UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

WORKSHOP 1 REPORT

NAME	**MUHAMMAD ZULFAHMI BIN ZAMBRI**
MATRIC NUMBER	**B031910124**
COURSE	**BACHELOR OF COMPUTER SCIENCE (SOFTWARE DEVELOPMENT)**
PROJECT TITLE	**CAR DEALERSHIP SYSTEM**
SUPERVISOR NAME	**PROFESOR DR. MOHD KHANAPI BIN ABD GHANI**
EVALUATOR NAME	**TS. MUHAMMAD SUHAIZAN BIN SULONG**

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction

Development of an interactive car sale system that lets a customer to find a car and its details is the main objective of this project. The user can search the details of every car. Administrators are responsible of maintaining the details of vehicles like the Car Model, Car Brand and Car Price. The system's main functions includes signing in and log in functionality, customers can look up various cars listing with details included. The system provides a search algorithm that enables the user to find a car with all details displayed matching the car model. Users can also view the information of vehicles purchased and their particulars. Menu's and toolbars are part of the excellent user interface implemented in the project.

1.2 Problem Statement

In today's fast-paced society, it is very hard to be competitive without using cutting-edge technology available in market. It is becoming a challenge for automotive company to manage that data in an effective way. To be more productive in order processing, automotive company needs a solution that can facilitate their current processes with use of technology. With increased amount of orders, it will be difficult for automotive company to manage orders in an effective and efficient manner. It will be very hard to go through all paper work and back tracking orders. If there is any complain or review of any order, it takes large amount of effort and time to backtrack and fix the problem. This results in loss of resources, increased time and low output.

1.3 Background of Project

The title of this system is Car Dealership System, which is a system designed for a customer to make car booking arrangement easier. Once the customer registers their data such as first name, last name, id and number telephone to Car Dealership System, the data will save to the database and customer can login. After that customer can view and update his/her personal information. Customer can also search and book car. By implementing this system, the database system can easily manage and organize the entire customer's data. This system also ease the customer to store all their information rather than go to automotive store and fill the form manually.

1.4 Objectives

This project embarks on the following objectives:

1. To develop a system that allow the customer to create and manage their account on their own.
2. To make sure the customer have the information about cars.
3. To able the customer to book a car.
4. To resolve the loss of data and store it in a more organized way.

1.5 Scopes

1. Modules to be developed :
   - i. User access module – Registration and login
   - ii. Information module – Search and display available car
   - iii. Process module – Book car
   - iv. Report module – Display user's details, booked car details

2. Target user :

Limited for customer to:

i. Register to the system
ii. Able to login
iii. Able to update personal information
iv. View list of cars
v. View booked history

CHAPTER 2

ANALYSIS OF PROBLEM

2.1 Problem Description

Nowadays, automotive company has a problem to deal with having many customers. This will give problem in term of communicating with customers for their needs for buying a car. The company itself will have contact their customers to inform information about cars. The customers also has problem to check information about the car that they interested, to check the price of the car and to book a car that suitable with their needs.

In existing system, most of actions are manually written. The system used in the automotive company is outdated, as it requires paper, files and the binders, which will require the human workforce to maintain them. From the customers writing down their personal information in the registration form, then the automotive company's clerks will keep the form in the files, down to the record of the registration. This makes the process longer. This is time consuming, has much cost and consume a lot of workload.

For the proposed system, Car Dealership System is a system where the customers can organize; manage all the data of its own. The system can help customers for any process that need without going to the automotive company. Moreover, this system will help the process with more efficiency without time consuming. Furthermore, customers can check the information of the car without having to go to automotive company and ask the clerk regularly.

2.2 Problem Decomposition

No	Decomposition	Solution
1	For start up a automotive company, it will take a higher budget for hiring database architect.	This system can help to reduce the budget for open a automotive company.
2	For owning a automotive company, there will be a dozens of customer data need to be keep for future purposes.	This system can help to saves customer's data without being corrupted.
3	Customer always dissatified with their own choices when buying a new car.	The system's main functions includes, customers can look up various cars listing with details included.

Table 1.1 *Project decomposition and solution*

Table 1.1 shows the decomposition and solution based on the current system to make a new way to manage store management. For example, current system need to look up car information manually and now customers see can look up the information through system.

2.3 Structured Chart

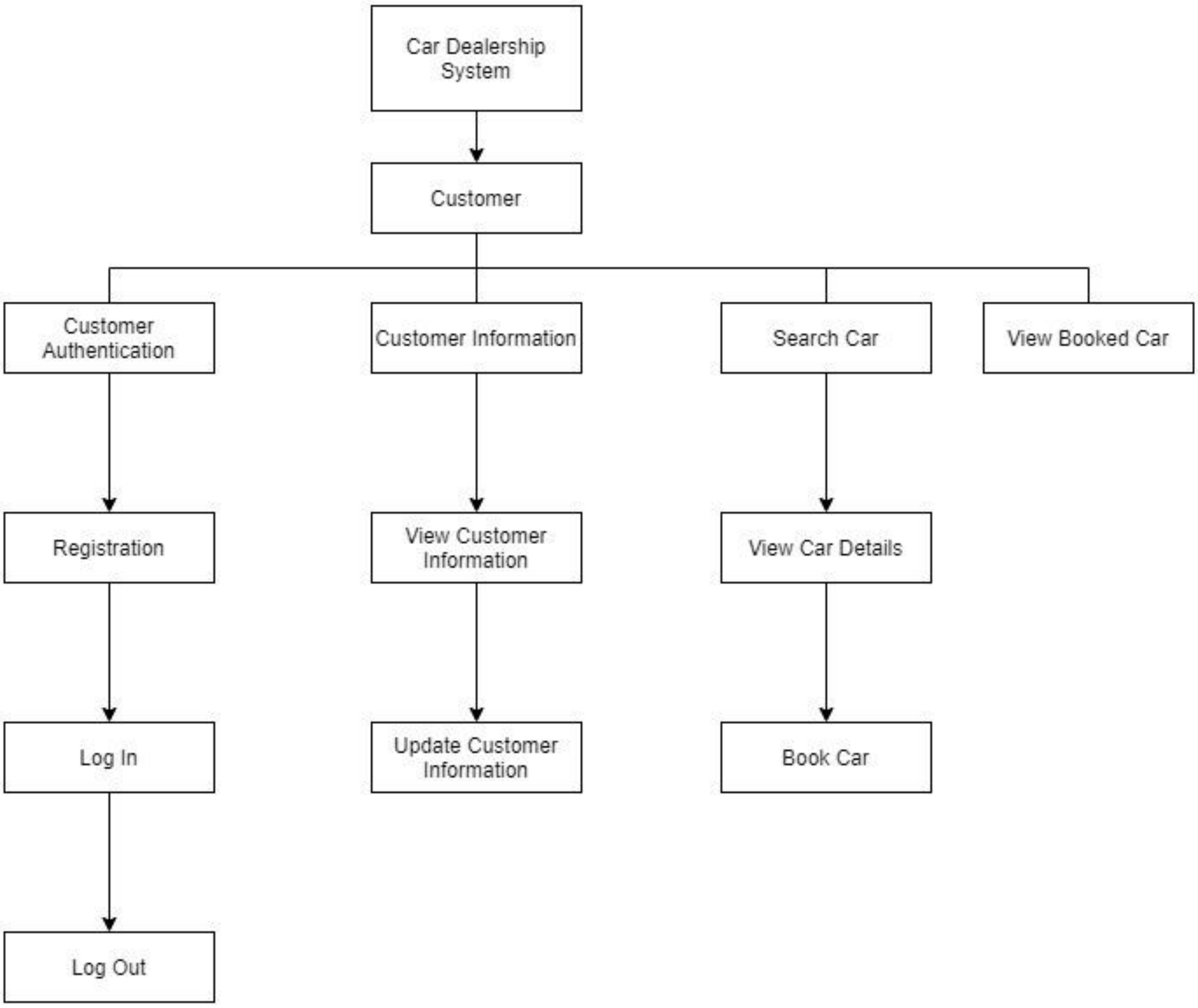


Figure 1.1 *Structured chart Car Dealership System*

CHAPTER 3

DESIGN

3.1 Flowchart

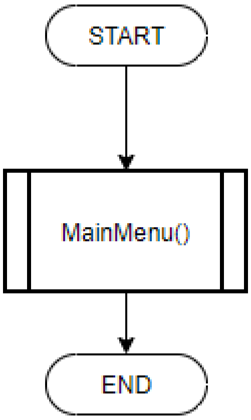


Figure 2.1 Flowchart for main function

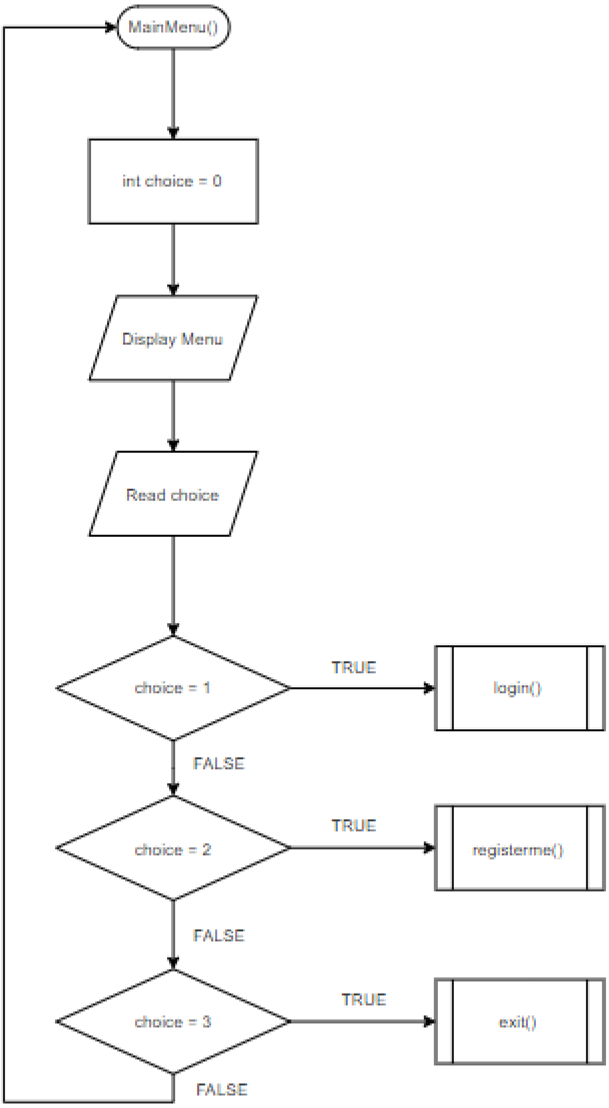


Figure 2.2 Flowchart for MainMenu function

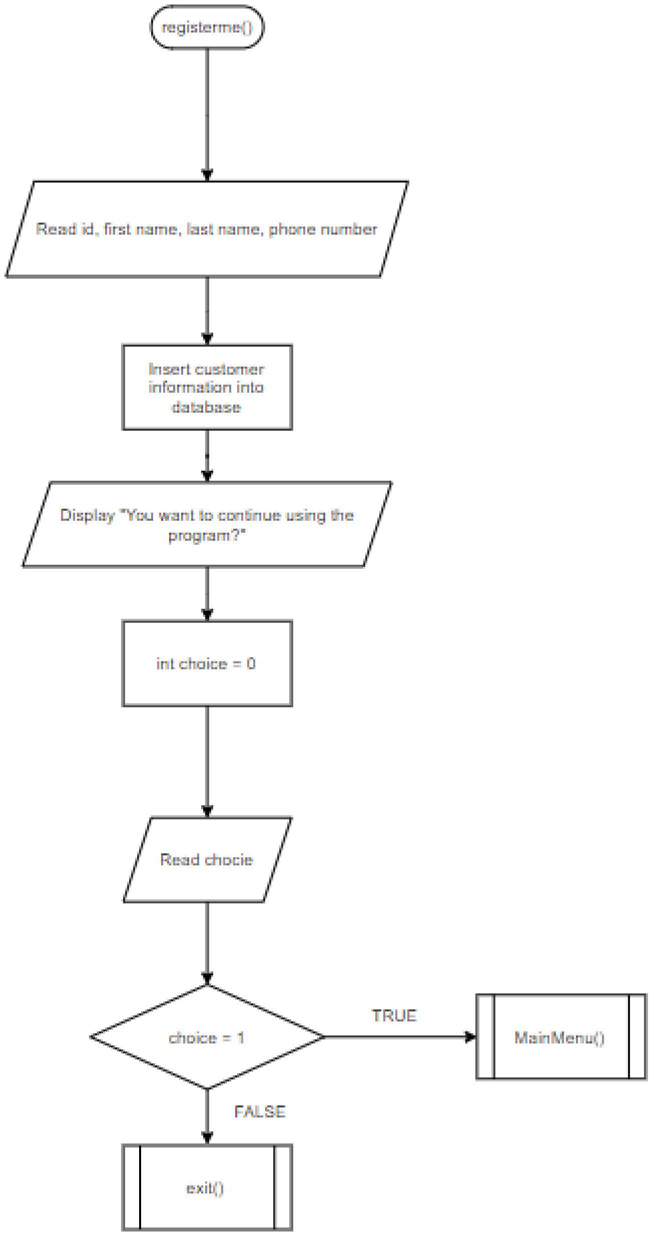

registerme()
Read id, first name, last name, phone number
Insert customer information into database
Display "You want to continue using the program?"
int choice = 0
Read chocie
choice = 1
TRUE
MainMenu()
FALSE
exit()

Figure 2.3 Flowchart for registerme function

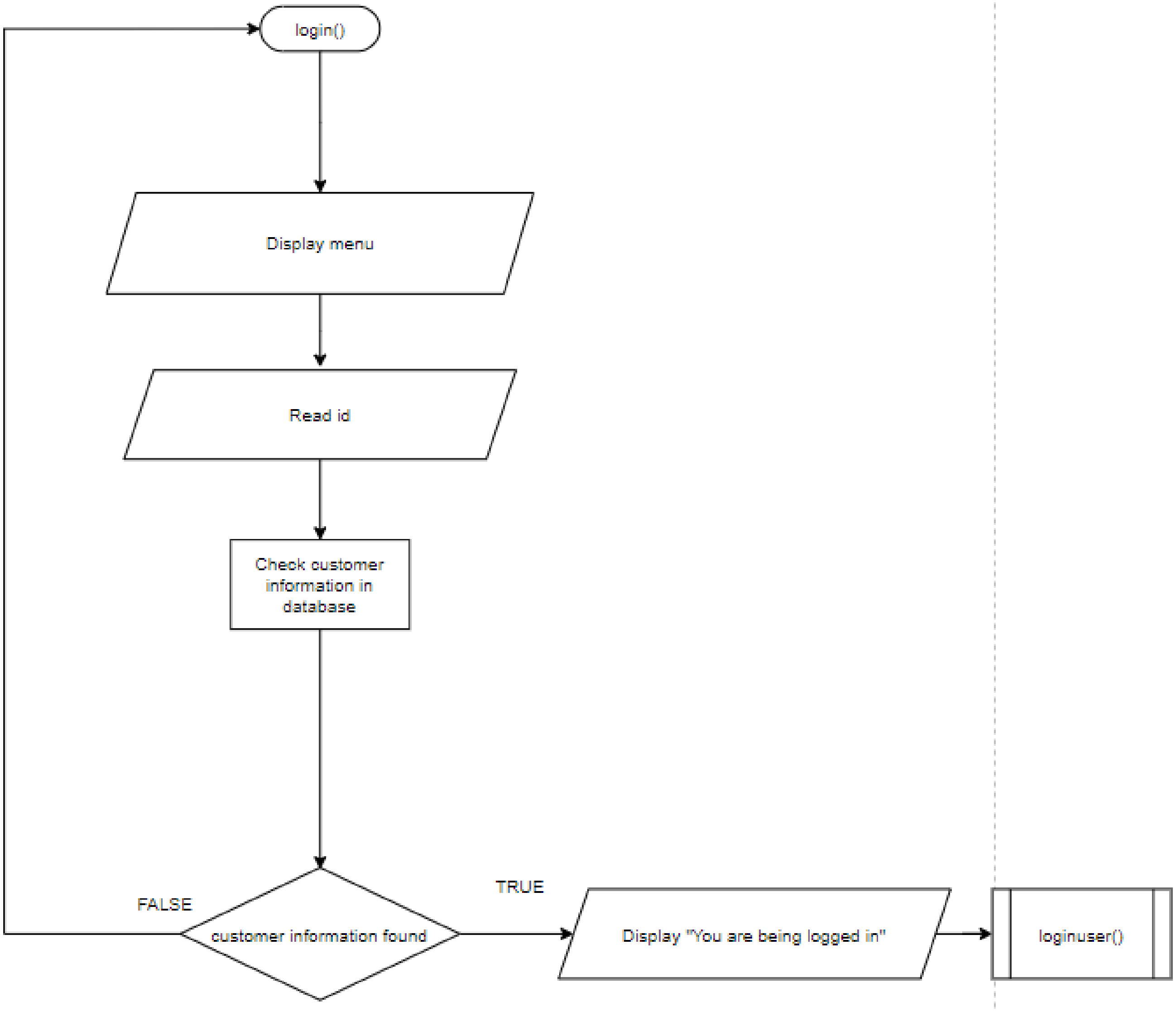

login()
Display menu
Read id
Check customer information in database
customer information found
FALSE
TRUE
Display "You are being logged in"
loginuser()

Figure 2.4 Flowchart for login function

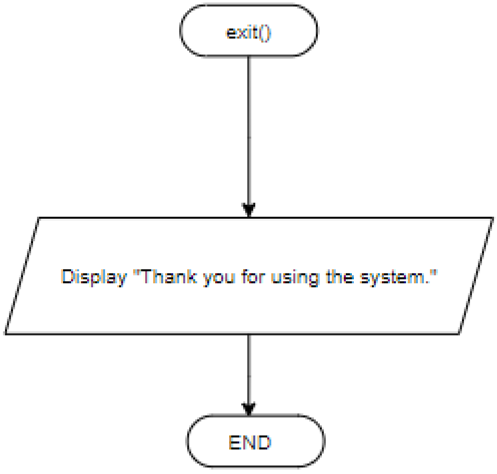


Figure 2.5 Flowchart for exit function

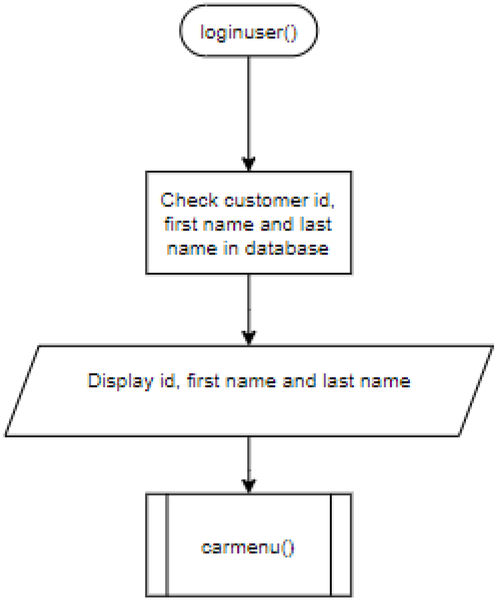

loginuser()
Check customer id, first name and last name in database
Display id, first name and last name
carmenu()

Figure 2.6 Flowchart for loginuser function

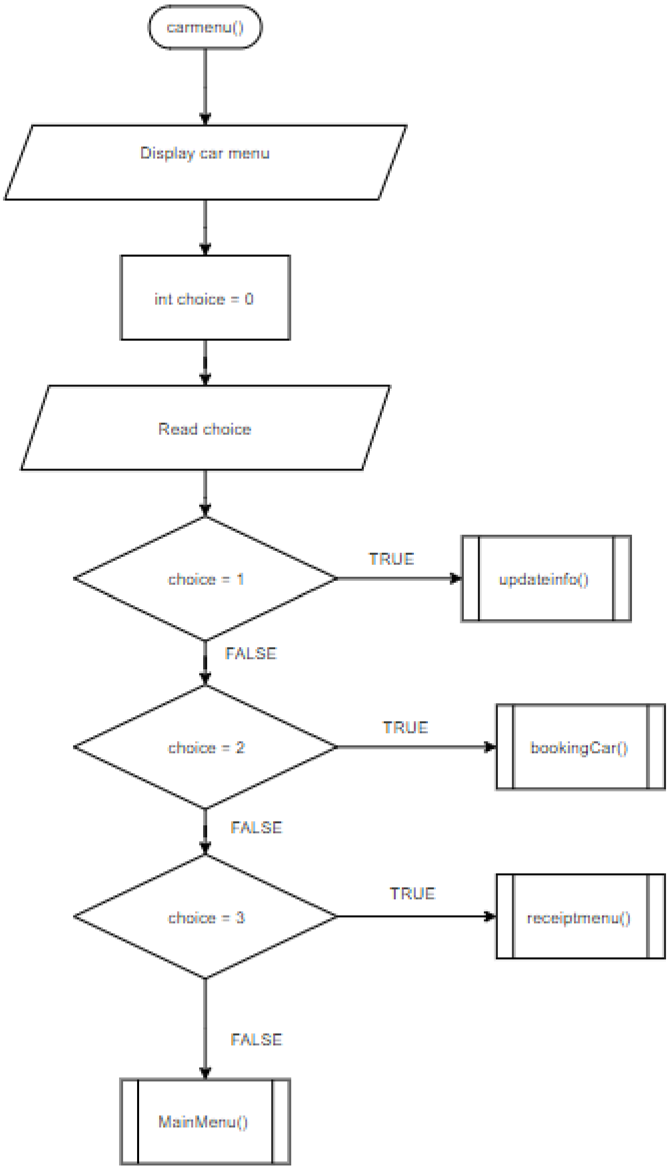


Figure 2.7 Flowchart for carmenu function

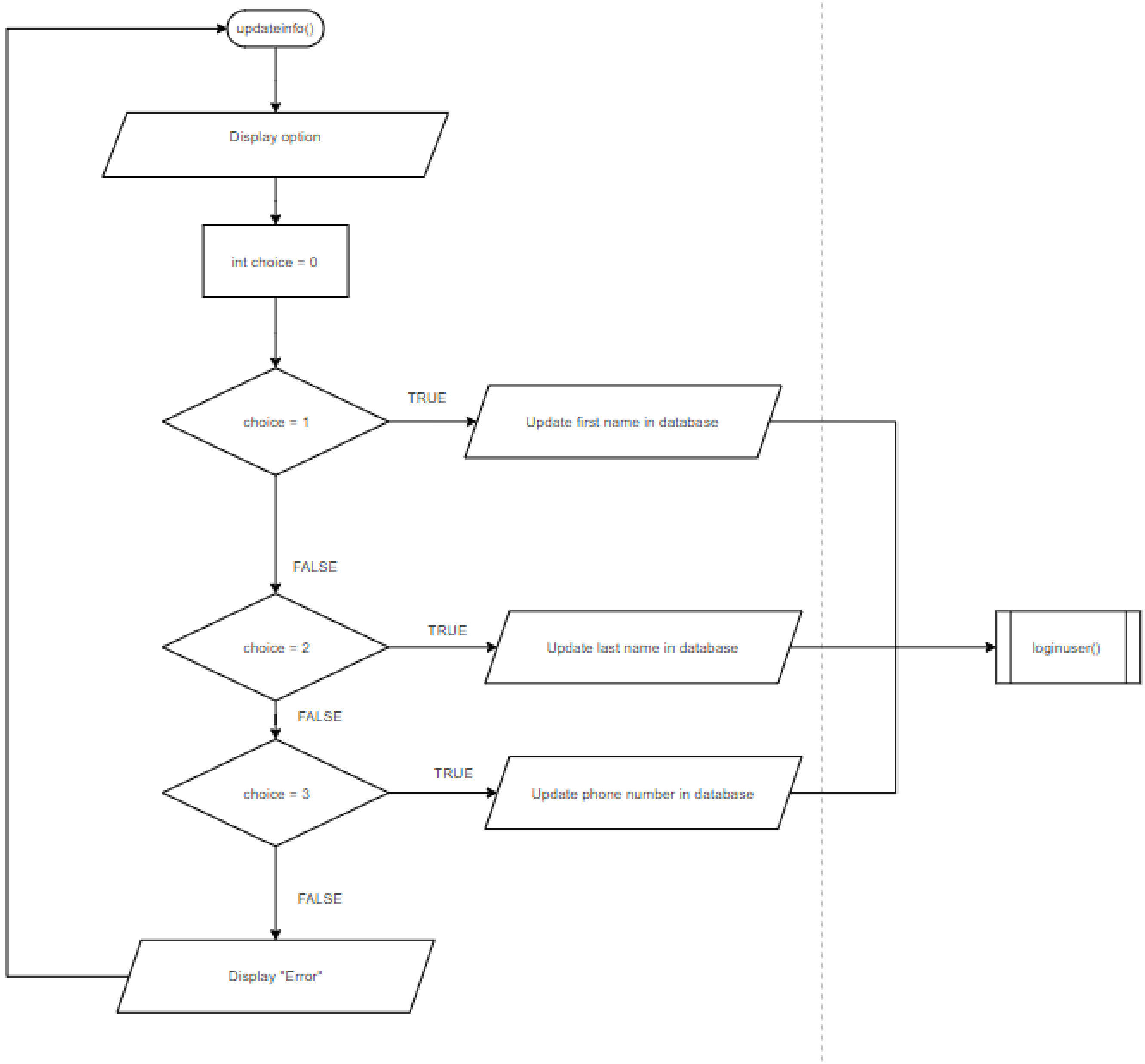


Figure 2.8 Flowchart for updateinfo function

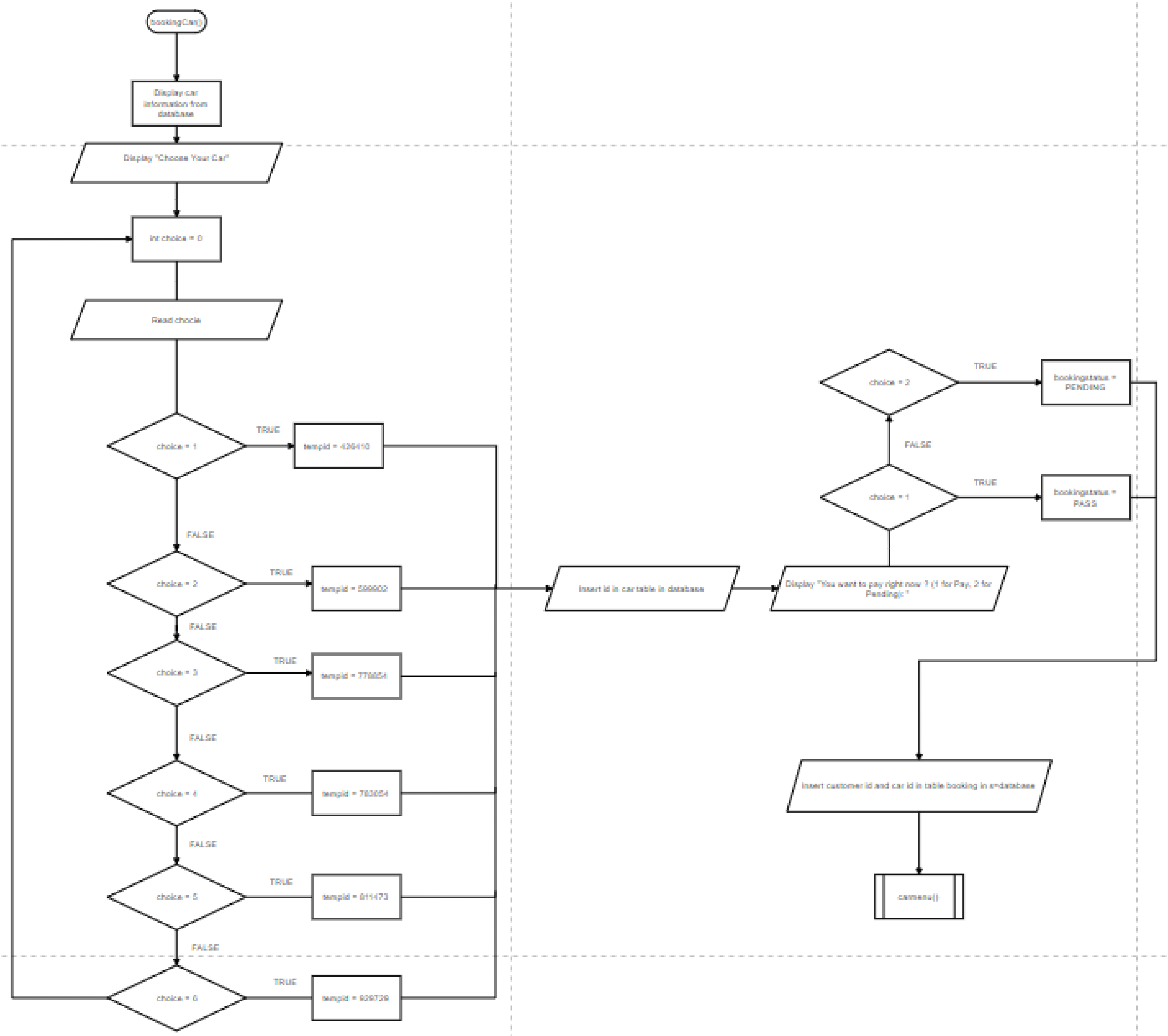


Figure 2.9 Flowchart for bookingcar function

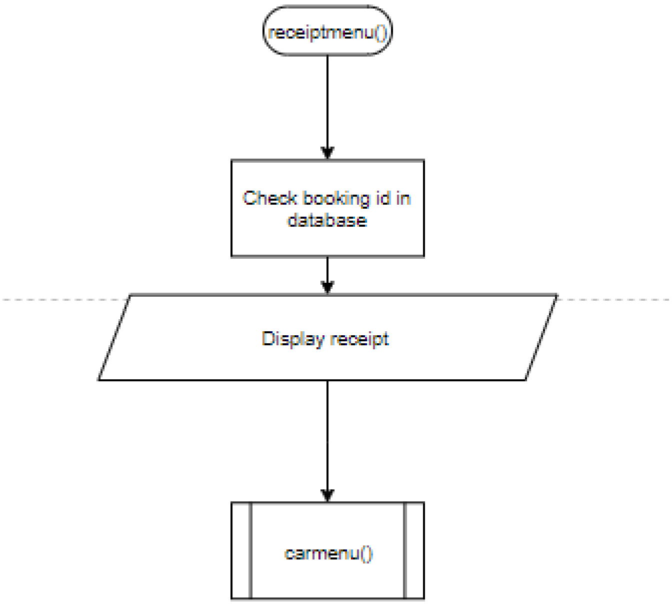


Figure 2.10 Flowchart for receiptmenu function

3.4 Entity relation diagram (ERD)

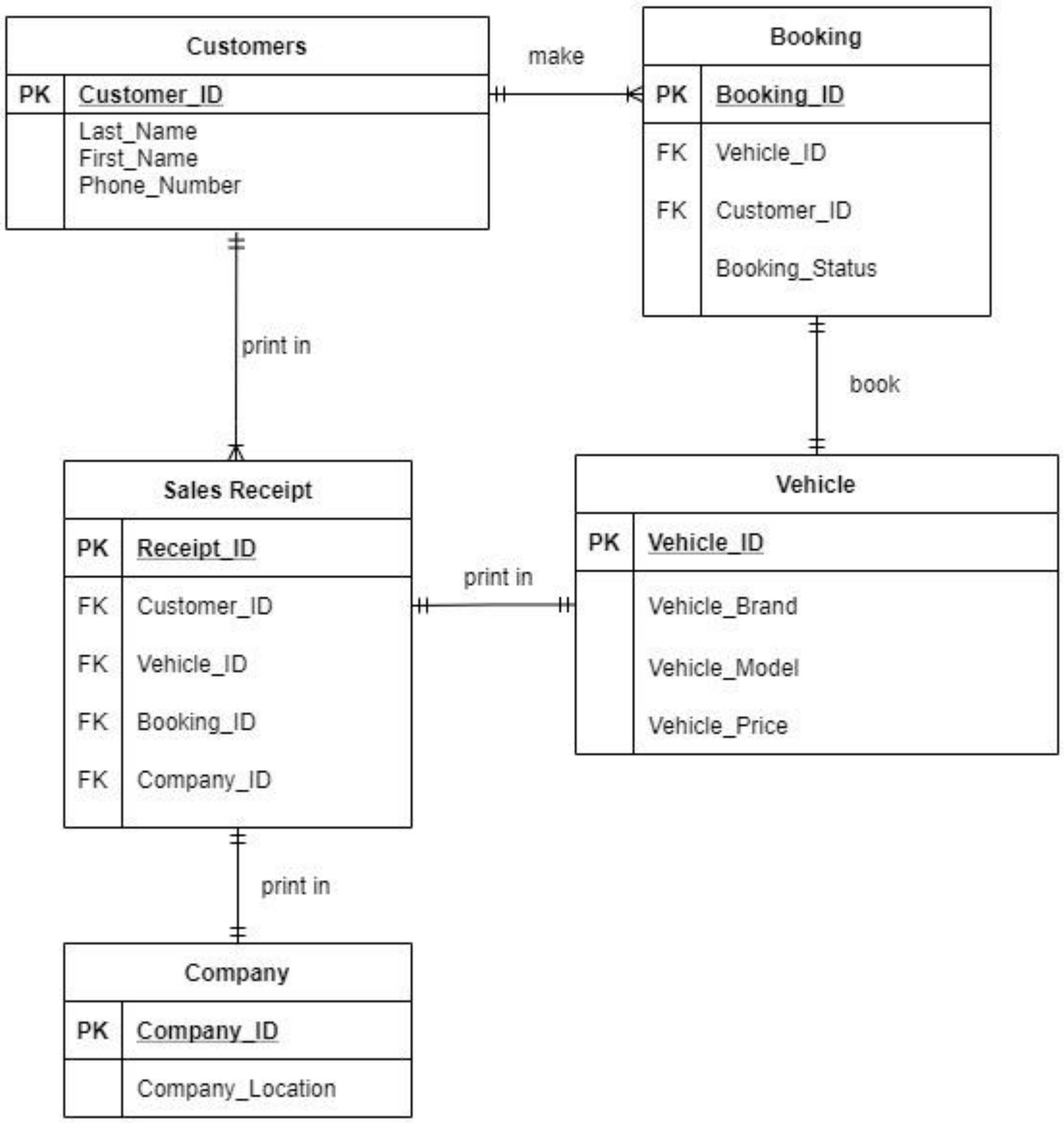


Figure 2.9 *Entity relationship design*

3.5 Data dictionary

Name	Type	Constraint	Description
Customer_ID	VARCHAR(20)	Primary Key	Store Customer ID
First_Name	VARCHAR(20)	Not Null	Store Customer First Name
Last_Name	VARCHAR(20)	Not Null	Store Customer Last Name
Phone_Number	VARCHAR(20)	Not Null	Store Customer Phone Number

Table 1.1 *Data dictionary for table customer*

Name	Type	Constraint	Description
Vehicle_ID	VARCHAR(20)	Primary Key	Store Vehicle ID
Vehicle_Brand	VARCHAR(20)	Not Null	Store Vehicle Brand
Vehicle_Price	DOUBLE	Not Null	Store Vehicle Price
Vehicle_Model	VARCHAR(20)	Not Null	Store Vehicle Model

Table 1.2 *Data dictionary for table vehicle*

Name	Type	Constraint	Description
Location_ID	VARCHAR(20)	Primary Key	Store Location ID
Company_Location	VARCHAR(20)	Null	Store Company Location

Table 1.3 *Data dictionary for table location*

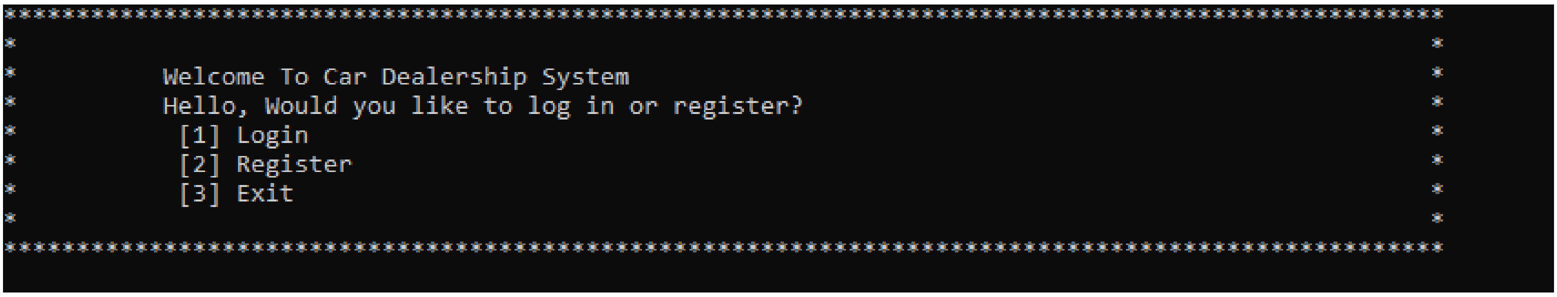
Name	Type	Constraint	Description
Booking_ID	VARCHAR(20)	Primary Key	Store Booking ID
Vehicle_ID	VARCHAR(20)	Foreign Key	Store Vehicle ID
Customer_ID	VARCHAR(20)	Foreign Key	Store Customer ID
Booking_Status	VARCHAR(20)	Not Null	Store Booking Status

Table 1.4 *Data dictionary for table booking*

Name	Type	Constraint	Description
Receipt_ID	VARCHAR(20)	Primary Key	Store Booking ID
Vehicle_ID	VARCHAR(20)	Foreign Key	Store Vehicle ID
Customer_ID	VARCHAR(20)	Foreign Key	Store Customer ID
Booking_ID	VARCHAR(20)	Foreign Key	Store Booking ID
Company_ID	VARCHAR(20)	Foreign Key	Store Location ID

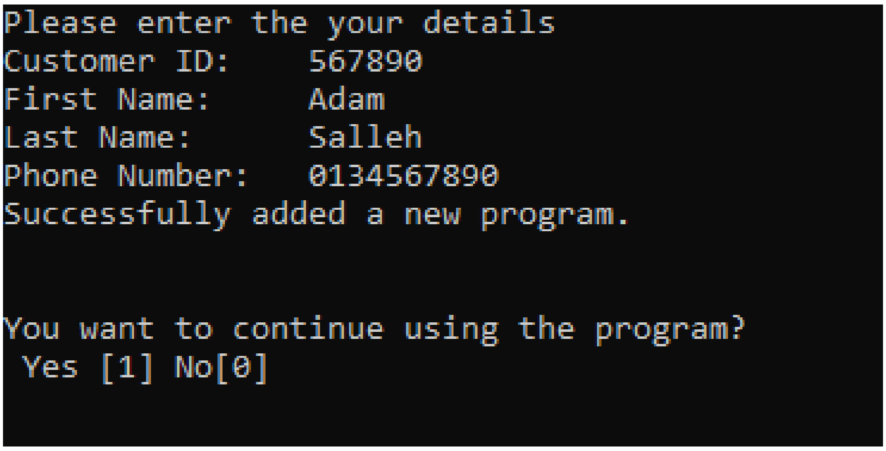
***Table 1.5** Data dictionary for table salesreceipt*

3.6 Interface Design



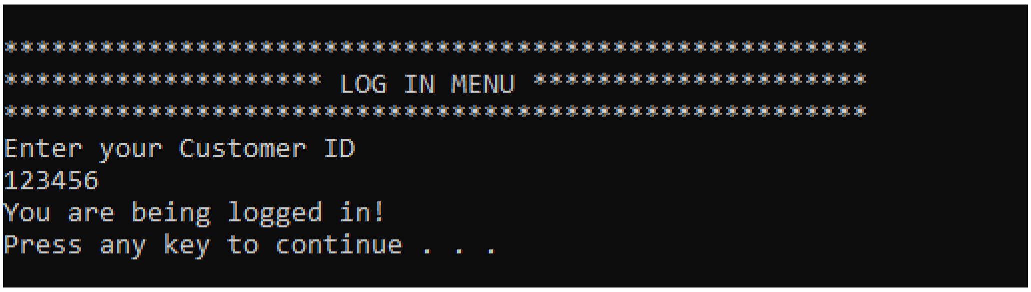
```
**************************************************************************************************
*                                                                                                *
*          Welcome To Car Dealership System                                                      *
*          Hello, Would you like to log in or register?                                          *
*           [1] Login                                                                            *
*           [2] Register                                                                         *
*           [3] Exit                                                                             *
*                                                                                                *
**************************************************************************************************
```

***Figure 3.0** Output for menu*



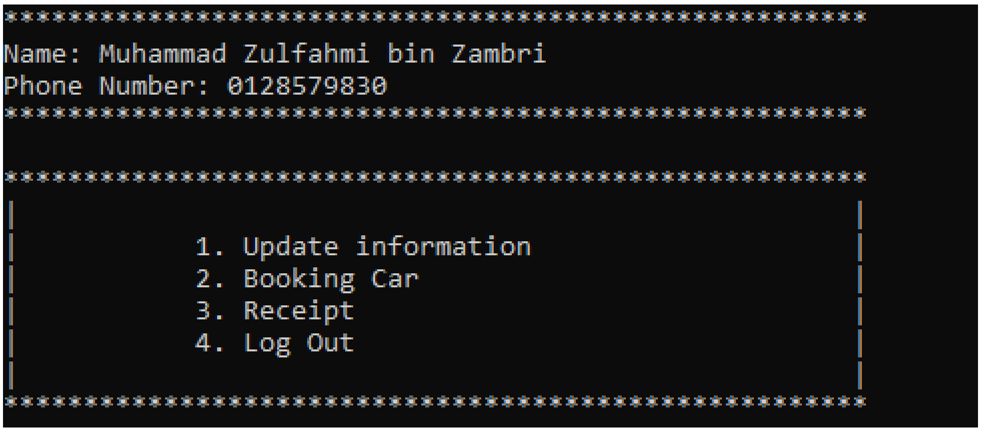
```
Please enter the your details
Customer ID:    567890
First Name:     Adam
Last Name:      Salleh
Phone Number:   0134567890
Successfully added a new program.


You want to continue using the program?
 Yes [1] No[0]
```

***Figure 3.1** Output for customer registration*



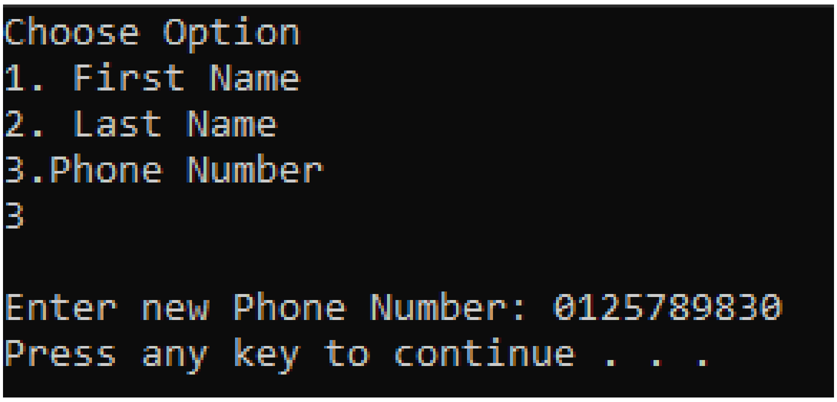
```
******************************************************
******************** LOG IN MENU *********************
******************************************************
Enter your Customer ID
123456
You are being logged in!
Press any key to continue . . .
```

Figure 3.2 *Output for customer login*



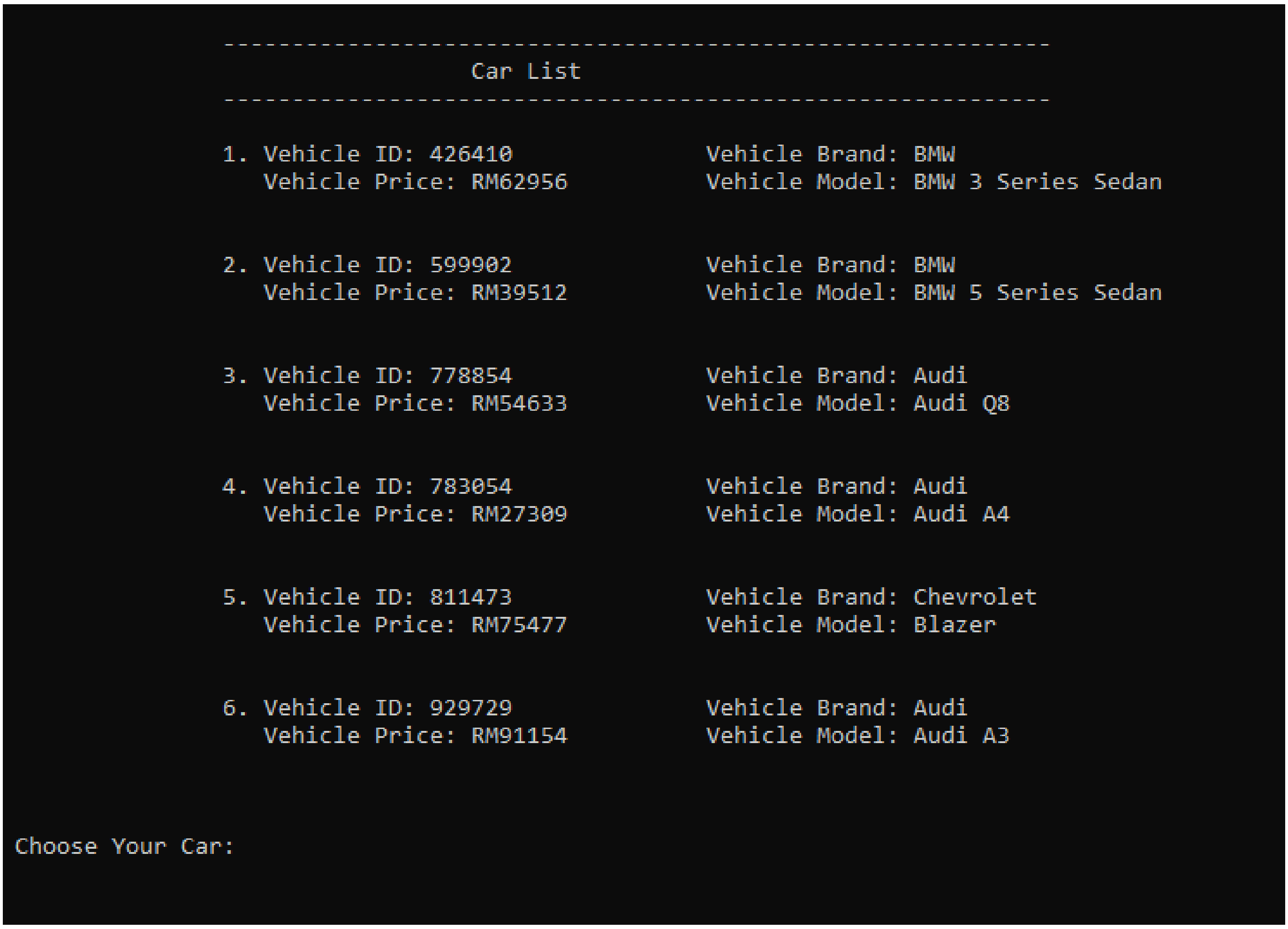
```
******************************************************
Name: Muhammad Zulfahmi bin Zambri
Phone Number: 0128579830
******************************************************

******************************************************
|                                                    |
|           1. Update information                     |
|           2. Booking Car                           |
|           3. Receipt                               |
|           4. Log Out                               |
|                                                    |
******************************************************
```

Figure 3.3 *Output for main menu*



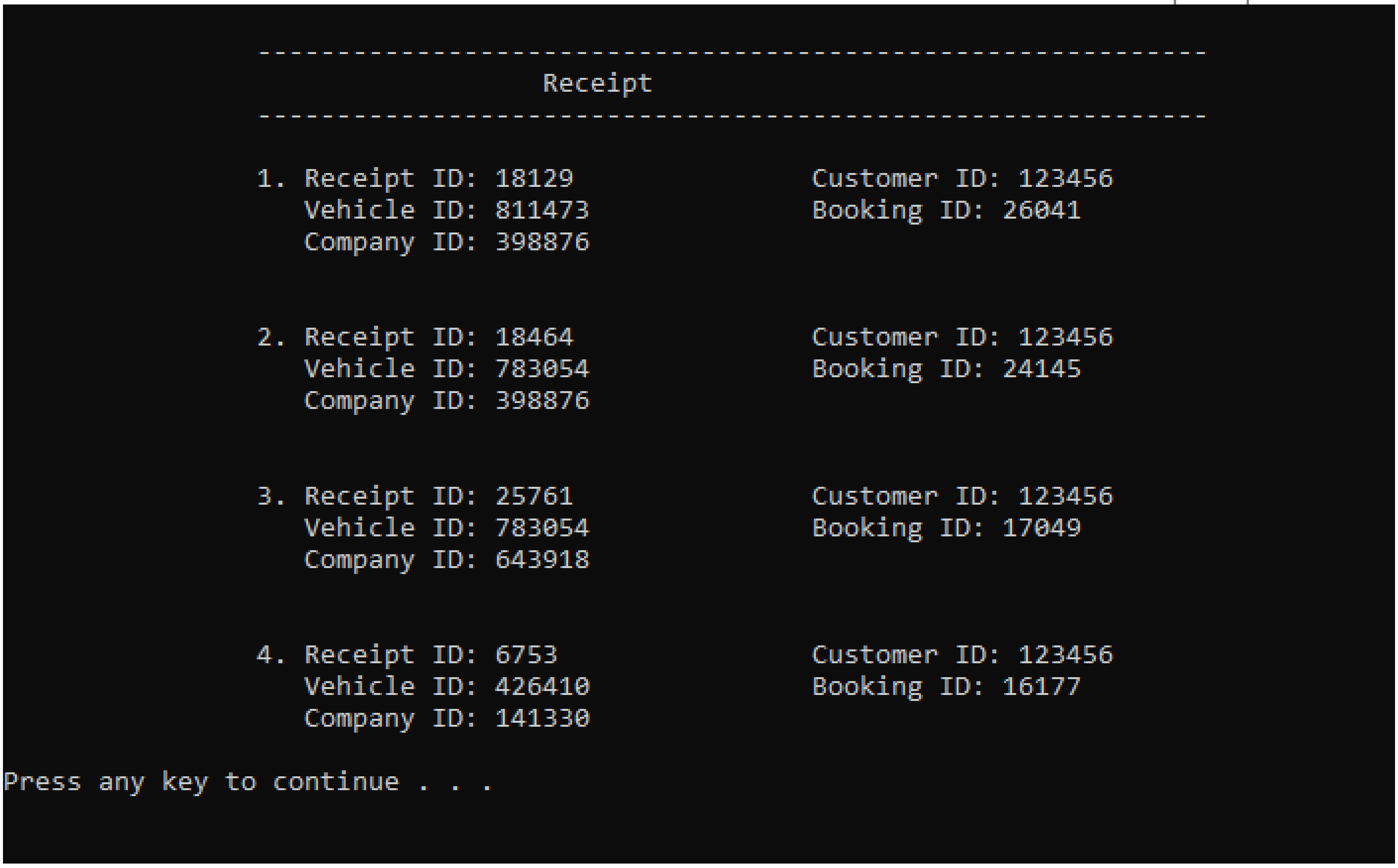
```
Choose Option
1. First Name
2. Last Name
3.Phone Number
3

Enter new Phone Number: 0125789830
Press any key to continue . . .
```

Figure 3.4 *Output for student update information*



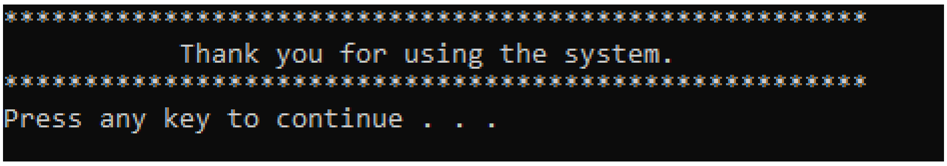
```
                ------------------------------------------------------------
                                  Car List
                ------------------------------------------------------------

                1. Vehicle ID: 426410               Vehicle Brand: BMW
                   Vehicle Price: RM62956           Vehicle Model: BMW 3 Series Sedan


                2. Vehicle ID: 599902               Vehicle Brand: BMW
                   Vehicle Price: RM39512           Vehicle Model: BMW 5 Series Sedan


                3. Vehicle ID: 778854               Vehicle Brand: Audi
                   Vehicle Price: RM54633           Vehicle Model: Audi Q8


                4. Vehicle ID: 783054               Vehicle Brand: Audi
                   Vehicle Price: RM27309           Vehicle Model: Audi A4


                5. Vehicle ID: 811473               Vehicle Brand: Chevrolet
                   Vehicle Price: RM75477           Vehicle Model: Blazer


                6. Vehicle ID: 929729               Vehicle Brand: Audi
                   Vehicle Price: RM91154           Vehicle Model: Audi A3



Choose Your Car:
```

Figure 3.5 *Output for booking menu*



```
                ------------------------------------------------------------
                                  Receipt
                ------------------------------------------------------------

                1. Receipt ID: 18129                Customer ID: 123456
                   Vehicle ID: 811473               Booking ID: 26041
                   Company ID: 398876


                2. Receipt ID: 18464                Customer ID: 123456
                   Vehicle ID: 783054               Booking ID: 24145
                   Company ID: 398876


                3. Receipt ID: 25761                Customer ID: 123456
                   Vehicle ID: 783054               Booking ID: 17049
                   Company ID: 643918


                4. Receipt ID: 6753                 Customer ID: 123456
                   Vehicle ID: 426410               Booking ID: 16177
                   Company ID: 141330

Press any key to continue . . .
```

Figure 3.6 *Output for receipt menu*



```
******************************************************
           Thank you for using the system.
******************************************************
Press any key to continue . . .
```

Figure 3.7 *Output for exit menu*

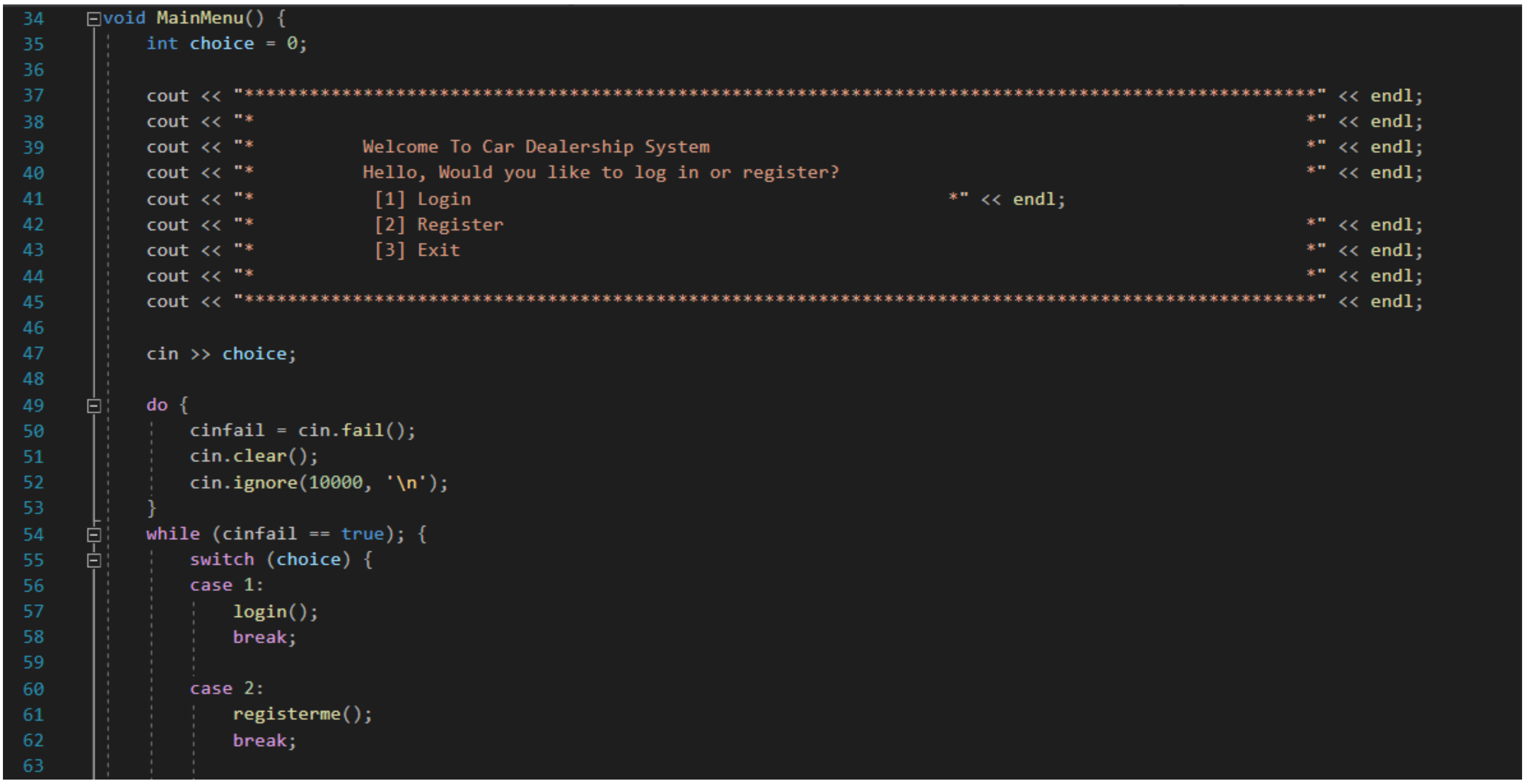
CHAPTER 4

IMPLEMENTATION

The programming language used to implement this system is C++. The database used is PhpMyAdmin (MySql) database in the xampp (localhost). To develop this system, there are a few programming technique that is applied such as function, selection, loop, pointer control and error handling. The programming technique that apply to this system are show below.

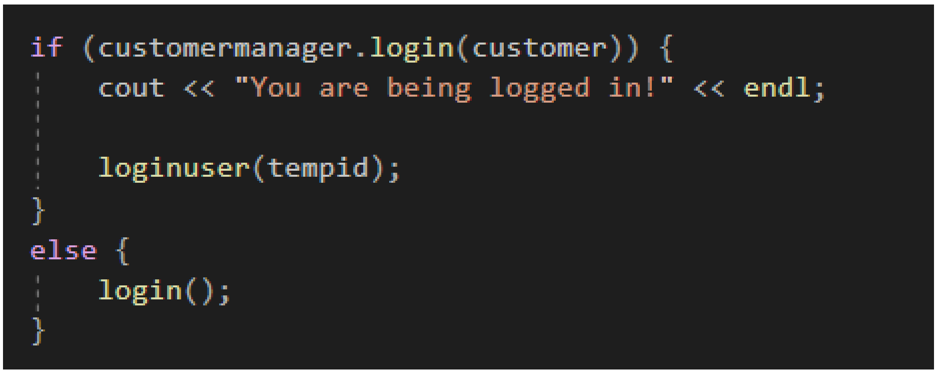
4.1 Programming technique

i. Function



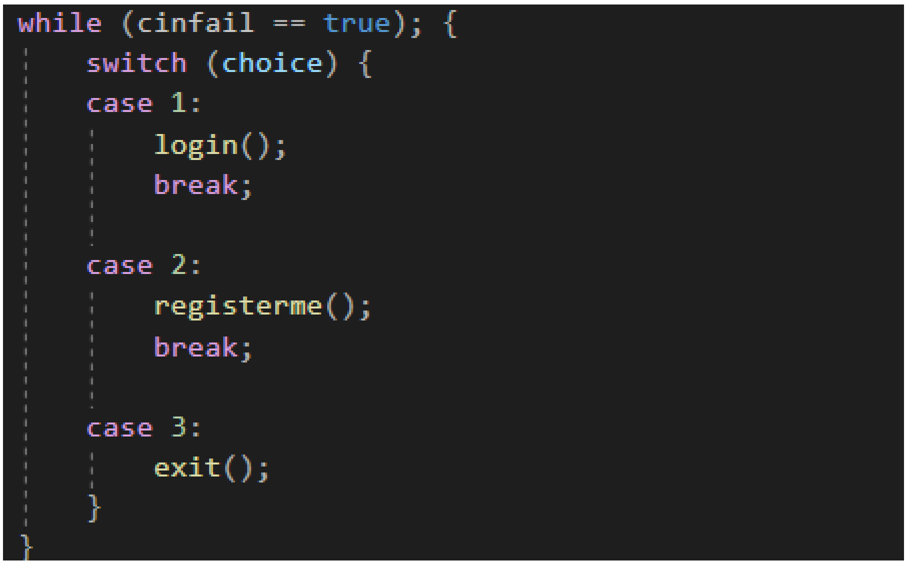
```
void MainMenu() {
    int choice = 0;

    cout << "**************************************************************************************************" << endl;
    cout << "*                                                                                                *" << endl;
    cout << "*          Welcome To Car Dealership System                                                      *" << endl;
    cout << "*          Hello, Would you like to log in or register?                                           *" << endl;
    cout << "*           [1] Login                                           *" << endl;
    cout << "*           [2] Register                                                                         *" << endl;
    cout << "*           [3] Exit                                                                             *" << endl;
    cout << "*                                                                                                *" << endl;
    cout << "**************************************************************************************************" << endl;

    cin >> choice;

    do {
        cinfail = cin.fail();
        cin.clear();
        cin.ignore(10000, '\n');
    }
    while (cinfail == true); {
        switch (choice) {
        case 1:
            login();
            break;

        case 2:
            registerme();
            break;

```

ii. Selection



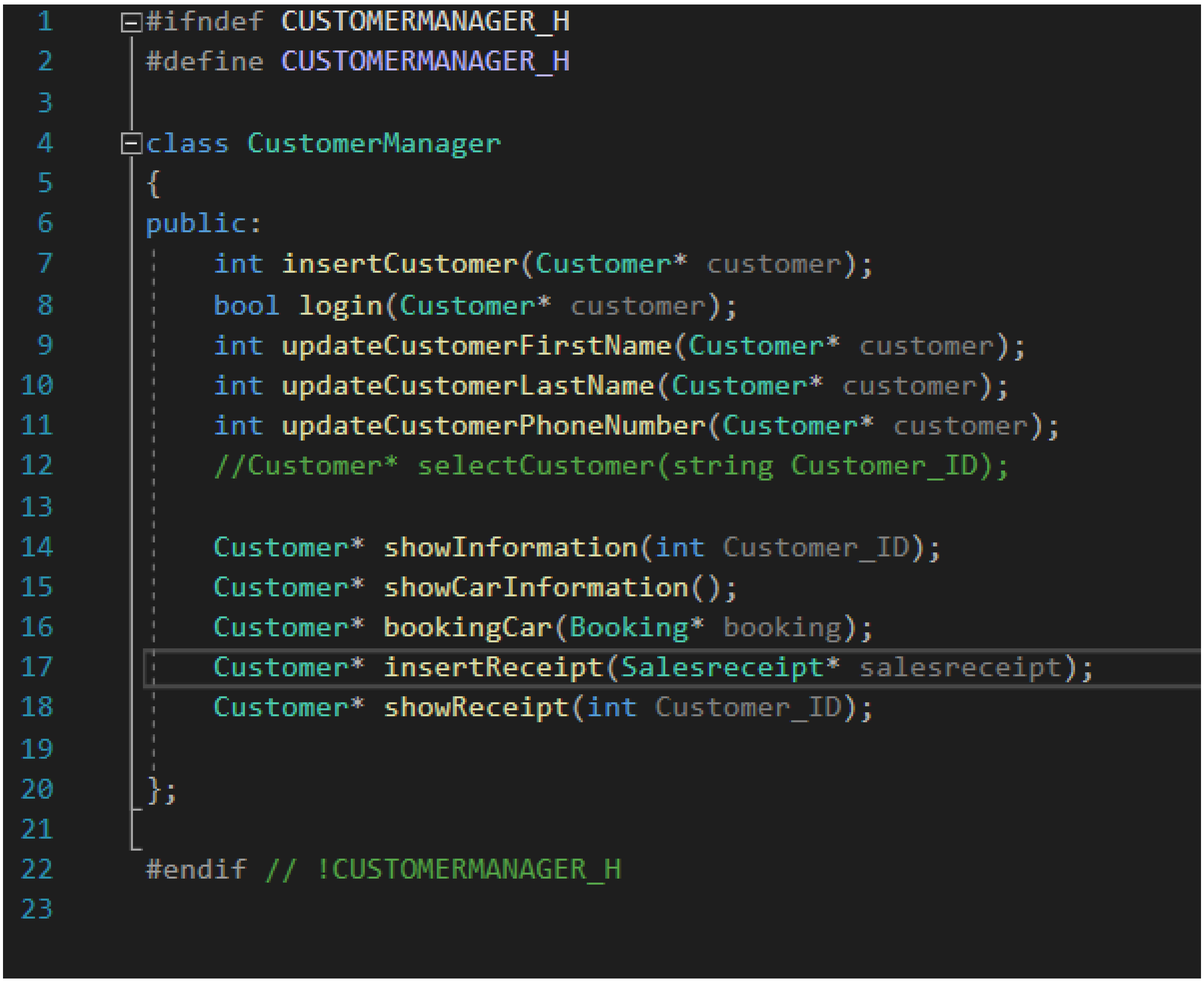
```
if (customermanager.login(customer)) {
    cout << "You are being logged in!" << endl;

    loginuser(tempid);
}
else {
    login();
}
```

iii. Loop



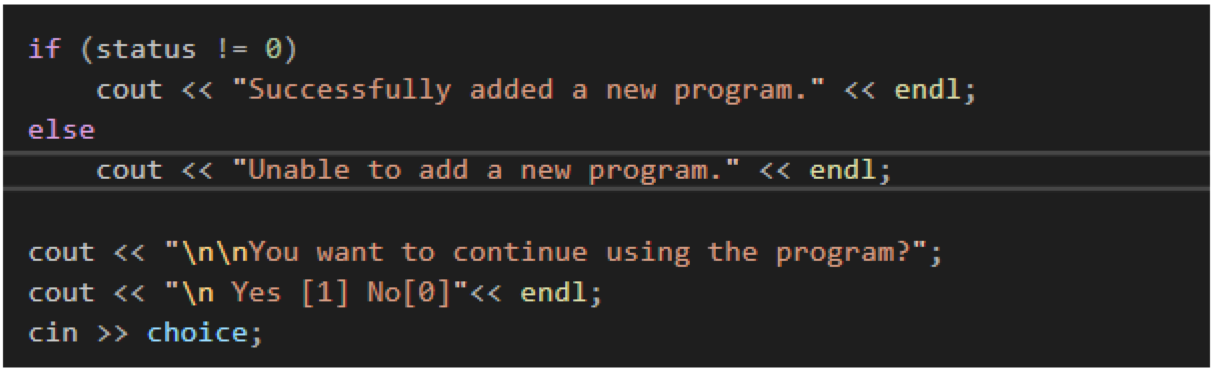
```
while (cinfail == true); {
    switch (choice) {
    case 1:
        login();
        break;

    case 2:
        registerme();
        break;

    case 3:
        exit();
    }
}
```

iv. Class



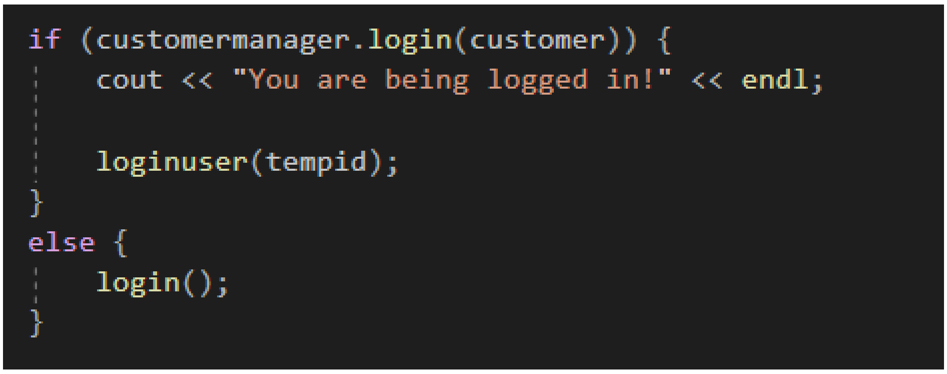
```
#ifndef CUSTOMERMANAGER_H
#define CUSTOMERMANAGER_H

class CustomerManager
{
public:
    int insertCustomer(Customer* customer);
    bool login(Customer* customer);
    int updateCustomerFirstName(Customer* customer);
    int updateCustomerLastName(Customer* customer);
    int updateCustomerPhoneNumber(Customer* customer);
    //Customer* selectCustomer(string Customer_ID);

    Customer* showInformation(int Customer_ID);
    Customer* showCarInformation();
    Customer* bookingCar(Booking* booking);
    Customer* insertReceipt(Salesreceipt* salesreceipt);
    Customer* showReceipt(int Customer_ID);

};

#endif // !CUSTOMERMANAGER_H
```

v. Control



```
if (status != 0)
    cout << "Successfully added a new program." << endl;
else
    cout << "Unable to add a new program." << endl;

cout << "\n\nYou want to continue using the program?";
cout << "\n Yes [1] No[0]"<< endl;
cin >> choice;
```

vi. Error Handling

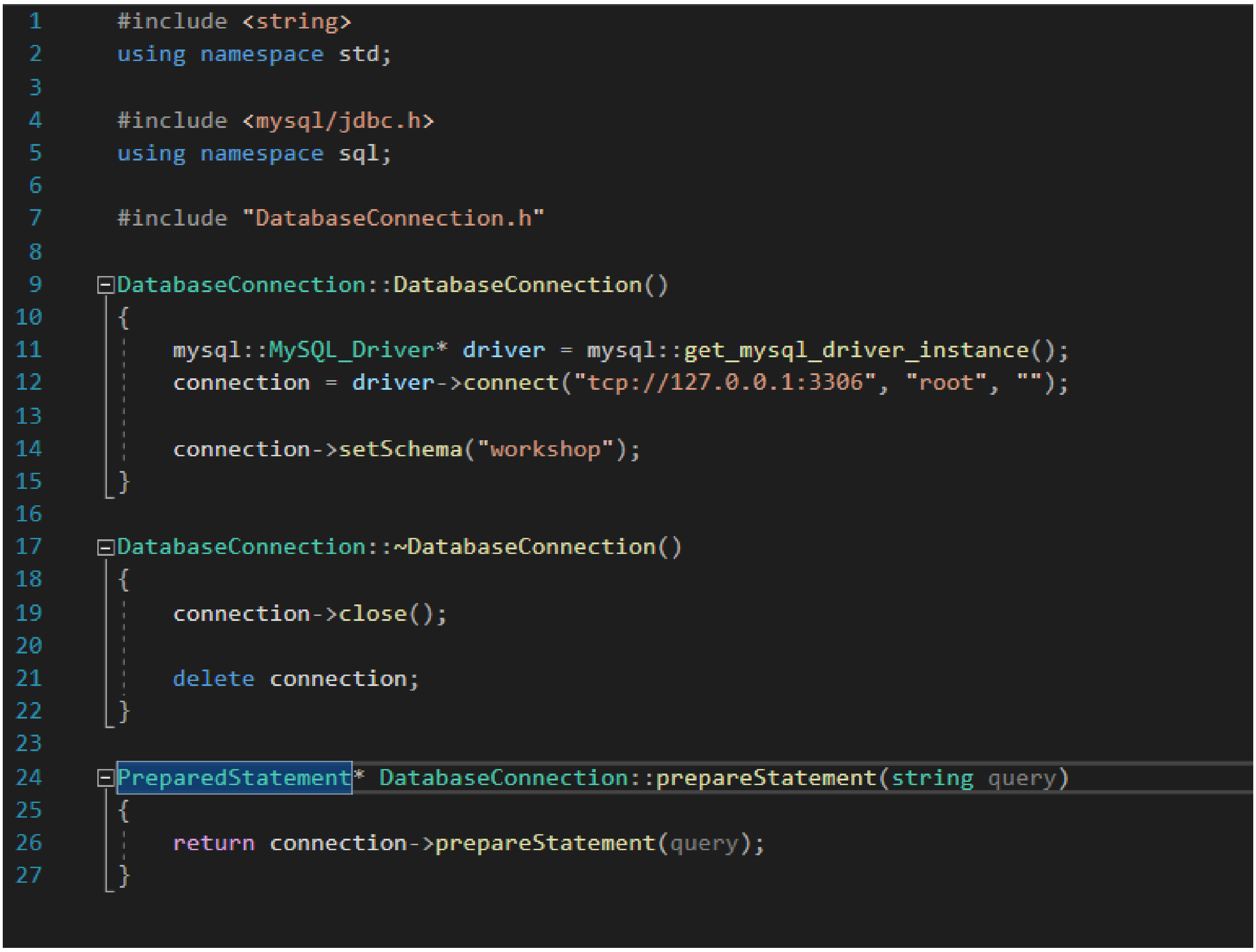


```
if (customermanager.login(customer)) {
    cout << "You are being logged in!" << endl;

    loginuser(tempid);
}
else {
    login();
}
```

4.2 Database implementation

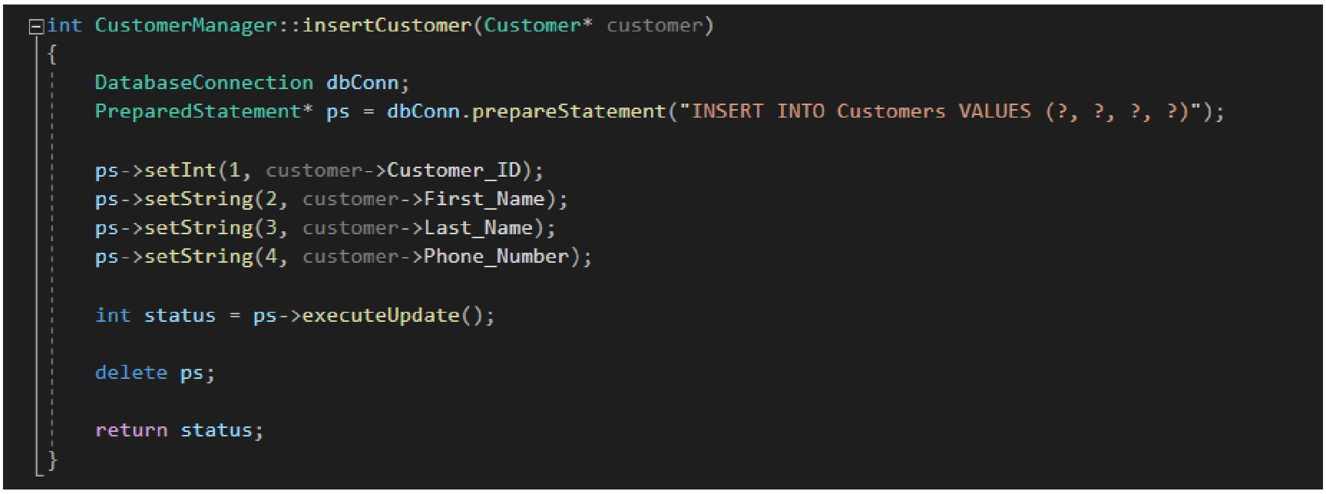
The implementation of this system is depends on the cooperation between php and Mysql to implement all those function such as insert, update, delete and also select to list out the data that inserted in database.

i. Database connection



```
#include <string>
using namespace std;

#include <mysql/jdbc.h>
using namespace sql;

#include "DatabaseConnection.h"

DatabaseConnection::DatabaseConnection()
{
    mysql::MySQL_Driver* driver = mysql::get_mysql_driver_instance();
    connection = driver->connect("tcp://127.0.0.1:3306", "root", "");

    connection->setSchema("workshop");
}

DatabaseConnection::~DatabaseConnection()
{
    connection->close();

    delete connection;
}

PreparedStatement* DatabaseConnection::prepareStatement(string query)
{
    return connection->prepareStatement(query);
}
```

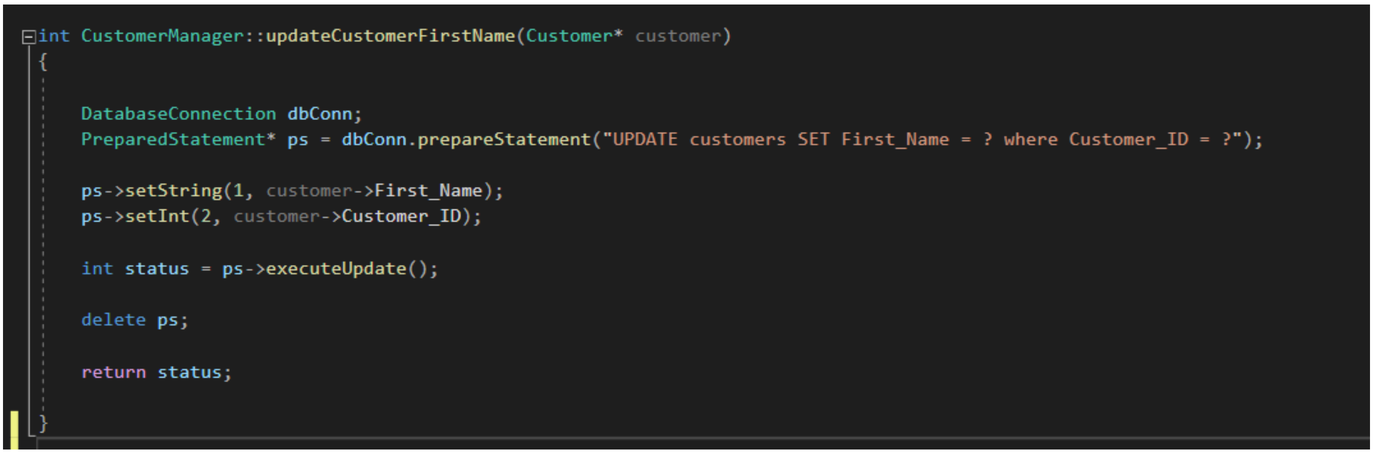
ii. Insert into database



```
int CustomerManager::insertCustomer(Customer* customer)
{
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("INSERT INTO Customers VALUES (?, ?, ?, ?)");

    ps->setInt(1, customer->Customer_ID);
    ps->setString(2, customer->First_Name);
    ps->setString(3, customer->Last_Name);
    ps->setString(4, customer->Phone_Number);

    int status = ps->executeUpdate();

    delete ps;

    return status;
}
```

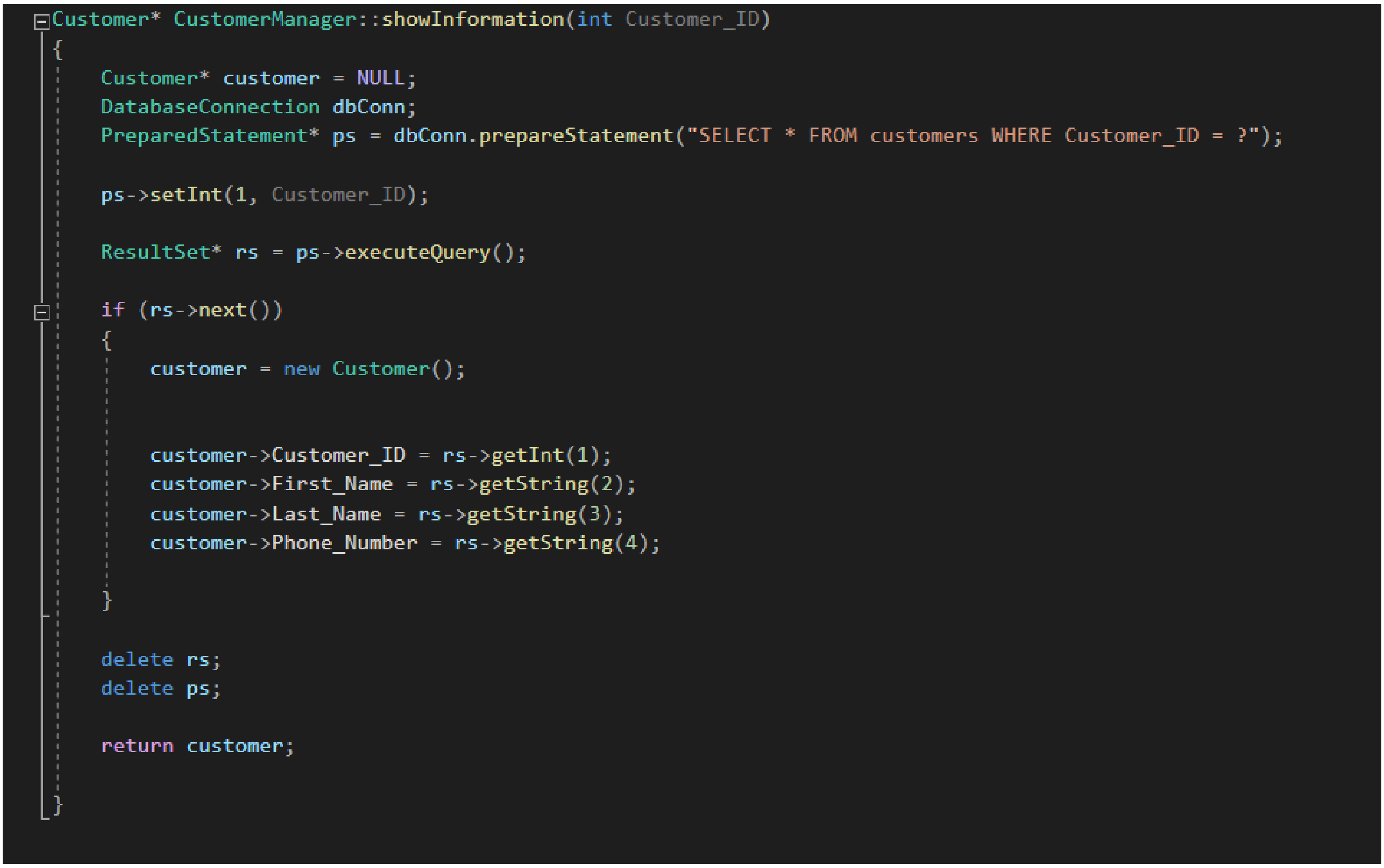
iii.

iii. Update database



```
int CustomerManager::updateCustomerFirstName(Customer* customer)
{

    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("UPDATE customers SET First_Name = ? where Customer_ID = ?");

    ps->setString(1, customer->First_Name);
    ps->setInt(2, customer->Customer_ID);

    int status = ps->executeUpdate();

    delete ps;

    return status;

}
```

iv. Select database

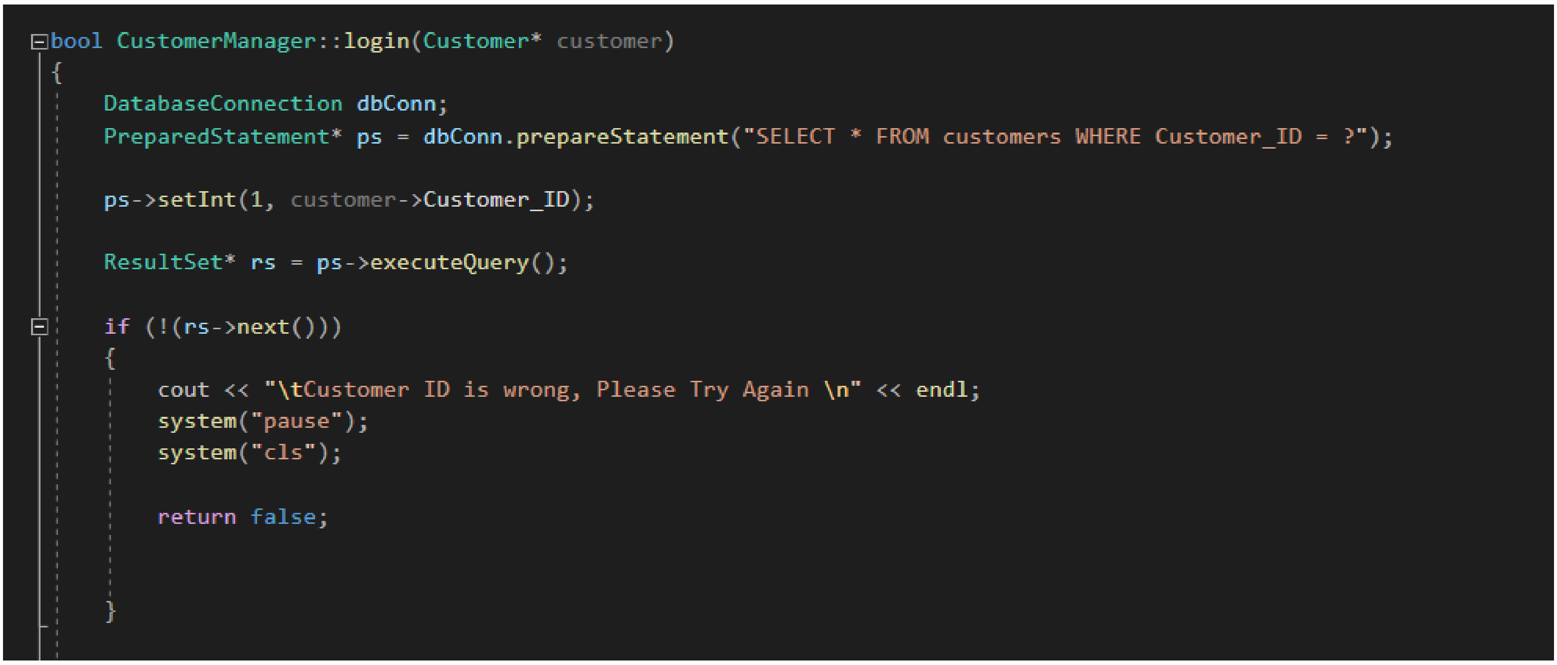


```
Customer* CustomerManager::showInformation(int Customer_ID)
{
    Customer* customer = NULL;
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("SELECT * FROM customers WHERE Customer_ID = ?");

    ps->setInt(1, Customer_ID);

    ResultSet* rs = ps->executeQuery();

    if (rs->next())
    {
        customer = new Customer();


        customer->Customer_ID = rs->getInt(1);
        customer->First_Name = rs->getString(2);
        customer->Last_Name = rs->getString(3);
        customer->Phone_Number = rs->getString(4);

    }

    delete rs;
    delete ps;

    return customer;

}
```

4.3 Security implementation

The problem that the user also encounter the security problem due to lack of security. For now, we implement the security feature in this system to avoid the unauthorized access.

i. Login security



```
bool CustomerManager::login(Customer* customer)
{
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("SELECT * FROM customers WHERE Customer_ID = ?");

    ps->setInt(1, customer->Customer_ID);

    ResultSet* rs = ps->executeQuery();

    if (!(rs->next()))
    {
        cout << "\tCustomer ID is wrong, Please Try Again \n" << endl;
        system("pause");
        system("cls");

        return false;


    }
```

CHAPTER 5

CONCLUSION

5.1 Conclusion

As conclusion, the system will help many customer can create and manage their account and can update their information. Besides, customers can have the information about the car, can check car information and book a car that is convenient with their needs. Finally, the system able to resolve the loss of data and store it in a more organized way. This undertaking expects to build up Car Database System, which will easily enable customer to manage the process in booking a car without helping from the staff.

5.2 Constraint

The limitation for this system is the system backup. For now, this system still unable to implement the system backup since the system is in development. Besides, there are other limitation in this system, which is the availability of the system. Customer needs to go to the kiosk at automotive company to access the system. Since the program developed in C++ only, it doesn’t have any security measure such as network security.

5.3 Further Enhancements

Enhancements that can be made based on my existing system few future. I would like to improve which customers can access this program using their mobile apps since nowadays a lot of system has been made into mobile apps, which are more convenient to user. Besides, in the future I would like to develop a graphical user interface for this program since it runs on console that the program don’t have an interactive user interface. Next, I would develop a guide feature so a new user can get to know how to use the system much easier.

REFERENCE

1. https://www.researchgate.net/publication/306077555_Design_and_Implementation_of_Computerizing_the_Dealership_Management_Software

2. https://projectsgeek.com/2016/02/car-sales-system-project-in-java.html

3. Noraswaliza Abdullah, Mashanum Osman, Zeratul Izzah Mohd Yusoh, Aniza Othman, Zarita Mohd Kosnin. (2019). Computer Programming. Durian Tunggal, Melaka : FTMK Universiti Teknikal Malaysia Melaka