**ONLINE SHOPPING MANAGEMENT SYSTEM**

BY

LOK WERN KAI

A RIGOROUS ASSESSMENT

SUBMITTED TO

Asia Pacific University

JULY 2022

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **No.** | **Content** | **Page** |
| 1 | Solution Design | 3 |
| 2 | C++ Programming Concept | 4 – 8 |
| 3 | Object-Oriented Concept | 9 - 13 |
| 4 | Program Screenshot | 14 - 37 |
| 5 | Limitation | 38 |
| 6 | Conclusion | 38 |

**SOLUTION DESIGN**

Diagram

Description automatically generated

Figure System Architecture Flow

The objective of this project is to develop an Online Shopping Management System for both business and customer user using C++ Object-Oriented Programming (OOP) concepts. The system should allow business user to manage user, inventory, and order. On the other side, customer user should be able to browse product, place order and check bill via the system.

Since the main objective is to focus on application layer with OOP concepts, a simple console application with text file serve as data storage can meet the requirement and ease the development process. Console application can reduce the time on User Interface (UI) design and text file as data storage can avoid the time on setting up a Relational Database Management System (RDBMS). Hence, developer will have more time focus on application layer.

The console application will retrieve instruction from the user then save the input into the relative text file as data storage for future purpose. The user will be able to view the data stored in the text file via the application as well.

**C++ PROGRAMMING CONCEPT**

1. **Variable**

Variable is one of the basic concepts in C++ programming. It is used to store and hold data values. In the application, variable such as username, password, file was declared to store the data for later use.

Text

Description automatically generated

1. **Function**

A function is a block of code that are ready to be re-use. In C++, function can be used to perform certain action. Function such as toLower(), removeNewLine() and getNewOrderId were defined to fulfill the needs to convert string value to lower case, remove new line in the data and get new order id for new order.

Text

Description automatically generated

1. **Control Statement**

Control statement in C++ is common use to redirect the flow of the program. It is heavily used in the application to deal with different condition and scenario.

* 1. **Conditional Statement**
     1. **If … else / else if … Statement**

Below screenshot shows an example of a nested If Else statement which return different string value with different condition such as is the product exists in database and is the class method successfully delete the product.

Text

Description automatically generated

* + 1. **Switch Statement**

Switch statement is another common conditional statement to deal with multiple condition. In the application, it is used to route user to the correct menu based on the user permission.

Text

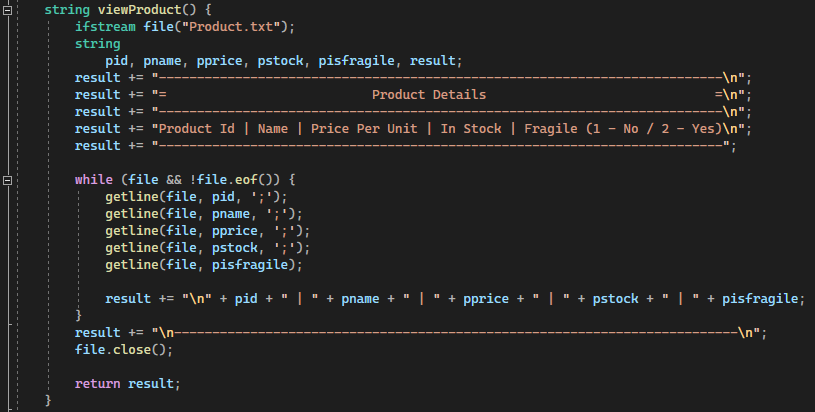
Description automatically generated

* 1. **Loops**

In C++, looping is one of the control statements that will execute the code until it meets the defined condition.

* + 1. **While Loop**

A while loop in C++ will check the condition before it execute the code. Below screenshot is an example of while loop, it will retrieve the data in the text file line by line until it reaches the end of the file.



* + 1. **For Loop**

In C++, for loop can take up to 3 statement to execute the code. Below screenshot shows a for loop statement that will loop every single character in a string data, convert it to lower case and append the converted character to a string data.

Text

Description automatically generated

* + 1. **Do…While Loop**

Do While loop is a reverse version of while loop. Do while loop will execute the block code for the first time then loop in if the condition is met. In the application, it was used to repeat menu screen to user until user opt to exit the application.

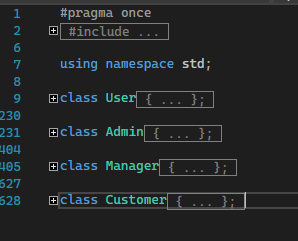
Text

Description automatically generated

**OBJECT ORIENTED CONCEPT**

1. **Class**

A Class in C++ programming is the main aspect of OOP. It allows the developer to follow the “Don’t Repeat Yourself” principle during development. In below screenshot, multiple class such as User, Admin, Manager and Customer were defined to create different object in the application.



1. **Constructor**

A Constructor is a special class method that will be execute when a new object is created.

In the application, the Order class have different constructor to handle different scenario when the application creates an Order object.

Text

Description automatically generated

1. **Encapsulation**

Encapsulation in OOP is to ensure that data are hidden from the users and can only access by the class itself. From below screenshot, variable such as password, isExit, permission and username fall under private access specifier. User only allow to access the data via in-class get and set method.

Text

Description automatically generated

1. **Inheritance**

Inheritance in OOP is to allow developer defined sub-classes from a base class, so that the method and variable in the based class can be re-use by the sub-classes. From below screenshot, Admin class is a sub-class of User class. When an Admin object require to list out customer, it can call the viewCustomer() method without create another User object.

Text

Description automatically generated

1. **Polymorphism**

Polymorphism allow OOP in C++ to have method with same name but work in different way. From below screenshot, there are 3 different Order method which behave differently and overload each other. It was designed to fit different use case in the application.

Text

Description automatically generated

1. **Abstraction**

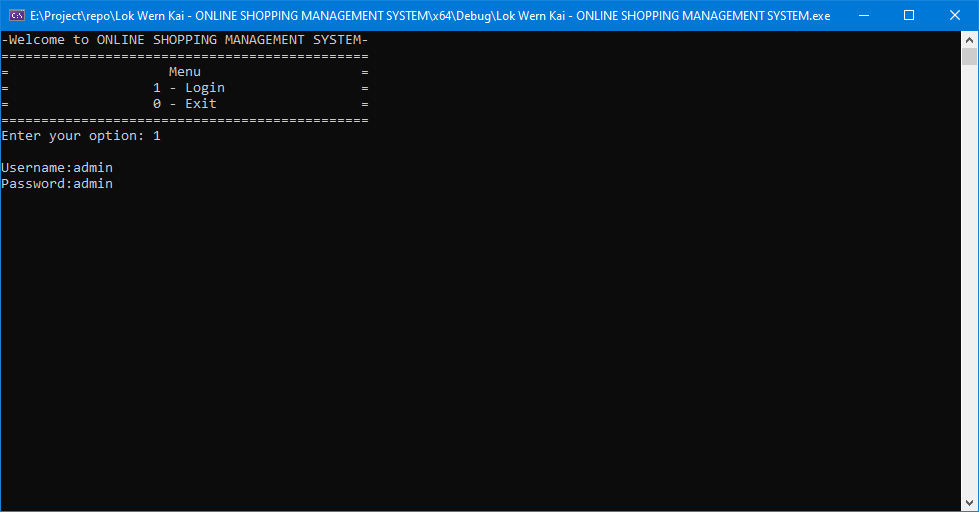
Encapsulation in OOP is to ensure that data are hidden from the users and not accessible outside from the class. From below screenshot, variable such as password, isExit, permission, username and delCart() method falls under private access specifier which user can’t access the data from the outside. However, public method such as Cart are callable when the user creates a Cart object.

Text

Description automatically generated

**PROGRAM SCREENSHOT**

**Main Menu**



Above screen shot is the main menu of the application.

To login, user will need to insert “1” and press enter. The username and password will prompt for user input. If the user does exist in the system, it will route the user to correct menu, else the user will not be able to login.

The main menu screen will remain until user insert “0” then the application will shut down.

**Admin**

1. **Admin Menu**

Text

Description automatically generated

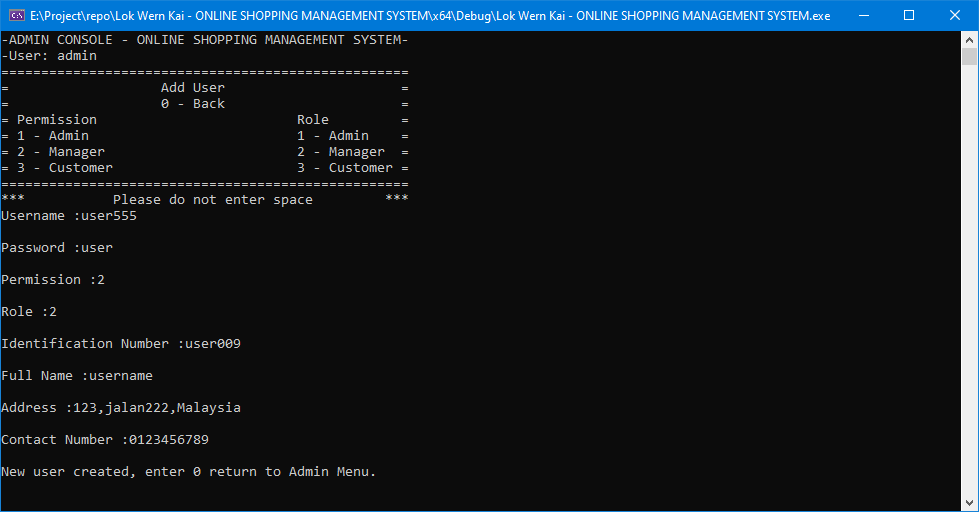
Above screenshot is the admin menu.

Only user with admin permission will be able to see the screen.

It provides option for admin to manage the user of the system by using the functionality such as Add User, Edit / View / Search User and Delete User.

If the user opts to logout from the admin account, insert “0” and the application will bring the user back to the main menu.

1. **Create User**



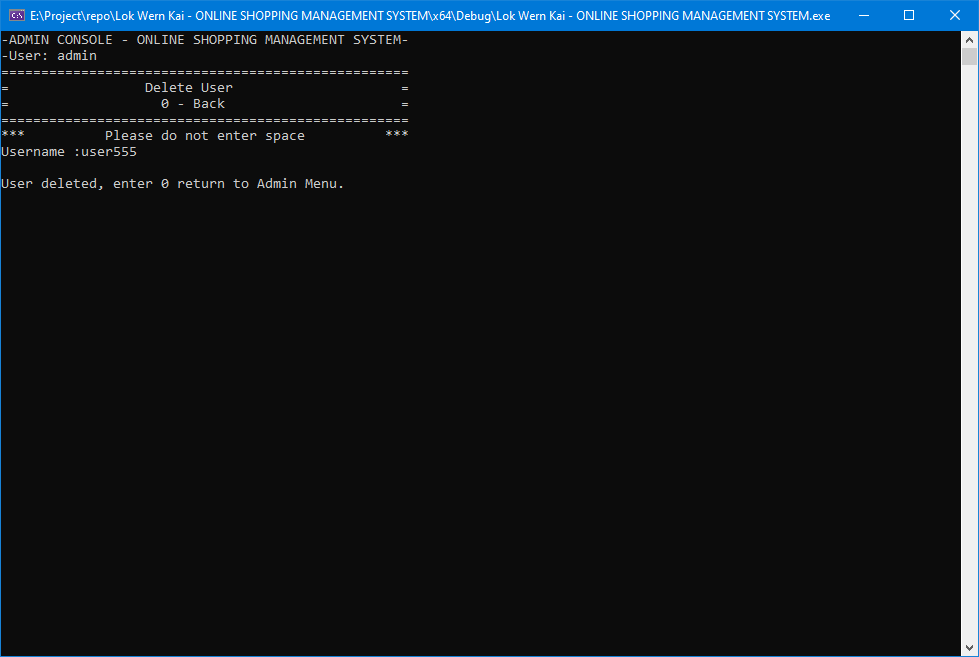
Above screenshot is the screen for admin to create new user.

Admin user will require to insert new user details such as username, password, user permission, user role, user identification number, etc.

These details will then be stored in the User.txt.

So that, the newly created user will be able to login the application with the username and password.

1. **Delete User**



Above screenshot is for the delete user function.

It is a simple screen; admin user is required to insert the username of the user in order to remove the user access.

Once the operation complete, the user details will be removed from the User.txt.

Hence, the user will no longer have the access to the application.

1. **View User**

Text

Description automatically generated

Above screenshot is the view user function.

Once the admin user enters the view option, all user details stored in User.txt will be retrieve and display in the application.

1. **Edit User**

Text

Description automatically generated

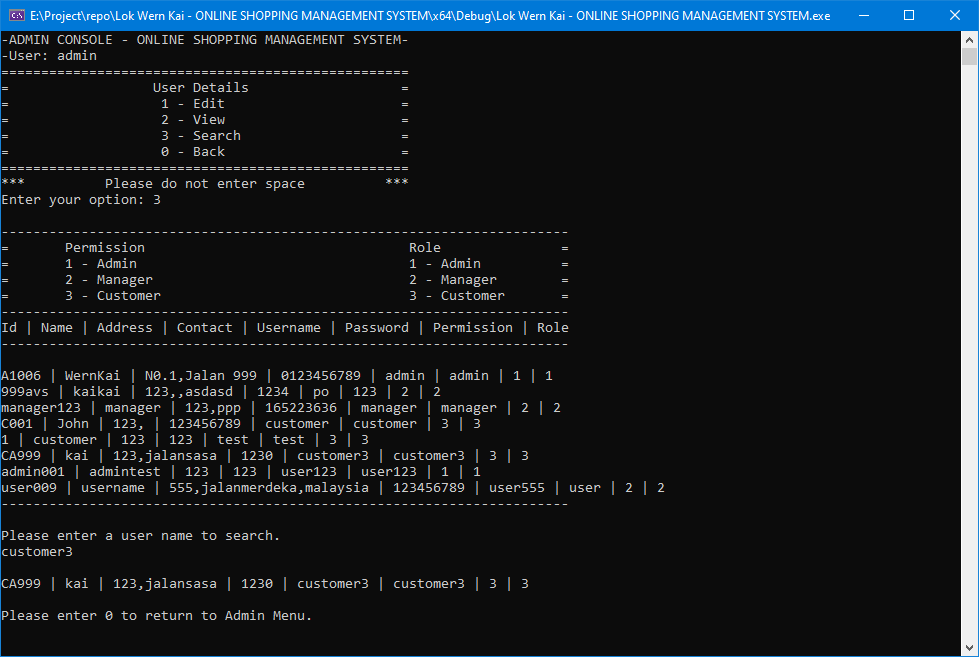
In this screen, admin user is required to enter the usernames that are going to be edit.

The application will show the user’s current details from User.txt.

Then multiple fields will prompt for admin user to update the user details.

Once the admin user completes the operation, the updated details of the user will prompt from application as a verification.

1. **Search User**



Search user screen which allows admin user to search for user.

Admin user will have to insert the username to search.

If the user does exist in the application, the details will display on the application.

**Manager**

**Manager Menu**

Text

Description automatically generated

Above screenshot is the manager menu.

Only user with manager permission will be able to see the screen.

It provides option for manager to manage the product, order or order item of the system by using the functionality such as Add, Edit / View / Search and Delete.

If the manager opts to logout from the manager account, insert “0” and the application will bring the user back to the main menu.

1. **Create Product**

Text

Description automatically generated

Above screenshot is the screen for manager to create new product.

Manager user will require to insert new product details such as product id, product name, price, quantity, fragile, etc.

These details will then be stored in the Product.txt.

So that, the newly created product will be available in the shopping system.

1. **Delete Product**

Text

Description automatically generated

Above screenshot is for the delete product function.

It is a simple screen; manager user is required to insert the product id of the product in order to remove the product.

Once the operation complete, the product details will be removed from the Product.txt.

Hence, the product will no longer available in the system.

1. **View Product**

Text

Description automatically generated

Above screenshot is the view product function.

Once the manager user enters the view option, all user details stored in Product.txt will be retrieve and display in the application.

1. **Edit Product**

Text

Description automatically generated

In this screen, manager user is required to enter the product name that are going to be edit.

The application will show the product’s current details from Product.txt.

Then multiple fields will prompt for manager user to update the product details.

Once the manager user completes the operation, the updated details of the product will prompt from application as a verification.

1. **Search Product**

Text

Description automatically generated

Search product screen which allows manager user to search for product.

Manager user will have to insert the product name to search.

If the product does exist in the application, the details will display on the application.

1. **Create Order**

Text

Description automatically generated

Above screenshot is the screen for manager to create new order.

Manager user will require to insert new order details such as order id and customer id.

These details will then be stored in the Order.txt.

So that, the newly created order will be ready to add order item.

1. **Delete Order**

Text

Description automatically generated

Above screenshot is for the delete order function.

It is a simple screen; manager user is required to insert the order id of the order to remove the order.

Once the operation complete, the order details will be removed from the Order.txt.

Hence, the order will no longer available in the system and the associated order item will be remove as well.

1. **View Order**

Text

Description automatically generated

Above screenshot is the view order function.

Once the manager user enters the view option, all order details stored in Order.txt will be retrieve and display in the application.

1. **Search Order**

Text

Description automatically generated

Search order screen which allows manager user to search for order.

Manager user will have to insert the order id to search.

If the order does exist in the application, the details will display on the application.

1. **Create Order Item**

Text

Description automatically generated

Above screenshot is the screen for manager to create new order item.

Manager user will require to insert new order item details such as order id, product id, price, and quantity.

These details will then be stored in the OrderItem.txt.

1. **Delete Order Item**

Text

Description automatically generated

Above screenshot is for the delete order item function.

It is a simple screen; manager user is required to insert the order id and product id to remove the order item.

Once the operation complete, the order item details will be removed from the OrderItem.txt.

Hence, the order item will no longer available in the system.

1. **View Order Item**

Text

Description automatically generated

Above screenshot is the view order item function.

Once the manager user enters the view option, all order item details stored in OrderItem.txt will be retrieve and display in the application.

1. **Edit Order Item**

Text

Description automatically generated

In this screen, manager user is required to enter the order id and product id that are going to be edit.

The application will show the order item’s current details from OrderItem.txt.

Then multiple fields will prompt for manager user to update the order item details.

Once the manager user completes the operation, the updated details of the order item will prompt from application as a verification.

1. **Search Order Item**

Text

Description automatically generated

Search order item screen which allows manager user to search for order item.

Manager user will have to insert the order id to search.

If the order does exist in the application, the details will display on the application.

**Customer**

Text

Description automatically generated

Above screenshot is the customer menu.

Only user with customer permission will be able to see the screen.

It provides option for customer to search for available product, add product to cart and view the bill.

If the customer opts to logout from the customer account, insert “0” and the application will bring the user back to the main menu.

1. **Search Product**

Text

Description automatically generated

Search product screen which allows customer user to search for product.

Customer user will have to insert the product name to search.

If the product does exist in the application, the details will display on the application.

This function share same Search Product functionality from Manager user.

1. **Add Product to Cart**

Text

Description automatically generated

Above screenshot is the screen that allow customer user to add product into cart.

Customer are allowed to add multiple but not duplicated product, if customer opt to stop adding product, insert “0” and the application will bring the user back to customer menu. These details will be stored in the Cart.txt

1. **View Bill**

Text

Description automatically generated

View bill screen is the screen that display the customer bills.

After customer finish adding product into cart, once the user exits the Cart function, a bill will be created and store the details in a file with “Bill-<username>.txt” naming format. The bill will show the total amount of each order including the packing fees.

**LIMITATION**

Beside of the functionality of the application, limitation comes together.

Below are some limitations with the possible solution to make the application more advance:

1. Implement more data verification such as address validation, phone number format checker to avoid potential data format issue.
2. In the current system, there’s no encryption feature for user password. Encryption such as AES should be implemented to protect user data.
3. Currently, each user bind to a specific permission to access the application. Once the application getting more usage and function, granting user with permission is not sufficient to manager user access. A role for each user should be introduce, so that we can group user in a specific role that share similar permission.
4. With the current design, customer is the only user that allow to generate bill. Even though manager is allowed to add order and order item, however, bill will not be generated. This feature should be added in order to make the application more complete.
5. Console application can only serve application with simple usage. Once more and more function come in, a better Graphic User Interface (GUI) should be introduced to ease the way user interact with the application. GUI library for C++ such Gtkmm or Qt will be a potential solution.
6. Limitation of data storage using text file is also one of the limitations, a RDBMS should be introduced and integrate with the application to provide better data storage and data security.

**CONCLUSION**

As a conclusion, a simple C++ console application was developed as an online shopping system with C++ programming and object-oriented concept. The application can fulfill the basic function as requested in the requirement. However, in application development, there’s no ending in development. More function and features should be considered and implement into the application to make it more complete.