Programming Concepts in C++

PURCHASE ORDER MANAGEMENT SYSTEM



UC2F1908(IS)

Matthew Axell

TP049057

Date Assigned: 24 March 2020

Date Completed: 17 May 2020

**Table of Contents**

[1 Object-Oriented Programming Concept 3](#_Toc40627995)

[1.1 Abstraction 3](#_Toc40627996)

[1.2 Encapsulation 3](#_Toc40627997)

[1.3 Inheritance 4](#_Toc40627998)

[1.4 Polymorphism 4](#_Toc40627999)

[2 Design 7](#_Toc40628000)

[2.1 Use Case Diagram 7](#_Toc40628001)

[2.2 Activity Diagram 8](#_Toc40628002)

[2.2.1 Manage User Database 8](#_Toc40628003)

[2.2.2 Manage Purchase Order Database 9](#_Toc40628004)

[2.2.3 View Purchase Order 10](#_Toc40628005)

[2.2.4 View Entries 10](#_Toc40628006)

[2.2.5 View Purchase Requisition 11](#_Toc40628007)

[2.2.6 Manage Purchase Requisition Database 12](#_Toc40628008)

[2.2.7 Manage Entries Database 13](#_Toc40628009)

[3 Screenshot of The Program’s Output 14](#_Toc40628010)

[4 Limitation 29](#_Toc40628011)

[5 Conclusion 30](#_Toc40628012)

[6 References 31](#_Toc40628013)

# Object-Oriented Programming Concept

## Abstraction

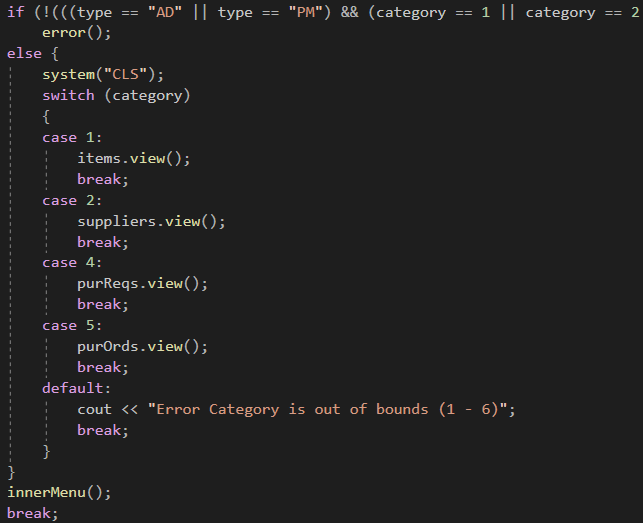
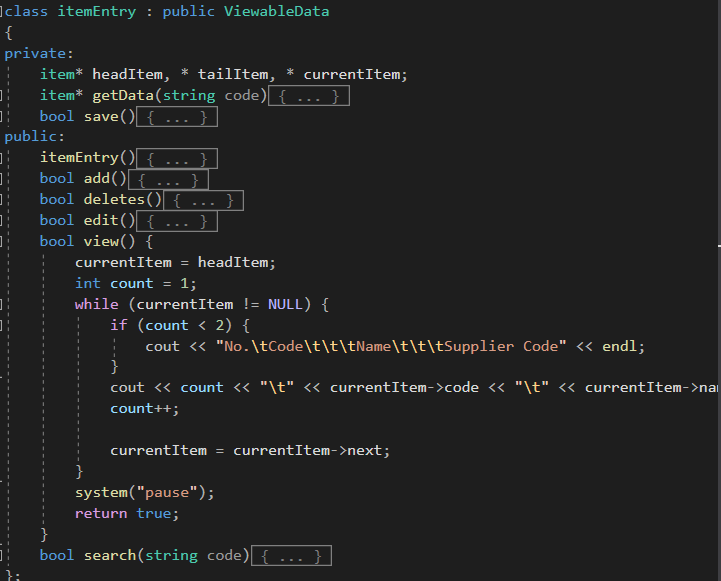
Abstraction is a concept that emphasizes on only showing things that are required and concealing irrelevant details to show the idea or purpose of the object (Mulonda, 2018). This is needed to show what an object can actually do, not to show how the object completes its functions. Another reason for this is to keep the code, neat and show what a programmer need, which is just the function. A programmer doesn’t actually need to know how the function work or how it produces the result it makes.

Figure 1.1a Abstraction of Viewing Function

## Encapsulation

Encapsulation means hiding any unnecessary or unneeded detail from a user or programmer and wrapping it to a single unit (Mulonda, 2018). The reason for this could be said to be similar to the concept of Abstraction, which is to keep the code simple enough for the programmer to use the function and not worry for messy and long programming code. This also helps to hide variables that the programmer doesn’t need to worry about as it is only used inside of the class ‘functions.

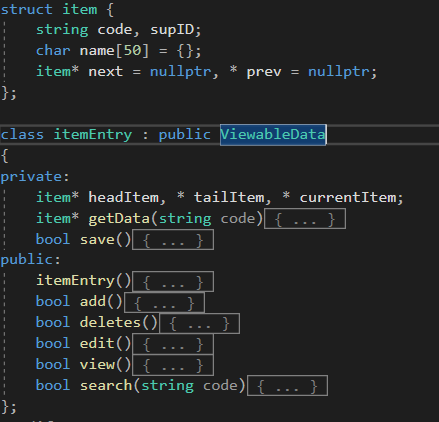


Figure 1.2a Item Entry Class Encapsulating Variables

## Inheritance

Inheritance is the concept that allows a code (variable and functions) from a class (called the parent class) to be used again in another class, called the child class (Mulonda, 2018). This helps the programmer use the same code for the same function in different classes in a program. It saves time and transform the code into a simpler and more readable code for others to contribute.

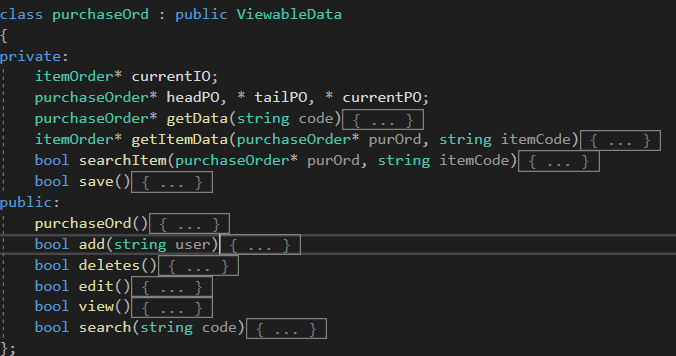


Figure 1.3a Purchase Order Class Inheriting Viewable Data Class

## Polymorphism

Polymorphism allows a user to redefine an existing function to work in a different way with the same name by changing parts of how it works, or parts used to get the function working (Mulonda, 2018). This is done by either *overloading* or *overriding*. The reason to use this is to have the same name for functions that works more or less the same way or leads to a similar result. This helps other programmer to recognize the name of the function with familiarity based on the previous classes or function without confusing them with new names of a function.

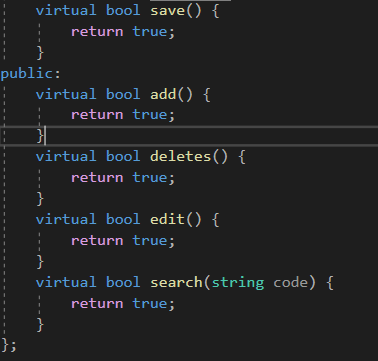


Figure 1.4a Virtual function



Figure 1.4b Item Entry Class Polymorph the Virtual function

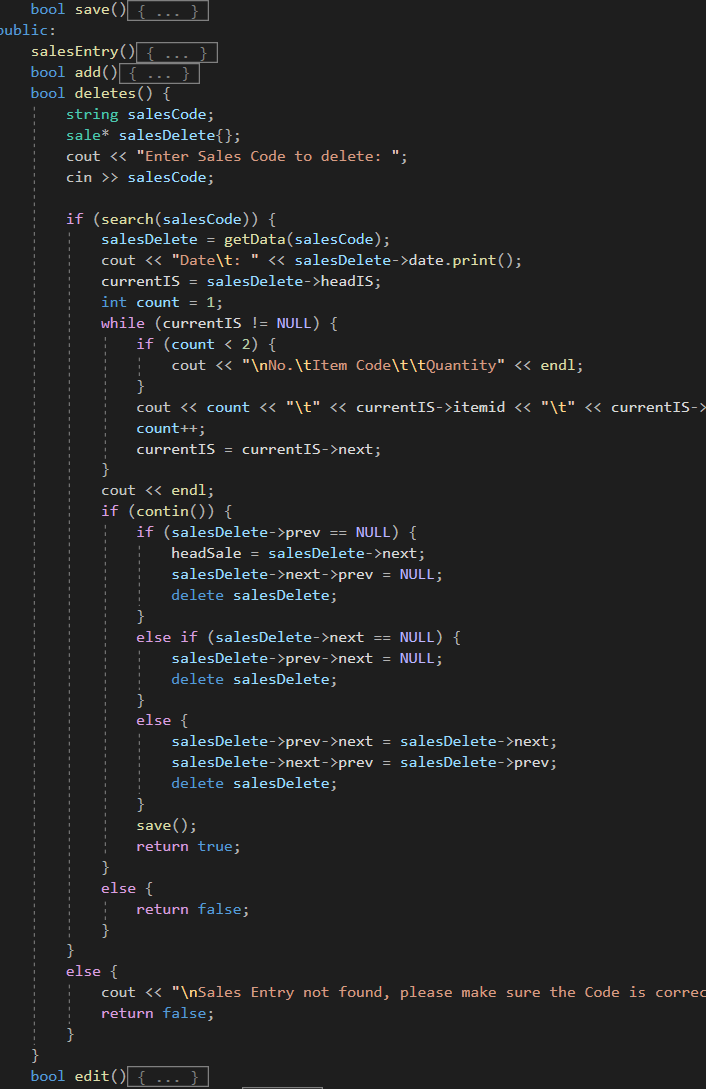


Figure 1.4b Sales Entry Class Polymorph the Virtual function

# Design

## Use Case Diagram

*A close up of a map

Description automatically generated* Figure 2.1a Use Case Diagram of The System

Figure 2.1a shows the Use Case Diagram. This diagram shows the simple relationship between use cases or the functions of the system and the actors/users (What is Use Case Diagram?, 2017). The diagram above represent how the admin can use every function in the system, while the Sales manager can only manage Entry databases (Item Entry, Supplier Entry, and Sales Entry), along with viewing the list of Purchase Requisitions. Finally, the purchase manager can access 1 database (sales entry) and view the list of entries (Item and Supplier Entries), purchase requisition, or purchase order.

## Activity Diagram

The diagrams below are called the Activity Diagrams. This UML diagram shows the flow of activity, similar to a flow chart (What is Activity Diagram?, 2017). The activity diagrams below will show the flow of each use case in the previous diagram (Figure 2.1a), each showing how a user completes a task in the use case.

### Manage User Database

A close up of text on a black background

Description automatically generated

Figure 2.2.1a Manage User Database Activity Diagram

In Figure 2.2.1a shows the activity diagram for the function, Managing the User’s Database. In the first menu, the user will be asked to pick either add, edit, or delete a user. If a user chooses Add, he or she will then be prompted to fill in all the information to create a new user. If the user selects edit, the system will show the current information and shows the user which information is editable. Afterwards, the system will ask the user to input in the new information, which then will ask a confirmation if the user wants to edit with the new info. If no, it will go back to the menu which asked the user which action he or she wants to do, if yes, it will proceed to save the new data into the database. When the user decides to the delete function, the system will ask the username and will display the information of the entered user. It will then ask a confirmation to delete this user. After the confirmation, the outcome will be the same as in the edit function.

### *Manage Purchase Order Database*

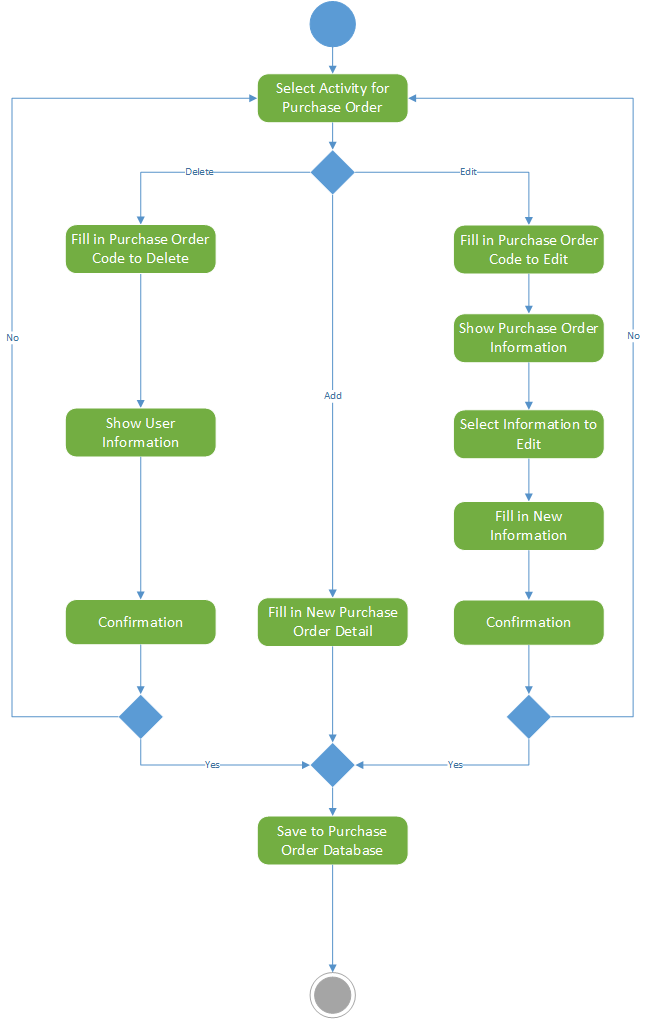


Figure 2.2.2a Manage Purchase Order Database Activity Diagram

In Figure 2.2.2a shows the function of Managing the Purchase Order’s Database. As you can see, the activity diagram is similar to Figure 2.2.1a, due to similar functions but different database. As so, the flow will be very similar with a few differences. First of all would be filling in the Purchase Order (PO) Code or the new PO details. Other difference would be all changes would be saving into the PO Database.

### A close up of a sign Description automatically generatedView Purchase Order

Figure 2.2.3a View Purchase Order Activity Diagram

The Figure 2.2.3a introduces the activity diagram on the function, Viewing Purchase Orders. This simple diagram explains that in this function, the system will get all the Purchase Orders in order to show it to the user as a list.

### A close up of a sign Description automatically generatedView Entries

Figure 2.2.4a View Entries Activity Diagram

The Figure 2.2.4a introduces the activity diagram on the function, Viewing the Entries, which is the Item Entry and the Supplier Entry only. This diagram reveals that the user will foremost select an Entry Database mentioned before, then the system will obtain all the Selected Entry for the sake of exhibiting the entries to the user on a list form.

### View Purchase Requisition

A close up of a sign

Description automatically generated

Figure 2.2.5a View Purchase Requisition Activity Diagram

The Figure 2.2.5a introduces the activity diagram on the function, Viewing Purchase Requisitions. This simple diagram only shows that in the function, the system would acquire all the Purchase Requisitions to show it to the user in a form of a list.

### Manage Purchase Requisition Database A close up of a map Description automatically generated

Figure 2.2.6a Manage Purchase Requisition Database Activity Diagram

In Figure 2.2.6a reveals the function of Managing the Purchase Requisition’s Database. As the figure shows, the activity diagram is identical to Figure 2.2.1a, due to identical functions but different database. As so, the flow of the function will be mostly identical with a few differences. The first would be filling in the Requisition (PR) Code or the new PR information. Lastly would be all changes would be saving into the PR Database.

### *Manage Entries Database*

A close up of text on a black background

Description automatically generated

Figure 2.2.7a Manage Entries Database Activity Diagram

In Figure 2.2.7a exhibits the function of Managing the Manage Entries’ Database. As the figure shows, the activity diagram is alike to Figure 2.2.1a, due to identical functions but different databases. As so, the flow of the function will be mostly alike with a few differences. Foremost is the selection of the database before the selection of action to the selected database. Secondly is the input of the Selected Entry’s Code or the new Selected Entry information. Finally, it is that all adjustments would be saving into the Selected Entry’s Database.

# Screenshot of The Program’s Output

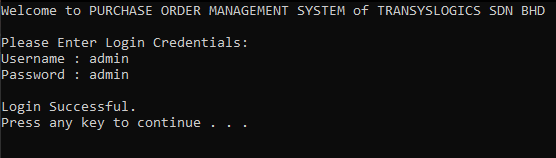


Figure 3a. Login Page

In Figure 3a, the user logins with his or her username and password. If the login succeeds in logging in the system will shows the message “Login Successful”, and if it fails, it will prompt the user the “username and/or password is wrong” message and the user will have to type the username and password again.

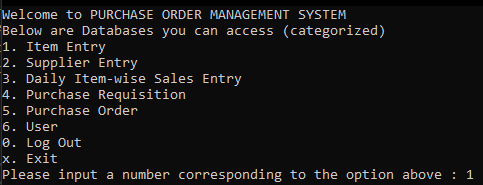


Figure 3b. Main Menu for Admin

In Figure 3b, the user was identified as an admin. The user is shown what Databases he or she can access and prompts the user to choose the database to access, Log Out, or Exit the program.

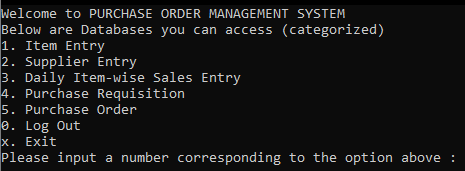


Figure 3c. Main Menu for Sales and Purchase Manager

In Figure 3c, the user is identified to be a Purchase Manager or Sales Manager. Same as Figure 3b, the user is shown what databases that the user can access, as well as the option to Log out or exit the program.

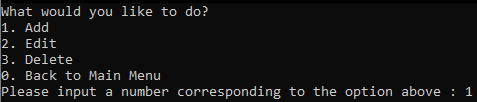


Figure 3d. Database Menu for Manage Only

In Figure 3d, actions to take in the selected database after the main menu. If the user is only able to manage it. The user is able to act according to the option shown in Figure 3d. The system will of course ask the user to choose an action. The user can also go back to the main menu.

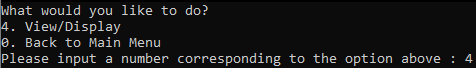


Figure 3e. Database Menu for Viewable Only

In Figure 3e, the only action a user can take if a user is only allowed to view, is of course viewing the database. The user will have to pick the action of either view the selected database or going back to the main menu.

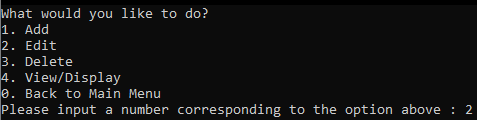


Figure 3f. Database Menu for All Access

The Figure 3f shows what a user that has the access to manage and view a database can do to the data in the selected database or go back to the main menu to choose another database again.

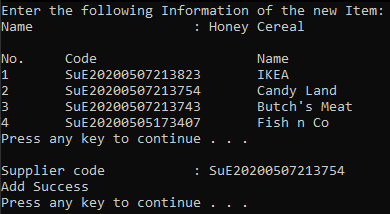


Figure 3g. Add Item Entry Page

In Figure 3g, it shows the menu after the user picks Item Entry as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new Item Entry. When filling in the Supplier Code, the system will show a list of suppliers to let the user know the option. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

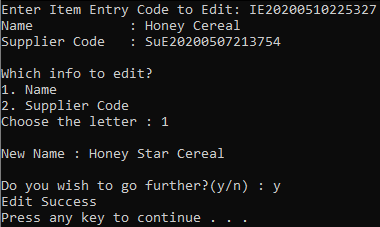


Figure 3h. Edit Item Entry Page

Figure 3h will show after the user picks Item Entry as the database to access and Edit as the action. The user will be prompted to type in the Item Entry Code, shown the full item information and choose an information to edit, then will be asked to fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

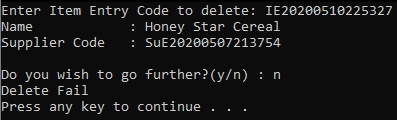


Figure 3i. Delete Item Entry Page

Figure 3h will show after the user picks Item Entry as the database to access and Delete as the action. The user will be prompted to type in the Item Entry Code to delete and the system will show the info of the item, If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

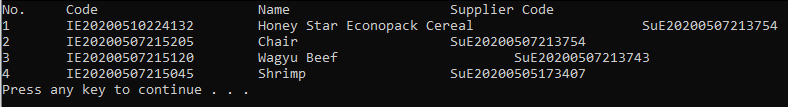


Figure 3j. View Item Entry Page

In Figure 3j, It shows the result when the user picks Item Entry as the database to access and View as the action. This will show all the Item with its information in a list

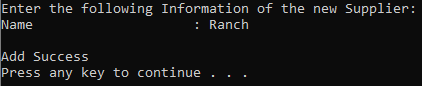


Figure 3k. Add Supplier Entry Page

In Figure 3k, it shows the menu after the user picks Supplier Entry as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new Supplier Entry. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

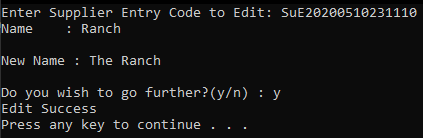


Figure 3l. Edit Supplier Entry Page

Figure 3l will show after the user picks Supplier Entry as the database to access and Edit as the action. The user will be prompted to type in the Supplier Entry Code and show the current full information of the supplier, then will be asked to fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

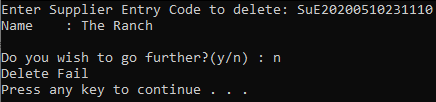


Figure 3m. Delete Supplier Entry Page

Figure 3h will show after the user picks Supplier Entry as the database to access and Delete as the action. The user will be prompted to type in the Supplier Entry Code to delete and the system will show the info of the supplier, If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

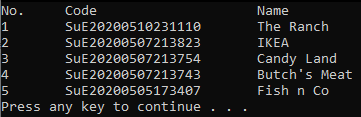


Figure 3n. View Supplier Entry Page

In Figure 3n, It shows the result when the user picks Supplier Entry as the database to access and View as the action. This will show all the Supplier with its information in a list.

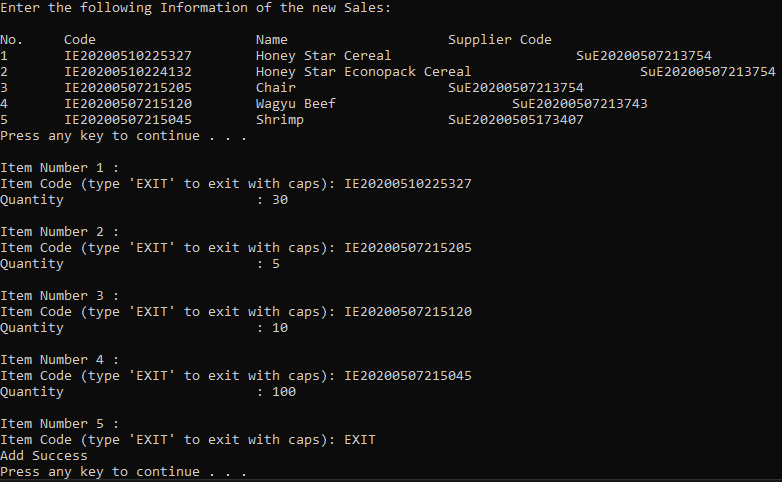


Figure 3o. Add Sales Entry Page

In Figure 3o, it shows the menu after the user picks Sales Entry as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new Sales Entry. When filling in the Sales Code, the system will show a list of items to let the user know the option. It will then proceed to ask the user to fill the item code as well as the quantity. It will continue to ask the user to fill the new items until the user type ‘EXIT’. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

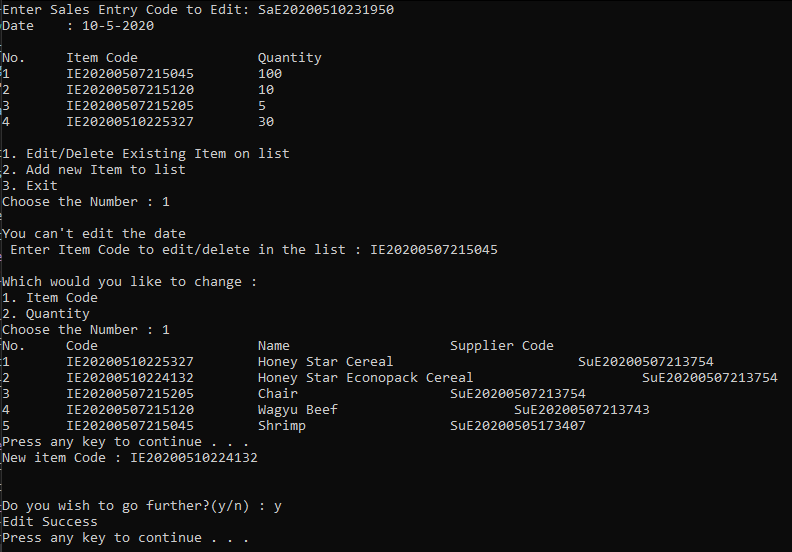


Figure 3p. Edit Sales Entry Page

Figure 3p will show after the user picks Sales Entry as the database to access and Edit as the action. The user will be prompted to type in the Sales Entry Code and show the current full information of the Sales, then will be asked to fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

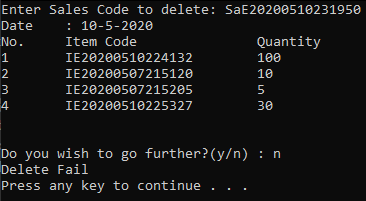


Figure 3q. Delete Sales Entry Page

Figure 3q will show after the user picks Sales Entry as the database to access and Delete as the action. The user will be prompted to type in the Sales Entry Code to delete and the system will show the info of the sales, If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

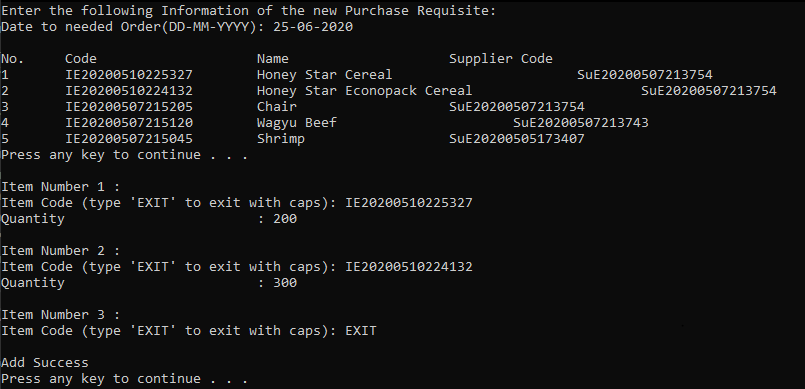


Figure 3r. Add Purchase Requisition Page

In Figure 3r, it shows the menu after the user picks Purchase Requisition as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new Purchase Requisition. When filling in the Purchase Requisition, the system will show a list of items to let the user know the option. It will then proceed to ask the user to fill the item code as well as the quantity. It will continue to ask the user to fill the new items until the user type ‘EXIT’. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

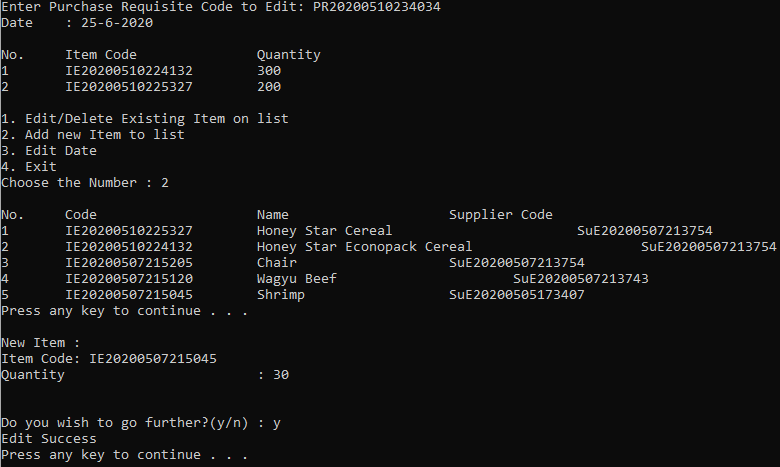


Figure 3s. Edit Purchase Requisition Page

Figure 3s will show after the user picks Purchase Requisition as the database to access and Edit as the action. The user will be prompted to type in the Purchase Requisition Code and show the current full information of the Purchase Requisition, then will be asked to fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

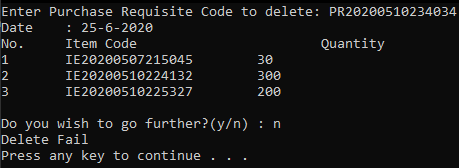


Figure 3t. Delete Purchase Requisition Page

Figure 3t will show after the user picks Purchase Requisition as the database to access and Delete as the action. The user will be prompted to type in the Purchase Requisition Code to delete and the system will show the info of the purchase requisition, If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

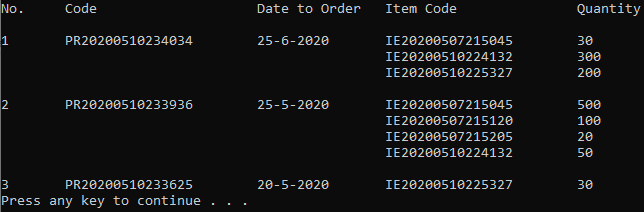


Figure 3u. View Purchase Requisition Page

In Figure 3u, It shows the result when the user picks Purchase Requisition as the database to access and View as the action. This will show all the Purchase Requisition with its information in a list.

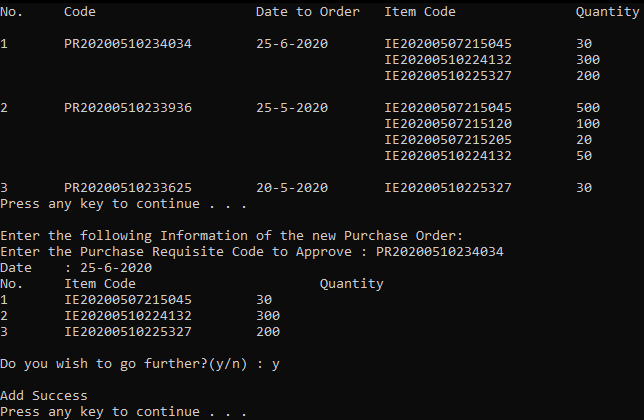


Figure 3v. Add Purchase Order Page

In Figure 3v, it shows the menu after the user picks Purchase Order as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new Purchase Order. When filling in the Purchase Requisition Code, the system will show a list of Purchase Requisition to let the user know the option. It will then proceed to show the user what items are included in the Purchase Requisition. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

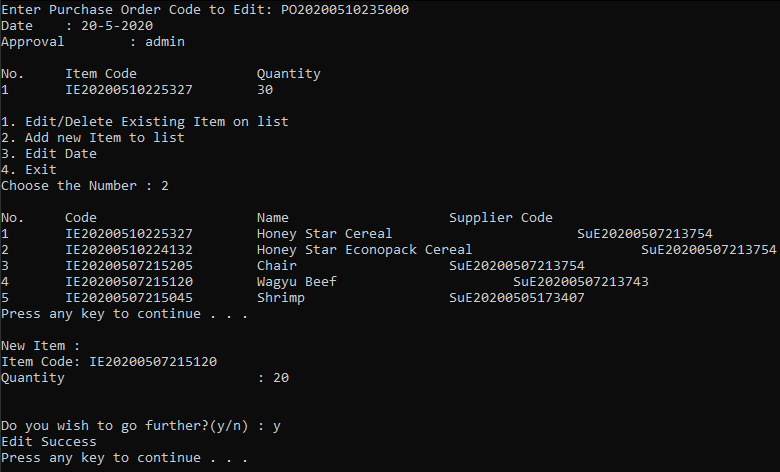


Figure 3w. Edit Purchase Order Page

Figure 3w will show after the user picks Purchase Order as the database to access and Edit as the action. The user will be prompted to type in the Purchase Order Code and show the current full information of the Purchase Order. The system will then ask the user which information to edit, then the user will be asked to fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

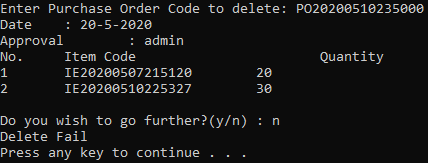


Figure 3x. Delete Purchase Order Page

Figure 3x will show after the user picks Purchase Order as the database to access and Delete as the action. The user will be prompted to type in the Purchase Order Code to delete and the system will show the info of the purchase order. If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

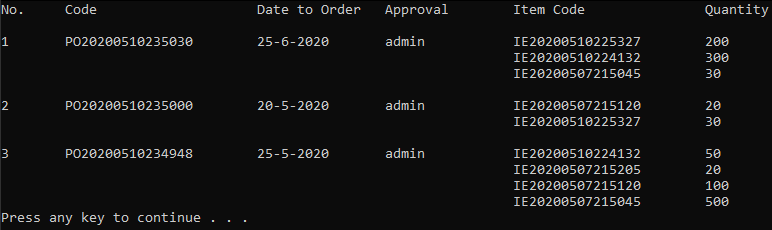


Figure 3y. View Purchase Order Page

In Figure 3y, It shows the result when the user picks Purchase Order as the database to access and View as the action. This will show all the Purchase Order with its information in a list.

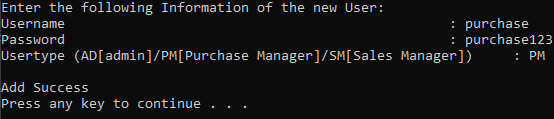


Figure 3z. Add User Page

In Figure 3z, it shows the menu after the user picks User as the database to access and Add as the action. The user will be prompted to fill in all the information to create a new User. If the information filled is correct, the system will show “Add Success”, if not, it’ll show “Add Fail”.

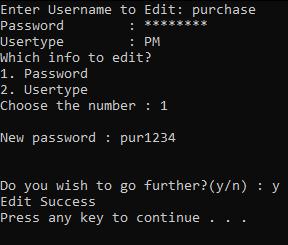


Figure 3A. Edit User Page

Figure 3A will show after the user picks User as the database to access and Edit as the action. The user will be prompted to type in the Username and show the current full information of the User. The system will then ask the user which information to edit, the user will fill in the new information. If the new information filled is correct and confirms it the new information, the system will show “Edit Success”, if not, it’ll show “Edit Fail”.

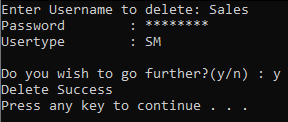


Figure 3B. Delete User Page

Figure 3B will show after the user picks User as the database to access and Delete as the action. The user will be prompted to type in the Username to delete and the system will show the info of the user. If the user confirms it, the system will show “Delete Success”, if not, it’ll show “Delete Fail”.

# Limitation

Although the system works as it is intended to do but there are some things that limits the capability of the system for further easier and simpler Usability for the user. For example, the Graphical User Interface the current system is just a Command Line Interface that is not user friendly nor easy to use. The GUI can only accept one input at a time. If a user makes a mistake, the user can't go back and fix it, which means he or she will have to start over to fix that mistake. Another point that is that if a user accidently presses command shortcut (ctrl + c, ctrl + b, etc.) in the program, it will crash. Thus, making this system not user-friendly.

Another example of the limitations of the system is how the user can't pick a choice with something similar to a drop box text. Because of this, the user must remember the code or ID of an entry in order to delete, edit, or search for it. A limitation that is connected to the one before this is that the user can't click the choices on the main menu using a mouse. Instead the user will have to choose by typing in the designated letter or number shown on the menu.

The program can also receive an empty input and record it without checking it. This is in most part my fault, as I didn’t program the system to not receive any empty inputs. But you can still delete the records if the mistake was made.

# Conclusion

In conclusion to this project, this has helped me further understand the concepts inside of object-oriented programming and further expand my knowledge on the programming language known as C++. Although this project seems simple at first, under journey it appears to show that it is more complex than it appears to be. This has taught me to never underestimate any projects given to me in the future and always understand a project before doing it beforehand. Although I have a limited knowledge on C++ and how to better the graphical interface, this project still turns out a good experience for me Anne my fellow classmates.

# References

Mulonda, Y., 2018. *Object-Oriented Programming Concepts “In Simple English”*. [online] Medium. Available at: <https://blog.usejournal.com/object-oriented-programming-concepts-in-simple-english-3db22065d7d0> [Accessed 9 May 2020].

Prabhu, R., 2019. *Object Oriented Programming (Oops) Concept In Java - Geeksforgeeks*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/> [Accessed 9 May 2020].

Visual paradigm. 2017. *What Is Activity Diagram?*. [online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/> [Accessed 9 May 2020].

Visual paradigm. 2017. *What Is Use Case Diagram?*. [online] Available at: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> [Accessed 9 May 2020].