Top Shelf

Warehouse Inventory Management System

by
Justin Poblete
Juheng Mo
Luis Camacho
Calvin Nguyen
John Gardner

CPSC 362 - 07
Professor Ghadami
California State University of Fullerton

Table of Contents

Document Revision History Table

REVISION HISTORY TABLE				
Rev. #	**Description of Change**	**Page No.**	**Author**	**Date**
0	Document Creation		Calvin N	02/27/2020
1	User Manual Added	14-24	John G	3/1/2020
2	Corrections To Use-Cases	6-9	Justin P	3/21/2020
3	Project Plan 2nd/3rd Sprint	6	Justin P	5/10/2020
4	CRC Diagram/Cards	23-33	Justin P	5/10/2020
5	Development Process	22	Justin P	5/10/2020

Project Plan

Overview

TopShelf is an inventory management program that will help users organize the extensive list of items found in a distributor warehouse. Item information will be readily available with our intuitive and easy to use application.

Target

Products are first shipped from suppliers to warehouses. From those warehouses the items are then shipped to retailers where the end consumer can purchase the product. We plan on capitalizing on Distributors/Wholesalers by providing our program to enhance their inventory management.

Goals

Provide a program that effectively and efficiently manages items by:

- Adding, removing, and updating products
- Search product information
- Implementing an easy-to-use ordering system
- Secure account management
- Barcode utilization
- Adding, removing, and updating suppliers
- Shipment management
- Printable invoices
- Alerts

Demonstration of the product software will be available for testing within 4 weeks from the start of the project.

Cost

Not all warehouses are the same so estimations can be made based on the information provided by the client. A quote can be given upon contacting us. The average cost per square foot of warehouse space was $6.53. The average starting hourly rate of warehouse staff was $11.44, and the average annual pay for a warehouse management staff was $47,478. The average corporate profit came in at 8.83% This is according to a 2017 warehouse survey.

Project Members

- Leader: Justin Poblete - MySQL
- Juheng Mo - MySQL
- Luis Camacho - C#
- Calvin Nguyen - C#, MySQL
- John Gardner - C#, MySQL

Specifications

TopShelf will be made using C# connected with a SQL database.
The customer will need to first provide all product information including:

- Product
    - ProductId
    - Product Name
    - Product Brand
    - Quantity
    - Price

    - Expiration Date
    - Storage Location
- User Information
    - Username
    - Password
- Supplier Information
    - Supplier ID
    - SupplierName
    - Supplier Email
    - Supplier Phone Number
- Shipping Records
    - Shipping ID
    - Departure time
    - Expected Arrival Time
    - Supplier ID

2nd Sprint

Our main objective during our 1st sprint was establishing the base of our program. This included implementing a simple database with some search functionalities we thought warehouse employees would use regularly. For this sprint we added to the database with suppliers but also applied two new user groups. The administrator would be able to secure the system by managing the different types of users interacting with the program, and clients would now be able to order items. The team also thought it was a good idea to give separate users a different GUI to reduce the possibility of lower clearance users from messing with the system. Stockers should not be able to add and remove accounts. Another function we added was barcode searches. Barcodes are an integral part of identifying items in a warehouse.

3rd Sprint

The primary focus of the final sprint was continuing to build upon our ordering system. Previously orders were simply records that kept track of when a client had ordered something, but not what they specifically ordered. The client form was also pretty bare. This time around we managed to include order histories available to both the clients and clerks from their respective forms. From their order forms an invoice can be printed which includes the total cost, tracking information, as well as an itemized list of all the products that were purchased. Clients could now update their personal information within their new account page too. Alerts were also added to notify the user of new orders, low inventory, and expired items.

Functional User Stories

1. As an HR Manager, I want users to login so we can prevent unauthorized users from accessing our information.
2. As an order selector, I want to look up current stock for various products so that I can adjust order quantities for future orders.
3. As a package handler, I want to quickly lookup item locations so that I can fulfill orders efficiently.
4. As a warehouse clerk, I want to be able to add, delete, and update inventory information so that everything will be up to date.
5. As a warehouse clerk, I want to look up the estimated arrival time so that I can properly handle incoming shipments.
6. As a customer/client I want to order from the distributor so I can manage my own stock.
7. As an administrator/HR, I want to create and delete accounts so I can manage employees and client access.
8. As a clerk, I want to be able to add shipment so I can keep track of the orders being sent to us from the supplier.
9. As a stocker, I want to print and read barcodes so I can identify products more efficiently.
10. As a clerk I want to be able to add, delete, and update the suppliers so that I can manage the suppliers that send items to our warehouse.
11. As a client, I want to view my old orders, so I can confirm items that I have purchased
12. As a clerk, I want to have invoices of orders, so I can keep track of shipments and finances.
13. As a client, I want to be able to view and update my personal information, so that I can verify my own account.
14. As a clerk, I want to view and update a new order so I can verify the client's orders and provide information about how the order will be delivered.
15. As a clerk, I want to be notified when products in the warehouse are close to expiring or if there are too few of them, so that I can properly update the inventory.

Non-Functional User Stories

1. As a clerk, I want to be able to maximize and adjust the program window size so that I can view the information better.
2. As a clerk , I want to have a drop down menu of suppliers so that I can more easily identify the suppliers.

3. As any user, I want to be able to hide my password while logging in so that I can prevent unauthorized users from getting my credentials.
4. As a clerk, I want to be able to have a dark mode of the program so that my vision isn't strained during the long hours of working.
5. As a clerk, I want new item identification numbers to auto increment so I don't have to remember which identification numbers are taken.

Use Cases

Use Case: User Login

ID: UC-01

Description:
The user input their designated username and password to access the system.
Primary Actor:
Client, Stocker, Admin, and Clerk
Pre-conditions:
Must have an account.
Post Conditions:
Success end condition:
User logs in and is able to use the program.
Failure end condition:
User is prompted that the username or password is incorrect.
Main Success Scenario:

1. System Displays Login Window on Startup.
2. User enters their username.
3. User enters their password.
4. User clicks on the Login button.
5. System authenticates username and password.

Use Case: Insert Stock

ID: UC-02

Description:
Clerk inputs items into the database.
Primary Actor:
Clerk
Pre-conditions:
Clerk is logged in to the system.

Post-conditions:
Success End Condition:
Item is added to the database
Failure End Condition:
Clerk inputs the wrong or incomplete information and the item is not added.
Main Success Scenario:

1. User clicks “INSERT” button
2. System displays Insert form
3. User enters ProductID
4. User enters ProductBrand
5. User enters ProductName
6. User enters Price of item
7. User enters Quantity of stock
8. User enters Expiration Date of product
9. User enters ProductLocation to be stored in warehouse
10. User enters SupplierID of the Supplier
11. User clicks “INSERT” button
12. System processes input data

Use Case: Search Item Stock

ID: UC-03

Description:
Stocker searches the amount of an item in the warehouse.
Primary Actor:
Stocker
Pre-conditions:
Stocker must be logged in to the system.
Post-conditions:
Success end condition:
Items are found and the amount is displayed to the user.
Failure end condition:
Stocker input an item that was not in the database and nothing is displayed
Main Success Scenario:

1. User clicks on “Search Product Stock” Button
2. User clicks on Product Name drop down menu
3. User clicks Product they want to look up
4. System processes input
5. System displays a pop-up window with stock of the item.
6. User clicks OK to close the window.

Use case: Update item information

ID: UC-04

Description:
Clerk wants to update the information of an item that was misinputed or has changed.
Primary Actor:
Clerk
Pre-conditions:
Clerk must be logged in to the system.
Post-conditions:
Success end condition:
Item is updated with the new correct information.
Failure end condition:
Item was not found in the database and not updated.
Main Success Scenario:

1. User clicks on Update in drop down menu
2. User clicks on Update Product
3. User enters Product ID for product to be edited
4. User enters information into desired field for edited
5. User clicks Update to modify
6. System processes input information
7. System displays pop-up window with new information
8. User presses OK to confirm edit
9. System modifies data inside the database

Use Case: Delete Item

ID: UC-05

Description:
Clerk wants to delete an item that is no longer available in the warehouse.
Primary Actor:
Clerk
Pre-conditions:
Clerk must be logged in to the system.
Post-conditions:
Success end condition:
Item is deleted from the database.

Failure end condition:
Item was not found in the database and not deleted.

Main Success Scenario:

1. User clicks on Delete in drop down menu
2. System displays delete features menu
3. User clicks on Delete Product
4. System displays pop-up window with a product list to be selected
5. User clicks on Product Name drop down menu
6. User selects which Product to delete
7. User clicks Delete
8. System processes input information
9. System displays pop-up window with new information
10. User presses OK to confirm deletion
11. System modifies data inside the database

Use Case: Order Item

ID: UC-06

Description:
Customer adds items to a cart and creates an order. The order is then stored until it is verified and processed.

Primary Actor:
Customer

Pre-conditions:
Customer must be logged in to the system.

Post-conditions:
Success end condition:
Order is created to later be processed by a clerk.
Failure end condition:
Customer is unable to create an order.

Main Success Scenario:

1. Customer must be login first
2. System displays order form
3. User selects quantity of a product from a drop down box
4. User clicks “Order Now” button
5. System processes the order

Use Case: Create Account

ID: UC-07

Description:
the admin creates a account in the system
Primary Actor:
Admin
Pre-conditions:
Admin must be logged in to system
Post-conditions:
Success end condition:
Admin creates an account for clerk,customer,stocker or admin
Failure end condition:
Admin is unable to create an account
Main Success Scenario:

1. User goes to “Admin” drop down box
2. User clicks “Create Account” button
3. System displays create account form
4. User enters username
5. User enters password
6. User enters group from the drop down menu
7. User clicks “Create” button
8. System creates a new entry in the database

Extension:

For step 7: if the username exists in the system

1. System will not allow the creation of the account
2. System will display a message that username already exists
3. User must input a unique username

Use Case: Insert Shipment

ID: UC-08

Description:
When the warehouse clerk makes orders with the supplier, he or she needs to be able to update shipments in the system. The clerk logs in and inserts shipment information: what the order contains, who the supplier is, and the expected arrival time
Primary Actor:
Warehouse Clerk
Pre-conditions:

Clerk must be logged in to system

Post-conditions:

Success end condition:

Clerk has entered shipment information and database is updated

Failure end condition:

Clerk does not have accurate or complete information (maybe the supplier does not have accurate shipment dates).

Main Success Scenario:

1. User login to system.
2. User clicks on the “Insert” drop-down menu.
3. User clicks on the “Insert Shipping Information” button.
4. System displays insert shipping form
5. User enters Departure Date
6. User enters Arrival Date
7. User selects the supplier from the drop-down menu.
8. User selects the quantity of the products from the drop-down menu from the table of products' name.
9. User clicks “Insert” button
10. System creates a new entry in the database

Extension:

For step 7:

1. If the supplier does not exist, it must be entered separately.

For step 8:

1. If a product does not exist, it must be entered separately.

Use Case: Search Barcode

ID: UC-09

Description:

Stocker scans barcodes of products in stock.

Primary Actor:

Stocker

Pre-conditions:

Barcode for item has been generated upon item entry into the database, stocker is logged in.

Post-conditions:

Success end condition:

Barcode is scanned and the system is updated.

Failure end condition:

Barcode is not scanned properly, system is not updated.

Main Success Scenario:

1. Stocker login to system.
2. User clicks the “Product Barcode” button.
3. User clicks the drop-down menu for Select Product.
4. User selects a product in stock.
5. User clicks the “View” button.
6. System displays selected product’s barcode.

Extension:

Stocker can also scan outgoing items.

1. Stocker clicks the “scan” drop down menu.
2. User clicks the “outgoing” button.
3. User scans the item being shipped out.
4. System stock is updated for that item.

Use Case: Insert Supplier

ID: UC-10

Description:

The Clerk inserts a new supplier that is now doing business with the warehouse

Primary Actor:

Clerk

Pre-conditions:

Clerk must be logged in to system

Post-conditions:

Success end condition:

Clerk has entered a supplier information and database is updated

Failure end condition:

Clerk is unable to add the new supplier

Main Success Scenario:

1.User goes to “Insert” drop down box
2.User clicks “Insert Supplier” button
3. System displays the insert form
4.User enters supplier name
5.User enters supplier e-mail
6.User enters supplier phone number
7.User clicks “Insert” button
8.System creates a new entry in the database

Use Case: View Client Order History

ID: UC-11

Description:
The Client checks on their order history of purchases
Primary Actor:
Client
Pre-conditions:
Client must be logged in to system
Post-conditions:
Success end condition:
Client sees their order history
Failure end condition:
Client didn't see no order history
Main Success Scenario:

1.User clicks “Order History” button
2.System displays the Order History form
3. The table set displays the users past order(s).

Extension:

For step 3:
3. If the table set doesn’t display past orders, the user must first order item(s).

Use Case: Invoice

ID: UC-12

Description:
The Clerk checks on a orders invoice
Primary Actor:
Clerk
Pre-conditions:
Clerk must be logged in to system
Client has placed an order from the warehouse
Post-conditions:
Success end condition:
Clerk sees the invoice of an order
Failure end condition:
Clerk didn't see the invoice
Main Success Scenario:

1.User clicks “Orders” button
2.System displays the Order Form
3. The table set displays the clients order(s).
4. User clicks on the OrderID of which invoice to see
5. User clicks “View” button
6. System displays the Order info Form

Use Case: View Client Account Page

ID: UC-13

Description:
The Client is able to view and modify their personal information in the account
Primary Actor:
Client
Pre-conditions:
Client must be logged in to system

Post-conditions:
Success end condition:
Client is able to view and update their personal information
Failure end condition:
Client is unable to update the account without the required information.
Main Success Scenario:

1. User clicks “Account” button
2. System displays the Account form
3. User clicks “Update” button
4. User enters phone number if it must be updated
5. User enters E-mail if it must be updated
6. User enters Address if it must be updated
7. User enters account password to authorize the update
8. User clicks “Save” button
9. System updates clients information

Use Case: View All Orders

ID: UC-14

Description:
The Clerk is able to view all the orders from the clients
Primary Actor:

Clerk
Pre-conditions:
Clerk must be logged in to system
Client has placed an order from the warehouse
Post-conditions:
Success end condition:
Clerk is able to see the clients orders
Failure end condition:
Clerk is unable to see the clients orders
Main Success Scenario:
1. User clicks " Orders" button
2. System displays the Order Form
3. The table set displays the clients orders
Extension:
For step 3:
3. If the table set doesn't display the client orders , there must first be clients that order item(s).

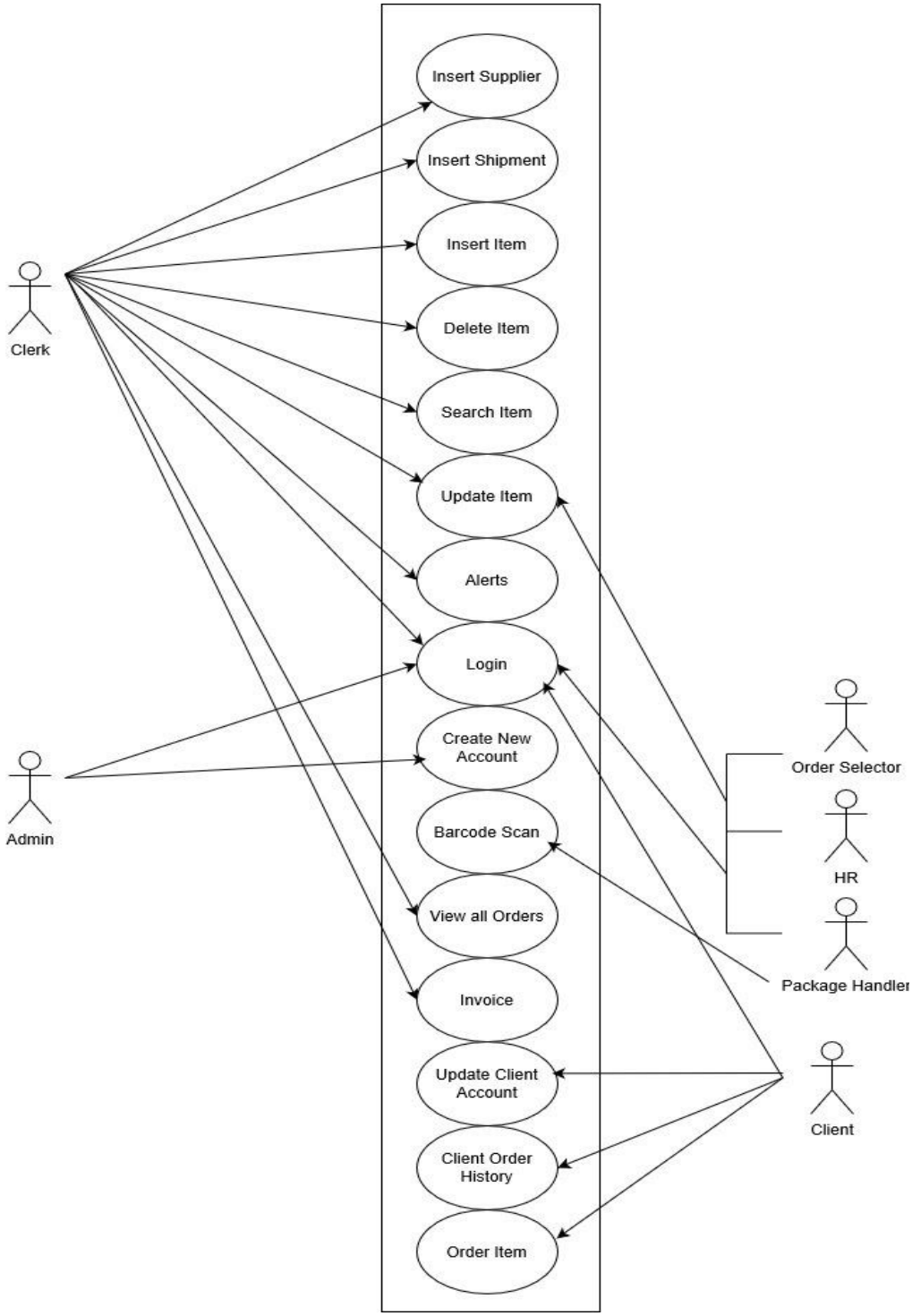
Use Case: Alert Low inventory/Expiration Date

ID: UC-15

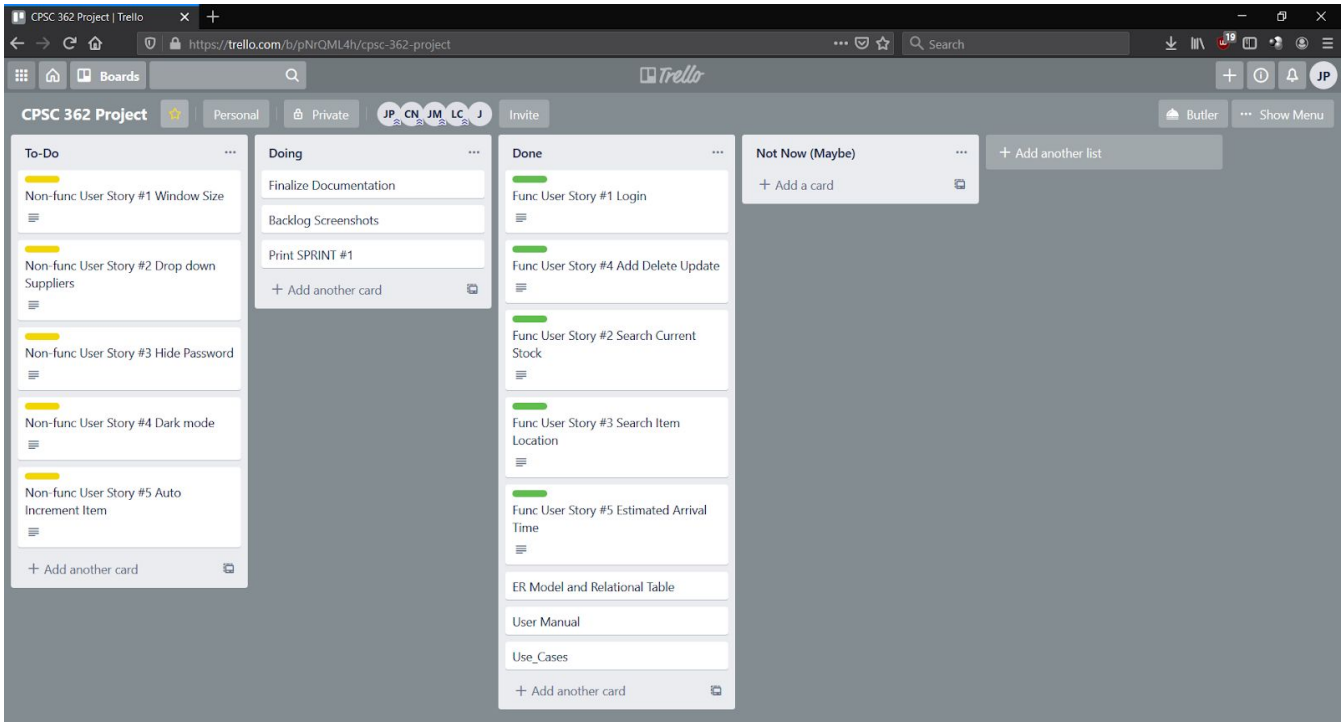
Description:
The Clerk is informed in a message automatically if the stock is low or the stock is about to expire
Primary Actor:
Clerk
Pre-conditions:
Clerk must be logged in to system

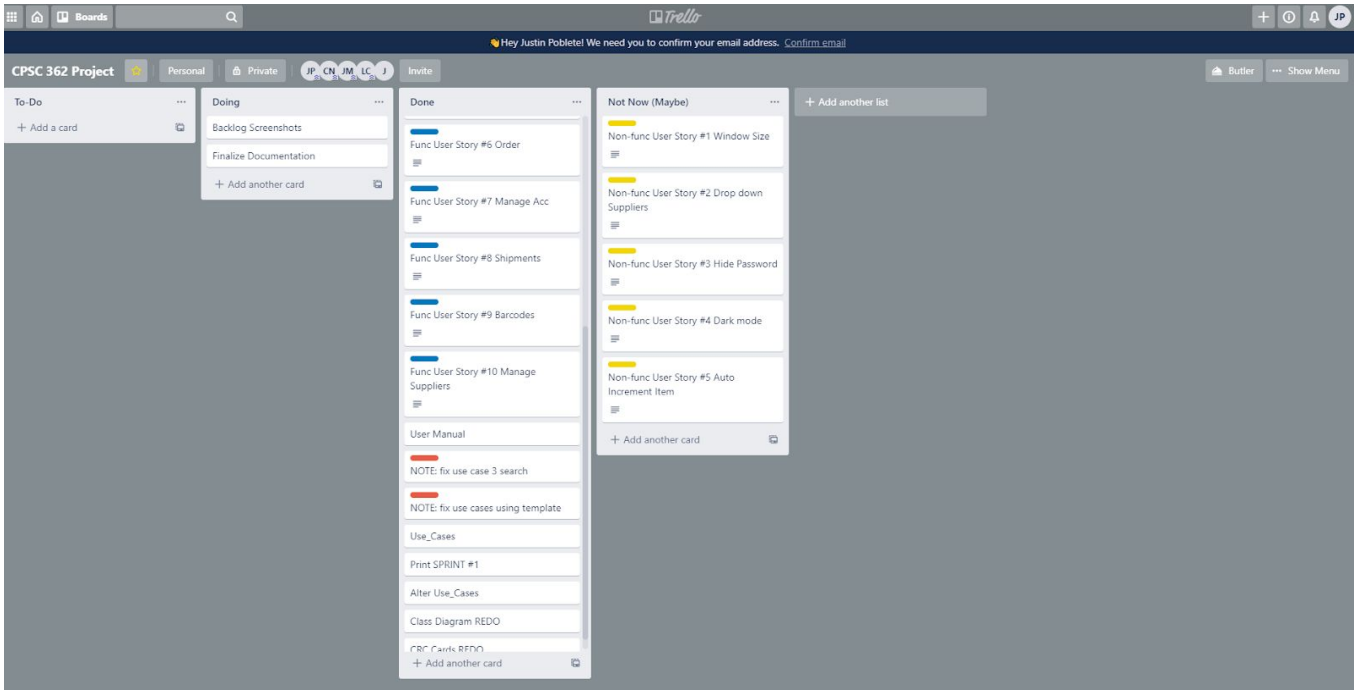
Post-conditions:
Success end condition:
Clerk is able to see the alert
Failure end condition:
Clerk is unable to see the alert
Main Success Scenario:
1. System displays the main menu
2. System then displays the alert message of low inventory
3. User clicks "Ok" button to confirm message
4. System then displays the alert message of soon to expire and expired items
5. User clicks "Ok" button to confirm message

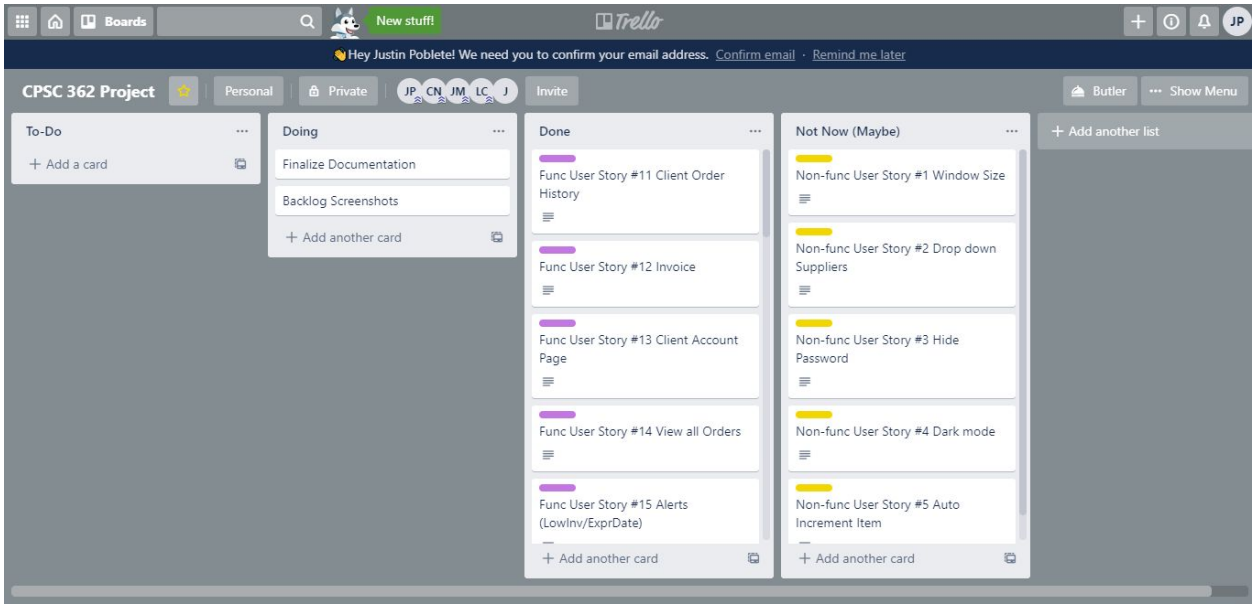
Use Cases Diagram



Backlog ScreenShots







Pre-Game Planning

Upon being tasked with creating an application we began to brainstorm. We had come up with review websites, dating applications, and much more. We eventually settled on creating an inventory program for grocery stores. We came to realize that we wanted more functionality so further down the line we decided to move onto a larger scaled warehouse food system similar to Costco. Each member in the project team came up with their own 5 user stories we believed were necessary for the required software.

Justin: The first 5 user stories I had chosen were only from the top of my head. Basic functions I believed to be the top priority in a warehouse inventory program. With this being the case, a few of the other user stories from my group were similar.
John: I chose these 5 user stories by thinking about the needs of individual employees. I tried to think about how different workers with different jobs could all benefit from this software.
Juheng: The first 5 user stories that I came up with are what I believe are the basic functionalities for an inventory program.
Calvin: The first 5 user stores were chosen because user access is the first priority of the warehouse system. Then the other basic functions were for keeping track of the warehouse's inventory and updating necessary details.
Luis: The reason for choosing my 5 user stories is that they will satisfy the user and are key features in the database. The reason for implementing a login account and changing password is

that the user can have peace of mind the database is secure. Also, implementing delete and edit as key features that help maintain a database

Staging/Grooming Plan

We then went on to research actual employees' occupations that pertained to the warehouse system to best emulate what would be useful for our program. After discussion, we chose 10 user stories from the collection and prioritized them accordingly. Among the chosen 10, we utilized the scrum poker game to pick and prioritize the most important functions to fit the first phase of the project.

These are our results from the scrum poker game:

Story #	Story Description	Poker Score	First Sprint (Top 5)
1	HR - Login	76	*
2	Order selector – look up stock	63.33	*
3	Package handler – stock location	72	*
4	Client manager – look up client markets	25.2	
5	Stocker – scan in received items	25.4	
6	Clerk – insert distributor stock	34.6	*
7	Clerk – look up shipment times	34	*
8	Clerk – create and print invoices	17.6	
9	Clerk – look up expiration dates on stock	18.8	
10	HR – employee scheduling	6.2	

Summary of Poker Results:

For our poker game, we assigned scores between 0-100 based on how important each individual member thought the story was. We used www.pointingpoker.com to record our votes. The five user stories we chose from the poker game do not completely match up with our initial planning stage. The first three user stories, #1-3, were our highest priority user stories during initial planning and remained our highest three scoring stories during the game. These three

stories will make it to the first sprint. The next two stories, #4 and #5, which were initially higher priority, actually received lower scores than #6 and #7. Therefore, #6 and #7 will be making it into the first sprint, while #4 and #5 will be addressed later.

Analysis of Poker Results

Login accounts having the highest score makes sense, since the most basic functionality of an inventory system is being able to log in and access the database. Account management also includes basic security, since managers can control who gets accounts.

The next two are also very important features. The order selectors must be able to look up current stock so that they can plan and adjust future orders. This prevents the warehouse from over/under stocking and wasting money. Next, the package handlers must be able to look up package locations for the warehouse to reach basic levels of efficiency. #6 was chosen because the clerk will be handling most database tasks, so their access is essential to populate and manage the database. These four stories, #1-3 and 6, we selected primarily based on their importance for the basic operation of a distributor.

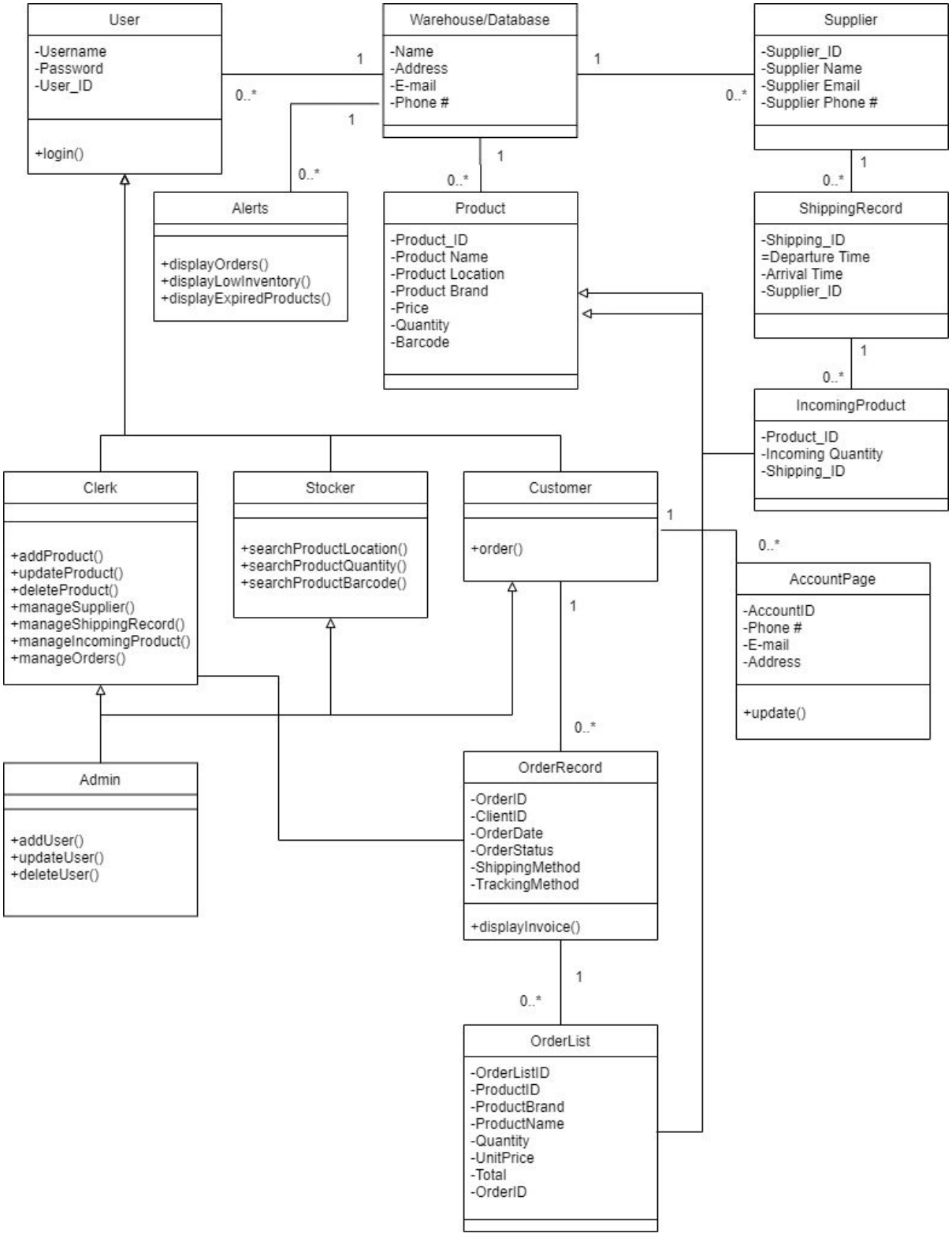
Clerks looking up shipment times received the last spot for our first sprint. Modern distributors work on very tight schedules, so being able to access incoming and outgoing shipment times is another important function of the warehouse.

The remaining stories were deemed non-essential and will either be implemented later, or have their functionality rolled into other portions of the program. #4 and #5 are fairly important and will probably be implemented in full down the line. #8-10 may be handled by other departments or software, so their inclusion or functionality may be adjusted later.

Development Process

When first developing the program, the team created an ER-diagram and Relational-diagram to distinguish all the items that are needed in a database. In this process, we organized the product information and supplier information to their specific tables. As the sprints went on, we added shipments and orders to these diagrams and tables with their corresponding attributes and keys. Search and account management functionalities were later implemented through research from various forums and tutorials. Everything was done in Windows Visual Studio with C# and connected to a SQL server.

Class Diagram



CRC Cards

Class Name: Warehouse	**ID: 1**	**Type:**
Description: Manages the entire system.		**Associated Use Cases: 10**
Responsibilities: Hold Users Hold Products Hold Suppliers Hold Orders		**Collaborators:** **Users** **Suppliers** **Product** **Alerts**

Attributes: Name Location Email Phone #
Relationships ● **Generalization:** ● **Aggregation: User, Supplier, Product** ● **Other Associations:**

Class Name: User	**ID: 2**	**Type:**
Description: Users will be individual account holders that interact with the system, by first logging in.		**Associated Use Cases: 10**
Responsibilities: Ability to login		**Collaborators:** **Clerk** **Stocker** **Customer** **Admin**

Attributes: Username Password

User_ID
Relationships ● **Generalization: Clerk, Stocker, Customer, Admin** ● **Aggregation:** ● **Other Associations:**

Class Name: Clerk	**ID: 3**	**Type:**
Description: User that Manages Products, Suppliers, and Orders.		**Associated Use Cases: 7**
Responsibilities: Add, delete, update all the products Add delete, update all the suppliers Delete and update orders		**Collaborators:** **Admin**

Attributes:
Relationships ● **Generalization: Admin** ● **Aggregation:** ● **Other Associations:**

Class Name: Stocker	**ID: 4**	**Type:**
Description: User that can view and search the products in order to maintain the warehouse.		**Associated Use Cases: 2**
Responsibilities: Search product location Search product quantity Search product barcodes		**Collaborators:** **Admin**

Attributes:
Relationships ● **Generalization: Admin** ● **Aggregation:** ● **Other Associations:**

Class Name: Customer	**ID: 5**	**Type:**
Description: User that places orders of items from the warehouse		**Associated Use Cases:1**
Responsibilities: Create Order		**Collaborators:** **Admin**

Attributes: Customer_ID Customer Name Customer Email Customer Phone #
Relationships ● **Generalization: Admin** ● **Aggregation:** ● **Other Associations:**

Class Name: Admin	**ID: 6**	**Type:**
Description: User that manages other users and has access to all other user's functions.		**Associated Use Cases:1**

Responsibilities: Manage the accounts of each user. Supervise all the actions of each user. Test system functions	**Collaborators:** **Clerk** **Stocker** **Customer**

Attributes:
Relationships ● **Generalization: Clerk, Stocker, Customer** ● **Aggregation:** ● **Other Associations:**

Class Name: Product	**ID: 7**	**Type:**
Description: The items being stored in the warehouse.		**Associated Use Cases:5**
Responsibilities: To be received into the warehouse To be sold to customers		**Collaborators:** **Warehouse** **IncomingProduct**

Attributes: Product_Id Product Name Product Location Price Quantity
Relationships ● **Generalization: IncomingProduct** ● **Aggregation:** ● **Other Associations:**

Class Name: Supplier	**ID: 8**	**Type:**
Description: Companies or individuals that ship out items to the warehouse.		**Associated Use Cases:1**
Responsibilities: Supply the warehouse.		**Collaborators:** **Warehouse** **ShippingRecord**

Attributes: Supplier_ID Supplier Name Supplier Phone #
Relationships ● **Generalization:** ● **Aggregation: Warehouse, ShippingRecord** ● **Other Associations:**

Class Name: ShippingRecord	**ID: 9**	**Type:**
Description: A record of a shipment from a supplier that is on its way to the warehouse.		**Associated Use Cases:1**
Responsibilities: Keep track of shipments		**Collaborators:** **Supplier** **IncomingProduct**

Attributes: Shipping_ID Departure Time Arrival Time Supplier_ID

Relationships ● **Generalization:** ● **Aggregation: Supplier, IncomingProduct** ● **Other Associations:**

Class Name: IncomingProduct	**ID: 10**	**Type:**
Description: Items that are being shipped and added to the warehouse.		**Associated Use Cases:1**
Responsibilities: To be added to the warehouse		**Collaborators:** **ShippingRecord** **Product**

Attributes: Product_ID Incoming Quantity Shipping_ID
Relationships ● **Generalization: Product** ● **Aggregation: Shipping Record** ● **Other Associations:**

Class Name: Alerts	**ID: 11**	**Type:**
Description: Alerts let the user know what orders have been placed, low inventory quantity, and expired products.		**Associated Use Cases:1**
Responsibilities: Notify the user Display orders Display low inventory Display expired products		**Collaborators:** **Warehouse**

Attributes:
Relationships ● **Generalization:** ● **Aggregation: Warehouse** ● **Other Associations:**

Class Name: **OrderRecord**	**ID: 12**	**Type:**
Description: A record that an order that has been placed by a client.		**Associated Use Cases:2**
Responsibilities: Keep track of the order that has been placed for both the employees and clients Display invoice of selected order		**Collaborators:** **Customer** **OrderList**

Attributes: -OrderID -ClientID -OrderDate -OrderStatus -ShippingMethod -TrackingMethod
Relationships ● **Generalization: Product** ● **Aggregation: Shipping Record** ● **Other Associations:**

Class Name: **OrderList**	**ID: 13**	**Type:**

Description: An itemized list of all the products within an order placed by the client	**Associated Use Cases:1**
Responsibilities: Keep track of all the items in an order and be retrieved through the invoice	**Collaborators:** **OrderRecord**

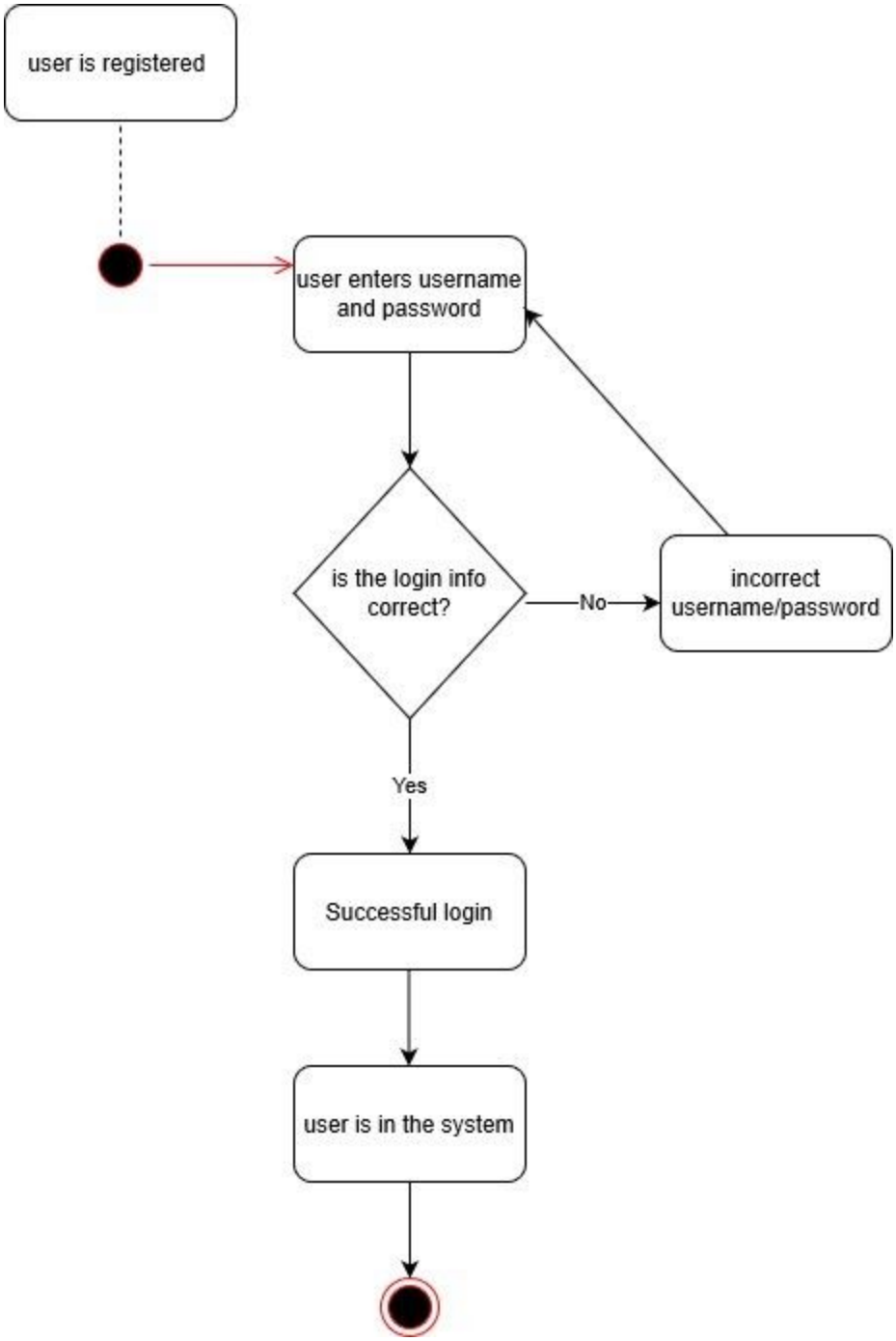
Attributes: -OrderListID -ProductID -ProductBrand -ProductName -Quantity -UnitPrice -Total -OrderID
Relationships ● **Generalization: Product** ● **Aggregation: OrderRecord** ● **Other Associations:**

Class Name: **AccountPage**	**ID: 14**	**Type:**
Description: Display the personal information of the client with the added ability to update their own information.		**Associated Use Cases:1**
Responsibilities: Display account information to clients Update information		**Collaborators:** **Customer**

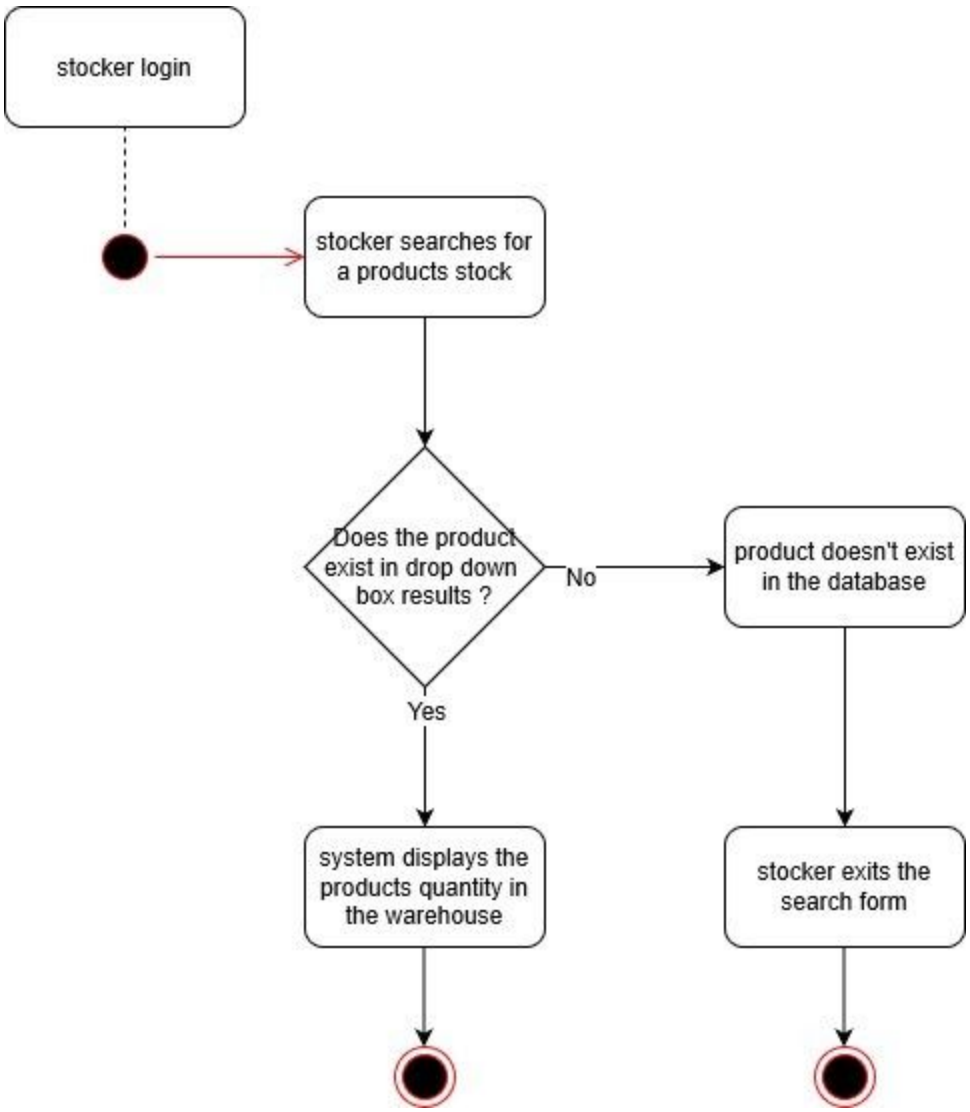
Attributes: -AccountID -Phone # -E-mail -Address
Relationships ● **Generalization:** ● **Aggregation: Customer** ● **Other Associations:**

Activity Diagrams

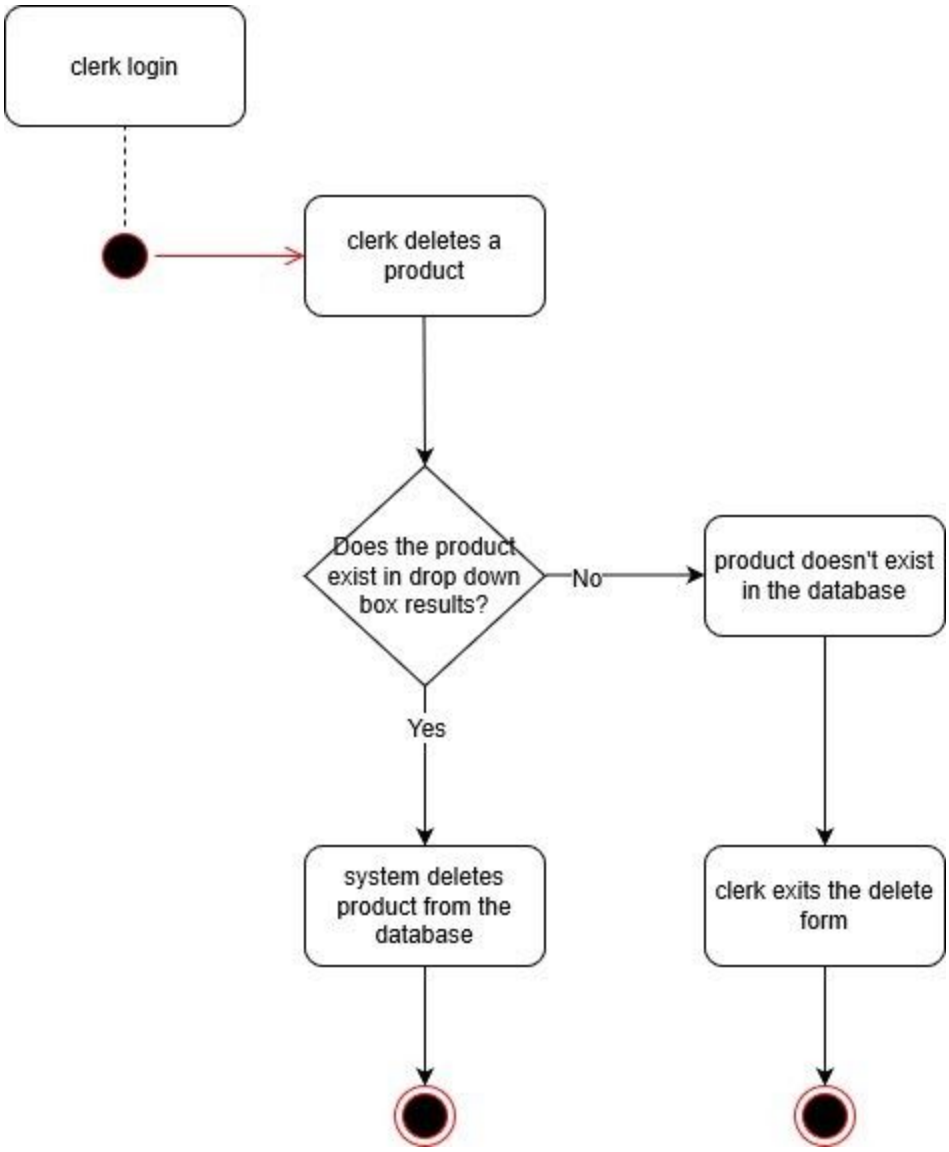
UC-01



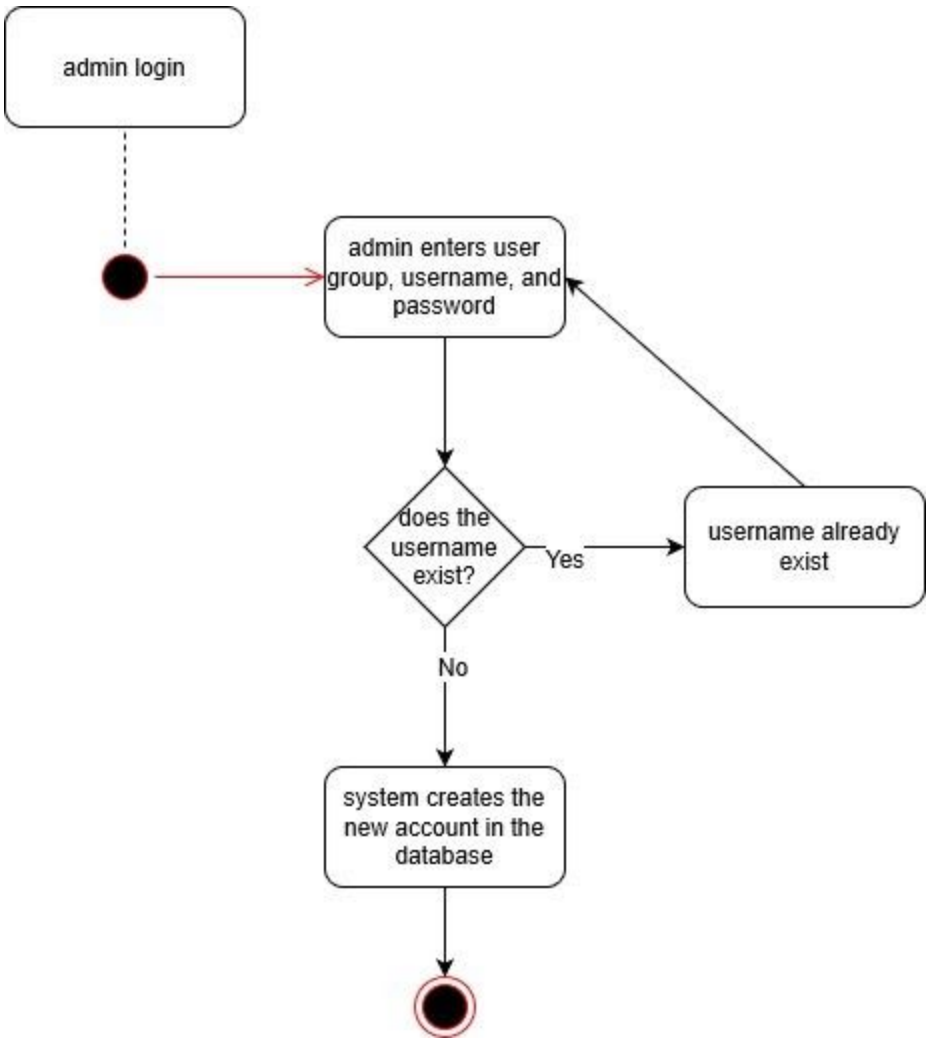
UC-03


stocker login
stocker searches for a products stock
Does the product exist in drop down box results ?
No
product doesn't exist in the database
Yes
system displays the products quantity in the warehouse
stocker exits the search form

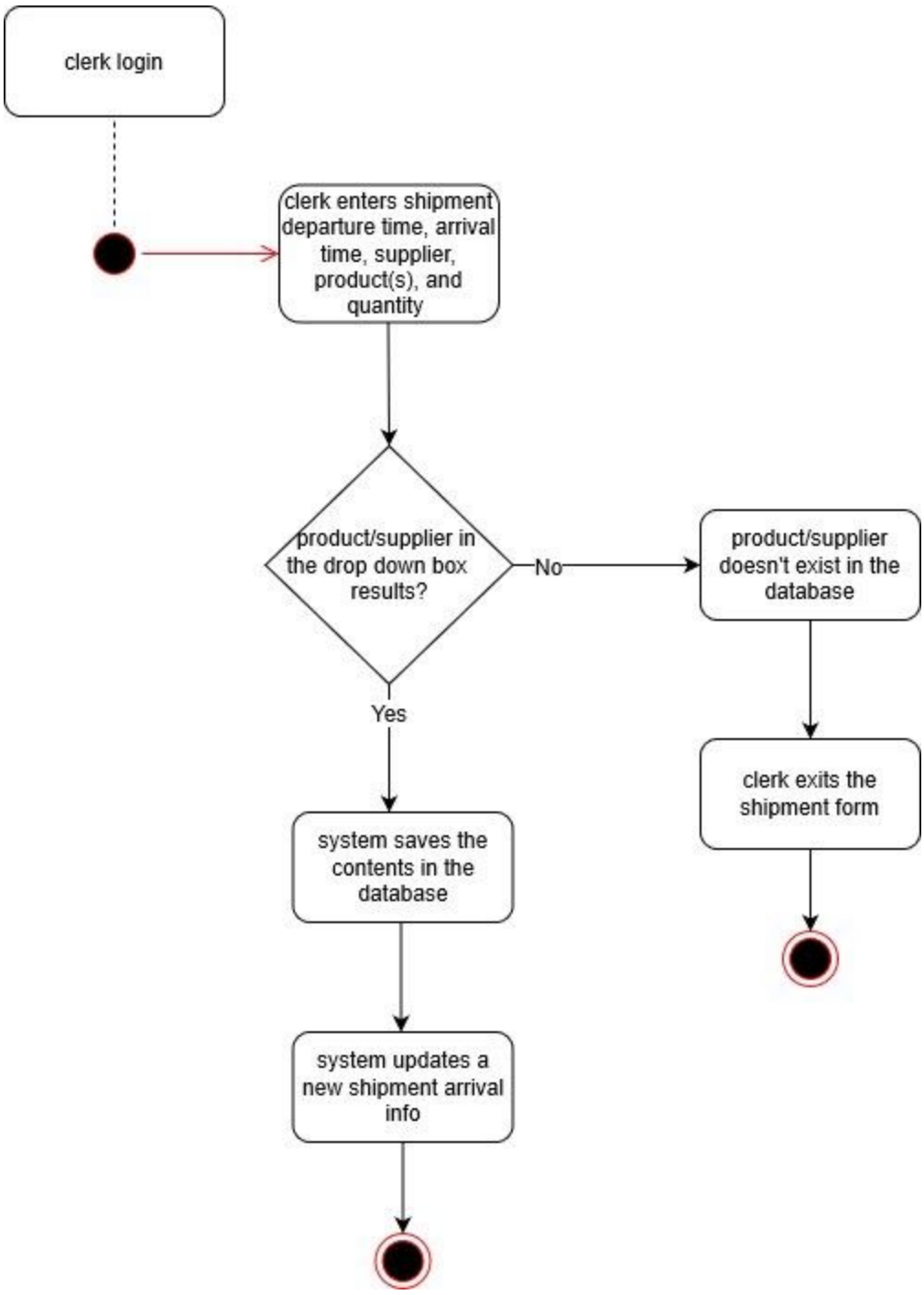
UC-05


clerk login
clerk deletes a product
Does the product exist in drop down box results?
No
product doesn't exist in the database
Yes
system deletes product from the database
clerk exits the delete form

UC-07


admin login
admin enters user group, username, and password
does the username exist?
Yes
username already exist
No
system creates the new account in the database

UC-08


clerk login
clerk enters shipment departure time, arrival time, supplier, product(s), and quantity
product/supplier in the drop down box results?
No
product/supplier doesn't exist in the database
clerk exits the shipment form
Yes
system saves the contents in the database
system updates a new shipment arrival info

Test Case #: 1 **System**: Warehouse Inventory Program **Designed by**: Juheng Mo **Executed by**: **Short Description**: Test the Warehouse Inventory Program login function	**Test Case Name**: Login **Subsystem**: Login Table **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
The system displays the login menu.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “Login” button	The system displays the message “You did not enter username and password!”	Pass	
2	Enter “calvin”	The username textbox shows “calvin”	Pass	
3	Click “Login” button	The system displays the message “You did not enter a password!”	Pass	
4	Clear the username textbox	The username textbox is empty	Pass	
5	Enter “123”	The password textbox shows “***”	Pass	
6	Click “Login” button	The system displays the message “You did not enter a username!”	Pass	

7	Enter “calvin”	The username textbox shows “calvin”	Pass	
8	Click “Login” button	The system displays the main menu	Pass	
9	**Check post-condition 1**		Pass	

Post-Conditions:
1. The user sees the main menu

Test Case #: 2 **System**: Warehouse Inventory Program **Designed by**: John Gardner **Executed by**: **Short Description**: Look up stock for products.	**Test Case Name**: Lookup **Subsystem**: Inventory Database **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
Inventory database is populated with some products and product amounts.
For this test case, database has 10 Diet Coke, product ID # 0003.
The System displays main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “search product stock” button	Display search box	Pass	

2	View drop down menu	Products are displayed in drop down menu, including “Coca Cola Diet Coke”	Pass	
3	Select “Diet Coke” from drop down menu and click “search”	Display dialogue box with product ID “#0003”, name “Diet Coke”, and stock “10”	Pass	
4	Click “OK” in dialogue box	Dialogue box closes, return to search box	Pass	
5	Click “Exit” in search box	Search box closes, return to main menu	Pass	
6	**Check post-condition 1**		Pass	

Post-Conditions:

1. Return to the main menu. Inventory contents should not be changed by this test.

Test Case #: 3 **System**: Warehouse Inventory Program **Designed by**: Luis Camacho **Executed by**: **Short Description**: Test to create a new account for a user	**Test Case Name**: Create Account **Subsystem**: Users **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:

Username “calvin” and password “123” are in the database
There is no user that uses the username “Bob”
The System displays main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “Create Account” button	System displays the Create Account form	Pass	
2	Enter username “Bob”	Username shows “Bob”	Pass	
3	Enter Password “321”	Password Shows “321”	Pass	
4	Choose User Group “Admin”	User Group shows “Admin”	Pass	
8	Click “Create” button	The system displays message of the account was created	Pass	
9	Select the “Ok” button from the confirmation message	The system displays the new User menu with an updated gridview of users	Pass	
10	**Check post-condition 1.**		Pass	

Post-Conditions:

1. New user account of “Bob” is saved in the user database

Test Case #: 4 **System**: Warehouse Inventory Program **Designed by**: Calvin Nguyen **Executed by**: **Short Description**: Test the Update product information	**Test Case Name**: Update Product Info **Subsystem**: Update Product Table **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
The Product “Coca Cola” is in the database and the current quantity is 10. Product Id #0009
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	User clicks on “Update” in drop down menu	The system displays drop down menu for User to select	Pass	
2	User clicks “Update Product”	The system displays the update form for User.	Pass	
3	User clicks on ProductID drop down menu to select the Product to be updated	The system displays all current products in the system.	Pass	
4	User clicks ProductID 9 for Coca Cola	The system selects ProductID 9	Pass	
5	User enters “50” in Quantity field and press Update	The system displays a message to notify user about the update	Pass	
6	User clicks OK to finish the update	The system empty all the fields for user to update next product	Pass	

7	**Check post-condition 1**		Pass	

Post-Conditions:
1. The new Quantity for the product Coca Cola is 50 in the database.

Test Case #: 5 **System**: Warehouse Inventory Program **Designed by**: John Gardner **Executed by**: **Short Description**: Delete an item from the database.	**Test Case Name**: Delete Item **Subsystem**: Inventory Database **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
Inventory is populated with some products, including name, amount, and IDs
For this test, inventory will have “Coca Cola Diet Coke”, ProductID #0003, quantity 10

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “Delete” drop down menu	“Delete Product” is displayed in the drop down menu	Pass	
2	Click “Delete Product”	“Delete Product” box is displayed	Pass	
3	User searches through product drop down menu	“Coca Cola Diet Coke” is among the products listed	Pass	

4	User selects “Coca Cola Diet Coke” and presses delete	Display dialogue box that says ProductID “#0003”, name “Diet Coke” has been deleted, “Diet Coke” deleted from inventory	Pass	
5	User exits “Delete” box	Return to main menu	Pass	
6	User checks inventory for post conditions	“Diet Coke” not found	Pass	
7	**Check post-condition 1**		Pass	

Post-Conditions:
1. Product “Diet Coke” is no longer in the inventory.

Test Case #: 6 **System**: Warehouse Inventory Program **Designed by**: Calvin Nguyen **Executed by**: **Short Description**: Test the client’s ability to order items.	**Test Case Name**: Order Item **Subsystem**: Order **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “traderjoe” and password “123” are in the database
The Product “Pepsi, Sierra Mist” is available in the database with more than 10 quantity.
The system displays the login menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	The user enters “traderjoe” and “123” for username and password respectively	The system authenticates username and password then displays the Order Form for client	Pass	
2	User selects 10 quantities of “Pepsi Sierra Mist.”	The display saves the Quantity 10 for “Pepsi Sierra Mist”	Pass	
3	User selects “USPS” from the Shipping method dropdown menu.	Shipping Method shows “USPS”	Pass	
4	The User clicks “Order Now” to finalize the order to be placed.	The system displays a message of successful operation	Pass	
5	The User clicks OK to confirm the purchase order.	The system saves the order and subtract product quantity that was purchased from the database.	Pass	
6	**Check post-condition 1 and 2**		Pass	

Post-Conditions:

1. The system saves the order that was placed into the database.
2. The system updates the quantity of the product that was purchased.

Test Case #: 7 **System**: Warehouse Inventory Program **Designed by**: Luis Camacho **Executed by**: **Short Description**: Test the functionality to delete an account.	**Test Case Name**: Delete Account **Subsystem**: Users **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
User account by the username “Bob” is in the database
System displays main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “ Delete account” button	System displays delete account form	Pass	
2	View drop down menu	Usernames are displayed in drop down menu, including “Bob”	Pass	
3	Select “Bob” from drop down menu	Drop down box shows “Bob”	Pass	
4	Click “delete” button	System displays message that account has been deleted	Pass	
5	Select the “Ok” button from the confirmation message	The system displays the Delete User menu with an updated gridview of users	Pass	
6	**Check post-condition 1**		Pass	

Post-Conditions:
1. The old user account “Bob” has been deleted in the user database

Test Case #: 8 **System**: Warehouse Inventory Program **Designed by**: Juheng Mo **Executed by**: **Short Description**: Test the Insert Shipment function	**Test Case Name**: Insert Shipment **Subsystem**: Shipping Record Table **Design Date**: 4/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
The product “Coca Cola” is in the database
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click “Insert ” button in drop down menu	The system displays the drop down menu for “insert”	Pass	
2	Click “Insert Shipping Information”	The system displays the insert shipment menu	Pass	
3	Enter “04/22/2020” for departure date	The departure date textbox shows “04/22/2020”	Pass	
4	Enter “04/23/2020” for arrival date	The departure date textbox shows “04/23/2020”	Pass	

5	Click the drop down box for supplier name	The dropdown box shows a list of supplier name in database	Pass	
6	Click “Coca-Cola Company” in drop down box	The supplier name dropdown box shows “Coca-Cola Company”	Pass	
7	Click the drop down box in the row that has a product brand “Coca Cola” and a product name “Coca Cola”	The dropdown box shows option 1 to 10	Pass	
8	Select “1” in the dropdown box	The dropdown box shows “1”	Pass	
9	Click “Insert” button	The system shows a messaging saying that shipment information is inserted	Pass	
10	**Check-post condition 1**		Pass	

Post-Conditions:

1. Shipment information for 1 case of Coca Cola from The Coca-Cola Company is saved in the database

Test Case #: 9 **System**: Warehouse Inventory Program **Designed by**: Justin Poblete **Executed by**: **Short Description**: Test the Warehouse Inventory Program search barcode function	**Test Case Name**: Search Barcode **Subsystem**: Product **Design Date**:04/21/2020 **Execution Date**:

Pre-Conditions:
Username “calvin” and password “123” are in the database
The product “Coca Cola” is in the database.
The barcode # for Coca Cola is 827369192
The system displays the main menu.

Step	Action	Expected System Response	Pass /Fail	Comment
1	Select “Product Barcode” Button.	The system displays the Product Barcode menu.	Pass	
2	Select the product “Coca Cola Coca Cola” using the dropdown menu.	The system expands the dropdown menu and Coca Cola is available.	Pass	
3	Select “View”	The system displays the barcode and number	Pass	
4	**Check post-condition 1**		Pass	

Post-Conditions:
1. The product barcode and barcode number as displayed.

Test Case #: 10 **System**: Warehouse Inventory System **Designed by**: Justin Poblete **Executed by**: **Short Description**: Test the Warehouse Inventory Program insert supplier function	**Test Case Name**: Insert Supplier **Subsystem**: Supplier **Design Date**: 04/21/2020 **Execution Date**:

Pre-Conditions:
The user is logged in
The test supplier information inputted will be “Test, test@gmail.com, (555) 555-5555” with the Supplier ID 0003
The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Select “Insert Supplier” tab	The system displays the Insert Supplier menu	Pass	
2	Enter “Test” into Supplier Name	Supplier Name Shows “Test	Pass	
3	Enter “test@gmail.com” into E-Mail	Phone Number shows “test@gmail.com”	Pass	
4	Enter “(555) 555-5555” into Phone Number	Phone Number shows “(555) 555-5555	Pass	
5	Select “Insert” Button	The system displays a confirmation message with the Supplier ID	Pass	

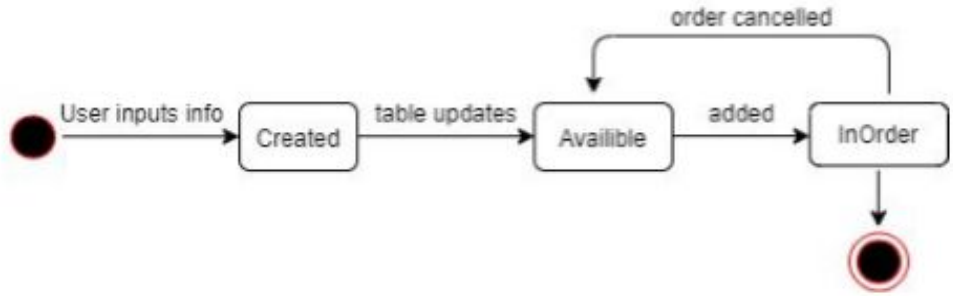
6	Select the “Ok” button from the confirmation message	The system displays the Insert Supplier menu with an updated gridview of Suppliers	Pass	
7	**Check post-condition 1**		Pass	

Post-Conditions:

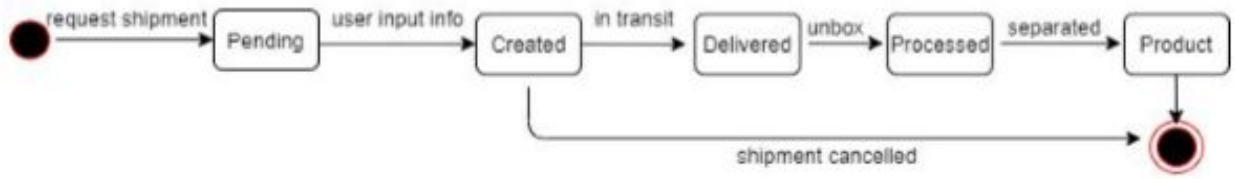
1. The new supplier “Test” is saved onto the database.

State Diagrams

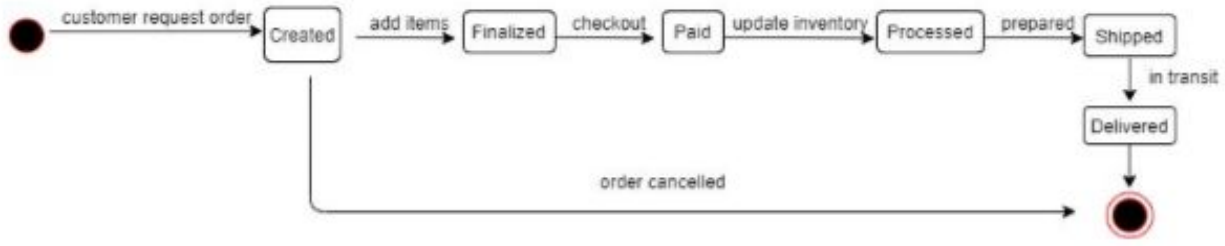
Product



Shipment

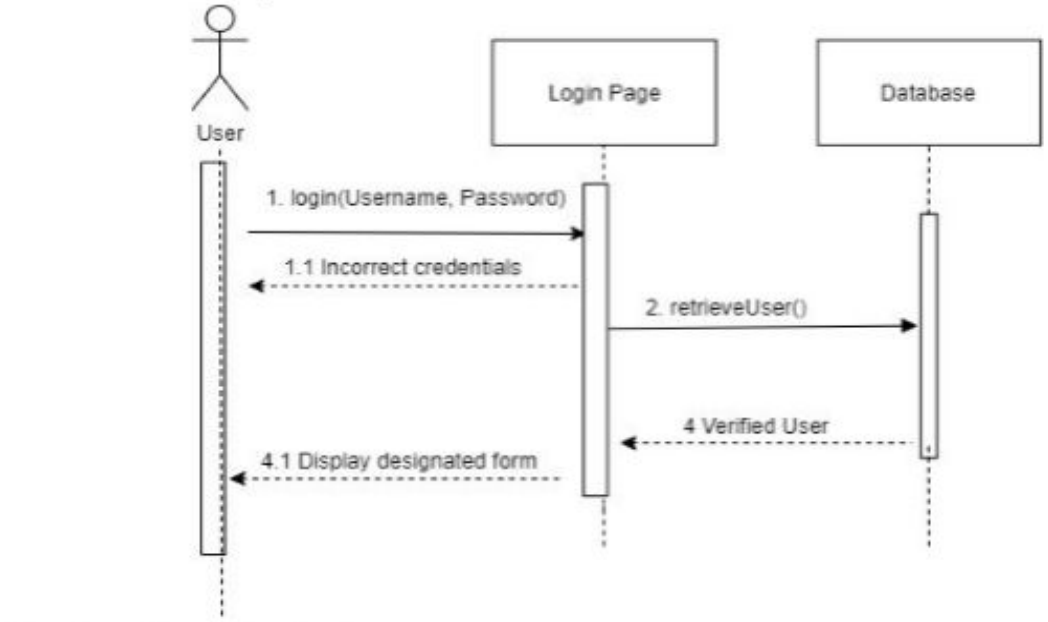


Order

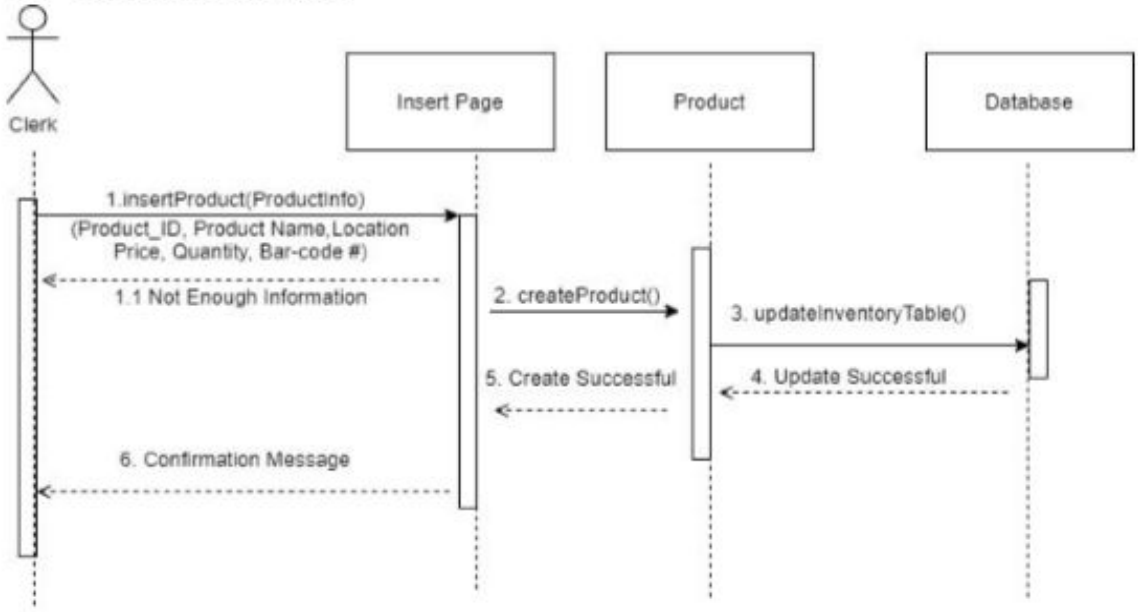


Sequence Diagrams

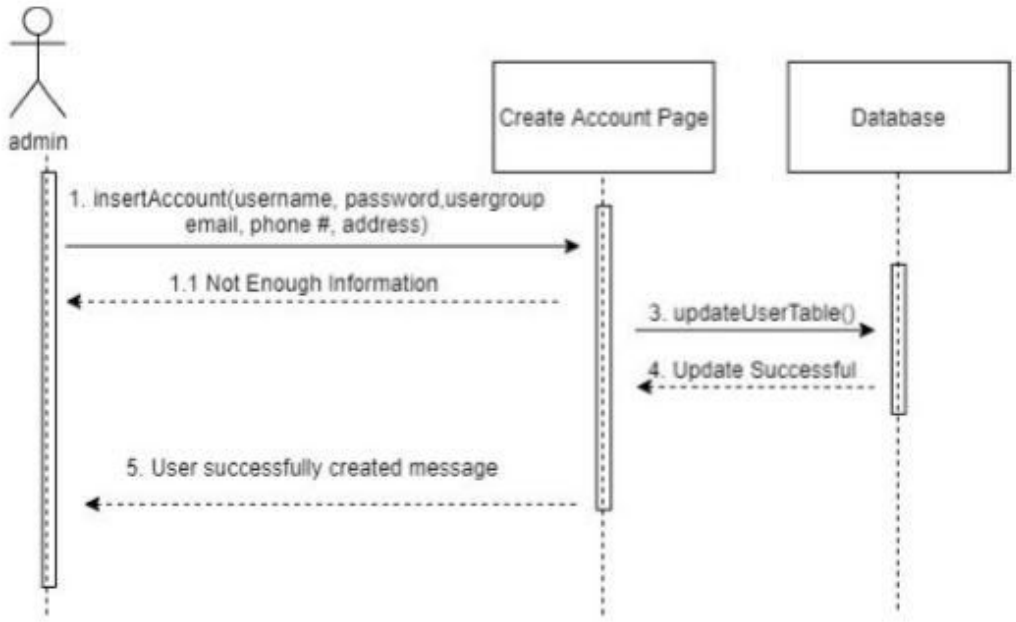
Use Case: User Login



Use Case: Insert Product

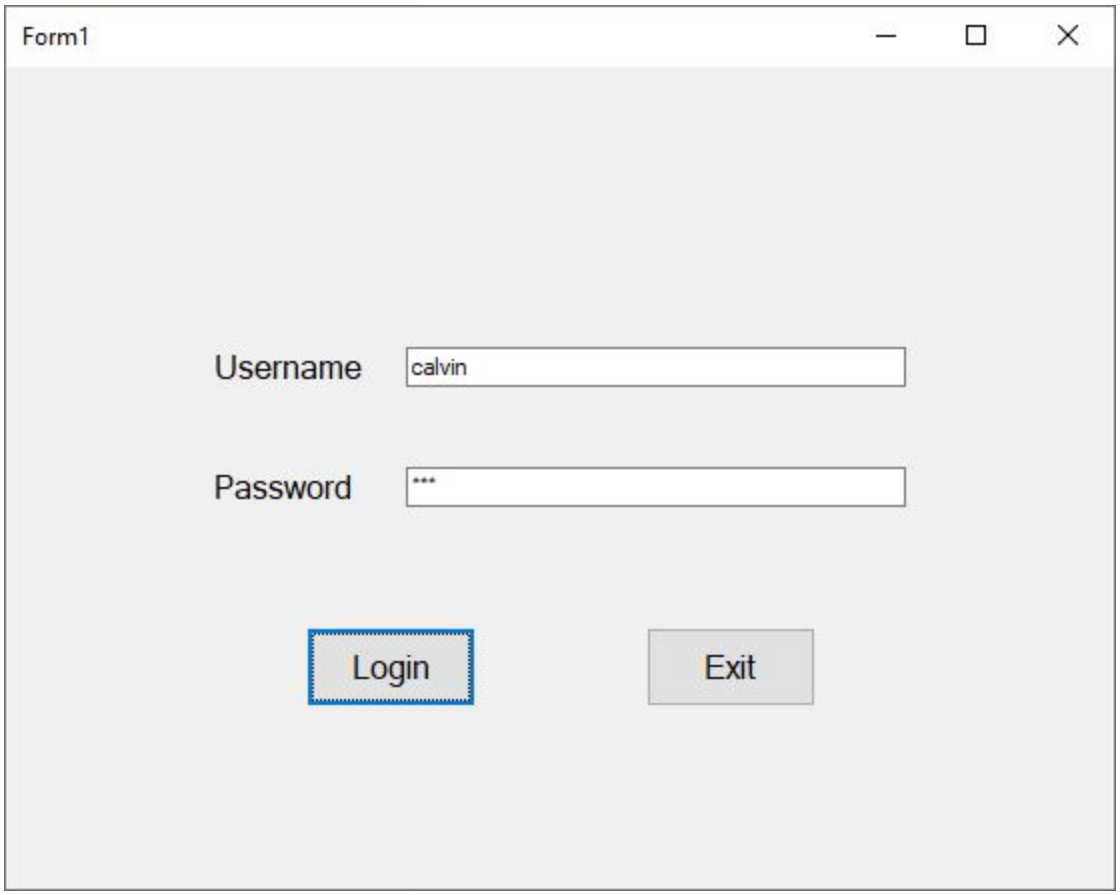


Use Case: Insert User



User Manual

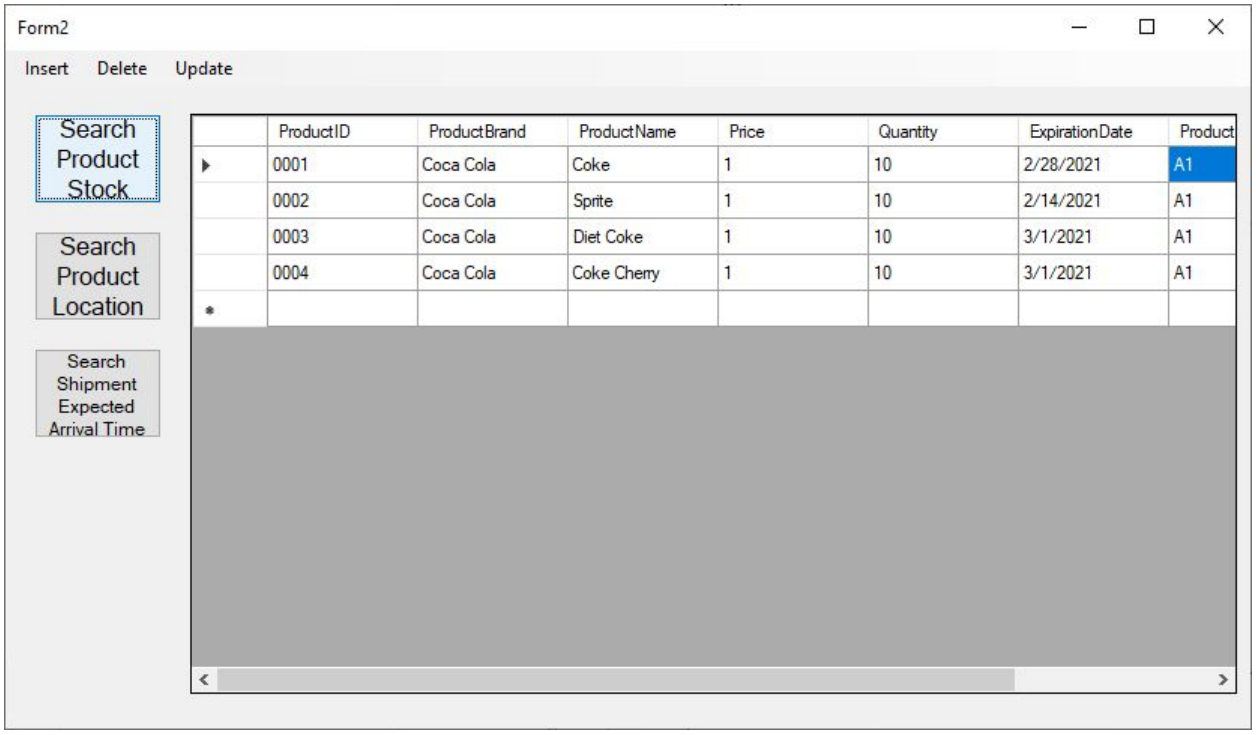
User Story 1: Login



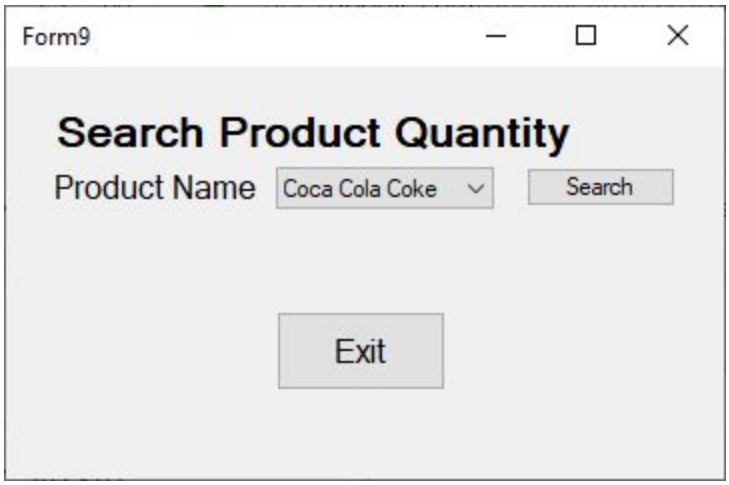
1. Input username in “Username” field.
2. Input password in “Password” field.
3. Click the “Login” button.
4. If the login was successful, the program will proceed to the main menu. Otherwise an error message will appear.

User Story 2: Search Product Stock

1. Click the “Search Product Stock” button on the main inventory screen.



2. Select the product to lookup in the drop down menu.

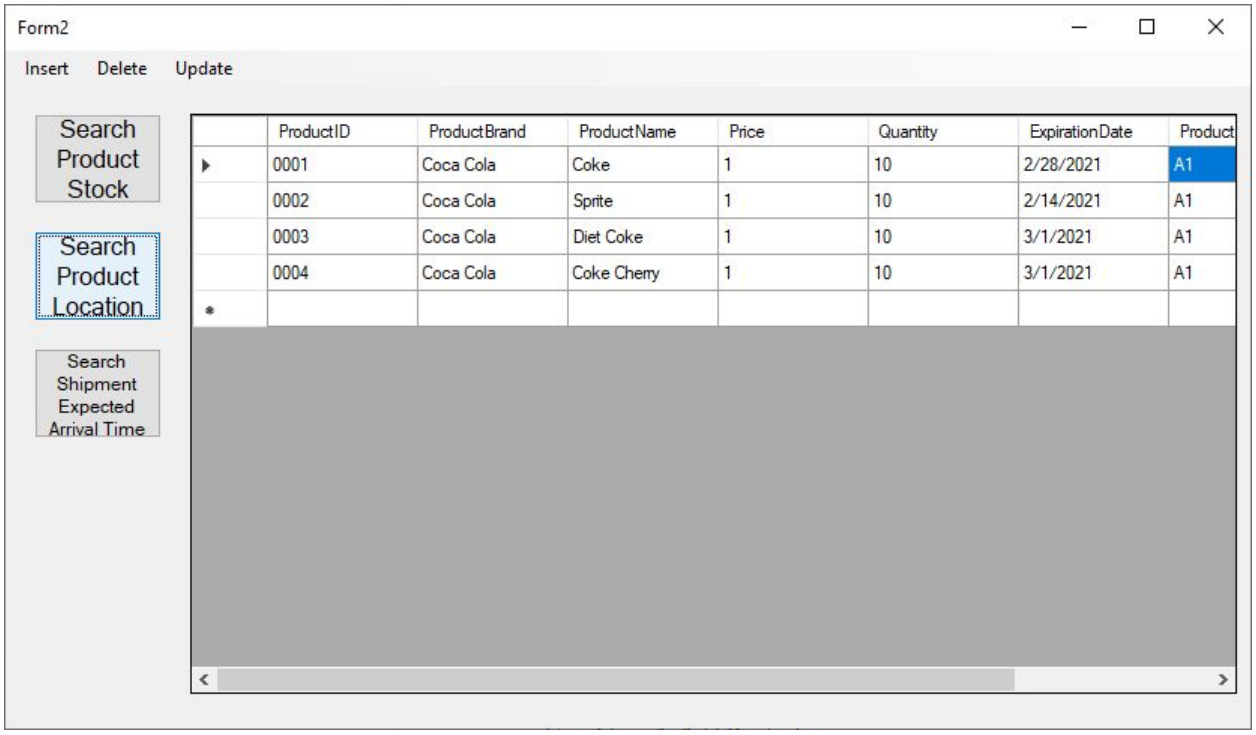


3. Click the “Search” button.

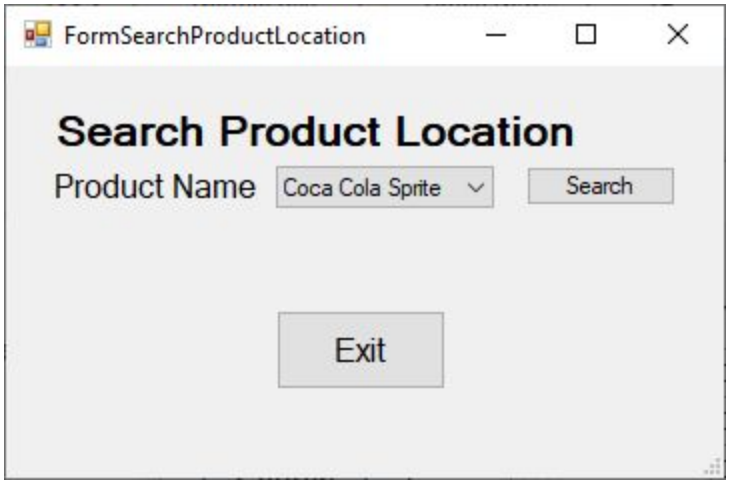
4. A dialogue box will appear displaying the item's current stock.

User Story 3: Search Product Location

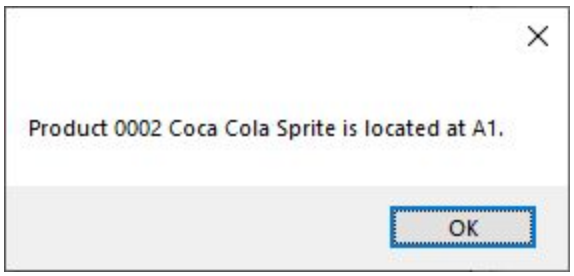
1. Click the "Search Product Location" button on the main menu.



2. Select the product to look up in the drop down menu.

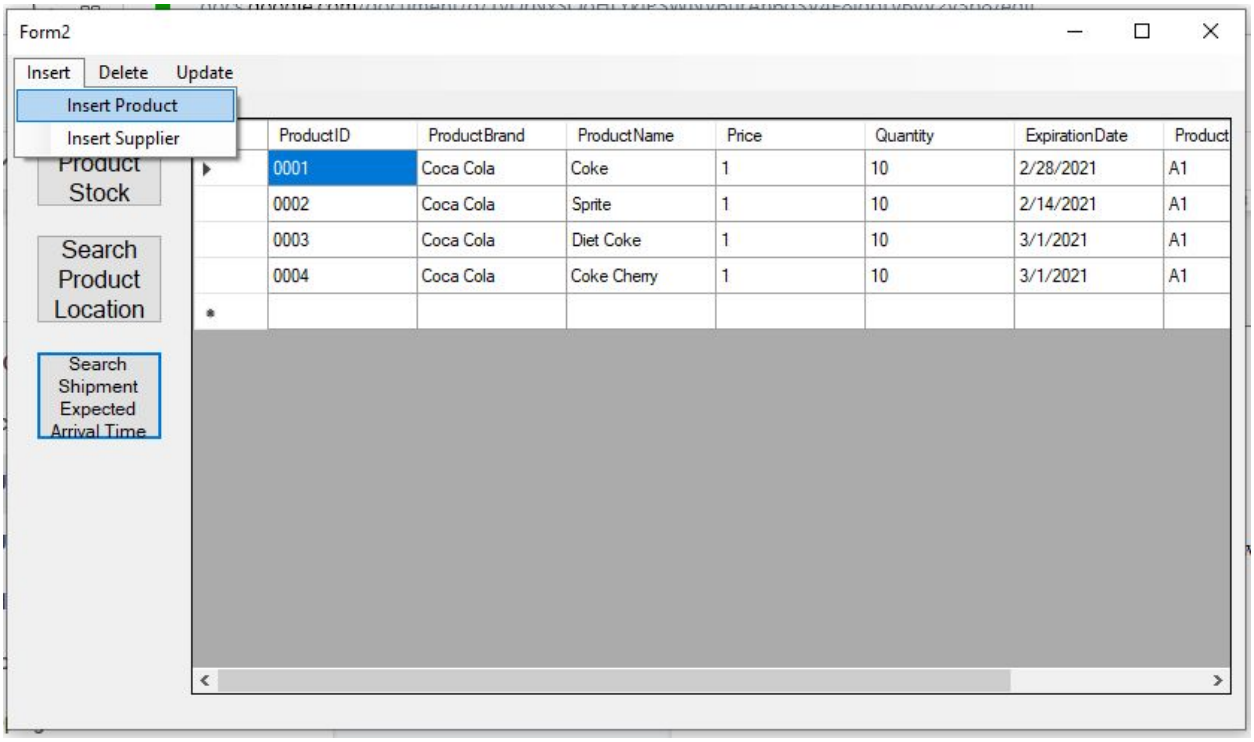


3. Click the “Search” button.
4. A dialogue box will appear displaying the item’s current location.

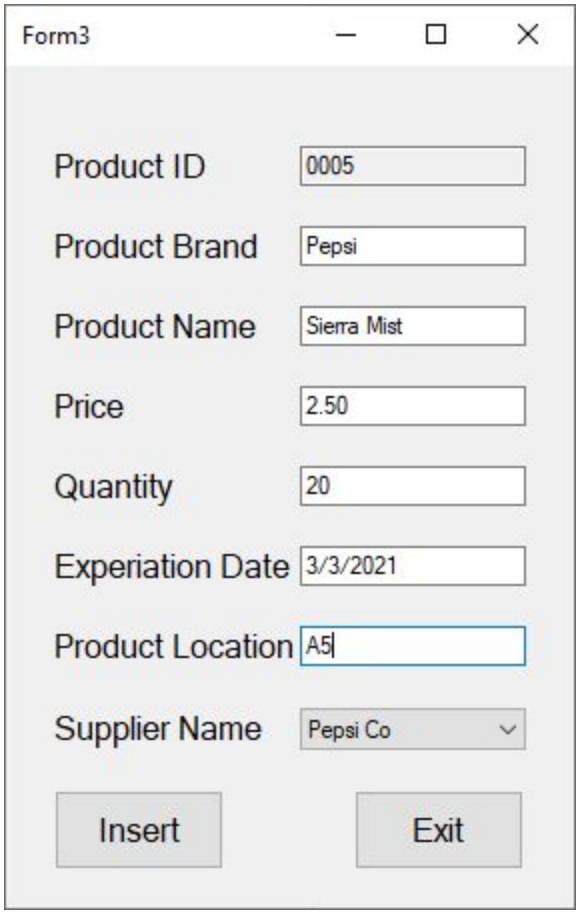


User Story 4: Add Product

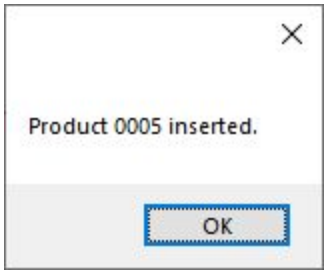
1. Click “Insert” drop down menu at the top left of the main inventory screen.



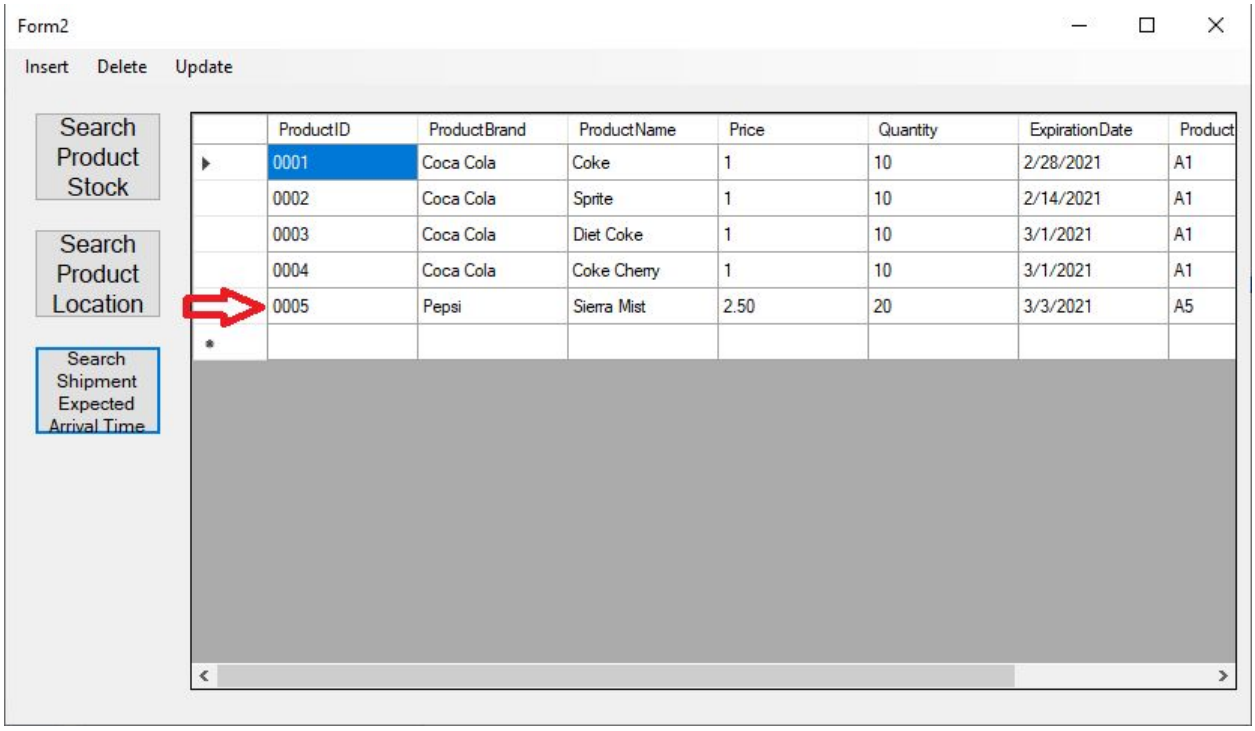
2. Click “Insert Product”.



3. Fill in each field with the appropriate information.
4. Supplier Name is filled with a drop down menu.
5. When all fields have been filled, click “Insert”.
6. If everything was inputted correctly, a confirmation pop-up will appear:

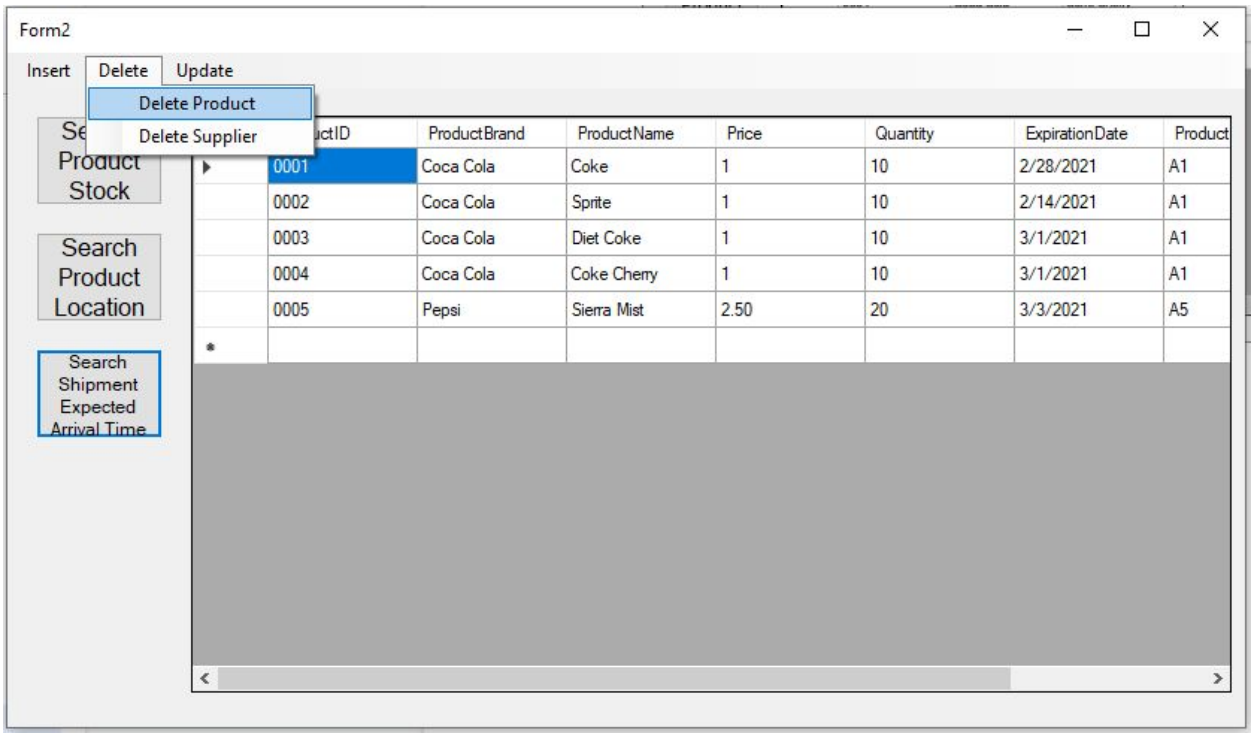


7. The product will now appear in the database:

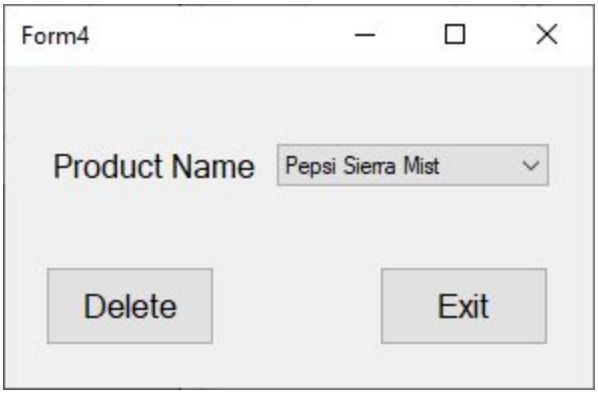


User Story 4: Delete Product

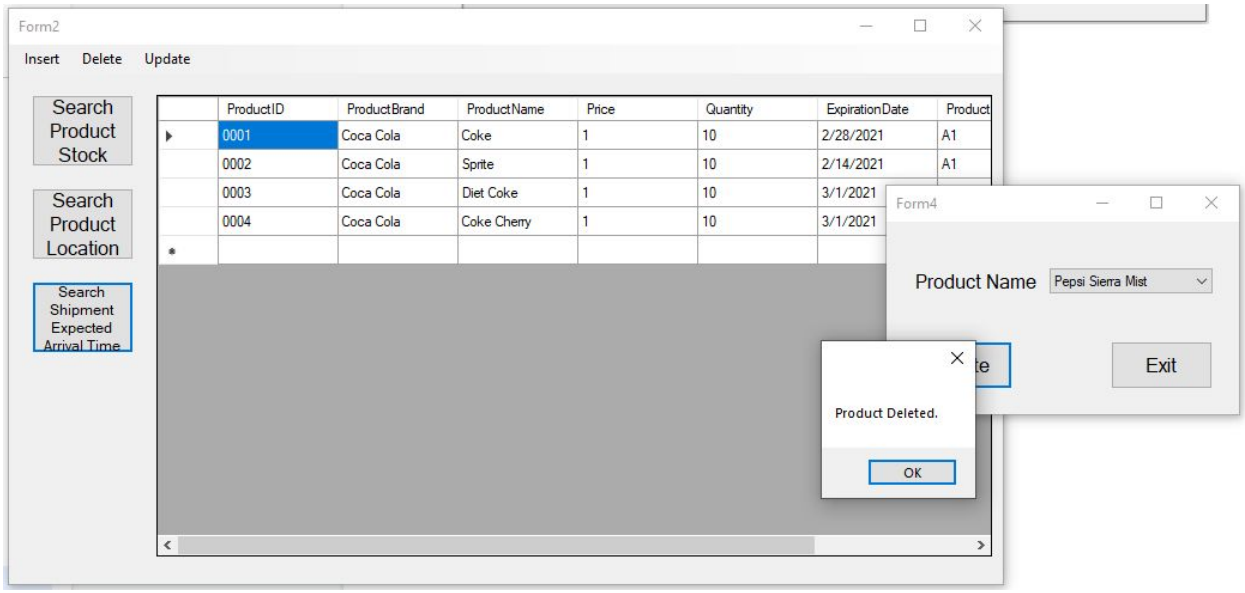
1. Click on the “Delete” drop down menu.
2. Click “Delete Product”.



3. The “Delete Product” page will appear:

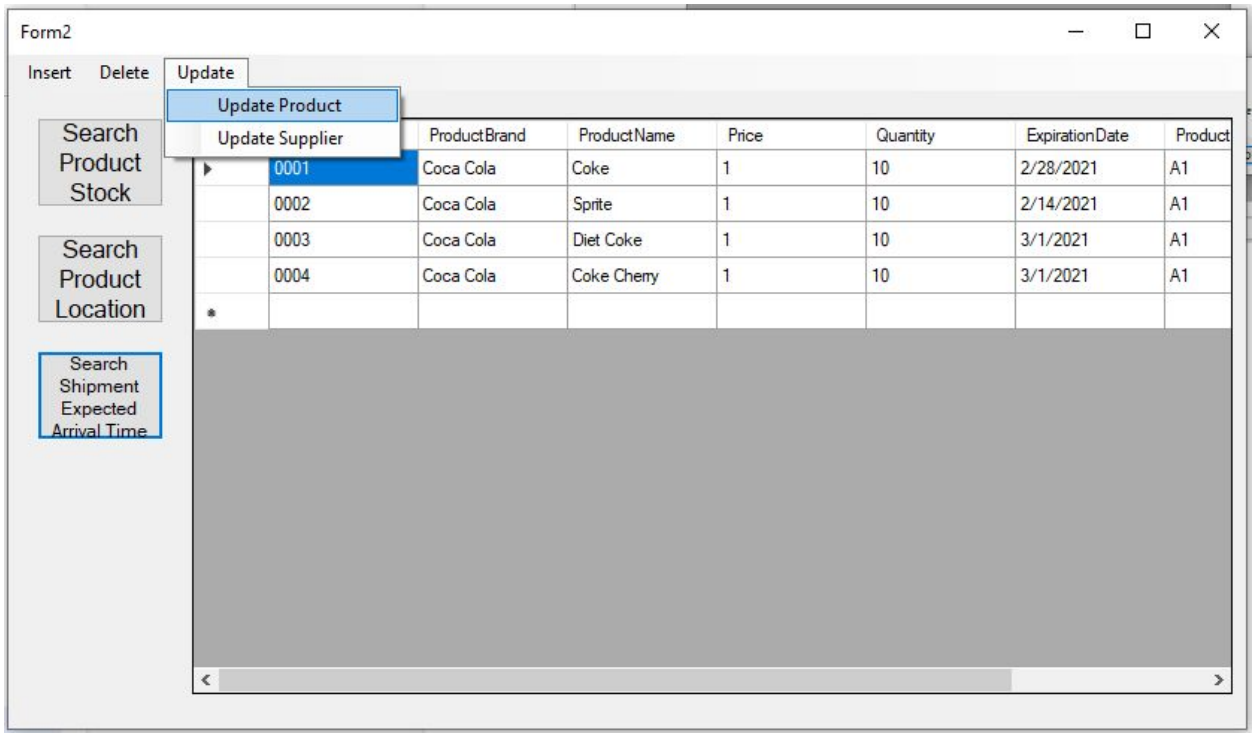


4. Select the name of the product to be deleted in the drop-down menu.
5. Click the “Delete” button.
6. A confirmation box will appear if the deletion is successful, and the item is removed from the database.

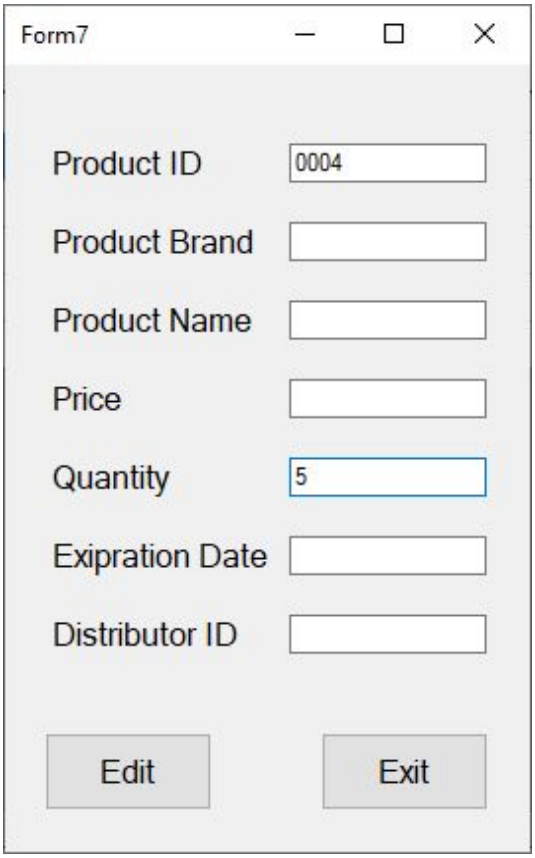


User Story 4: Update Product

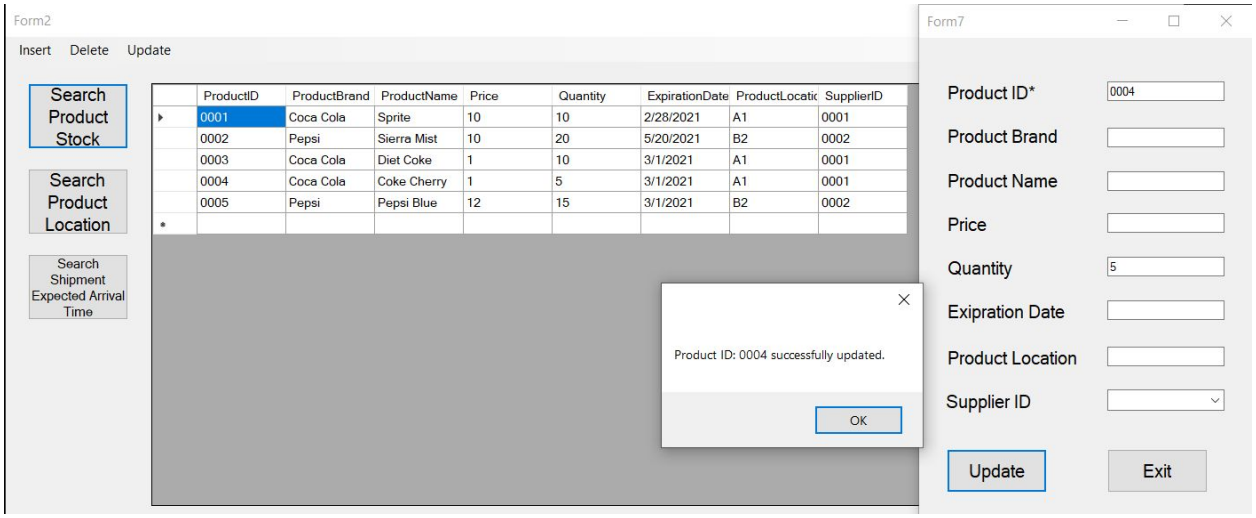
1. Click on the “Update” drown down menu.
2. Select “Update Product”.



3. Input the Product ID for the product that will be modified.
4. Fill in any fields that need to be modified in the product entry. Fields can be left blank if no change is to be made for that portion of the product's entry.

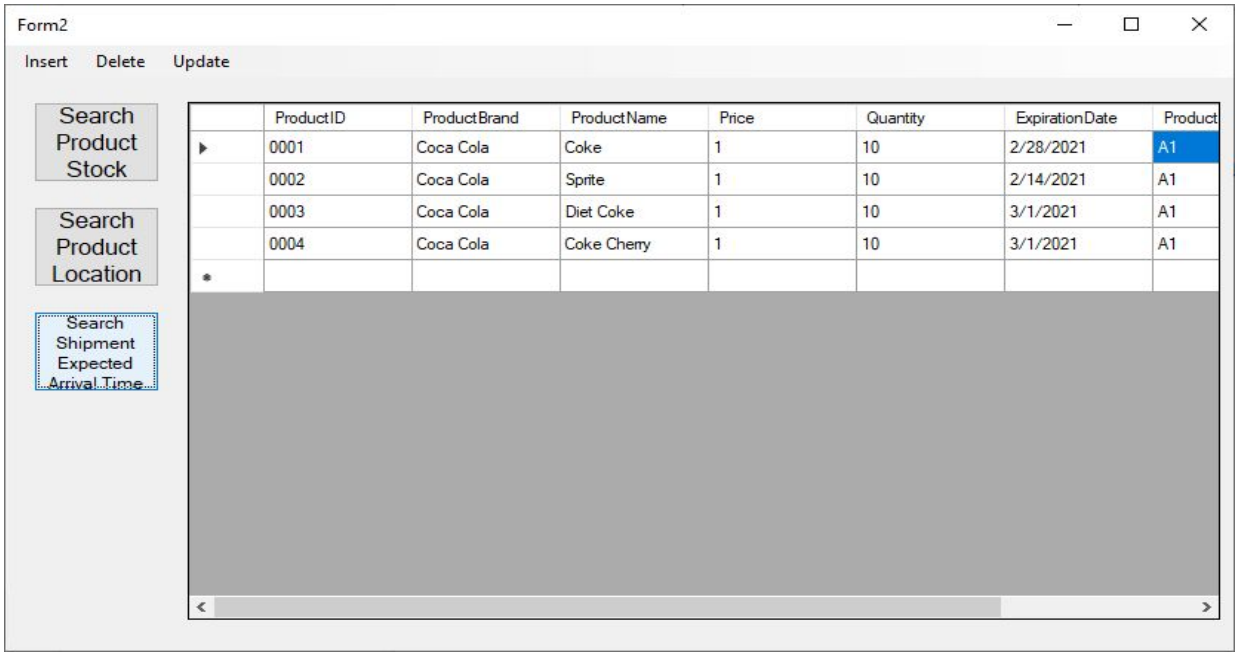


5. Click the “Update” button, and the entry for that product will be modified.

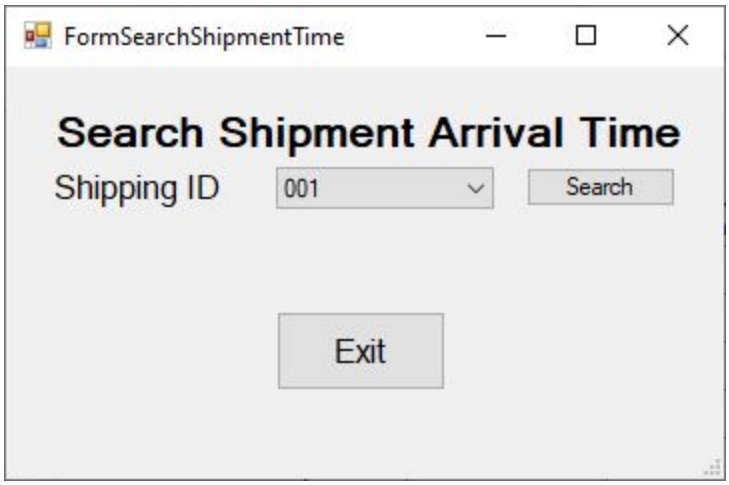


User Story 5: Search Shipment Times

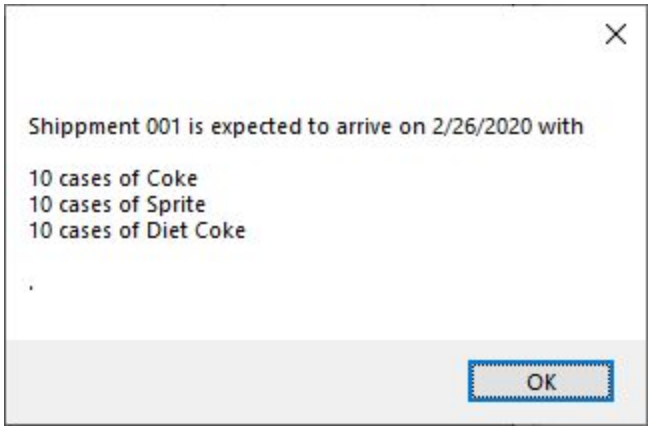
1. Click the “Search Shipment Expected Arrival Time” on the main inventory screen.



2. Select the shipment ID to look up in the drop down menu.

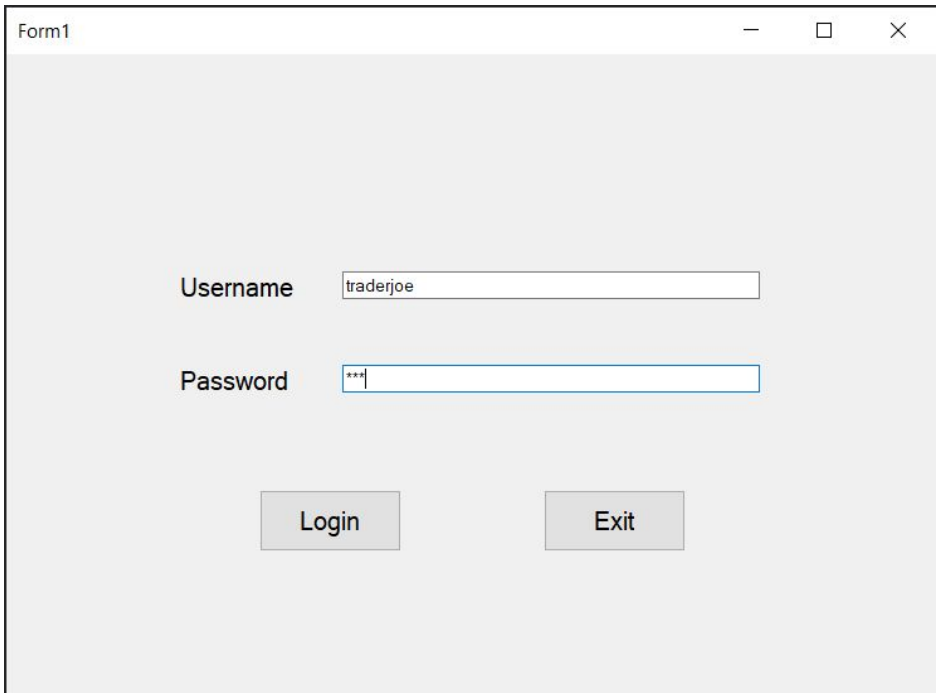


3. Click the “Search” button.
4. A dialogue will appear displaying the contents and arrival time for that shipment ID.

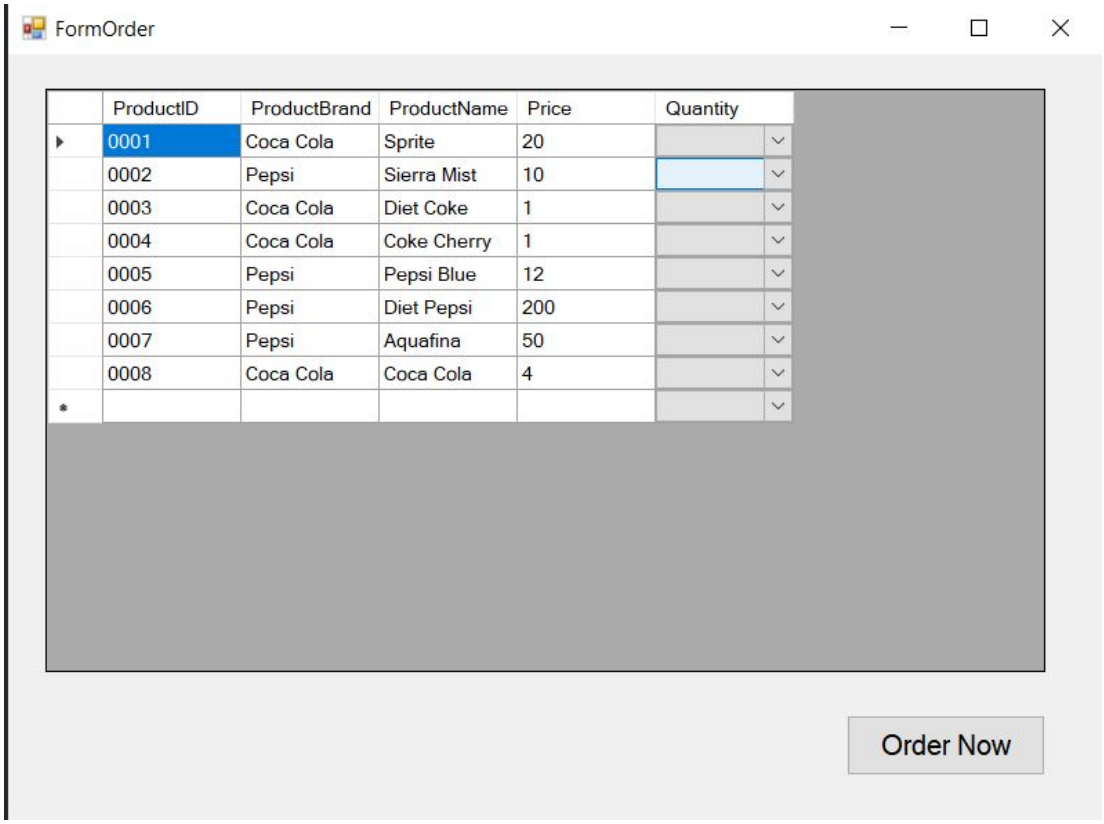


User Story 6: Customer Order

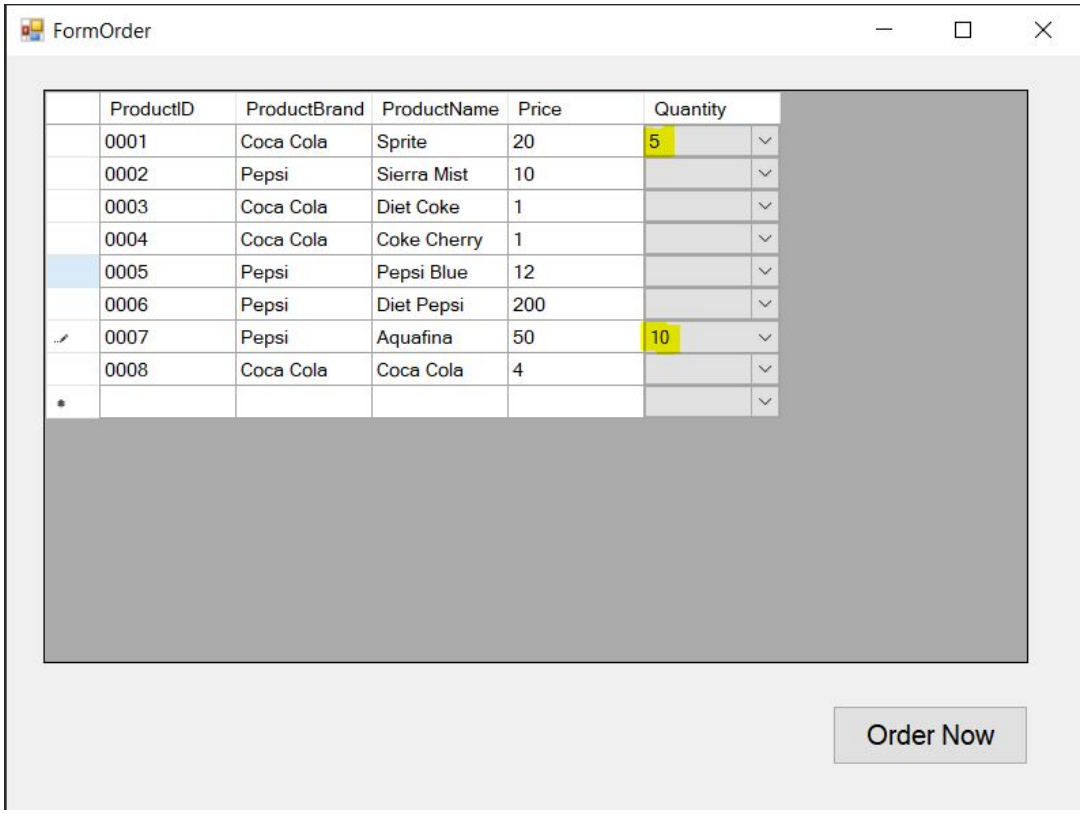
1. Users must login using a customer account.



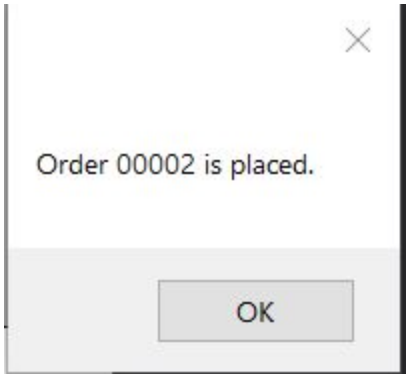
2. If the login was successful then the system will prompt the user with the FormOrder. The Form will have all the products available within the warehouse.



3. The user will then select the amount of each product they are willing to buy, but clicking on the quantity dropbox.



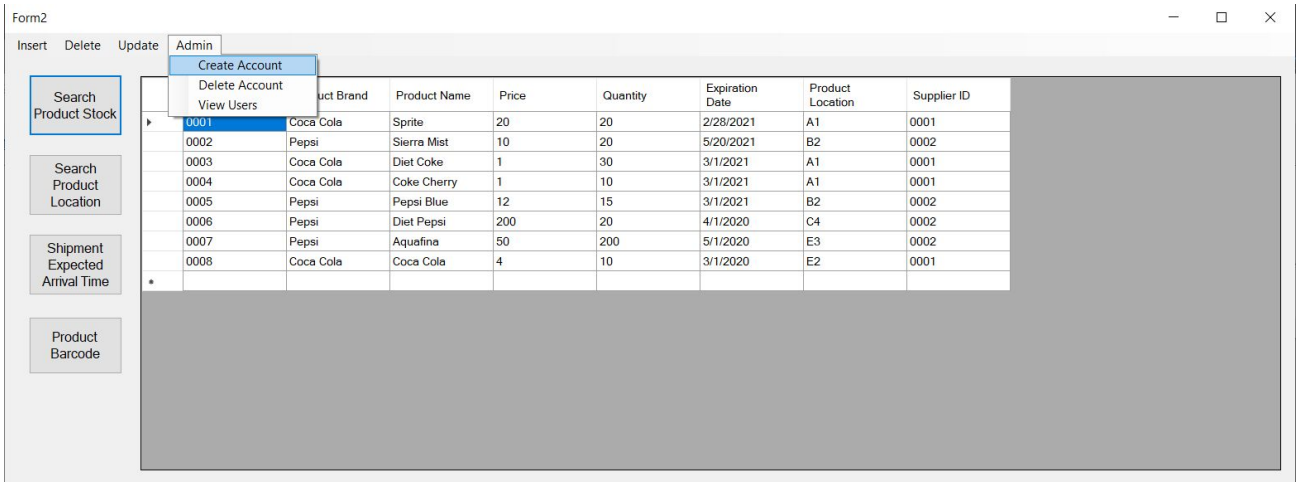
4. Once the user is satisfied with their choices, they may click “Order Now.” Then the system will prompt the customer with a confirmation message.



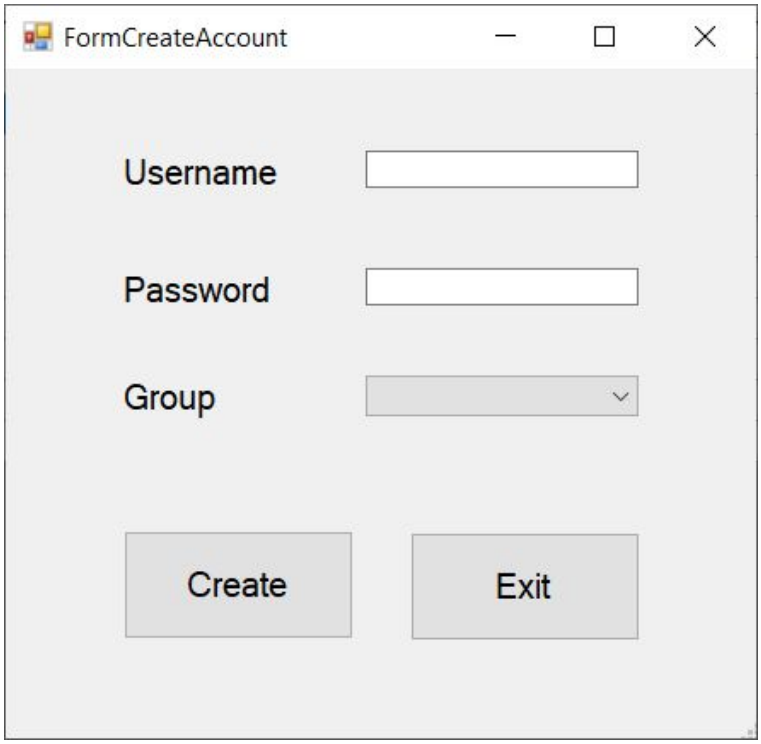
User Story 7: Add Accounts

1. Users must login with an administrator account.

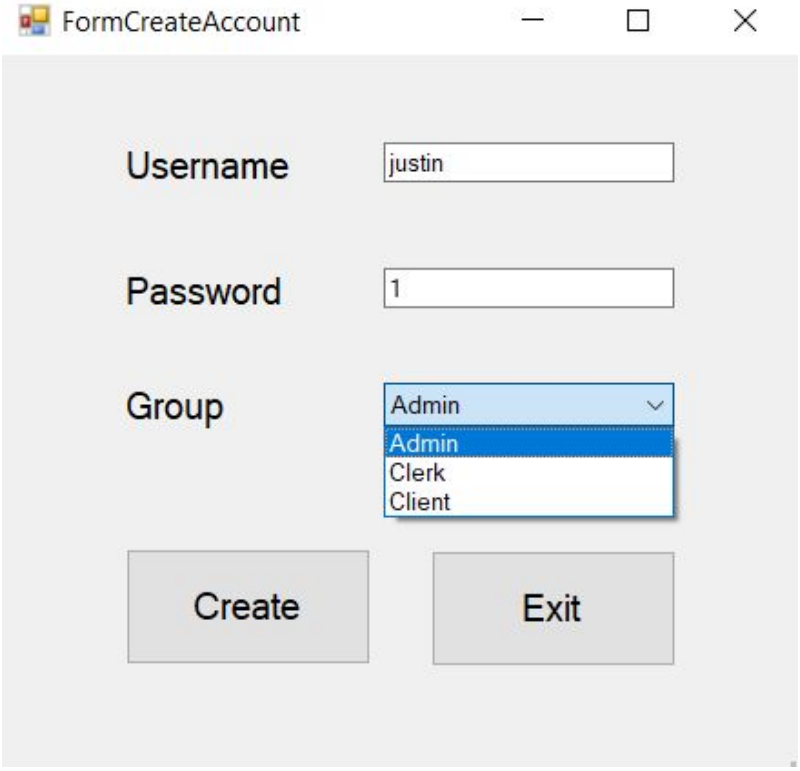
2. Select “Admin” and Click on “Create Account”



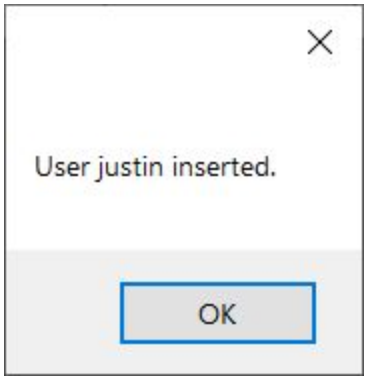
3. System will then display the FormCreateAccount



4. User then inputs the username, password and selects the type of user group.



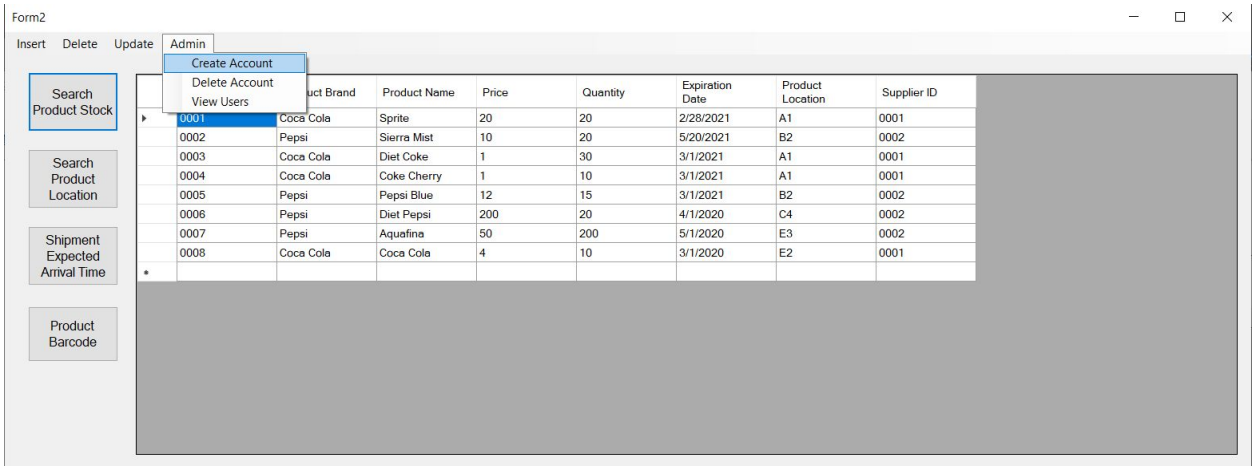
5. User then selects “Create” and the system prompts the user with a confirmation message.



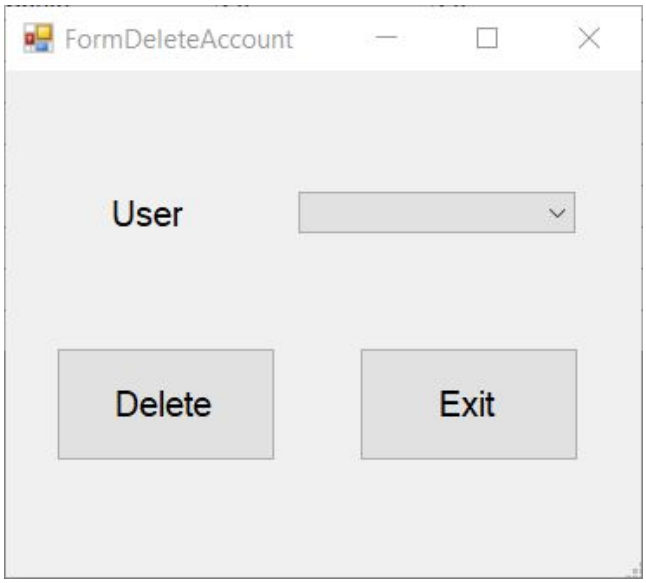
User Story 7: Delete User

1. Users must login with an administrator account.

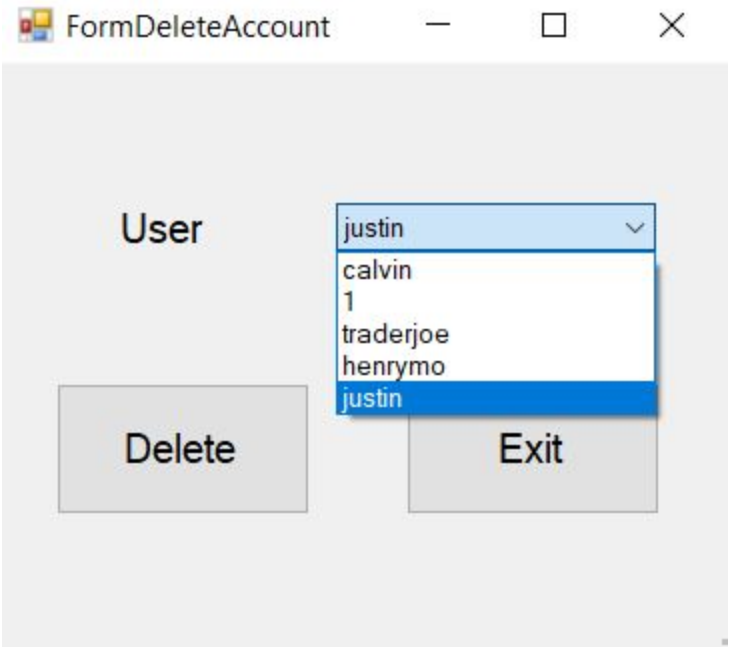
2. Select “Admin” and Click on “Delete Account”



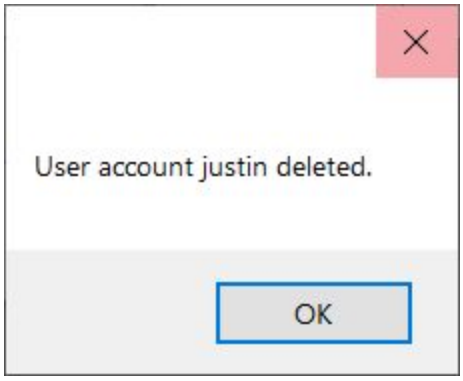
3. System will then display the FormDeleteAccount



4. User then selects the user they want to delete using the dropdown box.



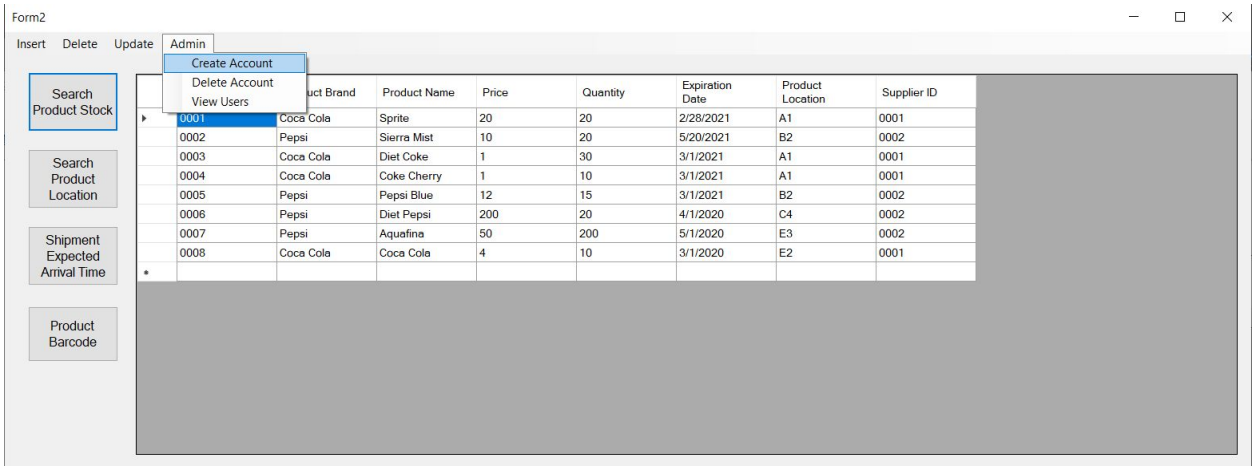
5. User selects “Delete” and the system prompts them with a confirmation message.



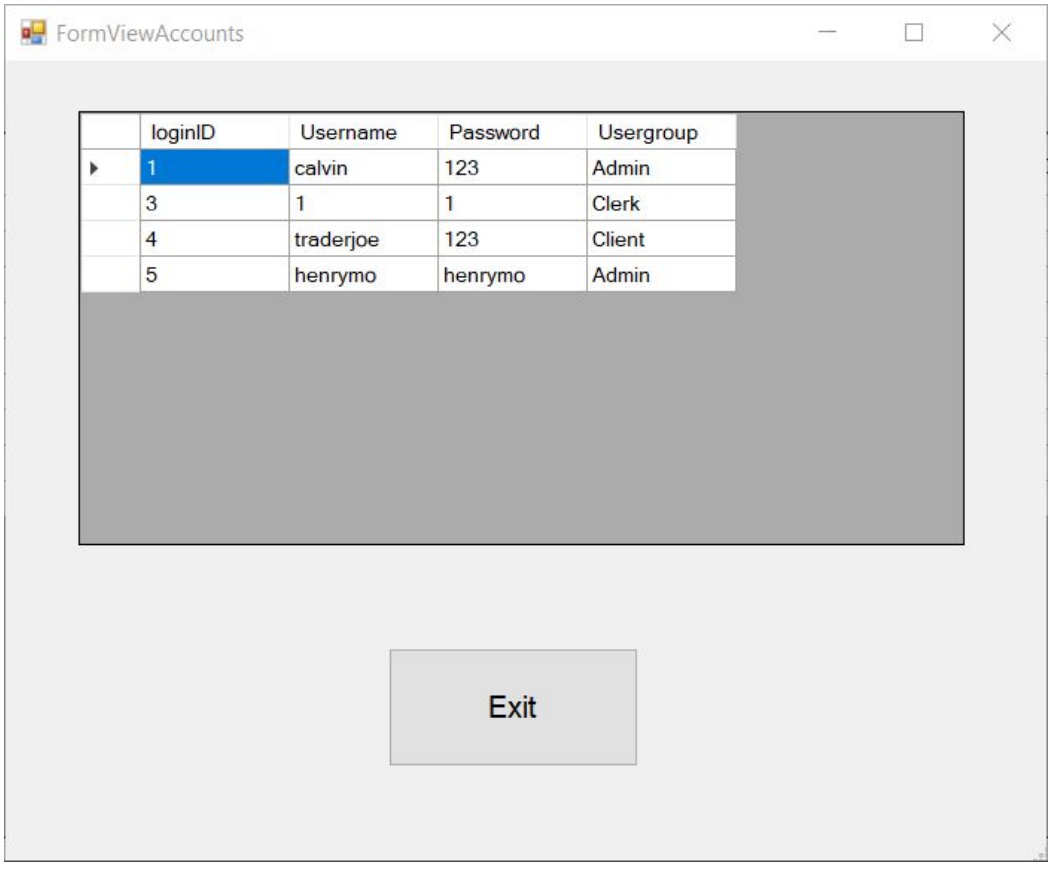
User Story 7: View Users

1. Users must login with an administrator account

2. Select “Admin” and Click on “View Users”



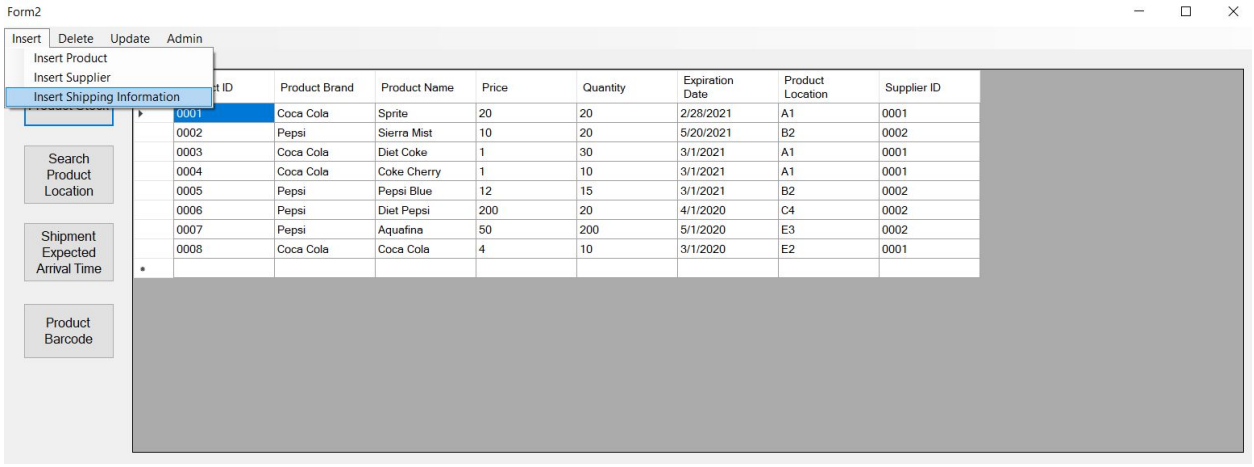
3. The system will display the FormViewAccounts where you can see all the users apart of the system.



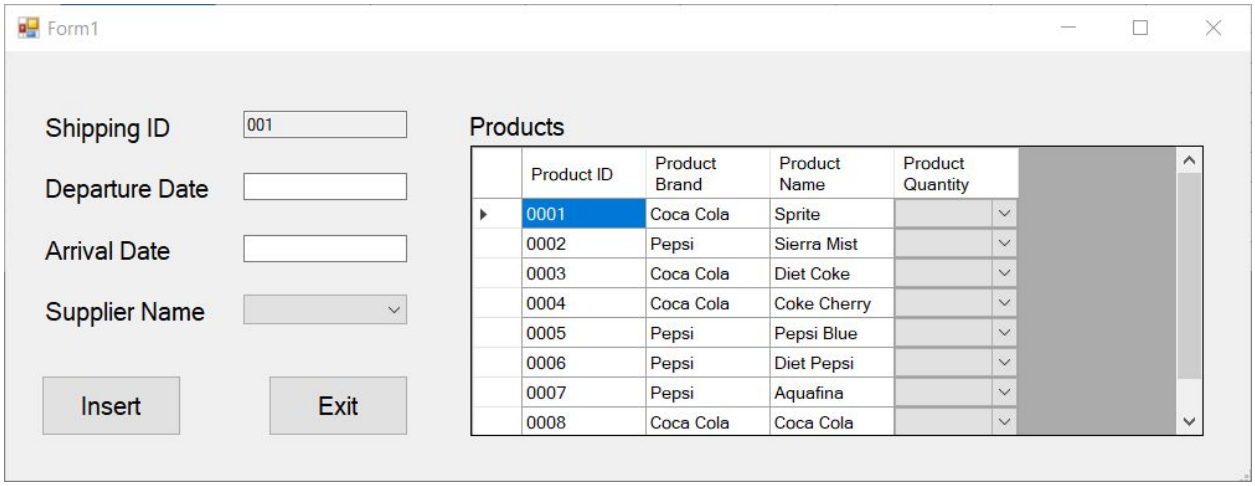
User Story 8: Insert Shipments

1. Users must login with a clerk or administrator account.

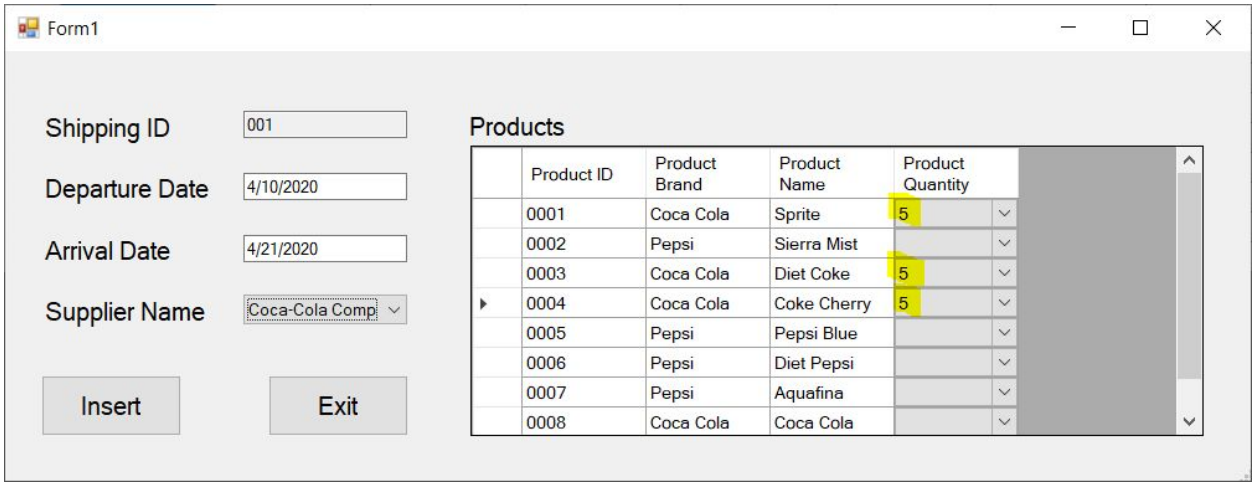
2. Select “Insert” and click on “Insert Shipping Information”



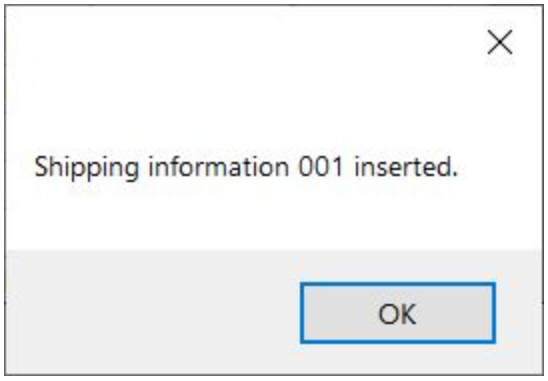
3. The system will display FormShipping where the user will input the Departure Date, Arrival Date, and Supplier.



4. The user will also select the amount of a product that is being sent to the warehouse.

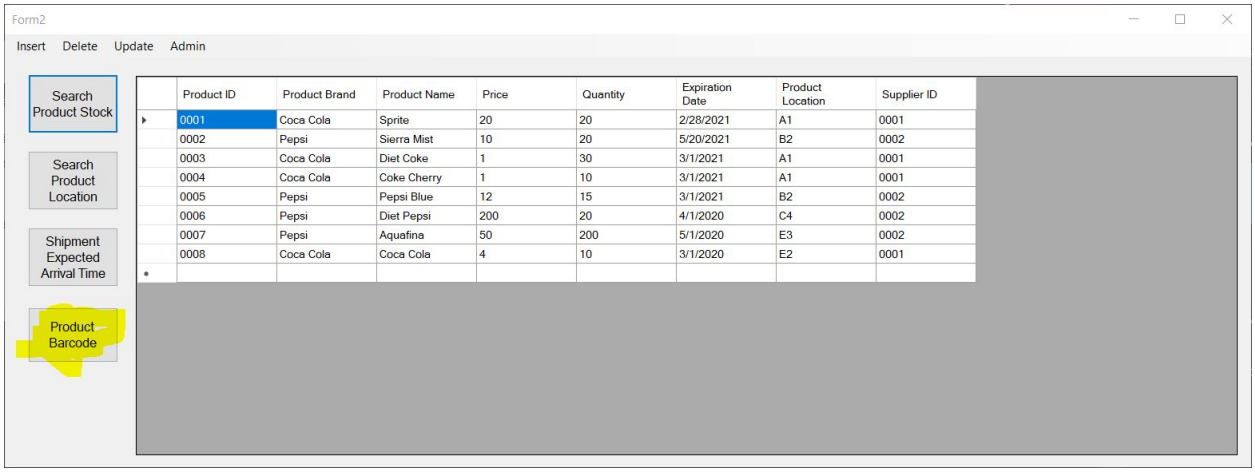


5. When the user is satisfied with their shipment information, they can click insert and the system will prompt them with a confirmation message.

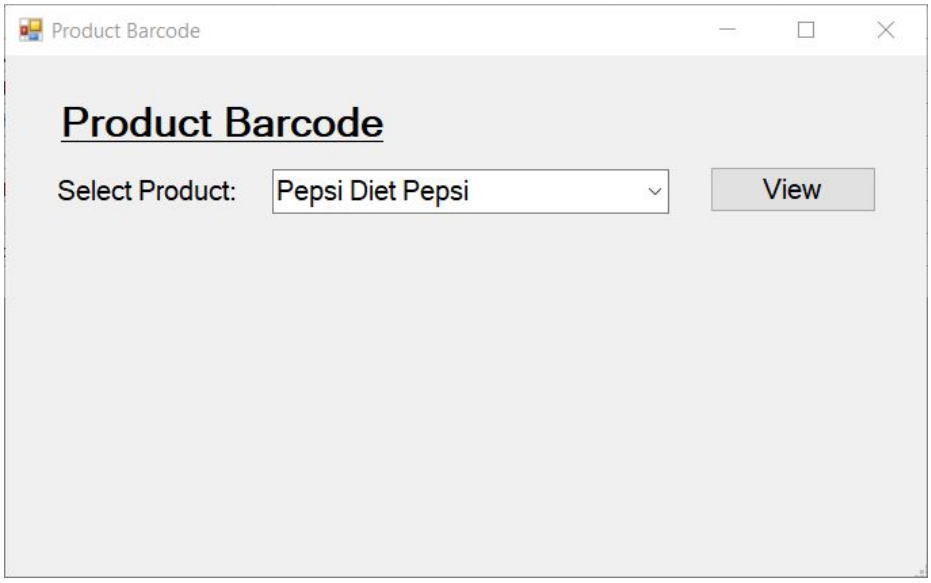


User Story 9: Barcodes

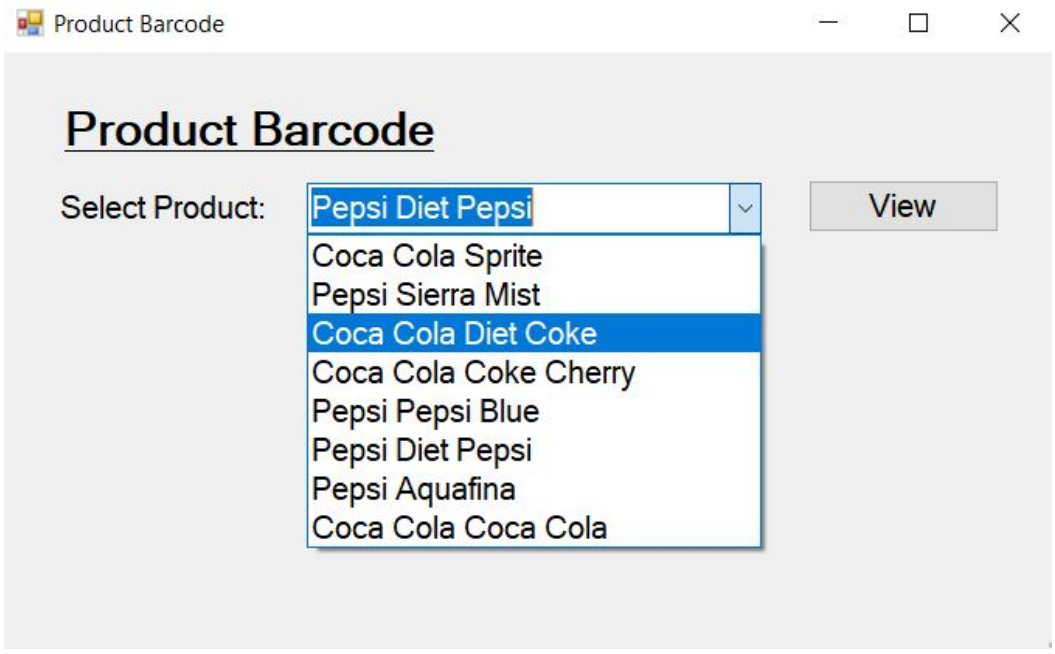
1. Users must login with a stocker/picker, clerk, or administrator account.
2. Click on “Product Barcode”



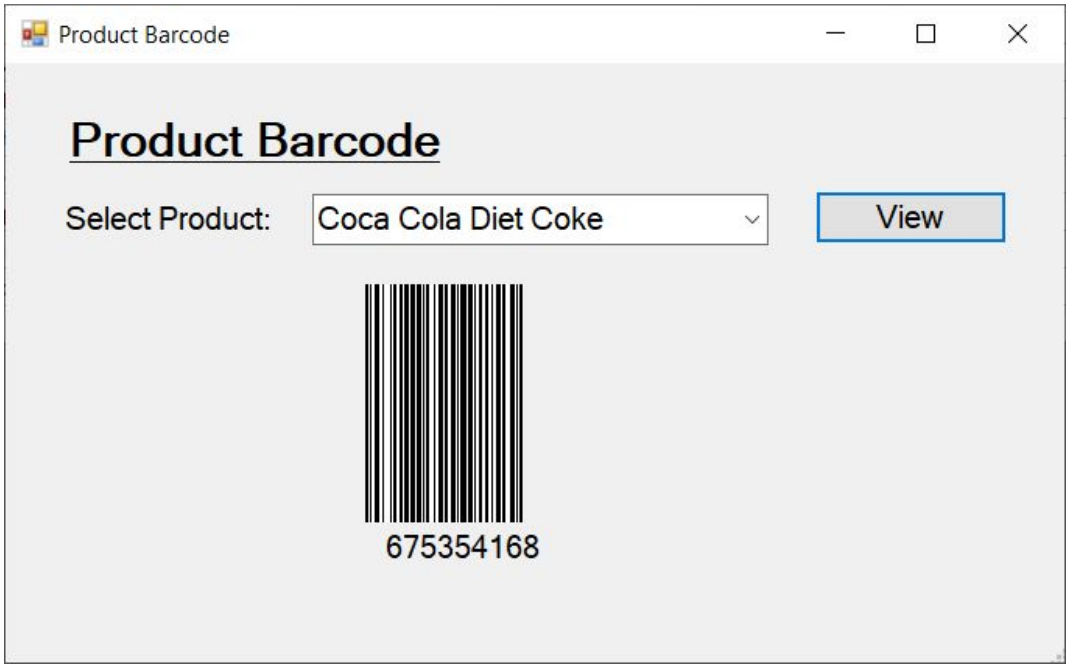
3. System will display FormProductBarcode.



4. Users will then select the desired product using the dropdown box.

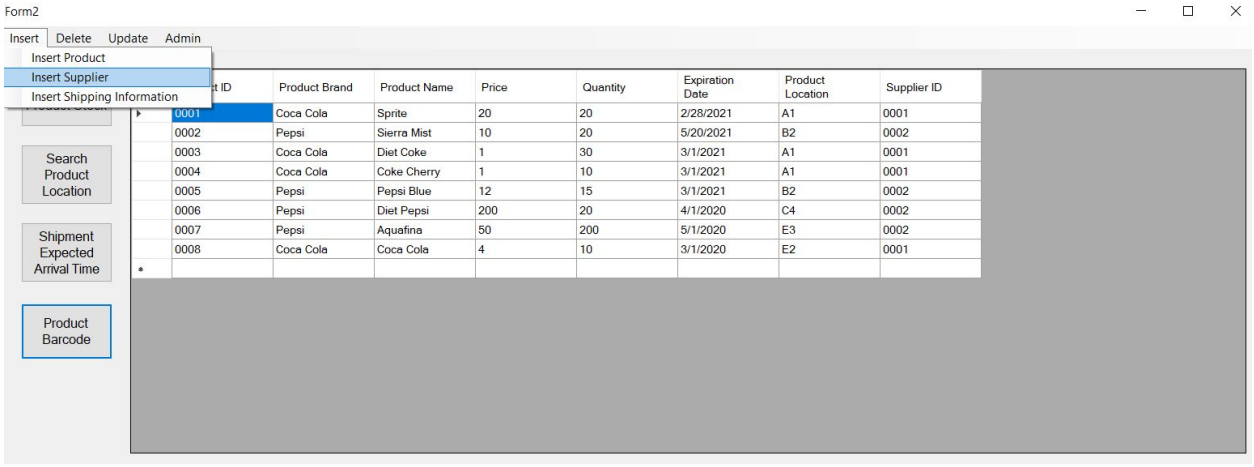


5. Click “View” and the system will prompt them with the associated barcode.

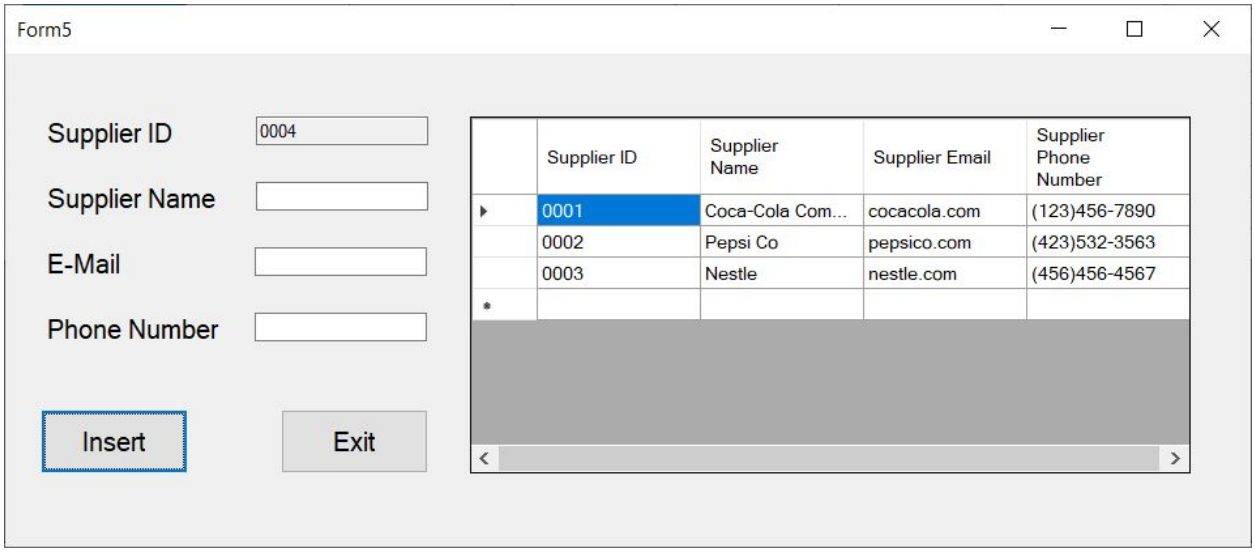


User Story 10: Add Suppliers

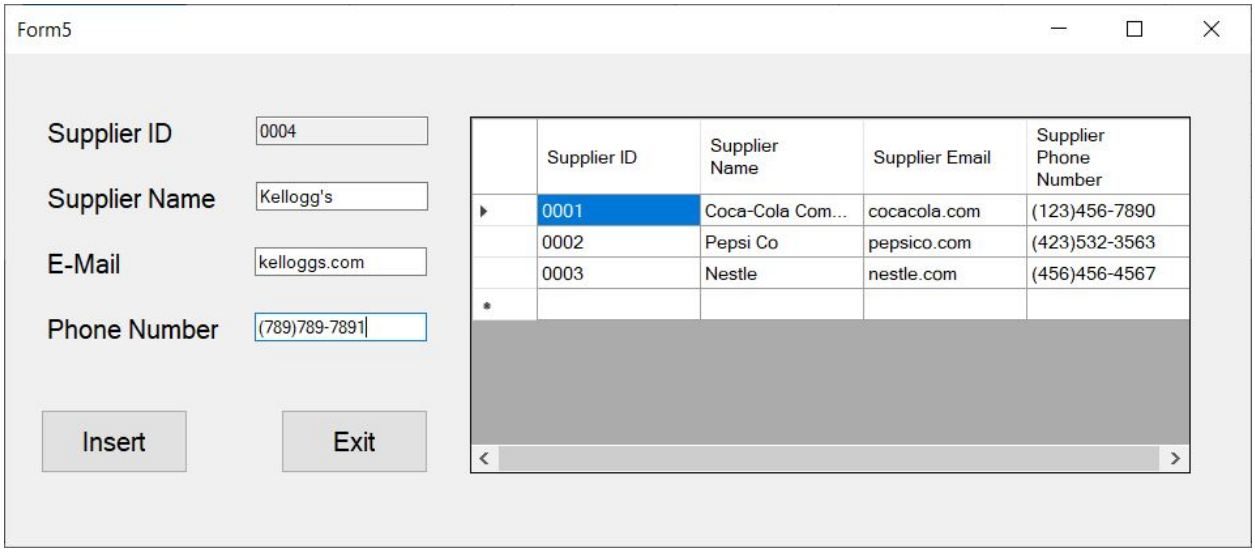
1. Users must login with a clerk or administrator account.
2. Select “Insert” and Click on “Insert Supplier”.



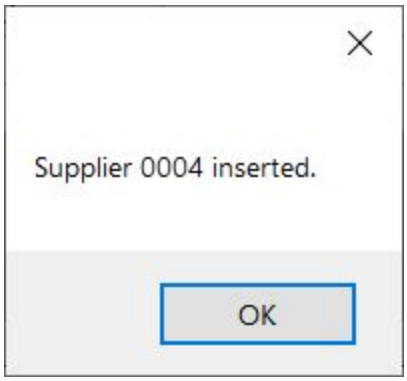
3. System will display the FormInsertSuppler.



4. User will input the Supplier name, Email, and Phone Number.

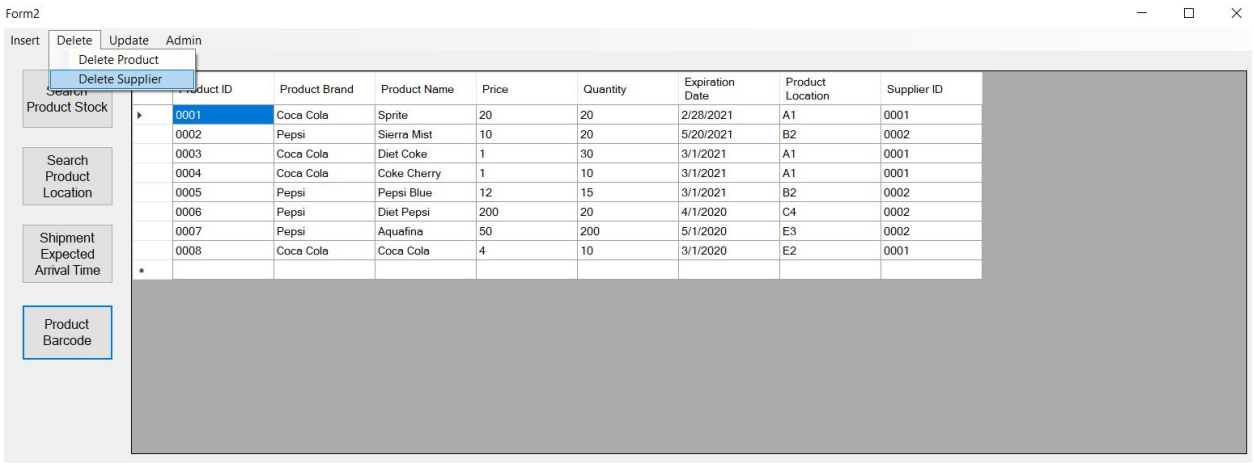


5. Click “Insert” and the system will prompt the user with a confirmation message,

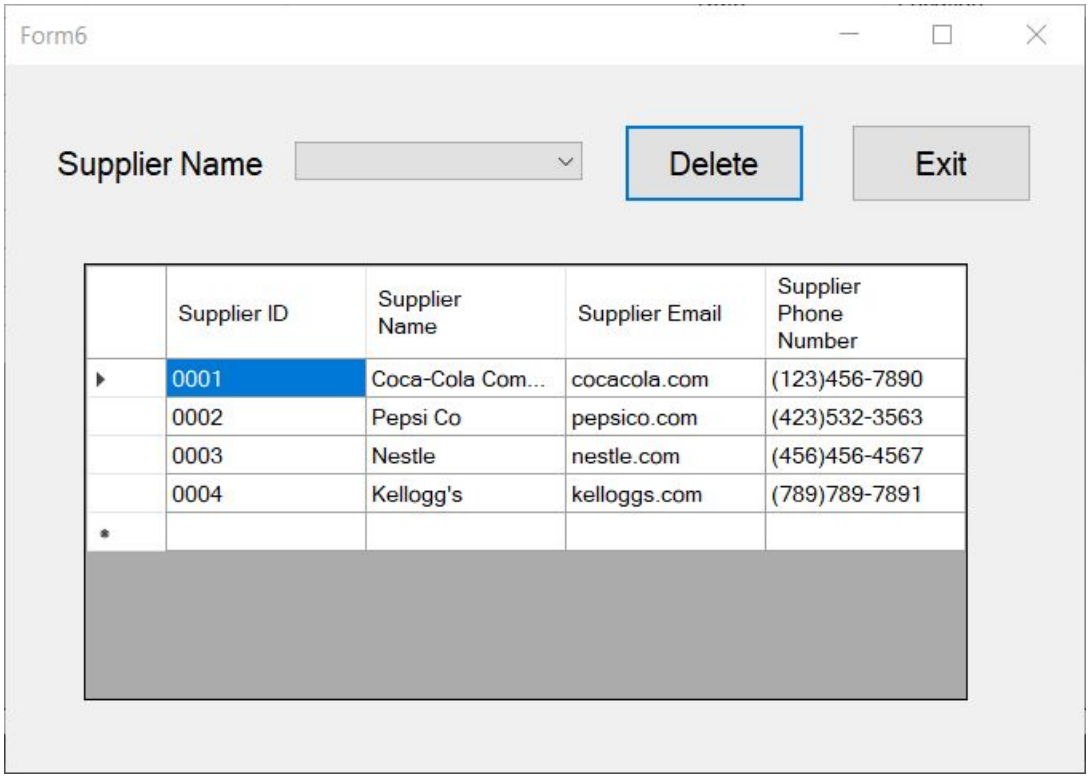


User Story 10: Delete Suppliers

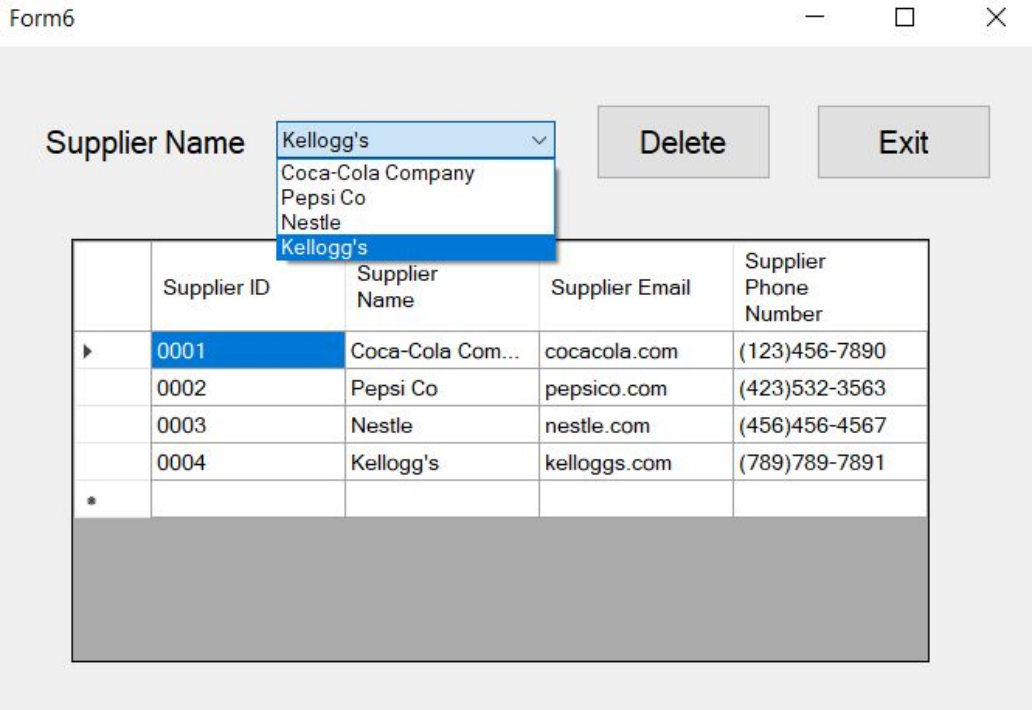
1. Users must login with a clerk or administrator account.
2. Select “Delete” and Click on “Delete Supplier”.



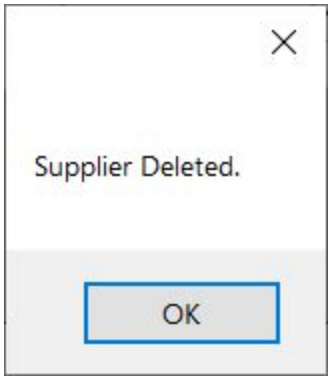
3. System will display FormDeleteSupplier.



4. Users will select the desired supplier to be deleted with the dropdown box.

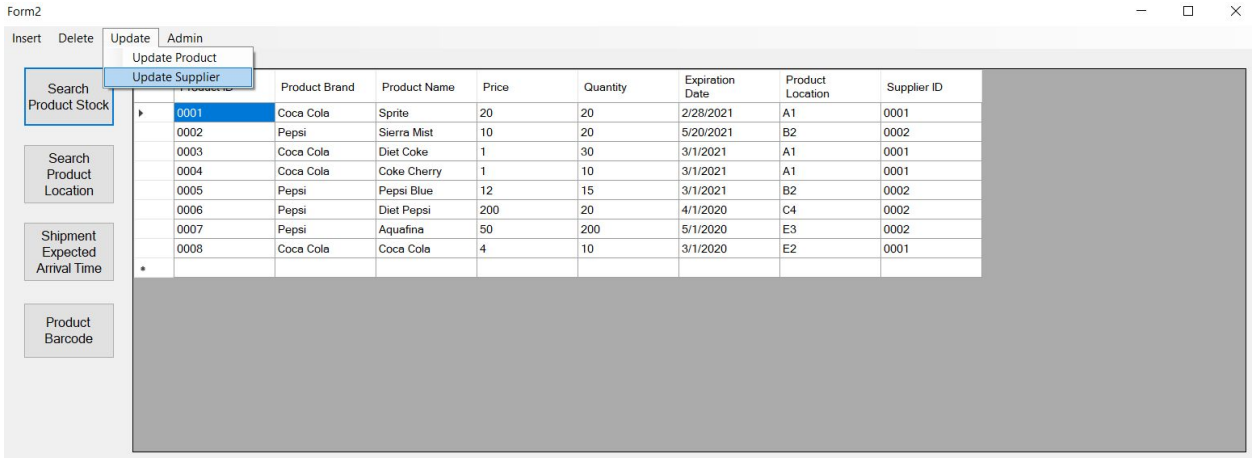


5. Click “Delete” and the system will prompt the user with a confirmation message.

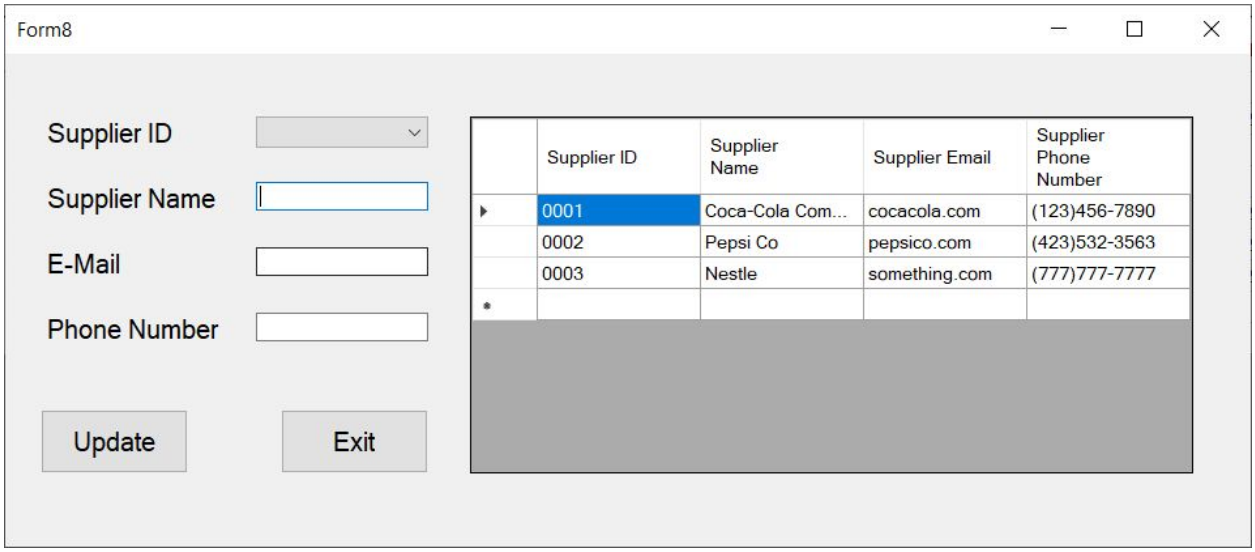


User Story 10: Update Suppliers

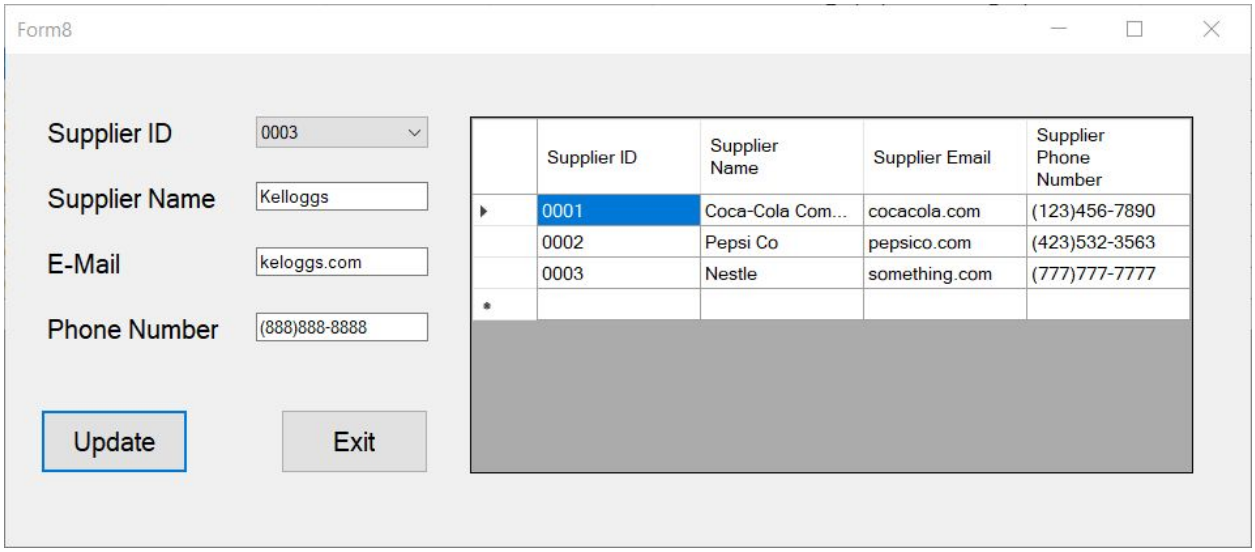
1. Users must login with a clerk or administrator account.
2. Select “Update” and Click on “Update Supplier”.



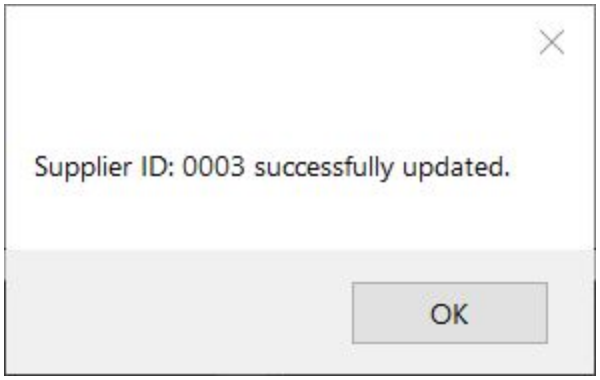
3. System will display FormUpdateSupplier.



4. Users will select the supplier with a dropdown box. Users can then input a new Name, Email, or Phone Number.

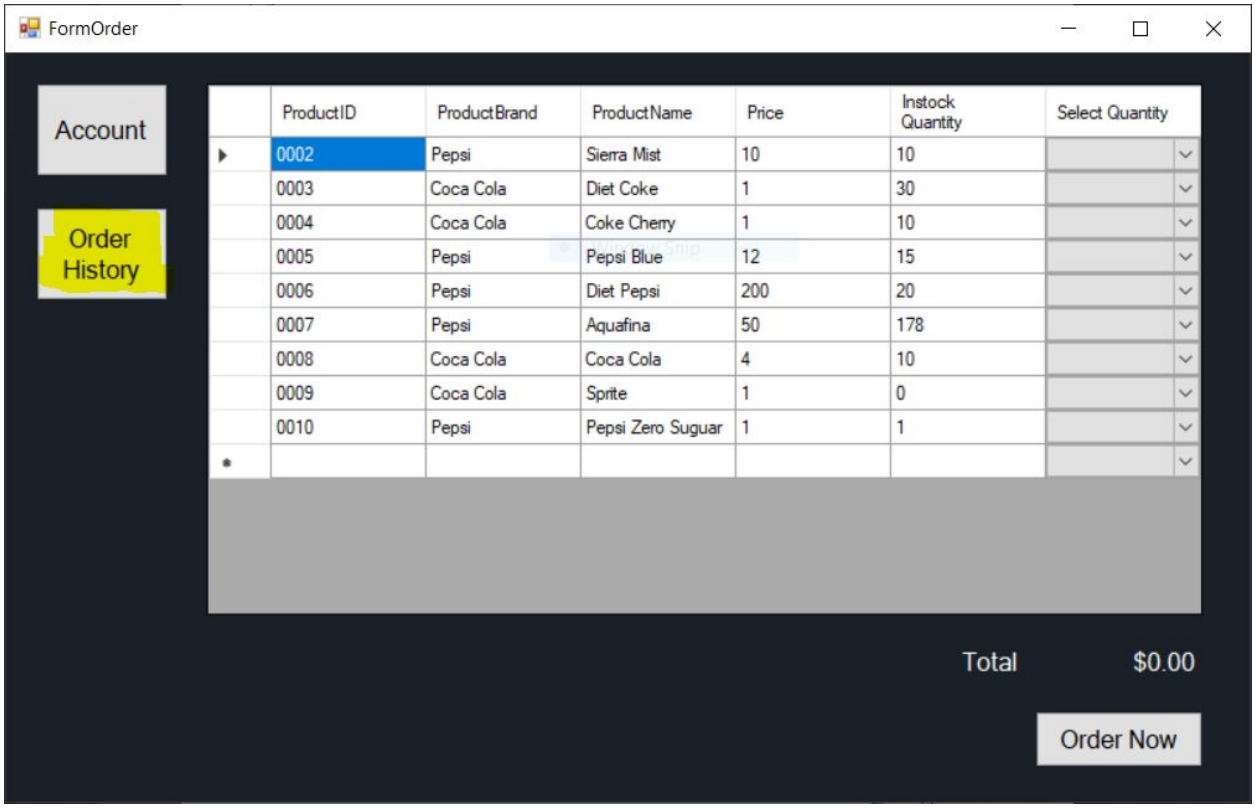


5. Click “Update” and the system will prompt the user with a confirmation message.

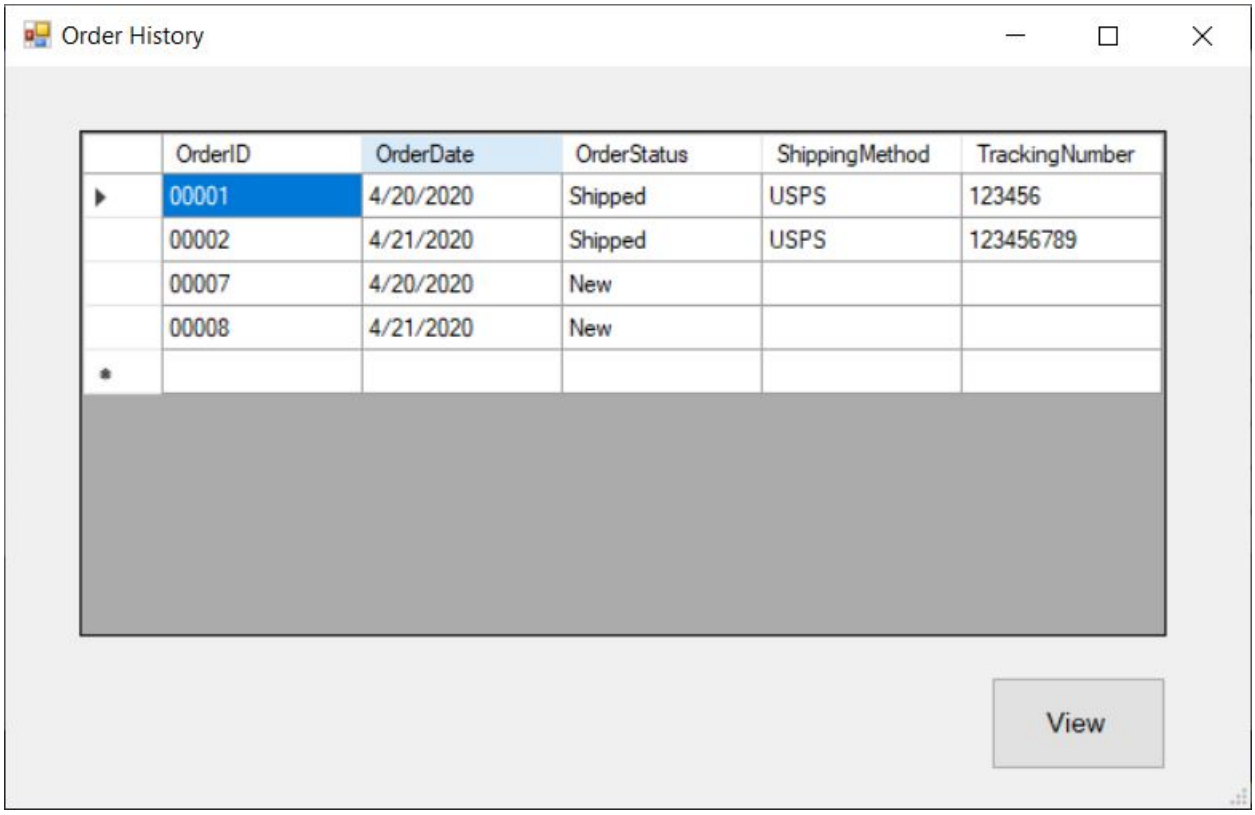


User Story 11: Client Order History

1. Users must login with a client account
2. Click on “Order History”

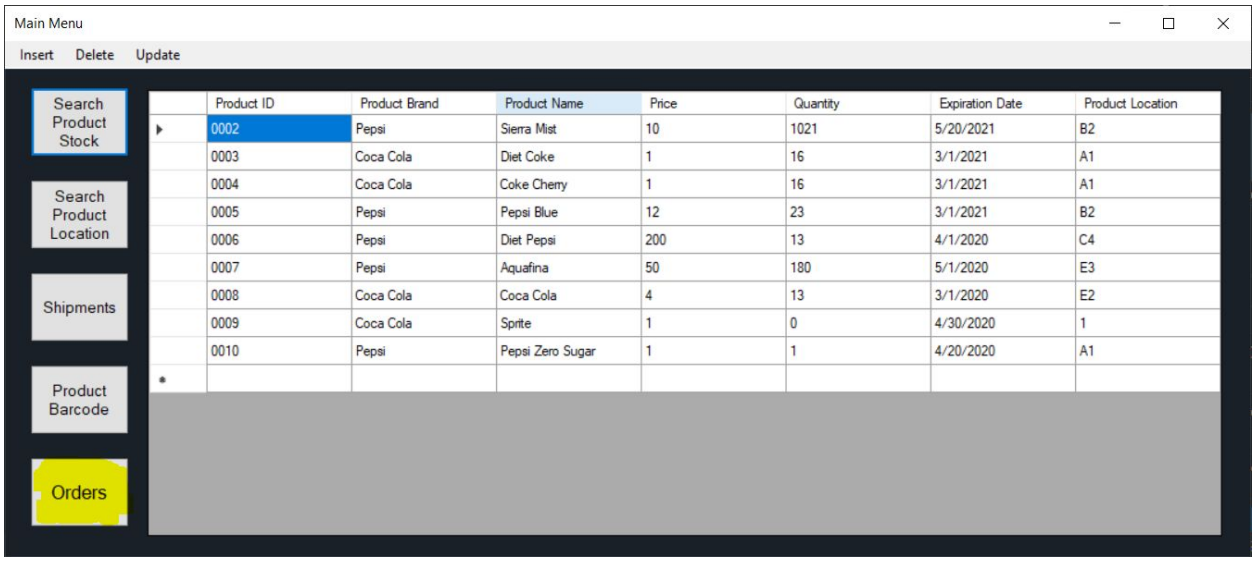


3. System will display FormOrder

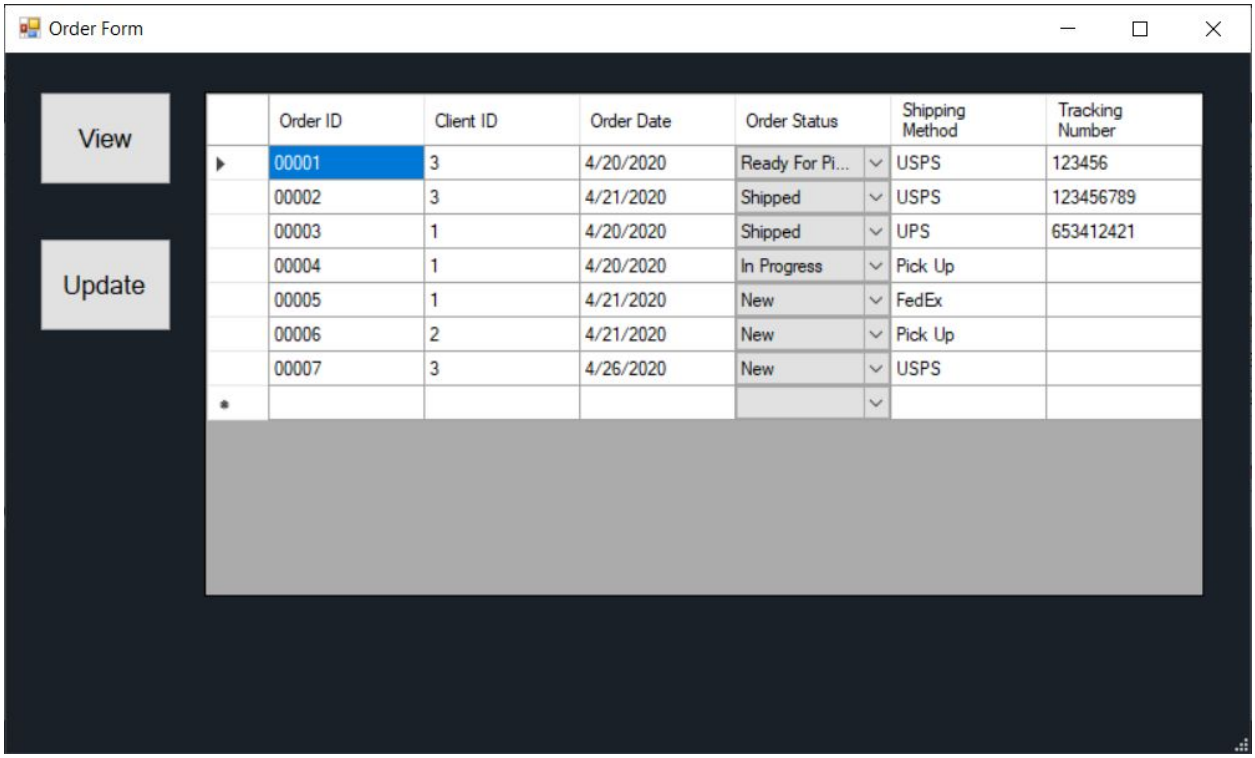


User Story 12: Invoice

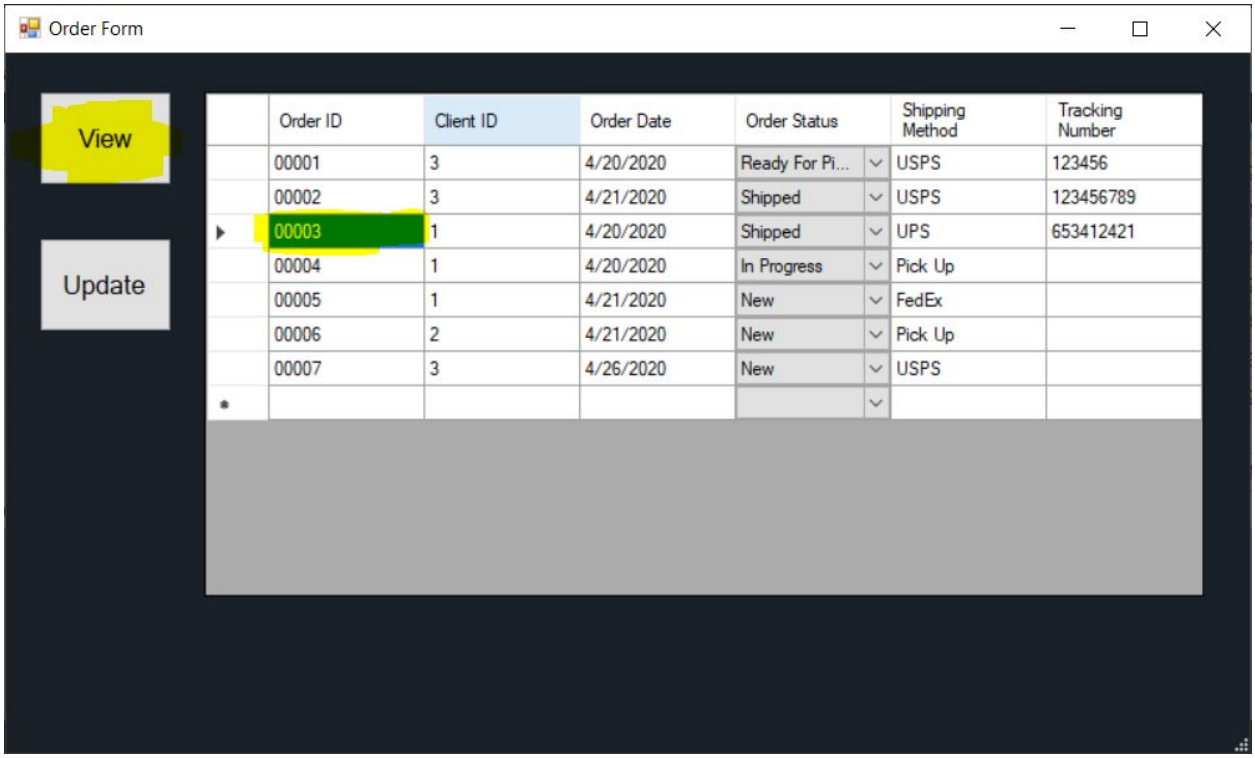
1. Users must login with a clerk or administrator account.
2. Click on “Orders”



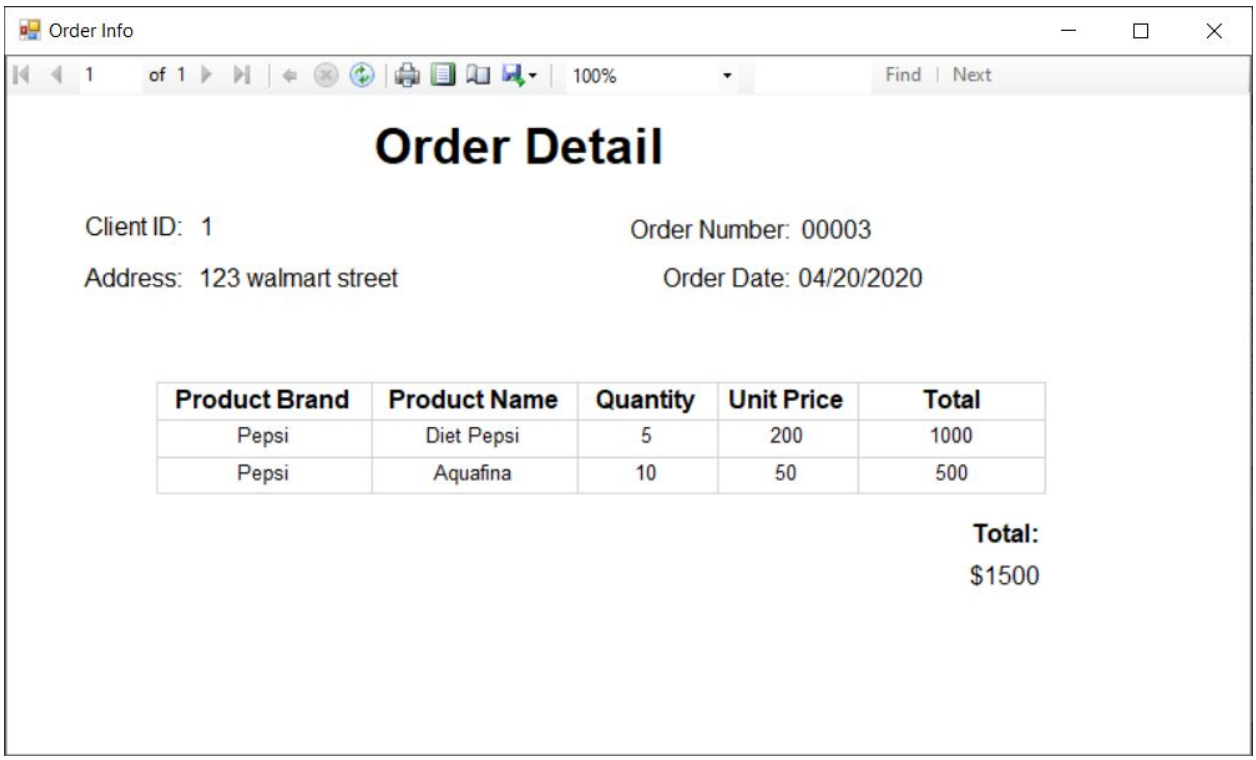
3. System will display Order Form



4. User clicks the OrderID number from the column to view an order
5. User clicks “View”

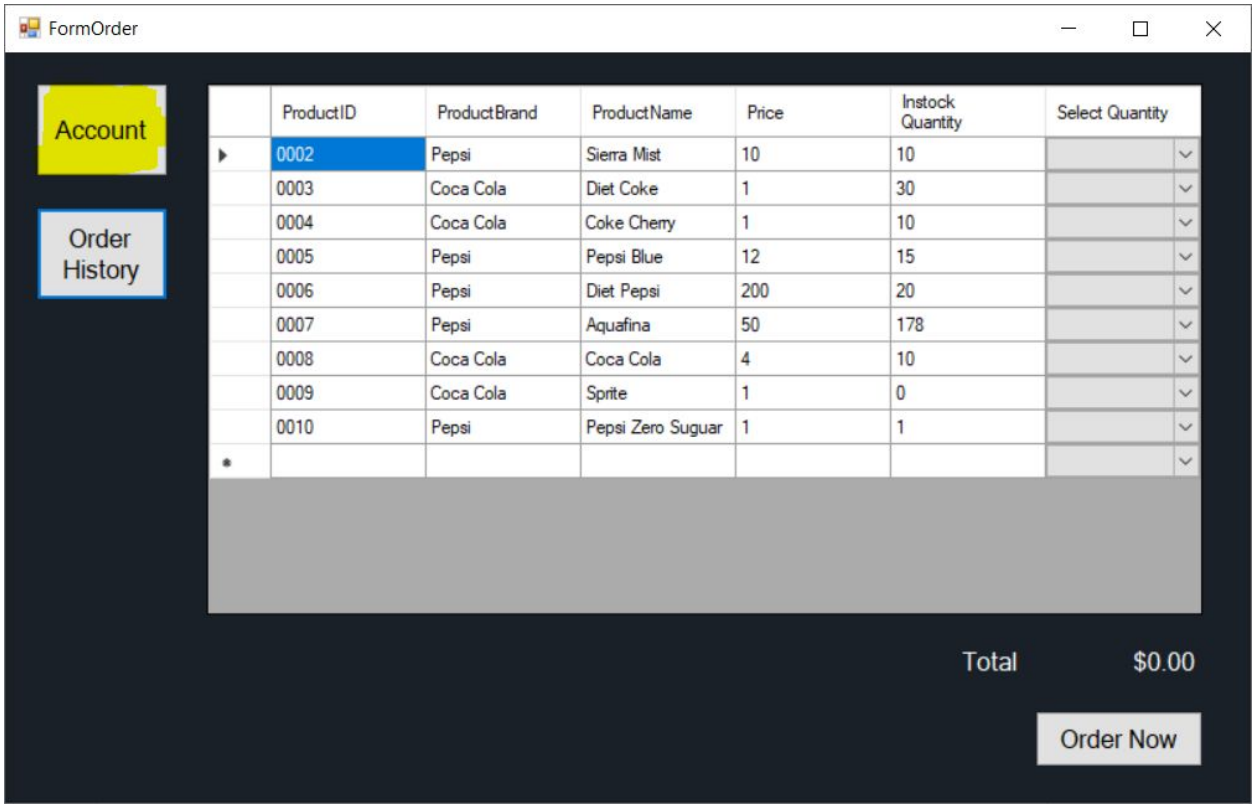


6. System will display the Order info Form

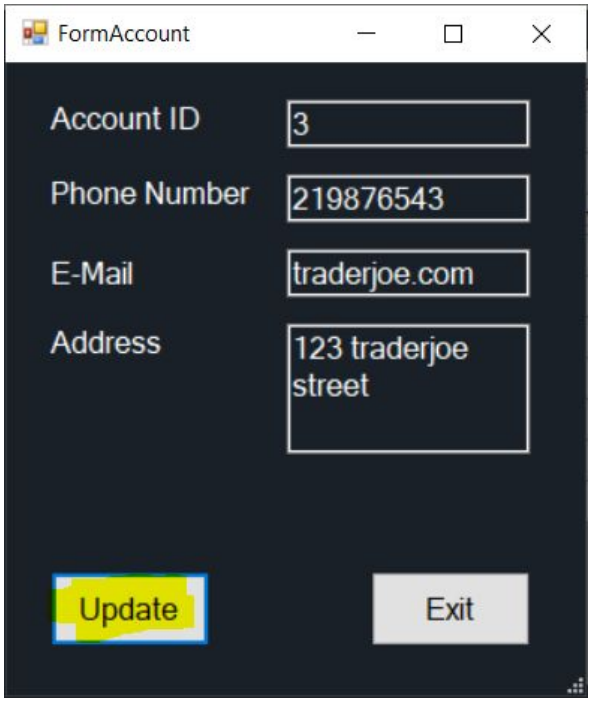


User Story 13: Client Account

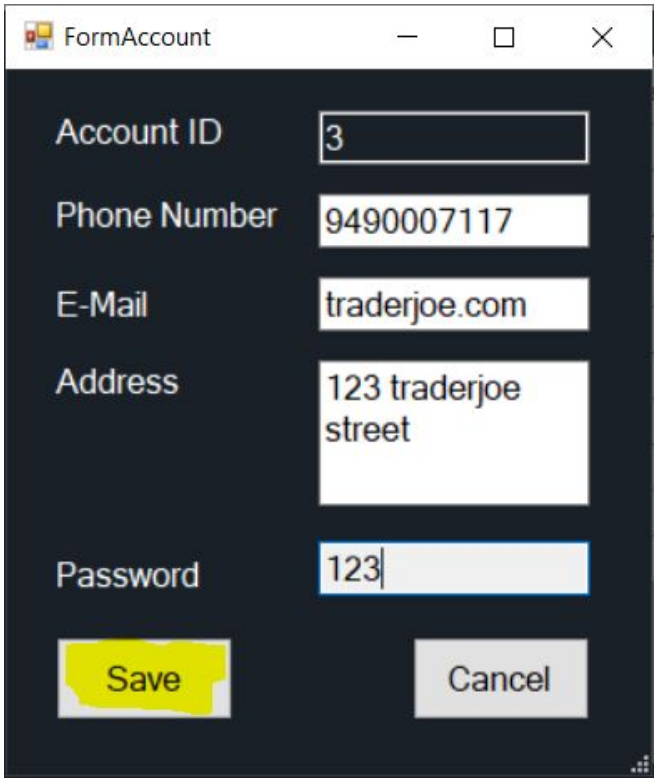
1. Users must login with a client account
2. Click on “Account”



3. System will display the FormAccount
4. Click on “Update”



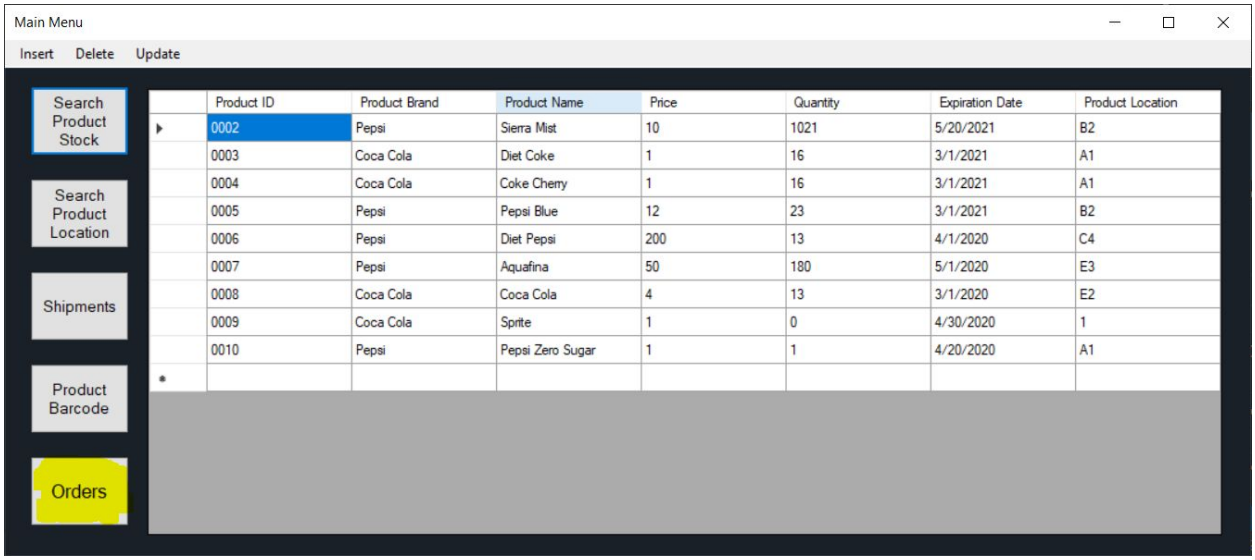
5. System will unlock the information to be modified
6. Users can then input a new phone number, e-mail, or address that is associated with the account
7. User enters the account's password
8. User clicks "Save"



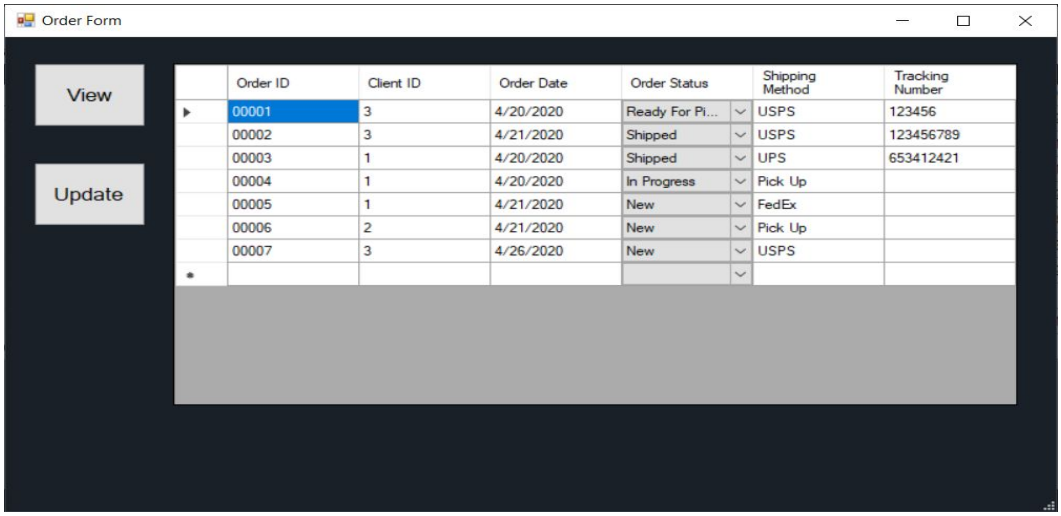
9. system will prompt the user with a confirmation message.

User Story 14: View all orders

1. Users must login with a clerk or administrator account.
2. Click on “Orders”

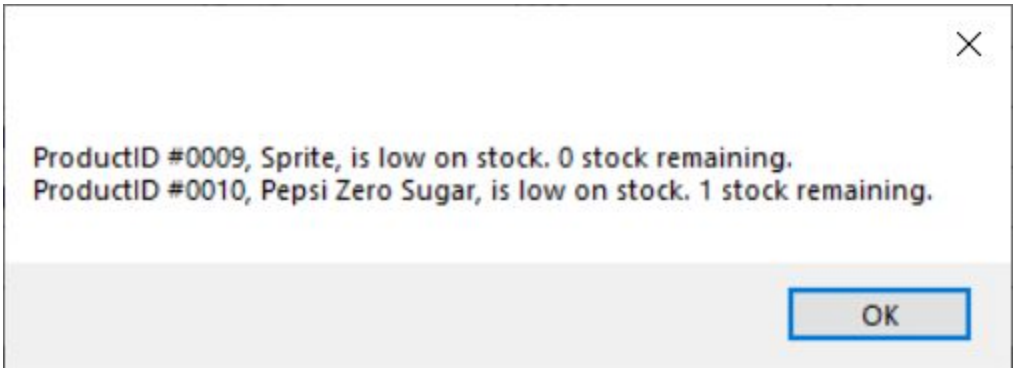


3. System will display the Order Form

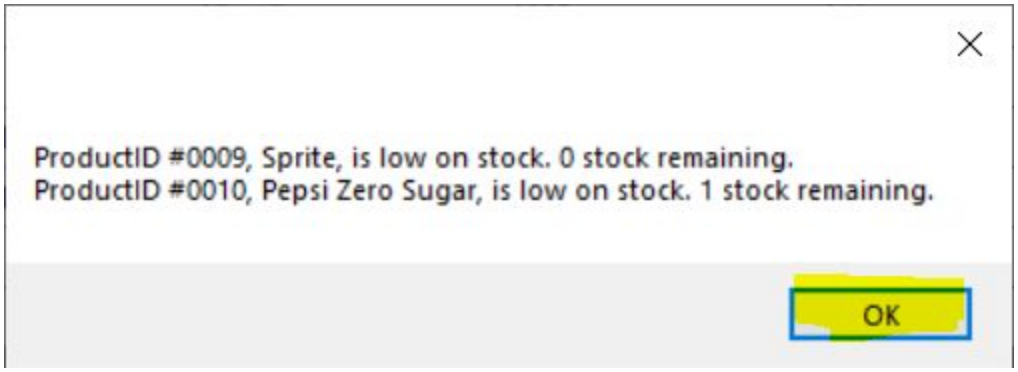


User Story 15: Alerts

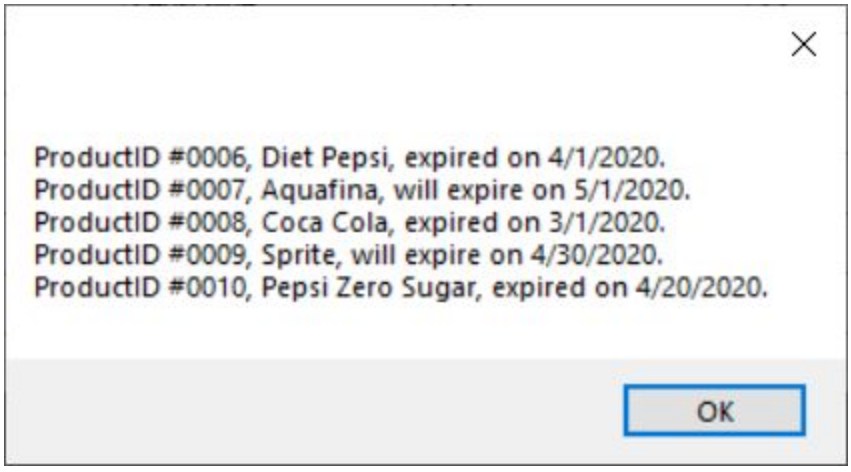
1. Users must login with a clerk or administrator account.
2. After system displays main menu, system will display message of items in low inventory



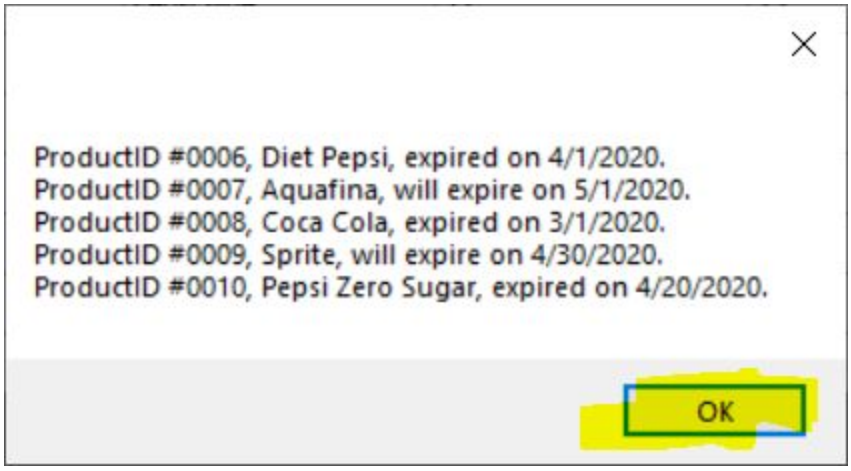
3. Click on “OK” to confirm message



4. System will then display message of items that are soon to expire or expired



5. Click on “OK” to confirm message



References

https://www.indeed.com/ - Used to look up various warehouse/distributor roles to get a better idea of who would be using what parts of the software.