Programming Concept & Design: White Market System

Yaashiniy A/P M Segaran

BIT20230901011

Faculty Business Management and Information Technology

Bachelor of Information Technology

Madam Hanisah

November 20, 2023

Table of Contents

Table of Figures

1.0 Introduction

This report explores a unique online marketplace called the White Market. This online market is not our average marketplace, it sells a variety of peculiar goods like uncommon medicines and unique self-defence equipment that are hard to fine in our country at affordable price. The report takes closer at the C++ codes that used to develop the system along with its functionality and it makes use of various functions and instructions to make it user-friendly. The White Market code's primary feature is a carefully designed system that askes users to choose categories, making it easier to buy prescriptions medicines and even strange goods like tobacco. Other than that, the code uses a menu-driven method with switch case to manage user input and move through different sections of the market.

The code uses well-defined functions to do specific jobs within the shopping experience. For example, there are functions like processDefenseToolsOrder and processMedicineOrder that handle things like choosing products, specifying quantity and calculating the total price. The use of functions makes the code clear and enable it to reuse for various tasks. Then, various control structures have been used to manage the flow of the program. For example, if-else statement that checks whether the variable 'age' is greater than or equal to 21 and executes different code blocks accordingly. Moreover, couple of arrays have been used in this system such as array of strings that declares strings named 'address' and 'name' with a size of 50. It is used to store addresses and names entered by the users. Next, the 'getline' function is used to read a line of input from the user because it is used to capture the full name of the customer with space and for capturing entire lines of text.

In conclusion, a detailed analysis of these elements will be provided in this report, providing information on the ideas behind the design decisions, the effectiveness of the functions, control structures and calculations that have been used in this program.

2.0 Objective

1. Automated Ordering System:

- **Objective:** Develop an automated ordering system for the White Market that allows customers to easily browse, select, and purchase items such as medicines and self-defense tools.

2. Order Processing and Invoicing:

- **Objective:** Develop order processing system that calculates costs, including item costs, shipping and generates detailed invoices for customers.

3. Customer Information Management:

- **Objective:** Implement a system to collect and manage customer information securely, including names, contact numbers, email addresses, and shipping details.

4. User Interface Enhancements:

- **Objective:** Enhance the user interface of the White Market system for a more visually appealing and user-friendly experience.

3.0 Price Tables

Category: Medicines

Name of Medicines	Prices
Cimicifuga racemosa	RM 67
Hydrastis canadensis	RM 50
Santalum ellipticum	RM 120

Table 1: Name of medicines and prices

Category: Self-Defence Tools

Name of Medicines	Prices
TASER Pulse+	RM 120
Hot Shot Tactical Axe	RM 67
Reapr Meridus Full Tang Sword	RM 146.67

Table 2: Type of self-defence tools and prices

Category: Tobacco/Cigarettes from Treasurer London

Name	Prices
Treasurer Luxury Black	RM 498/pack
Treasurer London Aluminium Black	RM 563.81
Porsche Design Jet Lighter P'3642	RM 850

Table 3: Items from Treasurer London and prices

4.0 Libraries

These libraries pre-define functions and classes for common tasks, such as handling files, working with strings, interacting with the operating system and provide any other special features.

1. **#include < iostream>**

   This library is used for input and output operations (cout and cin). It includes functionality for reading from the standard input (keyboard) and writing to the standard output (console)

2. **#include <fstream>**

   This library is used for file handling operations that allow us to from and write to files. In White Market program, it is utilized to write customer information to file named “White market customers.txt.” providing a record of transactions and allowing us to retrieve information about previous customers.

```
// file "white market customers.txt"
ofstream out("C:\\Users\\Yaashiniy Segaran\\Desktop\\White market customers.txt", ios::app);

out << "Name: " << name << endl;
out << "Contact Number: " << contactNum << endl;
out << "Email: " << email << endl;
out << "Address: " << address << endl;
out << "------------------------\n";
out.close();  // Close the file when done
```

Figure 1: fstream

3. **#include <iomanip>**

   Provides facilities to manipulate the format of input and output. It is often used for setting precision in floating-point numbers.

```
cout<< setprecision (2) << fixed ;
```

Figure 2: iomanip

4. **#include <string.h>**

   Used for handling string data types such as the customer's name,email and address.

```
string name[50], email, address[50] ;
int age, contactNum, category;
double medicineCost, defenseToolsCost, treasurerLondonCost,shipping, delivery;
int typeofmed, typeofTools, typeofItem, quantity;
```

Figure 3: String.h

5. **#include <windows.h>**

   Includes functions for interacting with the Windows operating system and this code, it is used for the 'Sleep' function to introduce delays. Other than that, it also provides functions and classes for common tasks such as handling files and working with the strings. System("cls") also uses windows.h to clear the console screen. This command is used to enhance the user interface by creating a cleaner and more organized appearance during the execution.

```
ManageMenu() {
system("color F0"); // yellow terminal colour
cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t                    W H I T E  M A R K E T                    " << endl;
cout << "\n\n\n\n\n\n\n\n";
Sleep(5000);
system("CLS");
menu(); // call to main function to load after executing the constructor
}
```

Figure 4: windows.h

5.0 Variables

string **name [50], email, address[50]** ;

int age, **contactNum, category**;

double **medicineCost, defenseToolsCost, treasurerLondonCost,shipping, delivery**;

int **typeofmed, typeofTools, typeofItem,quantity** ;

6.0 Comments

Provide explanation and context for different sections.

```
cout << "Enter your Full Name  : ";
cin.ignore(); // Ignore any leftover newline characters in the buffer
getline(cin, name);
cout << " " << endl;
```

```
//  defense tools order processing code...
// Function to process defense tools order
void processDefenseToolsOrder(double& delivery, double& defenseToolsCost)
{
    string name, email, address, payment;
    int typeofTools, quantity,shipping, age, contactNum;
```

Figure 5: Comments

7.0 Indentation

Indentation refers to the spaces at the beginning of a code line. Below picture is one of the example of how indentation is applied in my code.

```
int purchaseMedicine;

if(typeofmed == 1) {

    medicineCost = quantity * 67.30;
    delivery = medicineCost + shipping ;

    cout << " " << endl;
    cout << " Your purchase will cost RM " << delivery << " including delivery cost." << endl;
    cout << " " << endl ;

    Sleep(3500);
    }

if(typeofmed == 2) {

    medicineCost = quantity * 50.80;
    delivery = medicineCost + shipping;

    cout << " " << endl;
    cout << " Your purchase will cost RM " << delivery << " including delivery cost." << endl;
    cout << " " << endl ;

    Sleep(3500);

}

 if(typeofmed == 3) {

    medicineCost = quantity * 120;
    delivery = medicineCost + shipping;

    cout << " " << endl;
    cout << " Your purchase will cost RM " << delivery << " including delivery cost." << endl;
    cout << " " << endl ;

    Sleep(3500);
}
```

Figure 6: Indentation

8.0 Loop

White Market code includes a do-while loop which is a type of loop. The do-while loop in this program is responsible for displaying the categories and processing the user's input. This loop structure continues to execute the block of code inside it if the condition specified by while(true) is true. Other than that, the loop prompts the user to input a category, processes the chosen category and askes if the user wants to add more items. The loop repeats until the user decides not to add more items.

```
    // Display categories
    do
{

    displayCategories();

    cout << " " << endl;
    cout << "Enter the category (1/2/3): ";
    cin >> category;

    system("cls");

    switch (category)
    {
    case 1:
        processMedicineOrder(delivery, medicineCost);
        break;

    case 2:
        processDefenseToolsOrder(delivery, defenseToolsCost);
        break;

    case 3:
        processTreasurerLondonOrder(delivery, treasurerLondonCost);
        break;

    case 0:
        cout<< "Exiting the program. Thank you." << endl;

    default:
        cout << "Invalid category choice." << endl;
        cout << " " << endl;
        break;
    }

    char addMore;
        cout << "Do you want to add more items? (y/n): ";
        cin >> addMore;
        system ("cls");

        if (addMore != 'y' && addMore != 'Y')
        {
            break; // Exit the loop if the user doesn't want to add more items
        }

} while (true);
```

Figure 7: Loop (do-while)

9.0 Decision (if..else, switch..case)

In the ‘main’ function, there is a ‘switch’ statement that checks the value of the ‘category’ variable entered by the user.

```
switch (category)
{
case 1:
    processMedicineOrder(delivery, medicineCost);
    break;

case 2:
    processDefenseToolsOrder(delivery, defenseToolsCost);
    break;

case 3:
    processTreasurerLondonOrder(delivery, treasurerLondonCost);
    break;

case 0:
    cout<< "Exiting the program. Thank you." << endl;

default:
    cout << "Invalid category choice." << endl;
    cout << " " << endl;
    break;
}
```

Figure 8: Switch..case

if..else statement decision-making based on condition. In this program, the program checks the value of the ‘age’ variable. If the age is greater than equal to 21, it displays a message and proceeds with the programs. If the age is less than 21, it displays a message and terminate the program using exit (0).

```
int age;

cout << "Enter your age        : ";
cin >> age;
cout << " " << endl;
 if (age >= 21) {
    cout << "    " << endl;
    cout << " Congratulations! You're old enough to access the system." << endl;
    cout << " " << endl;
    cout << " Please wait while you are redirected to the White Market." << endl;
    cout << " " << endl;
    cout << " " << endl;
    system("pause");

     } else {
    cout << " Oops! We suggest you go age a bit more, like a fine wine and return when you're 21 or older." << endl;
    cout << " See you in few more years.. " << endl;
    system("pause");
    exit(0); // Terminate the program
```

Figure 9: if...else statement

10.0 Functions

1. ‘main’ function is a entry point for the program. When we run the program, execution starts from ‘main’ function. This function contain codes that will be executed when it runs. It must have return type of ‘int’ that shows the exit status of the program. return value of 0 typically signifies that the program executed successfully.

```
int main()
{
    string name[1000], email, address[1000] ;
    int age, contactNum, category;
    double medicineCost, defenseToolsCost, treasurerLondonCost,shipping, delivery;
    int typeofmed, typeofTools, typeofItem, quantity;

    ManageMenu admin;
    admin.changeTerminalColor();


    // Display categories
    do
{

    displayCategories(); // calling code- part of program where function is called
                         //              - is a 'main' function where displayCategories is invoked, so when the program runs,
                         //                it starts executing from the 'main' function
                         //              - when it sees displayCategories, it jumps to 'the displaycategories' function, execute its code,
                         //                and return to  the calling code.

    cout << " " << endl;
    cout << "Enter the category (1/2/3): ";
    cin >> category;

    system("cls");

    switch (category)
    {
    case 1:
        processMedicineOrder(delivery, medicineCost);
        break;

    case 2:
        processDefenseToolsOrder(delivery, defenseToolsCost);
        break;

    case 3:
        processTreasurerLondonOrder(delivery, treasurerLondonCost);
        break;

    case 0:
        cout<< "Exiting the program. Thank you." << endl;

    default:
        cout << "Invalid category choice." << endl;
        cout << " " << endl;
        break;
    }

    char addMore;
        cout << "Do you want to add more items? (y/n): ";
        cin >> addMore;
        system ("cls");

        if (addMore != 'y' && addMore != 'Y')
        {
            break; // Exit the loop if the user doesn't want to add more items
        }
} while (true);


    return 0;
```

Figure 10: ‘main’ function

2. In the given example of functions in medicine category, ‘processMedicineOrder’ function and the parameters ‘double& delivery’ and ‘double& medicineCost’ are passed by reference. Passed by reference provide function with memory address(reference) of the original variable. This is indicated using ampersand (&) in the parameter list. Any changes made to the variable inside function will affect the original variable in the calling code. Pass by reference for ‘delivery’ and ‘medicineCost’ allow functions to modify the original variables in the calling code.

```
// Medicine order processing code...
// Function to process medicine order
void processMedicineOrder(double& delivery, double& medicineCost)
{
    string name, email, address, payment;
    int typeofmed, quantity, shipping, contactNum, age;
```

Figure 11: Pass by reference

3. 'void changeTerminalColor()' is a member of the 'ManageMenu' class. 'ManageMenu' class is a function to change console colors. So the constructer, the new 'ManageMenu' sets the color with white background and black font for the first console(like a front page in White Market system) and cyan for the rest of the codes.

```
class ManageMenu{


    public:
        ManageMenu() {
        system("color F0"); // F=bright white which is backround, 0-black for the word
        cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t    W   H   I   T   E      M   A   R   K   E   T" << endl;
        cout << "\n\n\n\n\n\n\n\n";
        Sleep(5000);
        system("CLS");
        menu(); // call to main function to load after executing the constructor
        }

     void changeTerminalColor() {
        //
        system("color 0B"); // 0-black, B-cyan
    }

    ~ManageMenu() {} // destructor
};
```

Figure 12: 'void changeTerminalColor'()

4. 'void displayCategories ()' is a function that displays the available categories in the White Market system. This function does not return value when it performs its tasks and no specific data being sent back to the calling code.

```
// Function declarations/prototypes (defining functions)
void displayCategories();
void processMedicineOrder(double& delivery, double& medicineCost); // variables in function is known as parameters or arguments
void processDefenseToolsOrder(double& delivery, double& defenseToolsCost);
void processTreasurerLondonOrder(double& delivery, double& treasurerLondonCost);
void collectContactInformation(string& name, int& contactNum, string& email, string& address, int& age);
```

Figure 13: void displayCategories();

11.0 Arrays

'name' and 'address' array is used to store name of customer and address. It is defined to hold up to 50 string values, providing storage for multiple address and name.

```
string name[1000], email, address[1000] ;
int age, contactNum, category;
double medicineCost, defenseToolsCost, treasurerLondonCost,shipping, delivery;
int typeofmed, typeofTools, typeofItem, quantity;
```

Figure 14: Arrays

12.0 Special/Creative features

Colorful terminal output: system("color") function to set the color of the terminal. This adds a visual and creative touch to the user interface.

```
    string name; // hide admin name

public:
    ManageMenu() {
    system("color F0"); // yellow terminal colour
    cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t
    cout << "\n\n\n\n\n\n\n\n";
    Sleep(5000);
    system("CLS");
    menu(); // call to main function to load after executing the constructor
    }

 void changeTerminalColor() {
    //
    system("color 0B"); // Light green
```

Figure 15: Colorful Terminal

File Handling: Used file handling to store customer information in a text file ('White market customers.txt). This feature allows us to keep records of customers details for future purposes.

```
// file "white market customers.txt"
ofstream out("C:\\Users\\Yaashiniy Segaran\\Desktop\\White market customers.txt", ios::app);

out << "Name: " << name << endl;
out << "Contact Number: " << contactNum << endl;
out << "Email: " << email << endl;
out << "Address: " << address << endl;
out << "------------------------\n";
out.close();  // Close the file when done
```

Figure 16: File handling

Clear Screen: Used 'system("cls") to clear the screen at various points in the code. This contributes to a clean and organized display for the user.

```
}

if(typeofTools == 3) {

   defenseToolsCost = quantity * 146.20;
   delivery = defenseToolsCost + shipping;

   cout << " " << endl;
   cout << " Your purchase will cost RM " << delivery << " including delivery cost." << endl;
   cout << " " << endl ;
   cout << " " << endl ;

   Sleep(3500);
}

cout << " Loading....." << endl;
Sleep(5500);
system ("cls");
```

Figure 17: Clear screen

Invoices and Sleep: This program generates invoices with customer details and total costs. The use of the 'Sleep' function introduces a delay, creating a more interactive and visually appealing user experience.

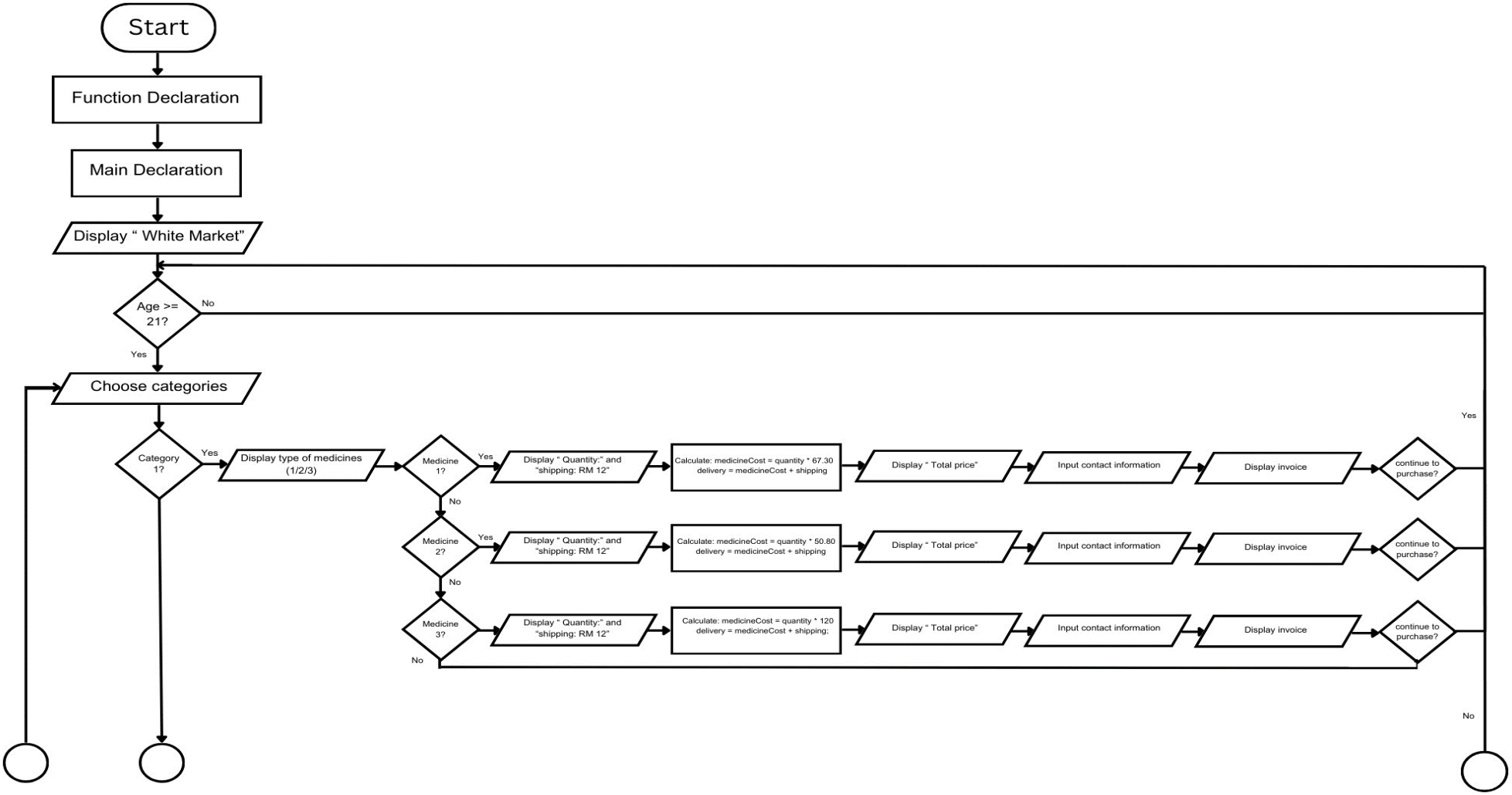
```
}

if(typeofTools == 3) {

   defenseToolsCost = quantity * 146.20;
   delivery = defenseToolsCost + shipping;

   cout << " " << endl;
   cout << " Your purchase will cost RM " << delivery << " including delivery cost." << endl;
   cout << " " << endl ;
   cout << " " << endl ;

   Sleep(3500);
}

cout << " Loading....." << endl;
Sleep(5500);
system ("cls");
```

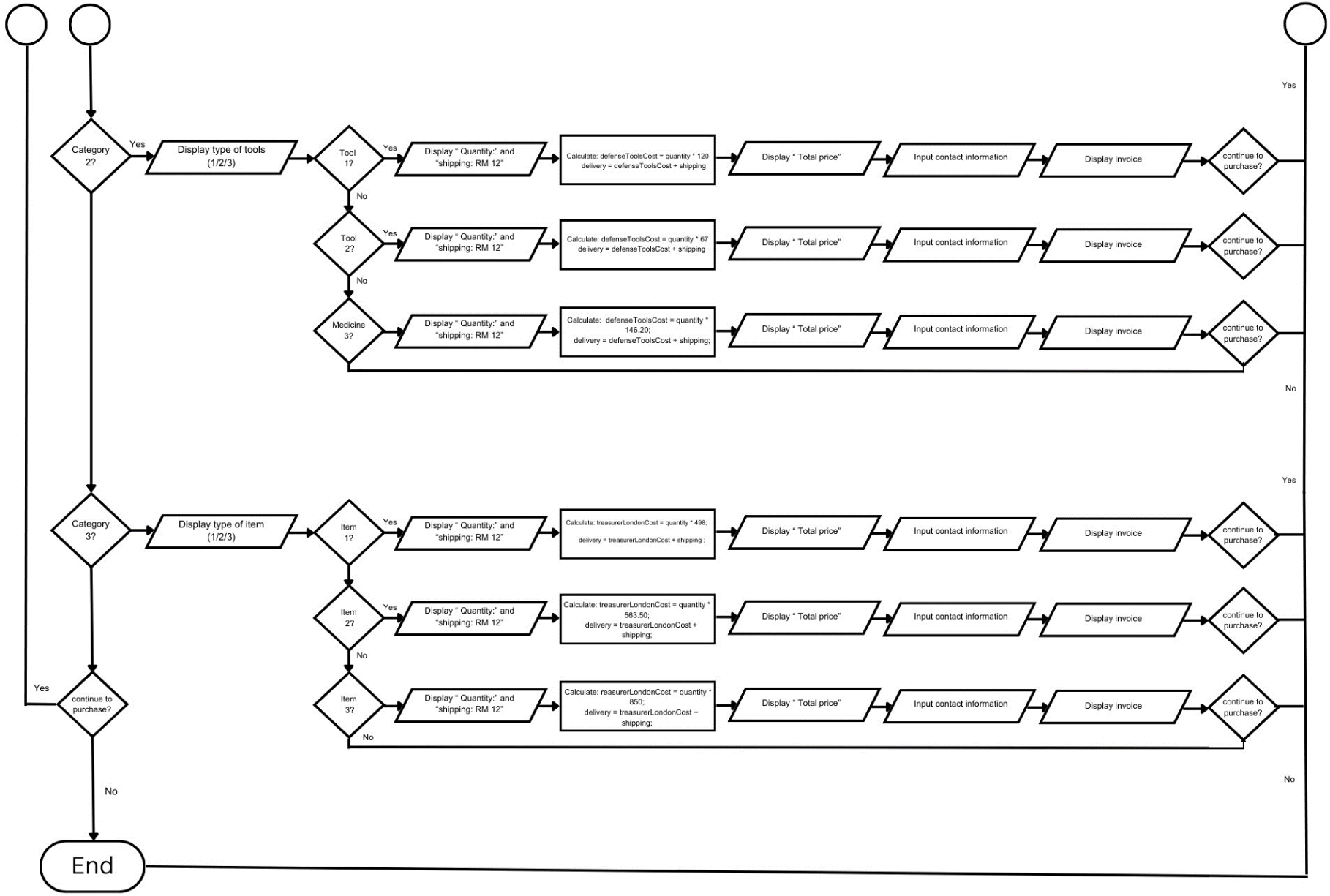
Figure 18: Sleep

```
cout << "                ----------------------------------------------------------------------------------------------------" << endl;
cout << "                                                  WHITE MARKET INVOICE                                              " << endl;
cout << "                ----------------------------------------------------------------------------------------------------" << endl;
cout << "                Name: " << name <<  endl;
cout << "                Address: " << address << endl;
cout << "                Email: " << email << endl;
cout << "                                                                                                                    " << endl;
cout << "                                                Amount   : " << defenseToolsCost << endl;
cout << "                                                Shipping : RM 7                                                     " << endl;
cout << "                                                Total    : RM " << delivery << endl ;
cout << "                                                                                                                    " << endl;
cout << "                                                                                                                    " << endl;
cout << "                ____________________________________________________________________________________________________" << endl;
cout << "                          Deliveries of ordered items are scheduled for one week after the date of purchase.        " << endl;
cout << "                           Your personal information are confidential and will retain for future purposes.          " << endl;
cout << "                ____________________________________________________________________________________________________" << endl;
cout << "                                                                                                                    " << endl;
cout << "                                                                                                                    " << endl;
```

Figure 19: Invoice

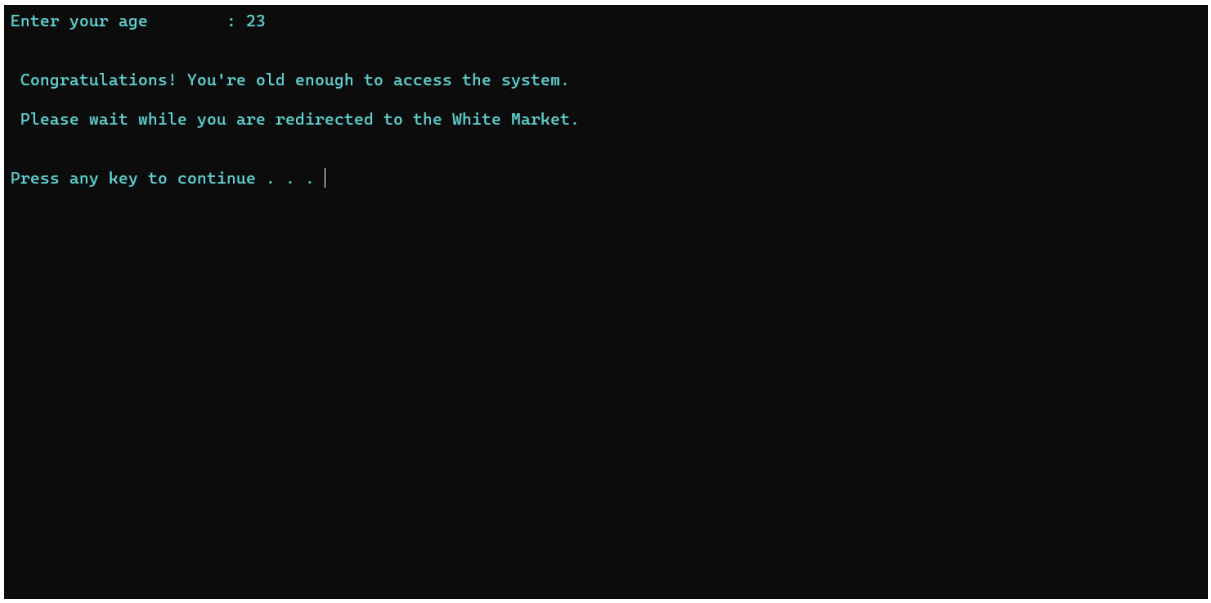
13.0 Flow chart




Category 2?
Yes
Display type of tools (1/2/3)
Tool 1?
Yes
Display " Quantity:" and "shipping: RM 12"
Calculate: defenseToolsCost = quantity * 120
delivery = defenseToolsCost + shipping
Display " Total price"
Input contact information
Display invoice
continue to purchase?
Yes
No
Tool 2?
Yes
Display " Quantity:" and "shipping: RM 12"
Calculate: defenseToolsCost = quantity * 67
delivery = defenseToolsCost + shipping
Display " Total price"
Input contact information
Display invoice
continue to purchase?
No
Medicine 3?
Display " Quantity:" and "shipping: RM 12"
Calculate: defenseToolsCost = quantity * 146.20;
delivery = defenseToolsCost + shipping;
Display " Total price"
Input contact information
Display invoice
continue to purchase?
No
Category 3?
Display type of item (1/2/3)
Item 1?
Yes
Display " Quantity:" and "shipping: RM 12"
Calculate: treasurerLondonCost = quantity * 498;
delivery = treasurerLondonCost + shipping ;
Display " Total price"
Input contact information
Display invoice
continue to purchase?
Yes
Item 2?
Yes
Display " Quantity:" and "shipping: RM 12"
Calculate: treasurerLondonCost = quantity * 563.50;
delivery = treasurerLondonCost + shipping;
Display " Total price"
Input contact information
Display invoice
continue to purchase?
No
Item 3?
Display " Quantity:" and "shipping: RM 12"
Calculate: reasurerLondonCost = quantity * 850;
delivery = treasurerLondonCost + shipping;
Display " Total price"
Input contact information
Display invoice
continue to purchase?
No
Yes
continue to purchase?
No
End

14.0 Output

W H I T E M A R K E T


Enter your age : 23
Congratulations! You're old enough to access the system.
Please wait while you are redirected to the White Market.
Press any key to continue . . .


YOUR LEGAL WHITE MARKET
White is black
SHOP AT YOUR OWN RISK
Categories
1: Medicines
2: Personal safety tools
3: Tobacco
Enter the category (1/2/3):

```
----------------------------------------------------------White Market Treasurer London----------------------------------------------------------

        We specialize in sourcing, distributing, and delivering rare medicines that are often challenging to obtain through traditional channels.


         .....................................     .....................................     .....................................
         .      1. Treasurer Luxury Black      .     .  2. Treasurer London Aluminium Black .     .  3. Porsche Design Jet Lighter P'3642 .
         .                                     .     .                                      .     .                                      .
         .    - 20x90mm premium cigarettes.    .     .      - 20x90mm premium cigarettes    .     .     - Special closed mechanism       .
         .    - Made in England                .     .      - Made in England               .     .       emitting a fine, consistent    .
         .    - Cardboard Pack                 .     .      - Gold Tipping Foil             .     .       flame.                         .
         .    - Special bland of Virginia      .     .      - Virginia Tobacco              .     .     - Cap is designed to protect the .
         .      Tobacco.                       .     .                                      .     .       inlet from dust.               .
         .                                     .     .                                      .     .     - Tank capacity: 1.0 gram        .
         .                                     .     .                                      .     .     - Black, Silver, Grey, Gold      .
         .                                     .     .                                      .     .                                      .
         .                        ~RM 498/pack .     .                    ~RM 563.50/pack .     .                              ~RM 850 .
         .....................................     .....................................     .....................................


 Type of item (1/2/3):  2
 Quantity             :  2
 Shipping (Malaysia) : RM 12


 Your purchase will cost RM 1139.00 including delivery cost.

 Loading.....
```

```
Enter your Full Name  : Yaashiniy Segaran

Contact Number        : 01322333443

Email address         : yaashu0508@gmail.com

Full Shipping Address : No 31, Jalan Tenang 9, Taman Damai Jaya, Skudai, Johor Bahru

Date : 19 November 2023


Please wait, your invoice is being processed and don't forget to save it.
 .........
```

```
                    ------------------------------------------------------------------------------------------------------------
                                                        WHITE MARKET INVOICE
                    ------------------------------------------------------------------------------------------------------------
                    Name: Yaashiniy Segaran
                    Address: No 31, Jalan Tenang 9, Taman Damai Jaya, Skudai, Johor Bahru
                    Email: yaashu0508@gmail.com
                    Date: 9 November 2023

                                                  Amount   : RM 1127.00
                                                  Shipping : RM 12
                                                  Total    : RM 1139.00


                    ------------------------------------------------------------------------------------------------------------
                                Deliveries of ordered items are scheduled for one week after the date of purchase.
                                 Your personal information are confidential and will retain for future purposes.
                    ------------------------------------------------------------------------------------------------------------


Do you want to add more items? (y/n): n
```

```
--------------------------------
Process exited after 313.5 seconds with return value 0
Press any key to continue . . .
```

15.0 Conclusion

In conclusion, the White Market system serves as an advanced platform designed to provide a unique shopping experience for customers seeking specialized items, including medicines and self-defense tools. The system employs a robust set of functionalities, blending user authentication, age verification and an automated ordering process to ensure a secure and convenient service.

The code is structured with clarity, utilizing various control structures such as loops and decision-making statements to manage user interactions effectively. The integration of functions including those handling medicines and self-defense tool orders showcases a well-organized approach to code design. Especially, the code incorporates a user-friendly interface, making it accessible to a diverse audience.

The implementation of file logging and customer information management demonstrates a commitment to maintaining a transparent and accountable business operation. Furthermore, the code employs libraries such as <fstream> and <string> to facilitate file operations and string manipulations respectively contributing to the overall functionality of the system.

The White Market system not only meets the requirements of a conventional online marketplace but also integrates age-related restrictions responsibly, aligning with legal and ethical considerations. The code's inclusion of sleep functions and colour manipulation adds a touch of creativity to the user experience, contributing to the system's overall appeal.

In summary, the White Market system stands as a testimony to effective coding practices, combining functionality, security and user engagement. The objectives set for this system have been met, providing a foundation for potential enhancements and expansions in the future. The development team has succeeded in creating an automated and secure marketplace, ensuring a positive and tailored experience for customers seeking unique and specialized products.

References

1. American Society for Quality (2019). *What is a Flowchart? Process Flow Diagrams & Maps | ASQ*. [online] Asq.org. Available at: https://asq.org/quality-resources/flowchart.

2. GeeksforGeeks. (2022). *How to Clear Console in C++?* [online] Available at: https://www.geeksforgeeks.org/how-to-clear-console-in-cpp/.

3. cppsecrets.com. (n.d.). *C++ program to print colored text . | C++ | cppsecrets.com*. [online] Available at: https://cppsecrets.com/article.php?id=13576 [Accessed 19 Nov. 2023].

4. www.javatpoint.com. (n.d.). Function Prototype in C++ - javatpoint. [online] Available at: https://www.javatpoint.com/function-prototype-in-cpp.

5. corob-msft (n.d.). Functions (C++). [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/cpp/cpp/functions-cpp?view=msvc-170.

6. corob-msft (n.d.). void (C++). [online] learn.microsoft.com. Available at: https://learn.microsoft.com/en-us/cpp/cpp/void-cpp?view=msvc-170.

7. Educative: Interactive Courses for Software Developers. (n.d.). Pass by value vs. pass by reference. [online] Available at: https://www.educative.io/answers/pass-by-value-vs-pass-by-reference.

8. Jayasekara, P.M. (2023). travel-management-system. [online] GitHub. Available at: https://github.com/pasinduweb/travel-management-system [Accessed 19 Nov. 2023].