UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

WORKSHOP 1 REPORT

Name:	Siew Chung Seng
Matric Number:	B031910342
Course:	BITS
Project Title:	Bookshop Stock Management System
Supervisor Name:	DR. SATRYA FAJRI PRATAMA
Supervisor Signature:	
Evaluator Name:	TS. MASLITA BINTI ABD AZIZ

Table Of Contents

Chapter 1: Introduction

1.1 Background

A Bookshop Stock Management System is basically for management of incoming and material from the Bookshop. This application will help the staff to allocate the records of the books in their stock accurately without any time constraint. This program is developed using Microsoft Visual Studio and Xampp. It can be used in any Bookshop for maintaining database details and their quantities. This project is used for handle user needs.

1.2 Problem Statement

Due to heavy demand of books in these competitive world, a non-digital system would be hard for the staff to keep track on their stock. This could lead to a mess on their work and lead to loss of trust of the customer. Also, non-digital stocking system will take a long time for the staff to check the stock one by one and the staff may make mistake while doing the work which cause the inaccuracy of the stock.

1.3 Objectives

This project embarks on few objectives. First, this program will help the staff to manage the stock system well in such a way that it increases the accuracy of the stock rather than a manual system. Second, this project will benefits the staff by decreasing their workload, reduce the usage of paper and it is cost and time effective as we have a computerized system. Last but not least, it will always make sure the details of the item are up to date as everything are stored in the database.

1.4 Scope of Project

This project provides a great help in managing the data in well-mannered order. It is designed specially to maintain the data in a sequential manner and to save the tome and efforts of database. This system will provide convenience for the staff in such a way that it provide functions such as view item details, search items and order items. The user will be able to manage their stock in a quicker way and more accurate.

Chapter 2: Analysis of Problem

2.1 Analysis of Problem

Manual system requires a lot of paperwork and consumes a lot of time and costly for any field of jobs. It would be a mess if there is a mistake made in doing the stock checking or information checking using manual system. It requires a lot of effort to search for the error and fix it. By establishing a digitalised system, the amount of work load of the user would be decreased and would save time for the user.

As the amount of the items are growing everyday, it is hard to maintain the information of every single items existed. So, a digital system would helps to maintain the data of the items and always up to date. It will also helps to reduce the risk of management of data of the items as human are tend to make mistakes.

2.2 Structure Chart

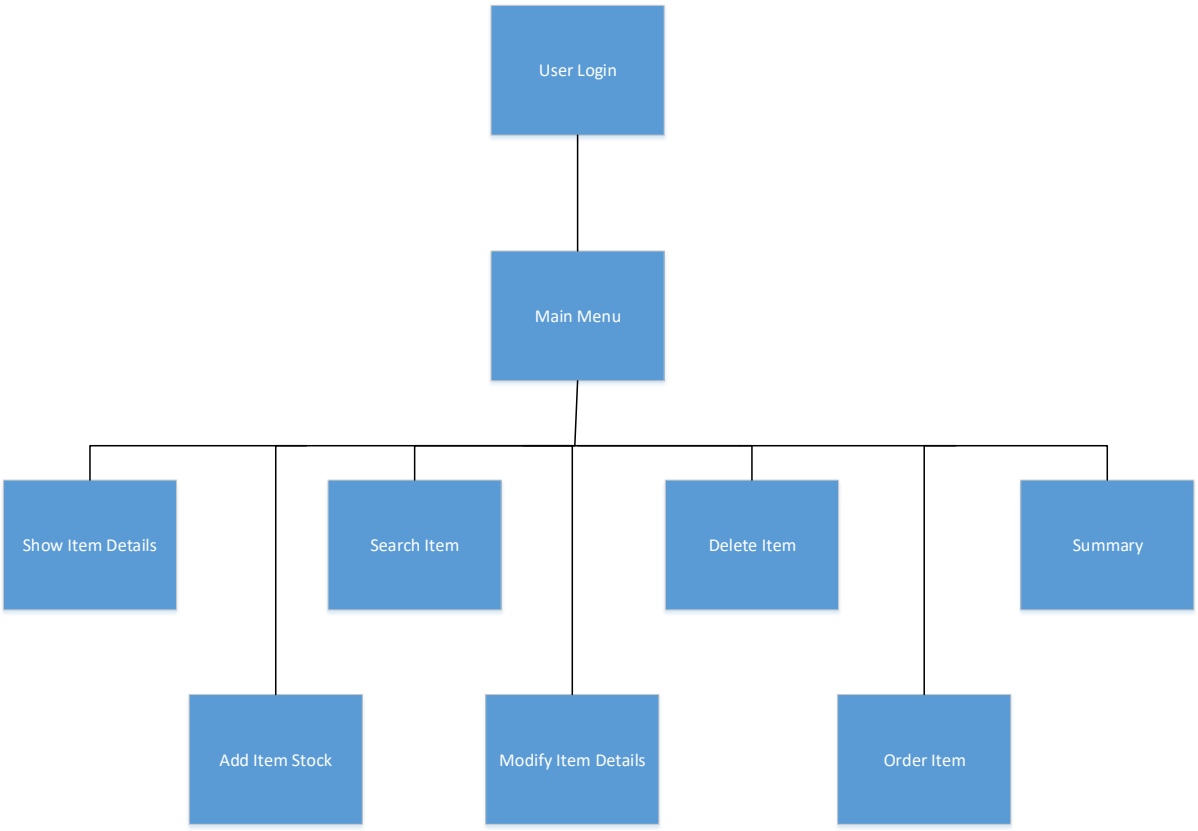


Figure 2.2.1 Structure Chart

Chapter 3: Design

3.1 Flowchart

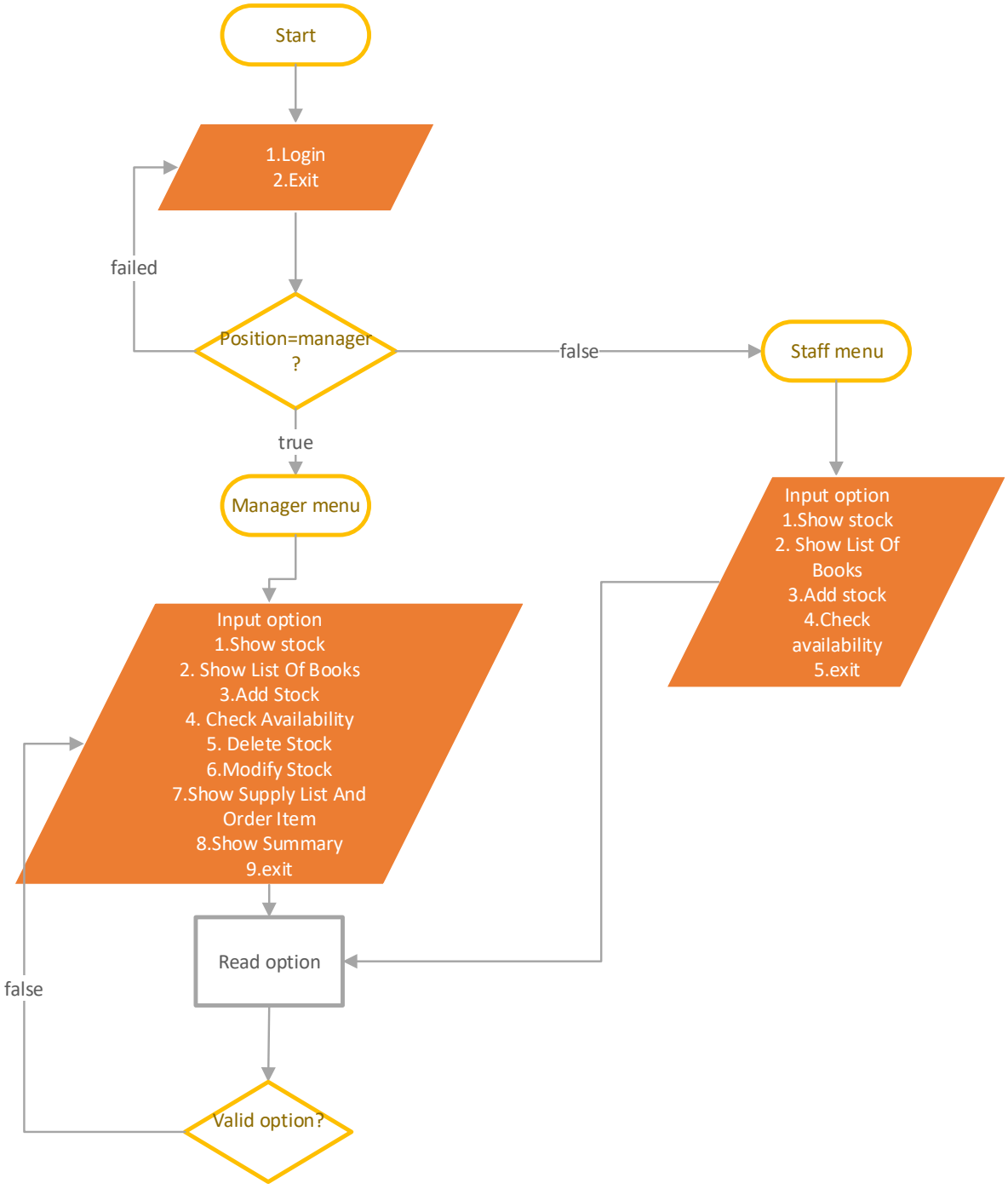
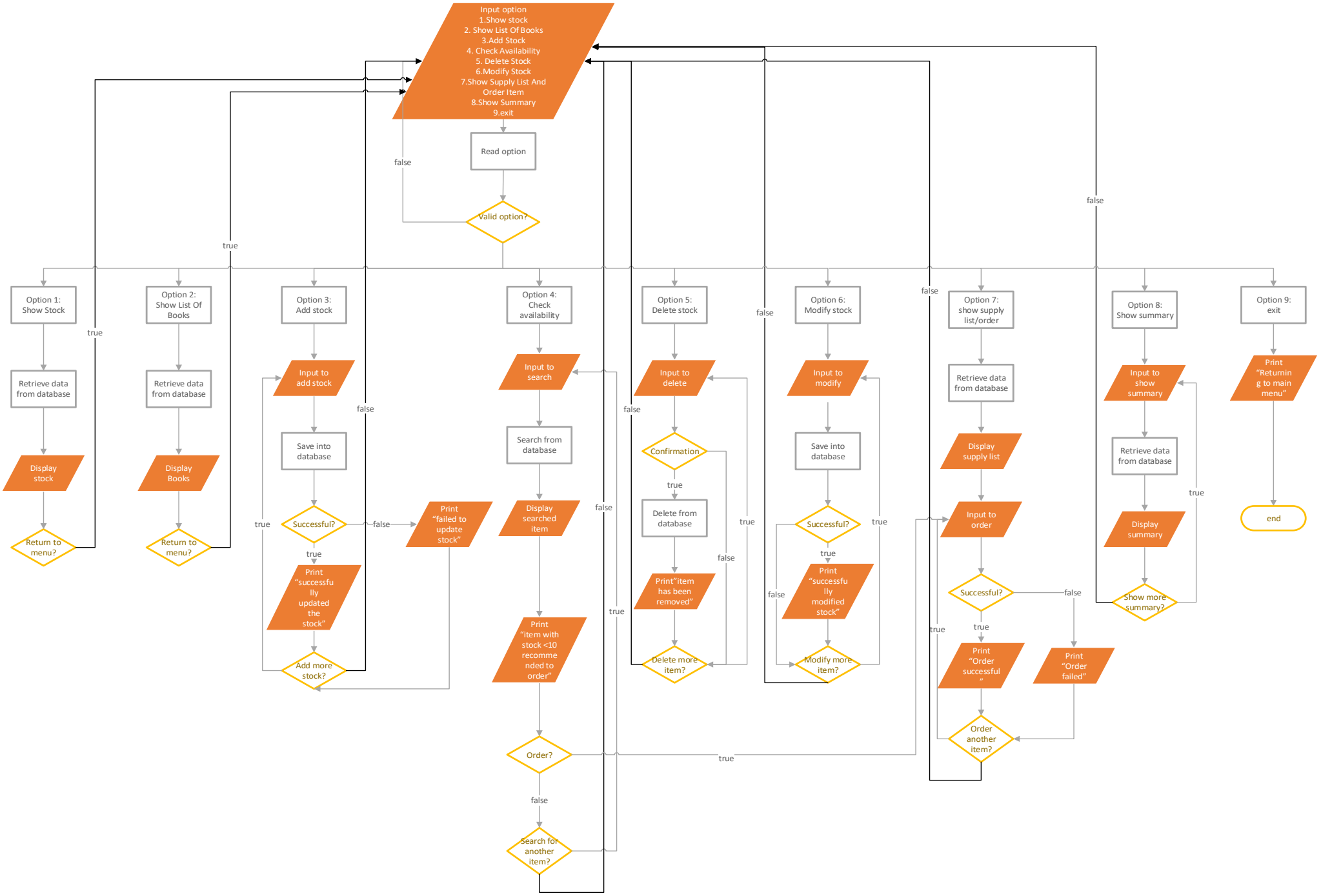


Figure 3.1.1 Flowchart


Input option
1.Show stock
2. Show List Of Books
3.Add Stock
4. Check Availability
5. Delete Stock
6.Modify Stock
7.Show Supply List And Order Item
8.Show Summary
9.exit
Read option
Valid option?
false
true
Option 1: Show Stock
Retrieve data from database
Display stock
Return to menu?
Option 2: Show List Of Books
Retrieve data from database
Display Books
Return to menu?
Option 3: Add stock
Input to add stock
Save into database
Successful?
Print "successfully updated the stock"
Print "failed to update stock"
Add more stock?
Option 4: Check availability
Input to search
Search from database
Display searched item
Print "item with stock <10 recommended to order"
Order?
Search for another item?
Option 5: Delete stock
Input to delete
Confirmation
Delete from database
Print"item has been removed"
Delete more item?
Option 6: Modify stock
Input to modify
Save into database
Successful?
Print "successfully modified stock"
Modify more item?
Option 7: show supply list/order
Retrieve data from database
Display supply list
Input to order
Successful?
Print "Order successful"
Print "Order failed"
Order another item?
Option 8: Show summary
Input to show summary
Retrieve data from database
Display summary
Show more summary?
Option 9: exit
Print "Returning to main menu"
end

3.2 Pseudo Code

3.2.1 Start Program

1. Start
2. Enter username and password
3. Check user's position
4. Display Main Menu
5. Select option 1-9
6. If option invalid, select again

3.2.2 Show Stock

1. Select query
2. Display data from database
3. Return to main menu

3.2.3 Show List of Books

1. Select query
2. Display data from database
3. Return to main menu

3.2.4 Add Stock

1. Select to add new stock for existing item or new item
2. Input to add stock
3. Insert query
4. Update database
5. Ask if user wanted to add another stock
6. Return to main menu

3.2.5 Check Availability

1. Select to search item by name, category or quantity
2. Input to search item
3. Select query
4. Display data from data base

5. Ask if user wanted to search more item
6. Return to main menu

3.2.6 Delete Stock

1. Select item to be deleted
2. Select query
3. Display item to be deleted from database
4. Confirmation on delete the item
5. Update database
6. Ask if the user wanted to delete more item
7. Return to main menu

3.2.7 Modify Stock

1. Select item to be modified
2. Select query
3. Display item to be modified from database
4. Input to modify the item
5. Insert query
6. Update database
7. Ask if the user wanted to modify more item
8. Return to main menu

3.2.8 Show Supply List and Order

1. Select to show order list or order item
2. If show order list
    2.1 Select query
    2.2 Display data from database
    2.3 Return to supply list and order
3. If order item
    3.1 Select to order item
    3.2 Input to order item

3.3 Insert query

3.4 Update database

3.5 Ask if the user wanted to order more item

3.6 Return to supply list and order

4. Return to main menu

3.2.9 Show Summary

1. Select to show summary
2. Select query
3. Display data from database
4. Return to main menu

3.2.10 Exit

1. Display exit message
2. Exit
3. End

3.3 ERD

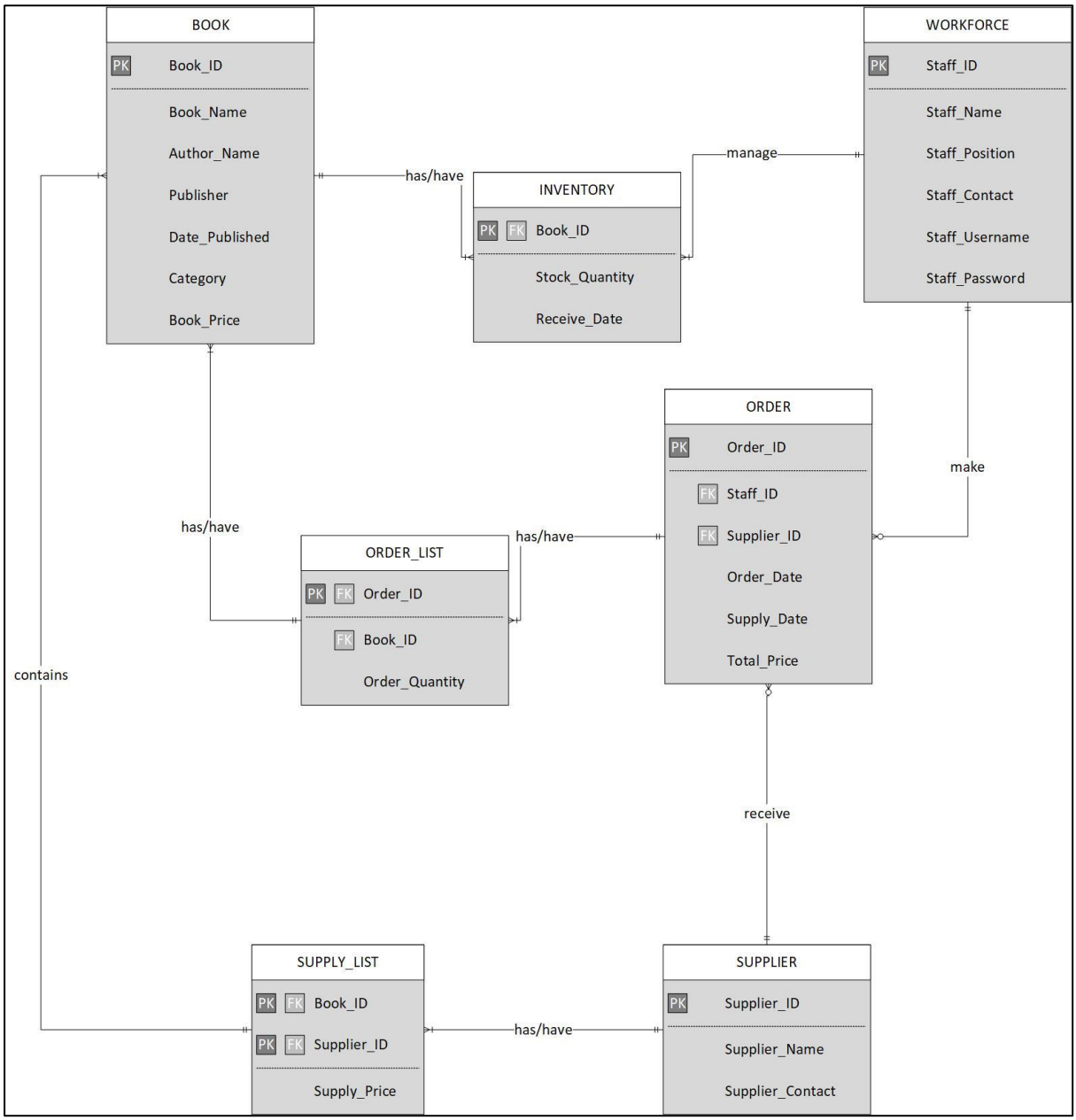


Figure 3.3.1 ERD

Business rule:

1. One workforce can manage one or many inventory of the shop.

2. One workforce can make no order or many order from the supplier.

3. When ordering books from the supplier the price should be more than RM100.

4. The supplier can receive either no order or many order from the shop.

5. An order must have one or many order list.

6. An order list can have one book or many book.

7. A book can have either one inventory or many inventory.

8. A supplier must have a supply list or many supply list which contain price of the item.

9. A supply list contains supply price of one book or many books

3.4 Data Dictionary

Inventory								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Book_ID	Varchar	6	Identity number of book	Y	N32114		PK, FK	Book
Stock_Quantity	int	4	Quantity of particular item	Y				
Receive_Date	Date		Date of the book received from the supplier	Y				

Book								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Book_ID	Varchar	6	Identity number of book	Y	N32114		PK	
Book_Name	Varchar	100	Name of book	Y				
Author_Name	Varchar	100	Name of author	Y				
Publisher	Varchar	100	Name of Pulisher	Y				
Date_Published	Date		Date of the book published	Y				
Category	Varchar	20	Category of the book	Y				
Book_Price	Number	6,2	Price of the book	Y				

Supplier								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Supplier_ID	Varchar	6	Identity number of supplier	Y	P45666		PK	
Supplier_Name	Varchar	100	Name of supplier	Y				
Supplier_Contact	int	11	Contact of supplier	Y	0128081123			

Workforce								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Staff_ID	Varchar	6	Identity number of staff	Y	S45672		PK	
Staff_Name	Varchar	100	Name of the staff	Y				
Staff_Contact	int	11	Contact of staff	Y	0103939412			
Staff_Position	Char	10	Position of the staff	Y				
Staff_Username	Varchar	50	Username of the staff	Y		Y		
Staff_Password	Varchar	50	Password of the staff	Y				

Order								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Order_ID	int	5	Identity number of order	Y			PK	
Supplier_ID	Varchar	6	Identity number of supplier	Y	P45666		FK	Supplier
Staff_ID	Varchar	6	Identity number of staff	Y	S45672		FK	Workforce

Order_Date	Date		Date when making order	Y				
Supply_Date	Date		Date to be filled by supplier when supplying					
Total_Price	Number	6,2	Total price of the item ordered	Y				

Order_List								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Order_ID	int	5	Identity number of order	Y			PK,FK	Order
Book_ID	Varchar	6	Identity number of book	Y	N32114		FK	Book
Order_Quantity	int	4	Quantity of the item ordered	Y				

Supply_List								
Field name	Data type	Field length	Description	Required?	Example	Unique?	PK or FK	FK reference table
Supplier_ID	Varchar	6	Identity number of supplier	Y	P45666		PK,FK	Supplier
Book_ID	Varchar	6	Identity number of book	Y	N32114		PK,FK	Book
Supply_Price	Number	6,2	Price of a particular book from when ordering from supplier	Y				

3.5 Interface design

Login Page:

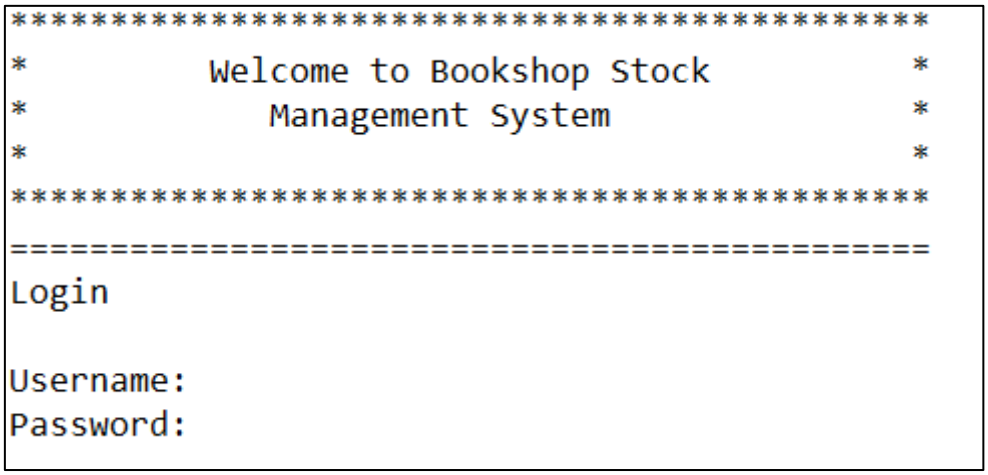


Figure 3.5.1 Interface Design of Login Page

Menu:

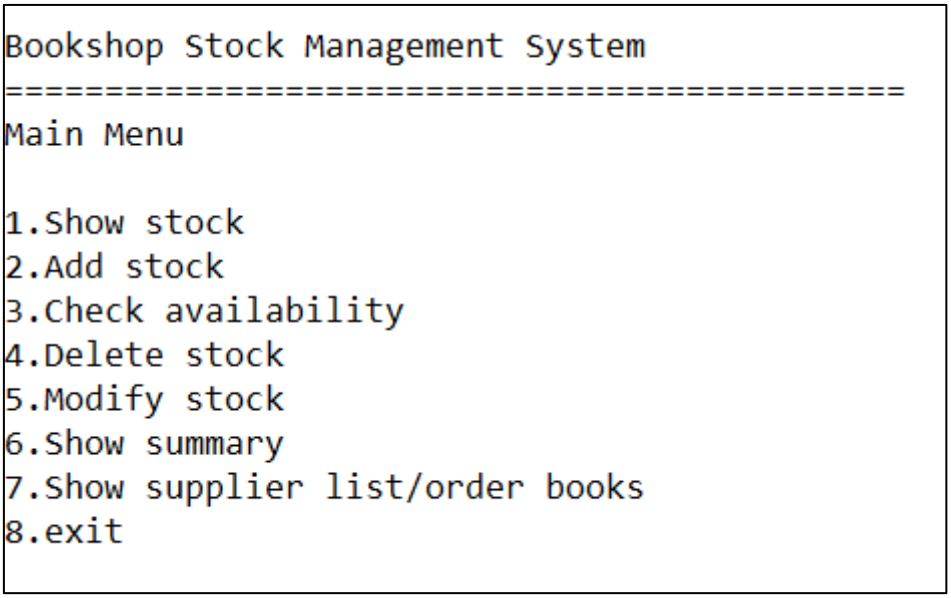


Figure 3.5.2 Interface Design of Menu

Chapter 4: Implementation

4.1 Function

```
21      //function declaration
22      void Connection();
23      void MainMenu();
24      void Login();
25      void AdminMenu();
26      void StaffMenu();
27      void ShowStock();
28      void ShowListOfBooks();
29      void ShowBook();
30      void SupplyList();
31      void AddStock();
32      void CheckAvailability();
33      void DeleteStock();
34      void ModifyStock();
35      void SupplyAndOrder();
36      void ShowSummary();
37      void ColorCode(unsigned short);
38      //End of function declaration
```

Figure 4.1.1 Function Declaration in Global Area

```
58      //Main Function
59      int main()
60      {
61          //connect to database
62          Connection();
63          MainMenu();
64
65          cout << "1. Login " << endl;
66          cout << "2. Exit " << endl;
67          cout << "Please select one option: ";
68          cin >> option;
69
70          switch (option)
71          {
72          case 1:
73              Login();
74              break;
75          case 2:
76              cout << "\n**Exiting program."<<endl;
77                  cout<<"Press Enter to continue...";
78              _getch();
79              exit(0);//exit program
80              break;
81          default:
82              cout << "\n**Please choose between 1 and 2.";
83                  cout<<"Press Enter to continue... ";
84              _getch();//pause console
85              system("cls");//clear screen
86              main();
87              break;
88          }
89      }
90
```

Figure 4.1.2 Example Section that Implements Function

A function in C++ can have a value, a side effect, or both. The side effect occurs before the value is returned. The function's value is the value of the expression in the return statement. A function can be called for its value, its side effect, or both. Figure 4.1.1 is the function declaration of the functions used in this project. It is a way to notify the compiler about a function before a call to the function. From figure 4.1.2, the function Connection() in line 62 is called to connect the database while function MainMenu() in line 63 is called to print interface of main menu. The function has been used for every module throughtout this project.

4.2 Control Technique (Selection)

4.2.1 Switch

```
153     //Admin Menu
154     void AdminMenu()
155     {
156         //admin menu and option to choose
157         system("cls");
158 
159         //print option
160         cout << "Bookshop Stock Management System" << endl;
161         cout << "============================================================" << endl;
162         cout << "Main Menu" << endl;
163         cout << "1. Show Stock" << endl;
164         cout << "2. Show List Of Books" << endl;
165         cout << "3. Add Stock" << endl;
166         cout << "4. Check Availability" << endl;
167         cout << "5. Delete Stock" << endl;
168         cout << "6. Modify Stock" << endl;
169         cout << "7. Show Supply List And Order Item" << endl;
170         cout << "8. Show Summary" << endl;
171         cout << "9. Exit" << endl;
172         cout << "\n Please select an option : ";
173         cin >> option;
174 
175         //option
176         switch(option)
177         {
178             case 1:
179             {
180                 cout << "Showing Stock.\nPress Enter to Continue..." << endl;
181                 _getch();
182                 system("cls");
183                 ShowStock();
184                 break;
185             }
186 
187             case 2:
188             { ... }
195 
196             case 3:
197             { ... }
203 
204             case 4:
205             { ... }
```

Figure 4.2.1 Example Section that Implements Switch

Selection is one of the most common control structure. Selection allows us to choose between two or more alternatives or in other words it enables us to make decisions. In figure 4.2.1, switch has been implemented to enable user to choose option from the list. The user can choose from 1 to 9 from the option and every option have different function. The control technique Switch are used in this project for those function that have multiple option.

4.2.2 if…else Statement

```
104     //login page
105     void Login()
106     {
107         string username;
108         string password;
109
110         cout << "Username: ";
111         cin >> username;
112         cout << "Password: ";
113         cin >> password;
114
115         //select query based on username and password entered
116         string checkUser_query = "SELECT * FROM workforce WHERE Staff_Username = '" + username + "' AND Staff_Password = '" + password + "' ";
117         const char* check = checkUser_query.c_str(); // convert checkUser and return to check
118         qstate = mysql_query(conn, check);//execute query
119
120         if (!qstate)
121         {
122             res = mysql_store_result(conn); //store result
123             if (res->row_count == 1) //found user
124             {
125                 while (row = mysql_fetch_row(res))//loop each row to get data
126                 {
127                     StaffID = row[0]; //store Staff_ID
128                     position = row[3]; //store position
129
130                         if (position == "Manager" || position == "manager")
131                             AdminMenu();
132                         else
133                             StaffMenu();
134                 }
135
136             }
137
```

Figure 4.2.2 Example Section that Implements if…else Statement

If…else statement is also a popular control technique that has been used for many programs. If the expression of the condition is true, the program will execute the statements within if clause otherwise the statement in else clause will be executed. In line 130 to 133 of figure 4.2.2, the section of code will check the position of the user logged in. If the position of the user is manager, function AdminMenu() will be called otherwise StaffMenu() will be called.

4.3 Control Technique (Repetition)

4.3.1 while Loop

```
1048        while (row = mysql_fetch_row(res))//loop each row to get data
1049        {
1050
1051            cout << "Book ID: " << row[0] << endl;
1052            cout << "Book Name: " << row[1] << endl;
1053            cout << "Author Name: " << row[2] << endl;
1054            cout << "Publisher: " << row[3] << endl;
1055            cout << "Date Published: " << row[4] << endl;
1056            cout << "Category: " << row[5] << endl;
1057            cout << "Price: RM " << row[6] << endl;
1058            cout << "Stock Quantity: " << row[7] << endl;
1059            cout << "Date Received: " << row[8] << endl;
1060        }
1061
```

Figure 4.3.1 Example Section that Implements While Loop

Repetition statement also known as looping statement or a loop allows the programmer to specify that a program should repeat an action while some condition remains true. In figure 4.3.1, the while loop will keep on looping until all the data has been retrieved from the database. Once the last row of the data has been reached the loop will exit.

4.3.2 for Loop

```
1624        //Start of table
1625        for (int i = 0; i < 158; i++)
1626        {
1627            cout << "_";
1628        }
1629        cout << endl;
1630        cout << "|" << setw(15) << "Supplier" << setw(15) << "|" << setw(60) << " Cost of Monthly order (RM) " << setw(67) << "|" << endl;
1631
1632        for (int i = 0; i < 155; i++)
1633        {
1634            if (i == 0 || i == 29)
1635            {
1636                cout << "|";
1637            }
1638
1639            cout << "_";
1640        }
1641        cout << "|" << endl;
```

Figure 4.3.2 Example Section that Implements for Loop

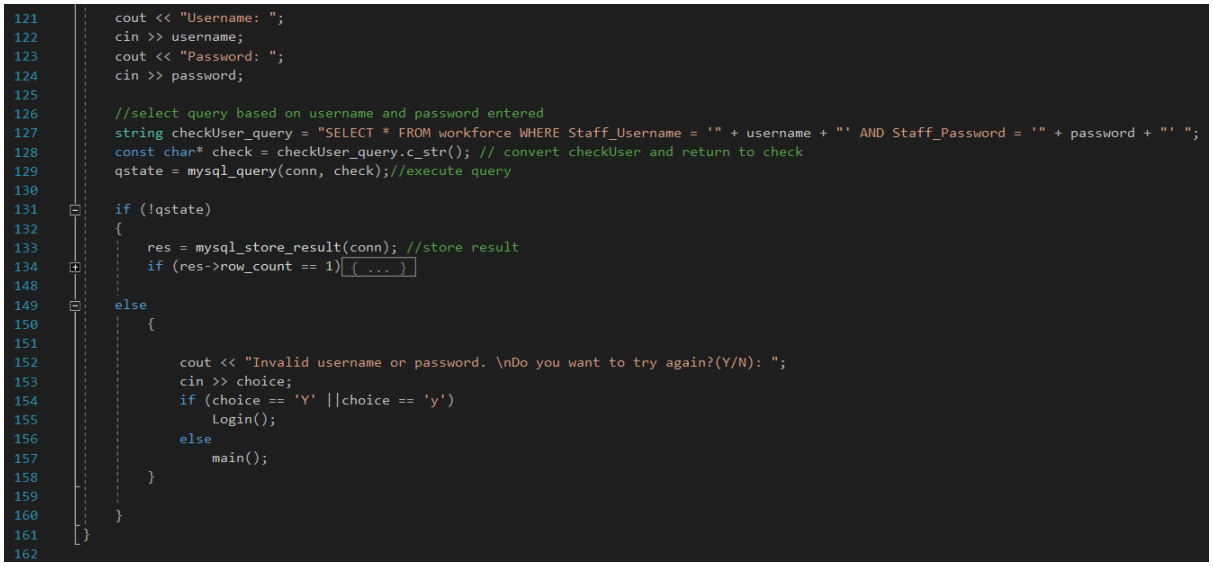
For loop is also one of the technique of repetition, similar to while loop it will repeat the action while the condition remains true. However, for loop requires index so that the loop are able to generate. In figure 4.3.2, it will print the header of a table using for loop. For line 1625 to 1628 the program will loop 157 times before proceeding to the next section of codes.

4.4 Error Handling

```
58      //Main Function
59      int main()
60      {
61          //connect to database
62          Connection();
63          MainMenu();
64
65          cout << "1. Login " << endl;
66          cout << "2. Exit " << endl;
67          cout << "Please select one option: ";
68          cin >> option;
69
70          switch (option)
71          {
72          case 1:
73              Login();
74              break;
75          case 2:
76              cout << "\n**Exiting program."<<endl;
77                  cout<<"Press Enter to continue...";
78              _getch();
79              exit(0);//exit program
80              break;
81          default:
82              cout << "\n**Please choose between 1 and 2.";
83                  cout<<"Press Enter to continue... ";
84              _getch();//pause console
85              system("cls");//clear screen
86              main();
87              break;
88          }
89      }
90
```

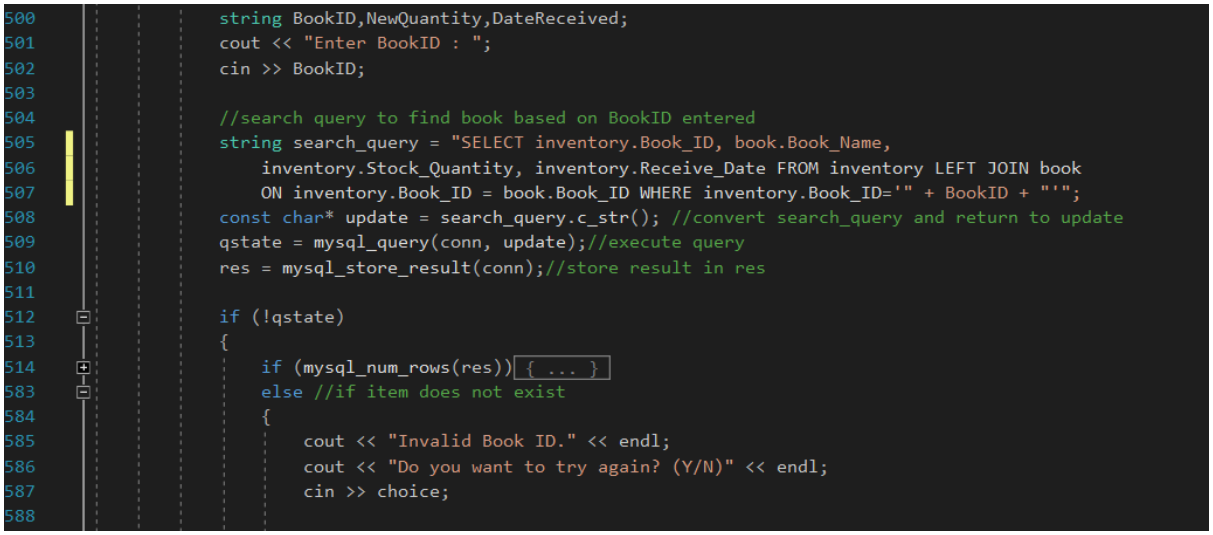
Figure 4.4.1 Example Section that Implement Error Handling

In figure 4.4.1, line 81 to 88 is the default case for switch which act as an error handling for this particular function and is implemented throughout the function that uses switch in this project. For this case, if the user enter option that are not 1 or 2, an error message will be printed and the user are asked to select the options again.



```
121     cout << "Username: ";
122     cin >> username;
123     cout << "Password: ";
124     cin >> password;
125
126     //select query based on username and password entered
127     string checkUser_query = "SELECT * FROM workforce WHERE Staff_Username = '" + username + "' AND Staff_Password = '" + password + "' ";
128     const char* check = checkUser_query.c_str(); // convert checkUser and return to check
129     qstate = mysql_query(conn, check);//execute query
130
131     if (!qstate)
132     {
133         res = mysql_store_result(conn); //store result
134         if (res->row_count == 1){ ... }
148
149     else
150         {
151
152             cout << "Invalid username or password. \nDo you want to try again?(Y/N): ";
153             cin >> choice;
154             if (choice == 'Y' ||choice == 'y')
155                 Login();
156             else
157                 main();
158         }
159
160     }
161 }
162
```

Figure 4.4.2 Example Section that Implements Error Handling



```
500                 string BookID,NewQuantity,DateReceived;
501                 cout << "Enter BookID : ";
502                 cin >> BookID;
503
504                 //search query to find book based on BookID entered
505                 string search_query = "SELECT inventory.Book_ID, book.Book_Name,
506                     inventory.Stock_Quantity, inventory.Receive_Date FROM inventory LEFT JOIN book
507                     ON inventory.Book_ID = book.Book_ID WHERE inventory.Book_ID='" + BookID + "'";
508                 const char* update = search_query.c_str(); //convert search_query and return to update
509                 qstate = mysql_query(conn, update);//execute query
510                 res = mysql_store_result(conn);//store result in res
511
512                 if (!qstate)
513                 {
514                     if (mysql_num_rows(res)){ ... }
583                     else //if item does not exist
584                     {
585                         cout << "Invalid Book ID." << endl;
586                         cout << "Do you want to try again? (Y/N)" << endl;
587                         cin >> choice;
588
```

Figure 4.4.3 Example Section that Implement Error Handling

In figure 4.4.2 and figure 4.4.3, an error handling are implemented in the if…else statement. If the user entered a wrong username or password, the program will display an error message and ask if the user wanted to try again. Similarly, if the user entered an invalid BookID, an error message will be displayed and ask if the user wanted to try again.

```
522             cout << "\nInsert quantity to be added : ";
523             cin >> quantity;
524
525             while (!cin) {
526                 cout << "\nInvalid Input";
527                 cout << "\nEnter quantity to be added : ";
528                 cin.clear();
529                 cin.ignore(256, '\n');
530                 cin >> quantity;
531
532             }
533
```

Figure 4.4.4 Example Section that Implements Error Handling

For figure 4.4.4, this particular section of code is to validate the user input. If the user entered input other than digits, an error message will be displayed and the program will keep on looping until the user entered a valid input.

Chapter 5: Conclusion

5.1 Conclusion

Bookshop stock management system is an attempt to overcome the present in efficient and time consuming in locating data of the items and managing the stocks. It provide a more convenient way of managing the stock rather than a manual system which consumes a lot of human resources and time. It is worth analysing and identifying the benefits as it would directly affect the productivity of the shop.

Unfortunately, due to time constraint and lack of knowledge in programming, the system is not as perfect as those system created by professional programmer and it is possible that some point is uncovered. In future, there will be user friendly graphic user interface for the system which enable the user to have a better experience using this program and the system will be improved to reduce bugs and malfunctions.

Chapter 6: References

1. “Replacing NULL with 0 in a SQL server query” Retrieved from https://stackoverflow.com/questions/16840522/replacing-null-with-0-in-a-sql-server-query
2. Ramakrishnan, R. and Gehrke, J. 2003. The Relational Model In Database Management Systems, 3rdEdition
3. Sin-Ingan, Jurlan Gumapo, Hanna Mae Las Penas, John Fort Neitez, Rodulf.(2019) “Bookstore inventory system documentation, Thesis for Java Programming” Retrieved from https://www.docsity.com/en/bookstore-inventory-system-documentation/5255115/

Appendixes

```
#include<iostream>
#include<iomanip>
#include<string>
#include<mysql.h>
#include<conio.h>

using namespace std;

//global variable declaration
int qstate;
MYSQL* conn;
MYSQL_ROW row;
MYSQL_RES* res;

string StaffID,position;
int option = 0;
char choice;


//function declaration
void Connection();
void MainMenu();
void Login();
void AdminMenu();
void StaffMenu();
void ShowStock();
void ShowListOfBooks();
void ShowBook();
void SupplyList();
void AddStock();
void CheckAvailability();
void DeleteStock();
void ModifyStock();
void SupplyAndOrder();
void ShowSummary();
void ColorCode(unsigned short);
//End of function declaration

//connection to database
 void Connection()
       {
        system("cls");
              conn = mysql_init(0);
              conn = mysql_real_connect(conn, "localhost", "root", "",
              dbworkshop1", 3306, NULL, 0);
              if (conn)
              {
                     cout << "Database Connected To dbworkshop1" << endl;
                     cout << "Press any key to continue..." << endl;
                     _getch();
                     system("cls");

              }
              else
                     cout << "Failed To Connect!" << mysql_errno(conn) << endl;
       }
```

```
//Main Function
int main()
{
	//connect to database
	Connection();
	MainMenu();

	cout << "1. Login " << endl;
	cout << "2. Exit " << endl;
	cout << "Please select one option: ";
	cin >> option;

	switch (option)
	{
	case 1:
		Login();
		break;
	case 2:
		cout << "\n**Exiting program."<<endl;
			cout<<"Press Enter to continue...";
		_getch();
		exit(0);//exit program
		break;
	default:
		if (!cin) {
			cout << "\nInvalid Option";
			cout << "\n**Please choose between 1 and 2." << endl;
			cout << "Press Enter to Continue...";
			cin.clear();
			cin.ignore(256, '\n');
			_getch();
			main();

		}
		cout << "\n**Please choose between 1 and 2."<<endl;
			cout<<"Press Enter to continue... ";
		_getch();//pause console
		system("cls");//clear screen
		main();
		break;
	}
}

//Print Interface for Main Menu
void MainMenu()
{

	cout <<"\t\t*************************************************************" <<
endl;
	cout << "\t\t* \t\t Welcome to Bookshop Stock \t\t*"<<endl;
	cout << "\t\t*   \t\t     Management System \t\t\t*" << endl;
	cout<< "\t\t*************************************************************" <<
endl;
	cout << "-------------------------------------------------------------------
------------------------"<<endl;
}


//login page
void Login()
{
	string username;
```

```
	string password;

	cout << "Username: ";
	cin >> username;
	cout << "Password: ";
	cin >> password;

	//select query based on username and password entered
	string checkUser_query = "SELECT * FROM workforce WHERE Staff_Username = '" +
	username + "' AND Staff_Password = '" + password + "' ";
	const char* check = checkUser_query.c_str(); // convert checkUser and return
	to check
	qstate = mysql_query(conn, check);//execute query

	if (!qstate)
	{
		res = mysql_store_result(conn); //store result
		if (res->row_count == 1) //found user
		{
			while (row = mysql_fetch_row(res))//loop each row to get data
			{
				StaffID = row[0]; //store Staff_ID
				position = row[3]; //store position

			 if (position == "Manager" || position == "manager")
					AdminMenu();
				else
					StaffMenu();
			}

		}

	else
		{

			cout << "Invalid username or password. \nDo you want to try
			again?(Y/N): ";
			cin >> choice;
			if (choice == 'Y' ||choice == 'y')
				Login();
			else
				main();
		}

	}
}
//Admin Menu
void AdminMenu()
{
	//admin menu and option to choose
	system("cls");

	//print option
	cout << "Bookshop Stock Management System" << endl;
	cout << "================================================================="
<< endl;
	cout << "Main Menu" << endl<<endl;
	cout << "1. Show Stock" << endl;
	cout << "2. Show List Of Books" << endl;
	cout << "3. Add Stock" << endl;
	cout << "4. Check Availability" << endl;
	cout << "5. Delete Stock" << endl;
```

```
	cout << "6. Modify Stock" << endl;
	cout << "7. Show Supply List And Order Item" << endl;
	cout << "8. Show Summary" << endl;
	cout << "9. Exit" << endl;
	cout << "\n Please select an option : ";
	cin >> option;


	//option
	switch(option)
	{
		case 1:
		{
			cout << "Showing Stock.\nPress Enter to Continue..." << endl;
			_getch();
			system("cls");
			ShowStock();
			break;
		}

		case 2:
		{
			cout << "Showing Book Details.\nPress Enter to Continue..." <<
endl;
			_getch();
			system("cls");
			ShowListOfBooks();
			break;
		}

		case 3:
		{
			system("cls");

			AddStock();
			break;
		}

		case 4:
		{
			system("cls");
			CheckAvailability();
			break;
		}

		case 5:
		{
			system("cls");
			cout << "Bookshop Stock Management System" << endl;
			cout <<
			"============================================================
			===" << endl;
			DeleteStock();
			break;
		}

		case 6:
		{
			system("cls");
			cout << "Bookshop Stock Management System" << endl;
```

```
                        cout <<
                        "============================================================
                        ==="<<endl;
                        ModifyStock();
                        break;
                }

                case 7:
                {
                        system("cls");

                        SupplyAndOrder();
                        break;

                }

                case 8:
                {
                        system("cls");

                        ShowSummary();
                        break;
                }

                case 9:
                {

                        cout << "\n**Exiting program."<<endl;
                        cout << "Press Enter to continue...";
                        _getch();
                        exit(0);//exit program
                        break;
                }

                default:
                {
                        if (!cin) {
                                cout << "\nInvalid Option";
                                cout << "\n**Please Select An OptionFrom 1-9." <<endl;
                                cout << "Press Enter to Continue...";
                                cin.clear();
                                cin.ignore(256, '\n');
                                _getch();
                                AdminMenu();

                        }
                        cout << "\nInvalid Option";
                        cout << "\n**Please Select An Option From 1-9." << endl;
                        cout << "Press Enter to Continue...";
                        _getch();
                        AdminMenu();
                        break;
                }


        }
}
//Staff Menu
void StaffMenu()
{
        //staff menu and option to choose
        system("cls");
```

```
	//print option
	cout << "Bookshop Stock Management System" << endl;
	cout << "================================================================="
	<< endl;
	cout << "Staff Menu" << endl<<endl;
	cout << "1. Show Stock" << endl;
	cout << "2. Show List Of Books" << endl;
	cout << "3. Add Stock" << endl;
	cout << "4. Check Availability" << endl;
	cout << "5. Exit" << endl;
	cout << "\nPlease select an option : ";
	cin >> option;

	//option
	switch (option)
	{
		case 1:
		{
		cout << "Showing Stock.\nPress Enter to Continue..." << endl;
		_getch();
		system("cls");
		ShowStock();
		break;
		}

		case 2:
		{
			cout << "Showing Book Details.\nPress Enter to Continue..." <<
			endl;
			_getch();
			system("cls");
			ShowListOfBooks();
			break;
		}

		case 3:
		{
		system("cls");

		AddStock();
		break;
		}

		case 4:
		{
		system("cls");
		CheckAvailability();
		break;
		}

		case 5:
		{
			cout << "\n**Exiting program."<<endl;
			cout << "Press Enter to continue...";
			_getch();
			exit(0);//exit program
			break;
		}

		default:
		{
```

```
                        if (!cin) {
                                cout << "\nInvalid Option";
                                cout << "\n**Please Select An Option From 1-5." << endl;
                                cout << "Press Enter to Continue...";
                                cin.clear();
                                cin.ignore(256, '\n');
                                _getch();
                                StaffMenu();

                        }
                cout << "\nInvalid Option";
                cout << "\n**Please Select An Option From 1-5." << endl;
                cout << "Press Enter to Continue...";
                _getch();
                StaffMenu();
                break;
                }

        }
}

//Show Stock
void ShowStock()
{
        cout << "Bookshop Stock Management System" << endl;
        cout << "=====================================================================" ;
        cout <<
        "======================================================================"<<endl<<e
        ndl;


                //execute query
              qstate = mysql_query(conn, "SELECT inventory.Book_ID, book.Book_Name,
              book.Category, book.Book_Price, inventory.Stock_Quantity,
              inventory.Receive_Date FROM inventory LEFT JOIN book ON
              inventory.Book_ID = book.Book_ID");

              cout << setw(7) << "Book ID" << setw(40) << "Book Name" << setw(38) <<
              "Category" << setw(13) << "Price" << setw(14) << "Quantity" << setw(16)
              << "Date Received" << endl;

               if (!qstate)
               {
                        res = mysql_store_result(conn); //store result in res
                        while (row = mysql_fetch_row(res))//loop every row to get data
                        {
                         cout << setw(6) << row[0] << setw(60) << row[1] << setw(20) <<
                         row[2] << setw(9) << "RM " << row[3] << setw(10) << row[4] <<
                         setw(17) << row[5] << endl;
                        }
               }

               cout << "\nPress Enter to Return to Main Menu...";
               _getch();
               if (position == "Manager" || position == "manager")
               {
                        system("cls");
                        AdminMenu();
               }
               else
               {
                        system("cls");
```

```
                        StaffMenu();
                }



}

//Show Book Details
void ShowListOfBooks()
{
        cout << "Bookshop Stock Management System" << endl;
        cout << "=====================================================================";
        cout << "=====================================================================";
        cout << "===============";
        cout << "=====================================================================";
        << endl<<endl;


        //execute query
        qstate = mysql_query(conn, "SELECT
        book.*,inventory.Stock_Quantity,inventory.Receive_Date FROM book LEFT JOIN
        inventory ON inventory.Book_ID = book.Book_ID");

        cout << setw(7) << "Book ID" << setw(40) << "Book Name" << setw(38) <<
        "Author Name" << setw(30) << "Publisher" << setw(20) << "Date Published" <<
        setw(18) << "Category" <<
        setw(17)<<"Price"<<setw(17)<<"Quantity"<<setw(20)<<"Date Received "<<endl;
        if (!qstate)
        {
                res = mysql_store_result(conn);
                        while (row = mysql_fetch_row(res))//loop every row to get data
                        {
                          cout << setw(6) << row[0] << setw(60) << row[1] << setw(25) <<
                          row[2] << setw(25) << row[3] << setw(17) << row[4] << setw(20)
                          << row[5] << setw(12)<<"RM
                          "<<row[6]<<setw(15)<<row[7]<<setw(18)<<row[8]<<endl;
                        }
        }

        cout << "\nPress Enter to Return to Main Menu...";
        _getch();
        if (position == "Manager" || position == "manager")
        {
                system("cls");
                AdminMenu();
        }
        else
        {
                system("cls");
                StaffMenu();
        }


}
//to be displayed in ModifyStock and DeleteStock
void ShowBook()
{
        cout << "Bookshop Stock Management System" << endl;
        cout << "=====================================================================";
        cout << "=====================================================================";
        cout << "===============";
```

```
        cout << "=================================================================="
<< endl<<endl;


        //execute query
        qstate = mysql_query(conn, "SELECT
        book.*,inventory.Stock_Quantity,inventory.Receive_Date FROM book LEFT JOIN
        inventory ON inventory.Book_ID = book.Book_ID");

        cout << setw(7) << "Book ID" << setw(40) << "Book Name" << setw(38) <<
        "Author Name" << setw(30) << "Publisher" << setw(20) << "Date Published" <<
        setw(18) << "Category" << setw(17) << "Price" << setw(17) << "Quantity" <<
        setw(20) << "Date Received " << endl;
        if (!qstate)
        {
                res = mysql_store_result(conn);
                while (row = mysql_fetch_row(res))//loop every row to get data
                {
                  cout << setw(6) << row[0] << setw(60) << row[1] << setw(25) << row[2]
                  << setw(25) << row[3] << setw(17) << row[4] << setw(20) << row[5] <<
                  setw(12) << "RM " << row[6] << setw(15) << row[7] << setw(18) <<
                  row[8] << endl;
                }
        }
}

//to be displayed in SupplyAndOrder
void SupplyList()
{
        cout << "Bookshop Stock Management System" << endl;
        cout << "==================================================================";
        cout << "==================================================================";
        cout << "===============";
        cout << "=================================================================="
      << endl<<endl;

        qstate =mysql_query(conn, "SELECT
        supply_list.Supplier_ID,supplier.Supplier_Name,supply_list.Book_ID,book.Book_
        Name,supply_list.Supply_Price FROM supply_list LEFT JOIN book ON
        supply_list.Book_ID = book.Book_ID LEFT JOIN supplier ON
        supply_list.Supplier_ID= supplier.Supplier_ID");
        res = mysql_store_result(conn);

        if (!qstate)
        {
           cout << setw(11) << "Supplier ID" << setw(30) << "Supplier Name" <<
           setw(17) << "Book ID" << setw(40) << "Book Name" << setw(40) << "Supply
           Price"<<endl;

                while (row = mysql_fetch_row(res))
                {
                   cout << setw(8) << row[0] << setw(40) << row[1] << setw(10) <<
                   row[2] << setw(60) << row[3] << setw(12) << "RM " << row[4]<<endl;
                }
        }
}

//Add Stock
void AddStock()
{
        int supplyprice = 0;
        int price = 0;
```

```
	int quantity = 0;
	int stock = 0;

	cout << "Bookshop Stock Management System" << endl;
	cout << "================================================================";

	//print option
	cout << "\nAdd Stock :- "<<endl<<endl;
	cout << "1. Add Stock for Existing Book"<<endl;
	cout << "2. Add Stock for New Book" << endl;
	cout << "3. Return to Main Menu" << endl;
	cout << "\nSelect An Option : ";
	cin >> option;

	//option to choose
	switch (option)
	{

		case 1:
		{
			string BookID,NewQuantity,DateReceived;
			cout << "Enter BookID : ";
			cin >> BookID;

			//search query to find book based on BookID entered
		       string search_query = "SELECT inventory.Book_ID, book.Book_Name,
		       inventory.Stock_Quantity, inventory.Receive_Date FROM inventory
		       LEFT JOIN book ON inventory.Book_ID = book.Book_ID WHERE
		       inventory.Book_ID='" + BookID + "'";
		       const char* update = search_query.c_str(); //convert
		       search_query and return to update
			qstate = mysql_query(conn, update);//execute query
			res = mysql_store_result(conn);//store result in res

			if (!qstate)
			{
			   if (mysql_num_rows(res)) //check if the BookID entered exist
			   in database
			       {
					row = mysql_fetch_row(res); //store data in row
					cout << "\nBook ID : " << row[0] << endl;
					cout << "Book Name : " << row[1] << endl;
					cout << "Quantity : " << row[2] << endl;
					cout << "Date Received : " << row[3] << endl;

					stock = stoi(row[2]);//convert string to
				    integer,stock now store result of Stock_Quantity

					cout << "\nInsert quantity to be added : ";
					cin >> quantity;

					while (!cin) {
					       cout << "\nInvalid Input";
					       cout << "\nInsert quantity to be added : ";
					       cin.clear();
					       cin.ignore(256, '\n');
					       cin >> quantity;

					}

				   NewQuantity = to_string(stock + quantity);
				   //calculate new quantity
```

```
                                string update_query = "UPDATE inventory SET
                                Stock_Quantity='" + NewQuantity + "' WHERE
                                Book_ID='" + BookID + "'"; //update query

                                 const char* StockUpdate =
                                 update_query.c_str();//convert update_query and
                                 return to StockUpdate

                                 qstate = mysql_query(conn, StockUpdate);//execute
                                 query

                                cout << "Insert new date received(YYYY-MM-DD) : ";
                                 cin.ignore(1, '\n');
                                 getline(cin, DateReceived);

                                 string update1_query = "UPDATE inventory SET
                                 Receive_Date='" + DateReceived + "' WHERE
                                 Book_ID='" + BookID + "'";//update query

                                 const char* DateUpdate =
                                 update1_query.c_str();//convert update1_query and
                                 return to DateUpdate
                                 qstate = mysql_query(conn, DateUpdate);//execute
                                 query

                                 if (!qstate)//notify user if query able/unable to
                                 run
                                cout << "Successfully updated the stock!" << endl;
                                 else
                                 cout << "Failed to update the stock!" << endl;

                                 //Prompt user to add more stock or return
                                 cout << "\nDo you want to add another
                                 stock?(Y/N): ";
                                 cin >> choice;
                                 if (choice == 'Y' || choice == 'y')
                                 {
                                        system("cls");
                                        AddStock();
                                 }
                                 else
                                 {
                                        //check position of the user and return to
the menu
                                        if (position == "Manager" || position ==
"manager")
                                        {
                                               cout << "Returning to Main Menu."
<< endl;
                                               cout << "Press Enter to
Continue..." << endl;
                                               _getch();
                                               system("cls");
                                               AdminMenu();
                                        }
                                        else
                                        {
                                               cout << "Returning to Main Menu."
<< endl;
                                               cout << "Press Enter to
Continue..." << endl;
                                               _getch();
```

```
                                                system("cls");
                                                StaffMenu();
                                        }
                                }
                        }
                        else //if item does not exist
                        {
                                cout << "Invalid Book ID." << endl;
                                cout << "Do you want to try again? (Y/N)" <<
endl;
                                cin >> choice;

                                //prompt user to try again
                                if (choice == 'Y' || choice == 'y')
                                {
                                        system("cls");
                                        AddStock();
                                }
                                else
                                {
                                        system("cls");
                                        AdminMenu();
                                }
                        }

                }
                break;
        }

        case 2:
        {
                string BookID, BookName, AuthorName, Publisher, DatePublished,
Category, Price, Quantity, DateReceived, SupplierID, SupplyPrice;

                //promt user to input new item
                cout << "\nInsert new stock" << endl;
                cin.ignore(1, '\n');
                cout << "Enter BookID : ";
                getline(cin, BookID);


                cout << "Enter Book Name : ";
                getline(cin, BookName);


                cout << "Enter Author Name : ";
                getline(cin, AuthorName);


                cout << "Enter Publisher : ";
                getline(cin, Publisher);


                cout << "Enter DatePublished(YYYY-MM-DD) : ";
                getline(cin, DatePublished);


                cout << "Enter Category : ";
                getline(cin, Category);

                cout << "Enter Price : RM ";
                cin >> price;
```

```
                        while (!cin) {
                                cout << "\nInvalid Input";
                                cout << "Enter Price : RM ";
                                cin.clear();
                                cin.ignore(256, '\n');
                                cin >> quantity;

                        }
                        Price = to_string(price);


                        cout << "Enter Quantity : ";
                        cin >> quantity;

                        while (!cin) {
                                cout << "\nInvalid Input";
                                cout << "\nEnter Quantity : ";
                                cin.clear();
                                cin.ignore(256, '\n');
                                cin >> quantity;

                        }
                        Quantity = to_string(quantity);

                        cin.ignore(1, '\n');
                        cout << "Enter Date Received(YYYY-MM-DD) : ";
                        getline(cin, DateReceived);


                        cout << "Enter Supplier ID: ";
                        getline(cin, SupplierID);


                        cout << "Enter Supply Price: RM ";
                        cin >> supplyprice;

                        while (!cin) {
                                cout << "\nInvalid Input";
                                cout << "\nEnter Supply Price : RM ";
                                cin.clear();
                                cin.ignore(256, '\n');
                                cin >> supplyprice;

                        }
                        SupplyPrice = to_string(supplyprice);

                        //query to insert new items into book, inventory, supply_list
                        string insert_query = "INSERT INTO book (Book_ID, Book_Name,
Author_Name, Publisher, Date_Published, Category, Book_Price) VALUES ('" + BookID +
"','" + BookName + "','" + AuthorName + "','" + Publisher + "','" + DatePublished +
"','" + Category + "','" + Price + "')";
                        string insert2_query = "INSERT INTO
inventory(Book_ID,Stock_Quantity,Receive_Date) VALUES ('" + BookID + "','" +
Quantity + "','" + DateReceived + "')";
                        string insert3_query = "INSERT INTO
supply_list(Supplier_ID,Book_ID,Supply_Price) VALUES ('" + SupplierID + "','" +
BookID + "','" + SupplyPrice + "')";
                        const char* insertBook = insert_query.c_str();//convert
insert_query and return to insertBook
                        const char* insertInventory = insert2_query.c_str();//convert
insert2_query and return to insertInventory
```

```
                        const char* insertSupplyList = insert3_query.c_str();//convert
insert3_query and return to insertSupplyList
                        qstate = mysql_query(conn, insertBook); //execute query
                        qstate = mysql_query(conn, insertInventory); //execute query
                        qstate = mysql_query(conn, insertSupplyList); //execute query

                        //notify user if query able/unable to run
                        if (!qstate)
                                cout << "Item successfully added!" << endl;
                        else
                                cout << "Failed to add item!" << endl;

                        //Promt user to add more stock or return
                        cout << "\nDo you want to add another stock?(Y/N): ";
                        cin >> choice;
                        if (choice == 'Y' || choice == 'y')
                        {
                                system("cls");
                                AddStock();
                        }
                        else
                        {
                                //check position of the user and return to the menu
                                if (position == "Manager" || position == "manager")
                                {
                                        cout << "Returning to Main Menu." << endl;
                                        cout << "Press Enter to Continue..." << endl;
                                        _getch();
                                        system("cls");
                                        AdminMenu();
                                }
                                else
                                {
                                        cout << "Returning to Main Menu." << endl;
                                        cout << "Press Enter to Continue..." << endl;
                                        _getch();
                                        system("cls");
                                        StaffMenu();
                                }
                        }
                        break;

                }

                case 3:
                {
                        //check position of the user and return to the menu
                        if (position == "Manager" || position == "manager")
                        {
                                cout << "Returning to Main Menu." << endl;
                                cout << "Press Enter to Continue..." << endl;
                                _getch();
                                system("cls");
                                AdminMenu();
                        }
                        else
                        {
                                cout << "Returning to Main Menu." << endl;
                                cout << "Press Enter to Continue..." << endl;
                                _getch();
                                system("cls");
                                StaffMenu();
```

```
                        }
                        break;
                }

                default:
                {
                        if (!cin) {
                                cout << "\nInvalid Option";
                                cout << "\n**Please Select An Option From 1-3." << endl;
                                cout << "Press Enter to Continue...";
                                cin.clear();
                                cin.ignore(256, '\n');
                                _getch();
                                system("cls");
                                AddStock();

                        }
                        cout << "\nInvalid Option"<<endl;
                        cout << "Please Select An Option From 1-3." << endl;
                        cout << "Press Enter to Continue..." << endl;
                        _getch();
                        system("cls");
                        AddStock();
                        break;
                }

        }



}
//Search book by particular input
void CheckAvailability()
{
        cout << "Bookshop Stock Management System" << endl;
        cout << "=================================================================";
        string BookName, Category,Quantity;
        int quantity;

        //print option
        cout << "\nSearch item by :-" << endl<<endl;
        cout << "1. Book Name \n2. Category \n3. Stock Quantity  \n4. Return to Main
Menu"<<endl;
        cout << "\nSelect An Option : ";
        cin >> option;

        switch (option)
        {

                case 1:
                {
                                cout << "Enter Book Name to be searched (at least 5
characters): ";
                                cin.ignore(1, '\n');
                                getline(cin, BookName);
                                while (BookName.length() <= 3)
                                {
                                        cout << "Enter Book Name to be searched (at least
5 character): ";
                                        cin.ignore(1, '\n');
                                        getline(cin, BookName);
```

```
                        }

                    cout << "\n";

                    //search query based on BookName
                    string search_query = "SELECT book.Book_ID, book.Book_Name,
book.Category, book.Book_Price, inventory.Stock_Quantity FROM inventory LEFT JOIN
book ON inventory.Book_ID = book.Book_ID WHERE book.Book_Name LIKE '%" + BookName +
"%'";
                    const char* SearchName = search_query.c_str();//convert
search_query and return to SearchName
                    qstate = mysql_query(conn, SearchName);//execute query

                    if (!qstate)
                    {
                        cout << setw(7) << "Book ID" << setw(40) << "Book Name"
<< setw(34) << "Category" << setw(13) << "Price" << setw(14) << "Quantity" << endl;
                        res = mysql_store_result(conn);//store result in res
                        while (row = mysql_fetch_row(res))//loop every row to
get data
                        {
                        cout << setw(6) << row[0] << setw(60) << row[1] <<
setw(16) << row[2] << setw(9) << "RM " << row[3] << setw(10) << row[4] << endl;

                        }

                        //Prompt user to add more stock or return
                            cout << "\nDo you want to search another
stock?(Y/N)";

                            cin >> choice;
                            if (choice == 'Y' || choice == 'y')
                            {
                                system("cls");
                                CheckAvailability();
            }
                            else
                            {
                                //check position of the user and return to
the menu

                                if (position == "Manager" || position ==
"manager")

                                {
                                    cout << "Returning to Main Menu."
<< endl;

                                    cout << "Press Enter to
Continue..." << endl;

                                    _getch();
                                    system("cls");
                                    AdminMenu();
                                }
                                else
                                {
                                    cout << "Returning to Main Menu."
<< endl;

                                    cout << "Press Enter to
Continue..." << endl;

                                    _getch();
                                    system("cls");
                                    StaffMenu();
                                }
                            }
```

```
                        }
                        break;



                }

                case 2:
                {
                        cout << "Enter Category to be searched (at least 3 character):
";
                        cin.ignore(1, '\n');
                        getline(cin, Category);

                        while(Category.length()<2){
                                cout << "Enter Category to be searched (at least 3
character): ";
                                cin.ignore(1, '\n');
                                getline(cin, Category);

                        }
                        cout << "\n";

                        //search query based on Category
                        string search_query= "SELECT book.Book_ID, book.Book_Name,
book.Category, book.Book_Price, inventory.Stock_Quantity FROM inventory LEFT JOIN
book ON inventory.Book_ID = book.Book_ID WHERE book.Category LIKE '%" + Category +
"%'";
                        const char* SearchCategory = search_query.c_str();//convert
search_query and return to SearchCategory
                        qstate = mysql_query(conn, SearchCategory);//execute query

                        if (!qstate)
                        {
                                cout << setw(7) << "Book ID" << setw(40) << "Book Name"
<< setw(38) << "Category" << setw(13) << "Price" << setw(14) << "Quantity" << endl;
                                res = mysql_store_result(conn);//store result in res
                                while (row = mysql_fetch_row(res))//loop every row to
get data
                                {
                                        cout << setw(6) << row[0] << setw(60) << row[1]
<< setw(18) << row[2] << setw(9) << "RM " << row[3] << setw(10) << row[4] << endl;

                                }
                                //Prompt user to add more stock or return
                                cout << "\nDo you want to search another stock?(Y/N)";
                                cin >> choice;
                                if (choice == 'Y' || choice == 'y')
                                {
                                        system("cls");
                                        CheckAvailability();
                                }
                                else
                                {
```

```
                                        //check position of the user and return to the
menu
                                        if (position == "Manager" || position ==
"manager")
                                        {
                                                cout << "Returning to Main Menu." << endl;
                                                cout << "Press Enter to Continue..." <<
endl;
                                                _getch();
                                                system("cls");
                                                AdminMenu();
                                        }
                                        else
                                        {
                                                cout << "Returning to Main Menu." << endl;
                                                cout << "Press Enter to Continue..." <<
endl;
                                                _getch();
                                                system("cls");
                                                StaffMenu();
                                        }
                                }

                        }
                        break;
                }

                case 3:
                {
                                cout << "Enter Stock Quantity to be searched : <";
                                cin >> quantity;

                                //keep looping if input error
                                while (!cin) {
                                        cout << "\nInvalid Input"<<endl;
                                        cout << "Enter Stock Quantity to be searched :
<";
                                        cin.clear();
                                        cin.ignore(256, '\n');
                                        cin >> quantity;

                                }

                        Quantity=to_string(quantity);
                        cout <<"\n";
                        string search_query= "SELECT book.Book_ID, book.Book_Name,
book.Category, book.Book_Price, inventory.Stock_Quantity FROM inventory LEFT JOIN
book ON inventory.Book_ID = book.Book_ID WHERE inventory.Stock_Quantity <='"+
Quantity +"' ORDER BY inventory.Stock_Quantity ASC";
                        const char* SearchQuantity = search_query.c_str();
                        qstate = mysql_query(conn, SearchQuantity);

                        if (!qstate)
                        {
                                cout << setw(7) << "Book ID" << setw(40) << "Book Name"
<< setw(38) << "Category" << setw(13) << "Price" << setw(14) << "Quantity" << endl;
                                res = mysql_store_result(conn);//store result in res
                                while (row = mysql_fetch_row(res))//loop every row to
get data
                                {
                                        cout << setw(6) << row[0] << setw(60) << row[1]
<< setw(18) << row[2] << setw(9) << "RM " << row[3] << setw(10) << row[4] << endl;
```

```
                        }

                        cout << "\n**Items with stock quantity less than 10 are
recommended to order." << endl;

                        if (position == "Manager" || position ==
"manager")//check position=manager,only manager can order item
                        {
                                cout << "\nDo you want to order?(Y/N): ";
                                cin>>choice;
                                if (choice == 'Y' || choice=='y')
                                {
                                        system("cls");
                                        SupplyAndOrder();
                                }
                                else
                                {
                                        cout << "\nDo you want to search for
another item?(Y/N): ";
                                        cin >> choice;
                                        if (choice == 'Y' || choice == 'y')
                                        {
                                                system("cls");
                                                CheckAvailability();

                                        }
                                        else
                                        {
                                                cout << "Returning to Main Menu."
<< endl;
                                                cout << "Press Enter to
Continue..." << endl;
                                                _getch();
                                                system("cls");
                                                AdminMenu();
                                        }
                                }
                        }
                        else
                        {
                                cout << "Returning to Main Menu." << endl;
                                cout << "Press Enter to Continue..." << endl;
                                _getch();
                                system("cls");
                                StaffMenu();
                        }




                }
                break;

        }

        case 4:
        {
                //check position of the user and return to the menu
                if (position == "Manager" || position == "manager")
                {
```

```
                        cout << "Returning to Main Menu." << endl;
                        cout << "Press Enter to Continue..." << endl;
                        _getch();
                        system("cls");
                        AdminMenu();
                }
                else
                {
                        cout << "Returning to Main Menu." << endl;
                        cout << "Press Enter to Continue..." << endl;
                        _getch();
                        system("cls");
                        StaffMenu();
                }
                break;
        }

        default:
        {
                if (!cin) {
                        cout << "\nInvalid Option";
                        cout << "\n**Please Select An Option From 1-4." << endl;
                        cout << "Press Enter to Continue...";
                        cin.clear();
                        cin.ignore(256, '\n');
                        _getch();
                        system("cls");
                        CheckAvailability();

                }
                cout << "\nInvalid Option"<<endl;
                cout << "Please Select An Option From 1-4." << endl;
                cout << "Press Enter to Continue..." << endl;
                _getch();
                system("cls");
                CheckAvailability();
                break;
        }
    }
}
//Delete Stock
void DeleteStock()
{

    string BookID;
    int quantity;
    system("cls");
    ShowBook();
    cout << "\nPlease Enter Book ID of the item to be deleted: ";
    cin >> BookID;

    //search query based on BookID
    string search_query = "SELECT
book.*,inventory.Stock_Quantity,inventory.Receive_Date FROM book LEFT JOIN inventory
ON inventory.Book_ID = book.Book_ID WHERE book.Book_ID='" + BookID + "'";
    const char* SearchBook = search_query.c_str();//convert search_query,return
to SearchBook
    qstate = mysql_query(conn, SearchBook);//execute query
    res = mysql_store_result(conn);//store result

    if (!qstate)
    {
```

```
		if (mysql_num_rows(res))//check if the BookID exist
		{
			system("cls");
			cout << "Bookshop Stock Management System" << endl;
			cout <<
"================================================================";
			cout << "\nItem to be deleted: " << endl<<endl;

			while (row = mysql_fetch_row(res))//loop each row to get data
			{

				cout << "Book ID: " << row[0] << endl;
				cout << "Book Name: " << row[1] << endl;
				cout << "Author Name: " << row[2] << endl;
				cout << "Publisher: " << row[3] << endl;
				cout << "Date Published: " << row[4] << endl;
				cout << "Category: " << row[5] << endl;
				cout << "Price: RM " << row[6] << endl;
				cout << "Stock Quantity: " << row[7] << endl;
				cout << "Date Received: " << row[8] << endl;
				quantity = stoi(row[7]);
			}

			if (quantity != 0) {

				cout << "\nItem quantity is not 0, item cannot be
deleted!"<<endl<<endl;
				cout << "Returning to Main Menu." << endl;
				cout << "Press Enter to Continue..." << endl;
				_getch();
				system("cls");
				AdminMenu();

			}
			else {
				cout << "\nAre you sure to delete this item?(Y/N): ";
				cin >> choice;
				if (choice == 'Y' || choice == 'y')
				{
					string delete_query = "DELETE FROM book WHERE
Book_ID='" + BookID + "'";
					const char* DeleteBook = delete_query.c_str();
					qstate = mysql_query(conn, DeleteBook);

					if (!qstate)
					{
						cout << "\nItem has been deleted." <<
endl;
						cout << "Do you want to delete another
item?(Y/N): ";
						cin >> choice;

						if (choice == 'Y' || choice == 'y')
						{

							DeleteStock();
						}
						else
						{
							cout << "\nReturning to Main Menu."
<< endl;
```

```
                                                cout << "Press Enter to
Continue..." << endl;
                                                _getch();
                                                system("cls");
                                                AdminMenu();

                                        }
                                }


                        }
                        else
                        {
                                cout << "\nReturning to Main Menu." << endl;
                                cout << "Press Enter to Continue..." << endl;
                                _getch();
                                system("cls");
                                AdminMenu();

                        }
                }



        }
        else //if item does not exist
        {
                cout << "Invalid Book ID." << endl;
                cout << "Do you want to try again? (Y/N)" << endl;
                cin >> choice;

                //prompt user to try again
                if (choice == 'Y' || choice == 'y')
                {
                        system("cls");
                        DeleteStock();
                }
                else
                {
                        system("cls");
                        AdminMenu();
                }
        }

}



}
//Modify Stock
void ModifyStock()
{
        int quantity;
        string
BookID,NBookID,BookName,AuthorName,Publisher,DatePublished,Category,Price,Quantity,D
ateReceived;

        system("cls");
        ShowBook();
        cout << "\nEnter Book ID  of the item to be edited: ";
        cin >> BookID;
```

```
        do
        {
        //search query based on BookID
        string search_query = "SELECT
book.*,inventory.Stock_Quantity,inventory.Receive_Date FROM book LEFT JOIN inventory
ON inventory.Book_ID = book.Book_ID WHERE book.Book_ID='" + BookID + "'";
        const char* SearchBook = search_query.c_str();//convert search_query,return
to SearchBook
        qstate = mysql_query(conn, SearchBook);//execute query
        res = mysql_store_result(conn);//store result

        if (!qstate)
        {


              if (mysql_num_rows(res))//check if the BookID exist
              {

                                system("cls");
                                cout << "Bookshop Stock Management System" << endl;
                                cout <<
"================================================================" << endl;
                                cout << "Item to be edited: " << endl << endl;

                                while(row=mysql_fetch_row(res))//loop each row to get
data
                                {

                                cout << "1. Book ID: " << row[0]<<endl;
                                cout << "2. Book Name: " << row[1] << endl;
                                cout << "3. Author Name: " << row[2] << endl;
                                cout << "4. Publisher: " << row[3] << endl;
                                cout << "5. Date Published: " << row[4] << endl;
                                cout << "6. Category: " << row[5] << endl;
                                cout << "7. Price: RM " << row[6] << endl;
                                cout << "8. Stock Quantity: " << row[7] << endl;
                                cout << "9. Date Received: " << row[8] << endl;
                                cout << "0. Return to Main Menu"<<endl;

                                }

                                cout << "\nSelect an option to edit: ";
                                cin >> option;

                                switch (option)
                                {
                                case 1 :
                                {
                                cout << "Enter New Book ID: ";
                                cin >> NBookID;

                                //update Book_ID
                                string update_query = "UPDATE book SET Book_ID='" +
NBookID + "'WHERE Book_ID='" + BookID + "'";
                                const char* UpdateBookID =
update_query.c_str();//convert update_query, return to UpdateBookID
                                mysql_query(conn, UpdateBookID);//execute query
                                break;
                        }

                                case 2:
                                {
```

```
                            cout << "Enter New Book Name: ";
                            cin.ignore(1,'\n');
                            getline(cin, BookName);

                            //update Book_Name
                            string update_query = "UPDATE book SET Book_Name='" +
BookName + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdateBookName =
update_query.c_str();//convert update_query, return to UpdateBookName
                            mysql_query(conn, UpdateBookName);//execute query
                            break;
                    }

                            case 3:
                            {
                            cout << "Enter New Author Name: ";
                            cin.ignore(1, '\n');
                            getline(cin, AuthorName);;

                            //update Author_Name
                            string update_query = "UPDATE book SET Author_Name='" +
AuthorName + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdateAuthorName =
update_query.c_str();//convert update_query, return to UpdateAuthorName
                            mysql_query(conn, UpdateAuthorName);//execute query
                            break;
                    }

                            case 4:
                            {
                            cout << "Enter New Publisher: ";
                            cin.ignore(1, '\n');
                            getline(cin, Publisher);;

                            //update Publisher
                            string update_query = "UPDATE book SET Publisher='" +
Publisher + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdatePublisher =
update_query.c_str();//convert update_query, return to UpdatePublisher
                            mysql_query(conn, UpdatePublisher);//execute query
                            break;
                    }

                            case 5:
                            {
                            cout << "Enter New Date Published(YYYY-MM-DD): ";
                            cin.ignore(1, '\n');
                            getline(cin, DatePublished);

                            //update Date_Published
                            string update_query = "UPDATE book SET Date_Published='"
+ DatePublished + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdateDatePublished =
update_query.c_str();//convert update_query, return to UpdateDatePublished
                            mysql_query(conn, UpdateDatePublished);//execute query
                            break;
                    }

                            case 6:
                            {
                            cout << "Enter New Category: ";
                            cin.ignore(1, '\n');
```

```
                            getline(cin, Category);

                            //update Category
                            string update_query = "UPDATE book SET Category='" +
Category + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdateCategory =
update_query.c_str();//convert update_query, return to UpdateCategory
                            mysql_query(conn, UpdateCategory);//execute query
                            break;
                    }

                            case 7:
                            {
                            cout << "Enter New Price: RM ";
                            cin.ignore(1, '\n');
                            getline(cin, Price);

                            //update Book_Price
                            string update_query = "UPDATE book SET Book_Price='" +
Price + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdatePrice = update_query.c_str();//convert
update_query, return to UpdatePrice
                            mysql_query(conn, UpdatePrice);//execute query
                            break;
                    }

                            case 8:
                            {

                            cout << "Enter New Quantity: ";
                            cin >> quantity;

                            //keep looping if error input
                            while (!cin){

                                        cout << "\nInvalid Input" << endl;
                                        cin.clear();
                                        cin.ignore(256, '\n');
                                        cout << "Enter New Quantity: ";
                                        cin >> quantity;

                                }

                            Quantity = to_string(quantity);

                            //update Stock_Quantity
                            string update_query = "UPDATE inventory SET
Stock_Quantity='" + Quantity + "'WHERE Book_ID='" + BookID + "'";
                            const char* UpdateQuantity =
update_query.c_str();//convert update_query, return to UpdateQuantity
                            mysql_query(conn, UpdateQuantity);//execute query
                            break;
                    }

                            case 9:
                            {
                            cout << "Enter New Date Received(YYYY-MM-DD): ";
                            cin.ignore(1, '\n');
                            getline(cin, DateReceived);

                            //update Date_Received
```

```
                        string update_query = "UPDATE inventory SET
Date_Received='" + DateReceived + "'WHERE Book_ID='" + BookID + "'";
                        const char* UpdateDateReceived =
update_query.c_str();//convert update_query, return to UpdateDateReceived
                        mysql_query(conn, UpdateDateReceived);//execute query
                        break;
                }

                        case 0:
                        {
                        cout << "\nReturning to Main Menu..." << endl;
                        cout << "Do you want to modify another item?(Y/N): ";
                        cin >> choice;

                        if (choice == 'Y' || choice == 'y')
                        {

                               ModifyStock();
                        }
                        else
                        {

                               cout << "Press Enter to Continue..." << endl;
                               _getch();
                               AdminMenu();
                        }
                }

                        default:
                        {

                        cout << "\nInvalid Option"<<endl;
                        cout << "Please Select An Option From 0-9."<<endl;
                        cout << "Press Enter to Continue..." << endl;
                        _getch();
                        }

                }


        }
        else
        {
                cout << "Invalid BookID entered."<<endl;
                cout << "Do you want to try again?(Y/N): ";
                cin >> choice;

                if (choice == 'Y' || choice == 'y')
                {
                        ModifyStock();
                }
                else
                {
                        cout << "Returning to Main Menu." << endl;
                        cout << "Press Enter to Continue..." << endl;
                        _getch();
                        AdminMenu();
                }

        }
```

```
	}
	}while (true);//keep looping if valid input
}

//Make Order
void SupplyAndOrder()
{
	cout << "Bookshop Stock Management System" << endl;
	cout <<
"================================================================="<<endl;
	cout << "1. Show order" << endl;
	cout << "2. Order item " << endl;
	cout << "3. Return to Main Menu " << endl;
	cout << "\nPlease Select An Option : ";
	cin >> option;

	switch (option)
	{
	case 1://show orderlist
	{
		system("cls");
		cout << "Bookshop Stock Management System" << endl;
		cout <<
"=================================================================";
		cout <<
"=================================================================";
		cout <<
"=================================================================" << endl <<
endl;

		//show order list - GROUP_CONCAT--> group ordered item in 1 row
		qstate = mysql_query(conn, "SELECT tblorder.Order_ID,
tblorder.Supplier_ID, tblorder.Staff_ID, GROUP_CONCAT(tblorderlist.Book_ID) AS
'Book_ID', GROUP_CONCAT(tblorderlist.Order_Quantity) AS 'Order_Quantity',
tblorder.Order_Date, tblorder.Supply_Date, Total_Price FROM tblorder JOIN
tblorderlist ON tblorder.Order_ID = tblorderlist.Order_ID GROUP BY
tblorder.Order_ID");
		res = mysql_store_result(conn);

		if (!qstate)
		{
			cout << setw(9) << "Order_ID" << setw(14) << "Supplier_ID" <<
setw(12) << "Staff_ID" << setw(35) << "Book_ID" << setw(27) << "Order_Quantity" <<
setw(19) << "Order_Date" << setw(15) << "Supply_Date" << setw(18) <<
"Total_Price"<<endl;

			while (row = mysql_fetch_row(res))
			{
				cout << setw(9) << row[0] << setw(11) << row[1] <<
setw(14) << row[2] << setw(40) << row[3] << setw(20) << row[4] << setw(21) << row[5]
<< setw(15) << row[6] << setw(11) << "RM " << row[7] << endl;
			}
		}

		cout << "\n*The item with supply date 0000-00-00 is the item that have
not being received yet.";
		cout << "\nPress Enter to Return to Menu...";
		_getch();
		system("cls");
		SupplyAndOrder();
		break;
```

```
	}

	case 2: //order item
	{
		string BookID;
		system("cls");
		SupplyList();
		cout << "\nEnter Book ID : ";
		cin >> BookID;

		//search query based on BookID
		string search_query = "SELECT supply_list.Supplier_ID,supplier.Supplier_Name,supply_list.Book_ID,book.Book_Name,supply_list.Supply_Price FROM supply_list LEFT JOIN book ON supply_list.Book_ID = book.Book_ID LEFT JOIN supplier ON supply_list.Supplier_ID= supplier.Supplier_ID WHERE supply_list.Book_ID='" + BookID + "' ";
		const char* SearchBookID = search_query.c_str();//convert search_query,return to SearchBookID
		qstate = mysql_query(conn, SearchBookID);//execute query
		res = mysql_store_result(conn);//store result in res

		if (!qstate)
		{

			int quantity;
			double price;

			string  OrderID, SupplierID, OrderDate, SupplyDate, TotalPrice, Quantity;

			if (mysql_num_rows(res))//check if item exist
			{
				system("cls");
				row = mysql_fetch_row(res);//get data

				cout << "Bookshop Stock Management System" << endl;
				cout << "================================================================" << endl;
				cout << "Details of the item to be ordered : " << endl;
				cout << "Supplier ID : " << row[0] << endl;
				cout << "Supplier Name : " << row[1] << endl;
				cout << "Book ID : " << row[2] << endl;
				cout << "Book Name : " << row[3] << endl;
				cout << "Supply Price : " << row[4] << endl;

				BookID = row[2];
				price = stoi(row[4]);//convert string to integer
				SupplierID = row[0];


				cout << "Enter quantity to be ordered : ";
				cin >> quantity;

				while (!cin){
				cout << "\nInvalid Input";
				cout << "\nEnter quantity to be ordered : ";
				cin.clear();
				cin.ignore(256, '\n');
				cin >> quantity;

				}
```

```
                        Quantity = to_string(quantity);

                        cout << "Enter OrderDate(YYYY-MM-DD) : ";
                        cin.ignore(1, '\n');
                        getline(cin, OrderDate);



                        TotalPrice = to_string(quantity * price);//calculate
total price


                        string insert_query = "INSERT INTO tblorder (Order_ID,
Supplier_ID, Staff_ID, Order_Date, Supply_Date, Total_Price) VALUES ('NULL','" +
SupplierID + "', '" + StaffID + "','" + OrderDate + "','NULL','" + TotalPrice +
"')";
                        string insert2_query = "INSERT INTO
tblorderlist(Order_ID, Book_ID, Order_Quantity) VALUES ('NULL','" + BookID + "','" +
Quantity + "')";
                        const char* InsertOrder = insert_query.c_str();
                        const char* InsertOrderList = insert2_query.c_str();
                        qstate = mysql_query(conn, InsertOrder);
                        qstate = mysql_query(conn, InsertOrderList);


                        if (!qstate)
                        {
                                cout << "Order successful" << endl;
                                cout << "Do you want to order another item?(Y/N):
";
                                cin >> choice;

                                if (choice == 'Y' || choice == 'y')
                                {
                                        system("cls");
                                        SupplyAndOrder();
                                }
                                else
                                {
                                        cout << "Returning to Main Menu." << endl;
                                        cout << "Press Enter to Continue..." <<
endl;
                                        _getch();
                                        AdminMenu();
                                }

                        }
                        else
                        {
                                cout << "Order Failed" << endl;
                                cout << "Do you want to order another item?(Y/N):
";
                                cin >> choice;

                                if (choice == 'Y' || choice == 'y')
                                {
                                        system("cls");
                                        SupplyAndOrder();
                                }
                                else
                                {
```

```
                                        cout << "Returning to Admin Menu." <<
endl;
                                        cout << "Press Enter to Continue..." <<
endl;
                                        _getch();
                                        AdminMenu();
                                }

                        }



                }
                else
                {
                        cout << "Invalid BookID entered." << endl;
                        cout << "Do you want to try again?(Y/N): ";
                        cin >> choice;

                        if (choice == 'Y' || choice == 'y')
                        {
                                system("cls");
                                SupplyAndOrder();
                        }
                        else
                        {
                                cout << "Returning to Admin Menu." << endl;
                                cout << "Press Enter to Continue..." << endl;
                                _getch();
                                AdminMenu();
                        }

                }


        }
        break;

    }

    case 3:
    {
        cout << "\nReturning to Main Menu." << endl;
        cout << "Press Enter to Continue..." << endl;
        _getch();
        system("cls");
        AdminMenu();
        break;
    }

    default:
    {
        if (!cin) {
                cout << "\nInvalid Option";
                cout << "\n**Please Select An Option From 1-3." << endl;
                cout << "Press Enter to Continue...";
                cin.clear();
                cin.ignore(256, '\n');
                _getch();
                system("cls");
                SupplyAndOrder();
```

```
                }
                cout << "\nInvalid Option"<<endl;
                cout << "Please Select From 1-3" << endl;
                cout << "Press Enter to Continue..." << endl;
                _getch();
                system("cls");
                SupplyAndOrder();
                break;
        }



        }









}


//show summary
void ShowSummary()
{

        cout << "Bookshop Stock Management System" << endl;
        cout << "================================================================="
<< endl;
        cout << "1. Show cost of monthly order from supplier" << endl;
        cout << "2. Show total cost ordered from different supplier" << endl;
        cout << "3. Show profit of each book" << endl;
        cout << "4. Return to Main Menu" << endl;

        cout << "\nPlease Select An Option : ";
        cin >> option;

        switch (option)
        {
        case 1:
        {
                system("cls");
                cout << "Bookshop Stock Management System" << endl;
                cout <<
"=================================================================";
                cout <<
"=================================================================";
                cout <<
"=================================================================" << endl;
                cout << "\n";

                //Start of table
                for (int i = 0; i < 158; i++)
                {
```

```
                        cout << "_";
                }
                cout << endl;
                cout << "|" << setw(15) << "Supplier" << setw(15) << "|" << setw(60)
<< " Cost of Monthly order (RM) " << setw(67) << "|" << endl;

                for (int i = 0; i < 155; i++)
                {
                        if (i == 0 || i == 29)
                        {
                                cout << "|";
                        }

                        cout << "_";
                }
                cout << "|" << endl;

                cout << "|" << setw(30) << "|";
                cout << setw(10) << "January |" << setw(10) << "February |" <<
setw(10) << "March |" << setw(10) << "April |" << setw(10) << "May |" << setw(10) <<
"June |" << setw(10) << "July |" << setw(10) << "August |" << setw(13) << "September
|" << setw(10) << "October |" << setw(12) << "November |" << setw(12) << "December
|" << endl;

                for (int i = 0; i < 144; i++)
                {
                        if (i == 0 || i == 29 || i == 38 || i == 47 || i == 56 || i ==
65 || i == 74 || i == 83 || i == 92 || i == 101 || i == 113 || i == 122 || i == 133)
                        {
                                cout << "|";
                        }



                        cout << "_";
                }
                cout << "|" << endl;

                //query for fetching data-COALESCE(,0)--->set NULL to 0
                qstate = mysql_query(conn, "SELECT Supplier_ID , COALESCE(SUM(CASE
WHEN month = 1 THEN Total_Price END),0) January , COALESCE(SUM(CASE WHEN month = 2
THEN Total_Price END),0) February , COALESCE(SUM(CASE WHEN month = 3 THEN
Total_Price END),0) March , COALESCE(SUM(CASE WHEN month = 4 THEN Total_Price
END),0)April , COALESCE(SUM(CASE WHEN month = 5 THEN Total_Price END),0) May ,
COALESCE(SUM(CASE WHEN month = 6 THEN Total_Price END),0) June , COALESCE(SUM(CASE
WHEN month = 7 THEN Total_Price END),0) July , COALESCE(SUM(CASE WHEN month = 8 THEN
Total_Price END),0) August , COALESCE(SUM(CASE WHEN month = 9 THEN Total_Price
END),0) September , COALESCE(SUM(CASE WHEN month = 10 THEN Total_Price END),0)
October , COALESCE(SUM(CASE WHEN month = 11 THEN Total_Price END),0) November ,
COALESCE(SUM(CASE WHEN month = 12 THEN Total_Price END),0) December FROM (SELECT
tblorder.* , EXTRACT(MONTH FROM Order_Date) month FROM tblorder ) tblorder GROUP BY
Supplier_ID");

                //body table
                if (!qstate)
                {

                        res = mysql_store_result(conn);
                        while (row = mysql_fetch_row(res))
                        {
                                cout << "|"<<setw(21) << row[0]<<setw(9)<<"|" << setw(9)
<< row[1]<<"|" << setw(9) << row[2] <<"|"<< setw(9) << row[3]<<"|" << setw(9) <<
```

```
row[4]<<"|" << setw(9) << row[5]<<"|" << setw(9) << row[6]<<"|" << setw(9) <<
row[7]<<"|" << setw(9) << row[8]<<"|" << setw(12) << row[9] <<"|"<< setw(9) <<
row[10]<<"|" << setw(11) << row[11]<<"|" << setw(11) << row[12] <<"|"<< endl;
                        }


                }
                for (int i = 0; i < 144; i++)
                {
                        if (i == 0 || i == 29 || i == 38 || i == 47 || i == 56 || i ==
65 || i == 74 || i == 83 || i == 92 || i == 101 || i == 113 || i == 122 || i == 133)
                        {
                                cout << "|";
                        }



                        cout << "_";
                }
                cout << "|" << endl;
                //endl of table

                cout << "\n\n";
                cout << "Supplier:" << endl;
                cout << "P42372 - Horizon Books Sdn Bhd" << endl;
                cout << "P69453 - Basheer Graphic Books Sdn Bhd" << endl;
                cout << "P77382 - Metro Book Distributor" << endl;

                cout << "\nPress Enter to Return to Menu";
                _getch();
                system("cls");
                ShowSummary();
                break;
        }

        case 2:
        {
                double TotalPrice;
                int bar = 0;
                system("cls");
                cout << "Bookshop Stock Management System" << endl;
                cout <<
"================================================================" << endl;

                //execute query
                qstate =mysql_query(conn, "SELECT tblorder.Supplier_ID,
supplier.Supplier_Name, SUM(tblorder.Total_Price)AS Total_Price FROM tblorder LEFT
JOIN supplier ON tblorder.Supplier_ID = supplier.Supplier_ID  GROUP BY
supplier.Supplier_Name");

                if (!qstate)
                {
                        res = mysql_store_result(conn);//store result

                        while (row = mysql_fetch_row(res))//fetch data
                        {
                                cout << endl << endl;
                                cout << setw(10) << row[0]<<"  ";//print out Supplier ID
                                TotalPrice = atof(row[2]);//convert string to double
                                bar = TotalPrice / 150;//set 1 bar = RM500
                                ColorCode(255);//change colour of the text to produce
white bar
```

```
                        for (int i = 0; i < bar; i++)//1st loop to produce white
bar
                        {
                               cout << "|";//white bar
                        }
                        cout << endl;

                        ColorCode(7);
                        cout << setw(12)<<"            " ;//space
                        ColorCode(255);

                        for (int i = 0; i < bar; i++)//2nd loop to make white
bar wider
                        {

                                cout << "|";//white bar

                        }

                        ColorCode(7);
                        cout << " RM " <<fixed <<setprecision(2)<< TotalPrice;
                }

                cout << "\n\nX-axis: Total price of the order over the
year"<<endl;
                cout << "Y-axis: Supplier" << endl;
                cout << "-------------------------------------------------------
-------------------------------------------------" << endl;
                cout << "\nLegend:"<<endl;
                cout << "P42372- Horizon Books Sdn Bhd" << endl;
                cout << "P69453- Basheer Graphic Books Sdn Bhd" << endl;
                cout << "P77382- Metro Book Distributor" << endl;


                cout << "\nPress Enter to Return to Menu";
                _getch();
                system("cls");
                ShowSummary();
                break;


        }
        else
        {
                cout << "Query error";
        }
    }

    case 3:
    {

        system("cls");
        cout << "Bookshop Stock Management System" << endl;
        cout <<
"============================================================";
        cout <<
"============================================================";
        cout <<
"============================================================" << endl<<endl;

        //execute query to show profit
```

```
		qstate = mysql_query(conn, "SELECT book.Book_ID,book.Book_Name, book.Book_Price, supply_list.Supply_Price,(book.Book_Price-supply_list.Supply_Price) AS Profit FROM book LEFT JOIN supply_list ON supply_list.Book_ID = book.Book_ID");

		cout << setw(10) << "Book ID" << setw(45) << "Book Name" << setw(40) << "Selling Price Per Unit" << setw(27) << "Supply Price Per Unit" << setw(20) << "Profit Per Unit"<<endl;//add total Profit

		if (!qstate)
		{
			res = mysql_store_result(conn);//store result in res

			while (row = mysql_fetch_row(res))//loop every row to get data
			{
				cout << setw(10) << row[0] << setw(60) << row[1] << setw(15) << "RM "<<row[2] << setw(20)<<"RM " << row[3] << setw(19) <<"RM "<< row[4]<<endl;
			}


		}

		cout << "\nPress Enter to Return to Menu";
		_getch();
		system("cls");
		ShowSummary();
		break;
	}

	case 4:
	{
		cout << "\nReturning to Main Menu." << endl;
		cout << "Press Enter to Continue..." << endl;
		_getch();
		system("cls");
		AdminMenu();
		break;
	}

	default:
	{
		if (!cin) {
			cout << "\nInvalid Option";
			cout << "\n**Please Select An Option From 1-3." << endl;
			cout << "Press Enter to Continue...";
			cin.clear();
			cin.ignore(256, '\n');
			_getch();
			system("cls");
			ShowSummary();

		}
		cout << "\nInvalid Option"<<endl;
		cout << "Please Select From 1-3" << endl;
		cout << "Press Enter to Continue..." << endl;
		_getch();
		system("cls");
		ShowSummary();
		break;
	}
```

```
        }



}

//change colour of the text
void ColorCode(unsigned short color)
{
        HANDLE hcon = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(hcon, color);
}
```