* **Faculty: Hồ Hoàn Kiếm**
* **Team Member:**
  + - * **Nguyễn Hoàng Nhật Tân**
      * **Nguyễn Hoàng Phúc Huy**

## Project Information

* Project name: Toy Management
* Project Code: **TM**
* Product Type: **Windows Forms Application**
* Start Date: **March 13rd, 2020**
* End Date: **April 4th, 2020**

1. Current Situation

Store owner is strenuous on controlling current products and bills. Managing a store isn’t easy, the store owner has to hire at least two staff to run the store. One cashier writes down the bill and calculates the total price, and the staff keeps track of the quantity of products in store.

## Problem Definition

**Below are disadvantages of current situation:**

* **Hard to remember products:** Our customer can not know how total of products that he is owning
* **Waste time to calculate how much totally and write a bill by hand:** When a customer comes to shop, and buys a lot. The staff has to calculate and write down the bill for them, sometimes the handwriting may be ugly. It will make the customer and the owner hard to read.
* **Hard to store all of bills:** Bills are made of paper, it not easy to preserve and store all the bills
* **Mistake from staff:** During work, mistakes more likely happen to the cashier, it can be wrong input for the price of the product.
* If the store is big enough and has a large amount of products, the store owner needs to hire more to keep track of them. The larger your store is the more staff you have to hire. Mistakes may come from staff: Miss out products, report wrong product quantity to the store owner,...

1. Proposed Solution

## Our proposed solution is to build an storing application named “ToyManagement Application”.

TM Application is Windows Form Application which provide functions for:

* The store owner to manage bills, products and staffs.
* The cashier to create bill and see all products that available in store.

### Feature functions

* **Register account:** Create a new account. If this account has already existed, the app will show “This account has already existed”. If not, the app will show “Create new account successfully”
* **Update / Delete / Restore account:** Update for admin or staff account, delete and restore for only staff account
* **View all of bills:** User can see all of current bills or they can search specific bills from day to day with no modify permission
* **View all of products of a specific category, add / update / delete / restore a product of a specific category or only the category:** User will see categories table first. After clicking on a specific category, the product table will show products of that category. And then user can use the Add / Update / Delete function for the product. Additionally, user can use the Add / Update / Delete function for the category also.
* **Create an order and automatic calculate total price:** Easy and fast to create an order, and give the payment for a customer
* **View all products of a specific category:** Staff account can view without allowing to do anything
* **Restore product:** User can restore a product if he wants
* **Restore category:** User can restore a category if he wants

### Advantages and disadvantages of Toy Management Application

* **Advantages:**
* **Easy to control products and bills.**
* **No Internet required**
* **Faster in calculate total price and create bill.**
* **Just need one staff to run the store**
* **Disadvantages:**
* **Not support barcode.**

Customer Request   
 Specification

1. Functional Requirements

Following is a list of functionalities of the application:

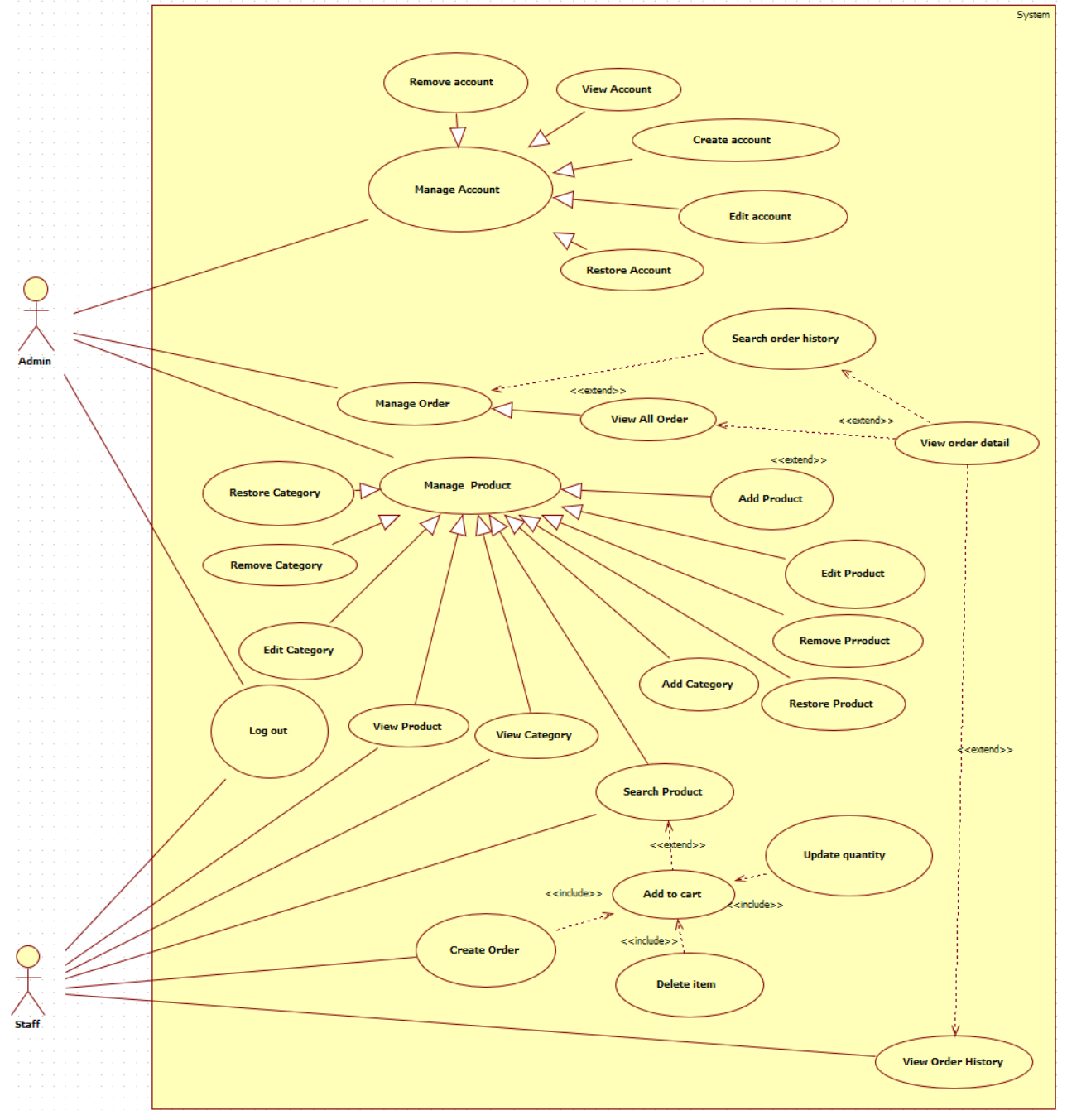
**Admin component:**

* Admin will have the specified account  to login to to the application
* After logging successful, the table will show:
* Product Management
* Order Management
* Account Management
* When the admin chooses “Product Management” on the table, the admin will see a category table first, when the admin clicks on Category, all of the products of that category will be shown. And then the admin can Add / Update / Delete / Restore a product of that category. In addition, the admin can Add a new category or Update / Delete / Restore a specific category or Search product by Product ID.
* When the admin chooses “Order Management” on the table, the admin will see all of the history orders which were created by the staff, and can search bills from this day to another day.
* When the admin chooses “Account Management” on the table, the admin can create a staff account which will be used for logging on Staff’s Table. Additionally, the admin can see a table which is shown with an account list. The admin can Update / Delete / Restore to the staff account, ONLY Update to ADMIN account.

**Staff component:**

* Staff has to know the account and password from being created by the Admin
* After logging successfully, the table will show “Cashier Form” which is used for creating a new order. When the staff types a keyword on “Product ID” Textbox, the textbox will show some suggestions for being easy to choose, then the staff makes a choice and type a quantity, then the quantity of that product which are being stored on the system will show. When clicking on the “Add to cart” button, the product will be shown on the table with some values “ProductID, Name, Price, Quantity, Sub total”. At the same time, Total will be shown the total price of all products which are added to cart. If the staff adds the wrong product or the customers change their mind, she can click on the “Remove Item” to remove the product in the cart. When finishing a cart, the staff will click on the “Create order” button, and all of the products from the cart will be made a bill and added to Order Management. If the staff clicks on the “View Bill” button, the staff will see specific orders which were created by that staff, and also the staff can see its details by clicking on the “View Detail”. In addition, the staff can reset the cart by clicking on the“New Cart” button. Furthermore, Staff can also view products which are available and being stored by clicking on the “View Product” button, but Staff can ONLY view products.

1. Use Case Diagram



**Use Case Specifications**

|  |  |
| --- | --- |
| **Use case** | **Login** |
| **Short Description** | **Login to app to use functions of Staff or Admin** |
| **Pre-Condition** |  |
| **Post-Condition** |  |
| **Main flow** | **\_Input:**  **.Username**  **.Password**  **\_Clicks on the Login button.**  **\_System check Username and Password**  **\_If Username and Password are valid and the role is admin, the system will move a user to AdminForm and be shown “Login successfully”**  **\_If Username and Password are valid and the role is staff, the system  will move the user to CashierForm and be shown “Login successfully”**  **\_If Username or Password is wrong a message will show up and tell the user “Invalid username or password”** |
| **Alternative flow(s)** | **\_If guess left username or password field blank, a message will be shown “Username or password is blank”** |

|  |  |
| --- | --- |
| **Use case** | **Create an account** |
| **Short Description** | **Create new account for only Staff** |
| **Pre-Condition** | **\_Login successfully as admin role**  **\_Click on the “New” button and choose “ADD”**  **\_Full name is only chars** |
| **Post-Condition** | **\_The message will show “Added successfully”** |
| **Main flow** | **\_User has to input:**   * **Username** * **Password** * **Staff name**   **\_The system will get Username, Password, Full name and shows on the table, and the box of status on the table is checked. At the same time inserting an new account and set Status which is true for this account to database** |
| **Alternative flow(s)** | **\_If username or password or full name is blank, a message will show “Username or Password or Fullname is blank”**  **\_If username has been existed, a message will show “This account is existed”**  **\_If full name is not chars, a message will show “Username is about chars and number, and Password is about chars and number, and Full name is ONLY chars”** |

|  |  |
| --- | --- |
| **Use case** | **Update an account** |
| **Short Description** | **Update a specific account** |
| **Pre-Condition** | **\_Login successfully as admin role**  **\_Click on the “New” button and choose “UPDATE”**  **\_Only update password and full name**  **\_The status of this account is true**  **\_Full name is only chars** |
| **Post-Condition** | **\_The message will show “Updated successfully”** |
| **Main flow** | **\_User has to input:**   * **Password** * **Full name**   **\_The system will get Password, Full name of that account on the table and then updating that account to database as on the table** |
| **Alternative flow(s)** | **\_If the account has Status = false, a message will show “You must restore this account before updating”**  **\_\_If full name is not chars, a message will show “Password is about chars and number, and Full name is ONLY chars”**  **\_If an error has occurred during processing that a message will show “Updated fail”** |

|  |  |
| --- | --- |
| **Use case** | **Delete an account** |
| **Short Description** | **Delete a Staff account** |
| **Pre-Condition** | **\_Login successfully as admin role**  **\_Status of this account is true** |
| **Post-Condition** | **\_A message will show “Are you sure to do this?” if a user chooses “Yes” on confirming message then the message will show “Deleted successfully”** |
| **Main flow** | **\_User will click on a row which he wanna remove**  **\_The system will set Status is false on the database and the box of status on the table is unchecked** |
| **Alternative flow(s)** | **\_If a user has deleted an admin account, a message will show “Warning: You can not delete an admin account!!!”**  **\_If the status of this account is false, a message will show “This account has already deleted”**  **\_If an error has occurred during processing that a message will show “Deleted fail”** |

|  |  |
| --- | --- |
| **Use case** | **Restore an account** |
| **Short Description** | **Restore a Staff account** |
| **Pre-Condition** | **\_Login successfully as admin role**  **\_The status of this account is false** |
| **Post-Condition** | **\_The message will show “Restored successfully”** |
| **Main flow** | **\_User will click on a row which he wanna restore**  **\_The system will set Status is true on the database and the box of status on the table is checked** |
| **Alternative flow(s)** | **\_If the status of this account is true, a message will show “This account has not deleted yet”**  **\_If an error has occurred during processing that a message will show “Restored fail”** |

|  |  |
| --- | --- |
| **Use case** | **Create Order** |
| **Short Description** | **Cashier creates an order** |
| **Pre-Condition** | **\_Login successfully as staff role**  **\_Cart must have one or more product** |
| **Post-Condition** |  |
| **Main flow** | **\_Clicks on the ‘Create Order’ button.**  **\_The system will get staffID, current date-time, products in cart and insert a new Order to database** |
| **Alternative flow(s)** | **\_If an error has occurred during processing that a message will show “Failed when creating an order”** |

|  |  |
| --- | --- |
| **Use case** | **Search Product** |
| **Short Description** | **Search a product which is based on a product’s id** |
| **Pre-Condition** | **\_User must login** |
| **Post-Condition** |  |
| **Main flow** | **\_Insert product ID to search bar and click on the Search button**  **\_The system shows the search results of the product and its information**  **\_If product is not found, the system will show a message “This Product is not existed”** |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Use case** | **View Order History** |
| **Short Description** | **\_Allow user to view the list of orders** |
| **Pre-Condition** | **\_User must login** |
| **Post-Condition** | **\_List of the route is viewed successfully** |
| **Main flow** | **\_The system shows a list of order, each order includes:**  **. OrderID**  **. AccountID**  **. Fullname**  **. OrderDate**  **. Total** |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Use case** | **View Order Detail** |
| **Short Description** | **\_View a detail of the order such as date create, cashier, products** |
| **Pre-Condition** | **\_User must login** |
| **Post-Condition** | **\_Order detail is showed** |
| **Main flow** | **\_The system show details of an order, include:**  **. OrderID**  **. Cashier**  **. Date**  **. List of Products: ProductID, ProductName, Quantity, Price** |
| **Alternative flow(s)** | dsadasd  sa |

|  |  |
| --- | --- |
| **Use case** | **Search Order** |
| **Short Description** | **\_Search specific orders which are based in the rage of dates** |
| **Pre-Condition** | **\_Use must login as Admin role** |
| **Post-Condition** | **\_Searching route results are shown.** |
| **Main flow** | **\_System searches which order was created in the certain date range**  **\_A list of result will be shown** |
| **Alternative flow(s)** | **\_ “Date from” is greater than the current date, the system will show a message “Invalid date input”**  **\_ “Date from” is greater than “Date to”, the system will show a message “Invalid date input”** |

|  |  |
| --- | --- |
| **Use case** | **Add Product** |
| **Short Description** | **\_Insert new product into the system** |
| **Pre-Condition** | **\_User must login as Admin role**  **\_User clicks on “New” button on the product side**  **\_Quantity and Price input must be an integer number**  **\_Price must be bigger than 0**  **\_The category status must be active** |
| **Post-Condition** | **\_Message will show “New Product is added”** |
| **Main flow** | **\_User has to input:**  **. CategoryID**  **. CategoryName**  **. ProductID**  **. ProductName**  **. Quantity**  **. Price**  **\_User clicks button “Add”**  **\_New product now is inserted to database** |
| **Alternative flow(s)** | **\_If ProductID has already existed, a message will show “This productID has already existed”**  **\_If an error has occurred during processing a message will show “Error when adding product !”** |

|  |  |
| --- | --- |
| **Use case** | **View Product** |
| **Short Description** | **\_The system will show all products which are based on the category user selected** |
| **Pre-Condition** | **\_User must login** |
| **Post-Condition** | **\_A list of products are shown** |
| **Main flow** | **\_System will show a list of product with the following attributes:**  **. ID**  **. Name**  **. Quantity**  **. Price** |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Use case** | **Edit Product** |
| **Short Description** | **\_Edit an available product** |
| **Pre-Condition** | **\_Use must login as Admin role**  **\_Product must be available**  **\_Quantity and Price input must be an integer number**  **\_Price must be bigger than 0**  **\_The category status and product status must be active** |
| **Post-Condition** | **\_A message will show “Product has been edited! ”** |
| **Main flow** | **\_User can edit the following information:**  **. Name**  **. Price**  **. Quantity** |
| **Alternative flow(s)** | **\_If error has occurred during processing a message will show “Error when editing product !”** |

|  |  |
| --- | --- |
| **Use case** | **Delete Product** |
| **Short Description** | **\_System will all products based on the category user selected** |
| **Pre-Condition** | **\_User must login as Admin role** |
| **Post-Condition** | **\_A message will show “Product Deleted” if a user chooses “Yes” on confirming message** |
| **Main flow** | **\_A message will show to confirm if user want to delete product or not**  **\_If user chooses “Yes”, the product will be deleted from the system** |
| **Alternative flow(s)** | **\_If error occurred during processing a message will show “Delete fail.”**  **\_User chooses “No”, the system closes the message** |

|  |  |
| --- | --- |
| **Use case** | **Restore Product** |
| **Short Description** | **\_Restore a product that has already been deleted from the system** |
| **Pre-Condition** | **\_User must login as Admin role**  **\_The product must be deleted before** |
| **Post-Condition** | **\_A message will show “Product restored”** |
| **Main flow** | **\_The system get the ProductID of a selected product that has been deleted and change the status of that from deleted to active** |
| **Alternative flow(s)** | **\_If error has occurred during processing that a message will show “Failed to restore Category”** |

|  |  |
| --- | --- |
| **Use case** | **View Category** |
| **Short Description** | **\_System will show all product’s category in the system** |
| **Pre-Condition** | **\_User must login** |
| **Post-Condition** | **\_A list of Category is shown** |
| **Main flow** | **\_System will show a list of Category with the following attribute:**  **. Name** |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Use case** | **Add Category** |
| **Short Description** | **\_Add a new category** |
| **Pre-Condition** | **\_User must login as Admin role**  **\_User clicked button “Action” in Category side**  **\_All category’s text fields must be filled** |
| **Post-Condition** | **\_A message will show “New Category added!”** |
| **Main flow** | **\_User have to input:**  **. CategoryID**  **. CategoryName**  **\_ When user click “Add” button on the category side, the system will insert new category to the database** |
| **Alternative flow(s)** | **\_If categoryID is duplicated, a message will show “This category ID is already existed”**  **\_If error occurred during processing a message will show “Error when adding Category”** |

|  |  |
| --- | --- |
| **Use case** | **Edit Category** |
| **Short Description** | **\_Edit information for a category** |
| **Pre-Condition** | **\_User must login as Admin role**  **\_All category’s text fields must be filled** |
| **Post-Condition** | **\_A message will show “Saved”** |
| **Main flow** | **\_User can edit the following information for category:**  **. Name**  **\_When the user click on the “Edit” button, the information of the category will be saved** |
| **Alternative flow(s)** | **\_If error occurred during processing a message will show “Error when editing Category”** |

|  |  |
| --- | --- |
| **Use case** | **Delete Category** |
| **Short Description** | **\_Delete an available category** |
| **Pre-Condition** | **\_User must login as Admin role** |
| **Post-Condition** | **\_A message will show “Category deleted” when user choose “Yes” on confirming message** |
| **Main flow** | **\_System will show a message to confirm if the user wants to delete the category or not**  **\_If user choose “Yes”, the category will be deleted along with its products** |
| **Alternative flow(s)** | **\_If user chose “No”, the system closes the message**  **\_If error occurred during processing a message will show “Delete fail.”** |

|  |  |
| --- | --- |
| **Use case** | **Restore Category** |
| **Short Description** | **\_Restore a category that has already been deleted from the system** |
| **Pre-Condition** | **\_User must login as Admin role**  **\_The category must be deleted before**  **\_User must provide the categoryID** |
| **Post-Condition** | **\_A message will show “Category restored”** |
| **Main flow** | **\_System get the CategoryID that has been deleted and change only the status of that category from deleted to active, the products in that category will not change** |
| **Alternative flow(s)** | **\_If error occurred during processing a message will show “Failed to restore Category”** |

|  |  |
| --- | --- |
| **Use case** | **Add to Cart** |
| **Short Description** | **Insert an item to cart** |
| **Pre-Condition** | **\_User must login as Staff role**  **\_Quantity must be number**  **\_Product status must be active** |
| **Post-Condition** | **\_The item you added appear on the grid view cart** |
| **Main flow** | **\_System get the ProductID and the quantity customer want to buy, compare with the product’s available at the store and insert it to cart** |
| **Alternative flow(s)** | **\_If the quantity customer want to buy is greater than the product’s available, a message will show “Product quantity must be greater than 0 and lower than product available”** |

|  |  |
| --- | --- |
| **Use case** | **Update Cart** |
| **Short Description** | **\_Edit quantity of an item in cart** |
| **Pre-Condition** | **\_User must login as Staff role**  **\_The item must be in the cart** |
| **Post-Condition** | **\_A message will show “Product has been updated”** |
| **Main flow** | **\_System get the ProductID and the quantity user want to change, compare with the product’s available at the store and update that product’s quantity in cart** |
| **Alternative flow(s)** | **\_If error occurred during processing a message will show “Failed to restore Category”** |

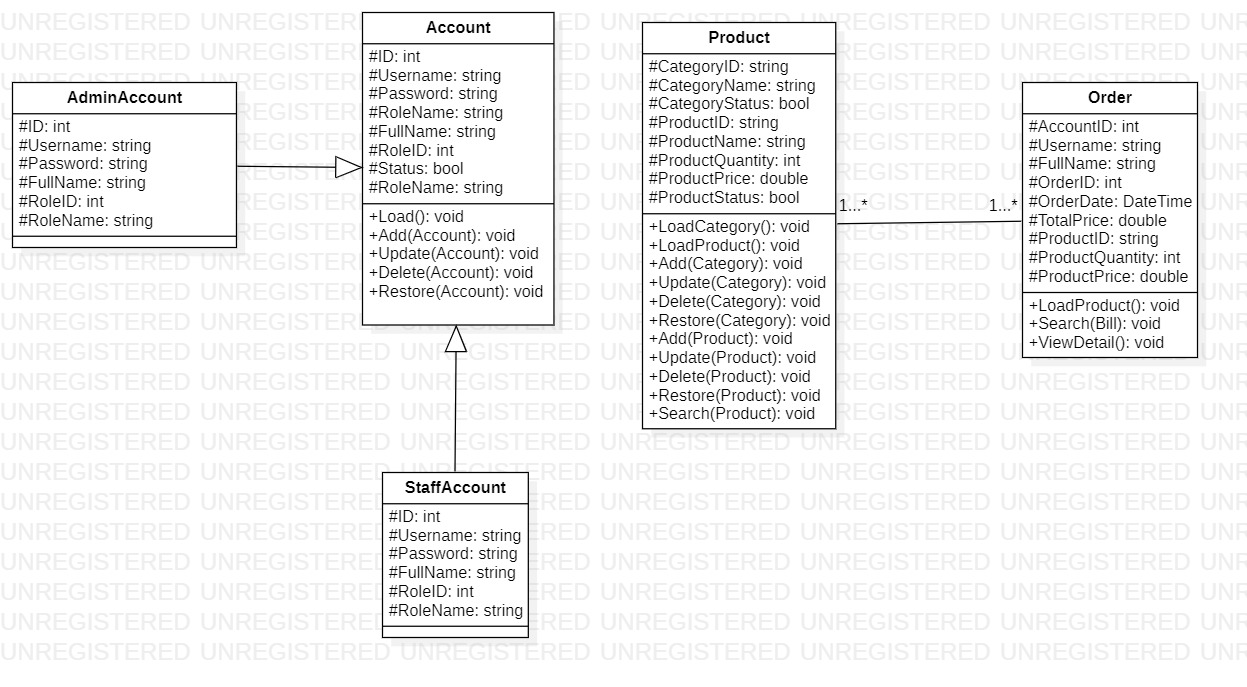
|  |  |
| --- | --- |
| **Use case** | **Delete Item** |
| **Short Description** | **Remove an item in cart** |
| **Pre-Condition** | **\_User must login as Staff role**  **\_The item must be in cart** |
| **Post-Condition** | **\_The item in the cart is removed** |
| **Main flow** | **\_System get the ProductID when a user clicked on the item in the cart and then remove that item from cart** |
| **Alternative flow(s)** | **\_If error occurred during processing a message will show “Fail to remove item”** |

|  |  |
| --- | --- |
| **Use case** | **Manage Product** |
| **Short Description** | **Manage accounts in the system** |
| **Pre-Condition** | **\_User must login as admin role** |
| **Post-Condition** |  |
| **Main flow** | **\_Administrator can view, insert, update, delete, search, and restore products.**  **\_Administrator can view, insert, update, delete and restore categories.** |
| **Alternative flow(s)** |  |

|  |  |
| --- | --- |
| **Use case** | **Manage Order** |
| **Short Description** | **\_Manage order in the system** |
| **Pre-Condition** | **\_User must login as admin role** |
| **Post-Condition** |  |
| **Main flow** | **\_Administrator can view, search, view detail orders.** |
| **Alternative flow(s)** |  |

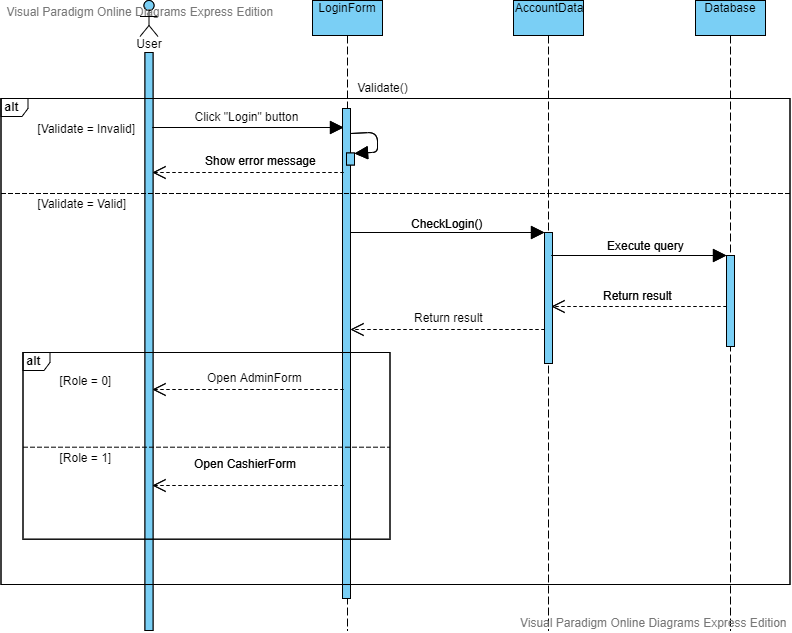
|  |  |
| --- | --- |
| **Use case** | **Manage Account** |
| **Short Description** | **\_Manage account in the system** |
| **Pre-Condition** | **\_User must login as admin role** |
| **Post-Condition** |  |
| **Main flow** | **\_Administrator can view, insert, update, remove accounts.** |
| **Alternative flow(s)** |  |

1. Class Diagram

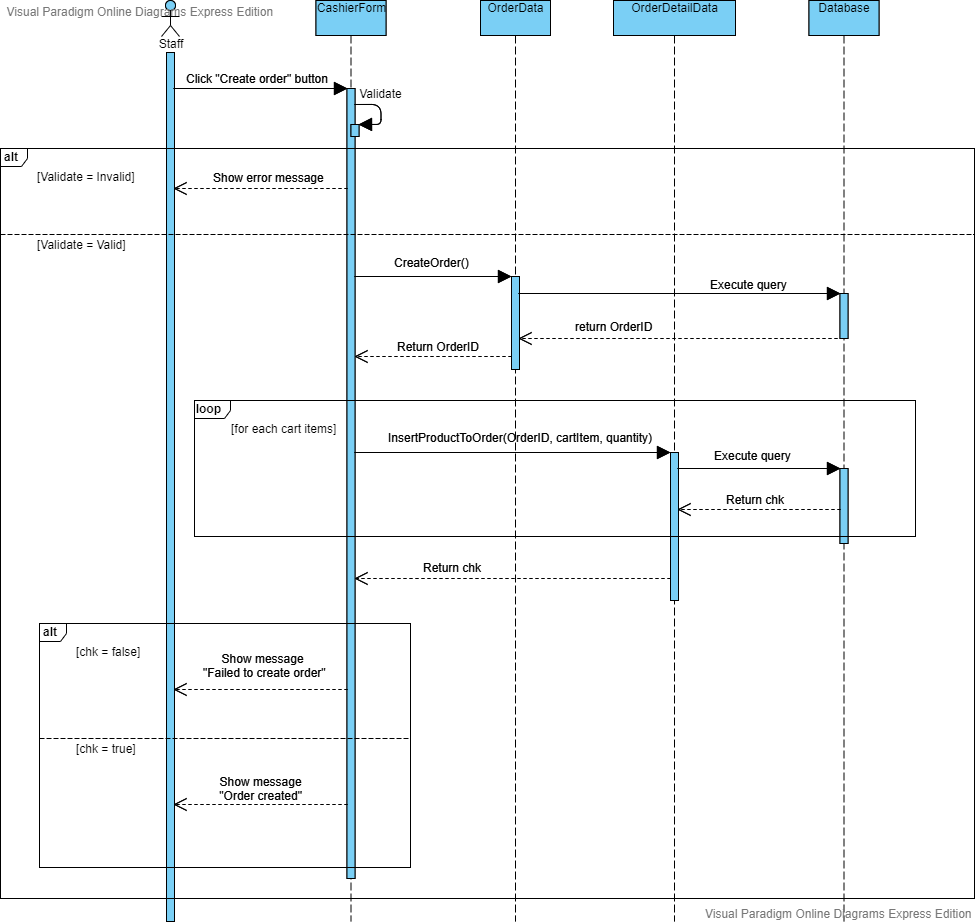


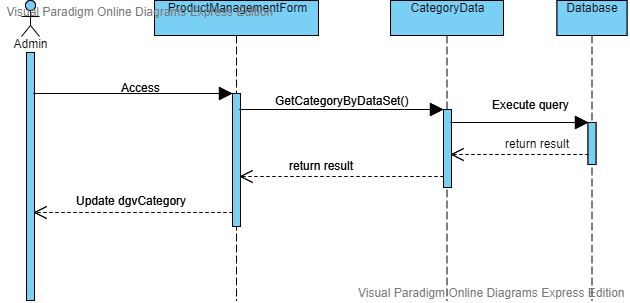
1. Sequence Diagram

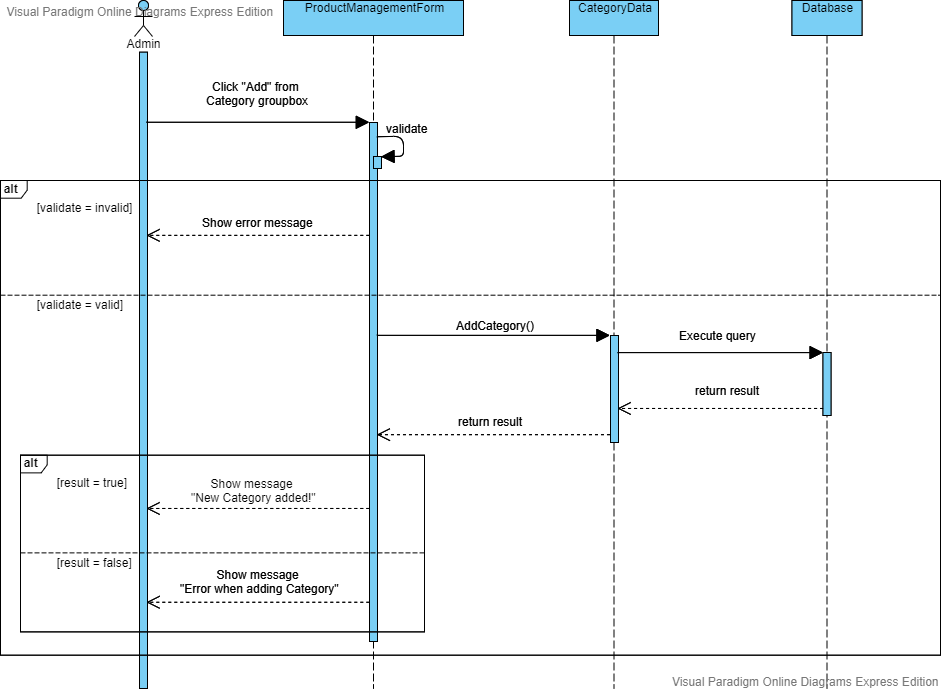
* **Login:**

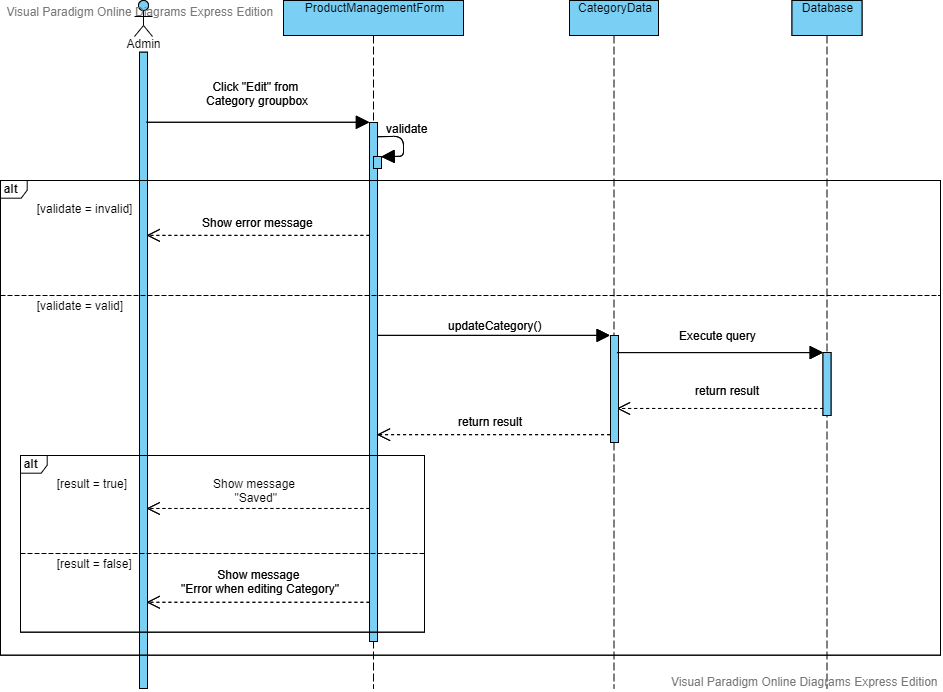


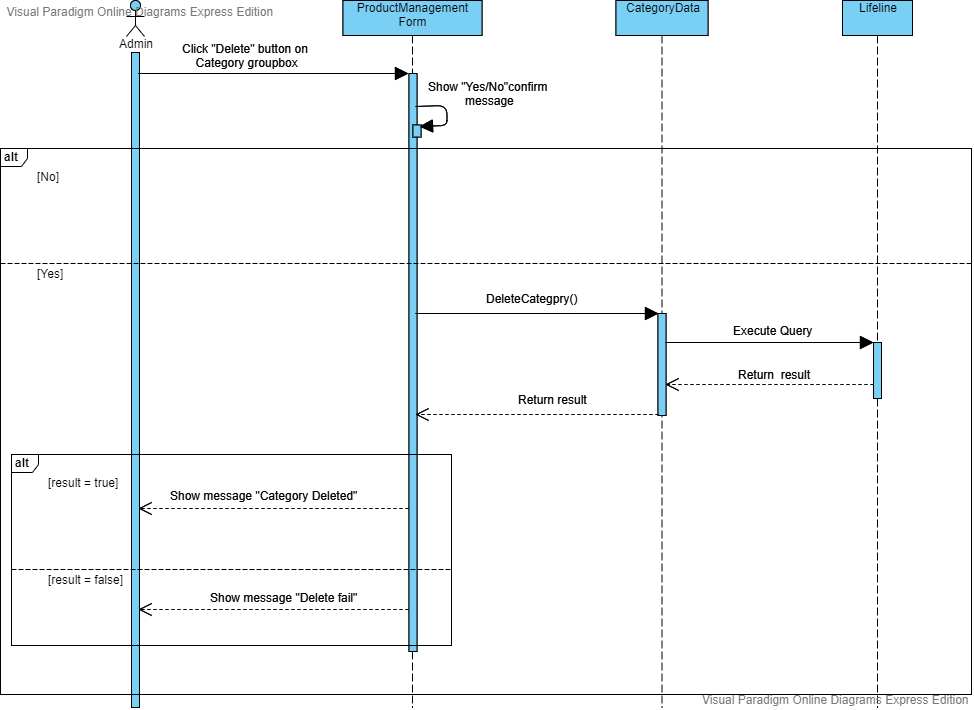
* **Create order:**

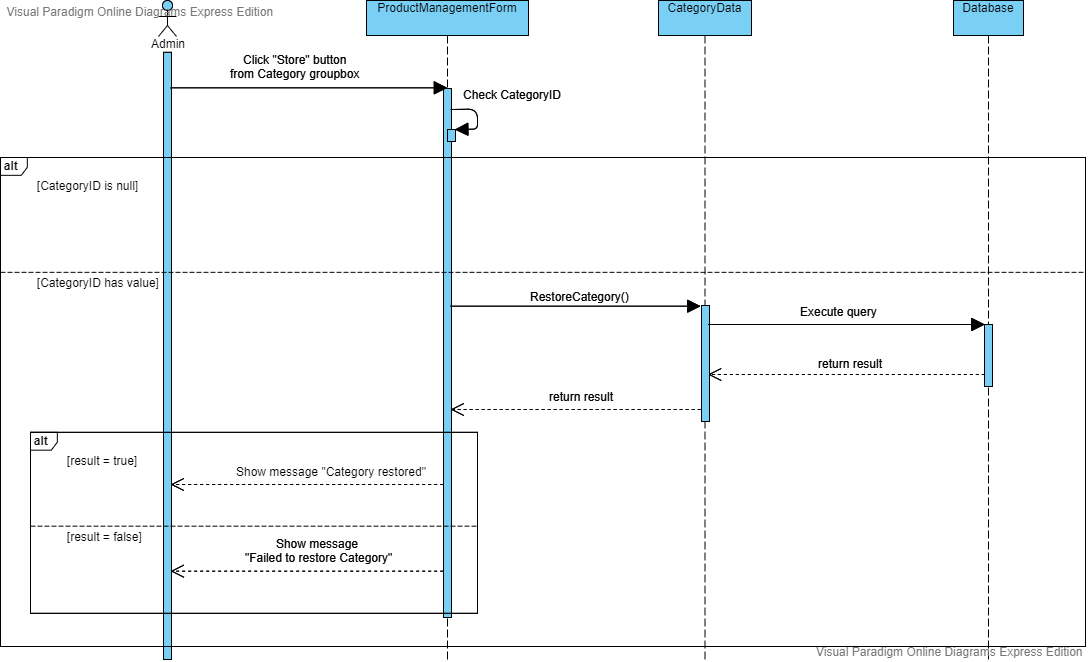


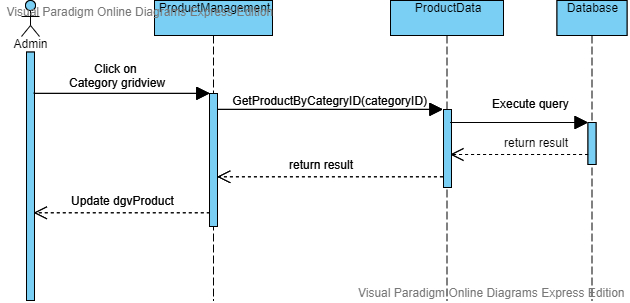
**.View Category:**

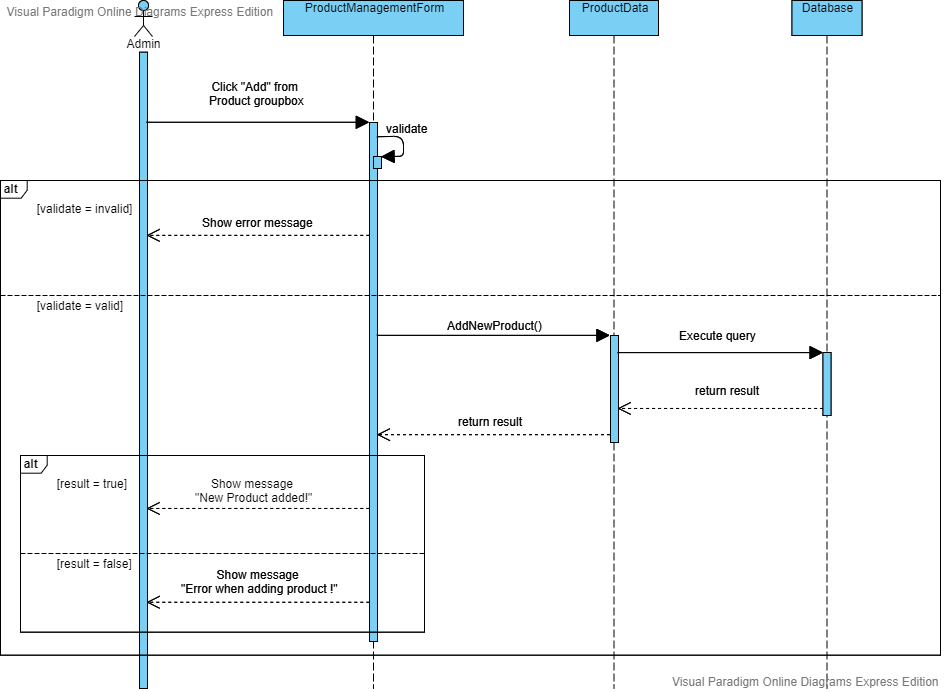
**.Add Category:** 

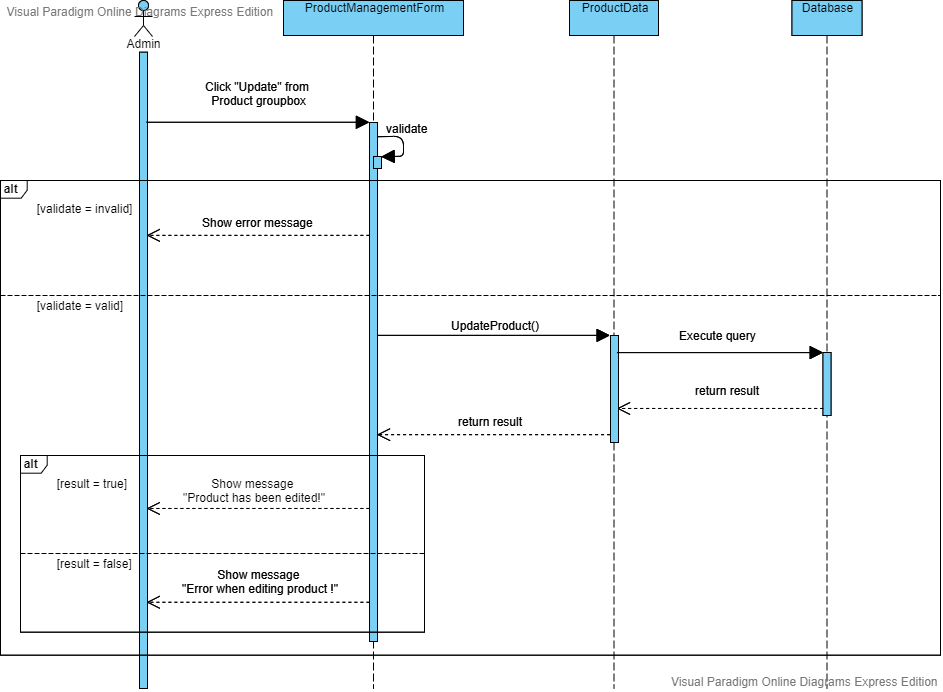
**.Edit category:** 

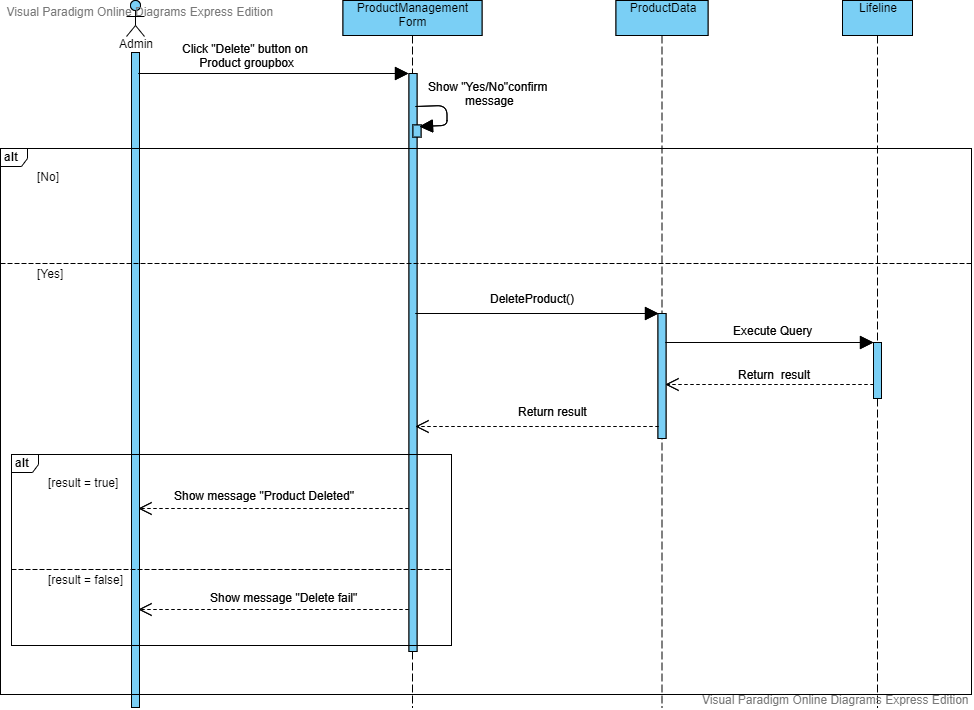
**.Delete Category:** 

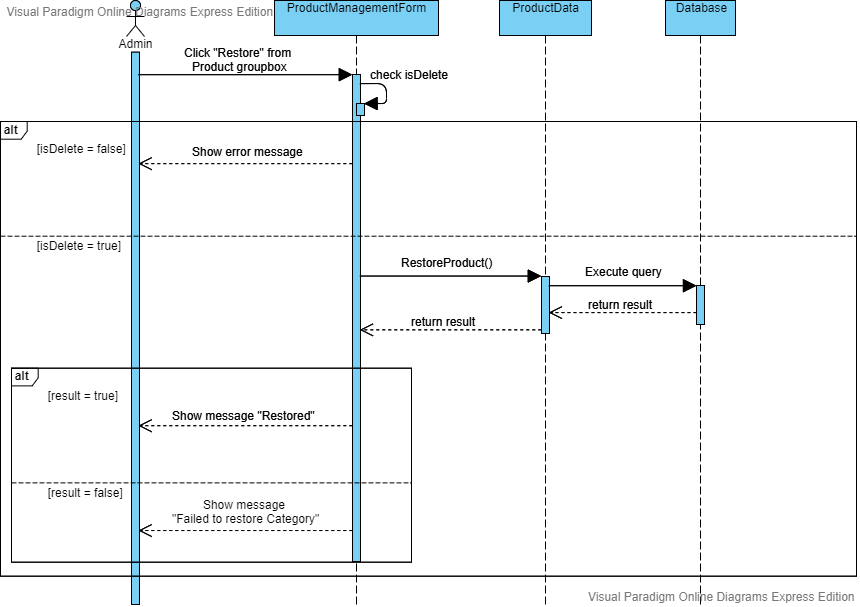
**.Restore Category:** 

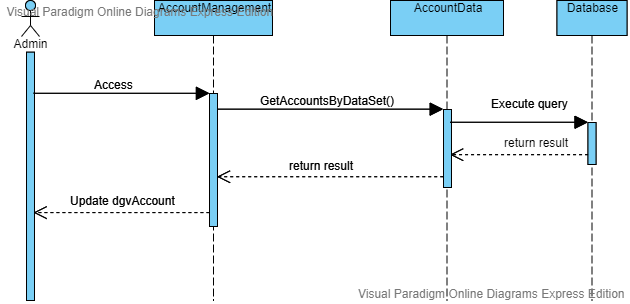
**.View Product:** 

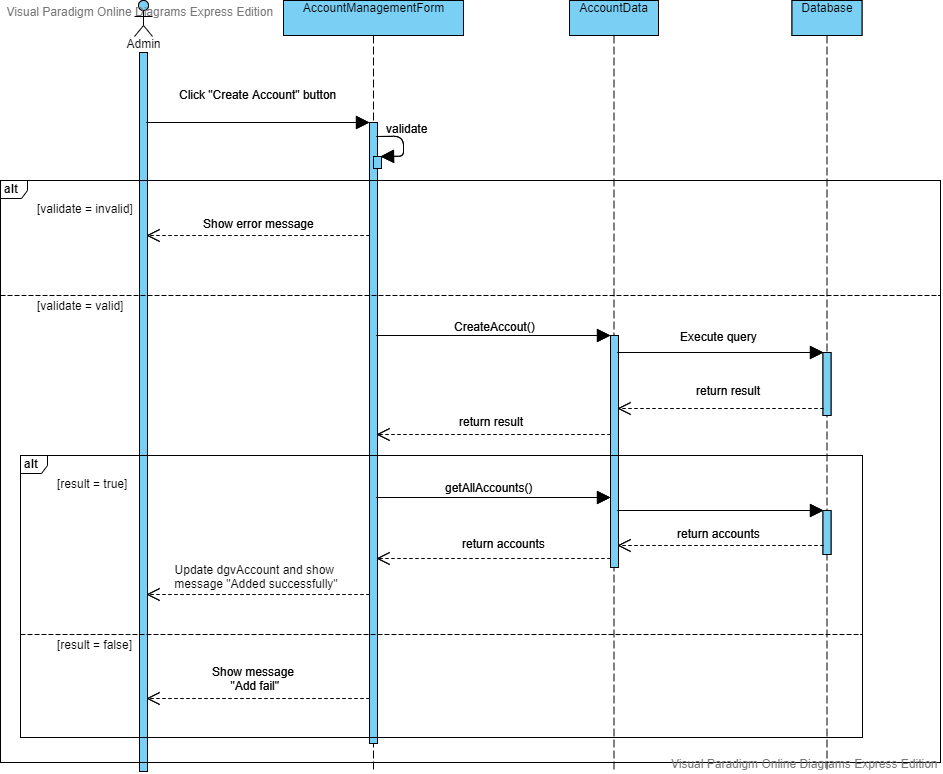
**.Add Product:** 

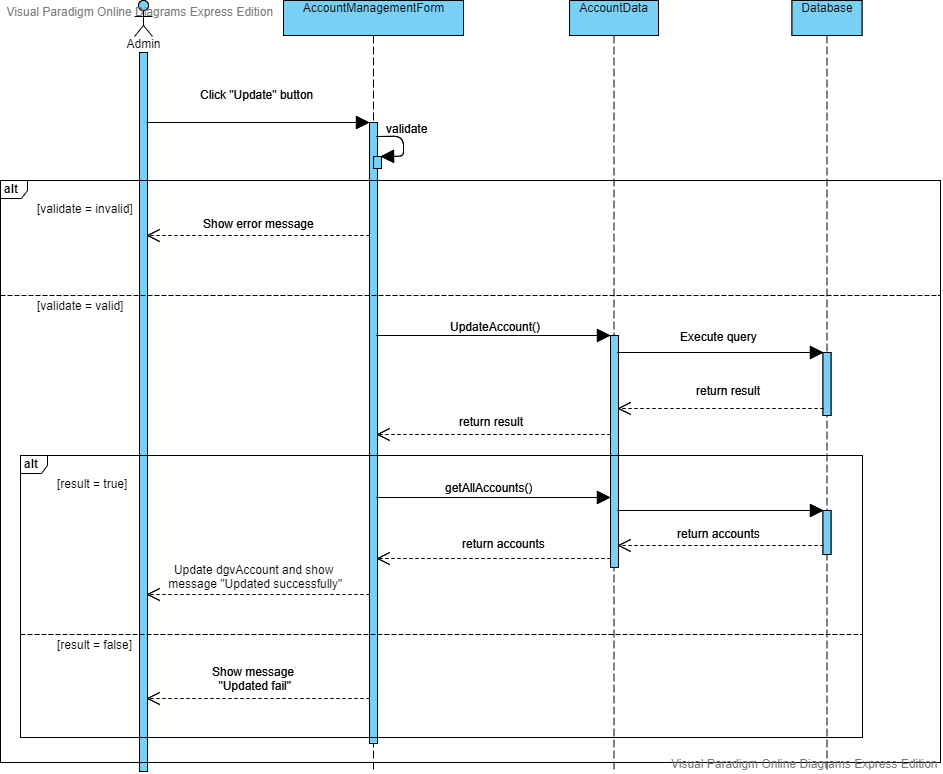
**.Edit Product:** 

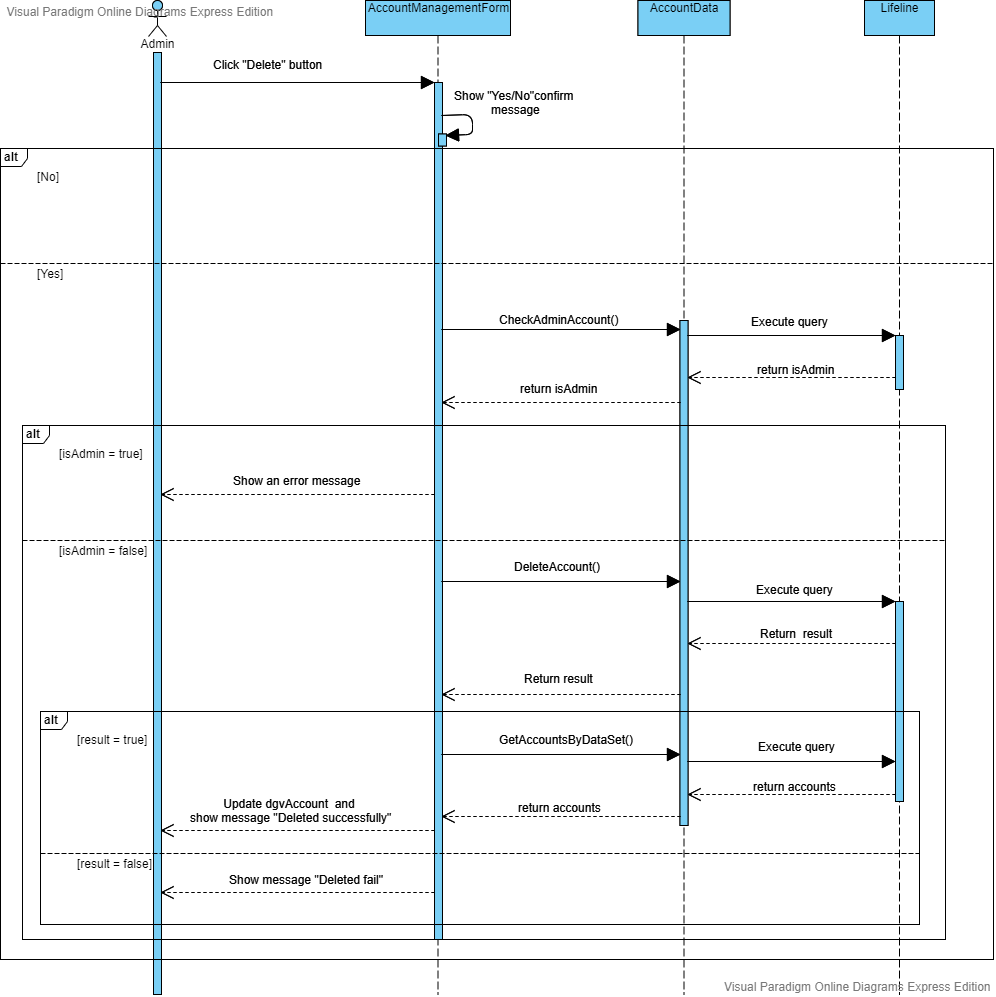
**.Delete Product:** 

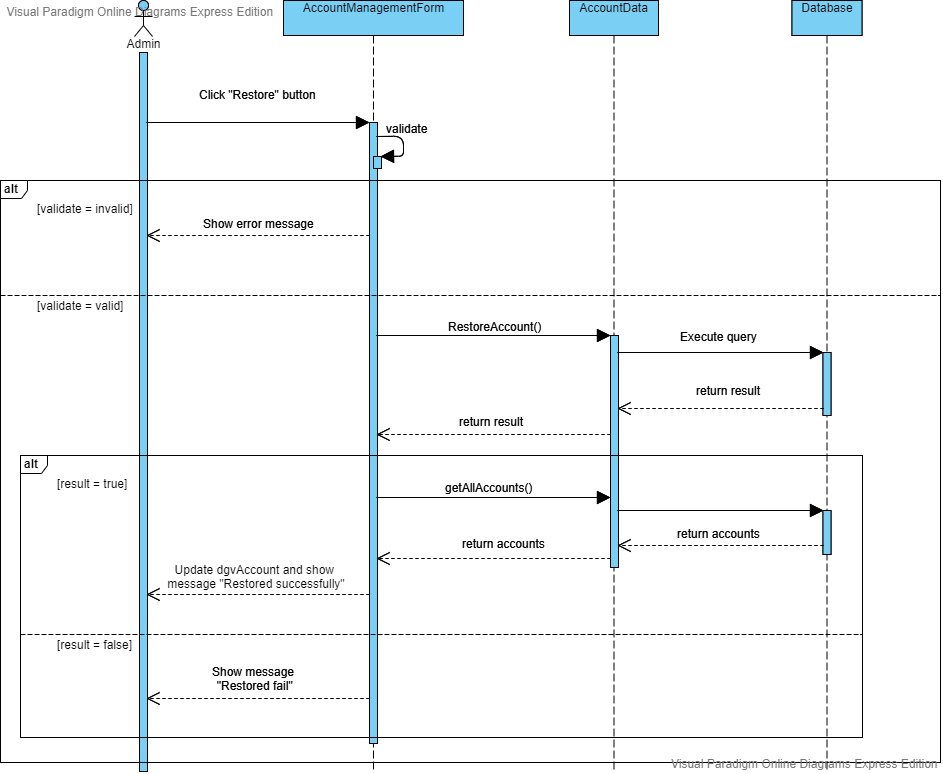
**.Restore product:** 

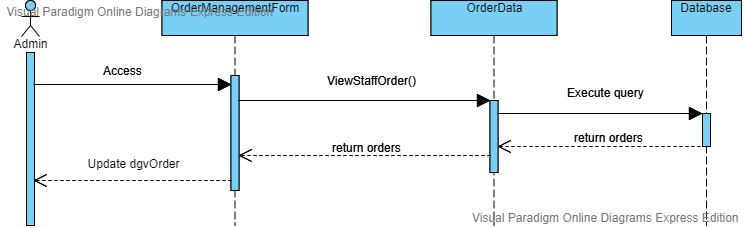
**.View Account:** 

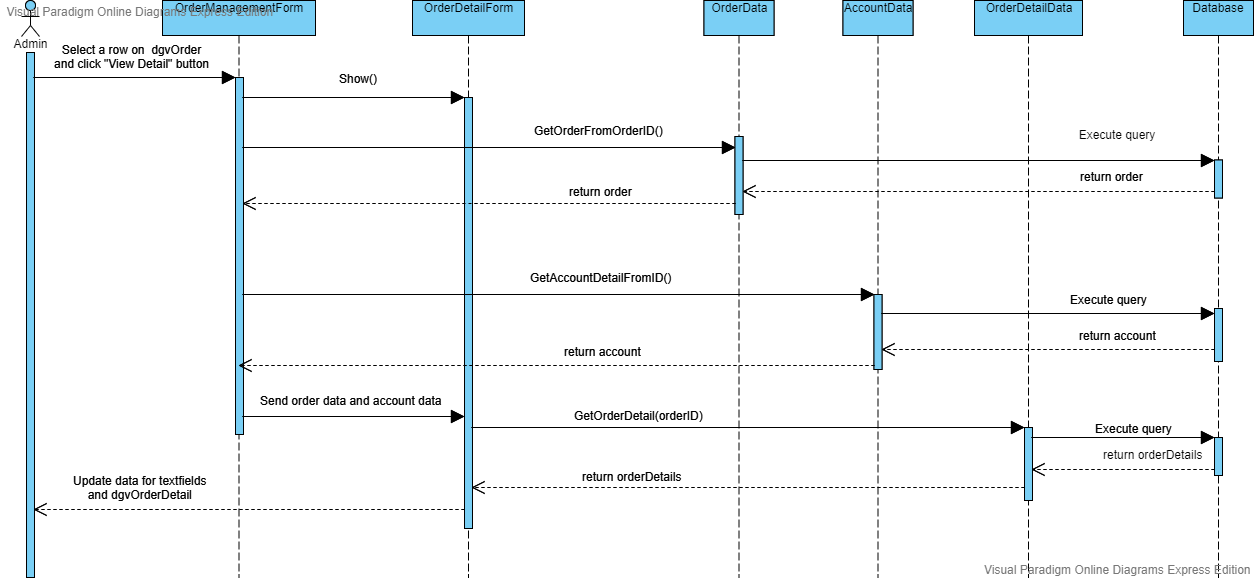
* **Create Account:** 

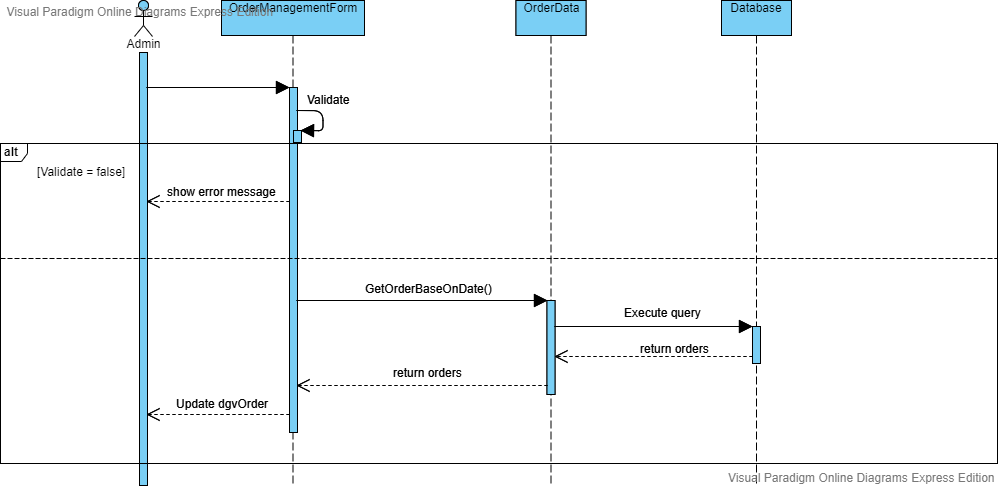
**.Edit Account:**

**.Delete Account:**

**.Restore Account:**

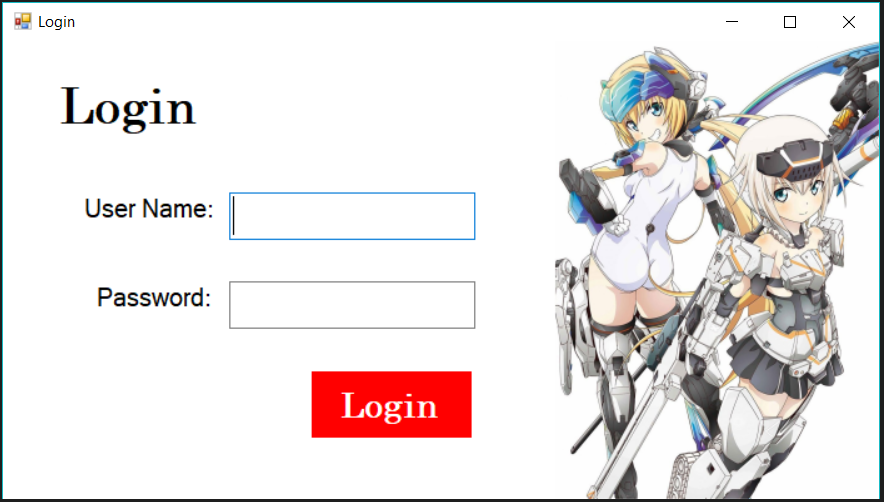
**.ViewOrder:**

**.ViewOrderDetail:**

**.Search Order:**

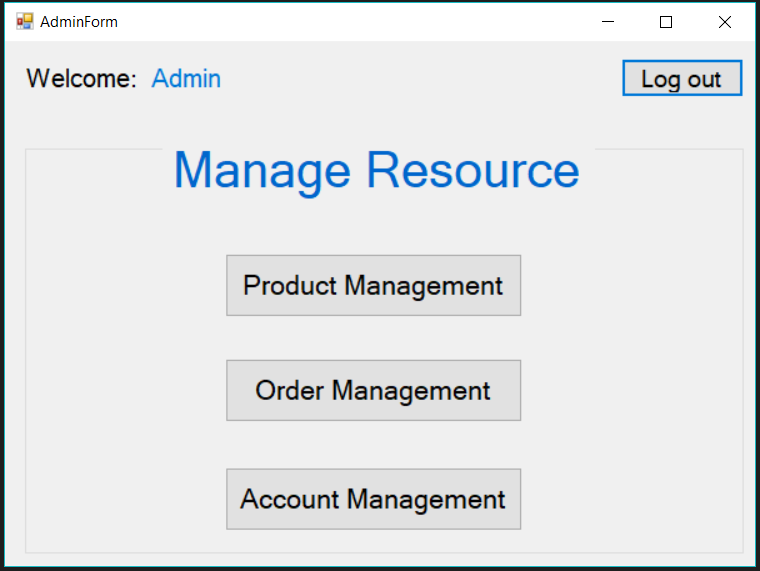
1. Interface Design

* **Login**



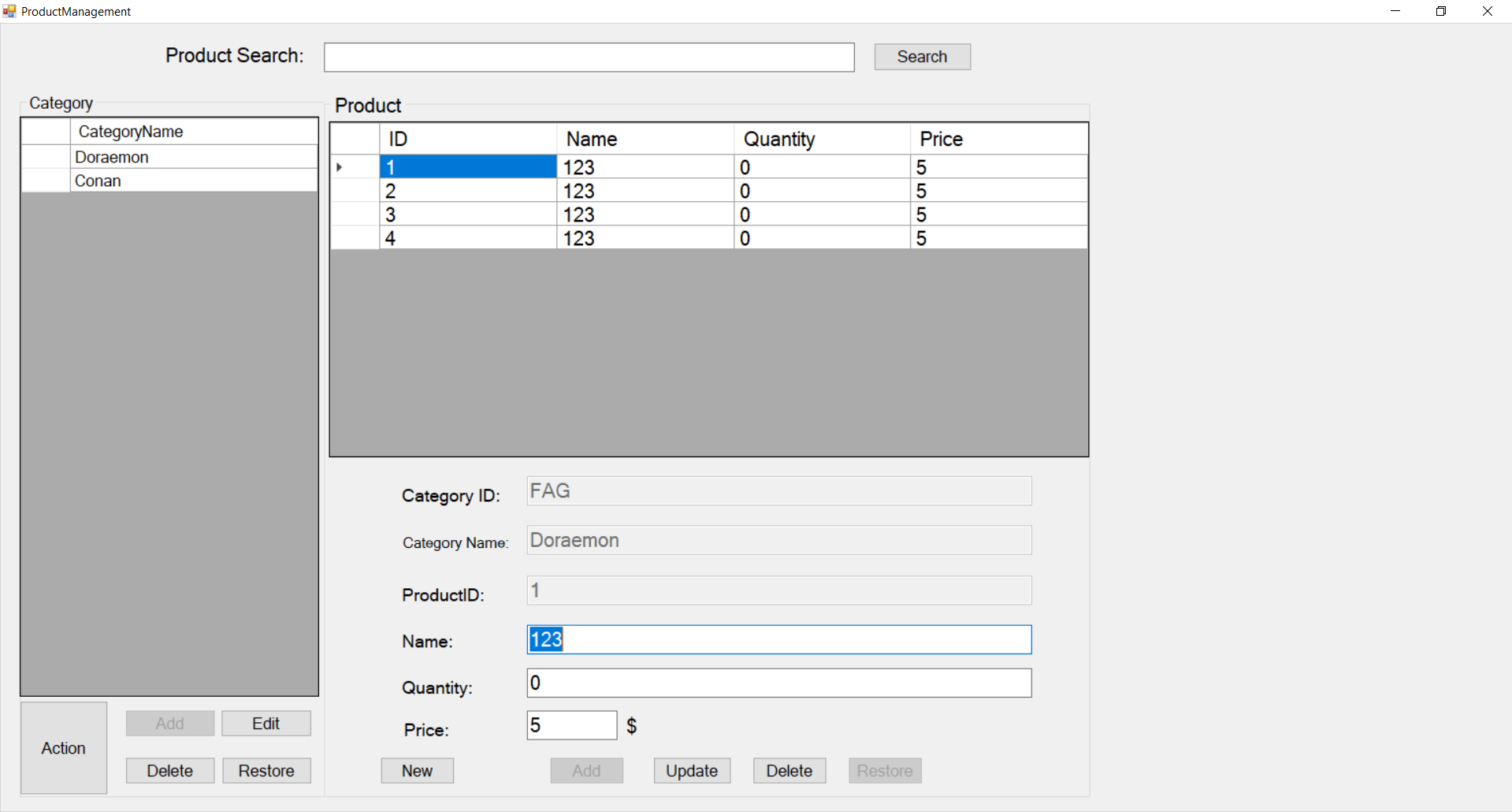
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| User Name | TextBox |  | String, not null | Input Customer’s user name |  |
| Password | TextBox |  | String, not null | Input Customer’s password |  |
| Login Object | Button |  |  | Login |  |

* **Admin role:**
  1. **Admin Form**



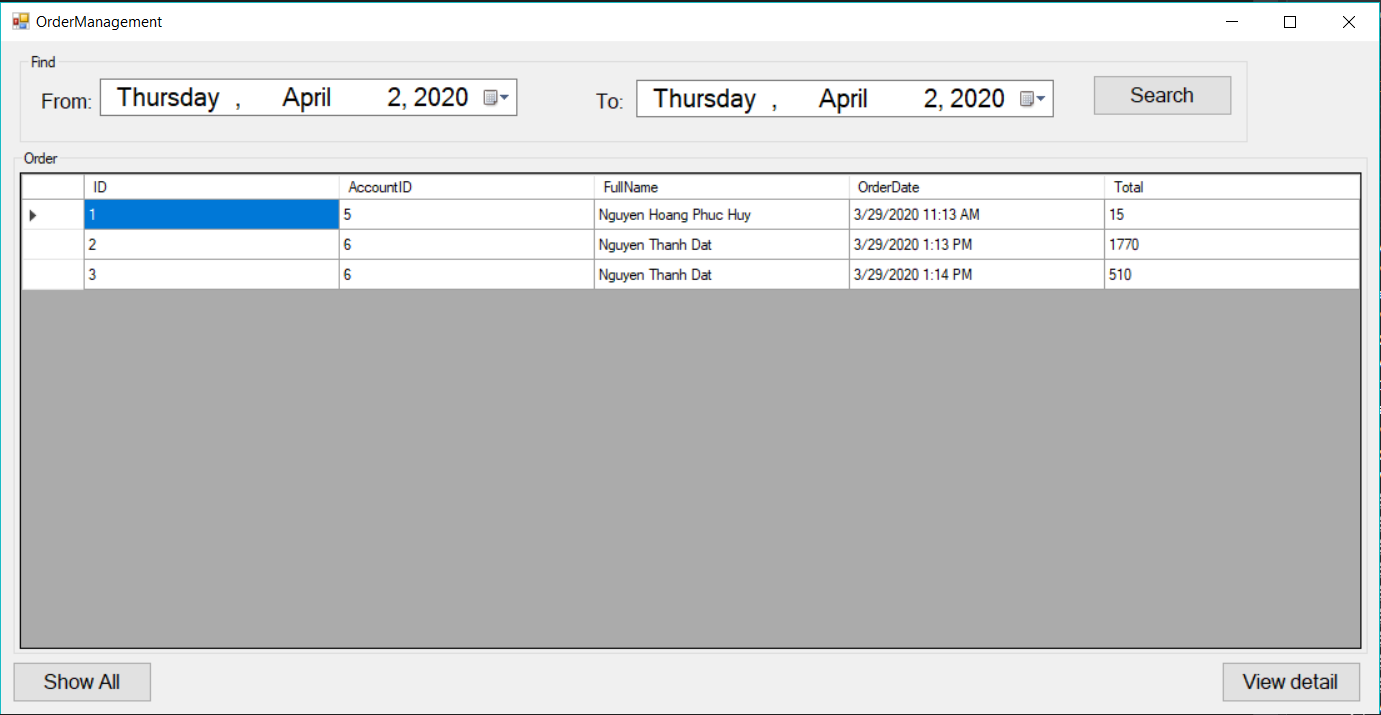
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Welcome | Label |  | Read Only | View a greeting |  |
| Admin | Label |  | Read Only | View full name of the account which logs in |  |
| Manage Resource | Label |  | Read Only | View a title |  |
| Product Management Object | Button |  |  | Open Product Manage Form |  |
| Order Management Object | Button |  |  | Open Order Management Form |  |
| Account Management Object | Button |  |  | Open Account Management Form |  |
| Log out Object | Button |  |  | Exit and Open Login Form |  |

* 1. **Product Management Form**



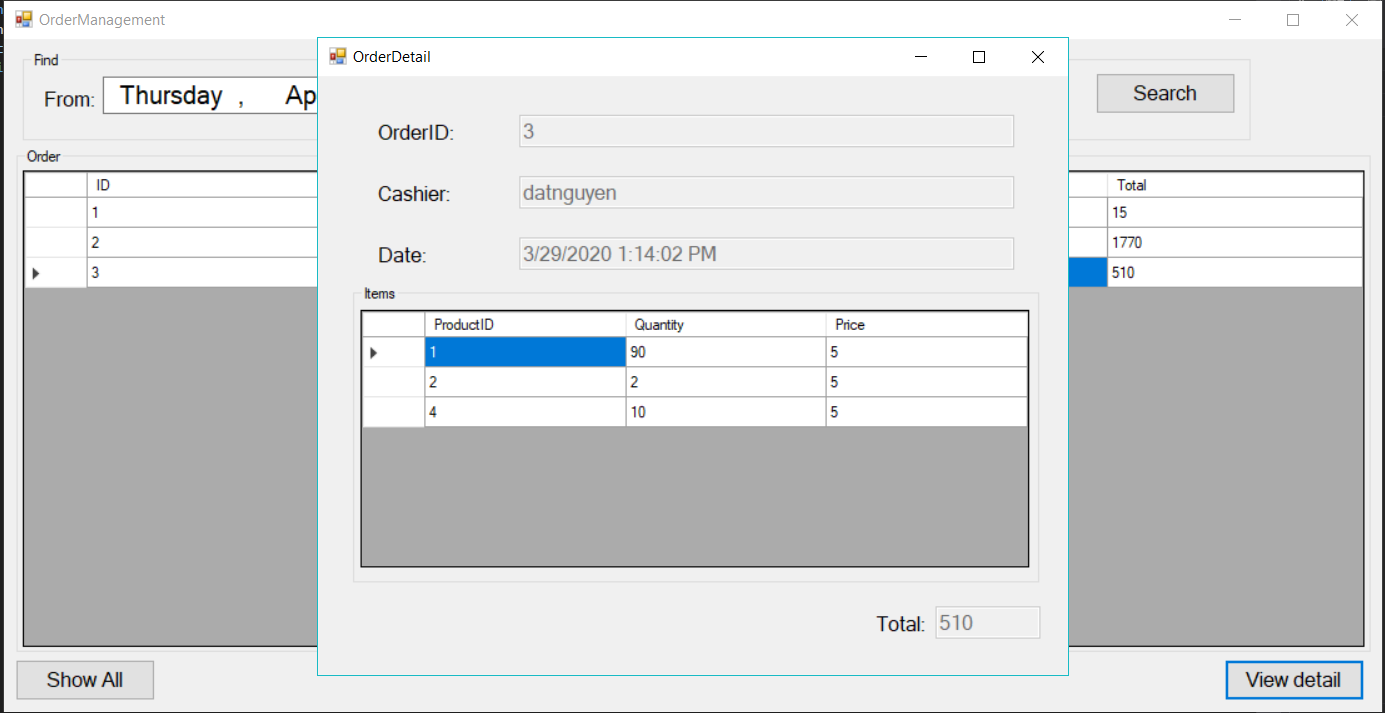
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Search | TextBox |  | String | Input ID’s Product |  |
| Search Object | Button |  | Click | Search ID’s Product is based on txtSearch |  |
| dgvCategory | DataGridView | SelectionChanged |  | View and select Category |  |
| Category ID | TextBox |  | Read Only if not using Action button  String, not null, not duplicate | Input Category ID |  |
| Category Name | TextBox |  | Read Only if not using Action button  String, not null | Input Category Name |  |
| Action Object | Button |  |  | Unlock Add’s function of Category |  |
| Add Object | Button |  |  | Add a Category |  |
| Edit Object | Button |  |  | Modify a Category | True |
| Delete Object | Button |  |  | Delete a Category and all of Products of this Category | True |
| Restore Object | Button |  |  | Restore a Category | Flase |
| dgvProduct | DataGridView | CellClick |  | View and select Product |  |
| Product ID | TextBox |  | Read Only if not using New button  String, not null, not duplicate | Input Product ID |  |
| Name | TextBox |  | String, not null | Input Product Name |  |
| Quantity | TextBox |  | Int, greater than 0 | Input Product Quantity |  |
| Price | TextBox |  | Double, greater than 0 | Input Product Price |  |
| New Object | Button |  |  | Unlock Add’s function of Product | True |
| Add Object | Button |  |  | Add a Product of specific Category |  |
| Update Object | Button |  |  | Modify a Product of specific Category | True |
| Delete Object | Button |  |  | Delete a Product of specific Category | True |
| Restore Object | Button |  |  | Restore a Product of specific Category | False |

* 1. **Order Management Form**



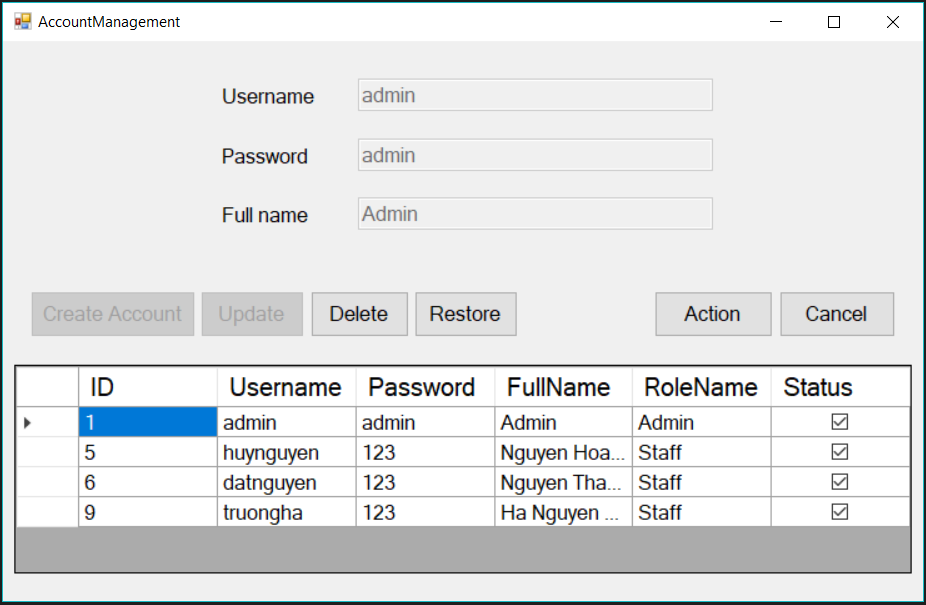
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Form | TextBox |  | Click | Select From Day |  |
| To | TextBox |  | Click | Select To Day |  |
| Search Object | Button |  |  | Select from day to day which is based on Form and To TextBox |  |
| dgvOrder | DataGridView | CellClick |  | View orders |  |
| Show All Object | Button |  |  | Show all orders |  |
| View detail Object | Button |  |  | View detail of specific order |  |

* 1. **Order Detail Form**



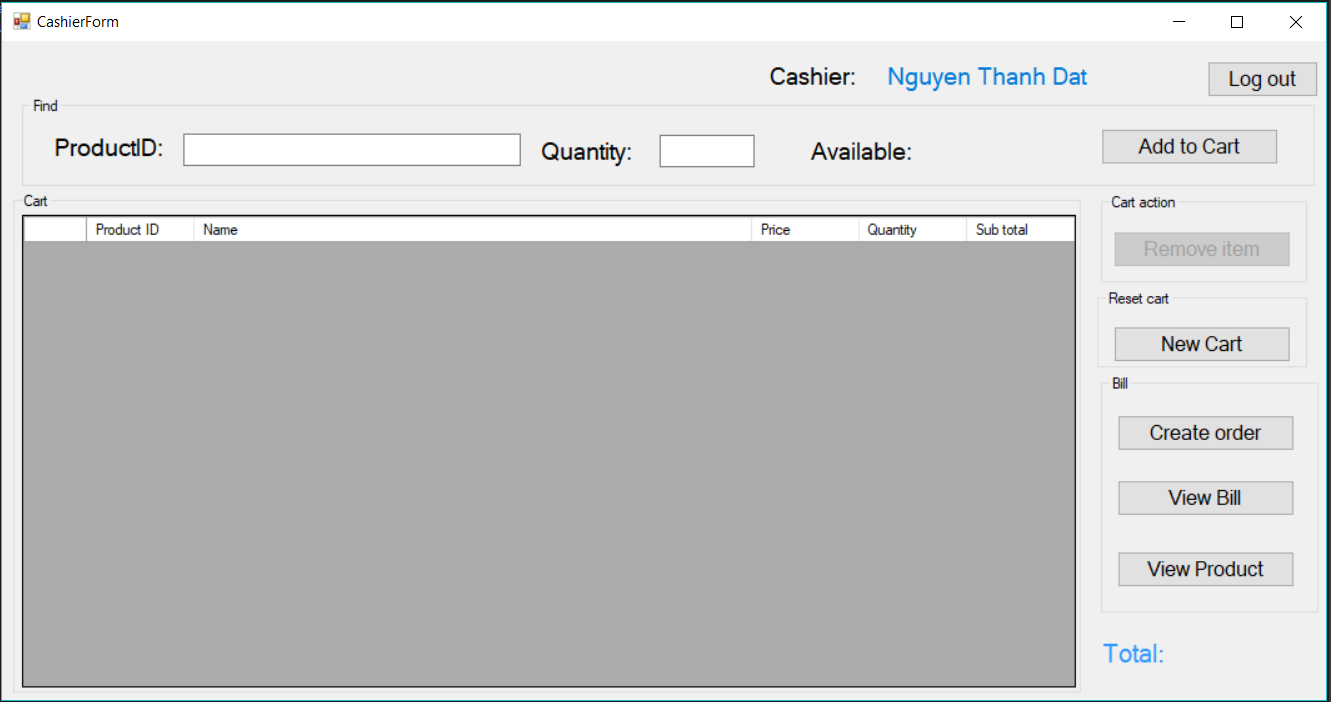
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| OrderID | Label |  | Read Only | View Order ID |  |
| Cashier | Label |  | Read Only | View full name of Cashier |  |
| Date | Label |  | Read Only | View Date Time |  |
| dgvOrderDetail | DataGridView |  |  | View products of this order |  |
| Total | Label |  | Read Only | View total money of this order |  |

* 1. **Account Management Form**



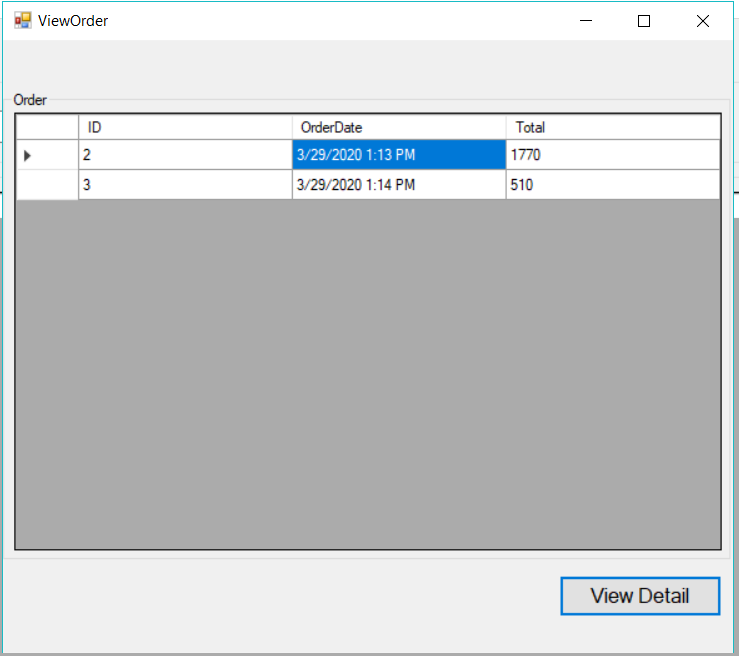
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Username | TextBox |  | Read Only if not using Action button  String, not null | View username |  |
| Password | TextBox |  | Read Only if not using Action button  String, not null | View password |  |
| Full name | TextBox |  | Read Only if not using Action button  String, not null, not number | View full name |  |
| Action Object | Button |  |  | Unlock Create Account or Update’s function |  |
| Create Account Object | Button |  |  | Add new Staff account |  |
| Update Object | Button |  |  | Update account | True |
| Delete Object | Button |  |  | Delete ONLY Staff account | True |
| Restore Object | Button |  |  | Restore Staff account | False |
| Cancel Object | Button |  |  | Cancel action |  |

* **Staff role**
  1. **Cashier Form**



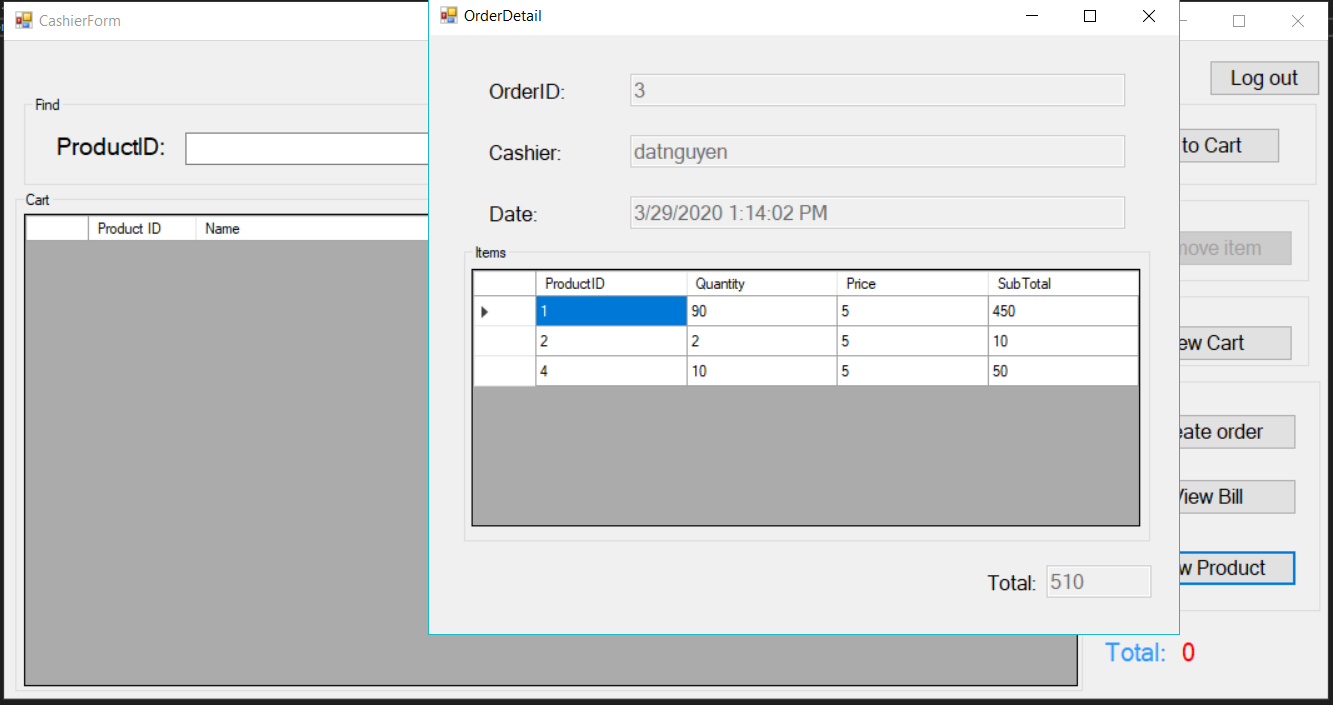
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Cashier | Label |  | Read Only | View full name of the account which logs in |  |
| Product ID | TextBox | KeyDown | String | Select Product ID |  |
| Quantity | TextBox |  | Click | Select Product Quantity |  |
| Available | Label |  | Read Only | Show current quantity product |  |
| Total | Label |  | Read Only | Show total money of the cart |  |
| dgvCart | DataGridView | CellClick |  | View cart |  |
| Add to Cart Object | Button |  |  | Add product to cart and Show it on DataGridView |  |
| Remove Item Object | Button |  |  | Remove product from cart and DataGridView |  |
| New Cart Object | Button |  |  | Reset cart |  |
| Create order Object | Button |  | Not empty cart | Create an order which is based on this cart |  |
| View Bill Object | Button |  | Not empty order | View specific orders which was created by specific staff |  |
| View Product Object | Button |  |  | View Product Management Form |  |
| Log out Object | Button |  |  | Exit and Open Login Form |  |

* 1. **View Order Form**



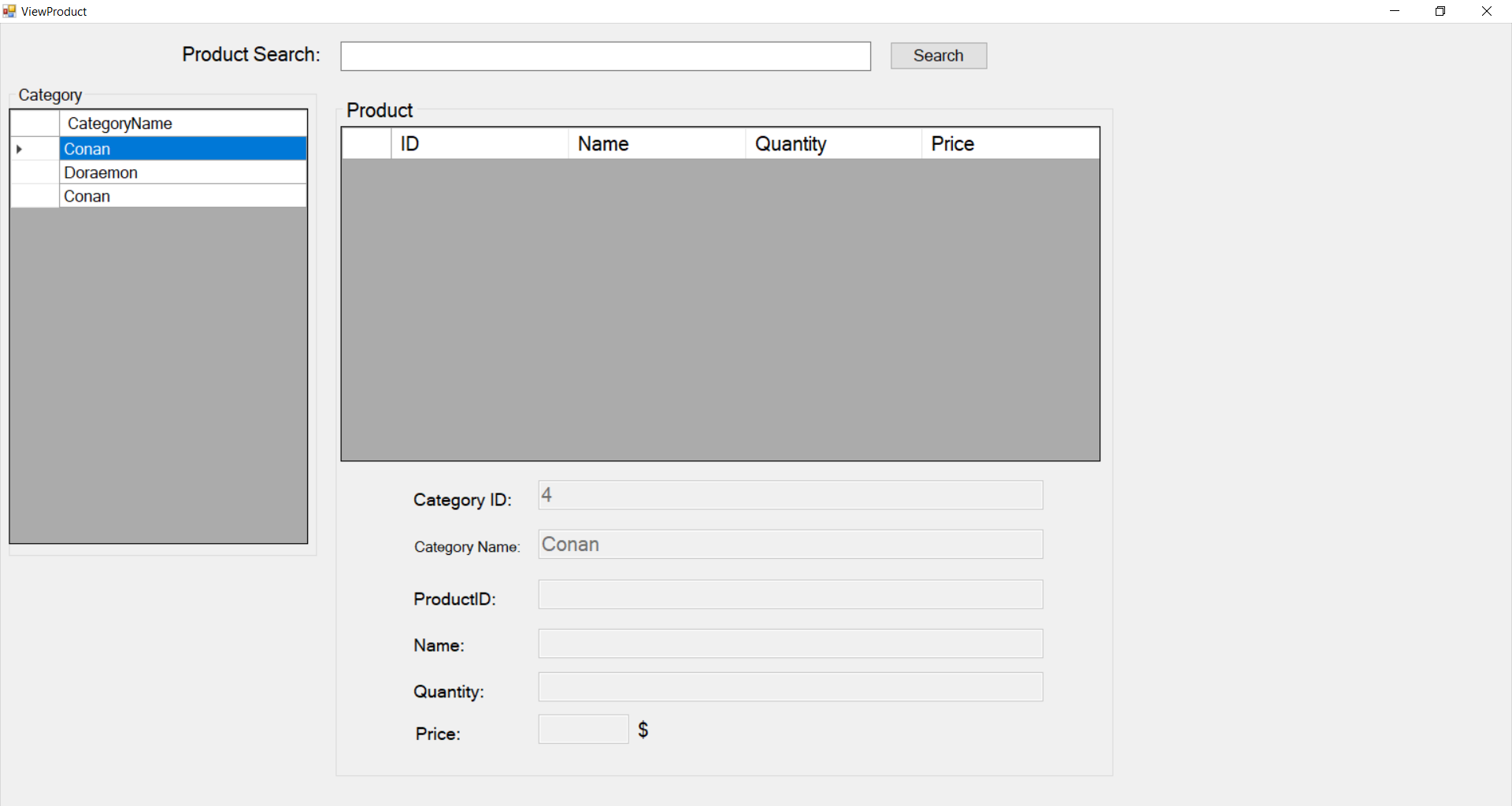
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| dgvOrder | DataGridView | CellClick |  | View specific orders |  |
| View Detail Object | Button |  |  | View detail specific order |  |

* 1. **View Order Detail Form**



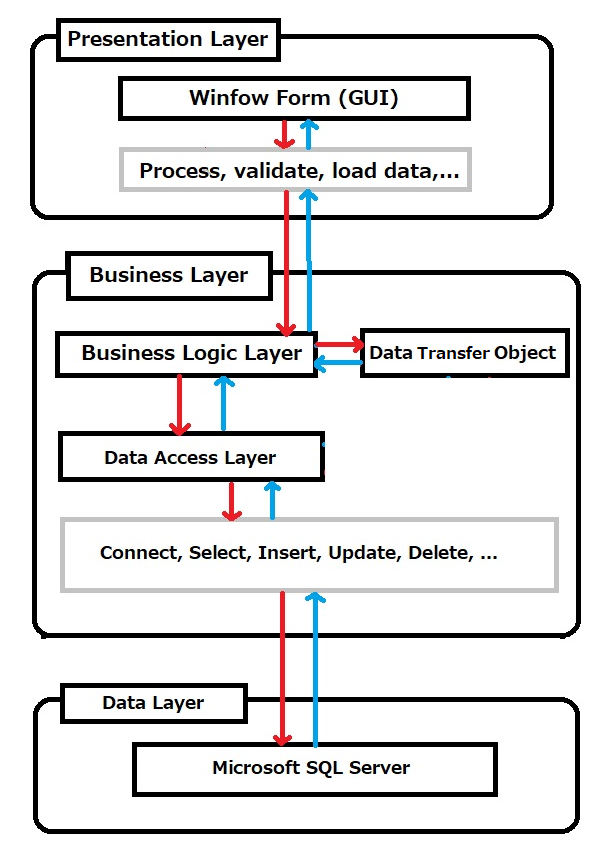
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| OrderID | Label |  | Read Only | View Order ID |  |
| Cashier | Label |  | Read Only | View full name of Cashier |  |
| Date | Label |  | Read Only | View Date Time |  |
| dgvOrderDetail | DataGridView |  |  | View products of this order |  |
| Total | Label |  | Read Only | View total money of this order |  |

* 1. **View Product Form**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **Event** | **Validation** | **Description** | **Status** |
| Search | TextBox |  | String | Input ID’s Product |  |
| Search Object | Button |  |  | Search ID’s Product is based on txtSearch |  |
| dgvCategory | DataGridView | SelectionChanged |  | View and select Category |  |
| Category ID | TextBox |  | Read Only | View Category ID |  |
| Category Name | TextBox |  | Read Only | View Category Name |  |
| dgvProduct | DataGridView | CellClick |  | View and select Product |  |
| Product ID | TextBox |  | Read Only | View Product ID |  |
| Name | TextBox |  | Read Only | View Product Name |  |
| Quantity | TextBox |  | Read Only | View Product Quantity |  |
| Price | TextBox |  | Read Only | View Product Price |  |

1. Architecture Model
   1. **System Architecture Design**



* 1. **Application architecture description**

**The system is developed under 3 Layer architecture style. We choose this architecture for this application because of following advantages:**

* Easy to change a component if we want
* Easy to maintain, and develop the application
* Easy to reuse

**This project follows 3 Layer architecture with following components:**

* **Presentation Layer:** is the part of the application that acts like **GUI**. It means Interface components, which are forms of the program that interact with the user
* **Business Logic Layer (BLL):** Handling program operations such as computation, processing and data integrity.
* **Data Access Layer (DAL):** The communication layer with the database management systems as making connection, executing query **Insert**, **Update**, **Delete**, etc
* **Data Transfer Object (DTO):** is used to transfer object or object list between layers (Presentation and Data Layer)