FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

SEMESTER 2 SESSION 2021/2022

BITU 2913 WORKSHOP

REPORT TITLE:

BOOKSHOP MANAGEMENT SYSTEM

SUPERVISOR NAME: TS. DR. NGO HEA CHOON

EVALUATOR NAME: DR.SITI AZIRAH BINTI ISMAI

PREPARED BY:

NO	NAME	MATRIC	SECTION/GROUP
2	NURSYAZA NISA BINTI ARFARIZAL	B032010244	S1G1

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Previously, the admin use manual management to record every book transaction. In addition, the admin needs at least a day to prepare a list of the available books in the bookshop. The main objective of this project is to develop a systemthat is fast and reliable for not only the admin but also other users to perform the task needed. Aside from that, to calculate the book selling in the bookshop. Additionally, to boost the efficiency in time and resources to manage the bookshop. The system also able to eliminate the paper- work in bookshop and at the same time, problems that arise from non-computerized system such as loss of missing files, inconsistency in data and lack of security that is avoidable.

1.2 PROBLEM STATEMENT

Unlinke in previous times, the availability of bookshop management system is known and advanced nowadays. However, the bookshop management system is still not commonly used even on a small bookshop. In managing the bookshop, all management is done manually. This of course, in the era of computerisation, has bring a lot of inconveniences for the admin and customer who are using the system. These inconveniences including:

i. Unsystematic way of purchasing books as customers need to queue in line in the bookshop to buy books.

ii. Lack of space to keep all the the book and purchase records.

iv. Misplacing and loss of documents and records.

v. Time taken consuming to find the records.

vi. Difficult to track and analyses the record.

1.3 BACKGROUND OF THE PROJECT

The background of the project is to develop a Bookshop Management System. This Bookshop Management System includes the management of book management and transaction section management. Bookshop Management System targeted in developing a computerized system which will automate for the daily work of the bookshop by improving the management of the book property so that it is more efficient. Commonly, the book and transaction section management from admin and from other users are simply written down on paper or books and arranged manually. Previously, the admin use manual management to record every book transaction. In addition, the admin needs at least a day to prepare a list of the available books in the bookshop. The main objective of this project is to develop a system that is fast and reliable for not only the admin but also other users to perform the task needed. Aside from that, the system will be use to calculate the book selling in the bookshop. Additionally, to boost the efficiency in time and resources to manage the bookshop. The system also able to eliminate the paper-work in bookshop and at the same time, problems that arise from non-computerized system such as loss of missing files, inconsistency in data and lack of security that is avoidable. The scope of the project is focused on the small shops in managing the bookshop and they are also the admin of the system. In addition, the customers also the user of the system to shop for books such as textbooks and storybooks. Overall, this system is made to help admin in managing bookshop conveniently to those bookshop with non- computerized system.

1.4 OBJECTIVES

The problem that arises when using non computerized system includes:

1. To develop and design convenient computerized system for bookshop system management.
2. To provide a system that can track and easily analyse book transaction.
3. To provide a full analysis of the bookshop management system.

1.5 SCOPES

The scope involved in Bookshop Management System will be focusing on the two parts which are the target user and the modules to be developed. The scope is described as follows:

1. Target User
    a. Administrator
    b. Customer

2. Module to be developed

    a. Login Module

    This module is made to verify the user before getting the user into the database. Admin is the only target user required to login the system using the username and password to perform other modules such as customer registration, add, delete and update book information. If username or password is entered wrongly, there will be an error message display about the unsuccessful attemp to login. This will maintain the authenication and privacy of the admin account.

    b. Registration Module

    Registration module will have only admin user to access it. Admin on behalf of the customer can register. The customer information such as customer name, phone and address will be promptly asked to be fill out before registering it in the system.

    c. Update Module

    This module is developed to edit the book's price and book's stocks in the system. As the price and the stock is always changing within time, the update module will be focusing on these two only. The information about the book details will be display first. Then , the admin can choose to update or not to update (the information will remain the same) the data in the system. As for customer, the update module is used to update the remaining book's stock after the book has been purchased.

d. Delete Module

This module is similarly to update module, it is particularly to be used by the admin which is to remove book information from the database. If the book that want to be remove is not in database, an error message will be display.

e. Admin Module

- Login as Admin
- Add, delete, update for books
- Generate Report

f. Customer Module

- Purchase Book
- View Book

g. Report Module

The report module is to let the admin to do a full analysis of the bookshop management system. The report analysis is divided into 3 which are stock analysis, book analysis and customer analysis. The report will be mainly about the detailed analysis on the purchase book and the customer who purchase the book.

CHAPTER 2: ANALYSIS OF THE PROBLEM

2.0 PROBLEM DESCRIPTION

The problem is in previous times, manual management is being used to record book transaction and this system mainly targeting on the small bookshop owners. In this proposed system, the target also includes the customers. Customers, as the user of the system will be able to view details of the books and purchase them. As the system is computerized, problems such as file lost, file corrupted and time- consuming can be avoided. Hence, it is more convenient and time-saving to users in managing transaction of the books.

2.1 PROBLEM DECOMPOSITION

The problem that arises when using non computerized system includes:

1) File managed is lost and damaged

Non computerized system would result to a file lost and file damaged. This might be because of carelessness or in other possibility it might be because of accidents. For example, sometimes the owners forgot where the file is placed as they did not keep it in their original place. There are also times when accidents happened such as the ink spilled or the paper accidentally torn which resulting to file damaged.

2) Time and space-consuming

When no computerized system is implemented, the time consuming to do bookshop activities will become much longer. The admins will need time to search for book's information to record the transaction done. Besides that, it is also space-consuming as the physical record was getting more and more after a really long time and place is needed to store the data.

3) Difficult to track and analyses record

Tracking and analyses record manually can be difficult. This is because one or more files need to be use or had accessed to find information required. A heavier workload and more resources needed to generate report, reducing the efficiency working.

2.1 STRUCTURE CHART

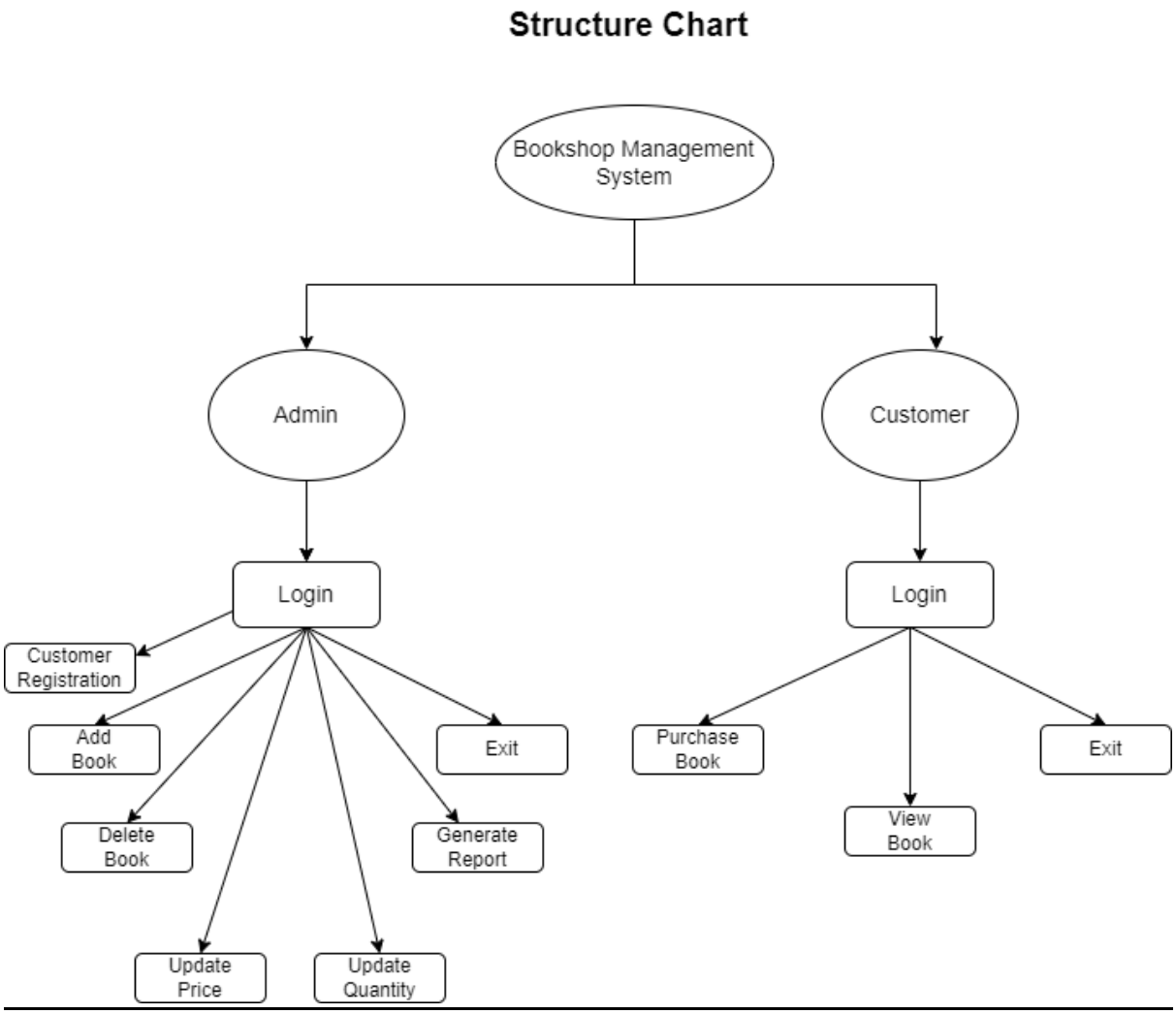

Figure 2.1: Structure Chart Bookshop Management System

CHAPTER 3: DESIGN

3.1 FLOWCHART

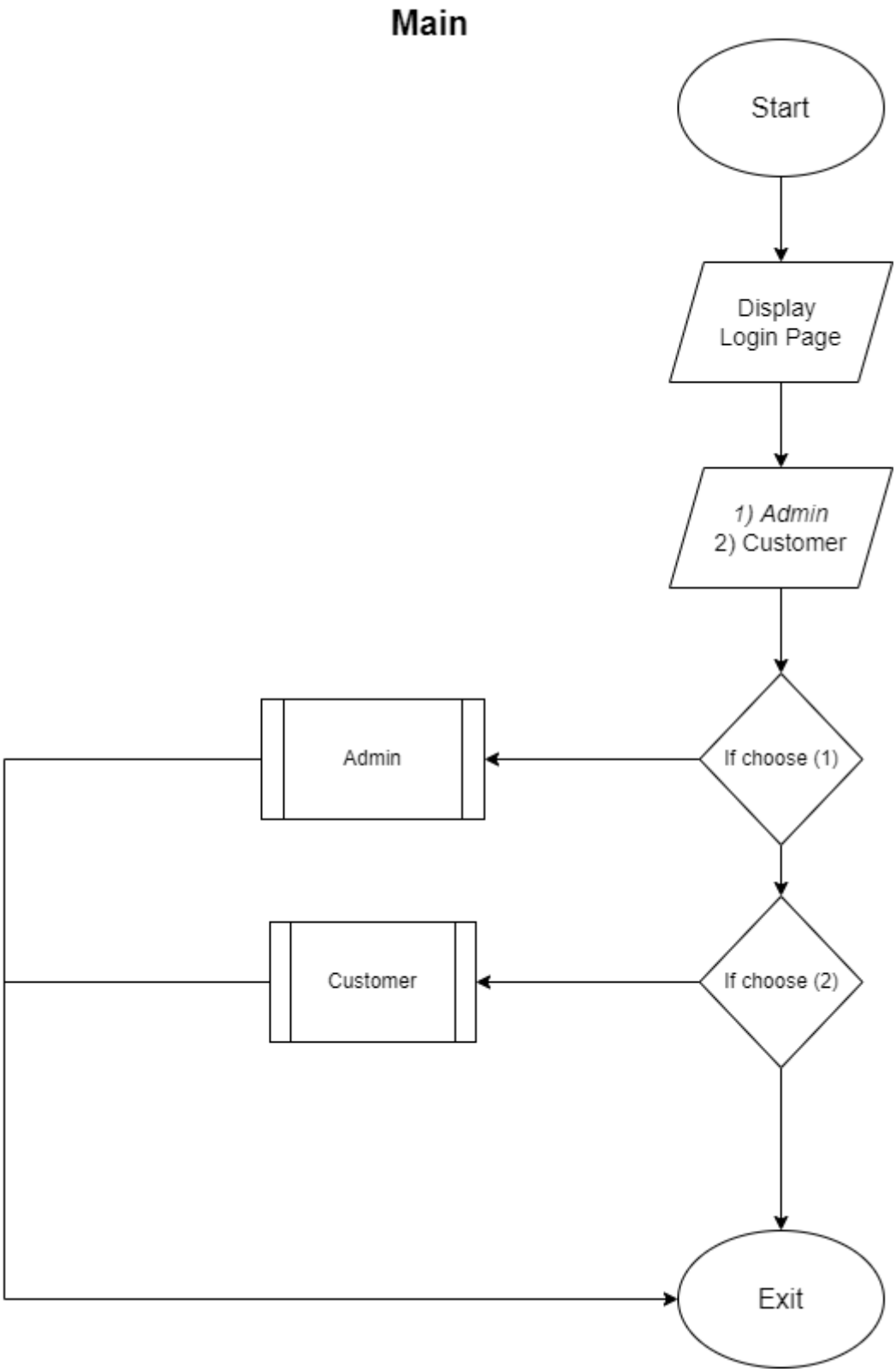

Figure 3.1.1: Main Flow Chart Bookshop Management System

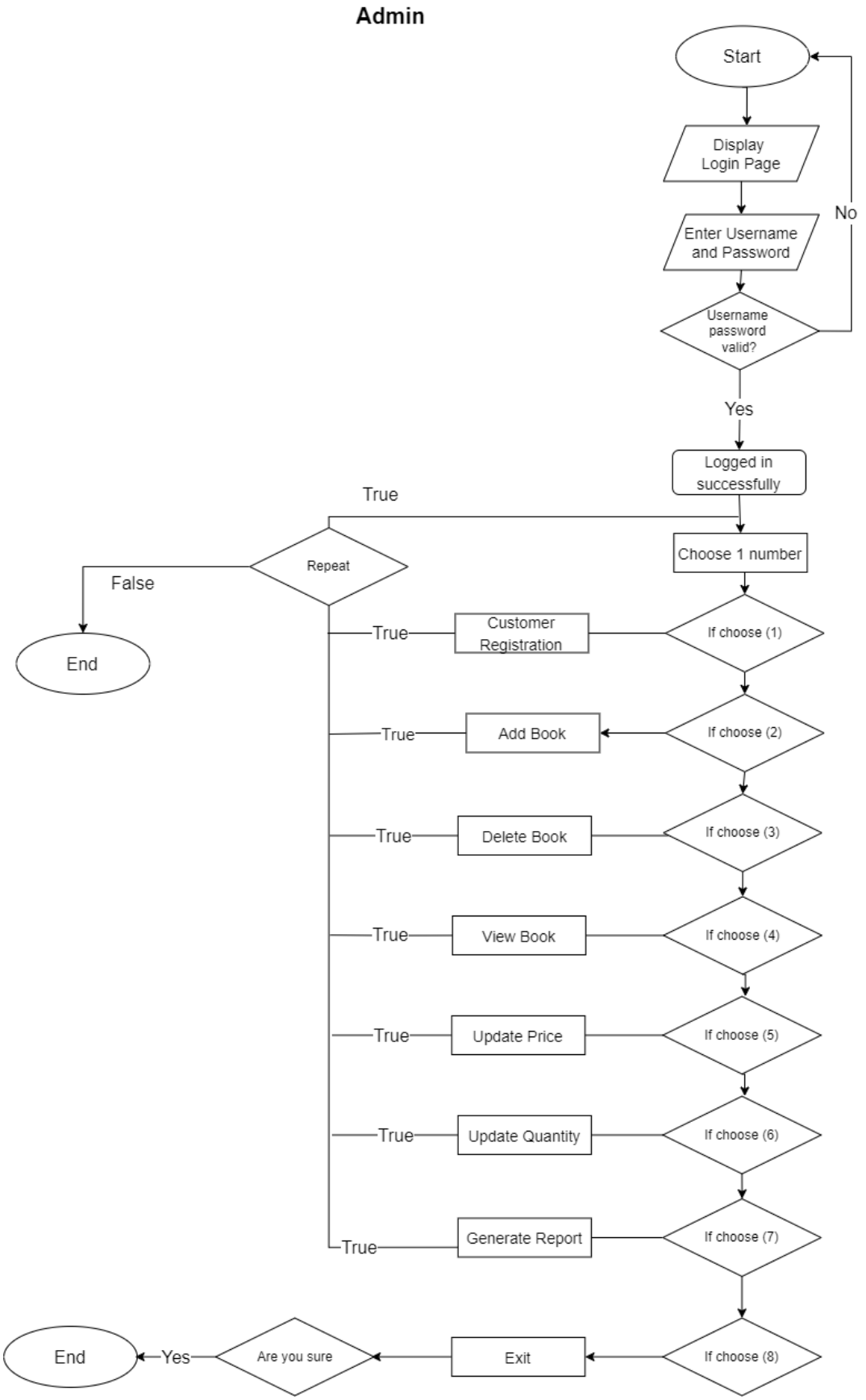

Figure 3.1.2: Admin Flow Chart Bookshop Management System

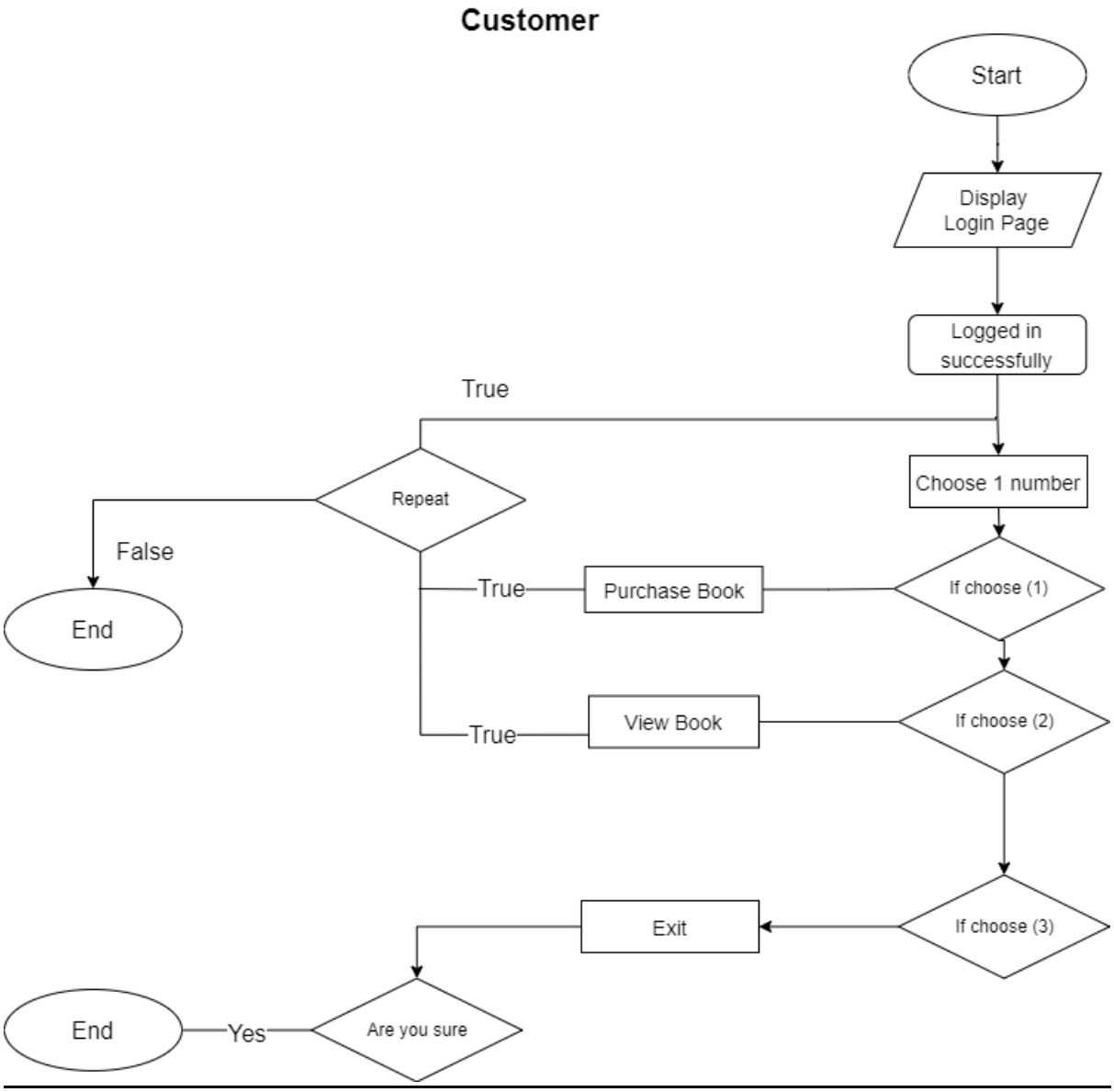

Figure 3.1.3: Customer Flow Chart Bookshop Management System

3.2 PSEUDO-CODE

A simple pseudo-based on flowchart that shown as above with some details of the system.

- **Login as Admin**

1. Start the program

2. Choose the option to Admin in main menu

3. Clear screen and go to the Administrator Page

4. . Choose the option to Login in admin menu

5. Clear screen and go to the Login Page

6. User need to enter correct username and password

7. If user enters invalid username and password

    Display the error message and enter again

    Else

    Login to account and display admin menu

6. Choose the option that had in admin menu

7. Choose to register customer

8. If want to register

    Fill in required details and username for verification

    If registered successfully

    Display successful message and go back to student menu

    Else

    Display error message and enter again

9. Else return to admin main menu

10. Choose to add book

11. If want to add book

    Enter the book ISBN

If data not found

Display successful message

Else

Display error message and enter again

12. Else return to admin main menu
13. Choose to delete book
14. If want to delete book

Display data and choose data that needed to delete

Enter the book ISBN

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

15. Else return to admin main menu
16. Choose to update book price
17. If want to update book price

Display data and choose data that needed to update book price

Enter the book ISBN

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

18. Else return to admin main menu
19. Choose to update book stock
20. If want to update book stock

Display data and choose data that needed to update book stock

Enter the book ISBN

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

21. Else return to admin main menu
22. Choose to generate report
23. If want to generate report

Choose to stock analysis

Enter the book ISBN

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

24. Else return to general report menu
25. If want to generate report

Choose to book analysis

Enter the book ISBN

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

26. Else return to general report menu
27. If want to generate report

Choose to customer analysis

Enter the customer ID

If data found

Display successful message

Else

Display error message and enter again

Else

Display error message and enter again

28. Else return to general report menu
29. Else return to admin menu
30. Exit

- **Login as Customer**

1. Start the program

2. Choose the option to Customer in main menu

3. Clear screen and go to the customer menu

4. . Choose the option to purchase

5. If want to purchase book

    Display data and choose data that needed to purchase

        Enter the book ISBN

            If data found

                Display successful message

            Else

                Display error message and enter again

        Enter the book quantity

            If data found

                Display successful message

            Else

                Display error message and enter again

    Else

        Display error message and enter again

6. Else return to customer menu

7. Choose to view book

8. If want to view book

    Display data

        If data found

            Display successful message

        Else

            Display error message and enter again

    Else

        Display error message and enter again

9. Else return to customer menu

10. Exit

3.3 ENTITY-RELATIONSHIP DIAGRAM

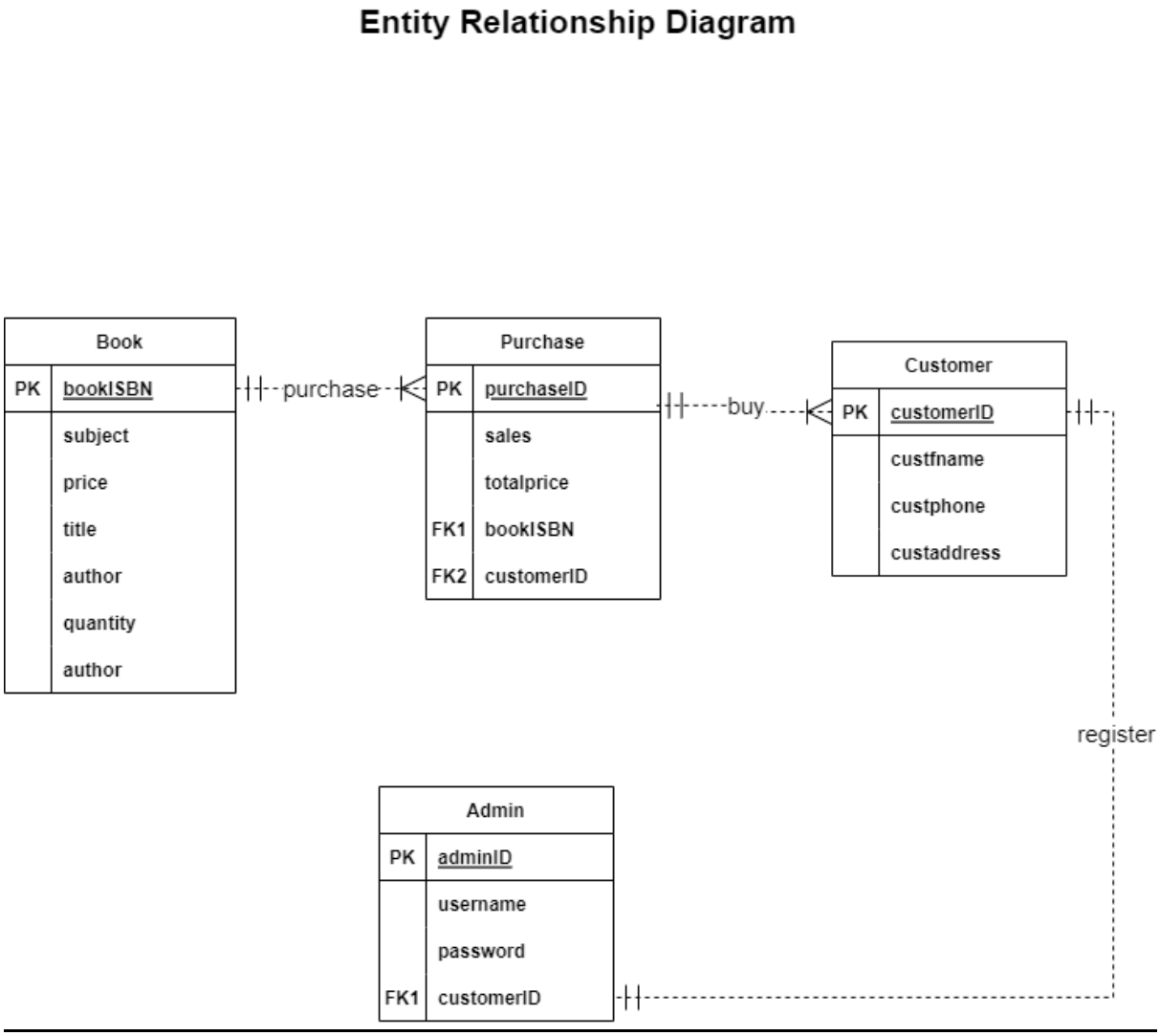

Figure 3.3.1 ERD of Bookshop Management System

3.3.1 BUSINESS RULES

The bussiness rules that related to ERD of this system are:

1. Admin can only have one adminID (primary key).
2. Customers can only purchase one book at a time. Each customer only has one customer id.
3. Book can be be purchase one or multiple times with bookISBN as primary key.
4. Admin can register for one or many customers at a time.

3.4 DATA DICTIONARY

admin

Column	Type	Null	Default	Comments
adminID *(Primary)*	int(15)	No		
username	varchar(15)	No		
password	varchar(15)	No		
name	varchar(40)	No		
phone	varchar(10)	No		
customerID	int(15)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	adminID	4	A	No	
Test	BTREE	No	No	customerID	4	A	No	

Figure 3.4.1: Data Dictionary for admin

books

Column	Type	Null	Default	Comments
bookISBN *(Primary)*	varchar(40)	No		
subject	varchar(40)	No		
price	float	No		
title	varchar(40)	No		
author	varchar(40)	No		
quantity	varchar(40)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	bookISBN	8	A	No	

Figure 3.4.2: Data Dictionary for books

customer

Column	Type	Null	Default	Comments
customerID *(Primary)*	int(15)	No		
custname	varchar(15)	No		
custphone	varchar(11)	No		
custaddress	varchar(40)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	customerID	8	A	No	

Figure 3.4.3: Data Dictionary for customer

purchase

Column	Type	Null	Default	Comments
purchaseID *(Primary)*	int(11)	No		
sales	varchar(40)	No		
totalprice	float	No		
bookISBN	varchar(40)	No		
customerID	int(15)	No		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	purchaseID	7	A	No	
bookISBN	BTREE	No	No	bookISBN	7	A	No	
customerID	BTREE	No	No	customerID	7	A	No	

Figure 3.4.4: Data Dictionary for purchase

3.5 INTERFACE DESIGN

The interface design of the bookshop management system is as shown below:

- Main Menu

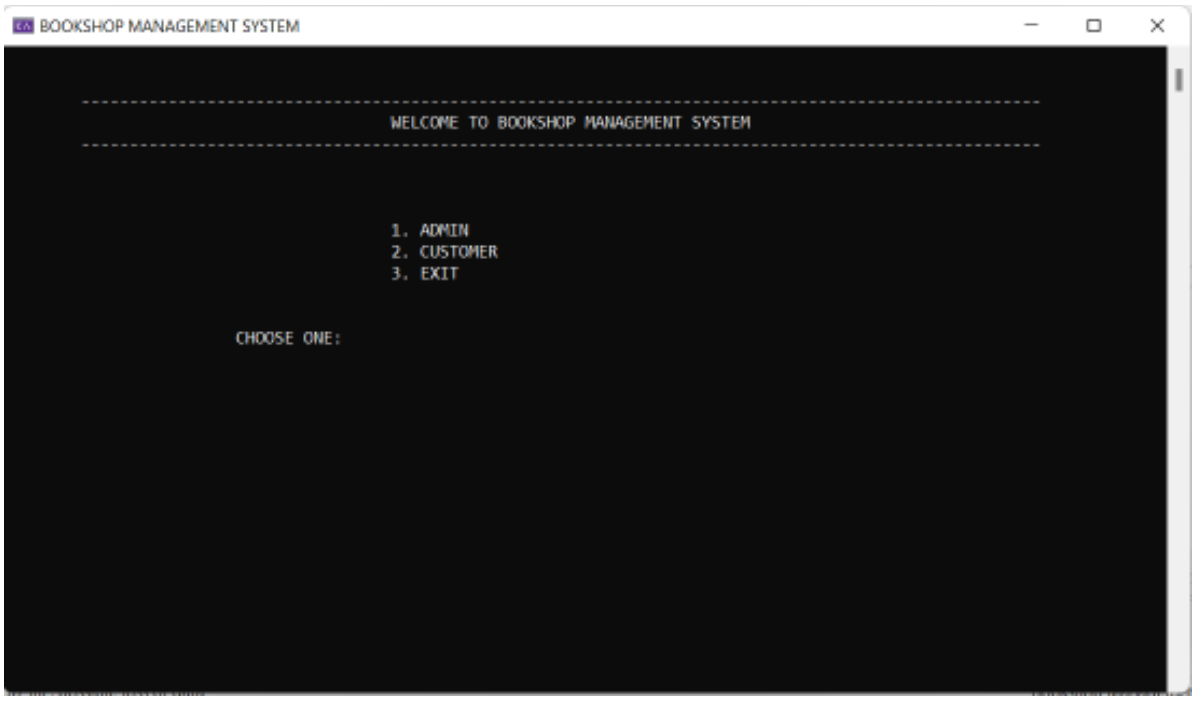

Figure 3.5.1: Main Menu Page

Admin Page

- Administrator Login

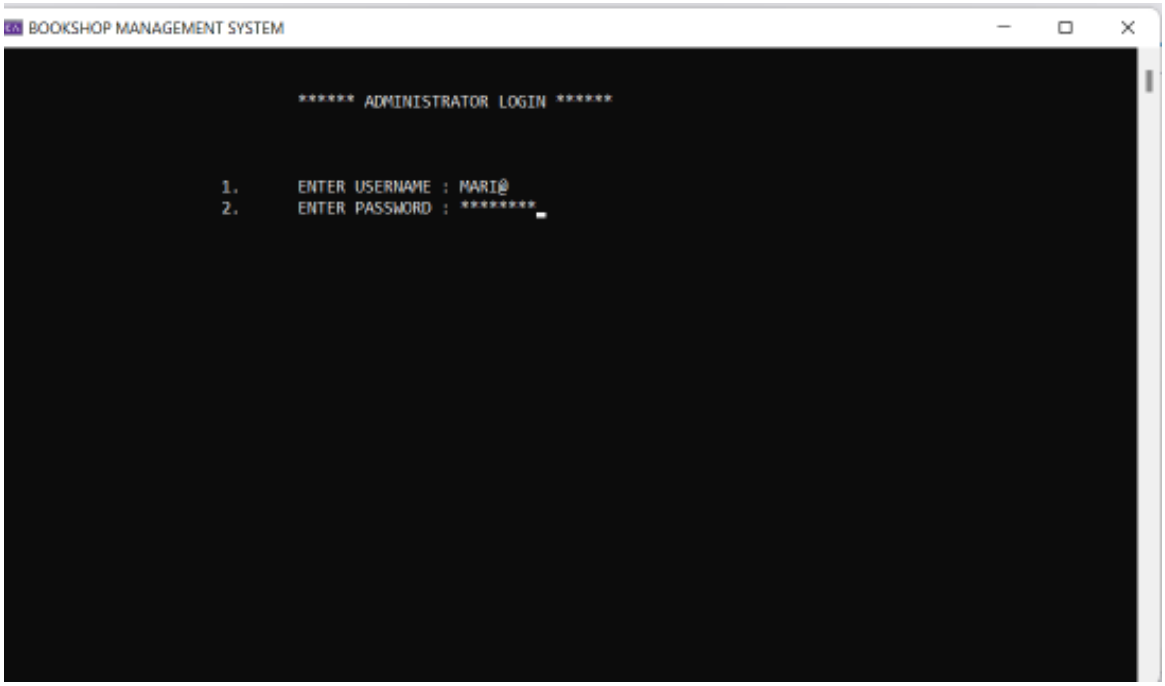

Figure 3.5.2: Administrator Login

- Administrator Menu

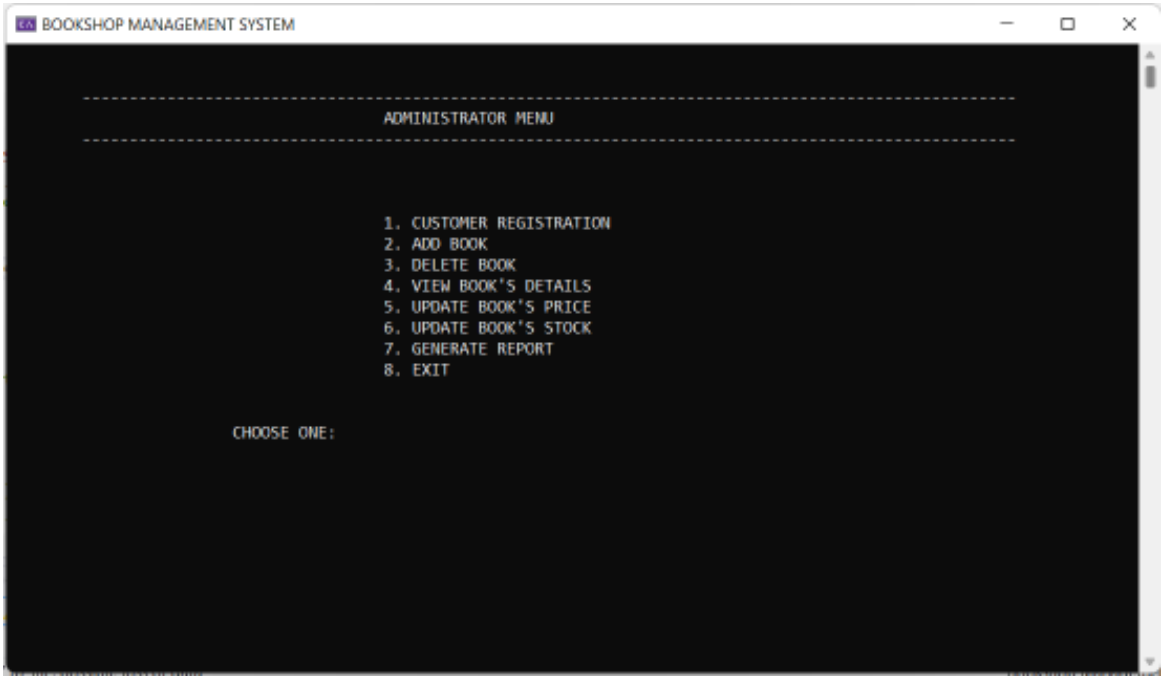

Figure 3.5.3: Administrator Menu

- Customer Registration

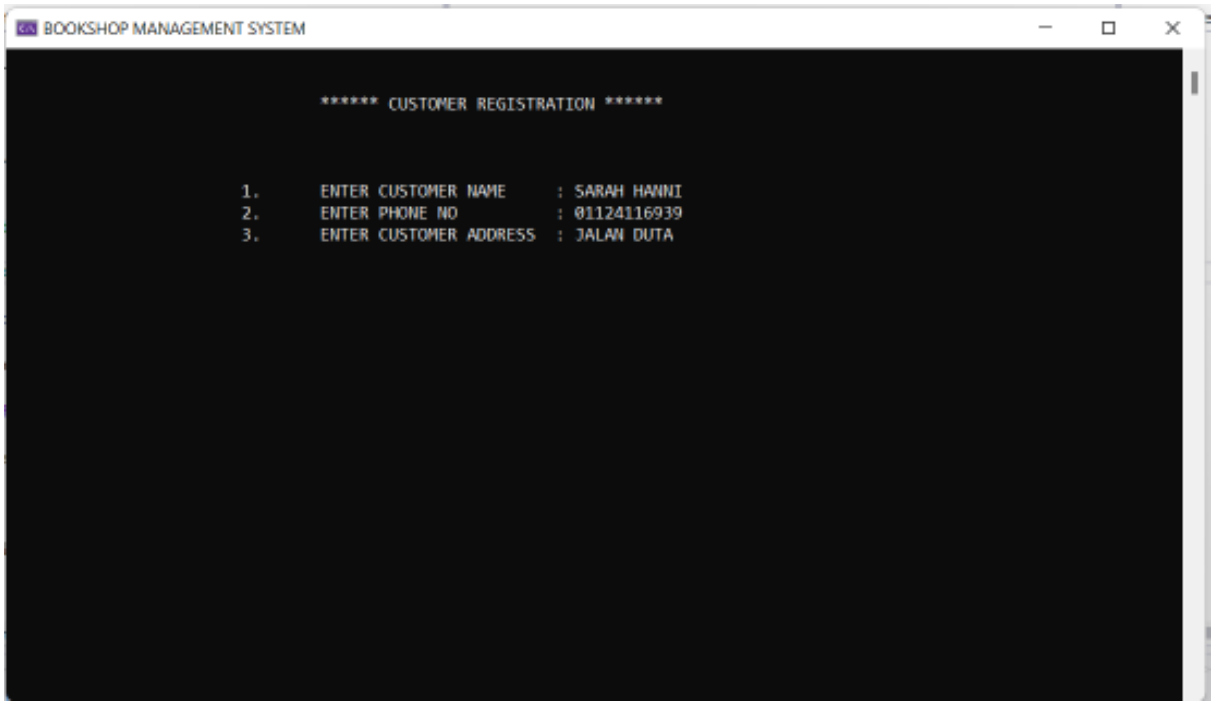

Figure 3.5.4: Customer Registration

- Adding Book

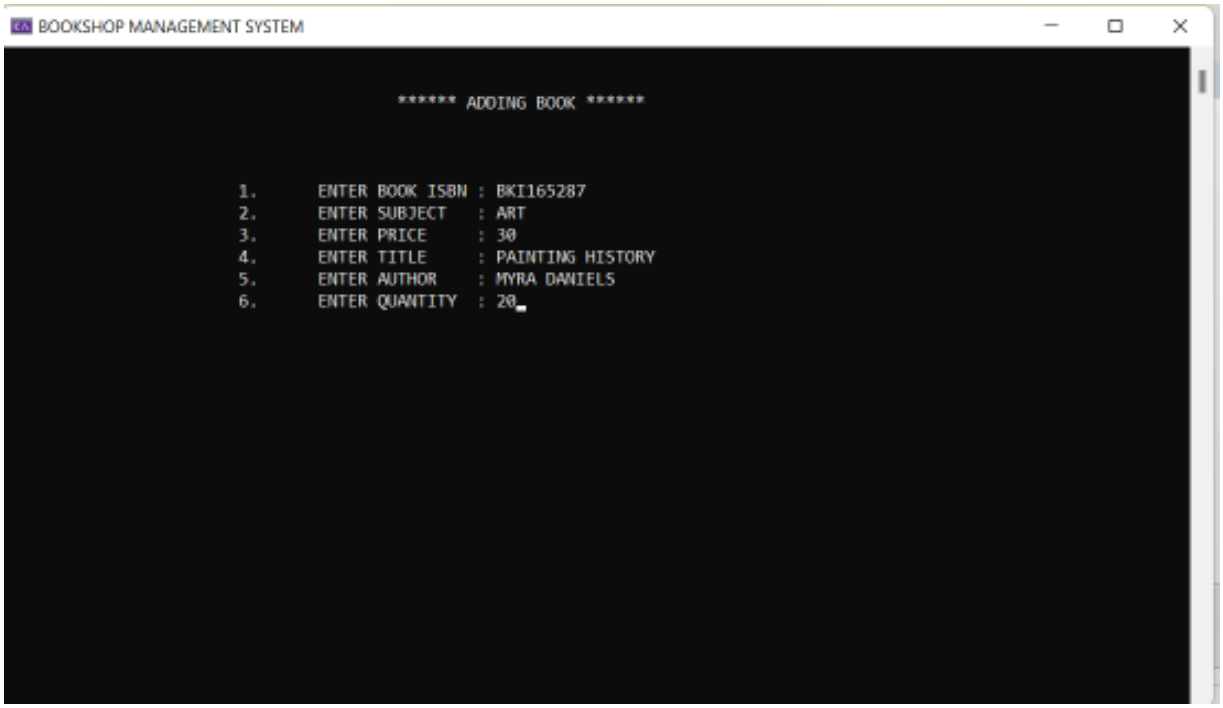

Figure 3.5.5: Adding Book Page

- Deleting Book

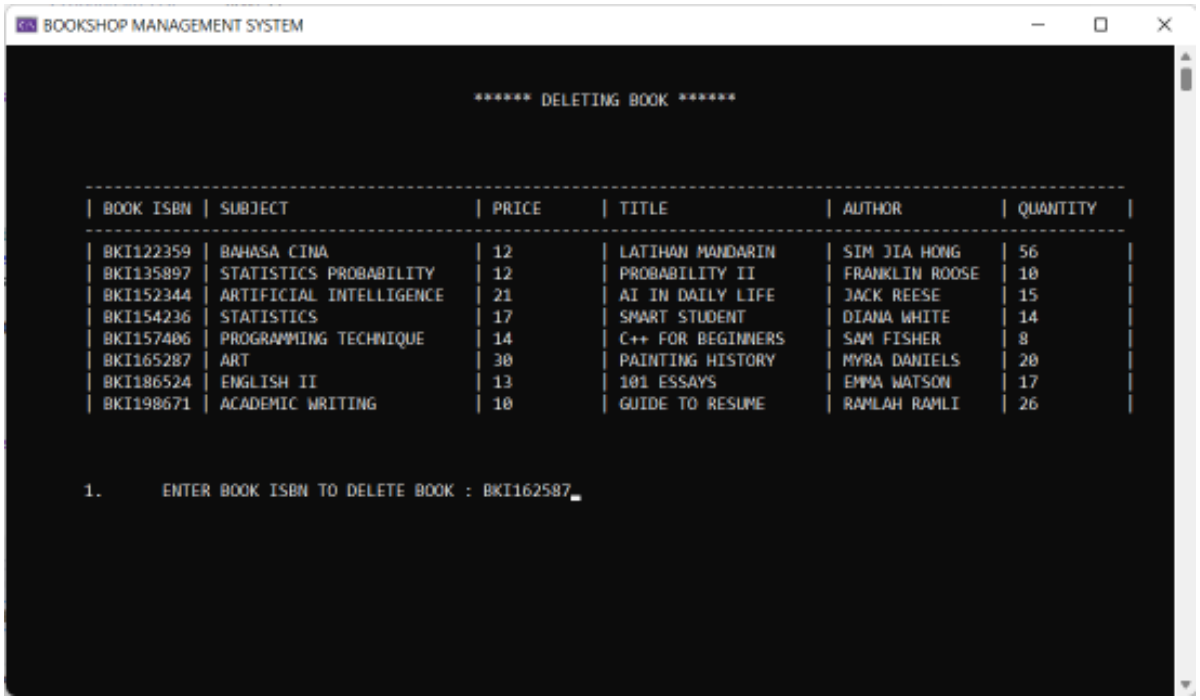

Figure 3.5.6: Deleting Book Page

- Update Book Price

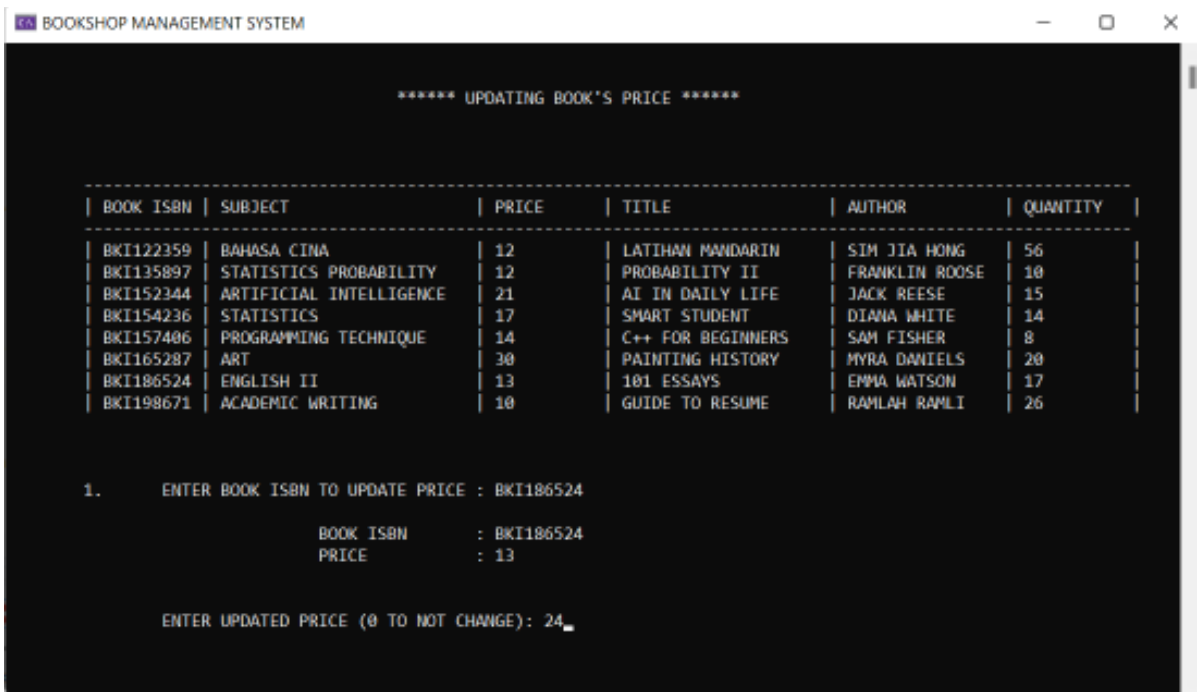

Figure 3.5.7: Update Book Price Page

- Update Book Stock

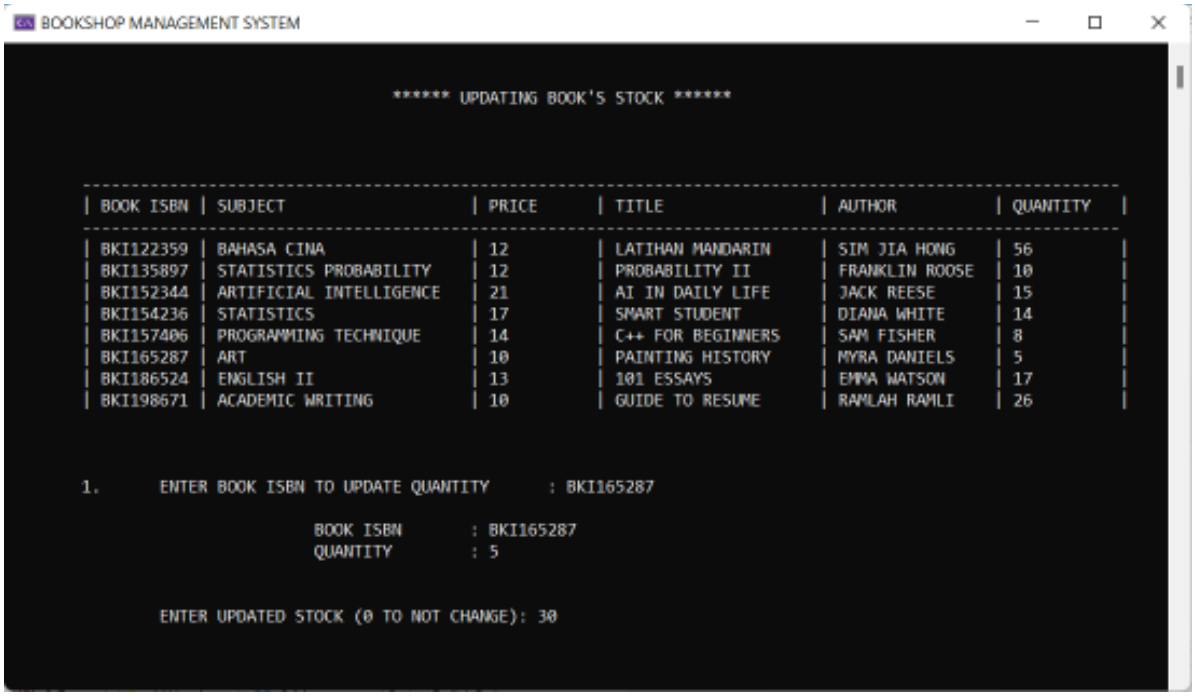

Figure 3.5.8: Update Book Stock Page

- Generate Report

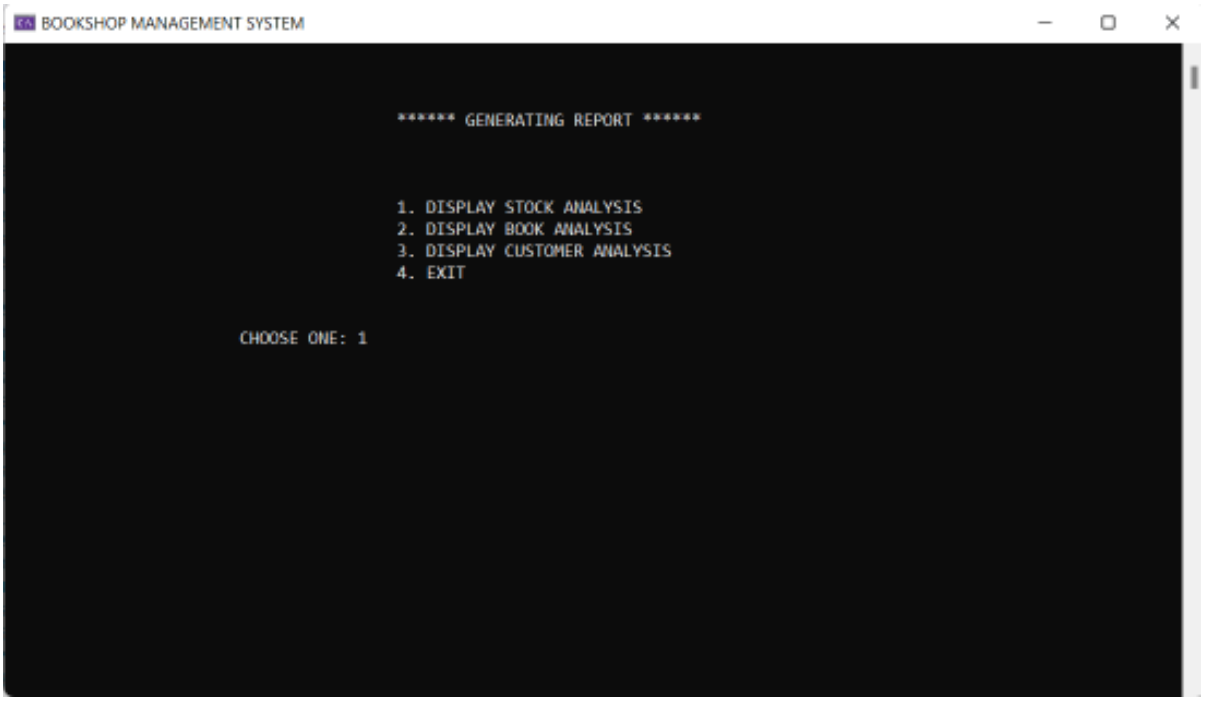

Figure 3.5.9: Generate Report Page

Figure 3.5.9.1: Generate Report Page

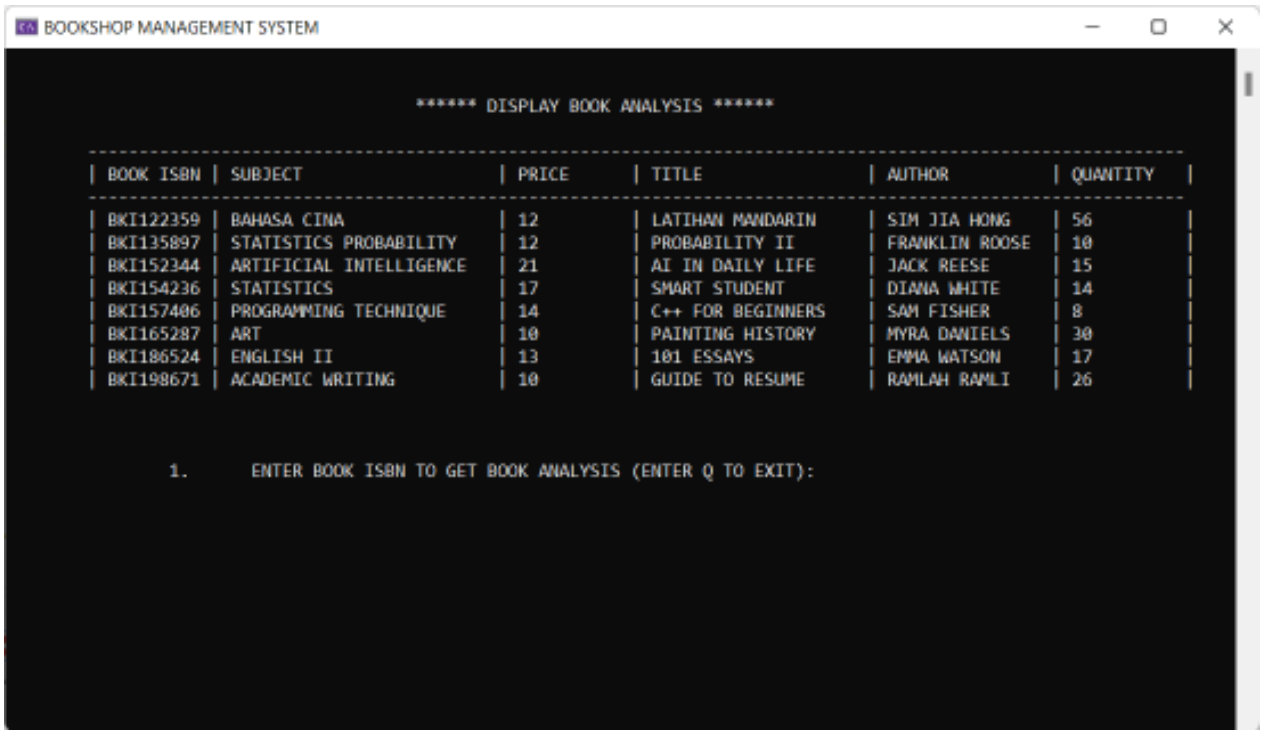

Figure 3.5.9.2: Generate Report Page

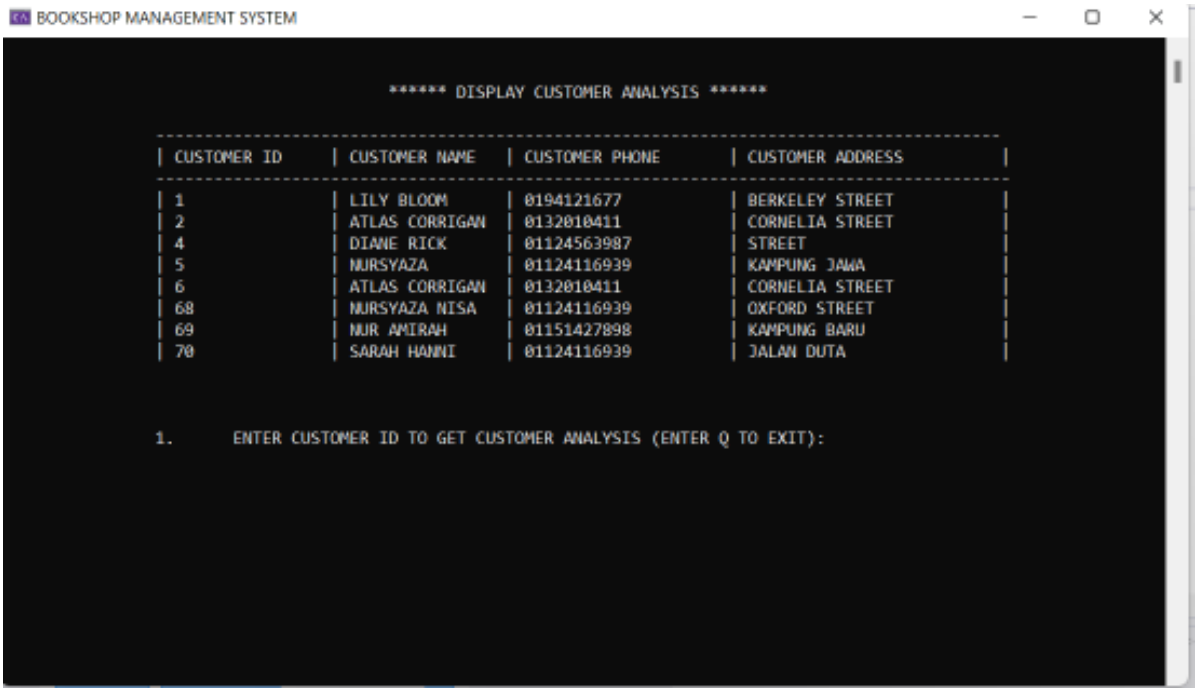

Figure 3.5.9.3: Generate Report Page

Customer Page

- Customer Menu

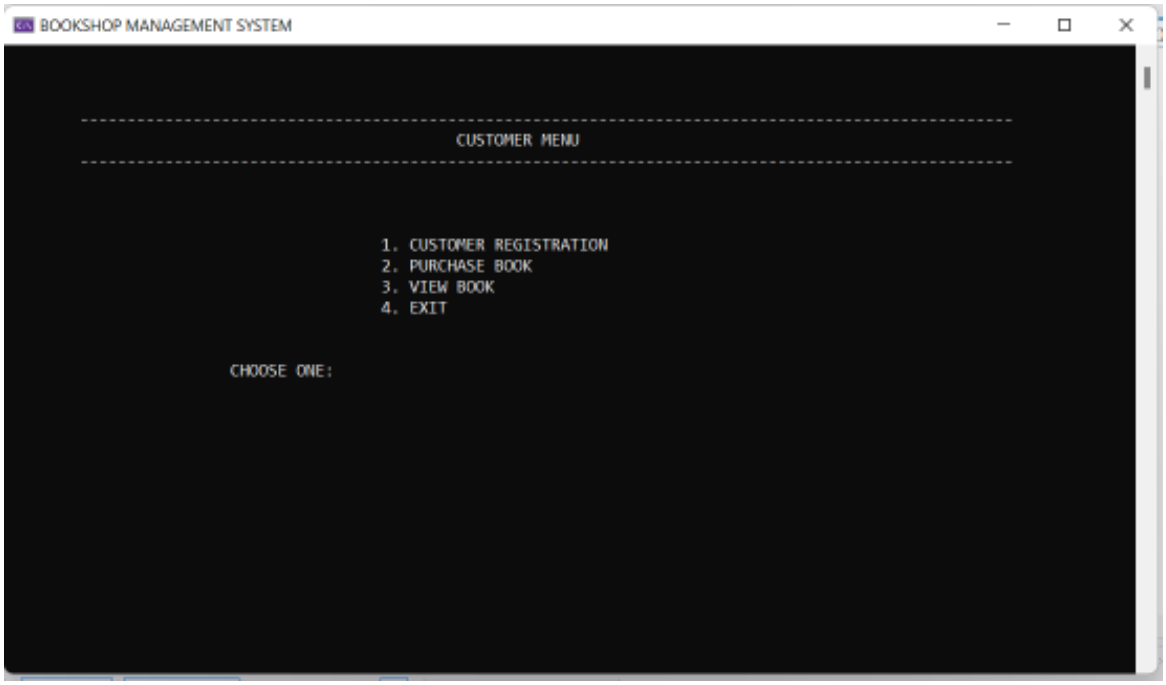

Figure 3.5.10: Customer Menu Page

- Purchase Book

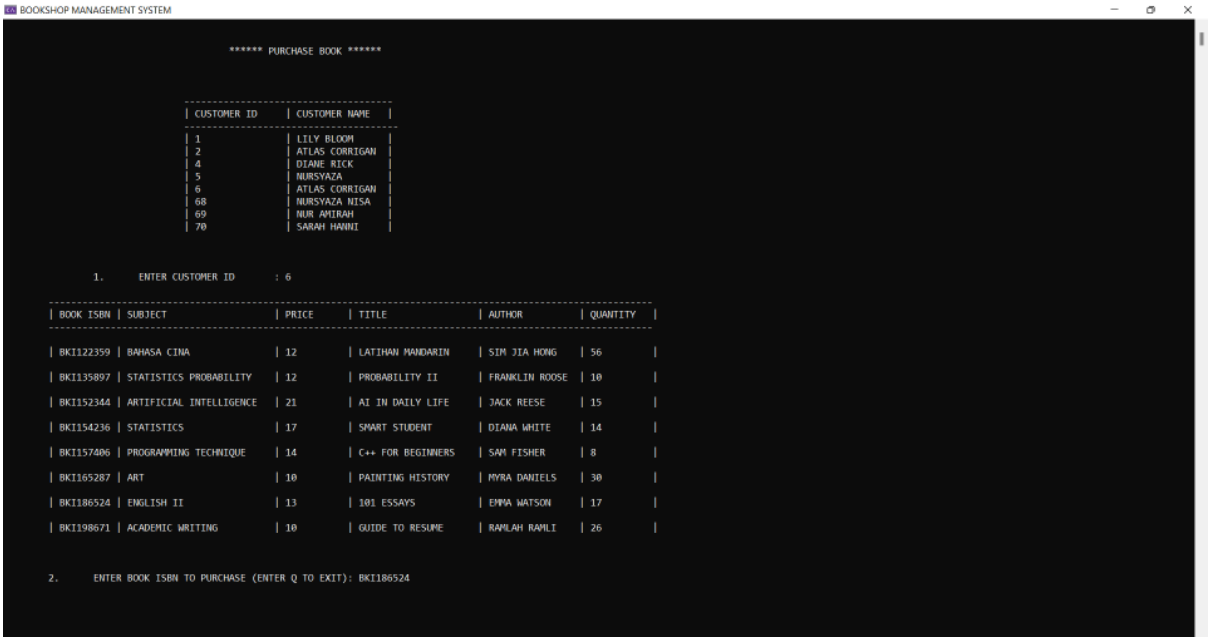

Figure 3.5.11.1: Purchase Book Page

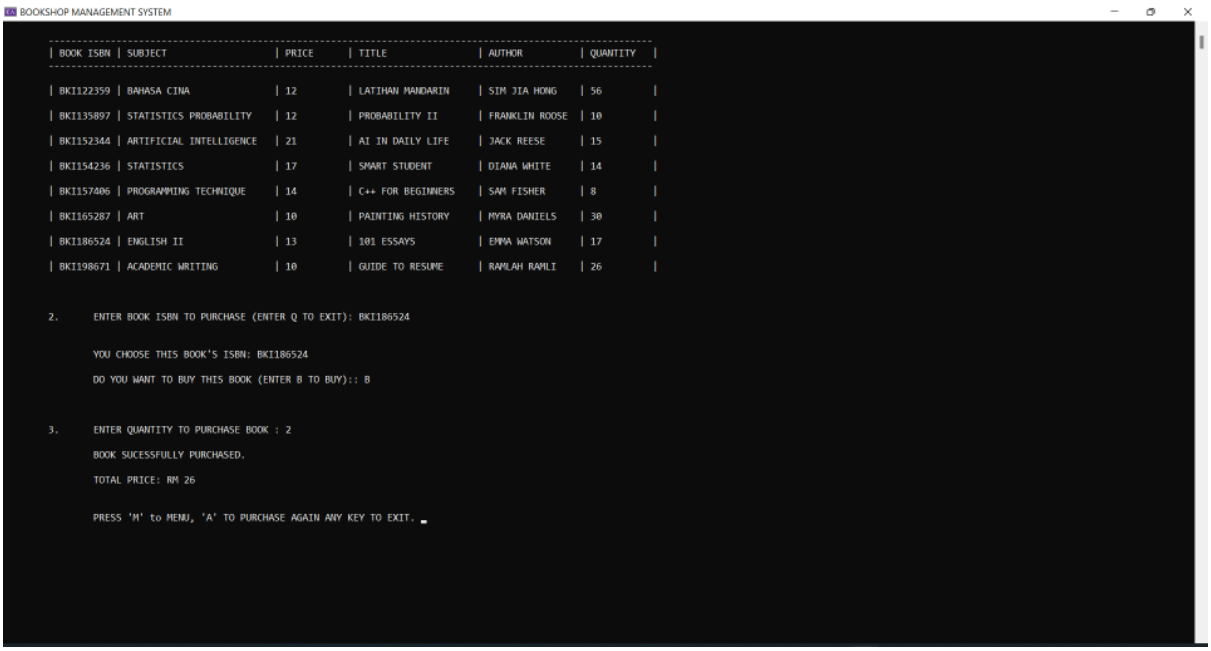

Figure 3.5.11.2: Purchase Book Page

- View Book Details

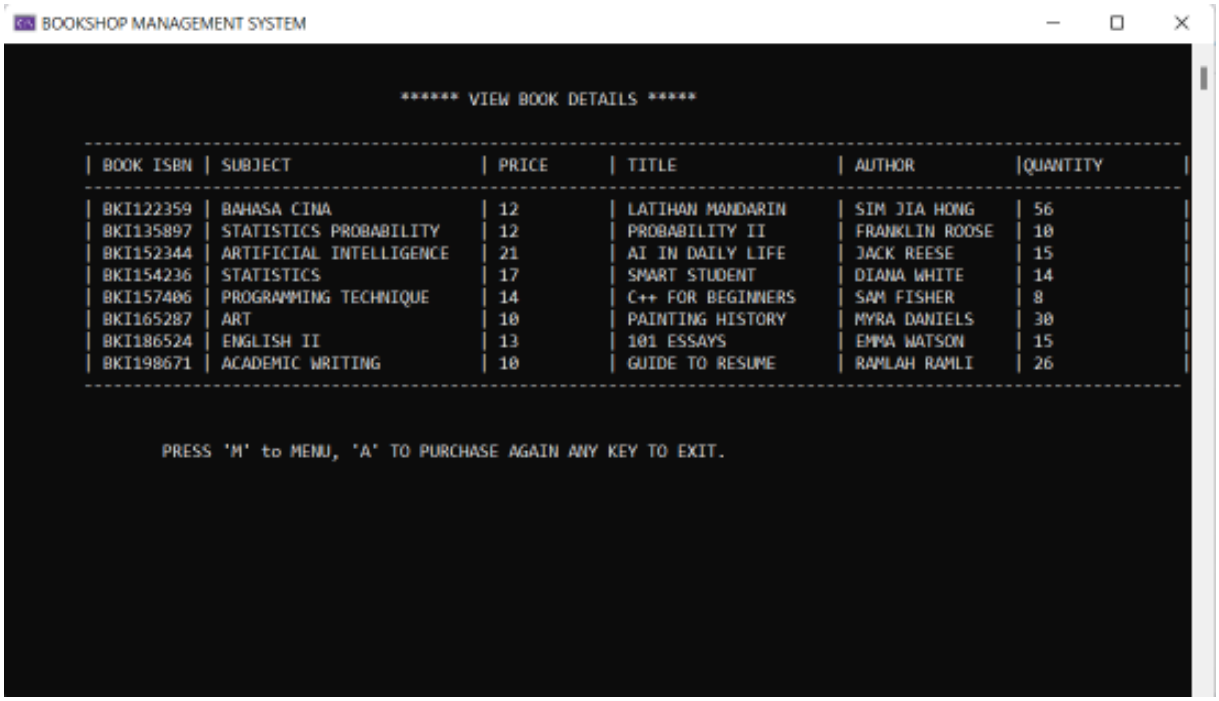

Figure 3.5.12: View Book Page

- Exit

Figure 3.5.13: Exit Page

CHAPTER 4: IMPLEMENTATION

4.1 SQL QUERY

In the implementation part, an example of each query which are insert, delete, update and delete that is created will be shown.

Before showing the inquiries, below is the coding database connection to phpMyAdmin which stores and receives data.

```
class db_response
{
public:
    static void ConnectionFunction()
    {
        conn = mysql_init(0);
        if (conn)
        {
            cout << "DATABASE CONNECTED" << endl;
            system("cls");
        }
        else
            cout << "FAILED TO CONNECT" << mysql_errno(conn) << endl;
        conn = mysql_real_connect(conn, "localhost", "root", "", "db_bookshop", 3306, NULL, 0);
        if (conn)
        {
            cout << "DATABASE CONNECTED TO MYSQL" << conn << endl;
            system("pause");
        }
        else
            cout << "FAILED TO CONNECT!" << mysql_errno(conn) << endl;
    }
};
```

Figure 4.1: Database Connection

```
string insert_query = "INSERT INTO books (bookISBN, subject, price, title, author, quantity) VALUES ('" + bookISBN + "', '" + subject + "', '" + price + "',  '" + title + "',  '" + author + "', '" + quantity + "')";

const char* q = insert_query.c_str();

qstate = mysql_query(conn, q);

if (!qstate)
{
    cout << endl << "\n     BOOK HAS SUCCESSFULLY ADDED!!" << endl;
}
else
{
    cout << "\n     QUERY EXECUTION PROBLEM!" << mysql_errno(conn) << endl;
}
```

Figure 4.1.1: Insert Query

Figure 4.1.1: Insert Query shown the insert query of books table for function Addbook();

```
string delete_query = "DELETE FROM books WHERE bookISBN = '" + strISBN + "'";
const char* qd = delete_query.c_str();
qstate = mysql_query(conn, qd);

if (!qstate)
{
    cout << endl << "\n     BOOK HAS SUCCESSFULLY DELETED!! " << endl;
    goto ExitMenu;
}
else
{
    cout << "\n     FAILED TO UPDATE!!" << mysql_errno(conn) << endl;
}
```

Figure 4.1.2: Delete Query

Figure 4.1.2 shown the delete query of books table for function Deletebook();

```
if (!qstate)
{

    string findbyISBN_query = "SELECT * FROM books WHERE bookISBN like '" + strISBN + "%'";
    const char* qi = findbyISBN_query.c_str();
    qstate = mysql_query(conn, qi);
    res = mysql_store_result(conn);
    while (row = mysql_fetch_row(res))
    {
        cout << "               BOOK ISBN   : " << row[0]
            << "\n              SUBJECT     : " << row[1]
            << "\n              PRICE       : " << row[2]
            << "\n              TITLE       : " << row[3]
            << "\n              AUTHOR      : " << row[4]
            << "\n              QUANTITY    : " << row[5]
            << endl << endl;

        storeISBN       = row[0];
        storesubject    = row[1];
        storeprice      = row[2];
        storetitle      = row[3];
        storeauthor     = row[4];
        storequantity   = row[5];

    }
}
else
{
    cout << "       QUERY EXECUTION PROBLEM!!" << mysql_errno(conn) << endl;
}
```

Figure 4.1.3: Select Query

Figure 4.1.3 shown the select query which had to search and get information of a specific book also in function Deletebook();

```
string update_query = "UPDATE books SET price = '" + price + "'WHERE bookISBN = '" + storeISBN + "'";
const char* qu = update_query.c_str();
qstate = mysql_query(conn, qu);

if (!qstate)
{
    cout << endl << "       BOOK PRICE HAS SUCCESSFULLY UPDATED!! \n" << endl;
    string  findbyISBN_query = "SELECT * FROM books WHERE bookISBN LIKE '" + storeISBN + "%'";
    const char* qi = findbyISBN_query.c_str();
    qstate = mysql_query(conn, qi);

    if (!qstate)
    {
        res = mysql_store_result(conn);
        while ((row = mysql_fetch_row(res)))
        {
            printf("                    -------------------------\n");
            printf("                    | %-5s | %-10s |\n", "BOOK ISBN", "PRICE");
            printf("                    -------------------------\n");
            printf("                    | %-5s | %-10s |\n", row[0], row[2]);
        }
    }
}
else
{
    cout << "\n     FAILED TO UPDATE!!" << mysql_errno(conn) << endl;
}
```

Figure 4.1.4: Update Query

Figure 4.1.3 shown the update query which had to update the price of a specific book in function UpdatePrice();

4.2 PROGRAMMING TECHNIQUES

The programming techniques that applied in this system are function, selection control and structured data. The coding for each programming techniques in this system is shown as below.

- Function

```
// Variables
string bookISBN;
string subject;
string price;
string title;
string author;
string quantity;
char choose;
// Variables end
```

Figure 4.2.1: Function without Parameters

Figure 4.2.1 shown the function without parameter for the book class. This function will be called upon the implementation.

```
// Variables
string customerID ;
string custname ;
string custphone;
string custaddress;
// Variables end
```

Figure 4.2.2: Function without Parameters

Figure 4.2.2 shown the function without parameter for the customer class. This function will be called upon the implementation.

- Selection Control

```
cout << endl << endl;
cout << "   ------------------------------------------------------------------------------------------------------------\n";
cout << "                    ADMINISTRATOR MENU        " << endl;
cout << "   ------------------------------------------------------------------------------------------------------------\n" << endl << endl;
cout << "\n                1. CUSTOMER REGISTRATION";
cout << "\n                2. ADD BOOK";
cout << "\n                3. DELETE BOOK";
cout << "\n                4. VIEW BOOK'S DETAILS";
cout << "\n                5. UPDATE BOOK'S PRICE";
cout << "\n                6. UPDATE BOOK'S STOCK";
cout << "\n                7. GENERATE REPORT";
cout << "\n                8. EXIT" << endl << endl;

cout << "\n         CHOOSE ONE: ";
cin >> choose;

switch (choose)
{
case 1:
    CustomerRegistration();
    break;
case 2:
    AddBook();
    break;
case 3:
    DeleteBook();
    break;
case 4:
    ViewBook();
    break;
case 5:
    UpdatePrice();
    break;
case 6:
    UpdateStock();
case 7:
    GenerateReport();
    break;
case 8:
    //ExitCode
ExitProgram:
    cout << "\n\n   PROGRAM TERMINATING. ARE YOU SURE? (y/N): ";
    cin >> exitSecurity;
    if (exitSecurity == 'y' || exitSecurity == 'Y') {
    }
    else if (exitSecurity == 'n' || exitSecurity == 'N') {
        system("cls");
        main();
    }
    else {
        cout << "   NEXT TIME CHOOSE AFTER CORRESPONING RULES." << endl;
        goto ExitProgram;
    }
    break;
default:
    cout << "\n PLEASE CHOOSE BETWEEN 1-6....PRESS TO CONTINUE";
    system("pause");
    system("cls");
    main();
    break;

}
```

Figure 4.2.3: Switch case for the Administrator Menu

```
system("cls");

// Variables
int choose;
cout << endl;
cout << endl<< "\n                ****** GENERATING REPORT ******            \n\n" << endl;

cout << "\n                1. DISPLAY STOCK ANALYSIS ";
cout << "\n                2. DISPLAY BOOK ANALYSIS ";
cout << "\n                3. DISPLAY CUSTOMER ANALYSIS ";
cout << "\n                4. EXIT" << endl << endl;

cout << "\n        CHOOSE ONE: ";
cin >> choose;

switch (choose)
{
case 1:
    StockAnalysis();
    break;
case 2:
    BookAnalysis();
    break;
case 3:
    CustomerAnalysis();
    break;
case 4:
    goto ExitMenu;
}
```

Figure 4.2.4: Switch case for the Generate Report

Figure 4.2.4 is the switch case for the generate report which runs inside the do-while loop in the main file. The user will go to the option page when a valid number is entered. If the user entered an invalid input and number, an error message wil be display and the user will need to enter number again by looping. Same technique used for the main menu, admin menu and customer menu.

- Class & Object

```
class db_response
{
public:
    static void ConnectionFunction()
    {
        conn = mysql_init(0);
        if (conn)
        {
            cout << "DATABASE CONNECTED" << endl;
            system("cls");
        }
        else
            cout << "FAILED TO CONNECT" << mysql_errno(conn) << endl;
        conn = mysql_real_connect(conn, "localhost", "root", "", "db_bookshop", 3306, NULL, 0);
        if (conn)
        {
            cout << "DATABASE CONNECTED TO MYSQL" << conn << endl;
            system("pause");
        }
        else
            cout << "FAILED TO CONNECT!" << mysql_errno(conn) << endl;
    }
};
```

Figure 4.2.5: Class

Figure 4.2.5 shows the class of database connection. This function will be called to perform instances to know whether the database is connected to phpMyAdmin.

CHAPTER 5: CONCLUSION

In conclusion, Bookshop Management System is reliable system that is developed to let both customer and admin to use the system in managing book and purchase transaction. Customer do not need to go to the bookshop and queue to purchase the book on the counter. Instead, the computerization bookshop management system allows the customer to check the availability and the price of the product before buying it. On the other side, the admin can save their time on data entry of each customer as they do have to manually record and save the data. Hence, the productivity and work efficiency of the admin will increase as the problem such as time-consuming has not become a problem. All the processes will become faster and easier to track and analyse back the records.

The limitation of this system is the colour of the interface that is a bit dull which in back and white. Besides that, this system is unable to generate a visual representation of analysis such as graph report, pie chart or histogram based on the report that has been generated. This as wells, has become the limitation of this system as a coloured analysis will be much easier to interpret compared to the black and white ones.

At last, the suggestion on improvement of this system is to design an interesting interface for this system. The colour of the system design should be bright, attractive and minimal so that the user will not feel bored looking at the black and white screen all the time when using this system. Furthermore, the system can be improved by abling to generate the visually coloured analysis such as pie chart or other forms for data analysis purpose. Besides that, adding more functions and features can enhance the system and increase the usability of the system by the users.

REFERENCES

1. Workshop 1 Committee. Guidance Book Implementation of Workshop 1. Fakulti Teknologi Maklumat dan Komunikasi Universiti Teknikal Malaysia Melaka.
2. https://www.geeksforgeeks.org/bookshop-management-system-using-file-handling/
3. https://www.codewithc.com/bookshop-management-system-project-in-c/