Even Semester (2023)

**BINUS UNIVERSITY**
**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Solo Group Work)**

| **Student Information**: | **Surname** | **Given Names** | **Student ID Number** |
|---|---|---|---|
| 1. | Frans | Christensen Mario | 2301963316 |

| | | | |
|---|---|---|---|
| **Course Code** | **:** COMP6571 | **Course Name** | **:** Data Structures and Algorithms |
| **Class** | **:** L2BC | **Name of Lecturer(s)** | **:** Mr. Tri Asih Budiono |
| **Major** | **:** CS | | |
| **Title of Assignment** | **:** Online Pharmacy System | | |
| **Type of Assignment** | **:** Final Project | | |
| **Submission Pattern** | | | |
| **Due Date** | **: 18-06-2019** | **Submission Date** | **:** |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer's instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

## Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offense which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other

possible penalties executed by the university. Please refer to the related course syllabus for further information.

## Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

| (Name of Student) | Signature of Student: |
| --- | --- |
| 1. Christensen Mario Frans | CHRISTENSEN MARIO FRANS |

# Table of Contents

# Project Problem & Solution

### Project Background

Ever since the introduction of the internet, online shopping has been increasing dramatically in modern days. It allows individuals to purchase the goods and services they want without having to leave their homes, which may be time consuming, also they have to pay for transportation fees (such as gas, taxi, train, bus, etc.).

It is very common to hear people buying fashion related items online, home improvement products, technology accessories, and many more. However, it is very rare that people to buy medicines/healthcare equipment online. Whereas healthcare is one of the most important necessity in our daily lives.

## Problem & Motivation

In recent situations, due to the COVID-19 pandemic, many pharmacies run out of stock for healthcare products, such as masks and hand sanitizers. It is very common to see customers coming into pharmacies, only to leave 5 seconds later after being told what they're looking for is out of stock. For the customer, not only this is a waste of time and transportation cost, it also increases the chances of them being infected (as they are leaving their homes to look for these items), and eventually spread the virus to their relatives at home. It was inefficient after all… so why can't we just find and buy the items online? This project motivates me as it can help people save their lives instead of risking it.

### Solution & Aim

My solution for this problem is to make a pharmacy system for people to buy medicines online with ease. The aim of this program is to show users that it is also entirely possible to shop online for medicines and healthcare equipment just like other niches. Moreover, to create a user-friendly interface for the user

### Project Scope

This system will include:

- Items as well as their stock count for users to see which products are available
- Create a customer's cart,  and checkout system
- User database to save login information (so users do not have to re-register)

- Pharmacy owner account (to view customers, items sold, current stocks for specific products, adding and removing stocks from the pharmacy, and to view revenue generated)

**Limitation of This Project**

**I will not** be implementing this program to an actual online pharmacy; this program is only a 'blueprint program' to show the system and how the logic will work.

# Data Structure & Implementation

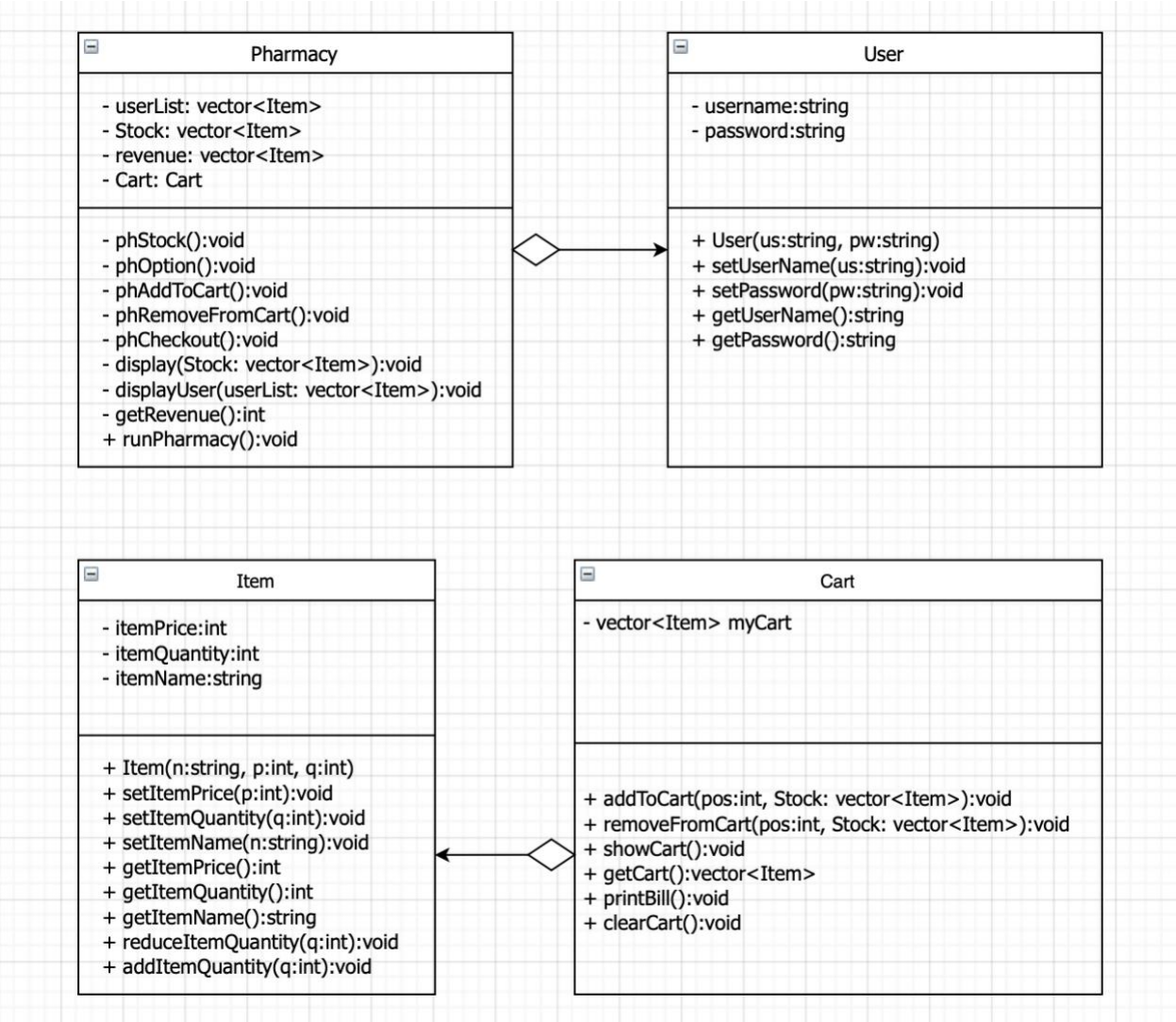
The data structure that I will be using for this program is Vector STL. Vectors are known as dynamic arrays with the ability to resize itself automatically when an element is inserted or deleted, with the storage is being handled automatically by the container.

In my opinion, this is a suitable data structure to be addressed to the solution as it is very similar to arrays, however, its size can be modified while the program is running, unlike arrays, which cannot be resized once instantiated. This is an important feature of vector that's crucial in this program as the main target of this program is to show the users the stocks and inventory and add/remove the inventory item/s from the Pharmacy that they are looking for, therefore, vector STL will be very helpful for this program.

The method I will be using to implement this Vector data structure reading the data from a text file, and use vector STL to modify these values dynamically. Other aspects that will be used in the program are aggregation, objects, classes, methods, and variables.

I will also be studying and using aggregation relationships. Aggregation is known as 'Has a' relationship in C++ programming. In my program, some classes will have an aggregation relationship with another class. This kind of relationship is useful for my program, as seen in the UML Diagram below; Pharmacy has a/many User, and Cart has a/many items.

**UML Diagram**




In the UML Diagram, the aggregation relationship is shown using an arrow with a diamond at the back, there are 4 different classes with different variables and methods in it. In addition, I will be adding 3 text files to be used as the database for the pharmacy stocks, user database, and the revenue generated. These files will be opened and the data in it will be put into a vector, modified by the vector, and updated into these files after the program is used.

# The Program

**Program Manual & Results of Execution**

```
=====================================
Welcome to the Pharmacy
To continue, please select one of the following:
1. Login
2. Register
=====================================
Input:
```

When the user enters the program, he/she is greeted with 2 options; login, or register. If this is the first time the user is using the program, and he/she has no account yet, simply choose the register option by inputting the chosen index. (Example: To register, input: 2) If the user already has an account, simply choose option 1 to login.

*Note: It's not necessary to always choose register; if the user had already registered in the past, they can choose login right away.*

```
=====================================
Welcome to the Pharmacy
To continue, please select one of the following:
1. Login
2. Register
=====================================
Input: 2
=====================================
To register, please create a new account
Create new username: mario
Create new password: frans
Confirm new password: frans
```

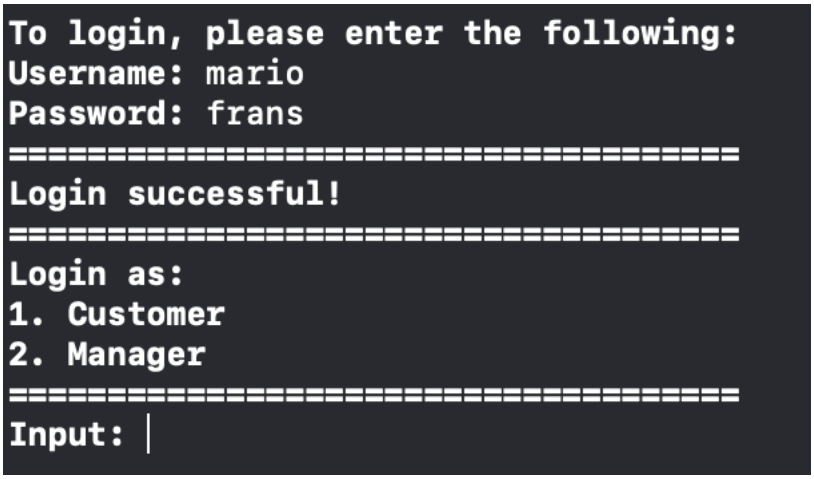
If we choose register, which means we create a new user account, we need to input a new username and password. The program will also ask the user to confirm the password to make sure that they typed in the correct password to be used.

```
To register, please create a new account
Create new username: mario
Create new password: frans
Confirm new password: frans
=====================================
Welcome to the Pharmacy
To continue, please select one of the following:
1. Login
2. Register
=====================================
Input:
```

Next, after registering, the program will take the user back to the welcome page again. Now with the registered username and password, the user is able to login with it. Simply input '1'.

```
=====================================
Welcome to the Pharmacy
To continue, please select one of the following:
1. Login
2. Register
=====================================
Input: 1
=====================================
To login, please enter the following:
Username: mario
Password: fran
=====================================
Your username/password is incorrect!
=====================================
To login, please enter the following:
Username:
```

Make sure to input the right password & username. If either one/both are incorrect, this error message will show up, and the user will be asked to input the username and password again




Once login is successful, the user will be asked to login either as the customer or manager. To add items to cart, proceed to checkout, pay and view bill, simply choose login as ‘Customer’. Otherwise, to view or add item/s to pharmacy stock, and view financials, simply pick login as ‘Manager’.

```
Login as:
1. Customer
2. Manager
====================================
Input: 1
================================================
    Product                 Price          Quantity
================================================
1 . Panadol               | Rp.5000        | 100
2 . Face Mask(50Pcs)      | Rp.50000       | 100
3 . Hand Sanitiser        | Rp.20000       | 100
4 . Tolak Angin           | Rp.20000       | 100
5 . Counterpain           | Rp.30000       | 100
6 . Enervon-C             | Rp.25000       | 100
7 . Thermometer           | Rp.100000      | 10
8 . Redoxon               | Rp.40000       | 100
9 . Hansaplast            | Rp.10000       | 100
10. Mineral Water         | Rp.3000        | 100
11. Mineral Water         | Rp.3000        | 100
================================================
====================================
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
====================================
Input:
```

If the user chooses login as customer, the program will automatically show the contents of the Pharmacy stock, this includes their respective item name, price (per quantity), and the quantity/stock of the items as shown above. Next, the user is able to add item/s to their cart, remove item/s from their cart (if they already have something in their cart, view their cart, and also initiate checkout. The user can also choose to exit, which leads them back to the welcome page.

```
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
==================================
Input: 1
============================
Cart Items:
============================
Input item index to be added to cart
2
Quantity:
3
Face Mask(50Pcs) successfully added to cart!
```

If the user chooses add item/s to cart, they will be asked to input the item index (shown in the contents of the Pharmacy stock, and the quantity that they choose to add to their cart. If the item index and quantity inputted matches the stock index and stock quantity respectively, then a message will show: “(item name) successfully added to cart!”

```
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
====================================
Input: 3
============================
Cart Items:
============================
1. Face Mask(50Pcs) Quantity: 3
```

Since we’ve added something to our cart, we are now able to view our cart, and the item we just added will show up here.

```
Cart Items:
============================
1. Face Mask(50Pcs) Quantity: 3
2. Panadol Quantity: 5
3. Hand Sanitiser Quantity: 2
================================================
    Product                 Price          Quantity
================================================
1 . Panadol               | Rp.5000        | 95
2 . Face Mask(50Pcs)      | Rp.50000       | 97
3 . Hand Sanitiser        | Rp.20000       | 98
4 . Tolak Angin           | Rp.20000       | 100
5 . Counterpain           | Rp.30000       | 100
6 . Enervon-C             | Rp.25000       | 100
7 . Thermometer           | Rp.100000      | 10
8 . Redoxon               | Rp.40000       | 100
9 . Hansaplast            | Rp.10000       | 100
10. Mineral Water         | Rp.3000        | 100
11. Mineral Water         | Rp.3000        | 100
================================================
====================================
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
====================================
Input:
```

We can also add multiple items to the cart and view them all at once! Moreover, as seen in the picture above, the quantity of the items left in the Pharmacy Stock decreased from their initial quantity , this is because the user had already added some of them to the stock.

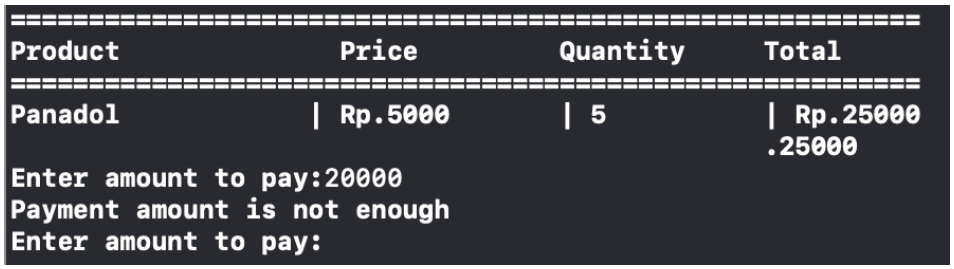
```
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
====================================
Input: 2
============================
Cart Items:
============================
1. Face Mask(50Pcs) Quantity: 3
2. Panadol Quantity: 5
3. Hand Sanitiser Quantity: 2
Input index number of item to be removed from the cart
1
Quantity:
3
Face Mask(50Pcs) removed from cart
```

To remove an item from the cart, enter index ‘2’ and the program will display the items in the cart, and the user chooses the item to be deleted as well as it’s quantity.

```
Input index number of item to be removed from the cart
1
Quantity:
3
Face Mask(50Pcs) removed from cart
================================================
    Product                Price         Quantity
================================================
1 . Panadol              | Rp.5000       | 95
2 . Face Mask(50Pcs)     | Rp.50000      | 100
3 . Hand Sanitiser       | Rp.20000      | 98
4 . Tolak Angin          | Rp.20000      | 100
5 . Counterpain          | Rp.30000      | 100
6 . Enervon-C            | Rp.25000      | 100
7 . Thermometer          | Rp.100000     | 10
8 . Redoxon              | Rp.40000      | 100
9 . Hansaplast           | Rp.10000      | 100
10. Mineral Water        | Rp.3000       | 100
11. Mineral Water        | Rp.3000       | 100
================================================
```

Now, as seen in this picture, after we removed all of the ‘Face Mask(50Pcs)’ item as an example, the quantity in the stock goes back up to 100, which is the initial quantity of the item.


Next, the user may want to proceed to checkout. They will have to input index ‘4’. They will then be taken to the see the bill page. They can now view all the items that they added to cart, with the price, quantity, and the total amount for each item (which is calculated by the price multiplied by quantity of that specific item). The grand total of the bill will also be shown at the bottom of all the total.




Then, the user will be asked to input the amount of money to pay the bill. If this amount less than the bill, an error message will show up to tell the user that their amount is not enough to purchase the items on the bill. They will then be asked to re-enter the amount to pay.

```
====================================
1. Add item/s to your cart
2. Remove cart item/s
3. View Cart
4. Initiate Checkout
5. Exit
====================================
Input: 4
============================================================
Product              Price         Quantity     Total
============================================================
Panadol            | Rp.5000       | 5          | Rp.25000
                                                .25000
Enter amount to pay:30000
                                          Change: .5000

============================================================
Thank you for purchasing from us!
We will be sending out your item as soon as possible!
============================================================
```

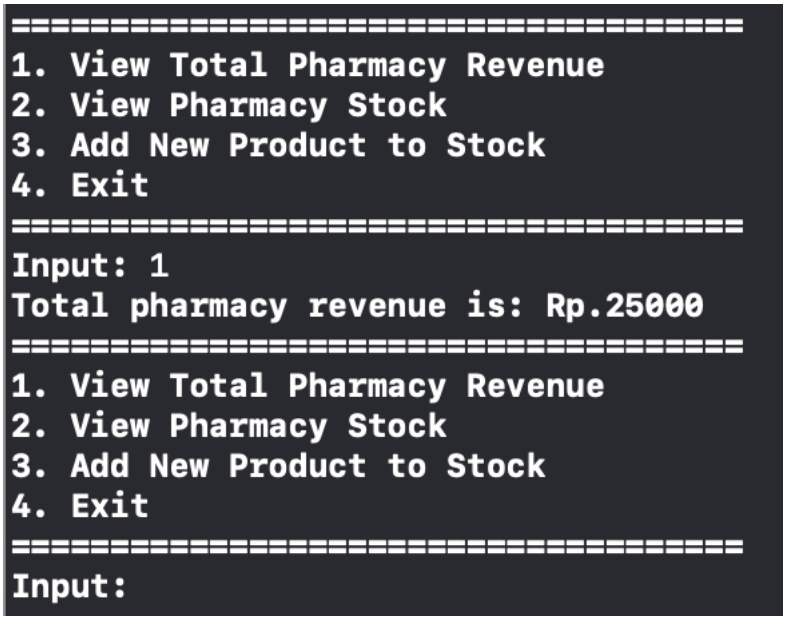
Otherwise if it is larger than the bill, the program will print the change for the customer, and the transaction is complete. A thank you message will be printed.

```
Thank you for purchasing from us!
We will be sending out your item as soon as possible!
============================================================
====================================
Welcome to the Pharmacy
To continue, please select one of the following:
1. Login
2. Register
====================================
Input: 1
====================================
To login, please enter the following:
Username: mario
Password: frans
====================================
Login successful!
====================================
Login as:
1. Customer
2. Manager
====================================
Input: 2
====================================
1. View Total Pharmacy Revenue
2. View Pharmacy Stock
3. Add New Product to Stock
4. Exit
```

Following the thank you message, the user will be directed back to the welcome page, they can repeat the process by logging in, or they can register for another account. Since we had already purchased from the store as a customer before, let us now login as the manager.

```
Login as:
1. Customer
2. Manager
====================================
Input: 2
====================================
1. View Total Pharmacy Revenue
2. View Pharmacy Stock
3. Add New Product to Stock
4. Exit
====================================
Input: 2
================================================
    Product                 Quantity
================================================
1 . Panadol               | 95
2 . Face Mask(50Pcs)      | 100
3 . Hand Sanitiser        | 100
4 . Tolak Angin           | 100
5 . Counterpain           | 100
6 . Enervon-C             | 100
7 . Thermometer           | 10
8 . Redoxon               | 100
9 . Hansaplast            | 100
10. Mineral Water         | 100
11. Mineral Water         | 100
================================================
```

As the manager, we are able to see the financials of the business, view leftover stocks of the pharmacy, and even add new products to stock. If we choose to view stock, we can see that ‘Panadol’ item is has only 95 quantity left. This is because we just bought 5 Panadols earlier.




As the manager, we can also view the Pharmacy's financials, such as its the revenue too. This is done by selecting 'View Total Pharmacy Revenue'. It can be seen that the current revenue generated is Rp.25,000. This is because we had purchased 5 Panadols earlier, totaling exactly Rp.25,000.

```
1. View Total Pharmacy Revenue
2. View Pharmacy Stock
3. Add New Product to Stock
4. Exit
====================================
Input: 3
Name of New Product: Candy
Price of New Product: 10000
Quantity of New Product: 100
====================================
1. View Total Pharmacy Revenue
2. View Pharmacy Stock
3. Add New Product to Stock
4. Exit
====================================
Input: 2
=============================================
    Product               Quantity
=============================================
1 . Panadol             | 100
2 . Face Mask(50Pcs)    | 100
3 . Hand Sanitiser      | 100
4 . Tolak Angin         | 100
5 . Counterpain         | 100
6 . Enervon-C           | 100
7 . Thermometer         | 10
8 . Redoxon             | 100
9 . Hansaplast          | 100
10. Mineral Water       | 100
11. Mineral Water       | 100
12. Candy               | 100
=============================================
```

Finally, the manager is also able to add new product/s to the pharmacy's stock. When doing this, he/she has to input the new item's name, selling price, and also its available quantity. Example: the user wants to add a new item; 'Candy' to the pharmacy, and its quantity is 100. Once the user had inputted those data, if we view the updated pharmacy stock, we will see that there will be a new 'Candy' on the bottom of the product list, which we just added.

This concludes the program manual. Thank you

# Program and Demo Link

## Link to Project in GitHub:

https://github.com/mariofrans/S2_Project_CPP_Pharmacy

## Link to Demo Video in OneDrive:

https://1drv.ms/u/s!AjvcnoBWp9nS3yOKi27Dij16KGAG?e=QwQCyS