LAPORAN PROYEK MATA KULIAH PEMROGRAMAN DASAR

SEMESTER GASAL 2025/2026

Online Shopping

Anggota Tim

1. 25/565780/TK/63818 | Muhammad Daffa Wirawan

Departemen Teknik Elektro dan Teknologi Informasi
Fakultas Teknik
Universitas Gadjah Mada
2025

Github Repository: https://github.com/davausername/Project-Online-Shop-Muhammad-Daffa-Wirawan-565780

1. DESKRIPSI APLIKASI

Aplikasi yang dikembangkan dalam projek ini merupakan simulasi aplikasi online shopping sederhana dimana users bisa berinterasi dengan fitur-fitur yang ada di dalam aplikasi tersebut seperti register, memilih category product, pembelian barang, dan simulasi proses delivery yang sangat sederhana.

Aplikasi ini dirancang untuk kemudahan user dalam berinteraksi dengan interface yang sederhana dan mudah digunakkan. Dalam awal pembuatannya, aplikasi ini dirancang dengan UML **Use Case** dan **Class Case** Diagram yang rancang di software Enterprise Architect. Setelah melalui tahapan tersebut, aplikasi ini dikonstruksikan dengan bahasa **C++** yang berbasis **console output**, sehingga seluruh interaksi program dilakukan melalui terminal atau console.

Projek dikembangkan sebuah aplikasi online shop mencakup analisis dan desain:

- Analisis dan pemodelan menggunakan UML Use Case & Class Case Diagram
- Pengembangan dengan menerapkan konsep OOP (Object Oriented Programming)
- Bahasa pemograman C++
- Penyimpanan data sederhana menggunakkan .txt file
- Error handling
- Interaksi user dengan terminal atau console

2. ANALISIS KEBUTUHAN

Aplikasi Online Shop berbasis C++ ini memiliki feature *user interactive* yang dirancang untuk mempermudah user dalam melakukan process pembelian barang. Aplikasi ini menampilkan menu dengan tampilan daftar product, product description, user expenses, dan melakukan proses simulasi delivery yang sangat sederhana.

2.1. Kebutuhan Fungsional

Kebutuhan fungsional yang terdapat di dalam aplikasi ini mencangkup berbagai functions yang digunakkan untuk menjalankan process tertentu dalam menanggapi input dari user seperti pembelian atau checkout suatu produk dalam Aplikasi.

Aplikasi online shop ini juga didesain untuk melakukan validasi terhadap user input, seperti validasi dari registrasi dan sign-in dari user. Selain itu, aplikasi ini juga menyimpan data yang di-*enter oleh* user dalam hal-hal tertentu.

Beberapa kebutuhan fungsional tersebut terletak dalam table di bawah ini.

Table 1 Kebutuhan Fungsional

Req-ID	Kebutuhan Fungsional	Deskripsi
F-01	Sign Up	Feature to add user to database
F-02	Sign In	Feature remember user and access all functionality
F-03	View User Details	Feature to view user details (full name, password, etc.
F-04	Remove Account	Feature to remove user from database
F-05	Save Product to Cart	Feature to save product to cart
F-06	Checkout	Feature to checkout all products in user cart
F-07	Buy	Feature to buy product
F-08	Simple Delivery Process	Simple feature to view delivery process
F-09	Track User Expenses	Feature to see how much user has spent in the shop

2.2. Kebutuhan Non Fungsional

Kebutuhan non-funsional yang ada dalam aplikasi online shop ini meliputi:

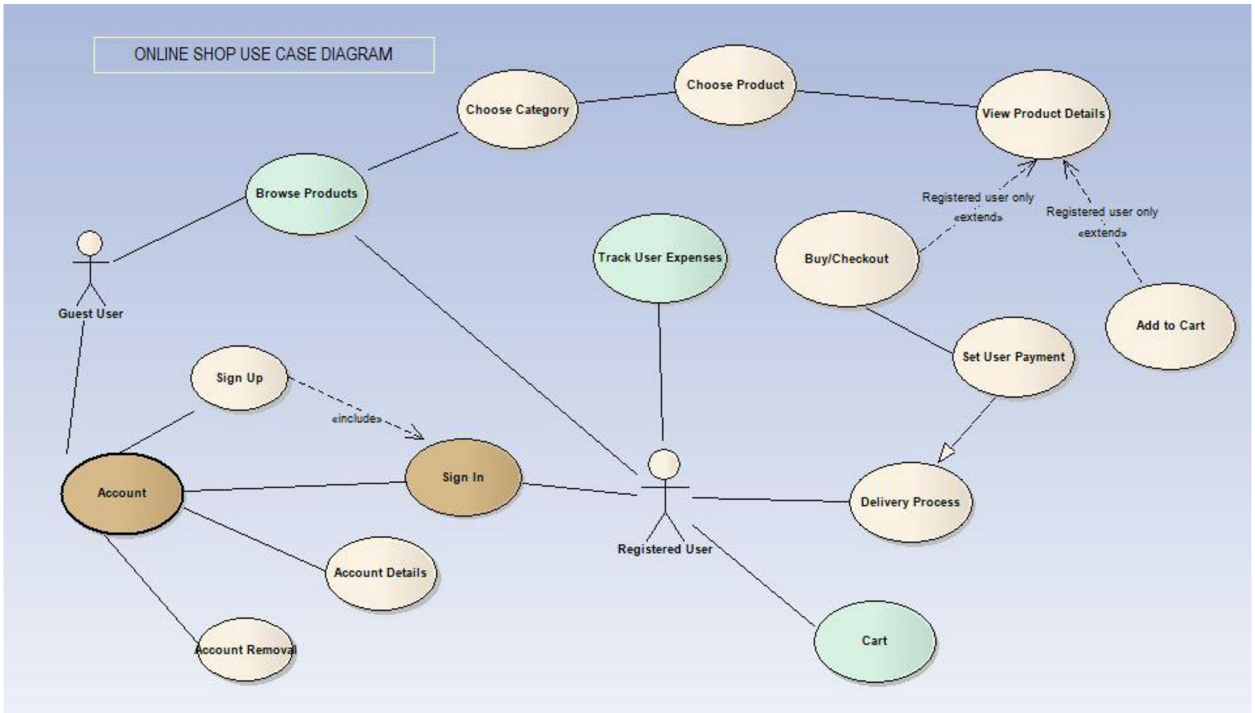
- Desaign UML Use Case & Class Case dalam memahami isi program
- Console or Terminal
- Quick performance and easy usability
- Portability (Windows, Linux, MacOs)
- Reability
- Maintability
- Flexible Usage

3. PEMODELAN DAN PERANCANGAN SISTEM

3.1. Use Case Diagram

Our application was designed by use case diagram to visually mapping, structuring, and defining system. It acts a bridge to help defining what system does, make program structure easily understood, and helping to identify system features. This diagram also assists programmer or developer in implementing and developing this online shop application.

Below is a picture of this app Use Case Diagram:



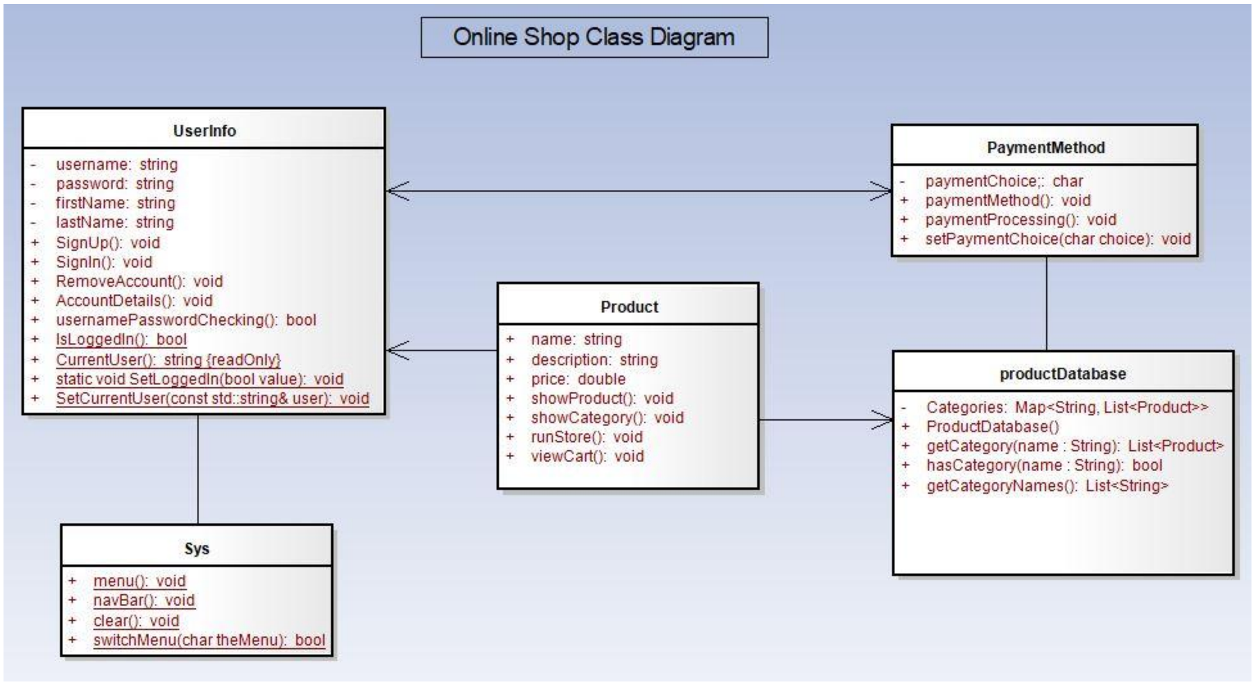
Use Case Skenario

Req-ID	Kebutuhan Fungsional	Skenario use case
F-01	Sign Up	1. User choose Account (if from menu/home) 2. Click Sign Up 3. Enter User Information. 4. Enter and it saves the data to database
F-02	Sign In	1. User choose Account 2. Click Sign In 3. Enter username and password 4. Check and successful then sends user to menu
F-03	View User Details	1. User choose Account 2. Click Account Details 3. Enter username and password 4. Fetch the data and shows the user Information and buying record
F-04	Remove User	1. User choose Account 2. Click Remove Account 3. Enter username and password 4. Remove User Account

F-05	Save Product to Cart (Registered Feature)	1. Browse product you want by choose the category. 2. Then, choose stuff you want to see 3. After that, click “1” to save to cart
F-06	Buy & Checkout (Registered Feature)	1. Firstly, you need to save products to cart or buy directly. 2. If you save products to cart, you visit the cart section and checkout all the stuff
F-07	Choose Payment Options (Registered Feature)	1. Buy stuff you want or checkout products in your cart. 2. Then, choose available payment options (Credit Card, Debit Card, PayPal, Wise) 3. Wait until it successful
F-08	Simple Delivery Process (Registered Feature)	1. Wait until your payment successful 2. It will redirect you to Delivery Process 3. You can leave and go back to delivery process in “Order” menu section.
F-09	Track User Expenses (Registered Feature)	1. In the homepage, enter “P” for visit the page 2. Then it shows your total expenses.

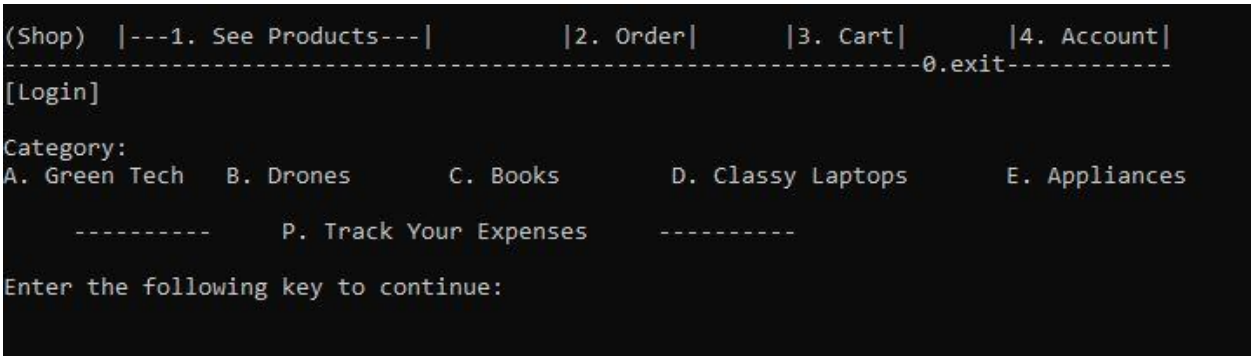
3.2. Class Diagram

Pada bagian ini dapat digunakan beberapa jenis UML diagram untuk menggambarkan aspek perilaku (behavioral) dari system atau aplikasi.



3.3. Perancangan User Interface

User Interface digunakkan pada aplikasi ini yaitu console dengan pilihan menu dengan simple and interactive design yang dapat diakses dengan memasukkan **char** yang terdapat pada menu dan click enter to continue the process.



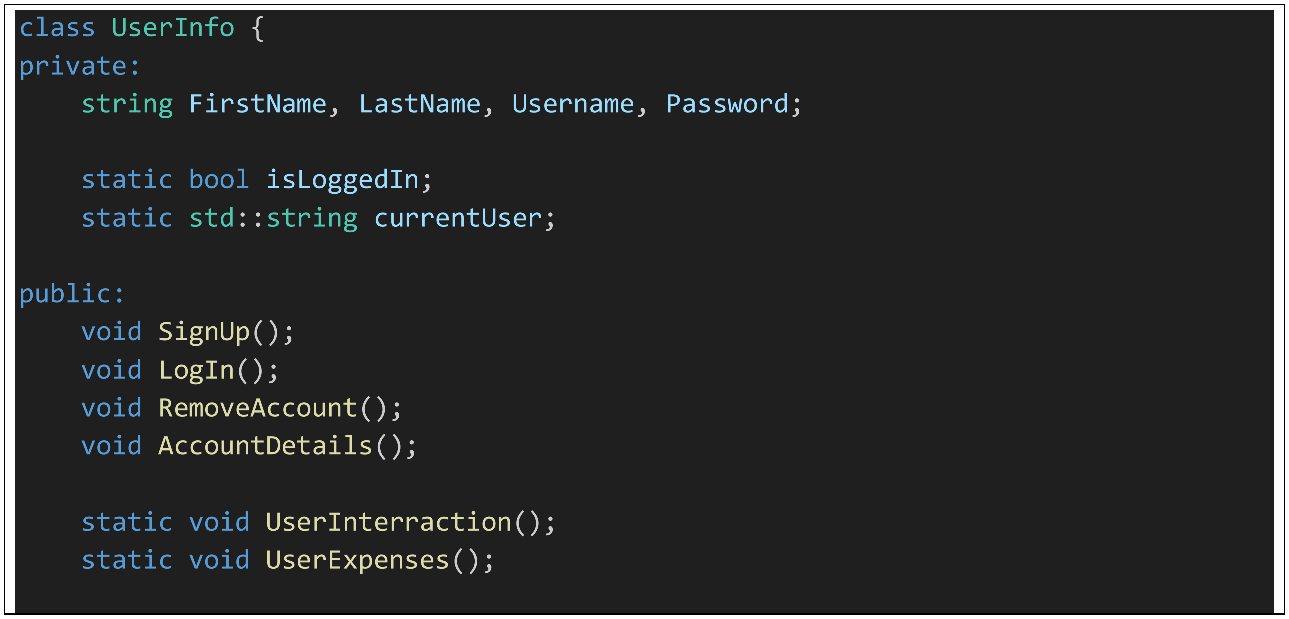
```
(Shop)  |---1. See Products---|          |2. Order|       |3. Cart|        |4. Account|
-------------------------------------------------------------------0.exit-----------
[Login]

Category:
A. Green Tech    B. Drones        C. Books         D. Classy Laptops        E. Appliances

     ----------      P. Track Your Expenses      ----------

Enter the following key to continue:
```

3.4. Transformasi Class Diagram Menjadi Kode Program

Class diagram yang telah rancang ditransformasikan dalam bentuk header files (.h) untuk mengimplementasikan ke dalam program c++(cpp). Dalam proses ini, terdapat sedikit perubahan nama dan sedikit syntax untuk menyesuaikan fungsionalitas ke dalam system.

Dibawah ini merupakan classes yang ada di dalam beberapa header files:

Account.h



```cpp
class UserInfo {
private:
    string FirstName, LastName, Username, Password;

    static bool isLoggedIn;
    static std::string currentUser;

public:
    void SignUp();
    void LogIn();
    void RemoveAccount();
    void AccountDetails();

    static void UserInterraction();
    static void UserExpenses();
    // checking
    static bool IsLoggedIn();
    static void SetLoggedIn(bool value);

    static const std::string& CurrentUser();
    static void SetCurrentUser(const std::string& user);

    bool usernamePasswordChecking(const string& inputUser, const string&
inputPass);
};
```

System.h

```
class Sys {
public:
    static void menu();
    static void navBar();
    static void clear();
    static void deliveryProcess();
    static bool switchMenu(char theMenu);
};
```

Product.h

```
class Product {
public:
    string name;
    string description;
    double price;

public:
    void showProduct(const Product& p);
    void showCategory(const vector<Product>& items);
    void runStore();
    void viewCart();

    void AddProductToUserCart(const std::string& username, const std::string&
productName);
    void buyProduct(const std::string& username, const double& productPrice);
    double getProductPrice(const string& productName);
    void clearCart(const string& username);
};
```

Payment.h

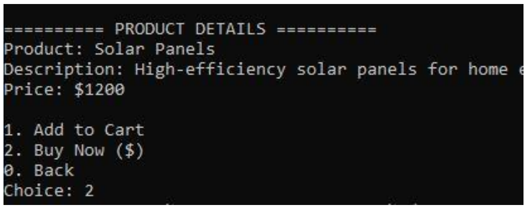
```
class PaymentMethod {
public:
    char paymentChoice;

public:
    void paymentMethod();
    void paymentProcessing();

    void setPaymentChoice(char choice) {
        paymentChoice = choice;
    }
};
```

4. IMPLEMENTASI APLIKASI

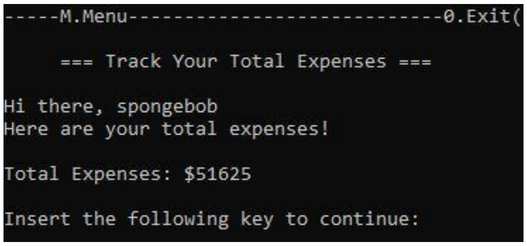
Pada bab ini silahkan dilaporkan luaran atau hasil dari pengembangan aplikasi yang telah dirancang pada tahap sebelumnya. Hal-hal yang dapat dijelaskan di sini:

- Perubahan yang dilakukan dari hasil UML tool *meliputi perubahan nama untuk beberapa header files, menambahakan libararies,* dan *beberapa syntax lainnya* seperti #include, #pragma once.
- Komponen yang telah diimplementasikan dari awal: Sign-up, Sign-in, Account-removal, Cart, Buy & checkout, Delivery process, dan track user expenses. Lalu terdapat penambahan beberapa features seperti Account Details dan Payment Options yang telah mengakitbatkan Class Diagram Renovation.
- Tangkapan layar dari sistem informasi:

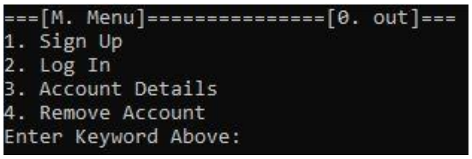
  Dibawah ini merupakan screenshot beberapa system informasi yang ada di dalam aplikasi, dibagian awal terdapat product details dimana users bisa beriteraksi dengan menyimpan barang ke cart atau membeli saat itu juga, di bagian kanan yang terdapat account menu di mana user bisa sign-up atau logged-in, selain itu terdapat juga track user expenses dan user cart yang menampilkan total price semua barang yang ada di cart.



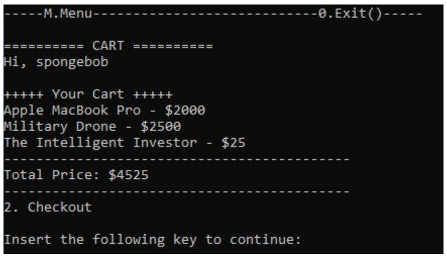
```
========== PRODUCT DETAILS ==========
Product: Solar Panels
Description: High-efficiency solar panels for home e
Price: $1200

1. Add to Cart
2. Buy Now ($)
0. Back
Choice: 2
```



```
-----M.Menu---------------------------0.Exit(

     === Track Your Total Expenses ===

Hi there, spongebob
Here are your total expenses!

Total Expenses: $51625

Insert the following key to continue:
```



```
===[M. Menu]===============[0. out]===
1. Sign Up
2. Log In
3. Account Details
4. Remove Account
Enter Keyword Above:
```

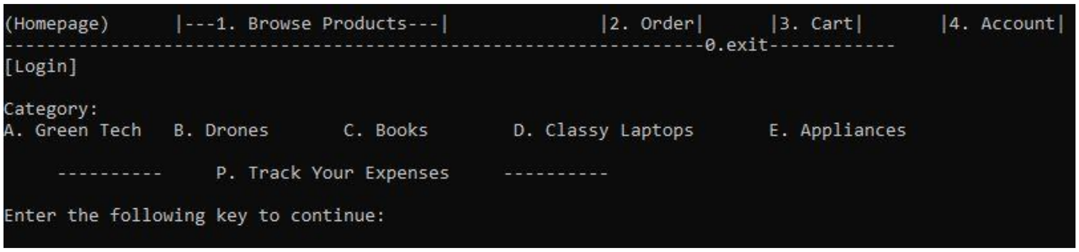


```
-----M.Menu---------------------------0.Exit()-----

========== CART ==========
Hi, spongebob

+++++ Your Cart +++++
Apple MacBook Pro - $2000
Military Drone - $2500
The Intelligent Investor - $25
-------------------------------------------
Total Price: $4525
-------------------------------------------
2. Checkout

Insert the following key to continue:
```

Untuk features lebih lengkapnya terdapat di bawah ini.

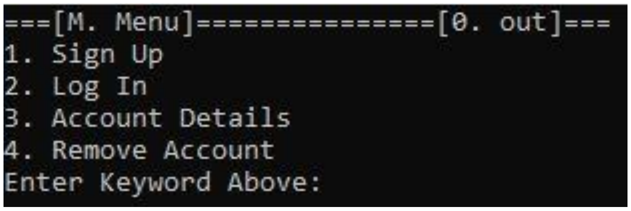
4.1. Fitur F-01 Register New Person / Sign Up

The **Sign up** feature serves as bridge to connect users to the database (.txt in our case). Once a user has registered, they gain full access to the app's functionality, allowing them to purchase products and track their records.



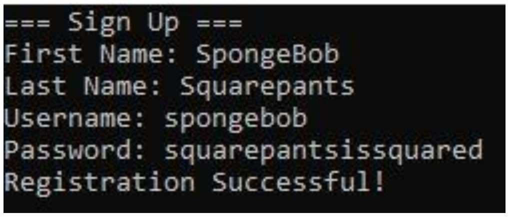
```
(Homepage)      |---1. Browse Products---|            |2. Order|      |3. Cart|       |4. Account|
-----------------------------------------------------------------0.exit-----------
[Login]

Category:
A. Green Tech   B. Drones       C. Books        D. Classy Laptops       E. Appliances

     ----------      P. Track Your Expenses     ----------

Enter the following key to continue:
```

From the menu, the user enters **'4'** to go to the account section.



```
===[M. Menu]===============[0. out]===
1. Sign Up
2. Log In
3. Account Details
4. Remove Account
Enter Keyword Above:
```

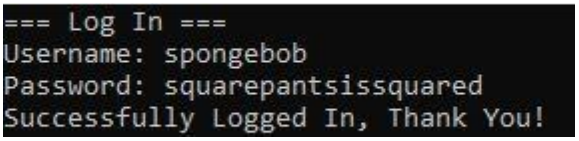
Next, select **'2'** to sign up and press enter. Then, enter your details.



```
=== Sign Up ===
First Name: SpongeBob
Last Name: Squarepants
Username: spongebob
Password: squarepantsissquared
Registration Successful!
```

4.2. Fitur F-02 Sign In

Our app also has feature to remember user and its data. To do it you need to go to the account section and click Sign in and then enter your username and password like below here.

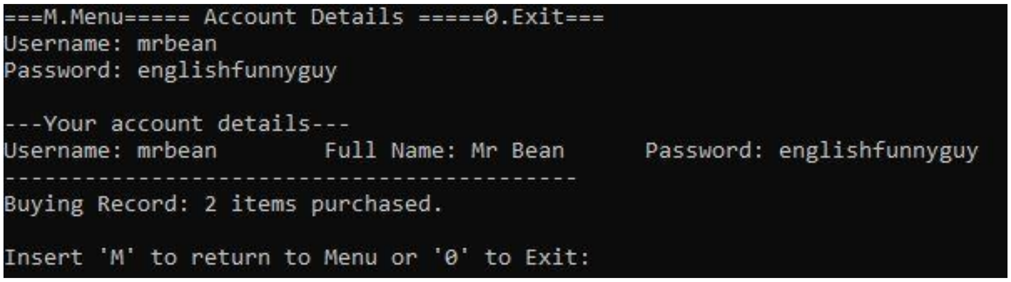


```
=== Log In ===
Username: spongebob
Password: squarepantsissquared
Successfully Logged In, Thank You!
```

After that, it automatically redirects you to the homepage!

4.3. Fitur F-03 View User Details

If you want to see your details, you can do it with this feature. What you need to do is go to the account section and choose User Details. Then insert your username and password. Here’s an example below!

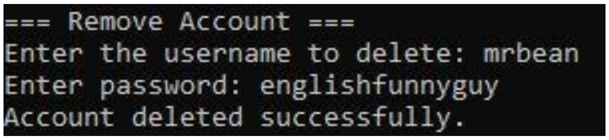


```
===M.Menu===== Account Details =====0.Exit===
Username: mrbean
Password: englishfunnyguy

---Your account details---
Username: mrbean        Full Name: Mr Bean      Password: englishfunnyguy
-------------------------------------------
Buying Record: 2 items purchased.

Insert 'M' to return to Menu or '0' to Exit:
```

4.4. Fitur F-04 Remove User Account

If you want to remove your user account, go the account section, then choose remove account. Enter your username and password. And yeah, your username has been removed.

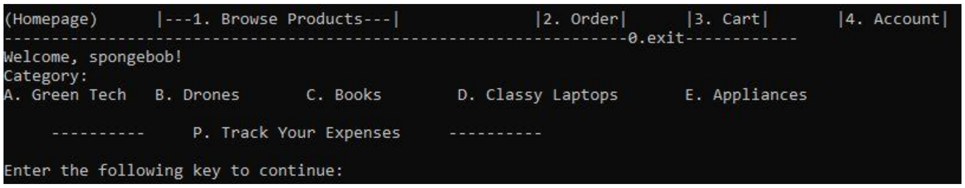
Ex. To remove the user account named **mrbean**, the user must enter the corresponding username and password for authentication.



```
=== Remove Account ===
Enter the username to delete: mrbean
Enter password: englishfunnyguy
Account deleted successfully.
```

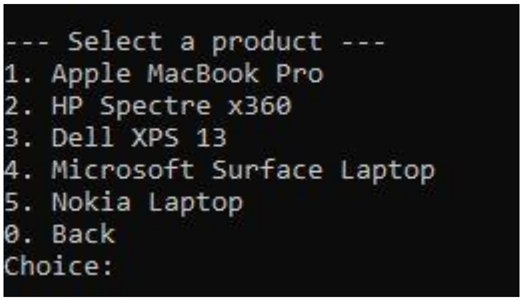
4.5. Fitur F-06 Save Product to Cart

This feature allows you to save any products or stuff you want to buy for later. Once you add an item to your cart, it will be remembered whenever you return to the app.



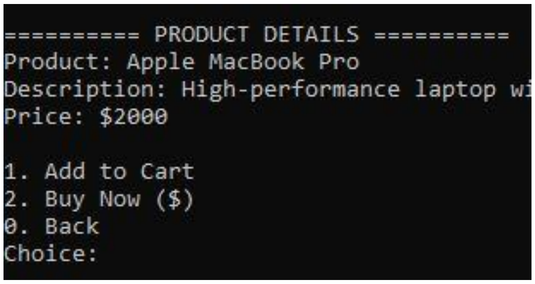
```
(Homepage)      |---1. Browse Products---|              |2. Order|      |3. Cart|       |4. Account|
------------------------------------------------------------------0.exit------------
Welcome, spongebob!
Category:
A. Green Tech   B. Drones       C. Books        D. Classy Laptops       E. Appliances

     ----------     P. Track Your Expenses     ----------

Enter the following key to continue:
```

In the home page you can browse more categories by visiting "Browse Products" or choose any category in the menu.

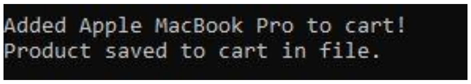


```
--- Select a product ---
1. Apple MacBook Pro
2. HP Spectre x360
3. Dell XPS 13
4. Microsoft Surface Laptop
5. Nokia Laptop
0. Back
Choice:
```

Then choose any desired product.



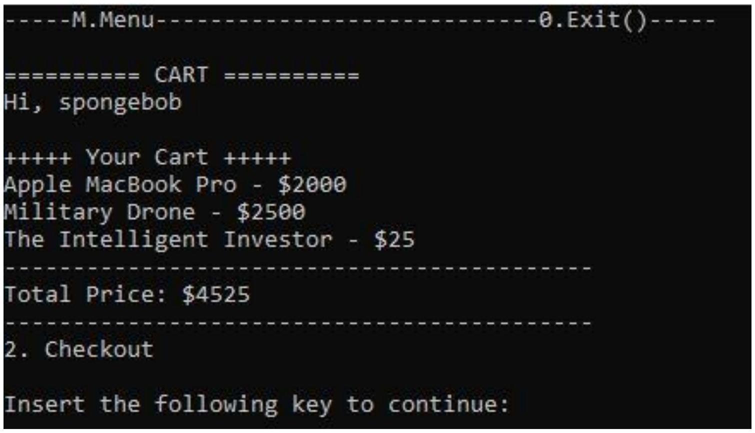
```
========== PRODUCT DETAILS ==========
Product: Apple MacBook Pro
Description: High-performance laptop wi
Price: $2000

1. Add to Cart
2. Buy Now ($)
0. Back
Choice:
```



```
Added Apple MacBook Pro to cart!
Product saved to cart in file.
```

After you choose a stuff you want to buy, you'll see a brief description. To add to your cart just enter '1' (as simple as it is).

If you want to check your cart, you may go back to the menu and enter '3' for cart. Once you've got into the cart page, you'll see such appearance below!

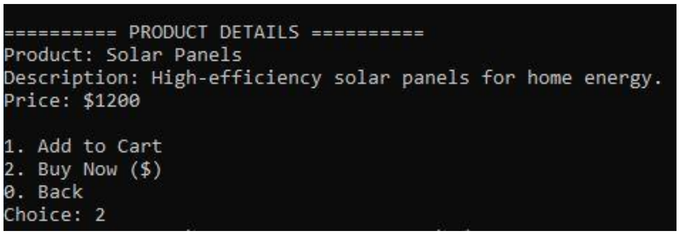


```
-----M.Menu---------------------------0.Exit()-----

========== CART ==========
Hi, spongebob

+++++ Your Cart +++++
Apple MacBook Pro - $2000
Military Drone - $2500
The Intelligent Investor - $25
-----------------------------------------
Total Price: $4525
-----------------------------------------
2. Checkout

Insert the following key to continue:
```

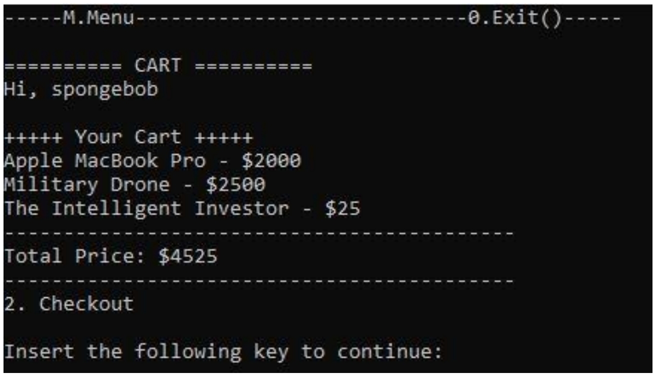
4.6. Fitur F-07 Buy & Checkout

Fitur buy & checkout merupakan fitur dimana user dapat membeli suatu barang dalam aplikasi online shop ini. Fitur tersebut dapat dilihat dengan mengakses product yang dinginkan dan click buy atau checkout pada cart section. Selanjutnya, pilih payment method, dan tunggu hingga proses berhasil.

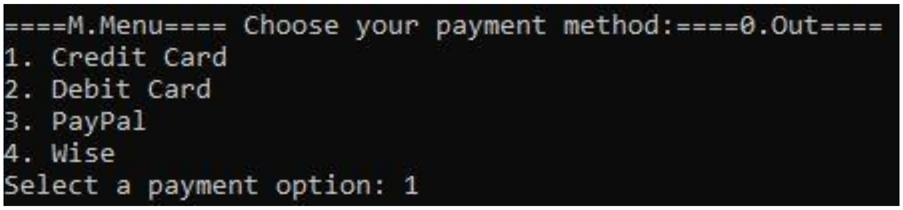
Untuk mencoba fitur tersebut pilih product yang ingin dibeli dan enter “2” untuk Buy now atau Checkout.



```
========== PRODUCT DETAILS ==========
Product: Solar Panels
Description: High-efficiency solar panels for home energy.
Price: $1200

1. Add to Cart
2. Buy Now ($)
0. Back
Choice: 2
```

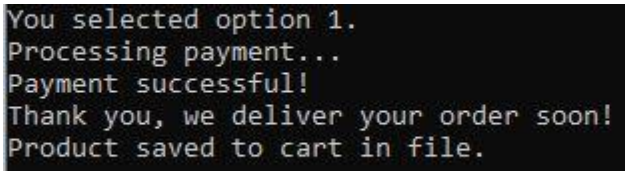


```
-----M.Menu---------------------------0.Exit()-----

========== CART ==========
Hi, spongebob

+++++ Your Cart +++++
Apple MacBook Pro - $2000
Military Drone - $2500
The Intelligent Investor - $25
-----------------------------------------
Total Price: $4525
-----------------------------------------
2. Checkout

Insert the following key to continue:
```



```
====M.Menu==== Choose your payment method:====0.Out====
1. Credit Card
2. Debit Card
3. PayPal
4. Wise
Select a payment option: 1
```

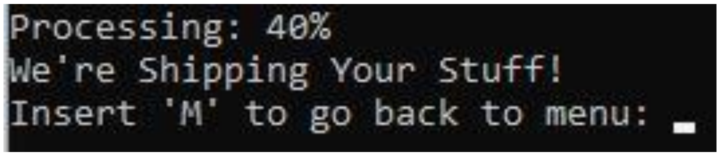
Selanjutnya, pilih metode pembayaran yang diinginkan dan enter. Setelah itu pembayaran diprocess dan successful.



```
You selected option 1.
Processing payment...
Payment successful!
Thank you, we deliver your order soon!
Product saved to cart in file.
```

4.7. Fitur F-09 Simple Delivery Process

Feature simple delivery process merupakan fitur untuk track process pengiriman barang yang sedang bekerja. Proses delivery dalam aplikasi ini sangat simple dengan waktu pengiriman yang berbasis fix (ten seconds) atau tidak random untuk kemudahan prediksi waktu.

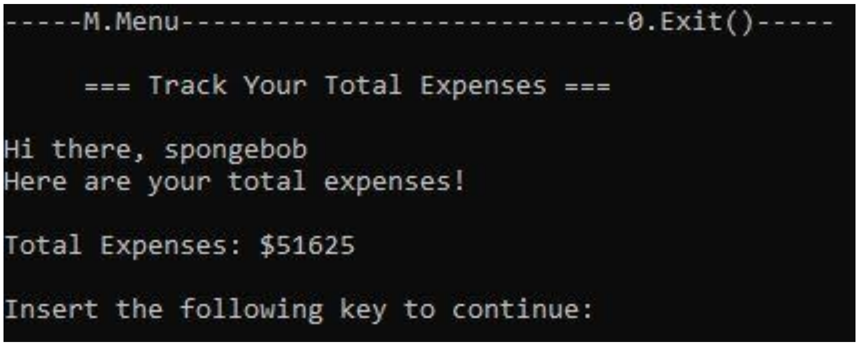


```
Processing: 40%
We're Shipping Your Stuff!
Insert 'M' to go back to menu: _
```

4.8. Fitur F-10 Track User Expenses

Fitur track user expenses merupakan feature untuk mengetahui berapa banyak uang atau total yang telah dihabiskan oleh seorang costumer di aplikasi ini.

Dibawah ini merukan contoh berapa total uang yang telah dihabiskan oleh user bernama Spongebob dengan total expenses mencapai $51625.



```
-----M.Menu---------------------------0.Exit()-----

     === Track Your Total Expenses ===

Hi there, spongebob
Here are your total expenses!

Total Expenses: $51625

Insert the following key to continue:
```

LAMPIRAN

- Full source code and beyond can be accessed through: https://github.com/davausername/Project-Online-Shop-Muhammad-Daffa-Wirawan-565780

- Brief Source code:

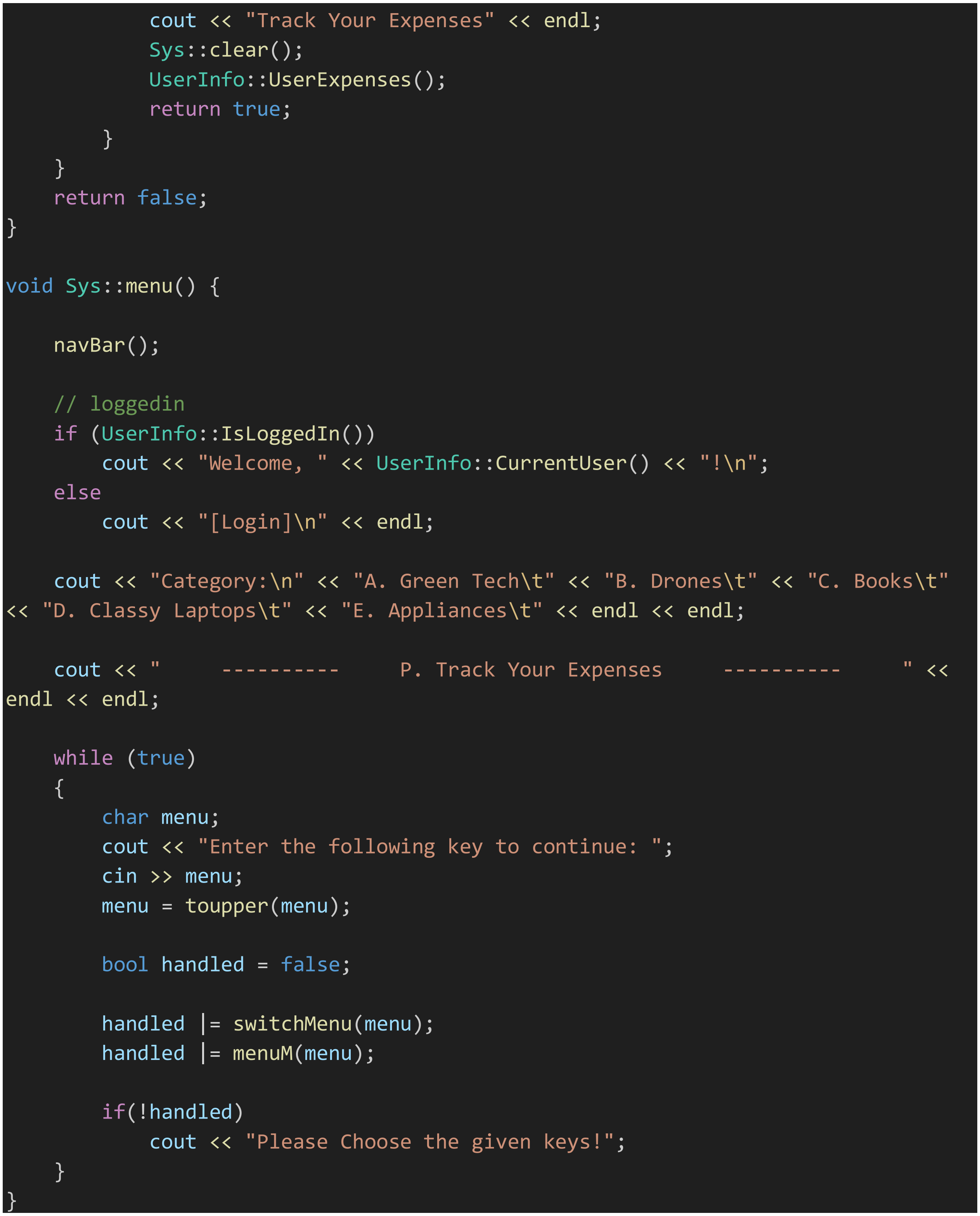
Main.cpp

```cpp
#include <iostream>
#include "Header/System.h"
using namespace std;

int main() {

    Sys::menu();

    return 0;

}
```

Interface.cpp

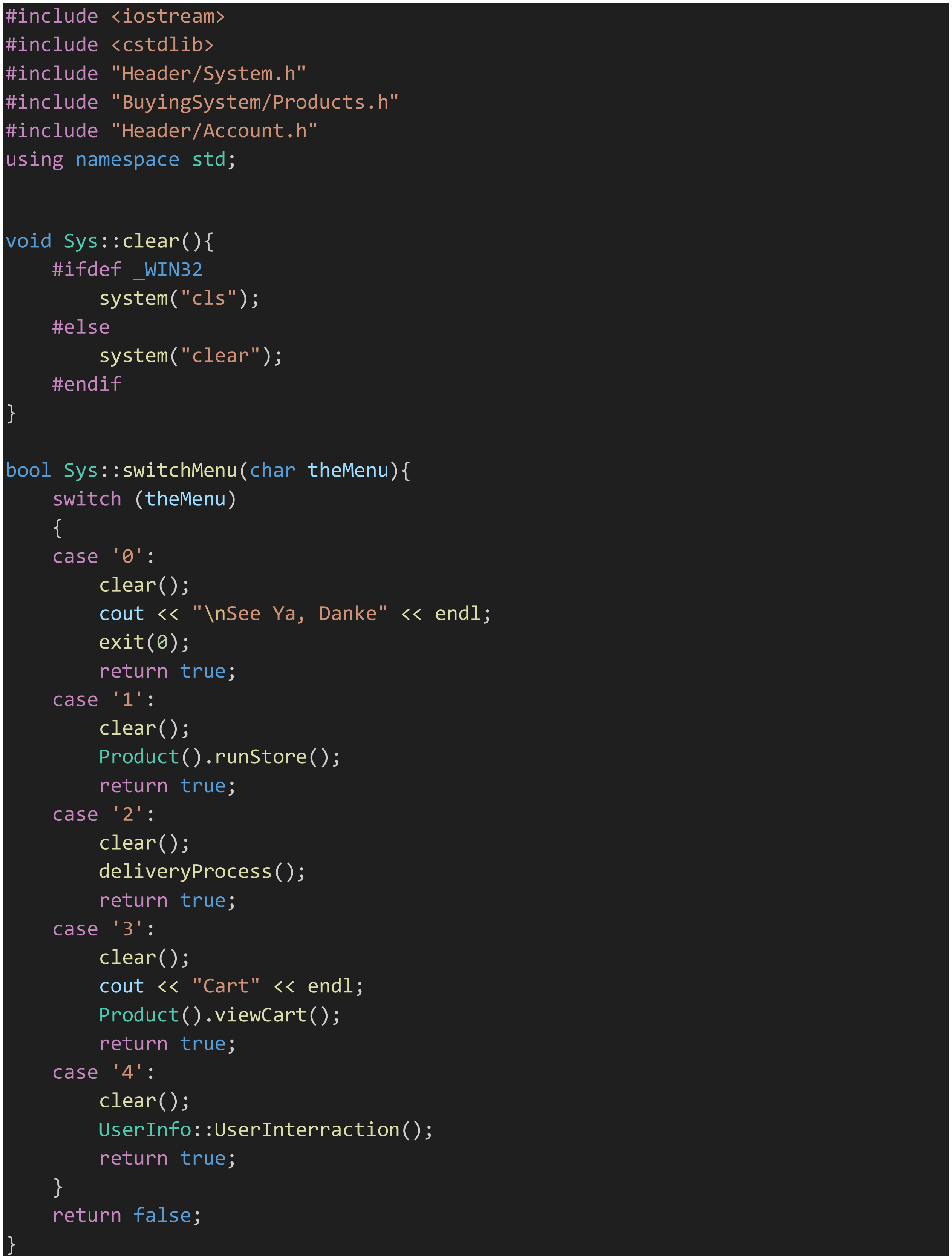
```cpp
#include <iostream>
#include "Header/System.h"
#include "BuyingSystem/ProductDataBase.h"
#include "BuyingSystem/Products.h"
#include "Header/Account.h"

bool UserInfo::isLoggedIn = false;
std::string UserInfo::currentUser = "";
```

```
using namespace std;


void Sys::navBar(){
    cout << endl << "(Homepage)\t" << "|---1. Browse Products---|\t\t" << "|2. 
Order|\t" << "|3. Cart|\t" << "|4. Account|" << endl;
    cout << "------------------------------------------------------------------
0.exit------------" << endl;
}

bool menuM (char menuM) {
    ProductDatabase db;
    switch (menuM) {
        case 'A': {
            Sys::clear();
            cout << "Green Tech" << endl;
            Product().showCategory(db.getCategory("Green Tech"));
            return true;
        }
        case 'B': {
            Sys::clear();
            cout << "Drones" << endl;
            Product().showCategory(db.getCategory("Drones"));
            return true;
        }
        case 'C': {
            Sys::clear();
            cout << "Books" << endl;
            Product().showCategory(db.getCategory("Books"));
            return true;
        }
        case 'D': {
            Sys::clear();
            cout << "Classy Laptops" << endl;
            Product().showCategory(db.getCategory("Classy Laptops"));
            return true;
        }
        case 'E': {
            Sys::clear();
            cout << "Appliances" << endl;
            Product().showCategory(db.getCategory("Appliances"));
            return true;
        }
        case 'P': {
```



```
            cout << "Track Your Expenses" << endl;
            Sys::clear();
            UserInfo::UserExpenses();
            return true;
        }
    }
    return false;
}

void Sys::menu() {

    navBar();

    // loggedin
    if (UserInfo::IsLoggedIn())
        cout << "Welcome, " << UserInfo::CurrentUser() << "!\n";
    else
        cout << "[Login]\n" << endl;

    cout << "Category:\n" << "A. Green Tech\t" << "B. Drones\t" << "C. Books\t"
<< "D. Classy Laptops\t" << "E. Appliances\t" << endl << endl;

    cout << "     ----------     P. Track Your Expenses     ----------     " <<
endl << endl;

    while (true)
    {
        char menu;
        cout << "Enter the following key to continue: ";
        cin >> menu;
        menu = toupper(menu);

        bool handled = false;

        handled |= switchMenu(menu);
        handled |= menuM(menu);

        if(!handled)
            cout << "Please Choose the given keys!";
    }
}
```

System.cpp



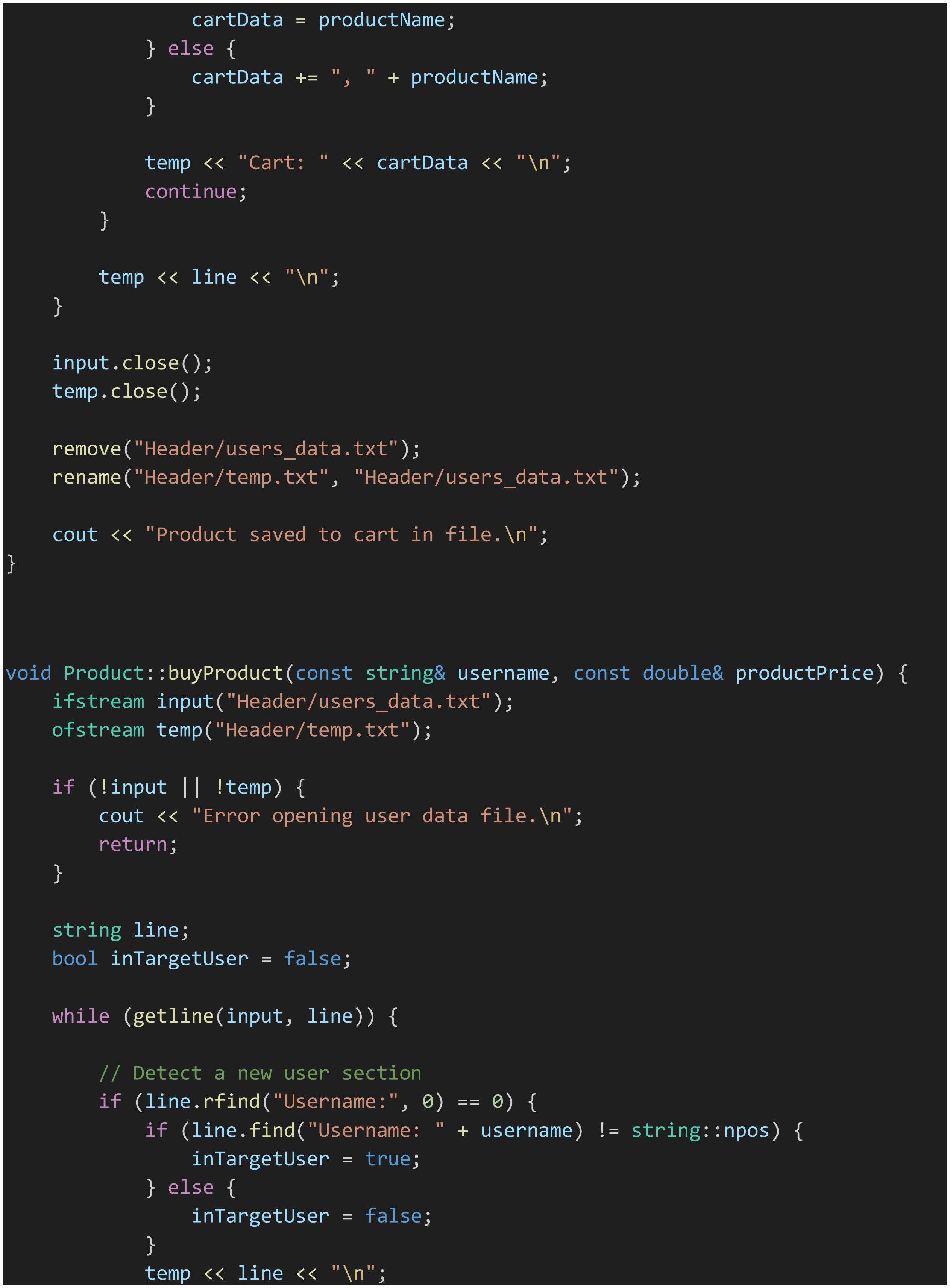
```cpp
#include <iostream>
#include <cstdlib>
#include "Header/System.h"
#include "BuyingSystem/Products.h"
#include "Header/Account.h"
using namespace std;


void Sys::clear(){
    #ifdef _WIN32
        system("cls");
    #else
        system("clear");
    #endif
}

bool Sys::switchMenu(char theMenu){
    switch (theMenu)
    {
    case '0':
        clear();
        cout << "\nSee Ya, Danke" << endl;
        exit(0);
        return true;
    case '1':
        clear();
        Product().runStore();
        return true;
    case '2':
        clear();
        deliveryProcess();
        return true;
    case '3':
        clear();
        cout << "Cart" << endl;
        Product().viewCart();
        return true;
    case '4':
        clear();
        UserInfo::UserInterraction();
        return true;
    }
    return false;
}
```

Payment.cpp

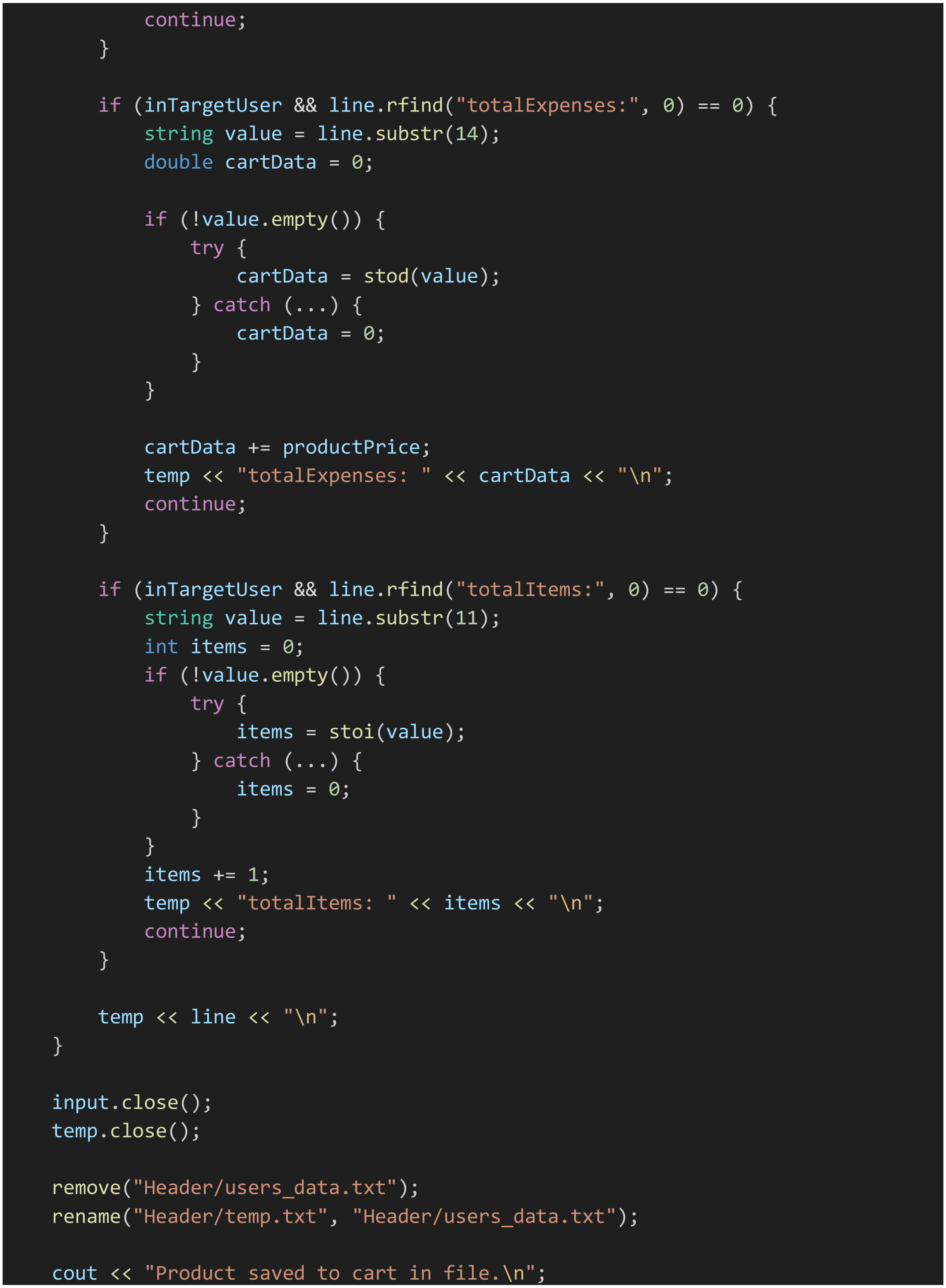
```
#include <iostream>
#include "payment.h"
#include "../Header/System.h"
using std::cout;
using std::cin;
using std::endl;

void PaymentMethod::paymentMethod() {
    cout << "====M.Menu==== Choose your payment method:====0.Out====" << endl;
    cout << "1. Credit Card\n2. Debit Card\n3. PayPal\n4. Wise" << endl;

    while (true) {
        char choice;
        cout << "Select a payment option: ";
        cin >> choice;

        PaymentMethod pm;
        pm.setPaymentChoice(choice);
        if (choice == '0') {
            cout << "See you, Danke." << endl;
            exit(0);
            return;
        } else if (choice == '1' || choice == '2' || choice == '3' || choice ==
'4') {
            cout << "You selected option " << choice << "." << endl;
            paymentProcessing();
            return;
        } else if (choice == 'M' || choice == 'm') {
            Sys::menu();
            return;
        } else {
            cout << "Invalid choice. Please try again." << endl;
        }
    }

}

void PaymentMethod::paymentProcessing() {
    cout << "Processing payment..." << endl;
    cout << "Payment successful!" << endl;
}
```

Processing.cpp

```
#include <iostream>
#include <fstream>
#include <string>
#include "Products.h"
using namespace std;

void Product::AddProductToUserCart(const string& username, const string&
productName) {
    ifstream input("Header/users_data.txt");
    ofstream temp("Header/temp.txt");

    if (!input || !temp) {
        cout << "Error opening user data file.\n";
        return;
    }

    string line;
    bool inTargetUser = false;

    while (getline(input, line)) {

        // Detect a new user
        if (line.rfind("Username:", 0) == 0) {
            if (line.find("Username: " + username) != string::npos) {
                inTargetUser = true;
            } else {
                inTargetUser = false;
            }
            temp << line << "\n";
            continue;
        }

        if (inTargetUser && line.rfind("Cart:", 0) == 0) {
            string cartData = line.substr(5); // after "Cart:"

            // trim
            while (!cartData.empty() && (cartData[0] == ' ' || cartData[0] ==
'\t'))
                cartData.erase(cartData.begin());

            // Append product
            if (cartData.empty()) {
```



```
                cartData = productName;
            } else {
                cartData += ", " + productName;
            }

            temp << "Cart: " << cartData << "\n";
            continue;
        }

        temp << line << "\n";
    }

    input.close();
    temp.close();

    remove("Header/users_data.txt");
    rename("Header/temp.txt", "Header/users_data.txt");

    cout << "Product saved to cart in file.\n";
}



void Product::buyProduct(const string& username, const double& productPrice) {
    ifstream input("Header/users_data.txt");
    ofstream temp("Header/temp.txt");

    if (!input || !temp) {
        cout << "Error opening user data file.\n";
        return;
    }

    string line;
    bool inTargetUser = false;

    while (getline(input, line)) {

        // Detect a new user section
        if (line.rfind("Username:", 0) == 0) {
            if (line.find("Username: " + username) != string::npos) {
                inTargetUser = true;
            } else {
                inTargetUser = false;
            }
            temp << line << "\n";
```

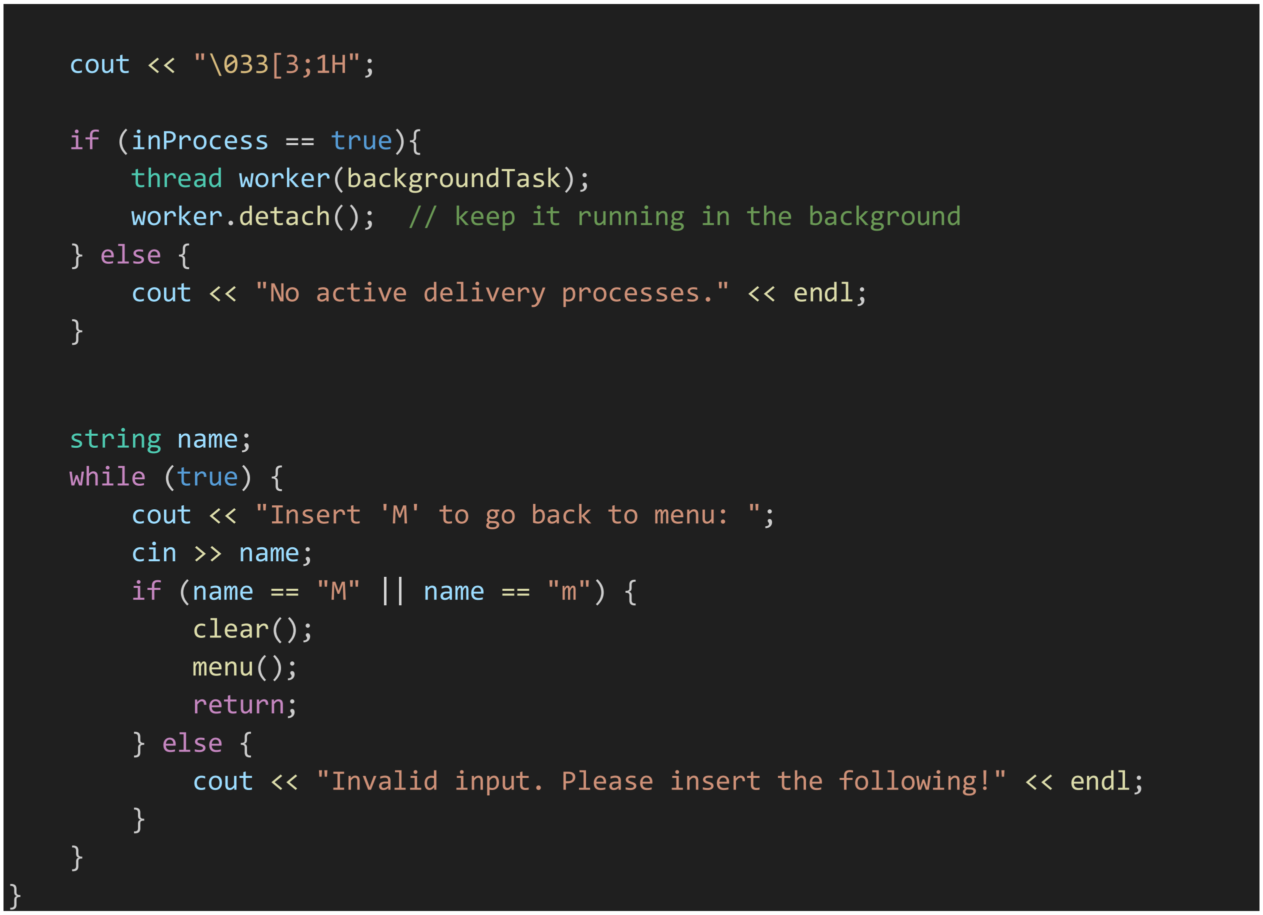


```
            continue;
        }

        if (inTargetUser && line.rfind("totalExpenses:", 0) == 0) {
            string value = line.substr(14);
            double cartData = 0;

            if (!value.empty()) {
                try {
                    cartData = stod(value);
                } catch (...) {
                    cartData = 0;
                }
            }

            cartData += productPrice;
            temp << "totalExpenses: " << cartData << "\n";
            continue;
        }

        if (inTargetUser && line.rfind("totalItems:", 0) == 0) {
            string value = line.substr(11);
            int items = 0;
            if (!value.empty()) {
                try {
                    items = stoi(value);
                } catch (...) {
                    items = 0;
                }
            }
            items += 1;
            temp << "totalItems: " << items << "\n";
            continue;
        }

        temp << line << "\n";
    }

    input.close();
    temp.close();

    remove("Header/users_data.txt");
    rename("Header/temp.txt", "Header/users_data.txt");

    cout << "Product saved to cart in file.\n";
```

```
}
```

Delivery.cpp

```
#include <iostream>
#include <thread>
#include <chrono>
#include <string>
#include <atomic>
#include "Header/System.h"
using namespace std;
bool inProcess = false;

atomic<int> currentNumber(0);

void backgroundTask() {
    inProcess = false;
    for (int i = 0; i <= 100; i+=10) {
        currentNumber = i;

        cout << "\033[s";

        cout << "\033[1;1H";
        cout << "\033[2K";
        cout << "Processing: " << currentNumber << "%" << flush;
        if (currentNumber == 20) {
            cout << "\nYour Stuff is About to be Delivered!" << endl;
        } else if (currentNumber == 40) {
            cout << "\nWe're Shipping Your Stuff!          " << endl;
        } else if (currentNumber == 50) {
            cout << "\nYour Stuff is on the Way!           " << endl;
        } else if (currentNumber == 100) {
            cout << "\nYour Stuff has been Delivered! Thank You!" << endl;
        }

        cout << "\033[u" << flush;

        this_thread::sleep_for(chrono::seconds(2));
    }
}

void Sys::deliveryProcess() {
    cout << "\033[2J";
```



```
    cout << "\033[3;1H";

    if (inProcess == true){
        thread worker(backgroundTask);
        worker.detach();  // keep it running in the background
    } else {
        cout << "No active delivery processes." << endl;
    }


    string name;
    while (true) {
        cout << "Insert 'M' to go back to menu: ";
        cin >> name;
        if (name == "M" || name == "m") {
            clear();
            menu();
            return;
        } else {
            cout << "Invalid input. Please insert the following!" << endl;
        }
    }
}
```

Total terdapat 13 c++ files dengan 7 .cpp files dan 5 header files

Semua files c++ project ini terdapat **di link Github:**
https://github.com/davausername/Project-Online-Shop-Muhammad-Daffa-Wirawan-565780

Thank You || Danke