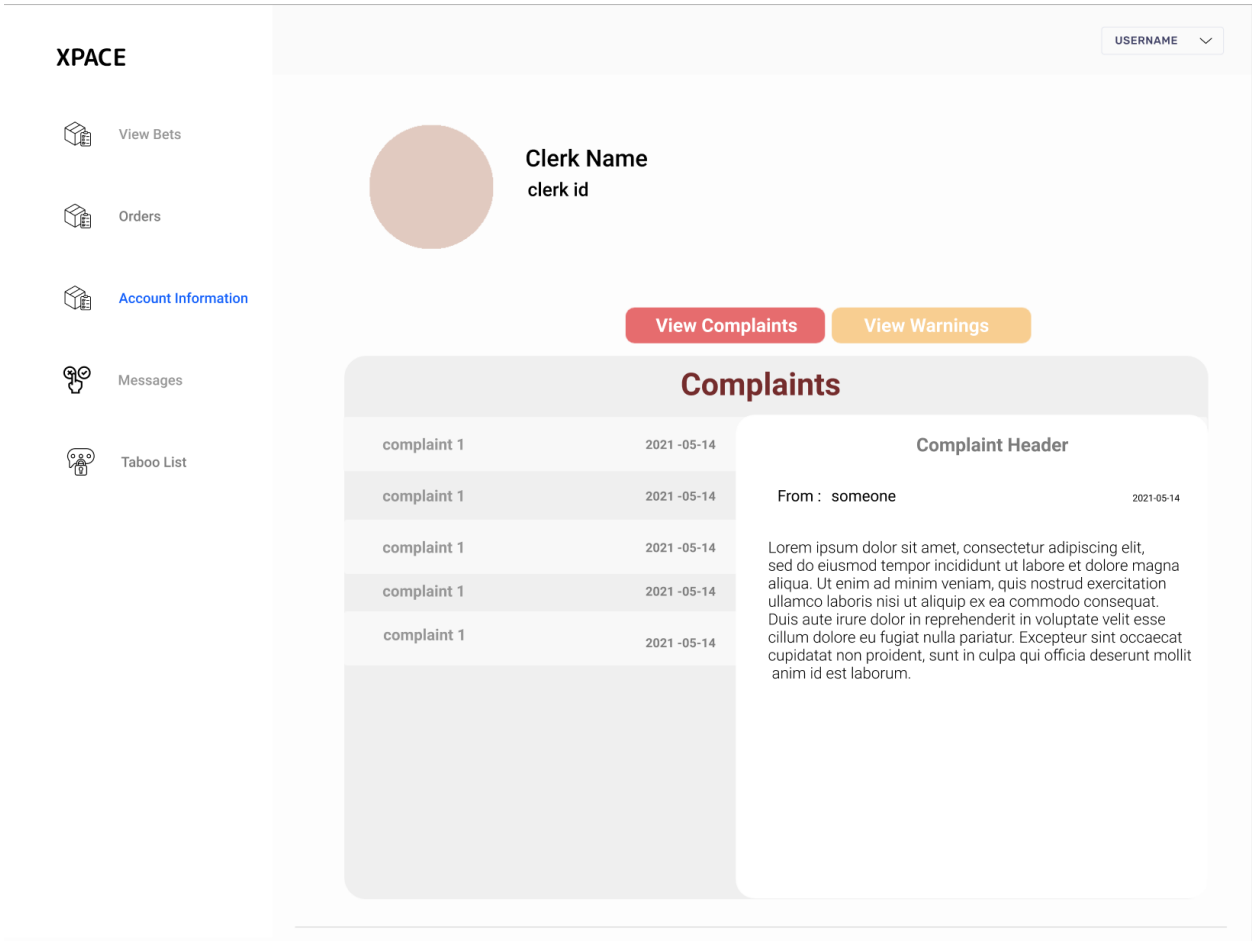
- View Bets



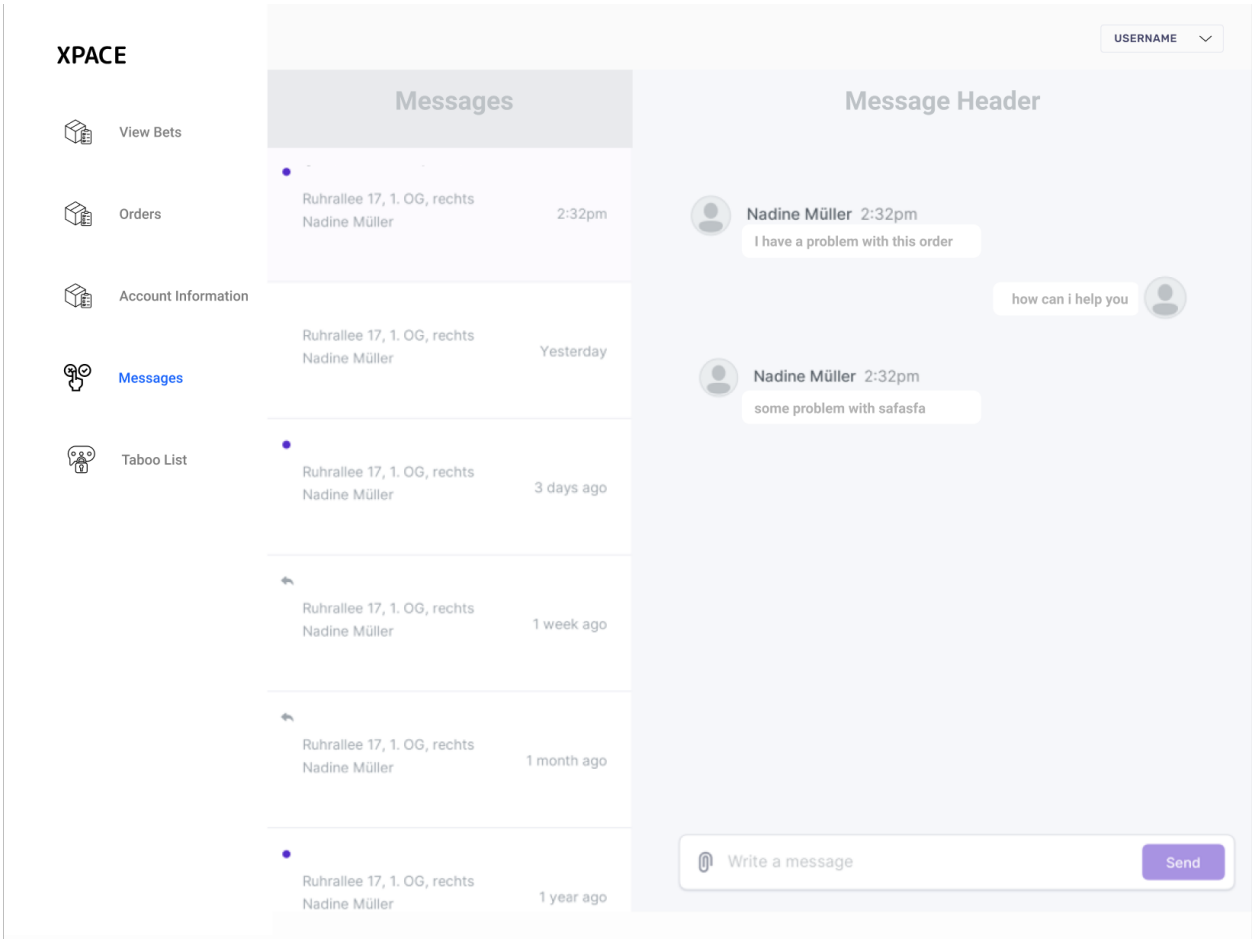
- View Order


XPACE
USERNAME
View Bets
Orders
Account Information
Messages
Taboo List
Search
Filter
ORDER
DATE
PRICE
STATUS
TRACKING ID
#0000000000
MM-DD-YYYY
$0.00
DELIVERED
1234567890
Order No.
#0000000000
Customer Info:
Name
Address Line 1
Address Line 2
City
State
Zip Code
Product Info:
Tracking id
Date
Package weight
Price
Status

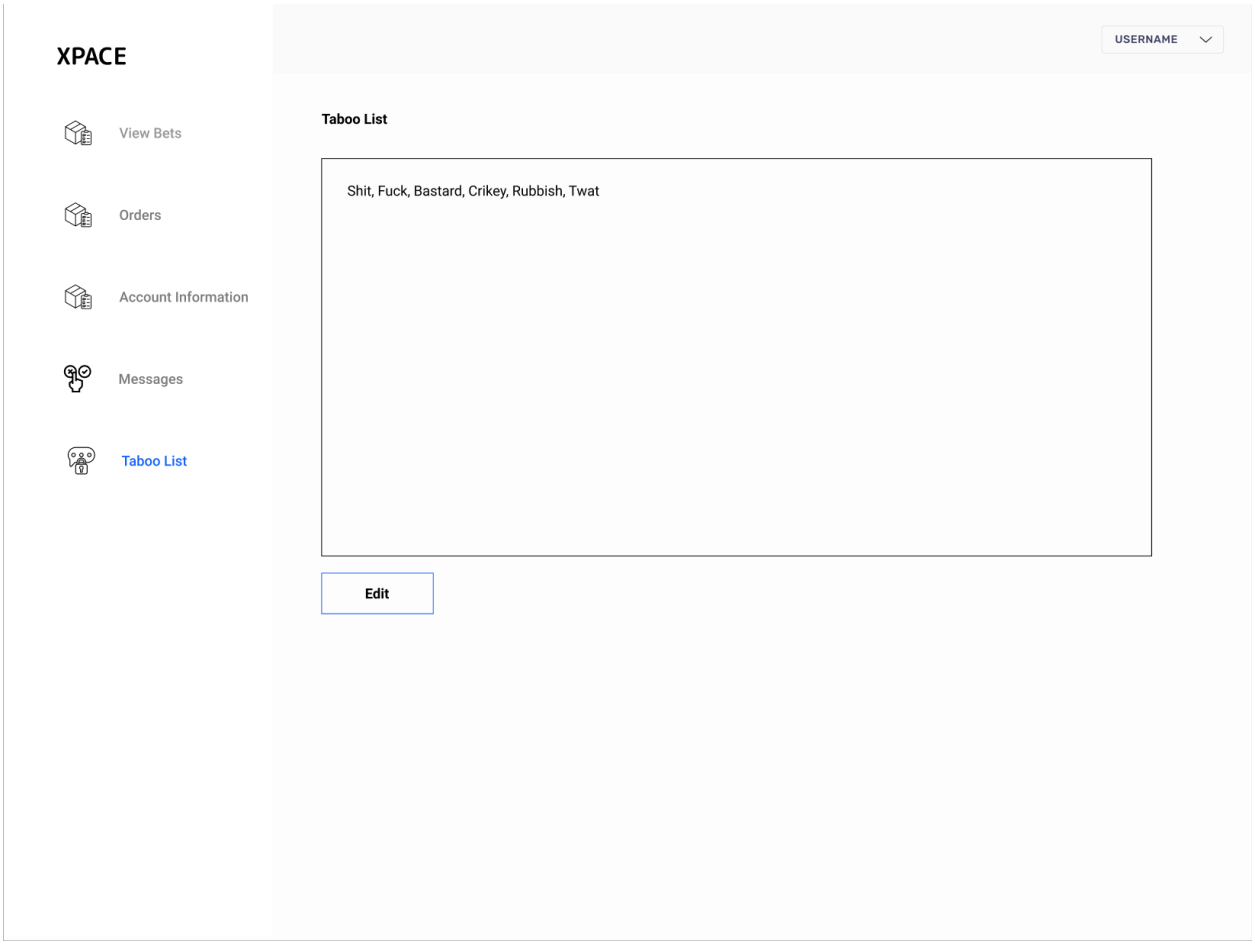
- Account Info



- Messages

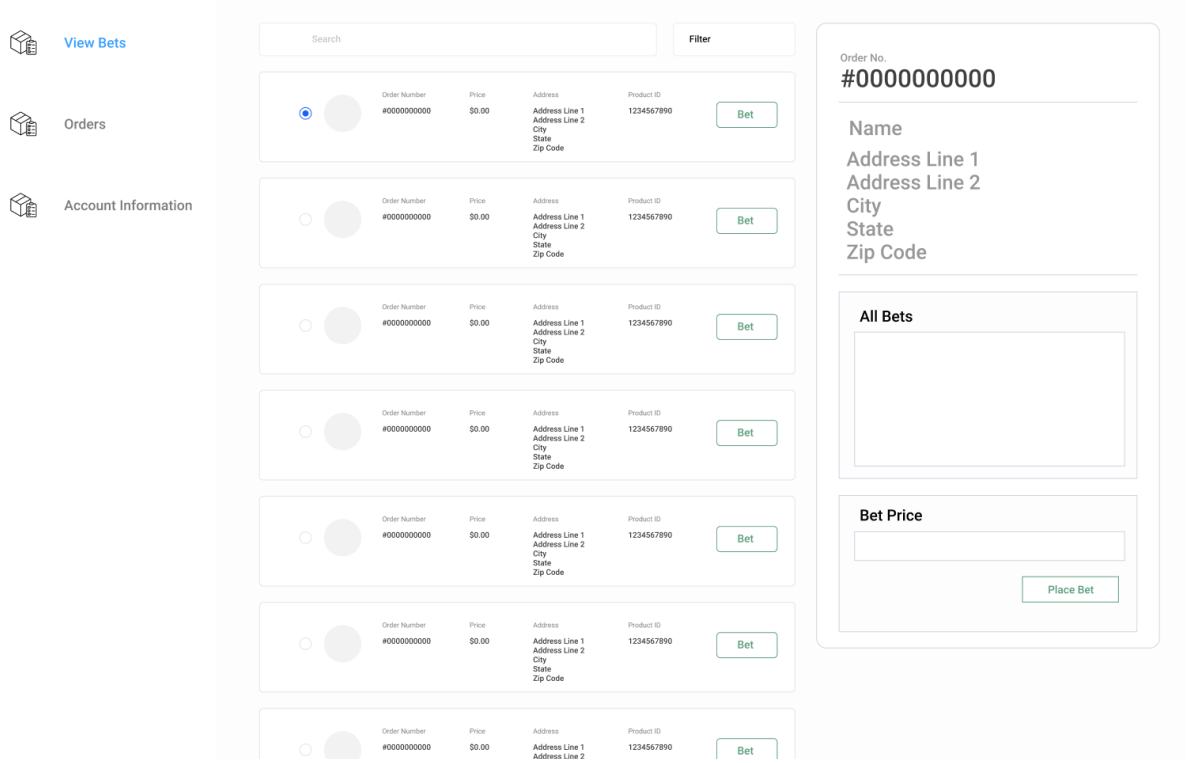

XPACE
View Bets
Orders
Account Information
Messages
Taboo List
USERNAME
Messages
Message Header
Ruhrallee 17, 1. OG, rechts
Nadine Müller
2:32pm
Yesterday
3 days ago
1 week ago
1 month ago
1 year ago
Nadine Müller 2:32pm
I have a problem with this order
how can i help you
Nadine Müller 2:32pm
some problem with safasfa
Write a message
Send

- Taboo List

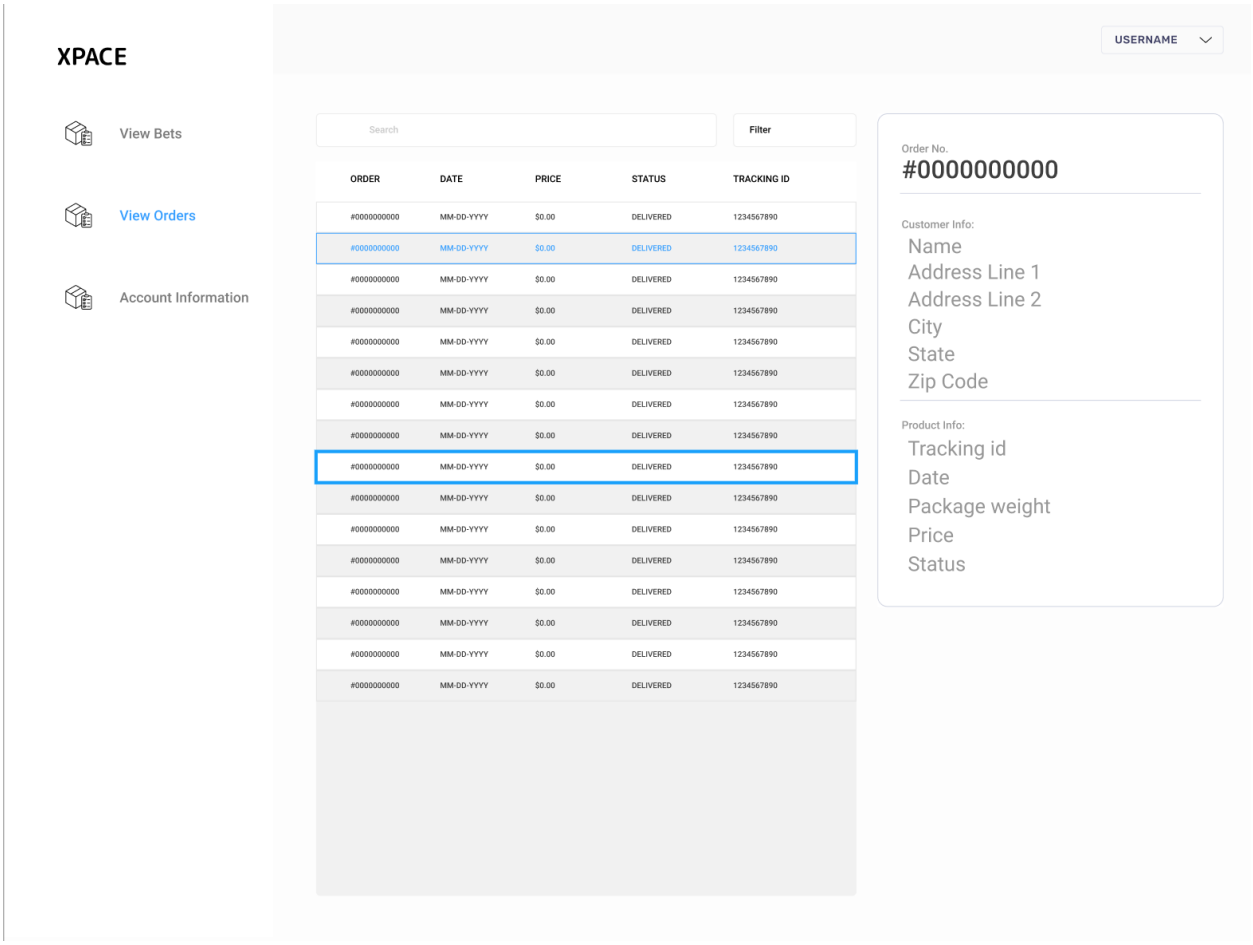

XPACE
USERNAME
View Bets
Orders
Account Information
Messages
Taboo List
Taboo List
Shit, Fuck, Bastard, Crikey, Rubbish, Twat
Edit

Delivery Company Dashboard GUI

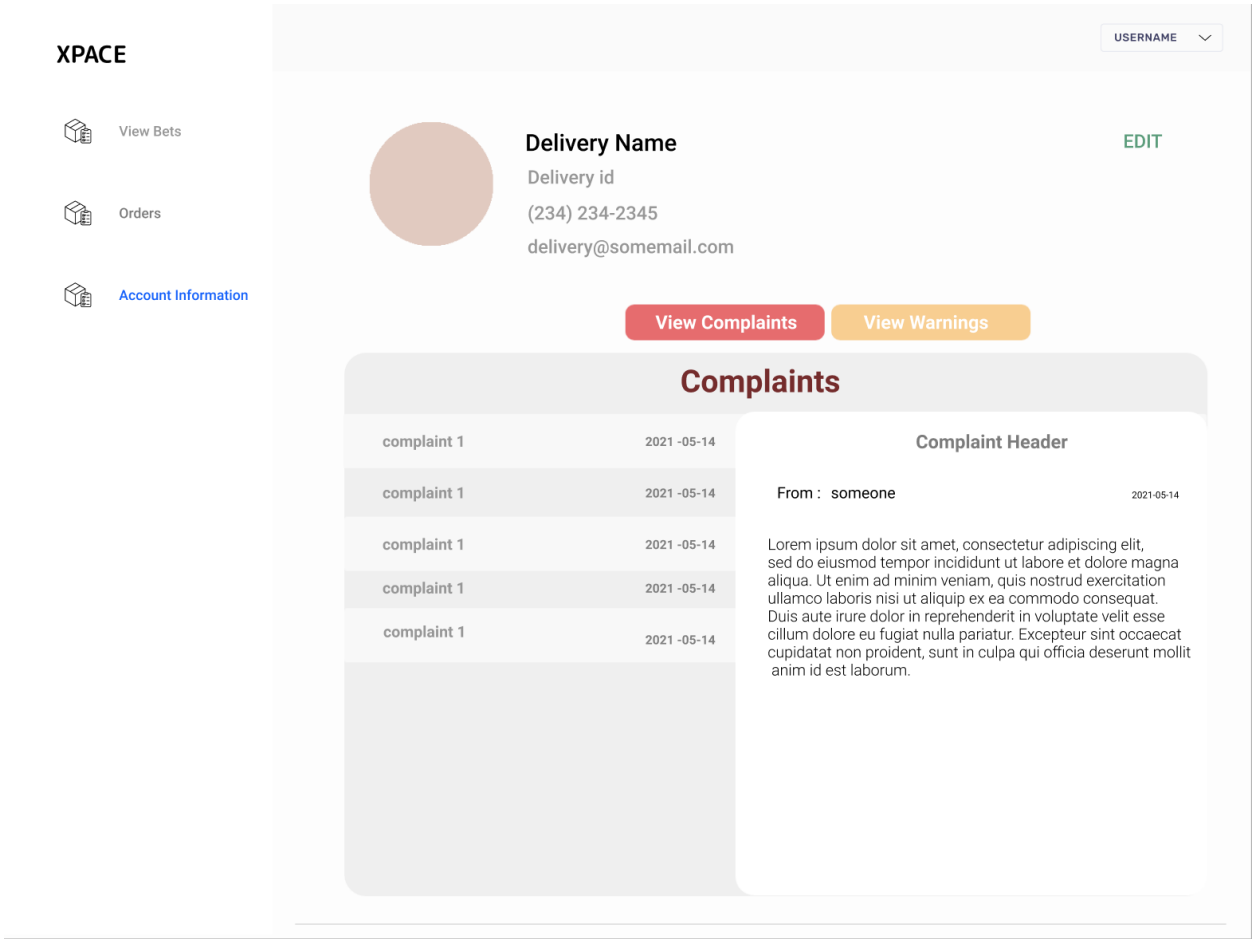
- View Bets



- View Order

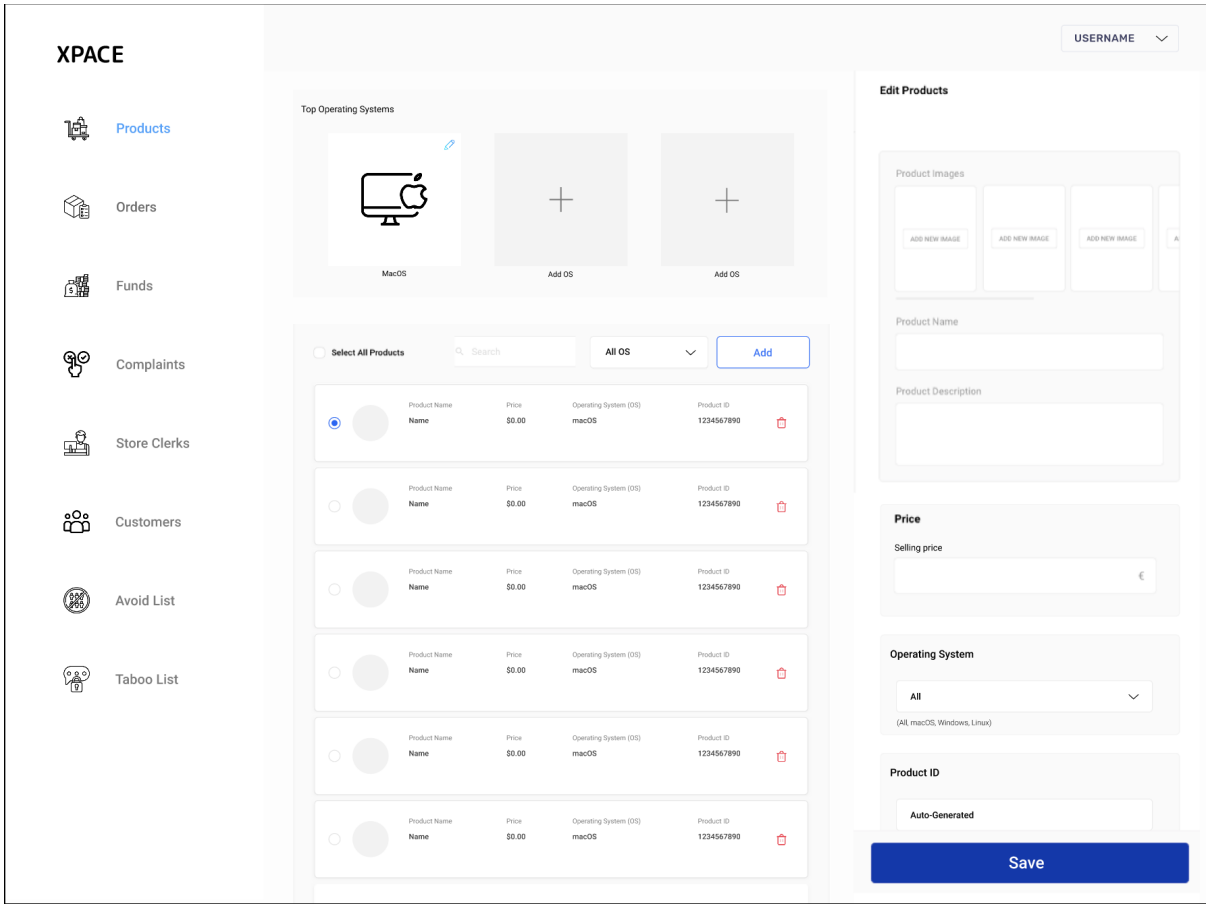

XPACE
USERNAME
View Bets
View Orders
Account Information
Search
Filter
ORDER
DATE
PRICE
STATUS
TRACKING ID
#0000000000
MM-DD-YYYY
$0.00
DELIVERED
1234567890
Order No.
#0000000000
Customer Info:
Name
Address Line 1
Address Line 2
City
State
Zip Code
Product Info:
Tracking id
Date
Package weight
Price
Status

- Account Info

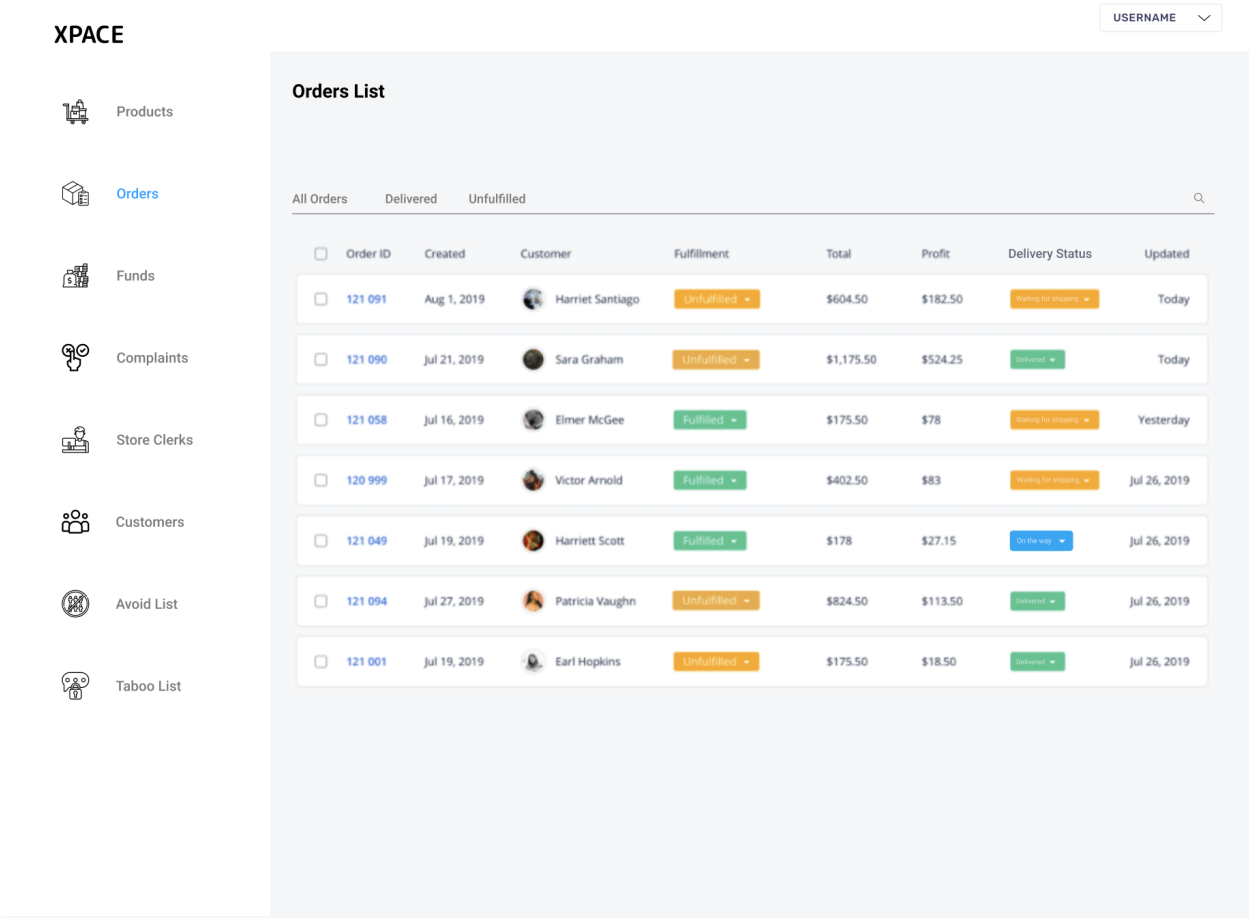


Manager Dashboard GUI

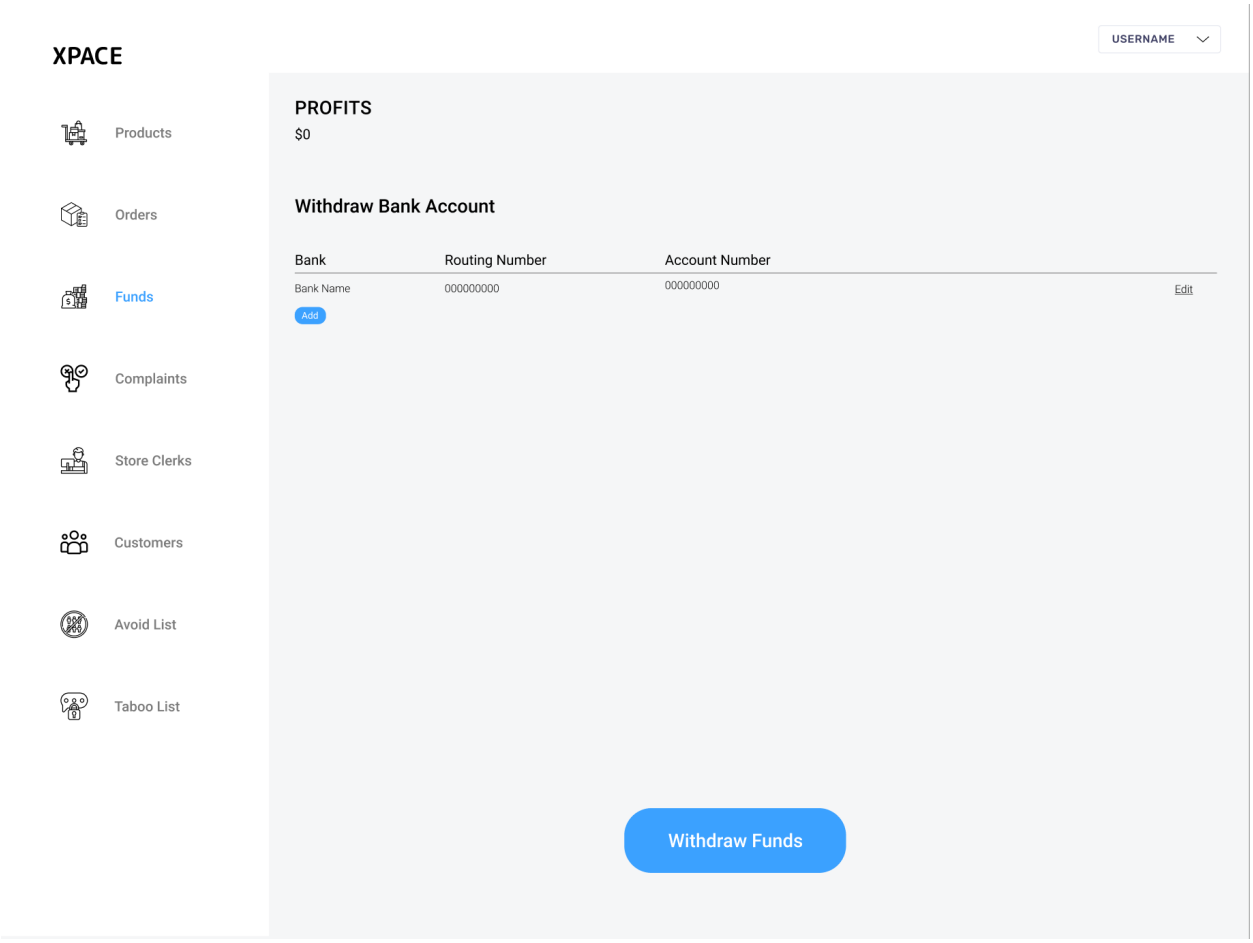
- Products



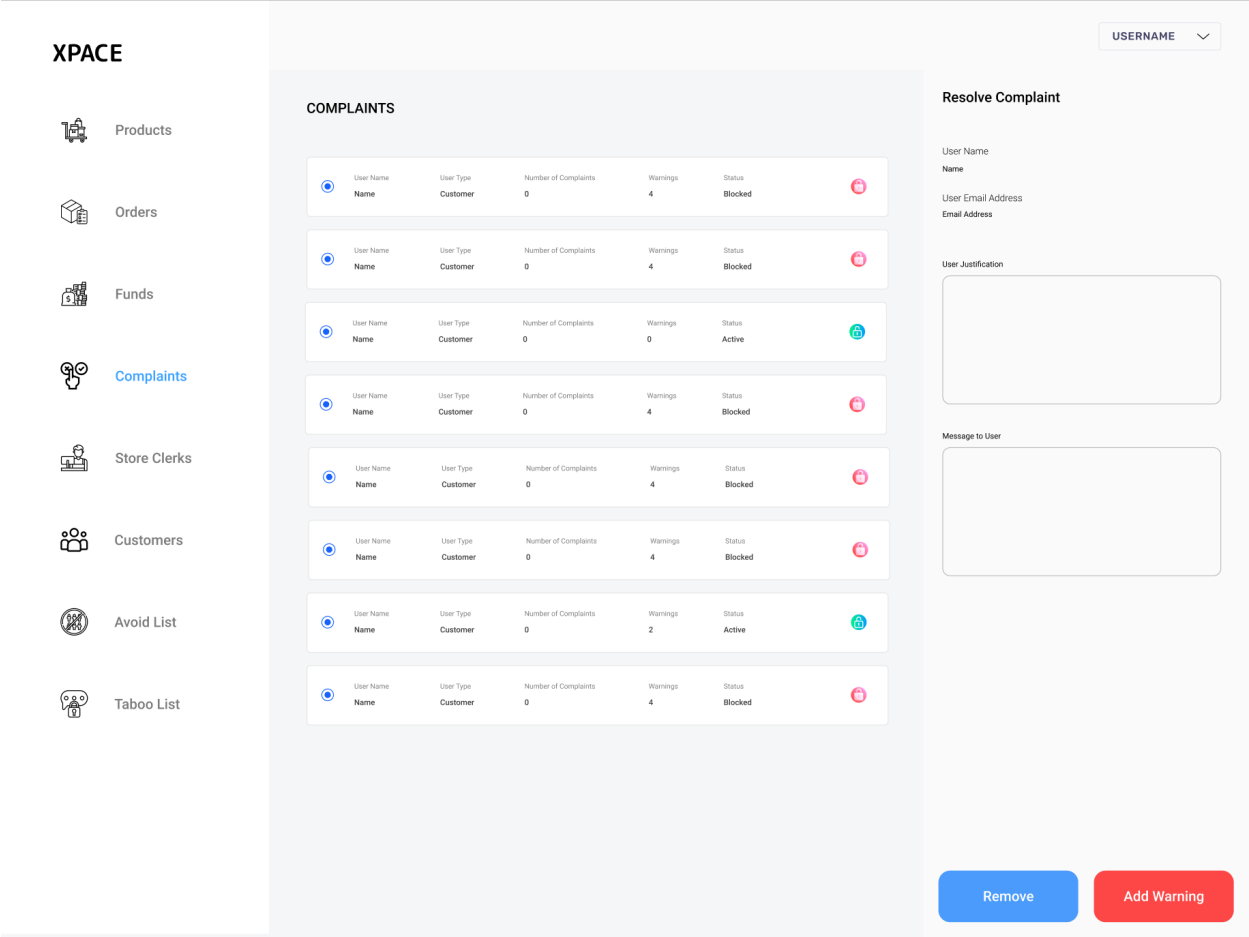
- Orders


XPACE
USERNAME
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
Orders List
All Orders
Delivered
Unfulfilled
Order ID
Created
Customer
Fulfillment
Total
Profit
Delivery Status
Updated
121 091
Aug 1, 2019
Harriet Santiago
Unfulfilled
$604.50
$182.50
Today
121 090
Jul 21, 2019
Sara Graham
Unfulfilled
$1,175.50
$524.25
Today
121 058
Jul 16, 2019
Elmer McGee
Fulfilled
$175.50
$78
Yesterday
120 999
Jul 17, 2019
Victor Arnold
Fulfilled
$402.50
$83
Jul 26, 2019
121 049
Jul 19, 2019
Harriett Scott
Fulfilled
$178
$27.15
Jul 26, 2019
121 094
Jul 27, 2019
Patricia Vaughn
Unfulfilled
$824.50
$113.50
Jul 26, 2019
121 001
Jul 19, 2019
Earl Hopkins
Unfulfilled
$175.50
$18.50
Jul 26, 2019

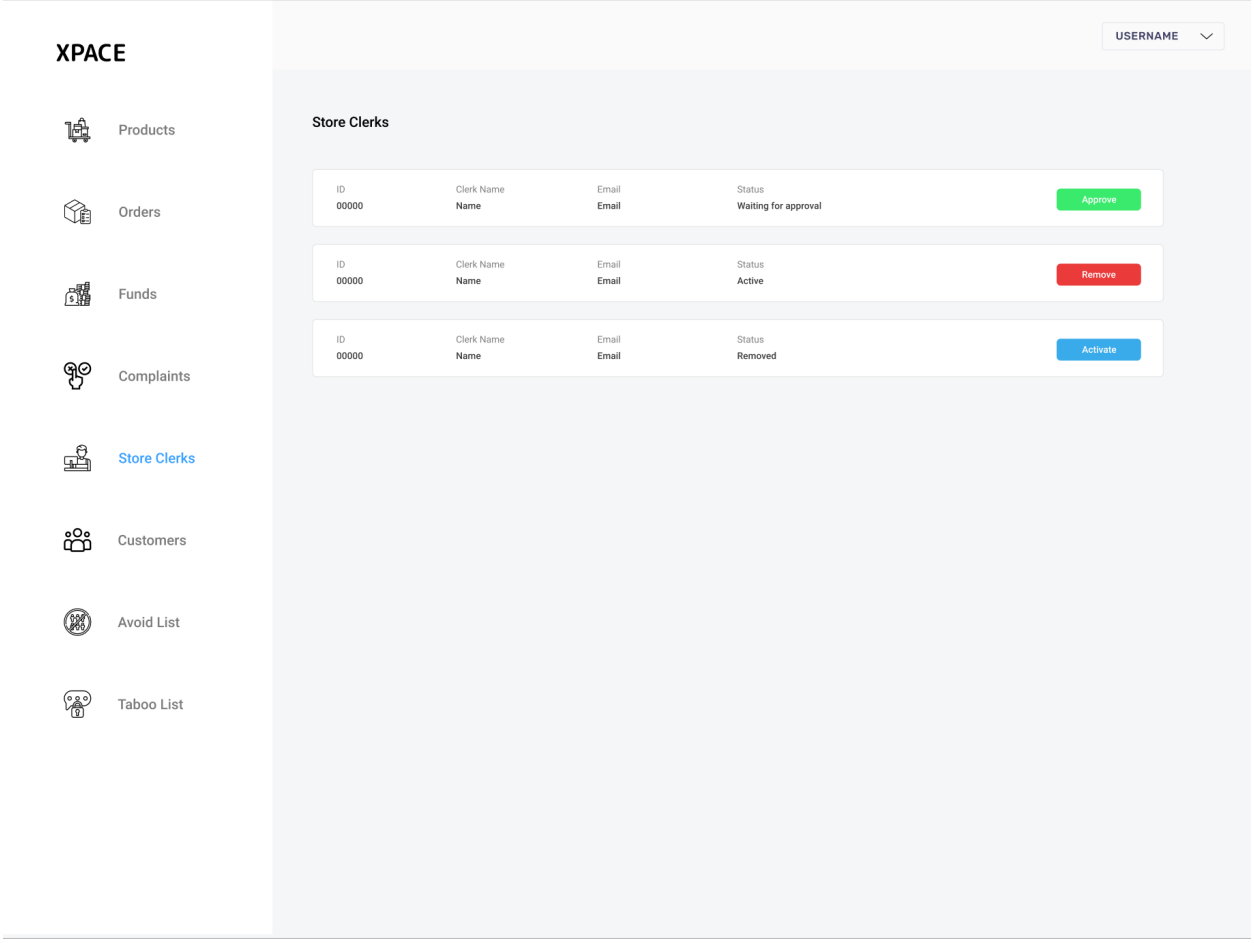
- Funds


XPACE
USERNAME
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
PROFITS
$0
Withdraw Bank Account
Bank
Routing Number
Account Number
Bank Name
000000000
000000000
Edit
Add
Withdraw Funds

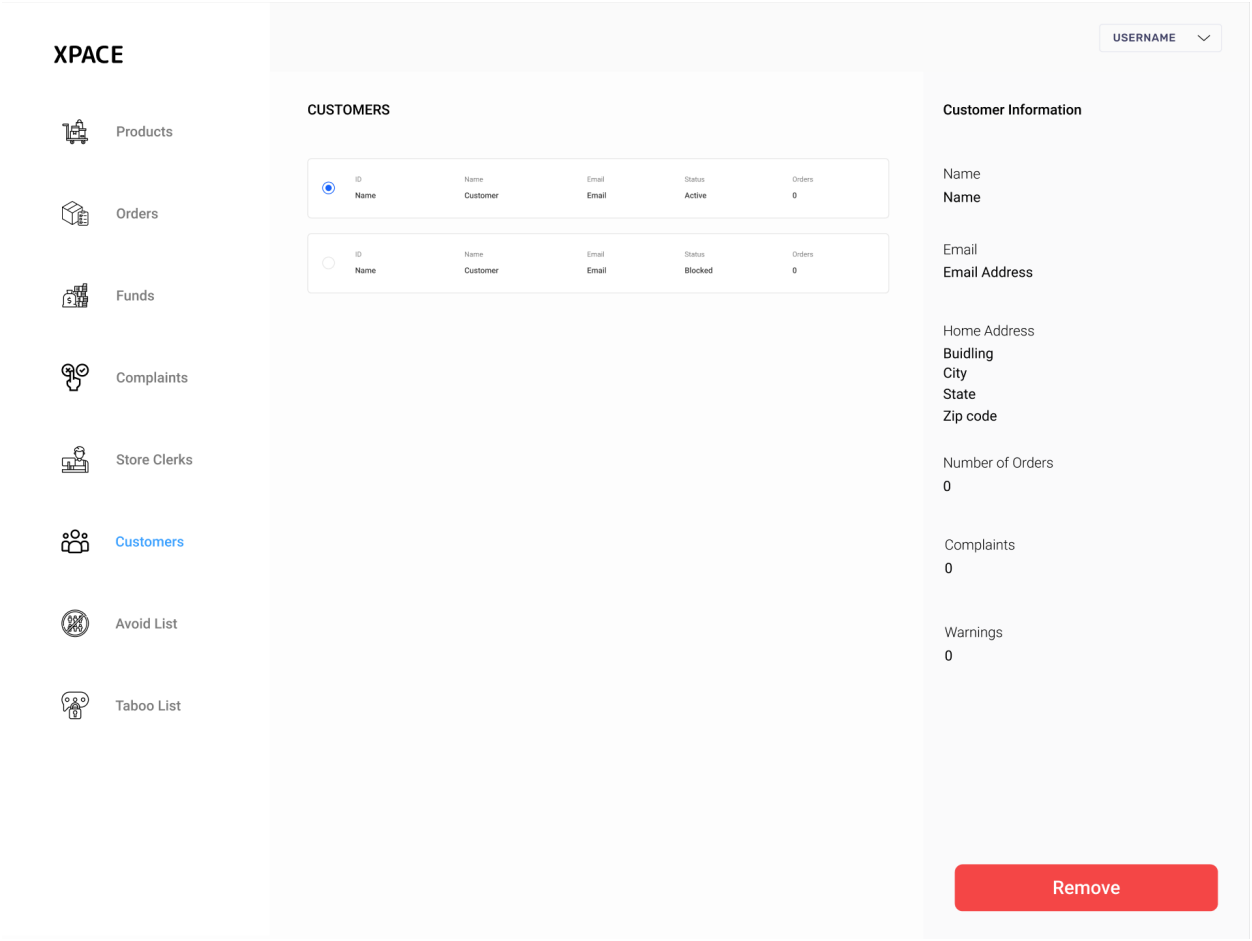
- Complaints


XPACE
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
USERNAME
COMPLAINTS
User Name
User Type
Number of Complaints
Warnings
Status
Name
Customer
0
4
Blocked
Active
2
Resolve Complaint
User Name
Name
User Email Address
Email Address
User Justification
Message to User
Remove
Add Warning

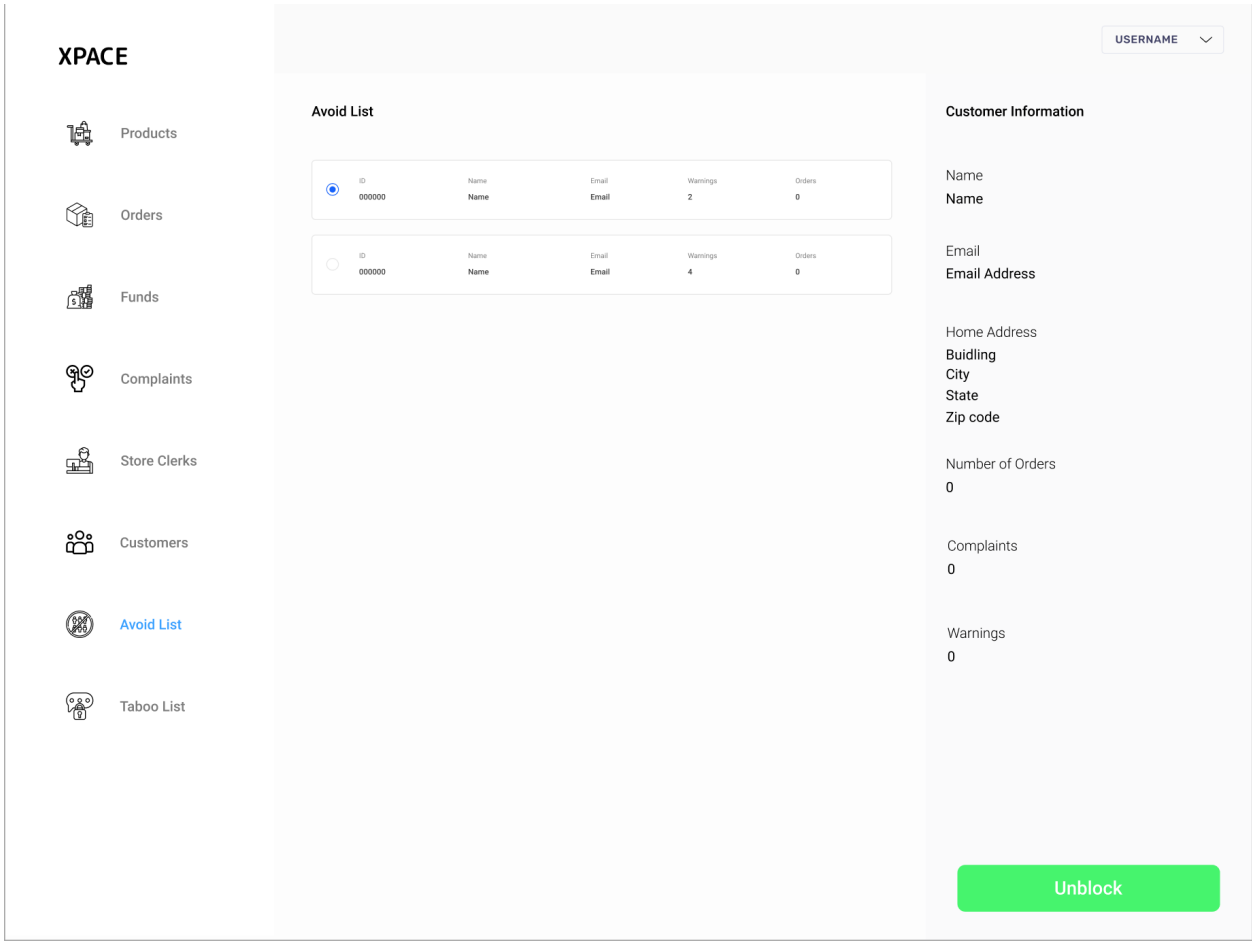
- Store Clerks



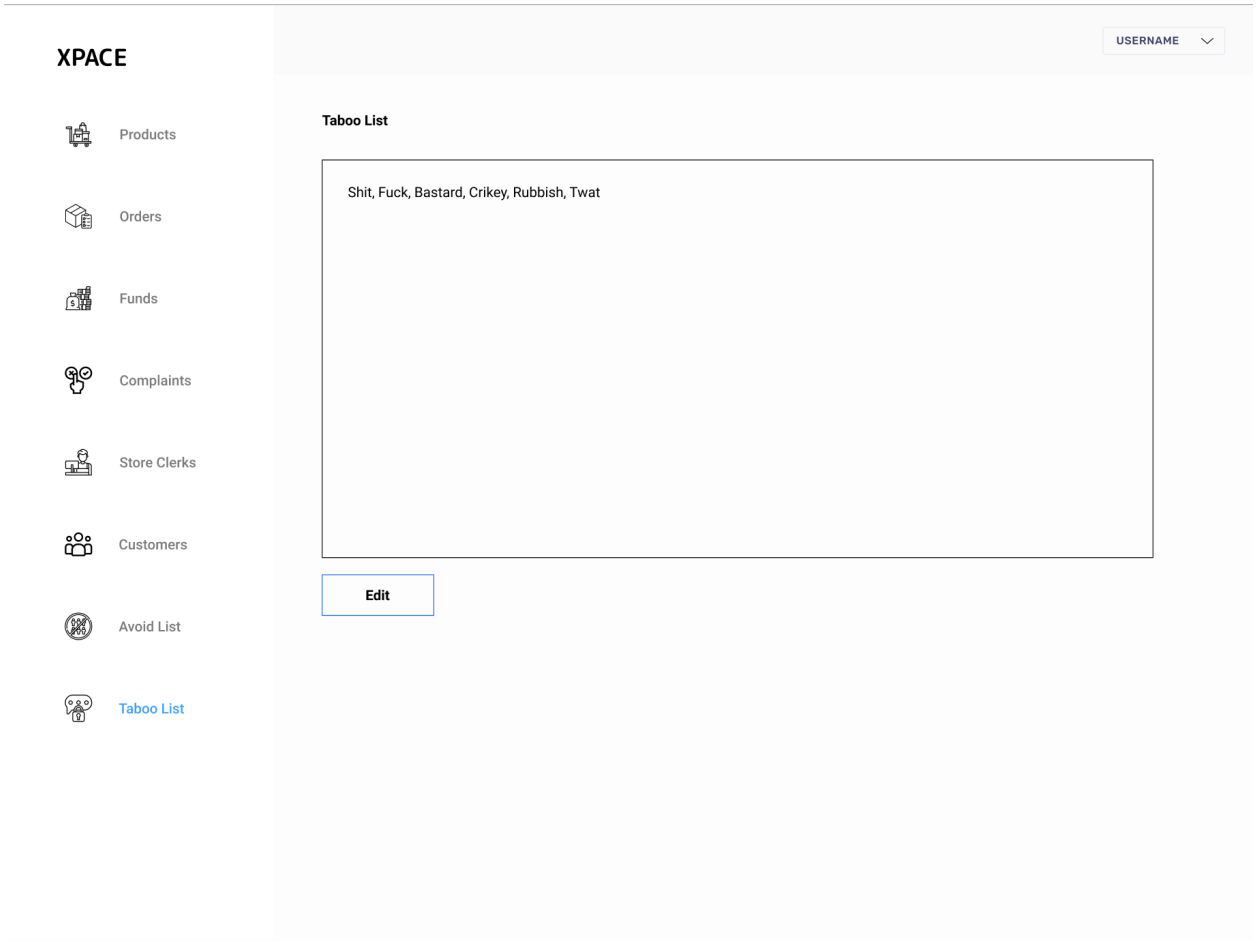
- Customers


XPACE
USERNAME
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
CUSTOMERS
ID Name Email Status Orders
Name Customer Email Active 0
ID Name Email Status Orders
Name Customer Email Blocked 0
Customer Information
Name
Name
Email
Email Address
Home Address
Buidling
City
State
Zip code
Number of Orders
0
Complaints
0
Warnings
0
Remove

- Avoid List

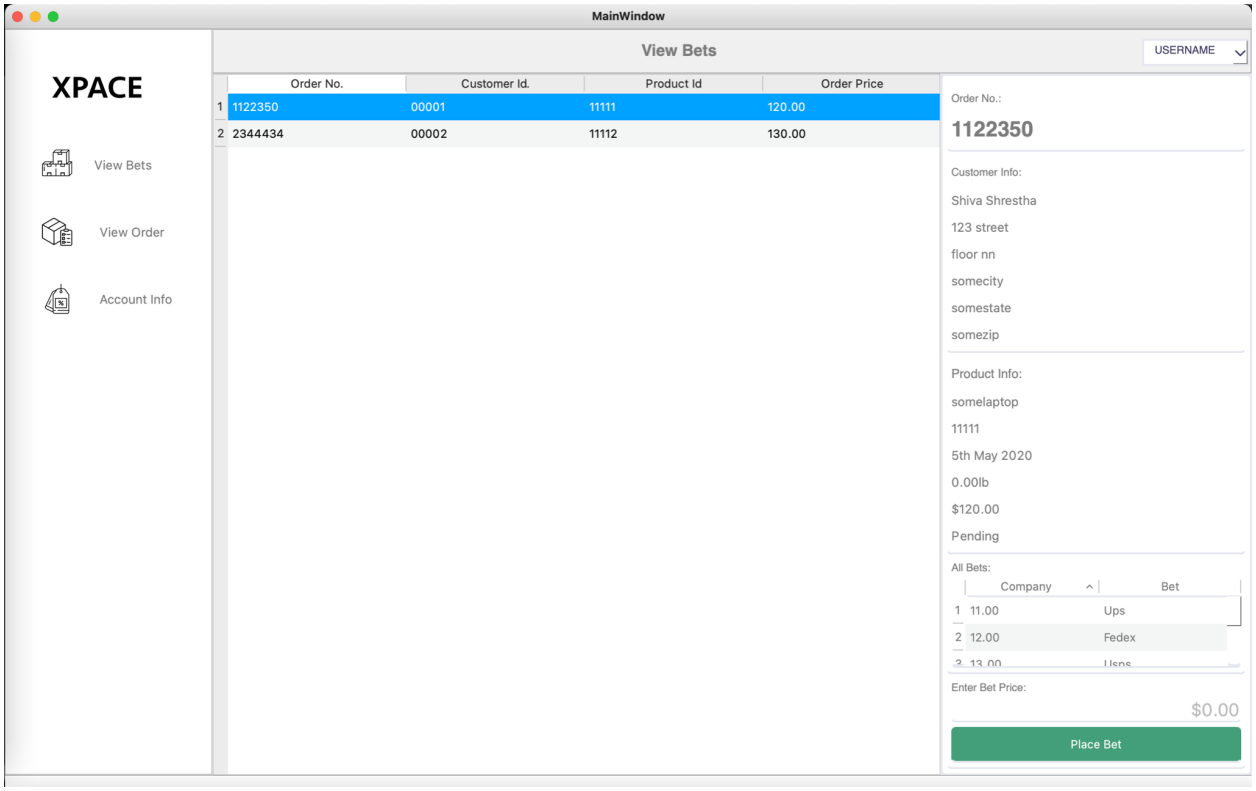

XPACE
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
USERNAME
Avoid List
ID
Name
Email
Warnings
Orders
000000
Name
Email
2
0
000000
Name
Email
4
0
Customer Information
Name
Name
Email
Email Address
Home Address
Buidling
City
State
Zip code
Number of Orders
0
Complaints
0
Warnings
0
Unblock

- Taboo List

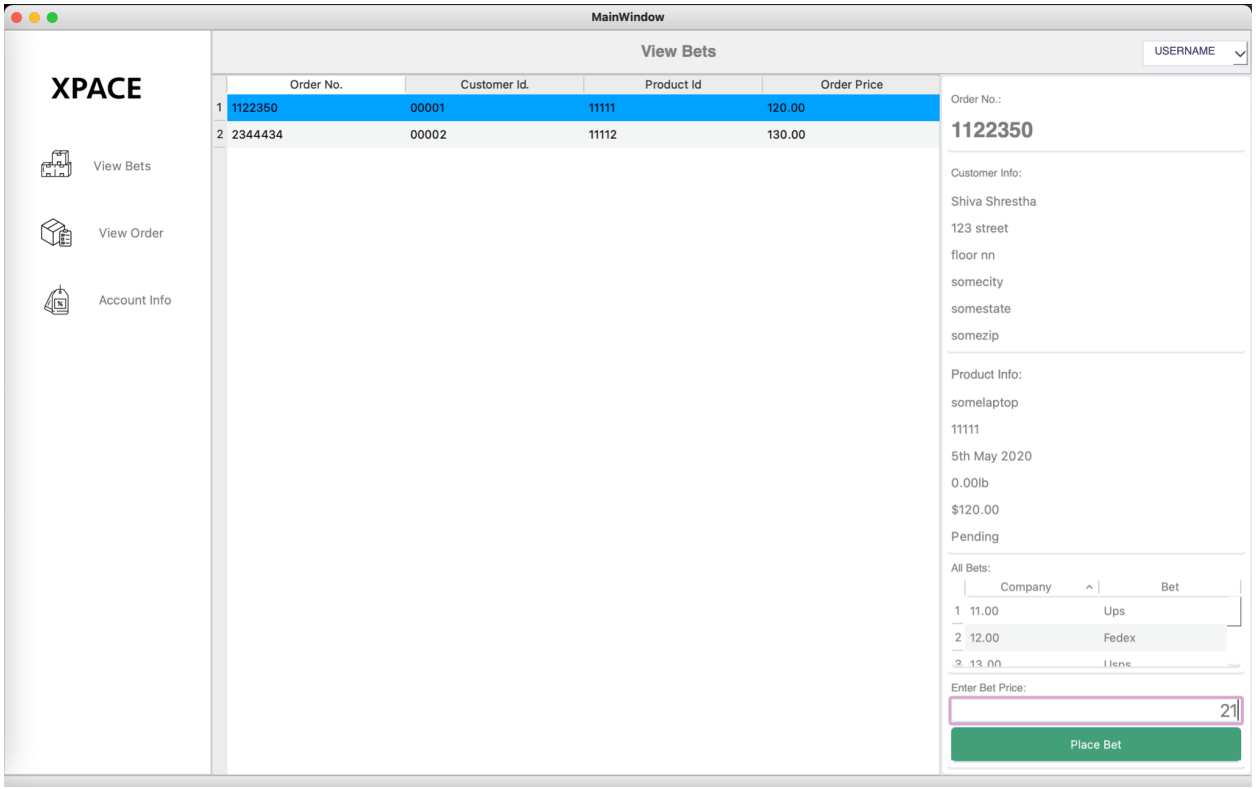

XPACE
USERNAME
Products
Orders
Funds
Complaints
Store Clerks
Customers
Avoid List
Taboo List
Taboo List
Shit, Fuck, Bastard, Crikey, Rubbish, Twat
Edit

Placing Bet Prototype

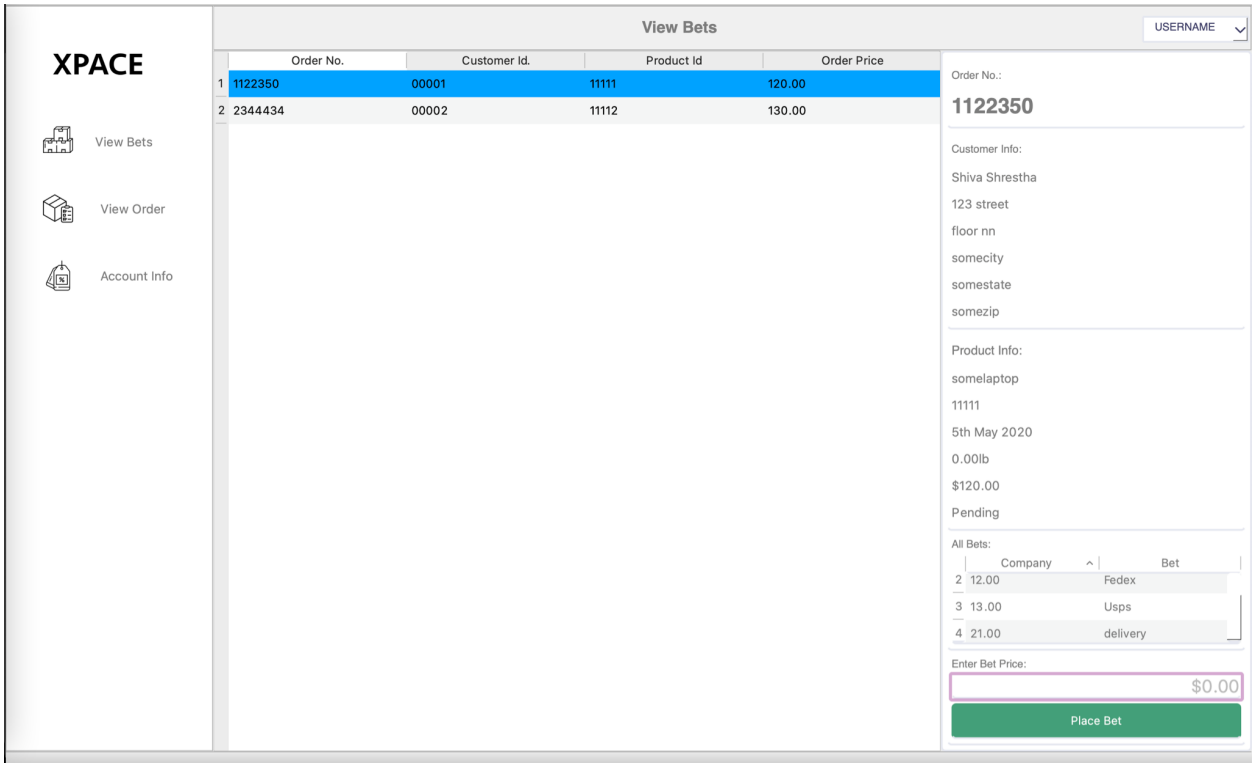
1. Select order to place bet



2. Write Delivery Price to bet



3. Bet placed successfully



Group meetings

Date	Time	Notes
03/18/21	1 hour 30 minutes	Analyze and break down work
03/23/21	1 hour 30 minutes	Work Revision and modifications
03/27/21	2 hours	Discuss user case models. Finishing phase 1 report
03/29/21	2 hours	Final adjustment and revisions for phase 1 report
04/07/21	1 hour 30 minutes	Divide application development
04/09/21	1 hour 30 minutes	Revision of prototype design
04/13/21	1 hour 30 minutes	Working on the implementation of the application front end
04/17/21	1 hour 30 minutes	Front end development and fixing encountered issues
04/21/21	1 hour 30 minutes	Follow up with team progress
04/24/21	1 hour	Revision of work done so far and planification of the phase 2 report
04/26/21	2 hours	Working on the project Phase 2 diagrams
04/27/21	2 hours	Final phase 2 revision and adjustments

Github (github):

https://github.com/CSC322-OnlineComputerStore/OnlineComputerStore.git