**Final Report**

Drink Store Management Software

*Team member:*  Phạm Vương Triều - 14520560

Hoàng Trọng Việt - 14520612

Huỳnh Đặng Công Khánh - 14520465

Contents

[1/ INTRODUCTION 1](#_Toc483513665)

[1.1 Purpose of the Requirements Document 1](#_Toc483513666)

[1.2 Product Scope 1](#_Toc483513667)

[1.3 Glossary 1](#_Toc483513668)

[1.4 References 1](#_Toc483513669)

[1.5 Overview 1](#_Toc483513670)

[2/ GENERAL DESCRIPTION 2](#_Toc483513671)

[2.1 Product Perspective 2](#_Toc483513672)

[2.2 Product Functions 2](#_Toc483513673)

[2.3 User Characteristics 3](#_Toc483513674)

[2.5 Assumptions & Dependencies 3](#_Toc483513675)

[3/ SPECIFIC REQUIREMENTS 4](#_Toc483513676)

[3.1 Problem Statements 4](#_Toc483513677)

[3.2 Functions 5](#_Toc483513678)

[3.3 System Functional Requirements 9](#_Toc483513679)

[3.4 System Non-functional Requirement 10](#_Toc483513680)

[3.5 System Interface 10](#_Toc483513681)

[4/ CONCLUSION 15](#_Toc483513684)

[4.1 For lecturer 15](#_Toc483513685)

[4.2 Thank you 16](#_Toc483513687)

# **1/ INTRODUCTION**

* 1. **Purpose of the Requirements Document**

The purpose of this document is:

* To state the requirements for the Drink Store Management System
* To show the analysis of the requirements
  1. **Product Scope**

The product is a software application for computers using Windows operating system and used for people who work at a drink store to manage the store.

* 1. **Glossary**

This subsection contains definitions of all the terms, acronyms, and abbreviations used in the document. Terms and concepts from the application domain are defined.

* Drink Store: an establishment which primary sell drinking product.
  1. **References**

IEEE/ANSI 830-1998

* 1. **Overview**

This document contains the following information:

* General description of product
* Functional & Non-functional requirements

# **2/ GENERAL DESCRIPTION**

The system is designed to be a facility for managing drink store and has many potential applications, such as managing orders, staffs and stocks, assigning sale-off information and generate reports.

This system is intended to support staffs and managers who work or manage the drink store, aims to increase the workload of staffs who work in this kind of store.

* 1. **Product Perspective**

The drink store management system will require users to work on computer. The users of the system do not need to invest in any other software to get the most out of the software system, but their computer must be Microsoft Windows based. Most of features are built-in and does not require any other software to function.

**System interfaces**: Computer must be windows based.

**User interfaces:** All user interfaces occur through an application screen.

**Hardware interfaces:** There is no hardware interface to this system.

**Software interfaces:** There is no software interface to this system.

* 1. **Product Functions**

Our system functions are divided into 5 main categories:

**Product Management**

* Cashier and Stock manager can view/search products
* Stock manager can add product information
* Stock manager can update product information
* Stock manager can delete product information

**Import Management**

* Stock manager can view/search import and import detail bill
* Stock manager can add import and import detail bill
* Stock manager can update import and import detail bill
* Stock manager can delete import and import detail bill

**Order Management**

* Cashier can view/search import and import detail bill
* Cashier can add import and import detail bill
* Cashier can update import and import detail bill
* Cashier can delete import and import detail bill

**Staff Management**

* Manager can view/search staff
* Manager can add staff information
* Manager can update staff information
* Manager can delete staff information

**Report Management**

* View/print imports summary
* View/print orders summary
  1. **User Characteristics**

There are 4 types of users in this system. All of users in the system are authorized users, which are: stock manager, cashier, manager, and admin.

Non-authorized users can’t use any functions of the system; they can only see the welcome & the login screen. After successfully log in to the system, non-authorized users will become authorized users, and their authorization level vary based on their roles.

One type of authorized user is stock manager. Since they are specialized in managing stocks and store ‘s facility, they have more additional privileges than regular staffs in the system, such as: create, update information or delete a product, create stock bills for managing stock and delete stock bills.

The next type of authorized user is cashier. Since they interact with customers, get the data from customers, they have some additional permissions specialize in managing customers, such as: create and cancel order, add, update or delete a product in order detail list, which is a list of products that customers want to have in their orders.

The third type of authorization user is manager. Their specialities managing staffs and report, they have some additional permission as well, such as: create, delete or change staff information, view any staff ‘s information, search for a specific staff, and retrieve report.

The third type of authorization user is admin. Their role is managing and testing all the system and only appear on development side, therefore, they have all permission above.

**General Constraints**

There are some constraints which the system must abide by during development. The system must be developed within these bounds. All are important to be aware of during the implementation of the software system.

* System is to be developed for distributed use as a windows-based application for computers.
* System is to be developed in C#, using Visual Studio.
* Data is stored in a local database.
* Each user has their own authorization based on their roles. A specific user can’t have permissions that is not necessary for his/her role.
  1. **Assumptions & Dependencies**
* System will be installed on a machine running Windows operating system with internet connection.
* Only staffs are authorized to use the system.
* Users are familiar in using window-based application.
* Database server working normally.

# **3/ SPECIFIC REQUIREMENTS**

* 1. **Problem Statements**

Smart book-coffee is a management system for staffs to manage the book-coffee shop with these following functions:

* **Order Management:**

Cashier can use the app to view lists and details of orders in their shift as well as search for item in order bill base on name. The app will display information of order such as: date order and total bill.

With every bill created, items are automatically checked by the system and show notifications to warehouse manager whether any products are out of stock.

Moreover, the warehouse manager is also in charge of updating or deleting whenever the shop needs to.

**Product Management:**

Stock manager can use the app to view product lists, details as well as search for product.

The app will display: Product name, brand, category, and unit price.

Cashiers also have the right to search and select product in product search form for their order bills.

Stock manager is in charge of adding and updating product’ information when new books are imported.

* **Staff Management:**

Shop manager can search for employees’ information or update employees’ information (if needed) based on name and position.

Staff lists can be added, updated and deleted.

The app displays staffs’ information including name, username, gender, birthday, position, salary, address and phone number.

* **Import Management:**

Stock receiving has their own type of bills that are different from selling bills. Each bill MUST have these information:

* Types of items
* Quantities
* Price
* Supplier
* Name of warehouse manager (created the bill)
* Stock receiving time
* Total payment

Stock managers are in charge of updating this information or cancelling if there are any mistakes happen.

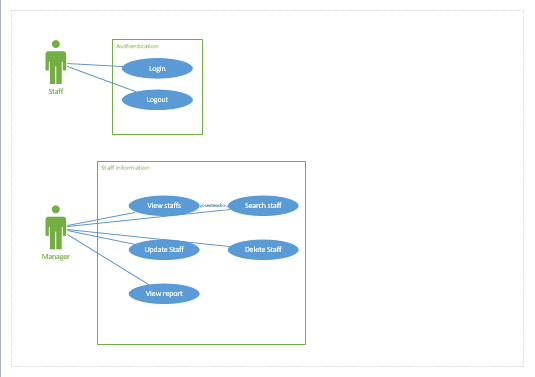
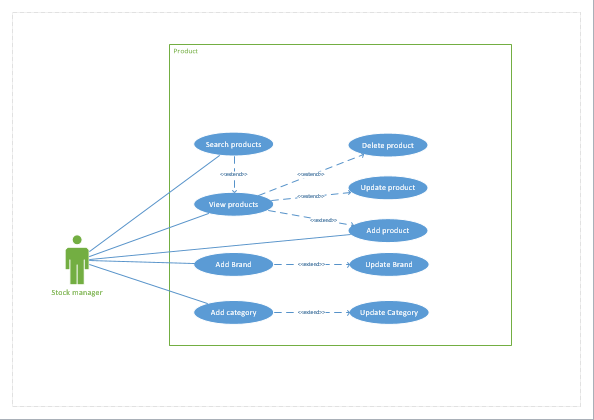
For each item (ingredients, books), the warehouse manager must add the item into the system before creating order.

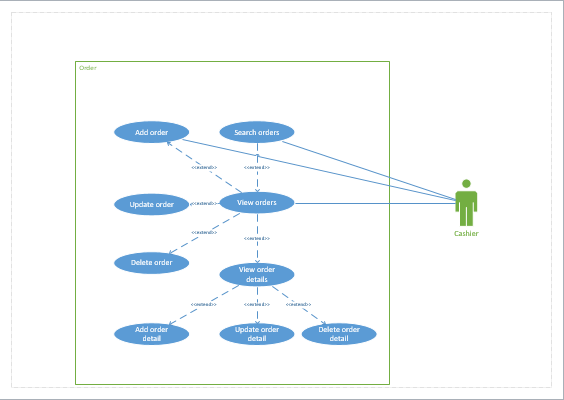
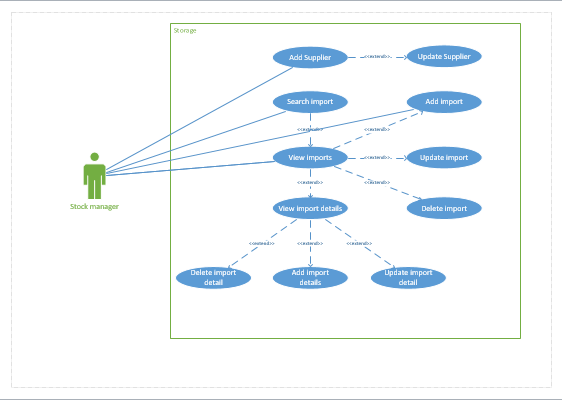
* 1. **Functions**

System functional requirements are specified by use cases and specific diagram. The use case helps understand system behaviours.

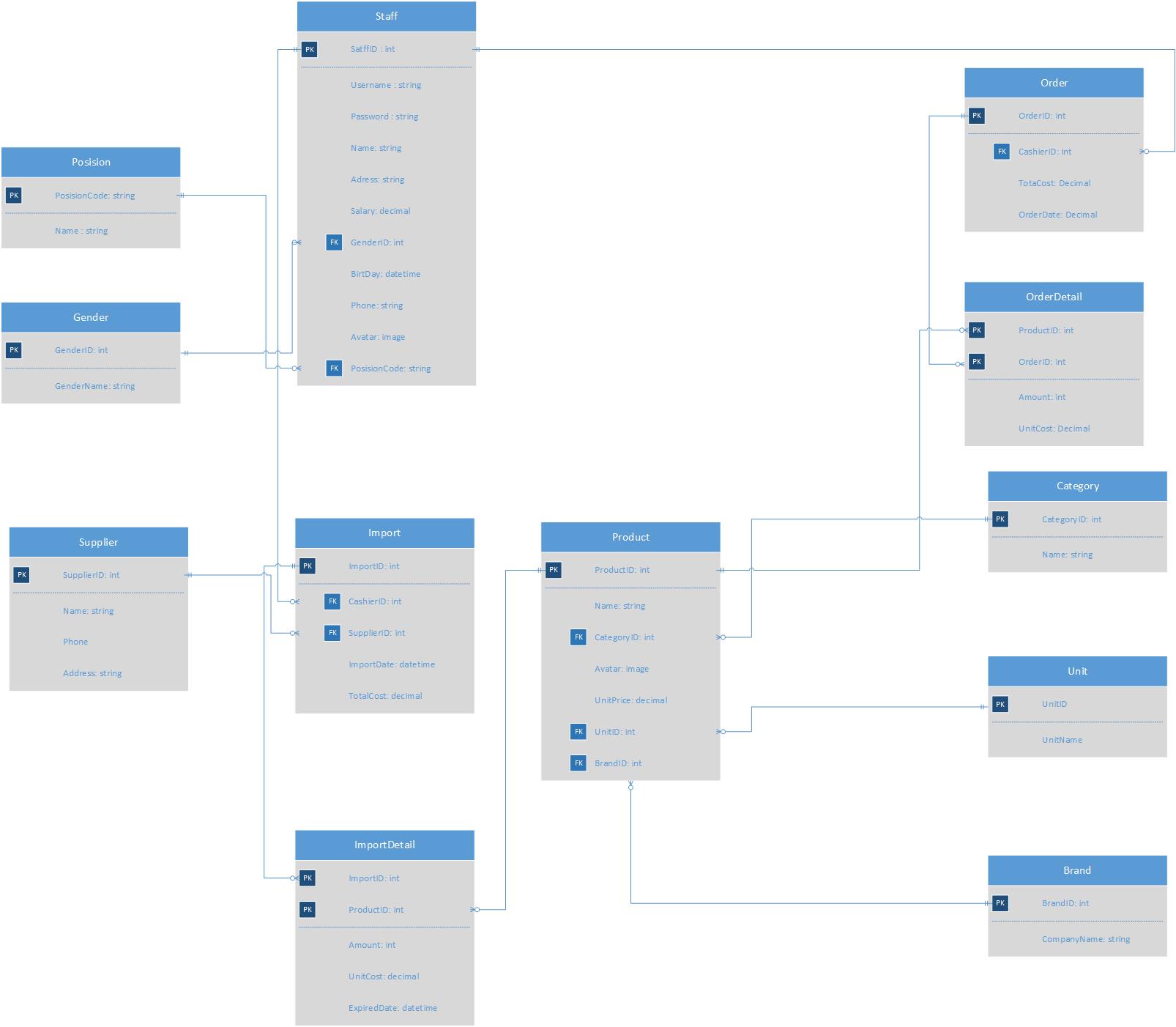
(Sequence diagram is too long to include in this document, so please open them in the same folder file to view details)

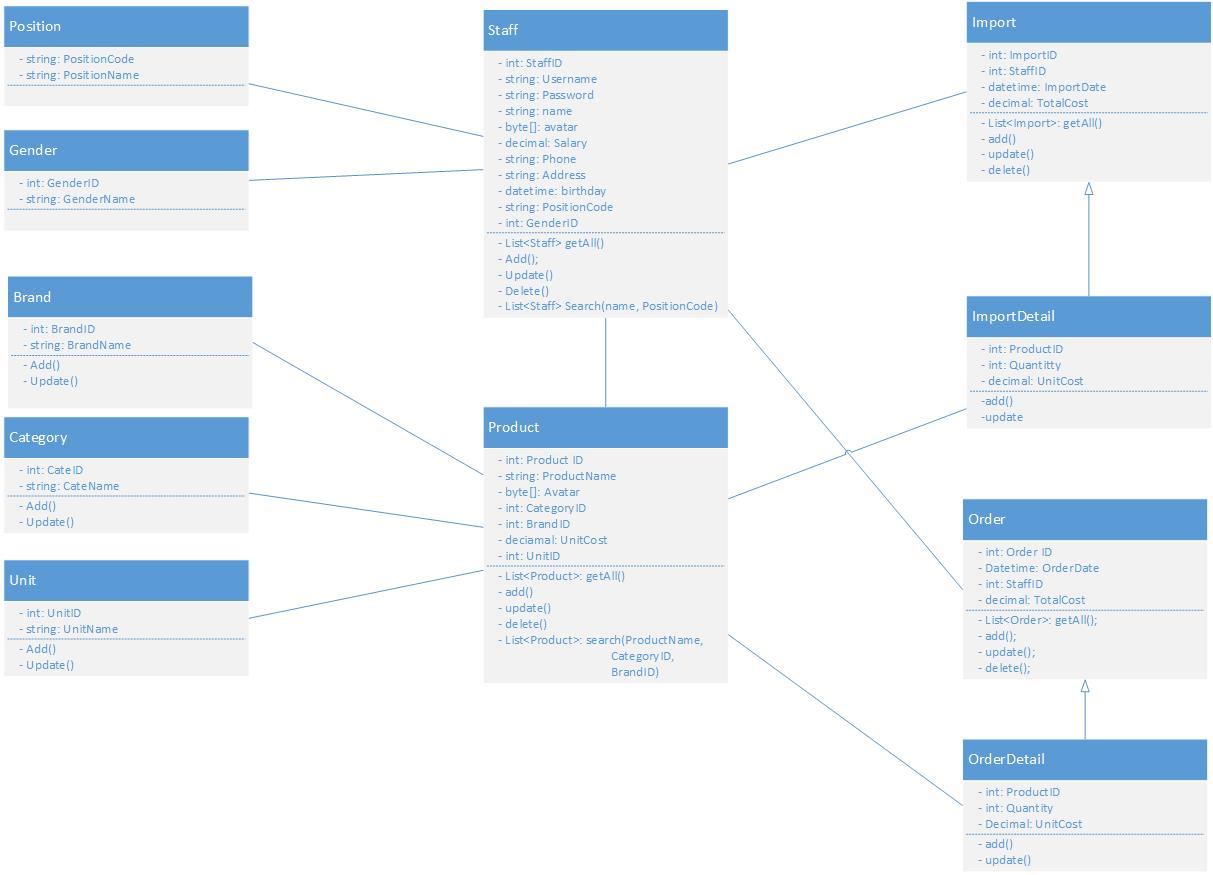
Below is the system use-case diagram to help understanding of the system function:





Below is the system ERD diagram :

****

****Below is the system UML diagram :

* 1. **System Functional Requirements**

Based on the previous problem statement, we state the following functional requirement for the system:

* **Create orders**

The system shall assist users with “cashier" role to create orders for customers. (All check-outs will be done at the reception counter, where cashier works)

An order will be saved with information input from the cashier and self-generated information, such as:

Order ID (automatically generated)

Products

Quantity (each item)

Price (each item) (generated based on products)

Customer’s total payment (automatically generated)

Staff ID of in charge cashier (get from the database)

Date that bill is created & exported (get current day)

When create a order, the system allows the cashier to obtain and choose product via a product list.

* **View/Search orders**

The system shall display the orders that are created.

The system also support search for orders in history based on date.

* **Delete orders**

The system assist users with “cashier” role to delete the orders that are not finished if some mistakes were made.

* **Update orders**

Cashier can also update other order information after the order was made.

* **Search order item**

The system assist users to search for a specific order item by name.

* **Out-of-stock notification**

The system can detect whether products are not enough and displays a warning message to users.

* **Add product information**

The system assist users with “stock manager” role to add a product information

* **Update product information**

The system assist users with “stock manager” role to update a product information.

* **Delete product**

The system assist users with “warehouse management” role to delete a product

* **View product list/details**

The system assist users to view a list of products the store owned or view a book details by clicking on a product item (product’s name, image, brand, category, price)

* **Search product**

The system assist users to search for a specific book based on stored information (product’s name, brand, category)

* **View/Search staff information/schedule**

Only managers can see information of any staffs in the store or search for staffs based on name and position.

* **Create/Cancel import bill**

The system allows warehouse managers to create an import bill when he/she restocks or cancel a stock bill if any mistake was made.

When a warehouse manager wants to create a new import, he/she must add a new product into the system first.

* **View/Search stock items**

The system allows users to view a list of available items in the stock or search for an item by name.

* **Add/Update/Delete ingredients**

The system allows stock manager to view, update or delete an import detail bill

* 1. **System Non-functional Requirement**

Based on the previous problem statement, we state the following non-functional requirement for the system:

* **Automatically update**

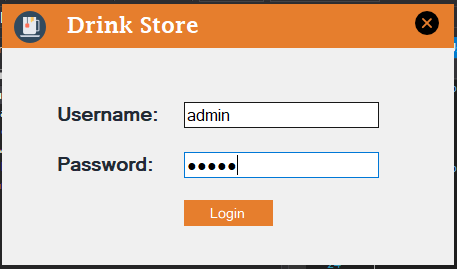
Whether an order is made, the system shall update the status/amount/information of related items immediately before any other events take place without user ‘s help.

* **Usability**

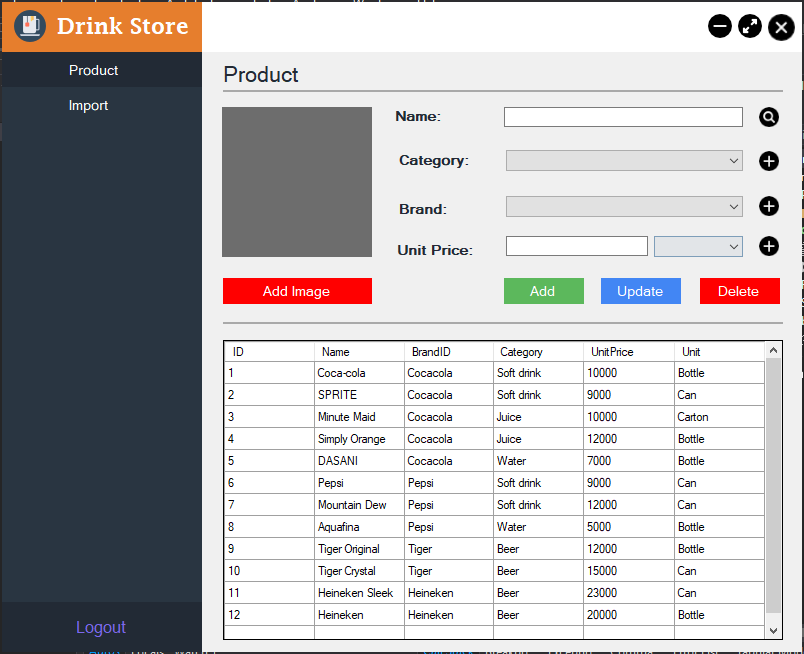
The system should be usable by non-experts with a maximum of 2 hours of training.

* 1. **System Interface**

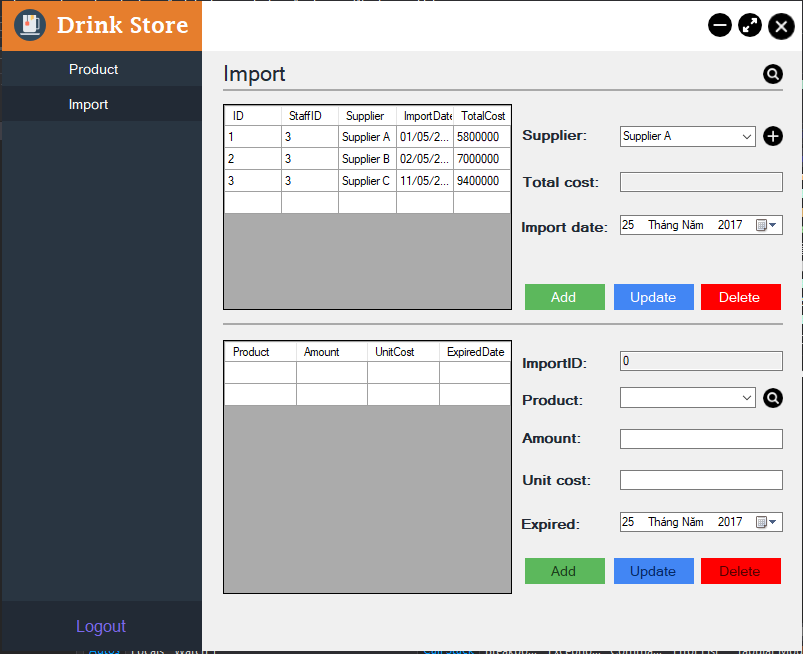
**Login Form:**

****

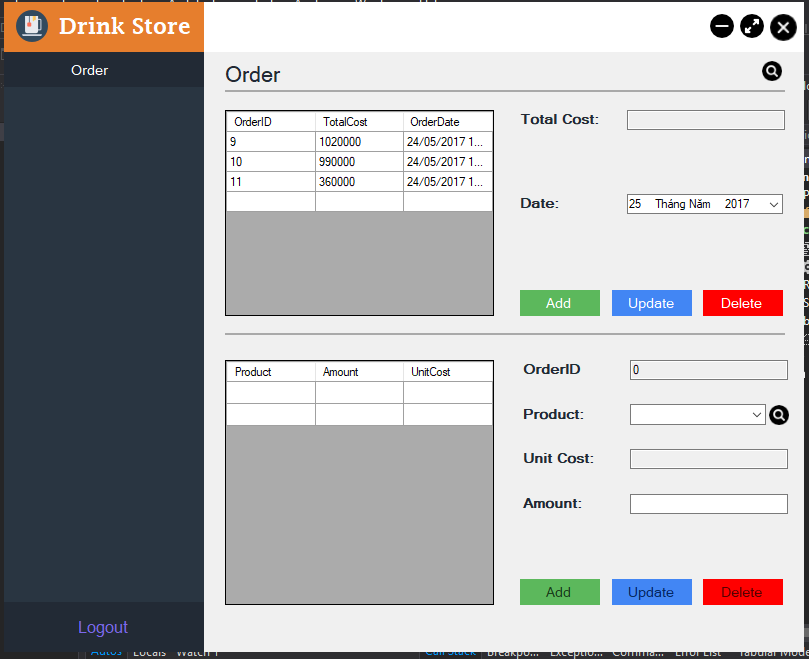
**Product form:**

****

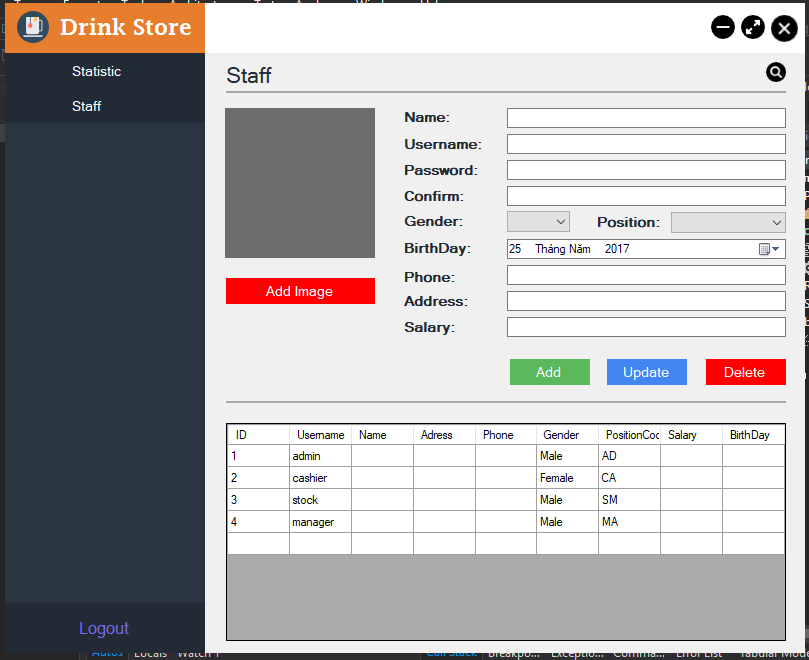
**Import form:**

****

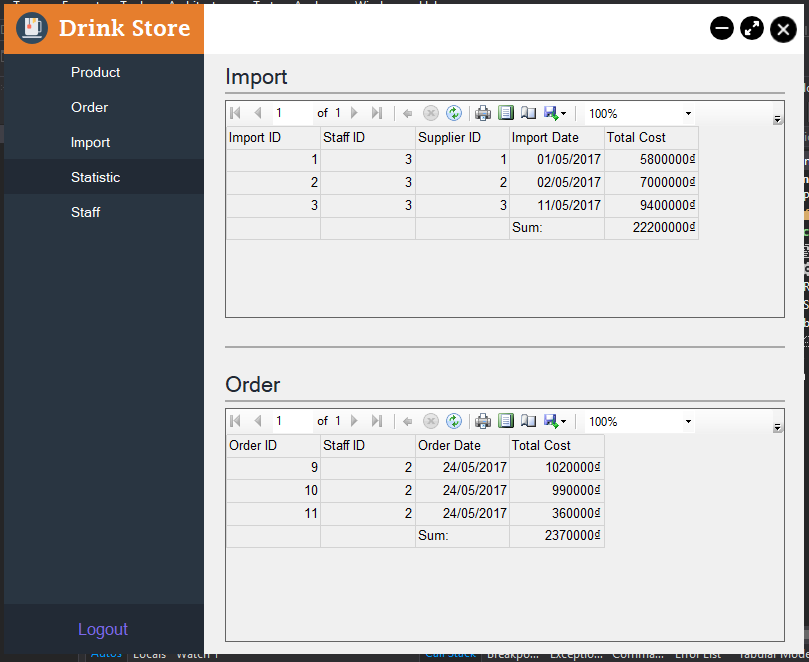
**Order form:**

****

**Staff form:**

****

**Report form**

****

# **4/ CONCLUSION**

**4.1 For lecturer**

Our program completed in three mouth and every member of the team had tried their best to learning and coding. Here is our work assignment:

|  |  |  |
| --- | --- | --- |
| Task | Person in charge | Status |
| Use-case diagram | Triều | Complete. |
| Class diagram | Việt | Complete. |
| ERD | Khánh | Complete. |
| Activity | Việt | Complete. |
| Sequence | Triều | Complete. |
| GUI designed | Triều | Complete. |
| Database | Khánh | Complete. |
| Data-Layer | Khánh | Complete. |
| Business-Layer | Việt | Complete. |
| Presentation-Layer | Triều | Complete. |
| Login Form | Khánh | Complete. |
| Import Form | Việt | Complete. |
| Product Form | Triều | Complete. |
| Order Form | Khánh | Complete. |
| Staff Form | Triều | Complete. |
| Testing and Validating | All team members | Complete. |
| Report | Việt | Complete. |

**4.2 Thank you**

Thanks Mr.Anh for your great lectures and advices. This project cannot complete without your guidance.