UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

WORKSHOP 1

FINAL REPORT

NAME	NURIN SYAFIQAH BINTI MOHD SAOFI
MATRIC NUMBER	B031910122
COURSE	BACHELOR OF COMPUTER SCIENCE (SOFTWARE)
PROJECT TITLE	ACCIDENT INFORMATION PROVIDERS SYSTEM
SUPERVISOR NAME	TS. DR. LIZAWATI BINTI SALAHUDDIN
SUPERVISOR SIGNATURE	
EVALUATOR NAME	MISS EMALIANA BINTI KASMURI

Table of Contents

CHAPTER 1: INTRODUCTION 1

1.1 Introduction 1

1.2 Problem Statement 1

1.3 Objective 1

1.4 Scope 2

1.5 Conclusion 2

CHAPTER 2: PROBLEM ANALYSIS 3

2.1 Introduction 3

2.2 Detailed Description of The Problem 3

2.3 Structure Chart 4

2.4 Conclusion 5

CHAPTER 3: DESIGN 6

3.1 Introduction 6

3.2 Flowchart 7

3.3 Entity Relationship Diagram 22

3.4 Data Dictionary 23

3.5 Interface Design 25

CHAPTER 4: IMPLEMENTATION 35

4.1 Introduction 35

4.2 Programming Techniques 35

4.3 User Interface 38

4.4 System Testing 53

CHAPTER 5: CONCLUSION 55

5.1 Project Summarization 55

5.2 Strength and Weaknesses 56

5.3 Problem Solved 56

5.4 Suggestions for Improvement 57

5.5 Conclusion 57

REFERENCES 58

CHAPTER 1: INTRODUCTION

1.1 Introduction

This system is to be used by police in Malaysia to write report about accident. Based on Mohd Azmi Abdul Hamid (2019, Jun 21) found that Malaysia will be one of the most accident occurs after Thailand and South Africa based on report Bloomberg (2017, June 20) recorded World Health Organisations (WHO) statistics for 2013. As we know, one of the polices jobs is to handle all accident cases in Malaysia. So, by this system will help all police to add new report of accident and change their information easily. Besides that, the system also provides the police to search and observe the report of accident. The current application process is done manually by all polices in Malaysia. The proposed system will ease the application process via online application.

1.2 Problem Statement

The problem is divided into several parts as follows:

- Difficulty of finding the location of a certain report.
- Retrieving a hardcopy data is time consuming.
- Wasting time for management officer to fill a new police or update policer profile.
- Hard to know the exact number of accidents occurs in Malaysia.

1.3 Objective

This project embarks on the following objectives:

- To ease get the information about the accident and report.
- To ease create a new profile or update police information.
- To fasten the process of checking numbers of accident in Malaysia.

1.4 Scope

1. Modules to be developed:

   - Register module
   - Login module
   - Create module
   - Display module
   - Search module
   - Update module
   - Delete module
   - Calculation module

2. Target user:

   - Police

1.5 Conclusion

This chapter describes the introduction, problem statement, objective, scope and significant of study. This chapter is to identify the problem of the previous system and make an objective to build a new system. It also states what are the scope of the system such as the system users and modules. Lastly, it describes the importance of researching about this project.

The output from this chapter is used to elaborate and analyse the problem statement that has been stated above. The structure chart of the system will also be discussed in the next chapter.

CHAPTER 2: PROBLEM ANALYSIS

2.1 Introduction

This chapter is consisting of detailed description of the problem and structure chart. In detailed description of the problem, the problem statement in Chapter 1 will decompose into parts that are easier to conceive, understand, program, and maintain. Meanwhile, structure chart shows the breakdown of Accident Information Providers System to its lowest manageable levels.

2.2 Detailed Description of The Problem

The reason why this application system needs to be developed because police officer needs to write a lot of report requests in one day. In the current system, the police need to check the report id by looking the lasts report in file room and then go to registration table to write a new report details and they need to double check the details so that there is no information left behind. Applicant must wait for the confirmation before leave and this is time consuming.

The second reason difficult for police officer to search any data in the right places at file room. To make a new report about accident, police officer needs to print the data in on a piece of paper and keep it in the file. As we know, at police station, there will be a lot of file which include for other cases. So, when police put the report in file, and later they want to use again the report, they need to find one by one in the file room. This process is not efficient and takes a lot of time consuming (up to 1 hour).

The third reason is because this system will provide a platform where new policer can register their profile and get the id number from the system. Then, they can use the id and password to login to this system. Beside that, they also can update their profile at any time. In the current system, new policer needs to go to office and write the form to get id number. Then if they move to a new place, they need to register again at new office. They also need to fill up the form to update any information like number telephone, address, or others personal information.

The fourth reason is that it is burdensome to generate report to use by the policer's upper management. The current system, they need to calculate number of report and draw graph manually which will take a lot of time to do it. From the report, they also need to sum and average all total cost monthly manually and it could be miscounting when do all the calculation.

2.3 Structure Chart

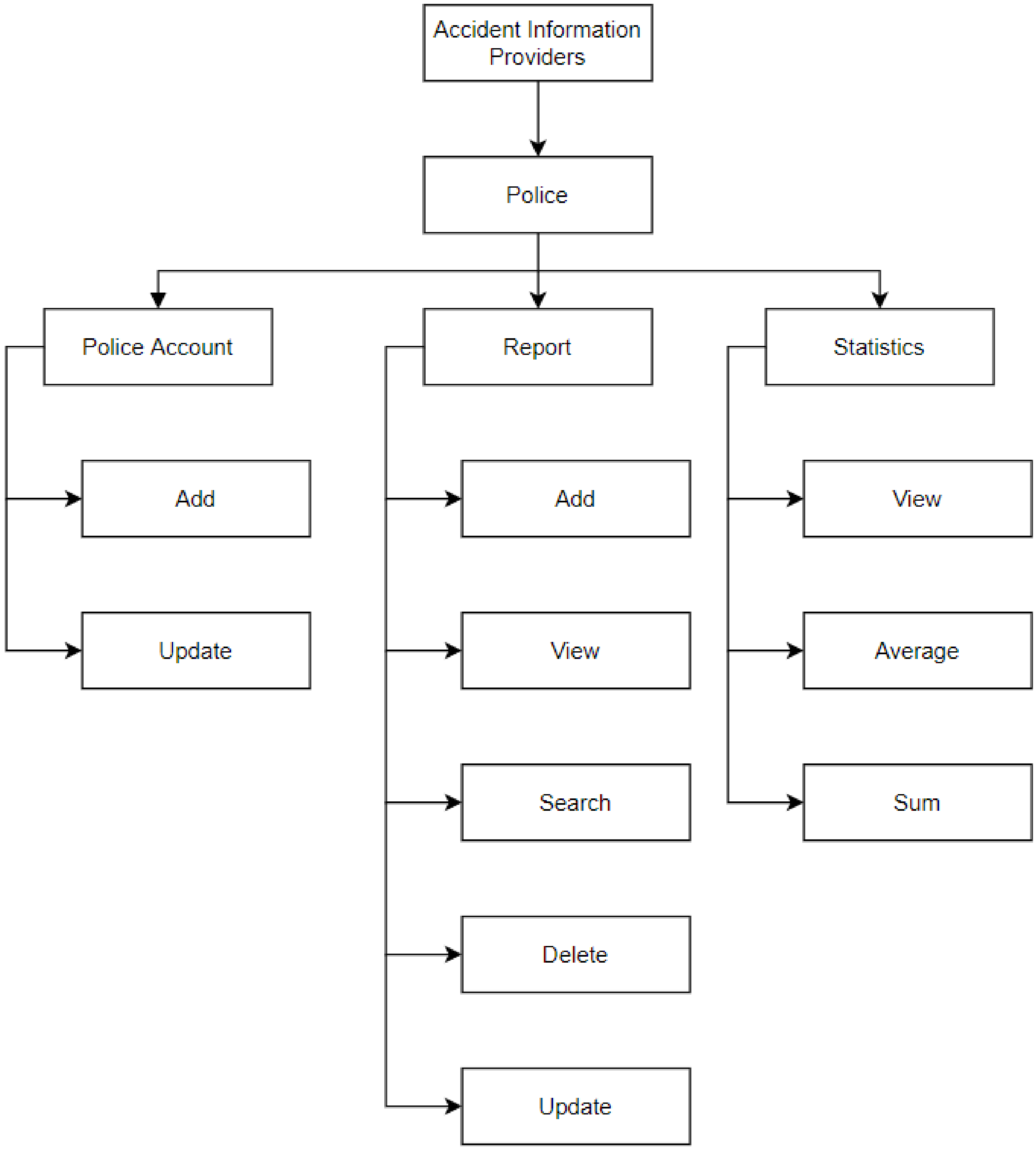


Figure 2.1: Structure Chart

A Structure Chart in software engineering is a chart which shows the breakdown of a system to its lowest manageable levels. This chart, at Figure 2.1 shows the breakdown of function for police.

2.4 Conclusion

This chapter specifies the analysis of the system. This chapter studies thoroughly the problem statement and develops a new concept for the new system. It also investigates all part of the system to ensure all components of the system work flawlessly to accomplish its aim.

The output of this chapter is used to design the modules, interface, and data for the system in the next chapter. Flowchart and database design will be also described thoroughly.

CHAPTER 3: DESIGN

3.1 Introduction

This chapter is about defining the architecture, modules, and data for a system to satisfy specified requirements. System design has two parts, physical and logical design. Physical design relates to the actual input and output processes of the system such as user interface design. Meanwhile, logical design of a system pertains to an abstract representation of the data flows, inputs, and outputs of the system. It includes entity-relationship diagram (ERD) and flowchart.

3.2 Flowchart

3.2.1 Main Flowchart

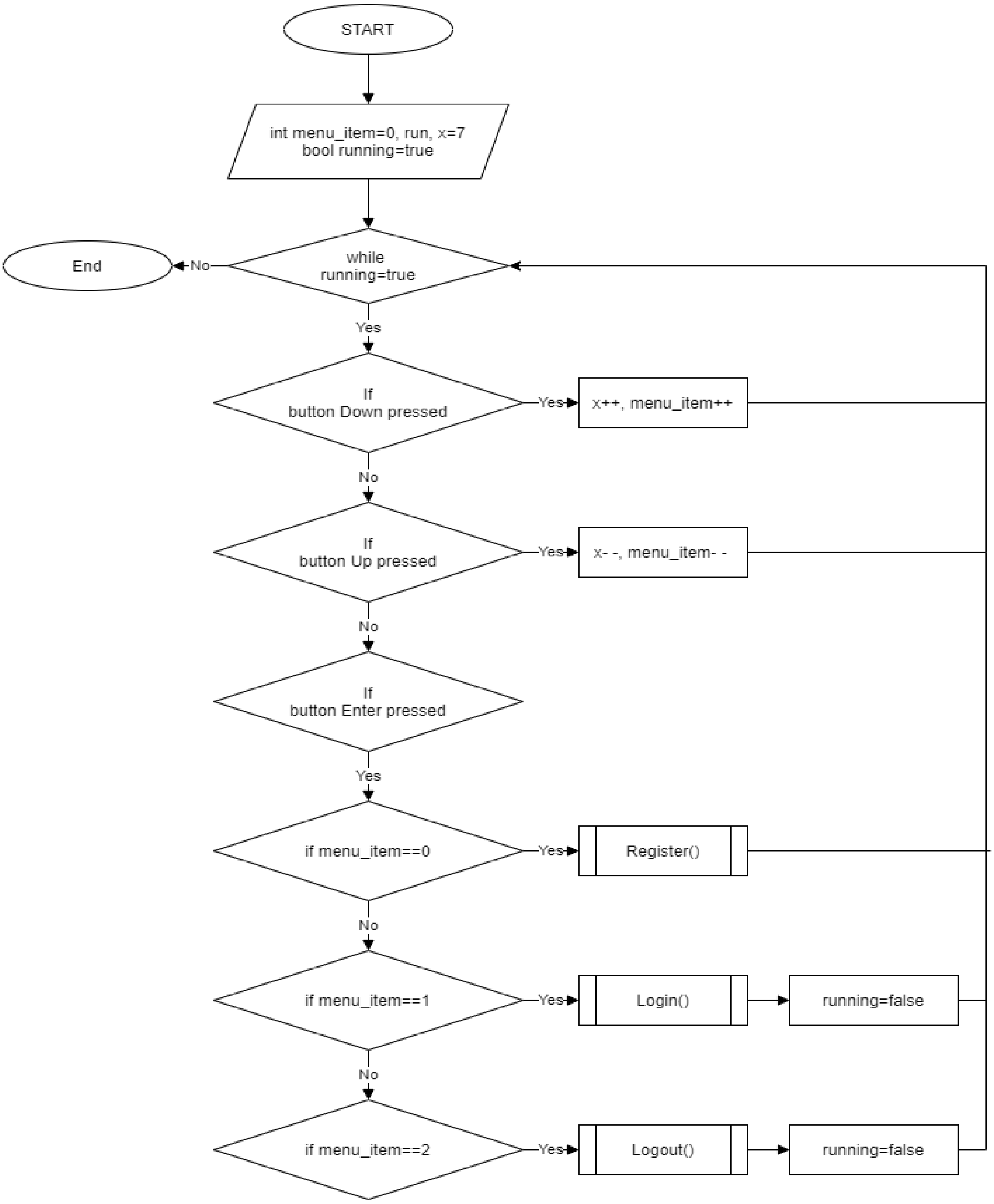


Figure 3.1: Main Flowchart

Figure 3.1 is about main. First the user can choose using Up and Down buttons on keyboard then press enter if choose that option. If user press on first option, the system will go to register function. If user press enters on second option, system will bring user to login function else logout function and end the program.

3.2.2 Register Flowchart

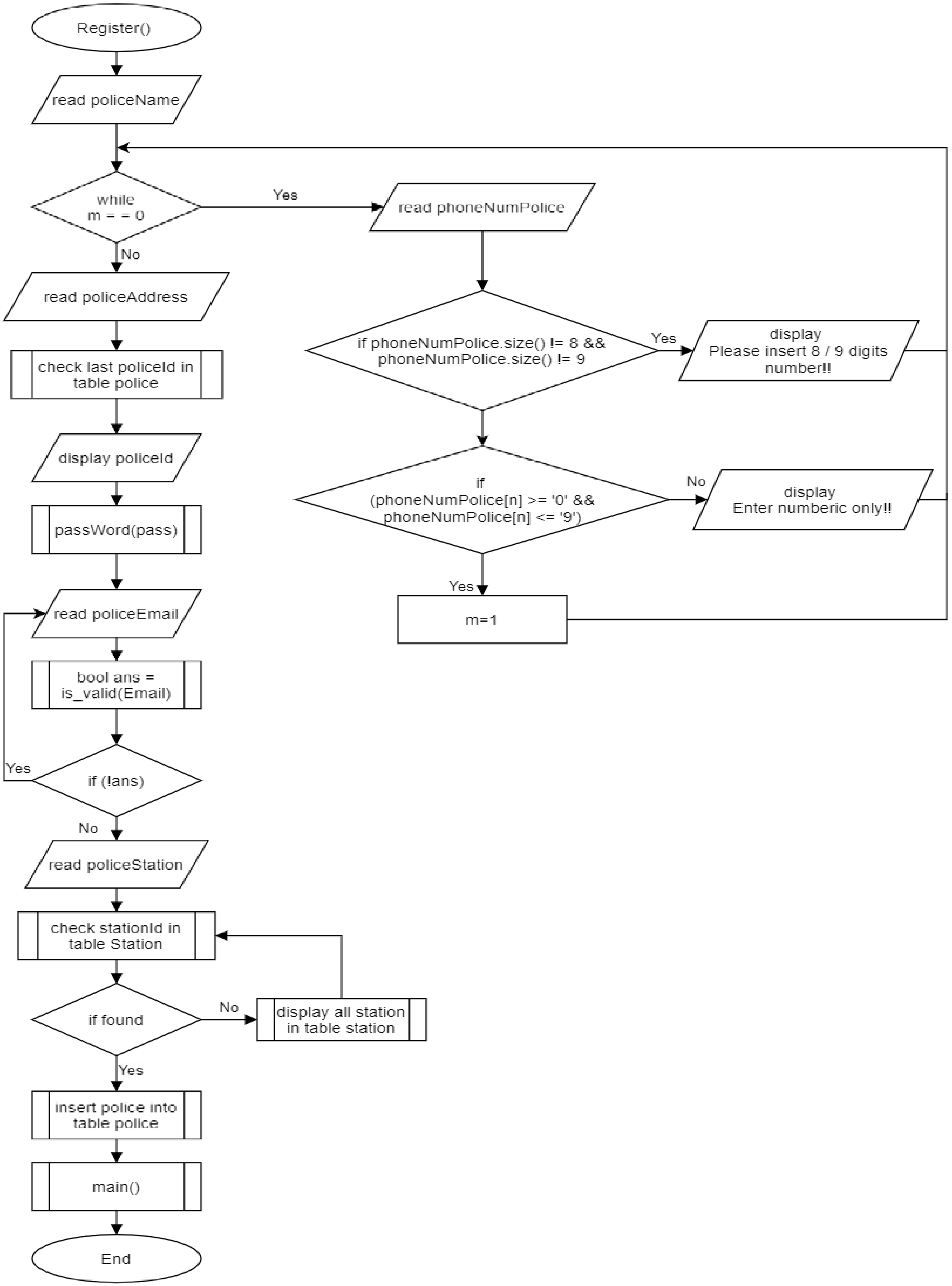


Figure 3.2: Register Flowchart

Figure 3.2 is about registering a user. Firstly, user need to enter their details such as name, address, email, password, and police station. Number phone must length 8 or 9 numbers and consist of all numbers. Id police will show after user put his/her address. So, the id shown will be use when user login. Then user will go to password function to setup the password for login. Then for email, it must have "@" and ".com" to get the valid email. User will be told to setup again the email until the email valid. For station id, user need to enter the right id, if not user will be displayed error message and list of station name and station id and user will ask to enter again the station id until it gets correct as in list.

3.2.3 Password Flowchart

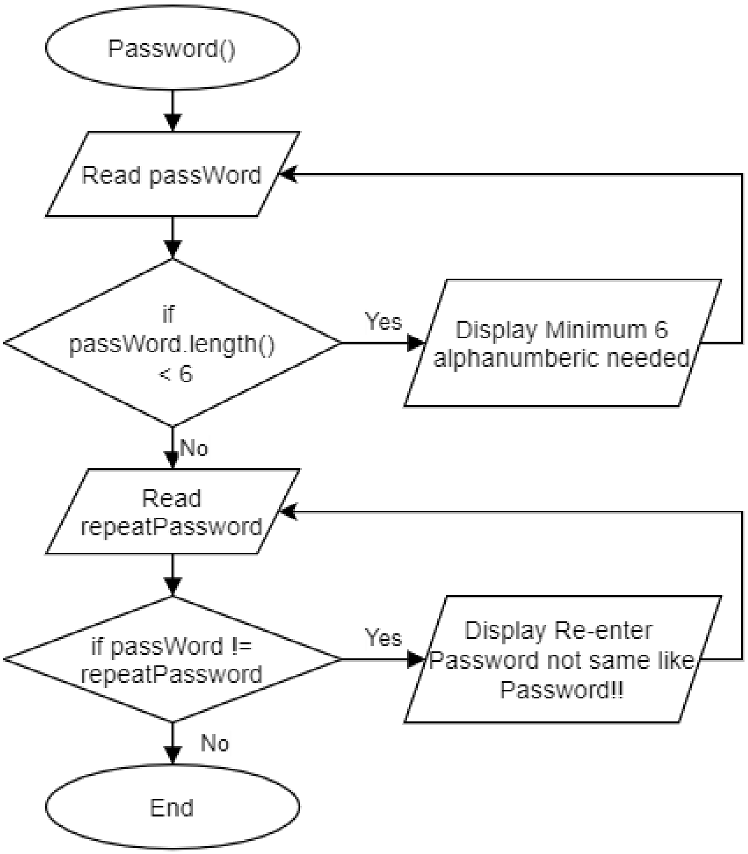

Password()
Read passWord
if passWord.length() < 6
Yes
Display Minimum 6 alphanumberic needed
No
Read repeatPassword
if passWord != repeatPassword
Yes
Display Re-enter Password not same like Password!!
No
End

Figure 3.3: Password Flowchart

Figure 3.3 is about setup a new password. Firstly, the program will ask user to enter the password. If user enter less than 6 alphabets, program will display error message and ask user to enter again the password. If password more than 6 alphabets, program will ask user to re-enter the password and if it not same as password, program will display error message and ask user to re-enter again the password.

3.2.4 Login Flowchart

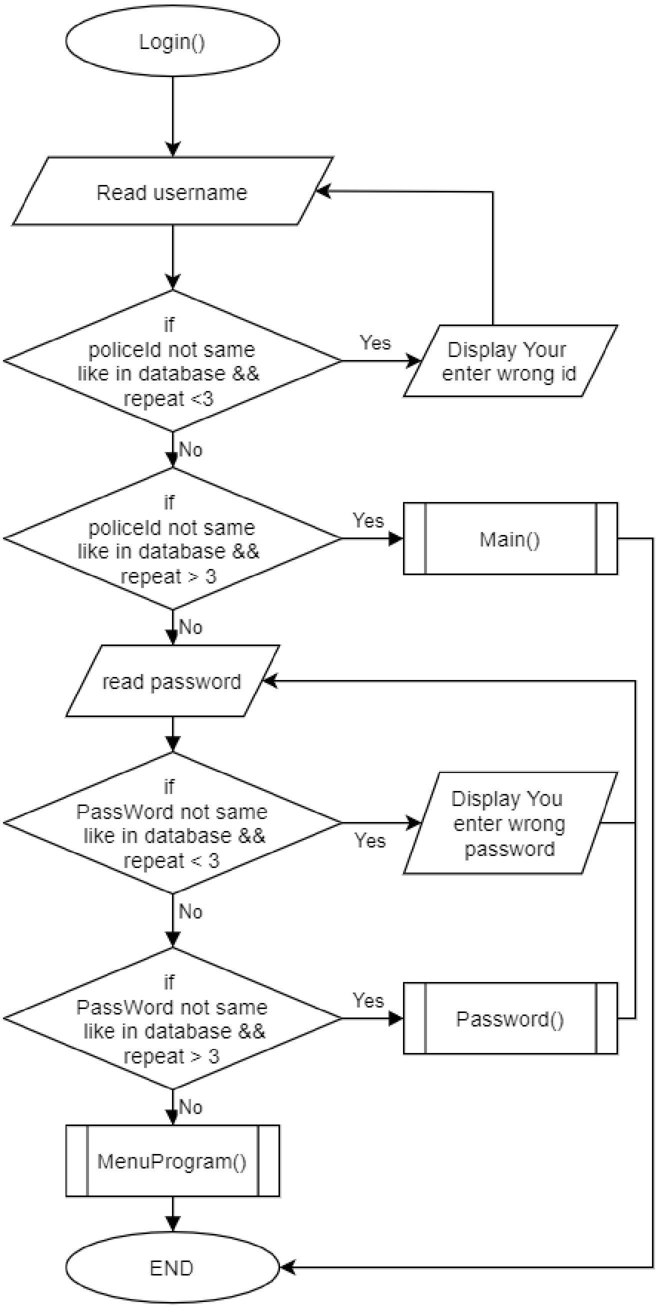


Figure 3.4: Login Flowchart

Figure 3.4 is about login for user. If policeId not same in database, the program will ask user to fill again. If user enter wrong id number, program will go back to main(), else program will ask user to enter password. If user enter wrong password for 3 times, user will ask to set up the new password. If policeId and password are same with database, user will go to MenuProgram().

3.2.5 Menu Program Flowchart

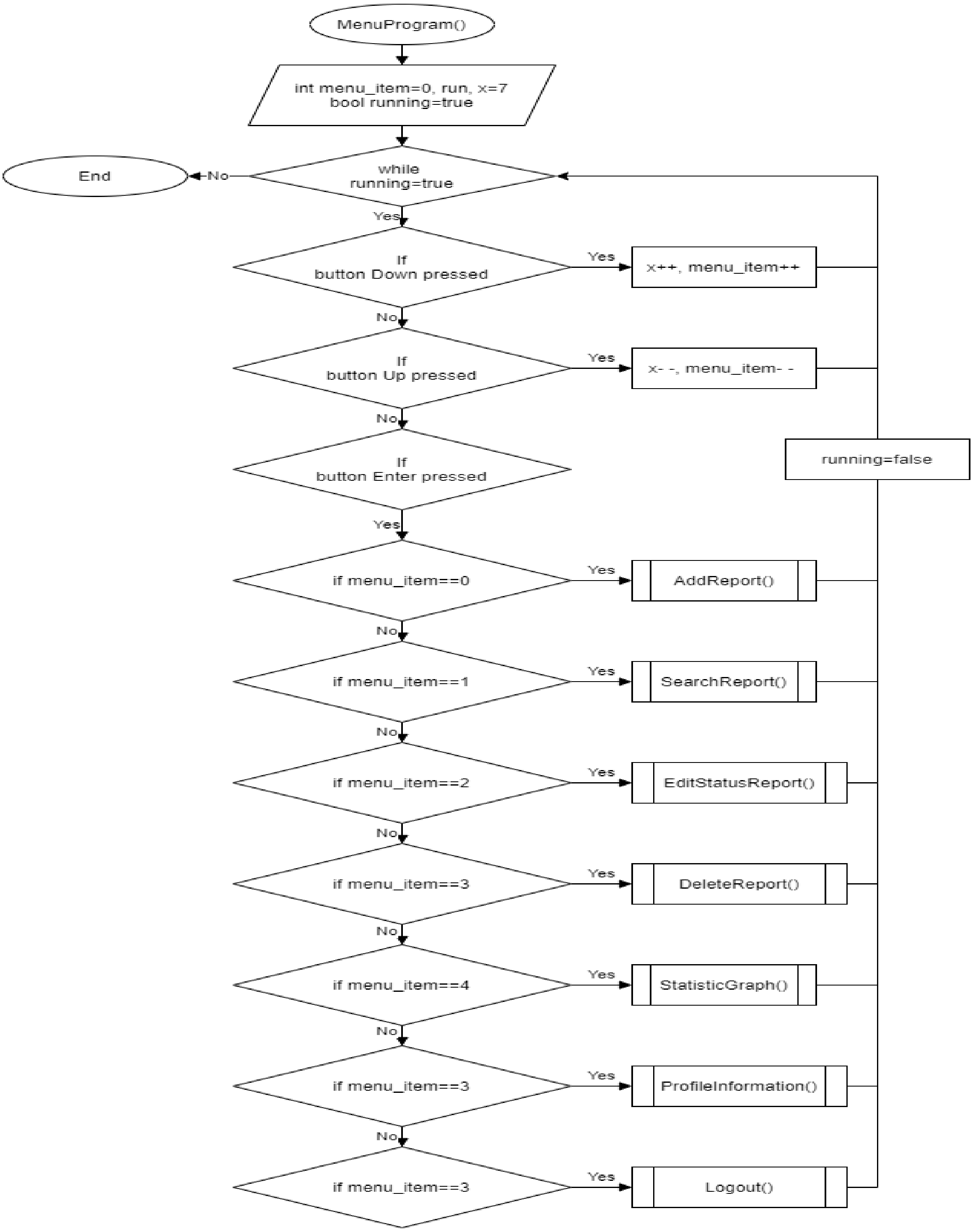


Figure 3.5: Menu Program Flowchart

Figure 3.5 is about menu program. User can choose option by controlling up and down button on keyboards and press enter if user want to choose the option. Program will execute at function that user press enters.

3.2.6 Add Report Flowchart

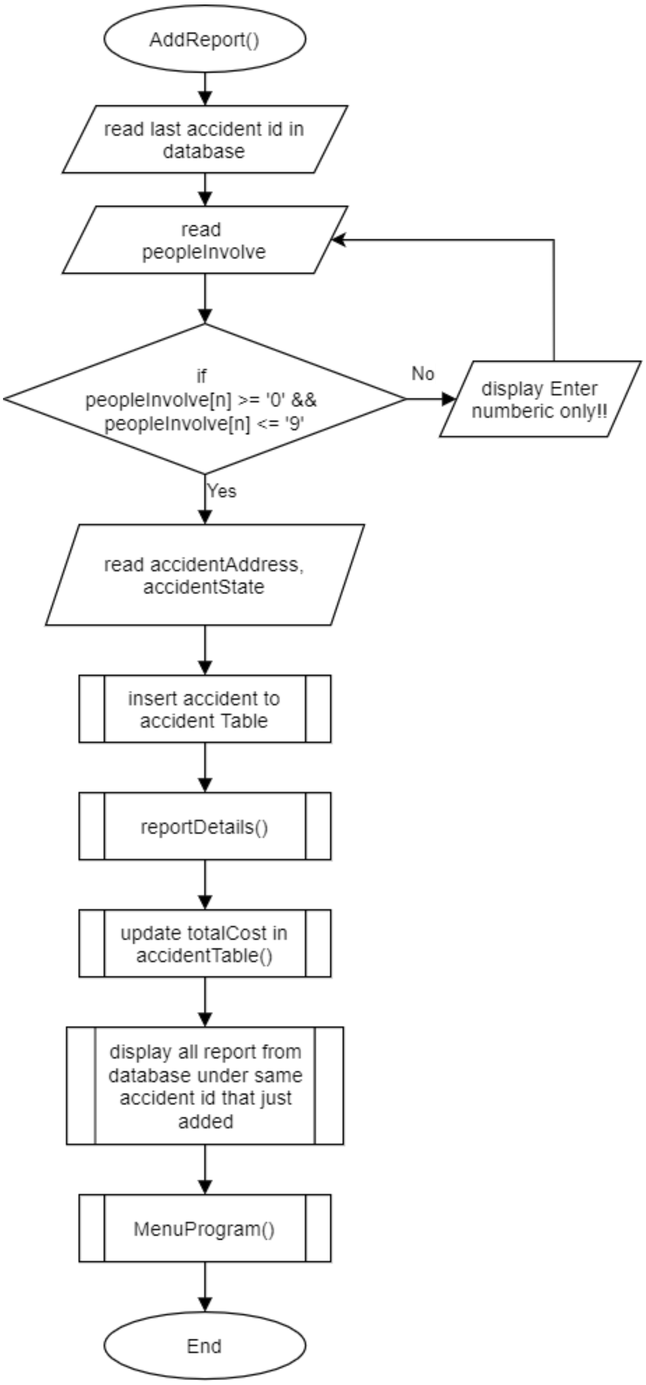

AddReport()
read last accident id in database
read peopleInvolve
if peopleInvolve[n] >= '0' && peopleInvolve[n] <= '9'
No
display Enter numberic only!!
Yes
read accidentAddress, accidentState
insert accident to accident Table
reportDetails()
update totalCost in accidentTable()
display all report from database under same accident id that just added
MenuProgram()
End

Figure 3.6: Add Report Flowchart

Figure 3.6 is about adding new accident details into accident table. Users need to enter number of people involve and if user enters others than numeric, it will send error message, else it will continue to enter accidentAddress and accidentState. Then, program will go to reportDetails and update the totalCost Accident Table. If succeed add update, user will go back to MenuProgram().

3.2.7 Report Details Flowchart

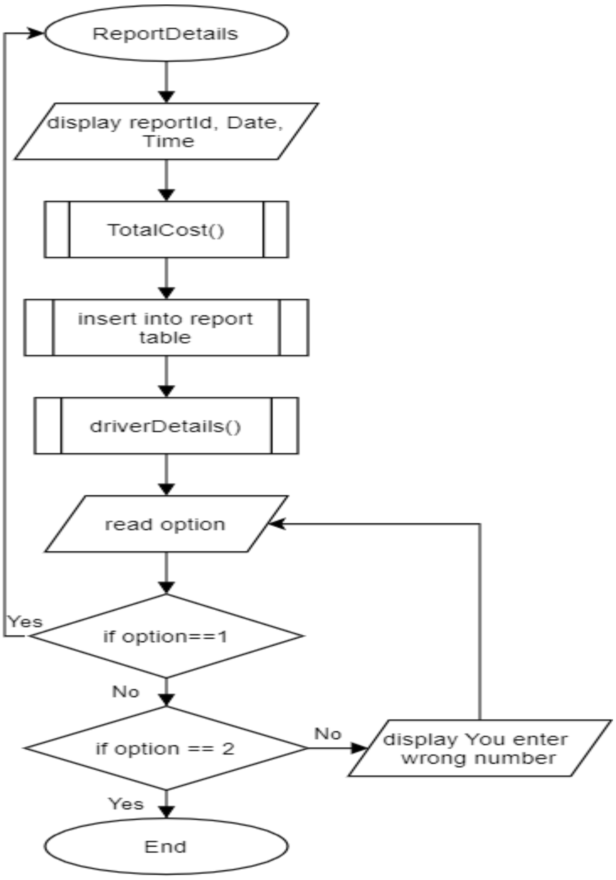


Figure 3.7: Report Details Flowchart

Figure 3.7 is about adding a new report details into table report. It will display report id, time and date add the report. It will go to function total cost to get cost details. It will user to continue adding a new report under the same accident id or done. This function will end if user press 2, if user press 1 it will continue adding a new report otherwise it will print error message and ask again the option.

3.2.8 Driver Details Flowchart

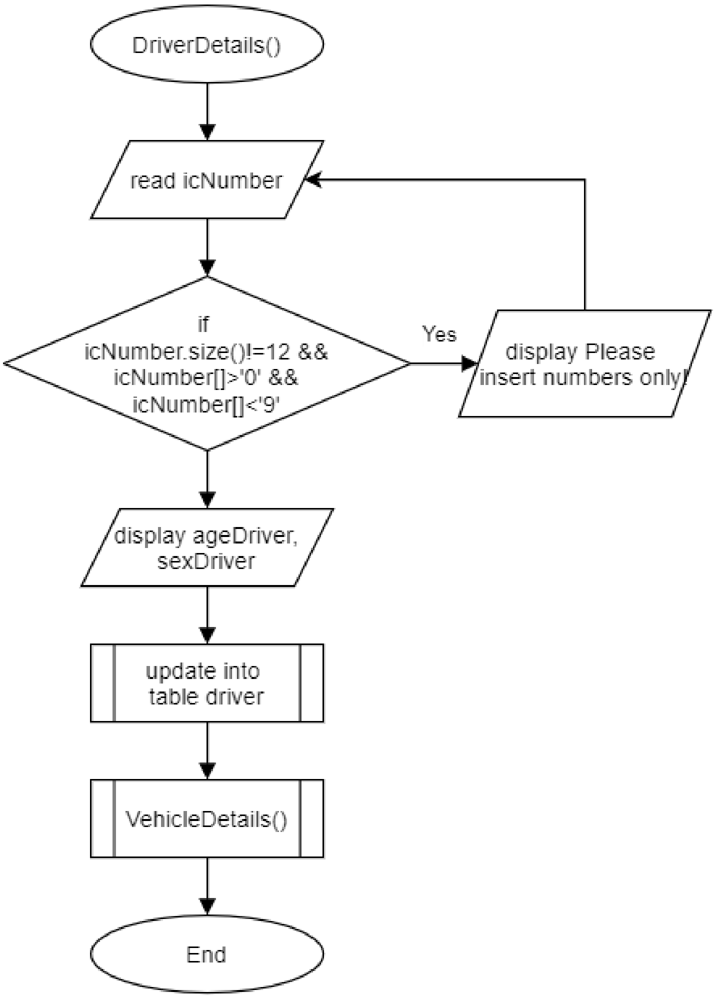
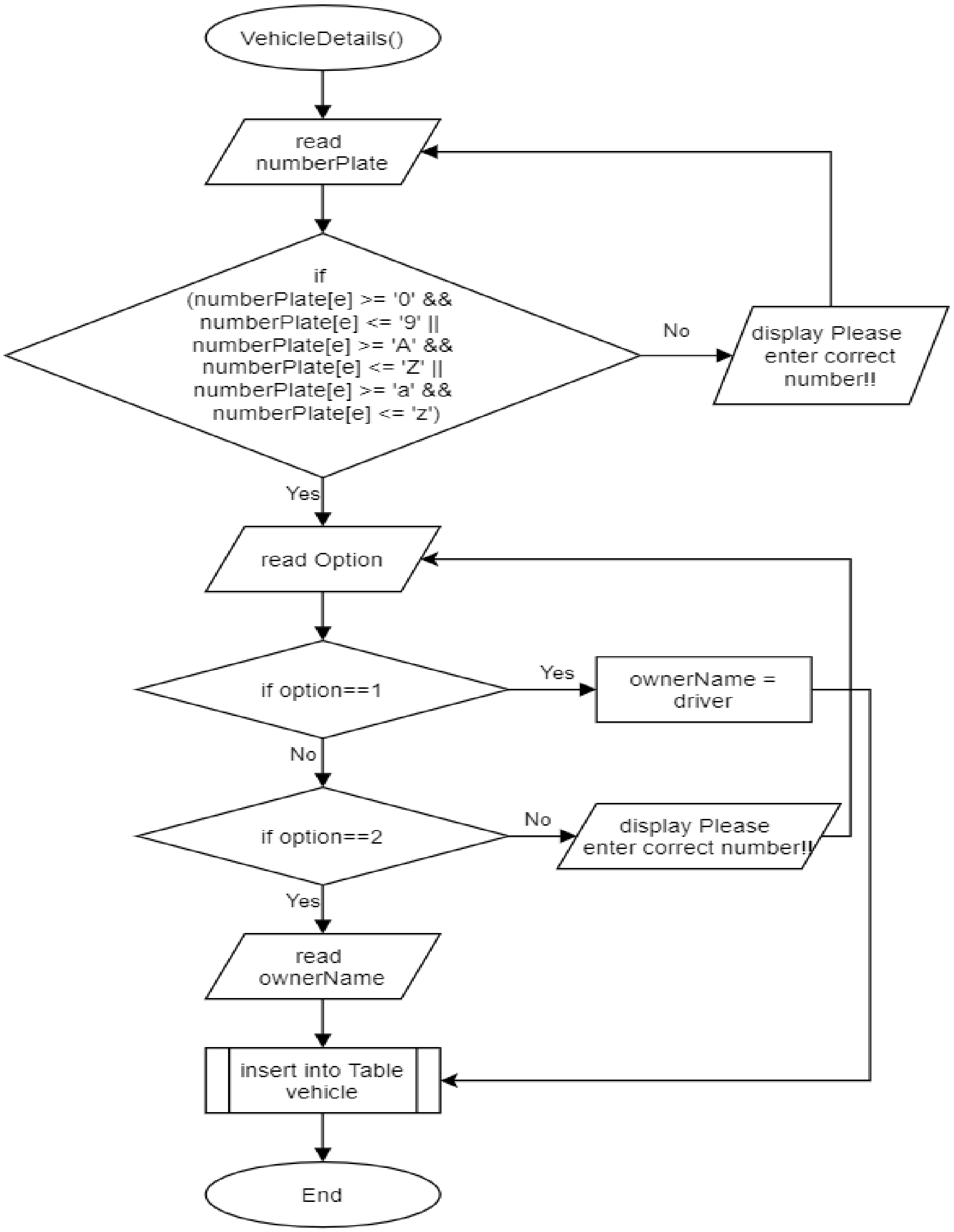


Figure 3.8: Drivers Details Flowchart

Figure 3.8 is about adding new driver details into table driver. Program will detect sex and age driver by ic number. So, ic number must enter with the correct number. It will ask user to enter again if number of ic is not equal to 12 and it contains alphabet. It will also loop if month or day in ic number that we fill is wrong.

3.2.9 Vehicle Details Flowchart



VehicleDetails()
read numberPlate
if (numberPlate[e] >= '0' && numberPlate[e] <= '9' || numberPlate[e] >= 'A' && numberPlate[e] <= 'Z' || numberPlate[e] >= 'a' && numberPlate[e] <= 'z')
No
display Please enter correct number!!
Yes
read Option
if option==1
Yes
ownerName = driver
No
if option==2
No
display Please enter correct number!!
Yes
read ownerName
insert into Table vehicle
End

Figure 3.9: Vehicle Details Flowchart

Figure 3.9 is about adding vehicle details into table vehicle. It will ask user either the name of owner car same as driver or not. If same user will press 1, if not user will press 2 and enter the new name otherwise program will display error message and ask user to enter again the option.

3.2.10 Total Cost Flowchart

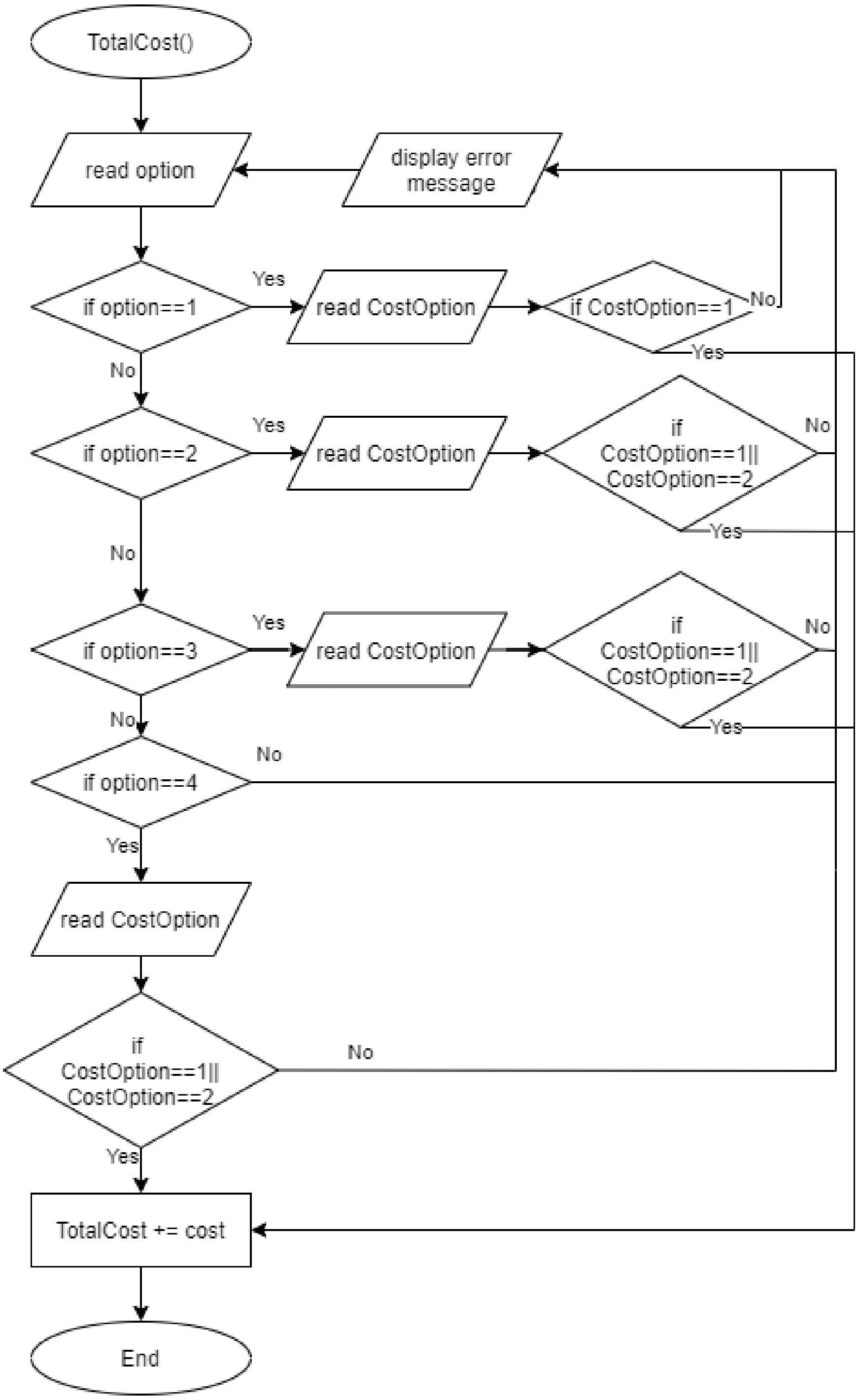


Figure 3.10: Total Cost Flowchart

Figure 3.10 is about payment for each report. There will be 4 category and user can choose only 1 for each report. Total cost will update in table accident once user done adding all reports under the same accident id.

3.2.11 Search Report Flowchart

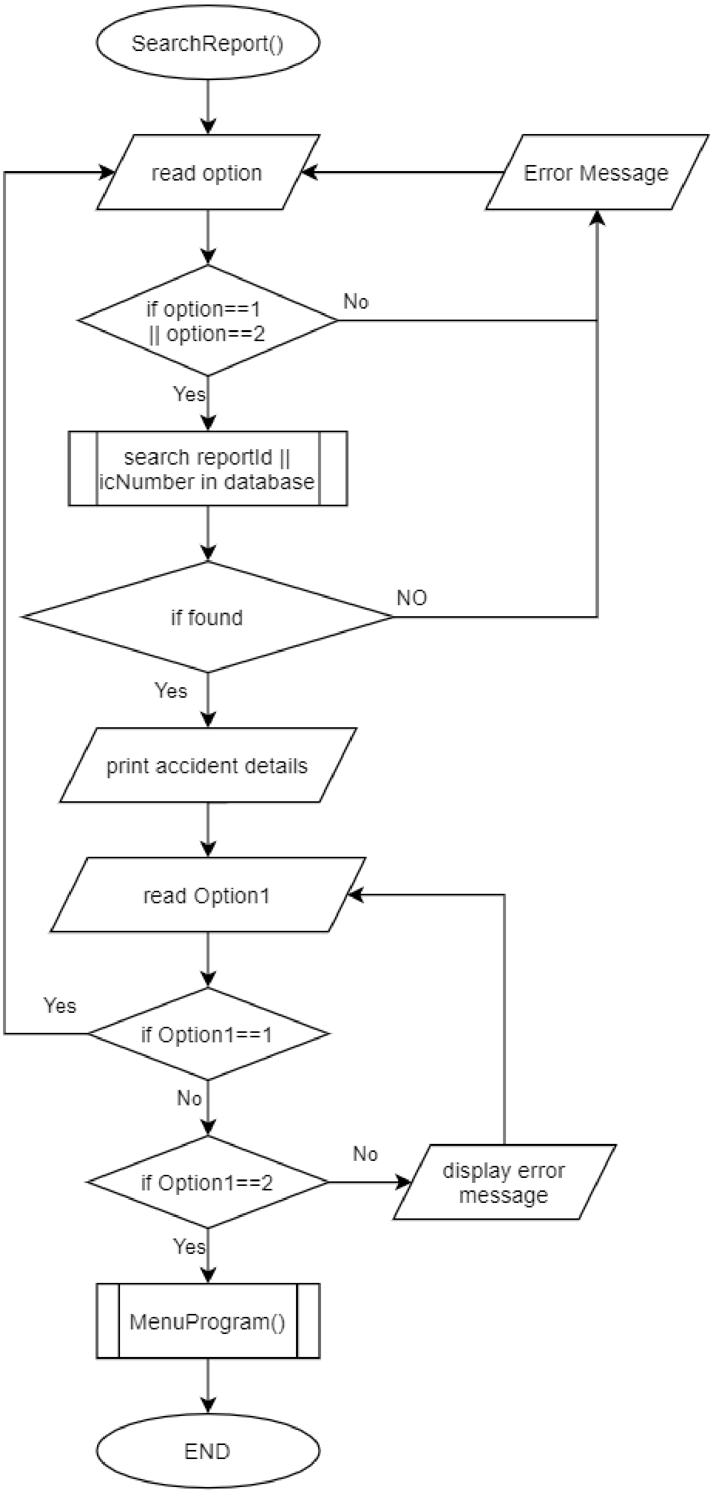


Figure 3.11: Search Report Flowchart

Figure 3.11 is about view report by searching using report id or ic number. If reportId or ic number not found it will print error message and ask to search again. If found, program will display searched id. Then it will ask user if want to continue search again or not. If yes it will continue again and if not it will go to menuProgram().

3.2.12 Edit Status Flowchart

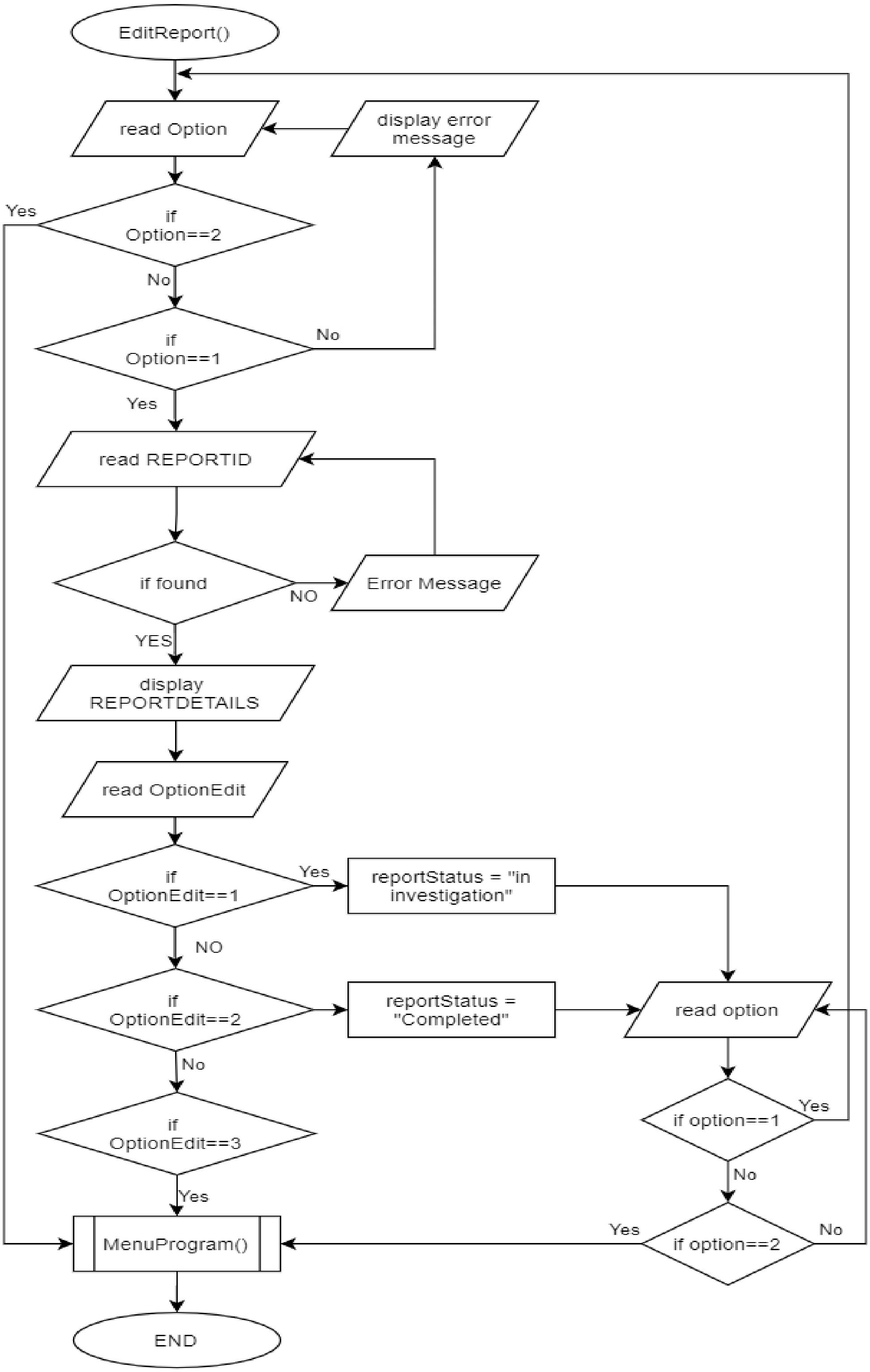


Figure 3.12: Edit Status Flowchart

Figure 3.12 is about edit status report. There will be 2 option for update status which in investigation or completed. Then, they can continue to menu program or continue update status report.

3.2.13 Delete Report Flowchart

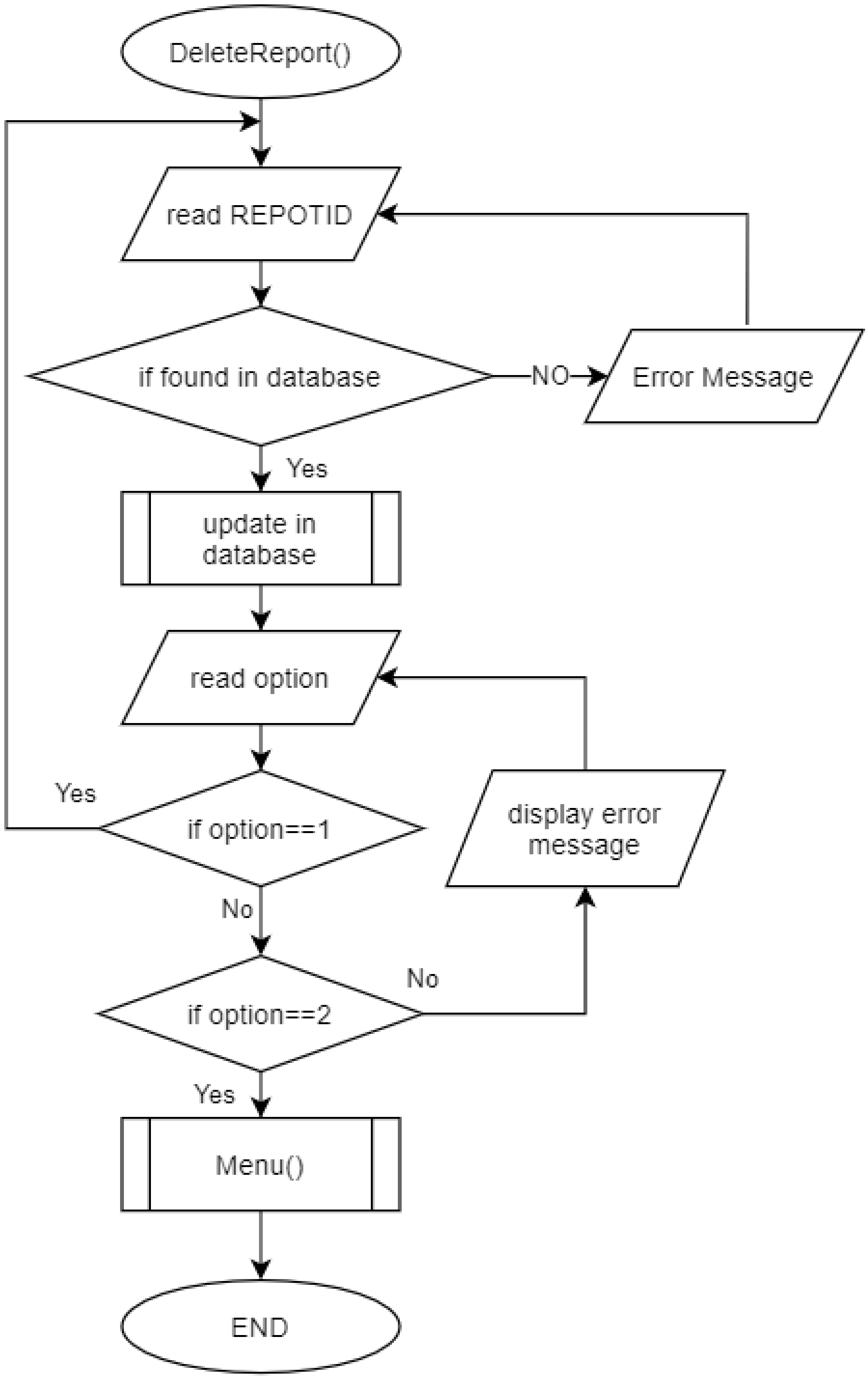


Figure 3.13: Delete Report Flowchart

Figure 3.13 is about deleting report. If user confirm want to delete report id, so that vehicle and driver table which have the same report id it will delete also.

3.2.14 Update Profile Flowchart

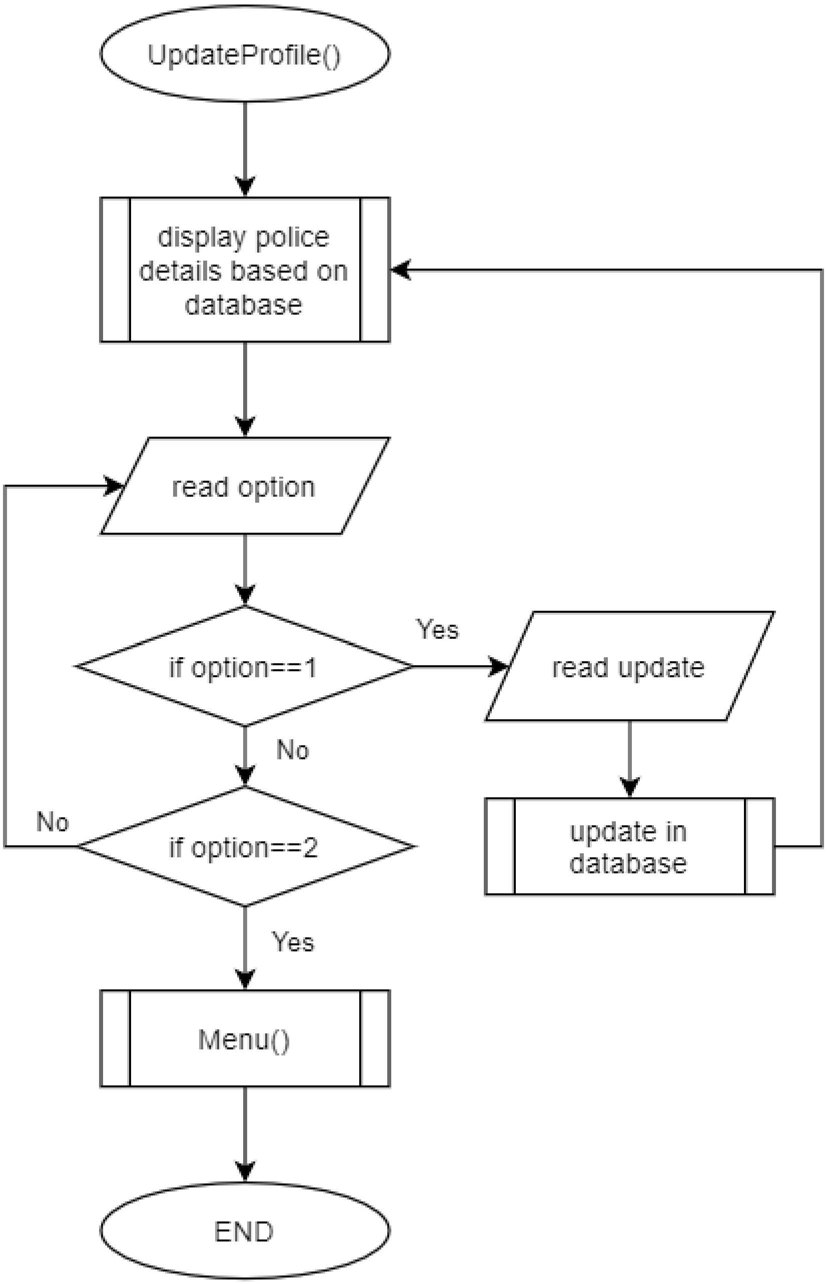


Figure 3.14: Total Cost Flowchart

Figure 3.14 is about update profile. User can change by option they want. They can update user’s name, station id, password, email, and number phone.

3.2.15 Statistics Graph Flowchart

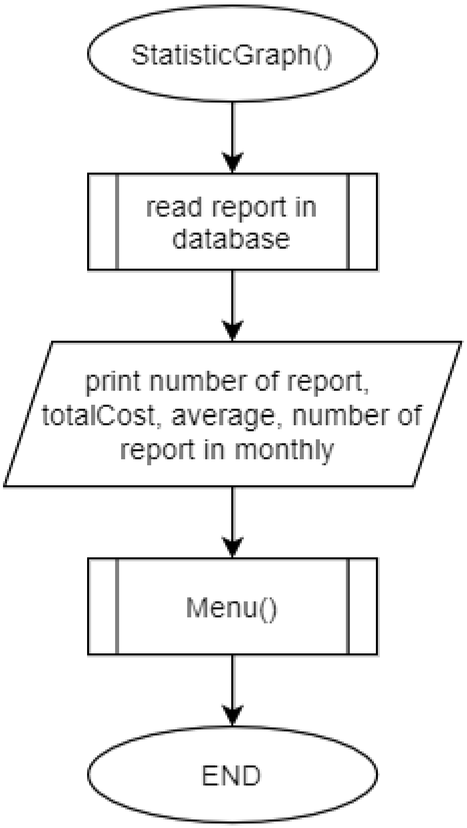

StatisticGraph()
read report in database
print number of report, totalCost, average, number of report in monthly
Menu()
END

Figure 3.15: Statistics Graph Flowchart

Figure 3.15 is about statistic graph flowchart. Program will display a table and a graph for report monthly. Then, it will display total cost, average and highest category was reported in that month.

3.3 Entity Relationship Diagram

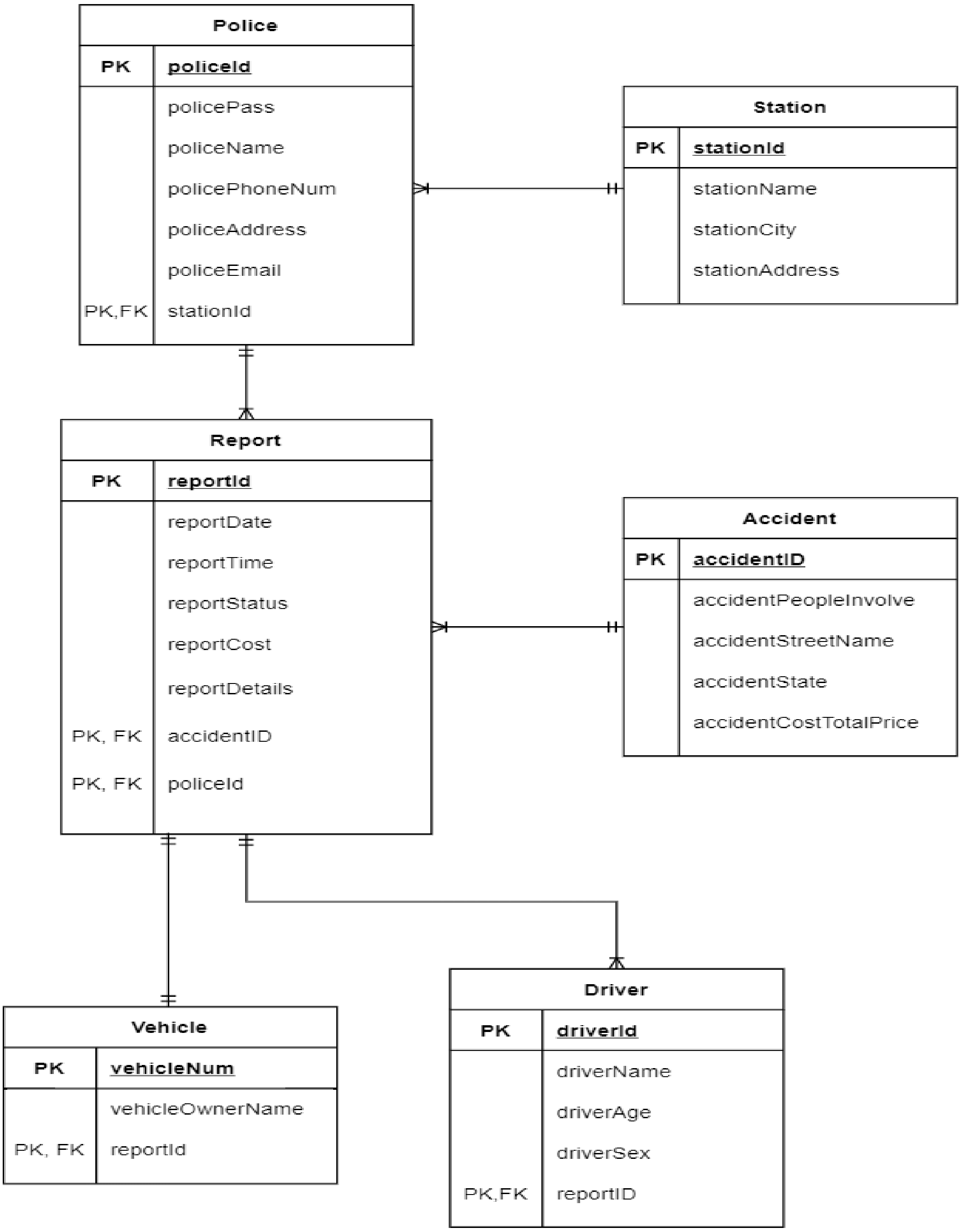


Figure 3.16: Accident Report Providers System ERD

3.4 Data Dictionary

3.4.1 Table Station

Attribute	Data type	Field length	Constraint	Null	Description
stationId	varchar	10	PK	-	-
stationName	varchar	50	-	-	-
stationCity	varchar	20	-	-	-
stationAddress	varchar	1000	-	-	-

Table 3.1: Data dictionary for table station

3.4.2 Table Police

Field name	Data type	Field length	Constraint	Null	Description
policeId	Varchar	10	PK	-	-
policePass	Varchar	100	-	-	-
policeName	Varchar	100	-	-	-
policePhoneNum	Varchar	10	unique	-	-
policeAddress	Varchar	250	-	-	-
policeEmail	Varchar	100	-	-	-
stationId	Varchar	10	PK, FK	-	-

Table 3.2: Data dictionary for table police

3.4.3 Table Report

Field name	Data type	Field length	Constraint	Null	Description
reportId	Varchar	10	PK	-	-
reportDate	Date	-	-	-	-
reportTime	Time	-	-	-	-
reportSatus	Varchar	20	-	-	-
reportCost	Int	11	-	-	-
reportDetails	Varchar	1000	-	-	-
accidentId	Varchar	10	PK, FK	-	-
policeId	Varchar	10	PK, FK	-	-

Table 3.3: Data dictionary for table report

3.4.4 Table Accident

Field name	Data type	Field length	Constraint	Null	Description
accidentId	Varchar	10	PK	-	-
accidentPeopleInvolve	Varchar	10	-	-	-
accidentStreetName	Varchar	100	-	-	-
accidentState	Varchar	20	-	-	-
accidentCostTotalPrice	Int	10	-	-	-

Table 3.4: Data dictionary for table accident

3.4.5 Table Driver

Field name	Data type	Field length	Constraint	Null	Description
driverId	Varchar	15	PK	-	-
driverName	Varchar	1000	-	-	-
driverAge	Int	3	-	-	-
driverSex	Varchar	10	-	-	-
reportId	Varchar	10	PK, FK	-	-

Table 3.5: Data dictionary for table driver

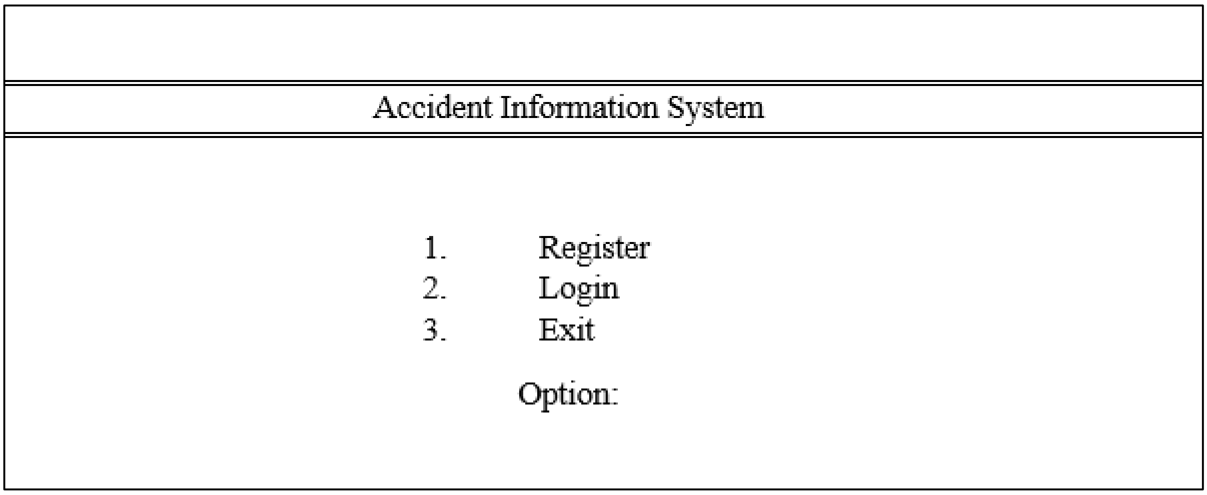
3.4.6 Table Vehicle

Field name	Data type	Field length	Constraint	Null	Description
vehicleNum	Varchar	10	PK	-	-
vehicleOwnerName	Varchar	100	-	-	-
reportId	varchar	10	PK,FK	-	-

Table 3.6: Data dictionary for table vehicle

3.5 Interface Design

3.5.1 Main Menu page



Accident Information System

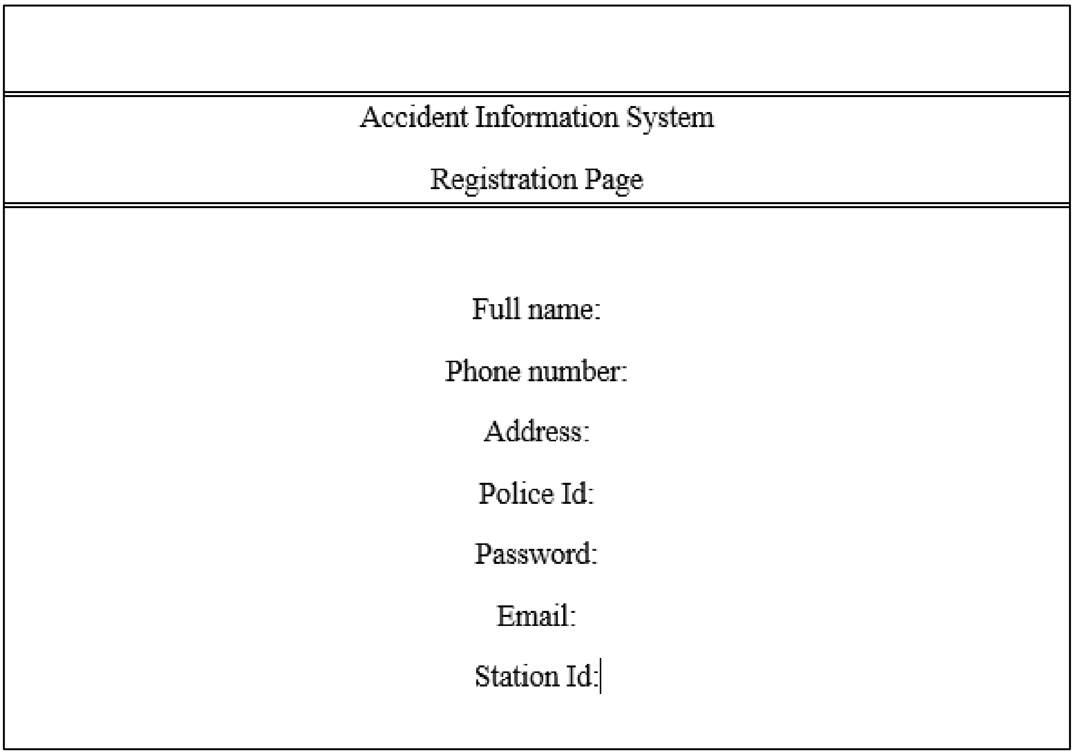
1. Register
2. Login
3. Exit

Option:

Figure 3.17: Output login and register option

[Figure 3.17 is the homepage for users to either login or register or exit]

3.5.2 Register page



Accident Information System

Registration Page

Full name:

Phone number:

Address:

Police Id:

Password:

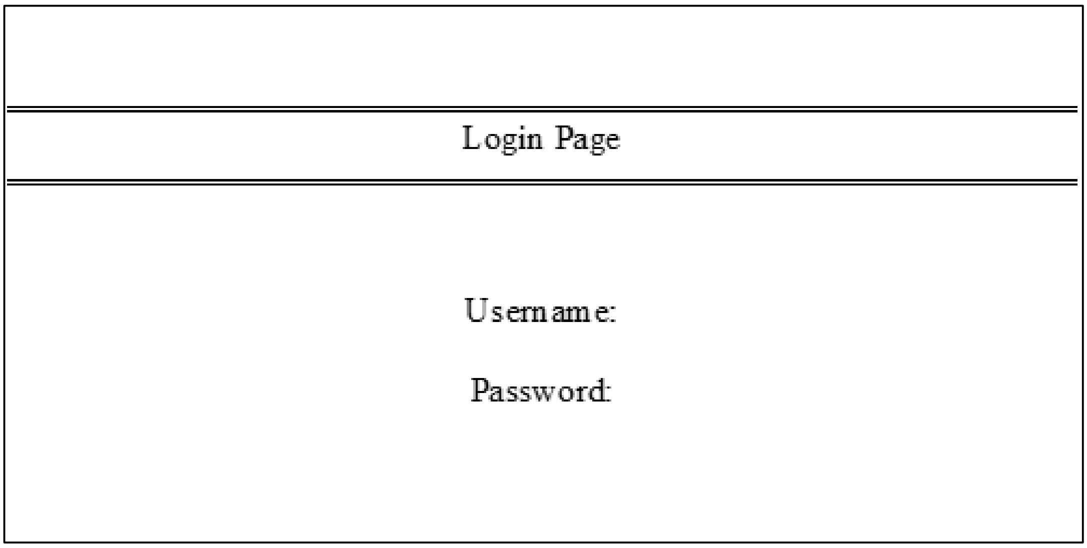
Email:

Station Id:

Figure 3.18: Output register page

[Figure 3.18 is to add a student that is not yet registered inside the system database]

3.5.3 Login page

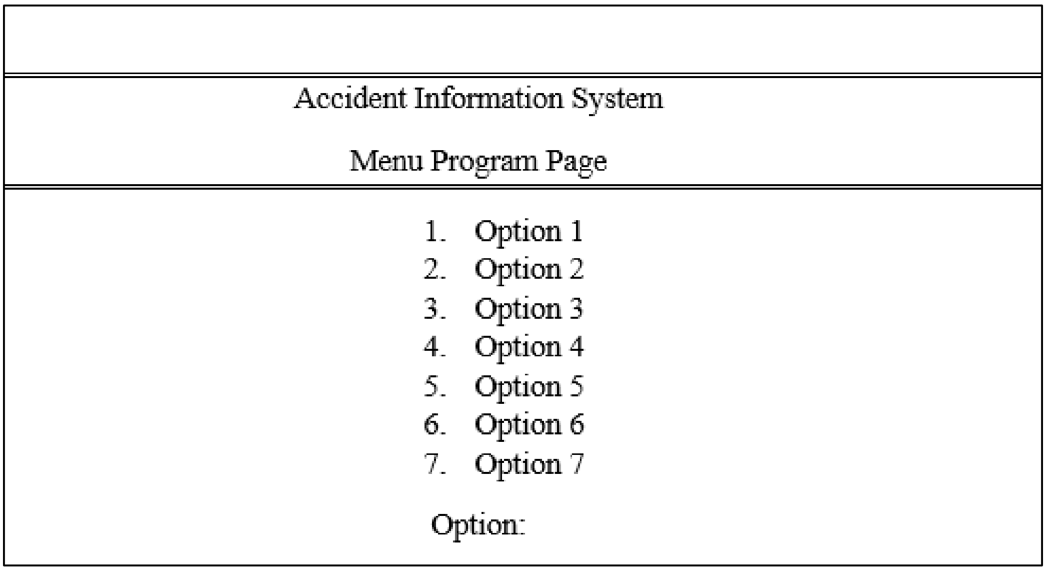


```
Login Page

Username:
Password:
```

Figure 3.19: Output login page

[Figure 3.19 is to make sure that only authorized user can log into the system]

3.5.4 Menu Program page

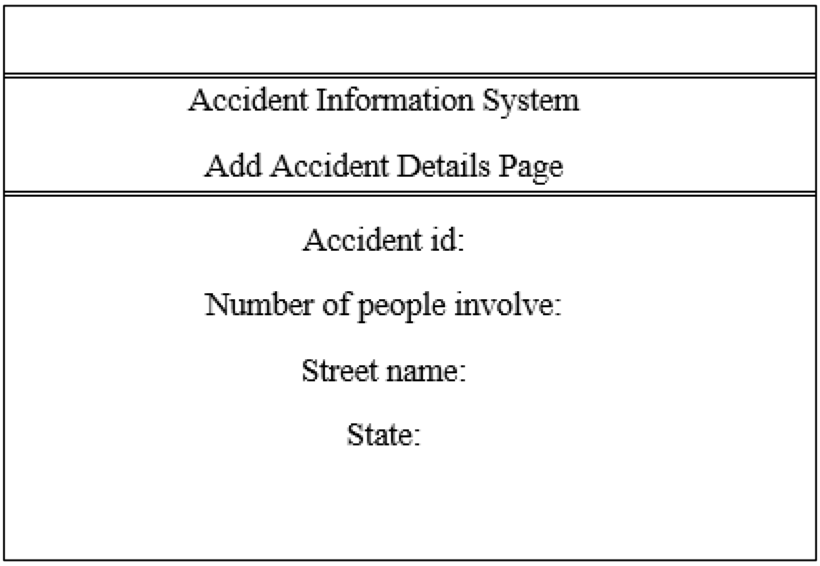


```
Accident Information System
Menu Program Page

1. Option 1
2. Option 2
3. Option 3
4. Option 4
5. Option 5
6. Option 6
7. Option 7

Option:
```

Figure 3.20: Output menu program

[Figure 3.20 is to show the option that a police officer can do]

3.5.5 Add accident Details page



Accident Information System

Add Accident Details Page

Accident id:

Number of people involve:

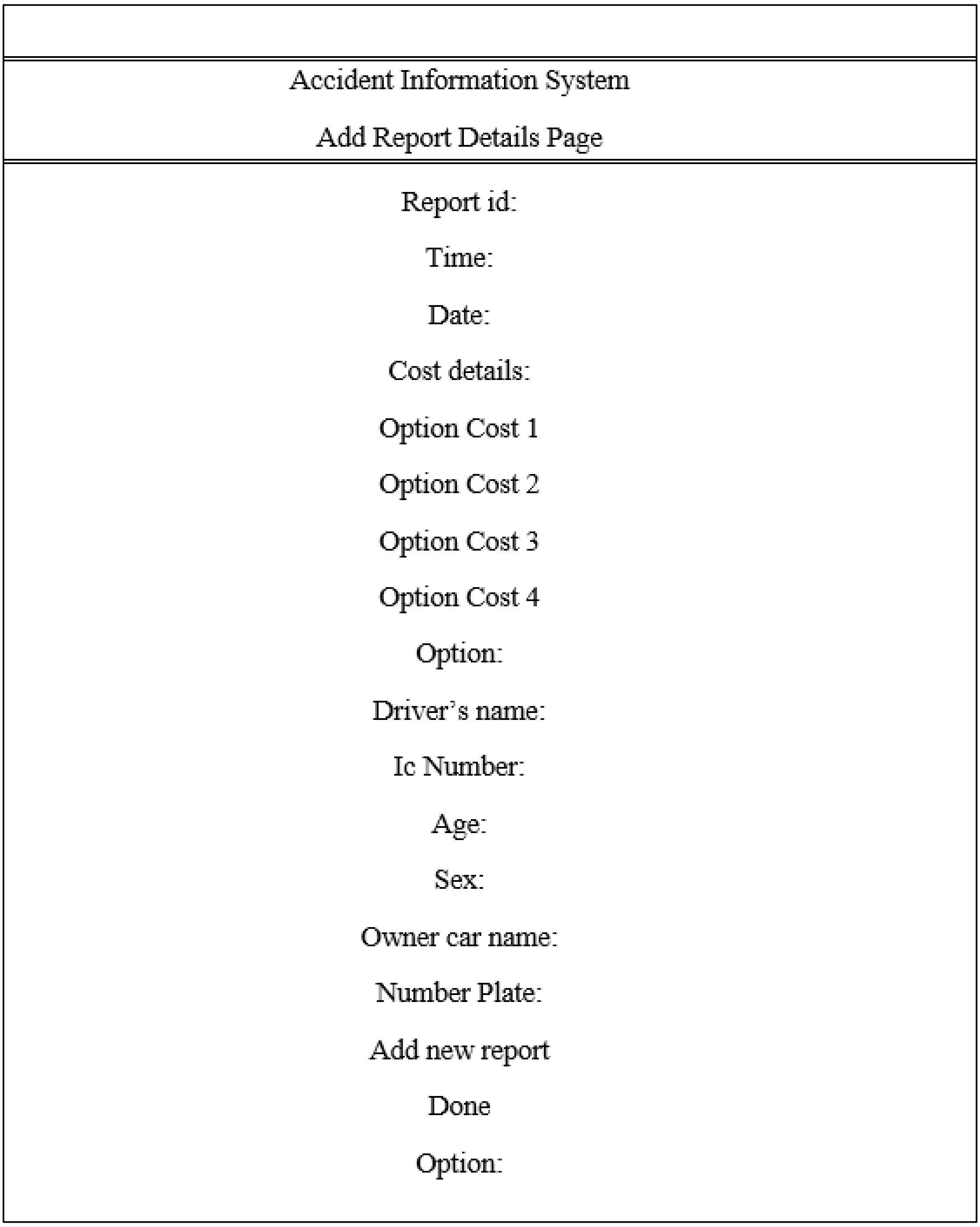
Street name:

State:

Figure 3.21: Output add accident details page

[Figure 3.21 is to allow the user to update user’s information]

3.5.6 Add report, vehicle, and driver details page



Accident Information System

Add Report Details Page

Report id:

Time:

Date:

Cost details:

Option Cost 1

Option Cost 2

Option Cost 3

Option Cost 4

Option:

Driver’s name:

Ic Number:

Age:

Sex:

Owner car name:

Number Plate:

Add new report

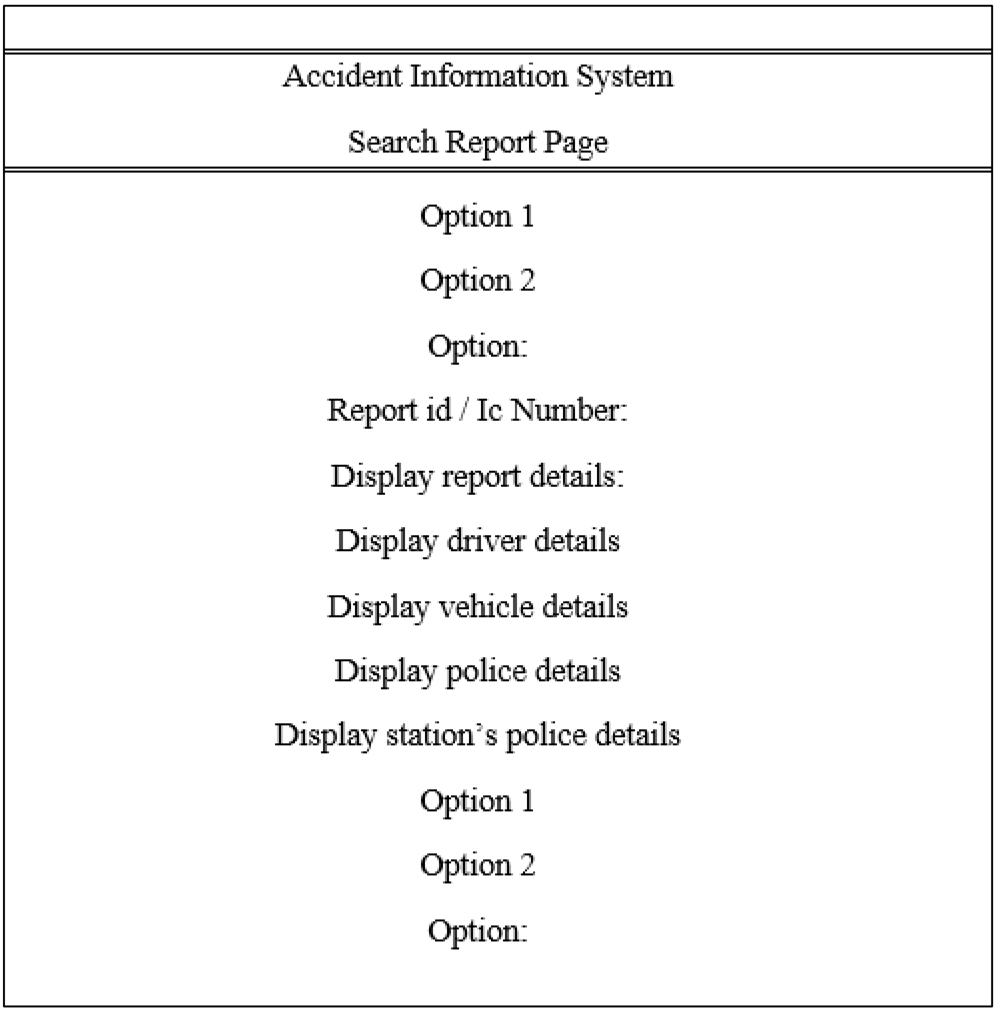
Done

Option:

Figure 3.22: Output add report, driver, vehicle page

[Figure 3.22 is to police to add new report]

3.5.7 Search Report page



Accident Information System

Search Report Page

Option 1

Option 2

Option:

Report id / Ic Number:

Display report details:

Display driver details

Display vehicle details

Display police details

Display station's police details

Option 1

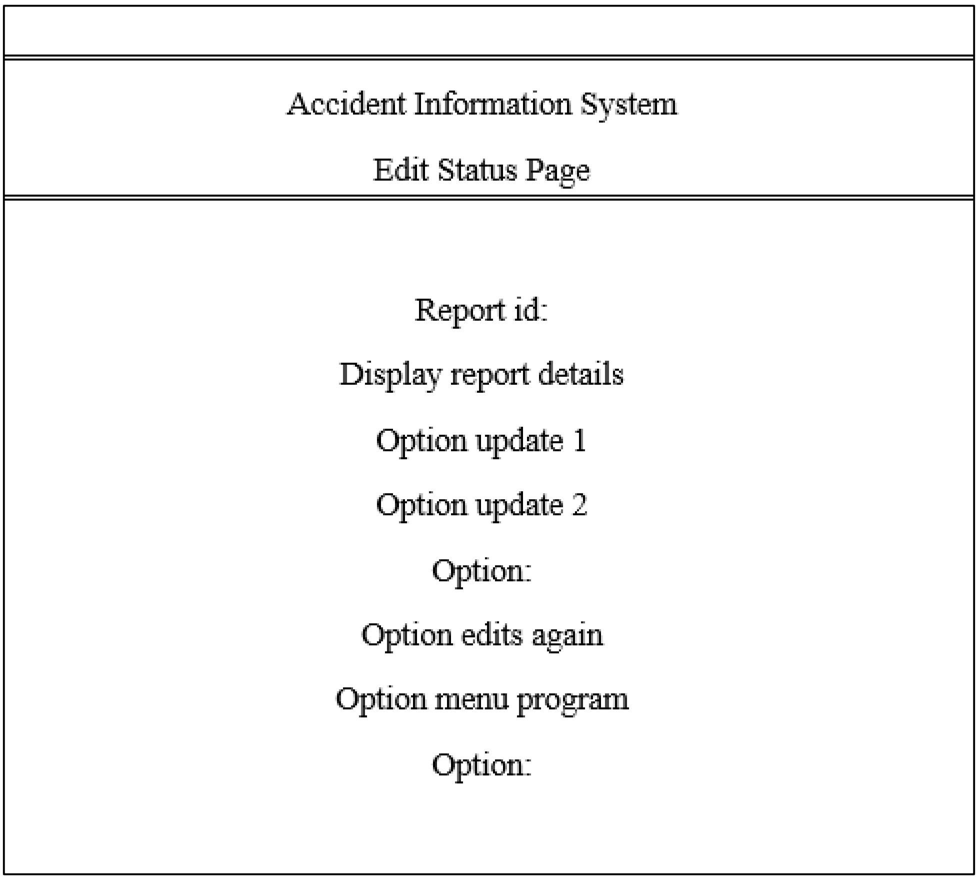
Option 2

Option:

Figure 3.23: Output search report page

[Figure 3.23 is to search id/ or ic number by the user and display the report details]

3.5.8 Edit Status report page



Accident Information System

Edit Status Page

Report id:

Display report details

Option update 1

Option update 2

Option:

Option edits again

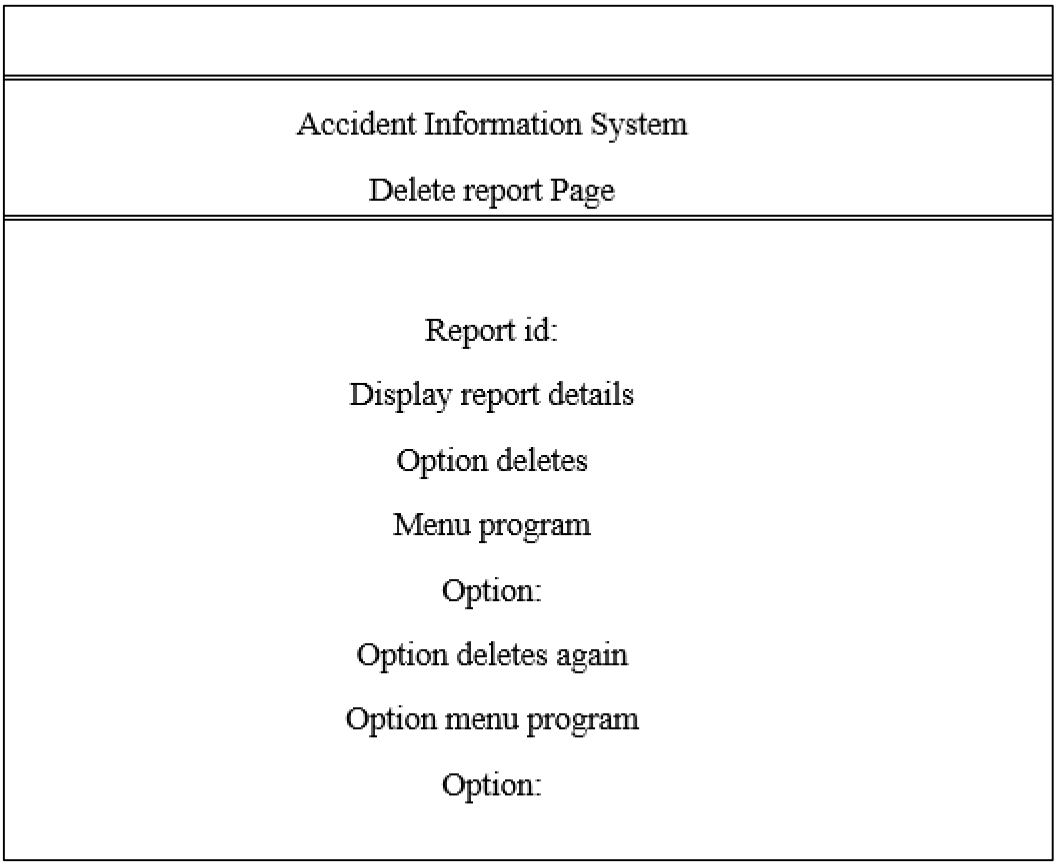
Option menu program

Option:

Figure 3.24: Output edit status page

[Figure 3.24 will update status for searched report]

3.5.9 Delete report id page



Accident Information System

Delete report Page

Report id:

Display report details

Option deletes

Menu program

Option:

Option deletes again

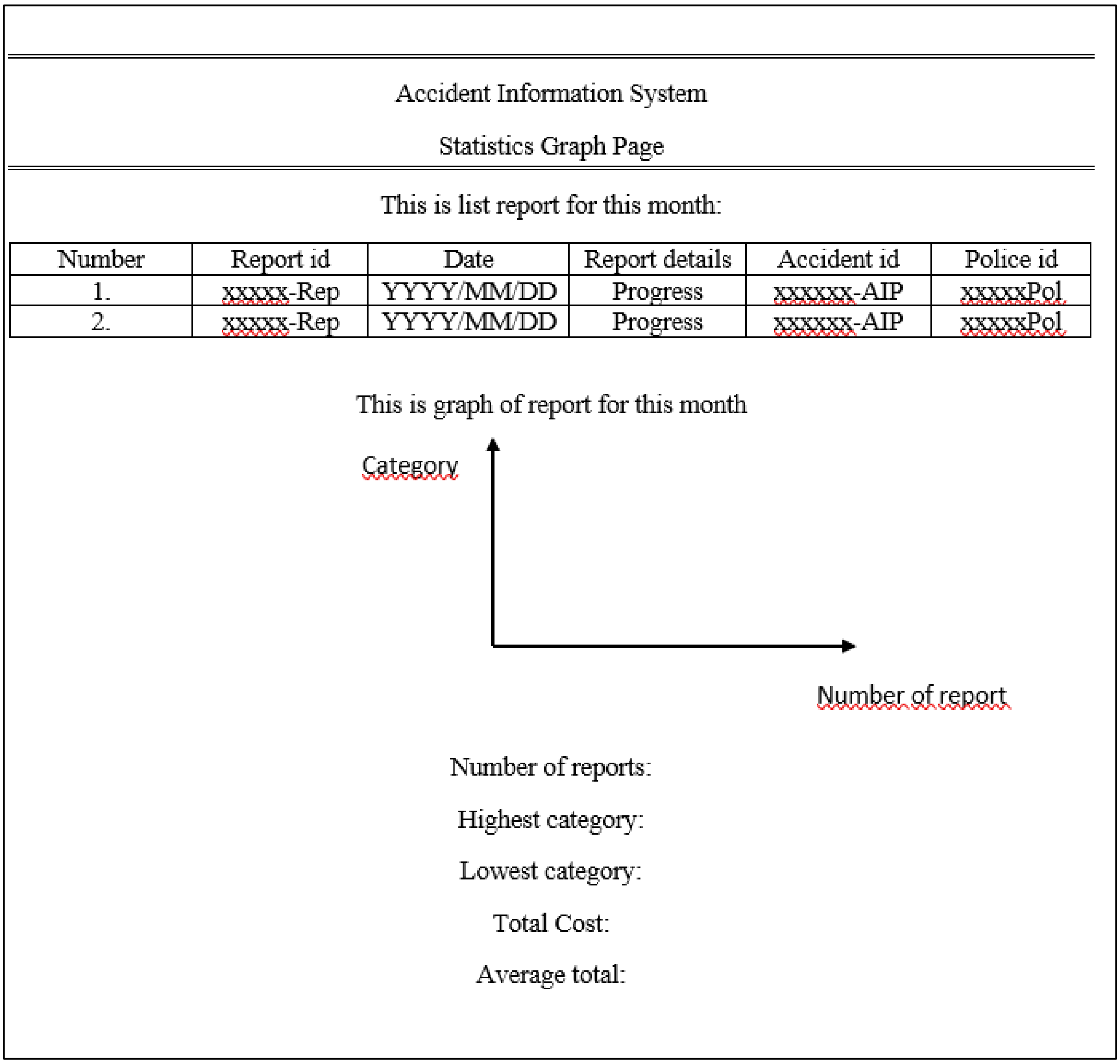
Option menu program

Option:

Figure 3.25: Output delete report page

[Figure 3.25 is deleting report, drivers and vehicle searched id report]

3.5.10 Statistic Graph page



Accident Information System

Statistics Graph Page

This is list report for this month:

Number	Report id	Date	Report details	Accident id	Police id
1.	xxxxx-Rep	YYYY/MM/DD	Progress	xxxxxx-AIP	xxxxxPol
2.	xxxxx-Rep	YYYY/MM/DD	Progress	xxxxxx-AIP	xxxxxPol

This is graph of report for this month

Number of reports:

Highest category:

Lowest category:

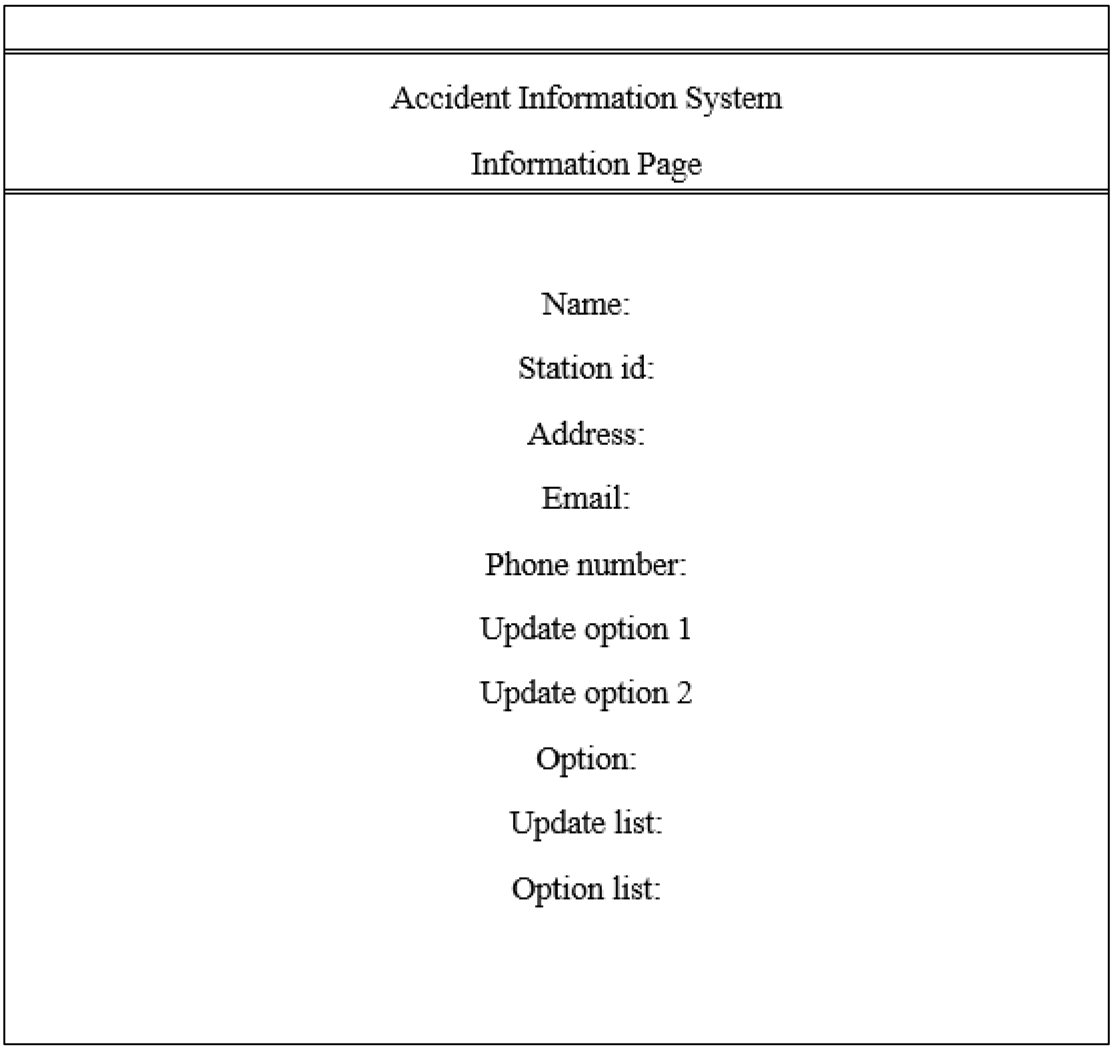
Total Cost:

Average total:

Figure 3.26: Output statistic graph page

[Figure 3.26 is to view graph and list report for current month in database]

3.5.11 Update Information page



Accident Information System

Information Page

Name:

Station id:

Address:

Email:

Phone number:

Update option 1

Update option 2

Option:

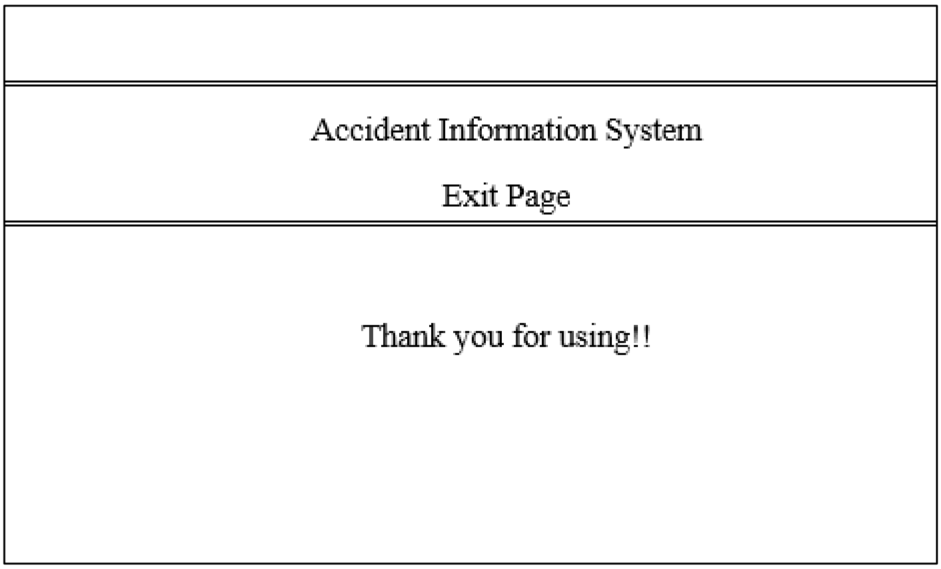
Update list:

Option list:

Figure 3.27: Output update information page

[Figure 3.27 displays the information of police]

3.5.12 Exit page



Accident Information System

Exit Page

Thank you for using!!

Figure 3.28: Output exit page

[Figure 3.28 displays the exit after done use the programs]

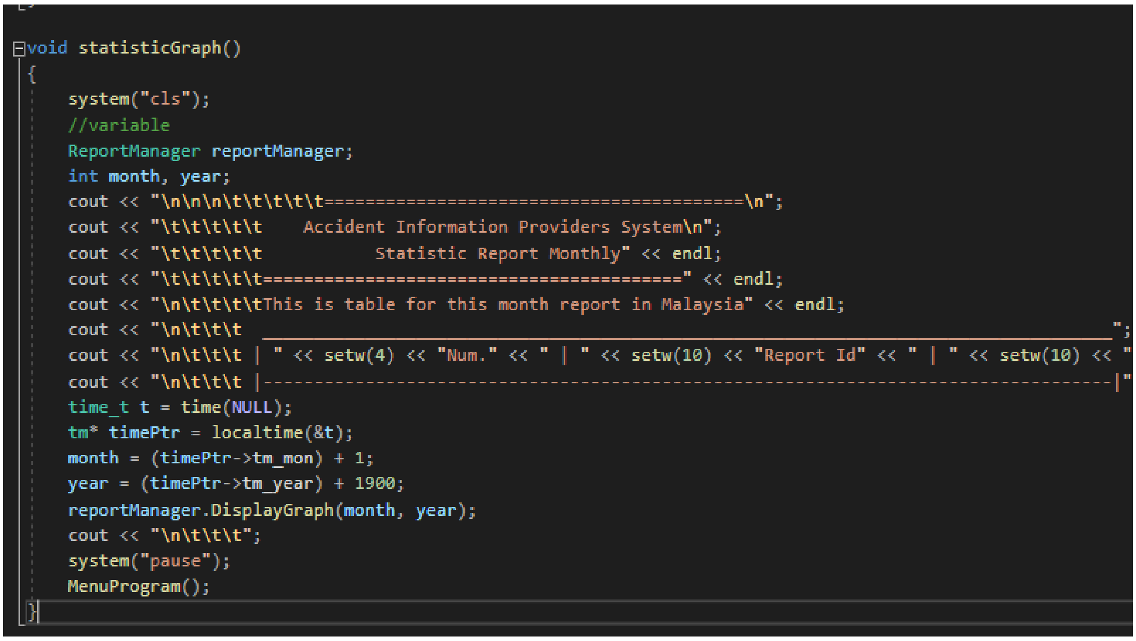
CHAPTER 4: IMPLEMENTATION

4.1 Introduction

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 techniques that is applied such as function, selection, loop, and error handling. The programming technique that applies to this system are show below.

4.2 Programming Techniques

4.2.1 Function

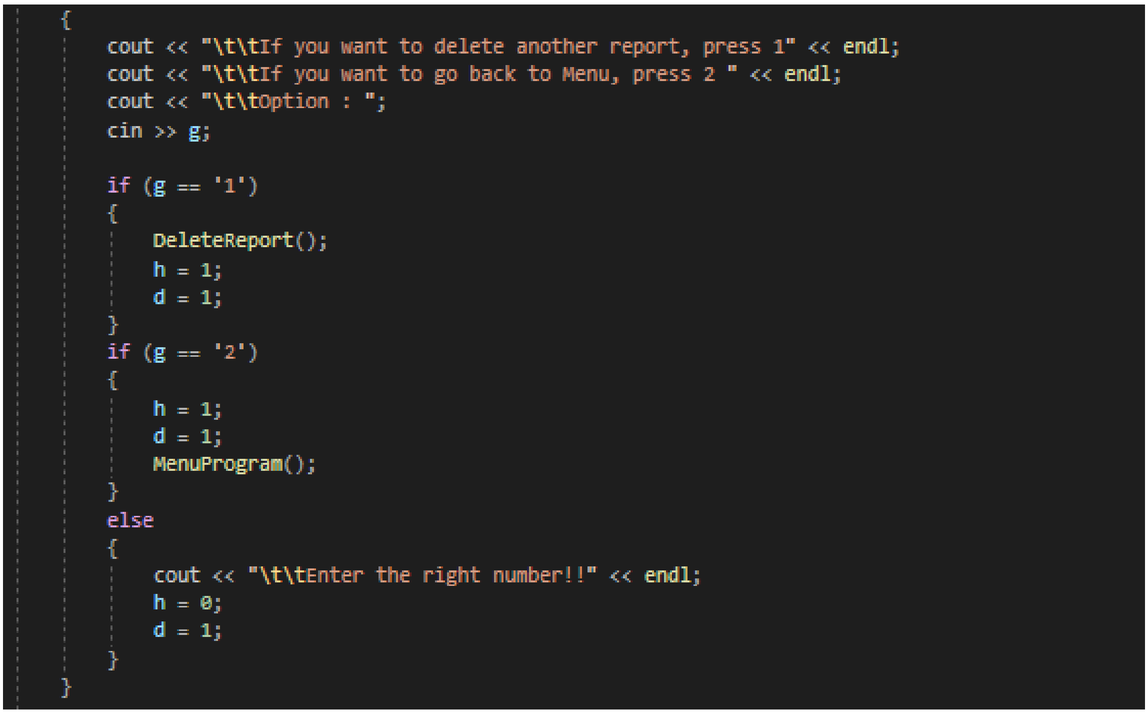


```
void statisticGraph()
{
    system("cls");
    //variable
    ReportManager reportManager;
    int month, year;
    cout << "\n\n\n\t\t\t\t\t=========================================\n";
    cout << "\t\t\t\t\t    Accident Information Providers System\n";
    cout << "\t\t\t\t\t           Statistic Report Monthly" << endl;
    cout << "\t\t\t\t\t=========================================" << endl;
    cout << "\n\t\t\t\tThis is table for this month report in Malaysia" << endl;
    cout << "\n\t\t\t  ____________________________________________________________________________________";
    cout << "\n\t\t\t | " << setw(4) << "Num." << " | " << setw(10) << "Report Id" << " | " << setw(10) << "
    cout << "\n\t\t\t |------------------------------------------------------------------------------------|"
    time_t t = time(NULL);
    tm* timePtr = localtime(&t);
    month = (timePtr->tm_mon) + 1;
    year = (timePtr->tm_year) + 1900;
    reportManager.DisplayGraph(month, year);
    cout << "\n\t\t\t";
    system("pause");
    MenuProgram();
}
```

Figure 4.1: Function implementation

[Function to display statistics report]

4.2.2 Selection

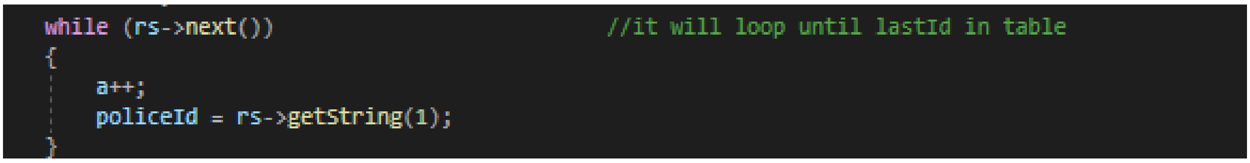


```
{
    cout << "\t\tIf you want to delete another report, press 1" << endl;
    cout << "\t\tIf you want to go back to Menu, press 2 " << endl;
    cout << "\t\tOption : ";
    cin >> g;

    if (g == '1')
    {
        DeleteReport();
        h = 1;
        d = 1;
    }
    if (g == '2')
    {
        h = 1;
        d = 1;
        MenuProgram();
    }
    else
    {
        cout << "\t\tEnter the right number!!" << endl;
        h = 0;
        d = 1;
    }
}
```

Figure 4.2: Selection implementation

[A selection to choose the page option in delete report function]

4.2.3 Looping

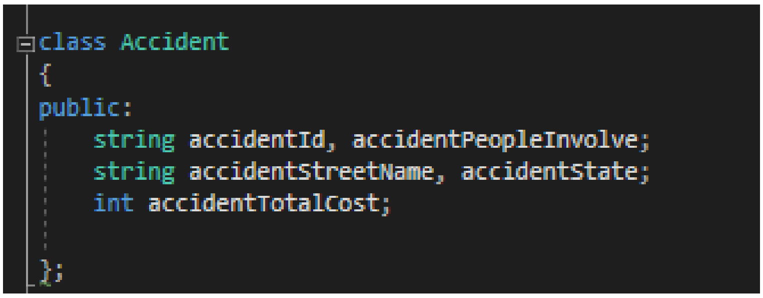


```
while (rs->next())                          //it will loop until lastId in table
{
    a++;
    policeId = rs->getString(1);
}
```

Figure 4.3: Looping implementation

[Looping for getting the last data of id police from table police]

4.2.4 Class

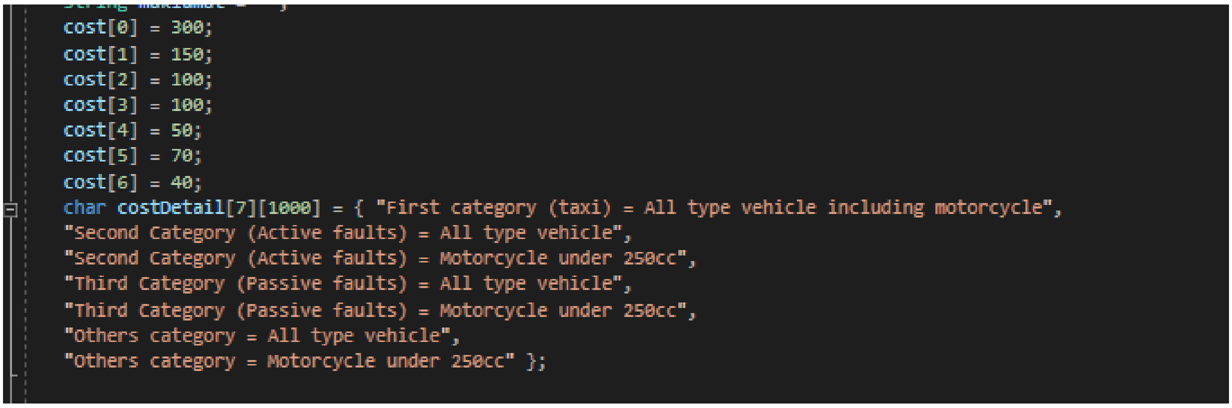


```
class Accident
{
public:
    string accidentId, accidentPeopleInvolve;
    string accidentStreetName, accidentState;
    int accidentTotalCost;

};
```

Figure 4.4: Class implementation

[A class accident to easily access accident’s information]

4.2.5 Array

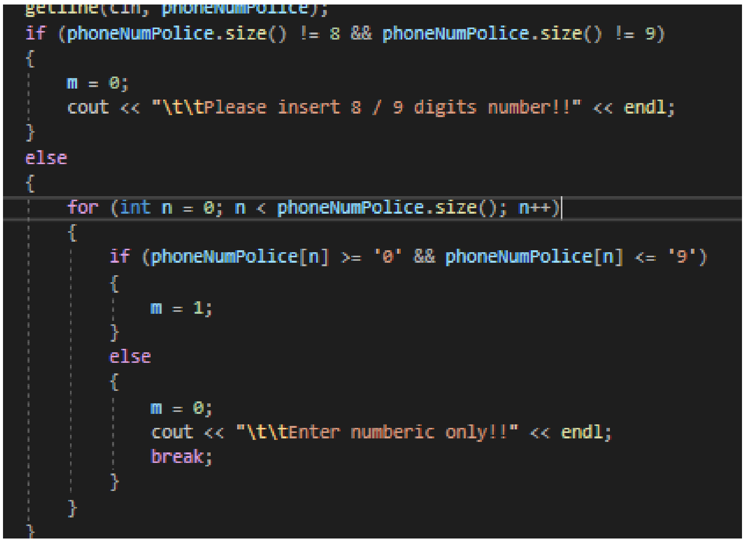


```
cost[0] = 300;
cost[1] = 150;
cost[2] = 100;
cost[3] = 100;
cost[4] = 50;
cost[5] = 70;
cost[6] = 40;
char costDetail[7][1000] = { "First category (taxi) = All type vehicle including motorcycle",
"Second Category (Active faults) = All type vehicle",
"Second Category (Active faults) = Motorcycle under 250cc",
"Third Category (Passive faults) = All type vehicle",
"Third Category (Passive faults) = Motorcycle under 250cc",
"Others category = All type vehicle",
"Others category = Motorcycle under 250cc" };
```

Figure 4.5: Array implementation

[An array to store cost and details]

4.2.6 Error Handling



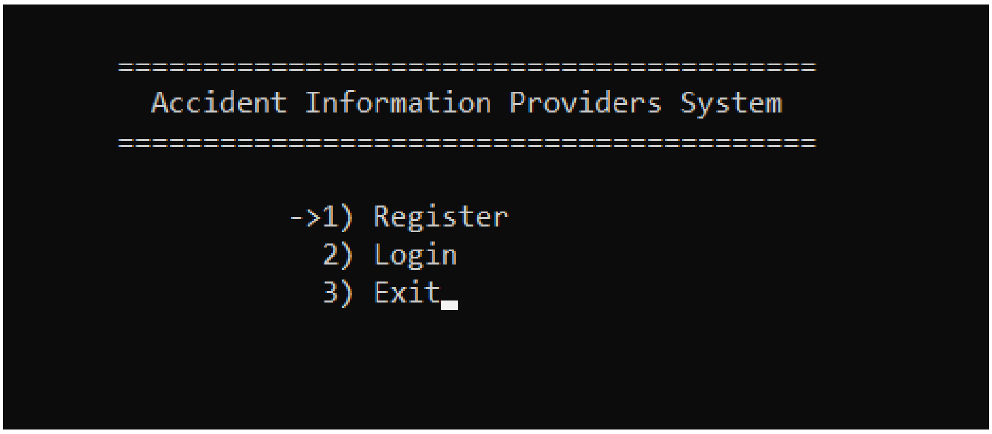
```
getline(cin, phoneNumPolice);
if (phoneNumPolice.size() != 8 && phoneNumPolice.size() != 9)
{
    m = 0;
    cout << "\t\tPlease insert 8 / 9 digits number!!" << endl;
}
else
{
    for (int n = 0; n < phoneNumPolice.size(); n++)
    {
        if (phoneNumPolice[n] >= '0' && phoneNumPolice[n] <= '9')
        {
            m = 1;
        }
        else
        {
            m = 0;
            cout << "\t\tEnter numberic only!!" << endl;
            break;
        }
    }
}
```

Figure 4.6: Error handling implementation

[Error handling code to make sure that the user only input the numeric and enough length]

4.3 User Interface

4.3.1 Main Menu

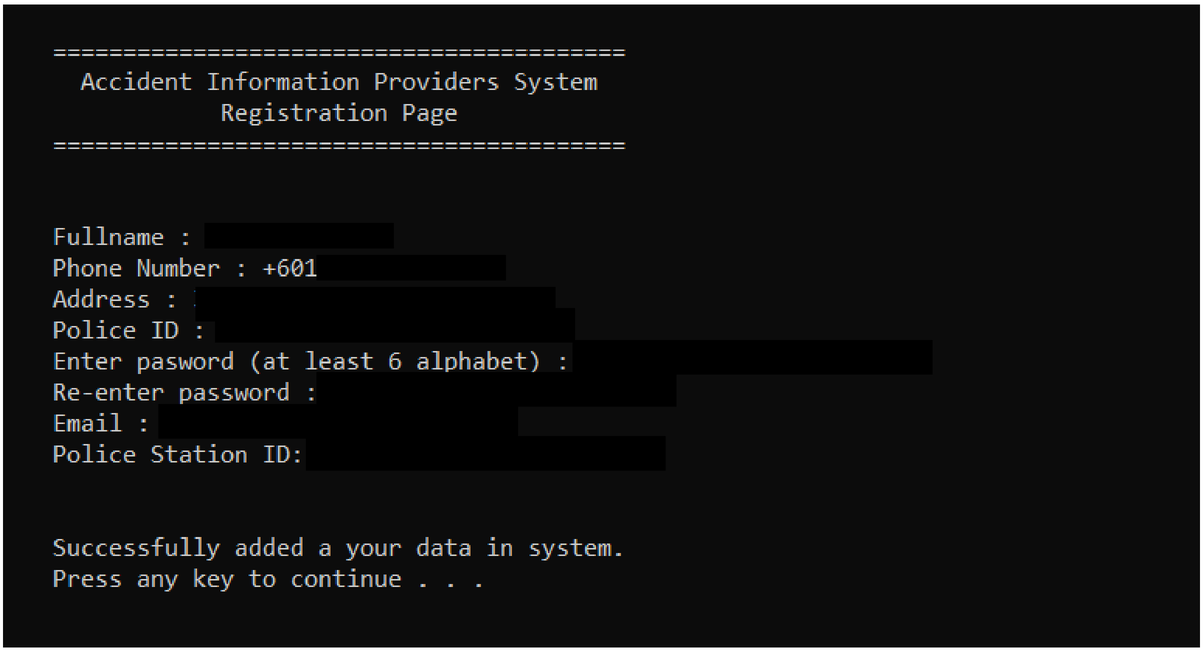


```
 ==========================================
   Accident Information Providers System
 ==========================================

            ->1) Register
              2) Login
              3) Exit_
```

Figure 4.7: Output login, register and exit option

[Figure 4.7 is the start-up page for the system]

4.3.2 Register page

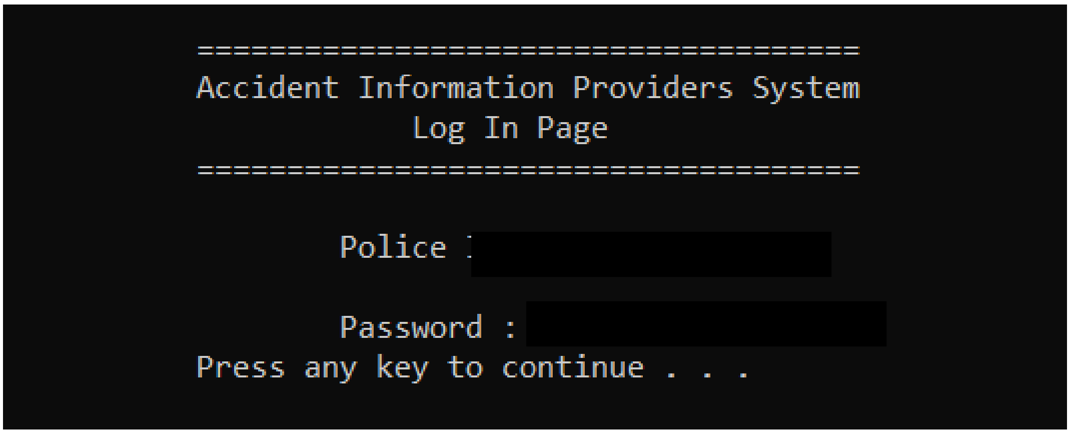


```
=========================================
  Accident Information Providers System
            Registration Page
=========================================


Fullname : [redacted]
Phone Number : +601[redacted]
Address : [redacted]
Police ID : [redacted]
Enter pasword (at least 6 alphabet) : [redacted]
Re-enter password : [redacted]
Email : [redacted]
Police Station ID:[redacted]


Successfully added a your data in system.
Press any key to continue . . .
```

Figure 4.8: Output register page

[Figure 4.8 is the register for new police. It's show that program already added new police in database]

4.3.3 Login Page

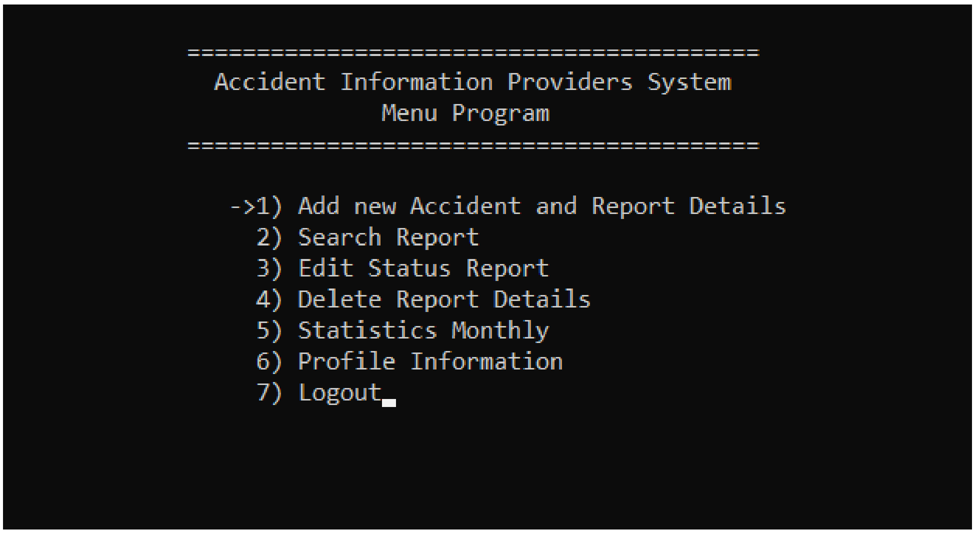


```
=====================================
Accident Information Providers System
           Log In Page
=====================================

        Police [redacted]

        Password : [redacted]
Press any key to continue . . .
```

Figure 4.9: Output Login page

[Figure 4.9 is login page for user. User can use Police id given and password when at register part]

4.3.4 Menu Program Page

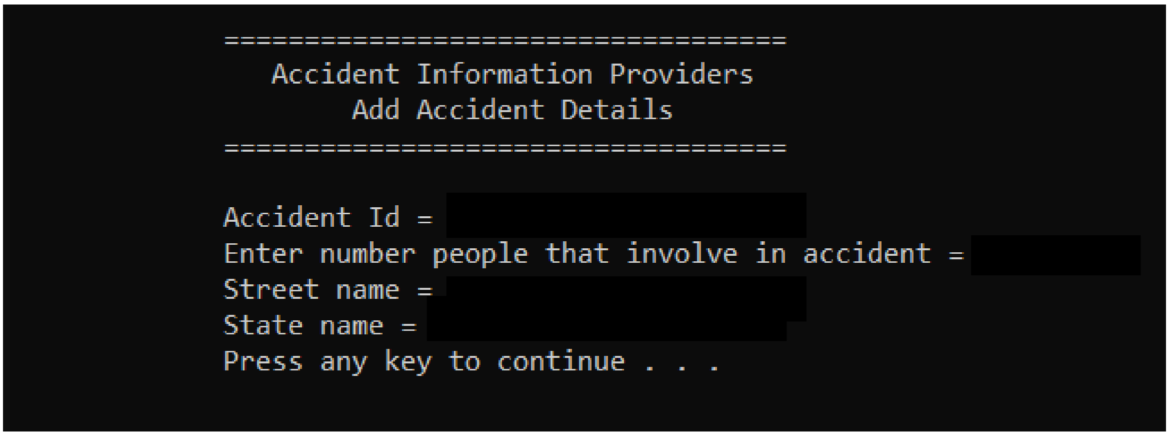


```
=========================================
  Accident Information Providers System
              Menu Program
=========================================

   ->1) Add new Accident and Report Details
     2) Search Report
     3) Edit Status Report
     4) Delete Report Details
     5) Statistics Monthly
     6) Profile Information
     7) Logout
```

Figure 4.10: Output Menu Program page

[Figure 4.10 is for Menu Program]

4.3.5 Add accident page

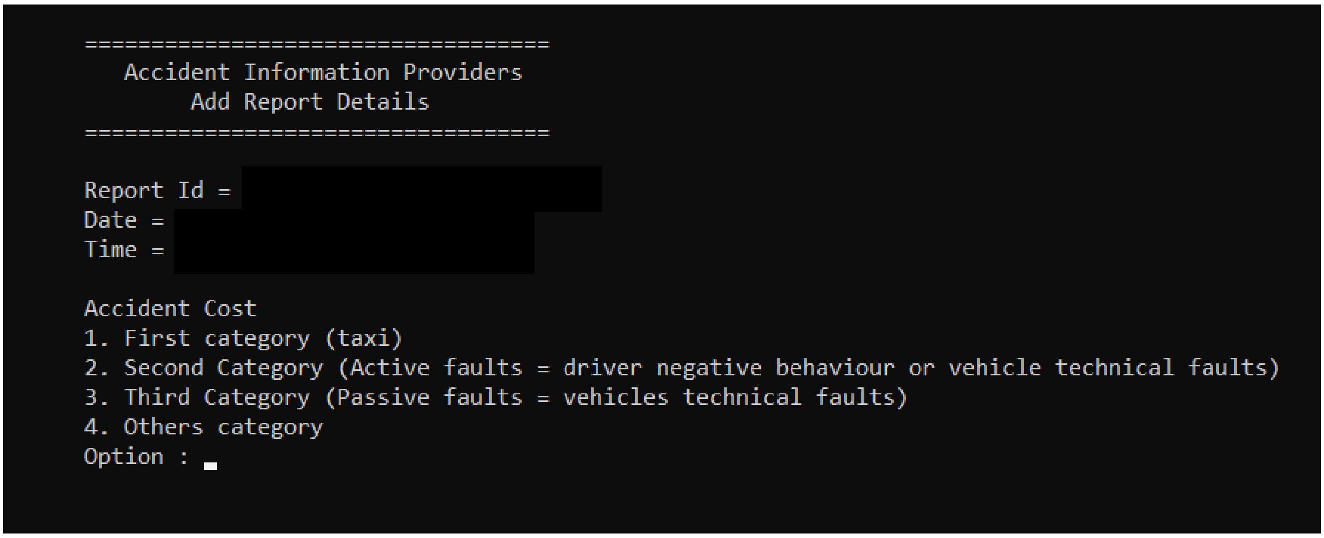


```
===================================
   Accident Information Providers
        Add Accident Details
===================================

Accident Id = [redacted]
Enter number people that involve in accident = [redacted]
Street name = [redacted]
State name = [redacted]
Press any key to continue . . .
```

Figure 4.11: Output add new accident page

[Figure 4.11 is adding new accident]

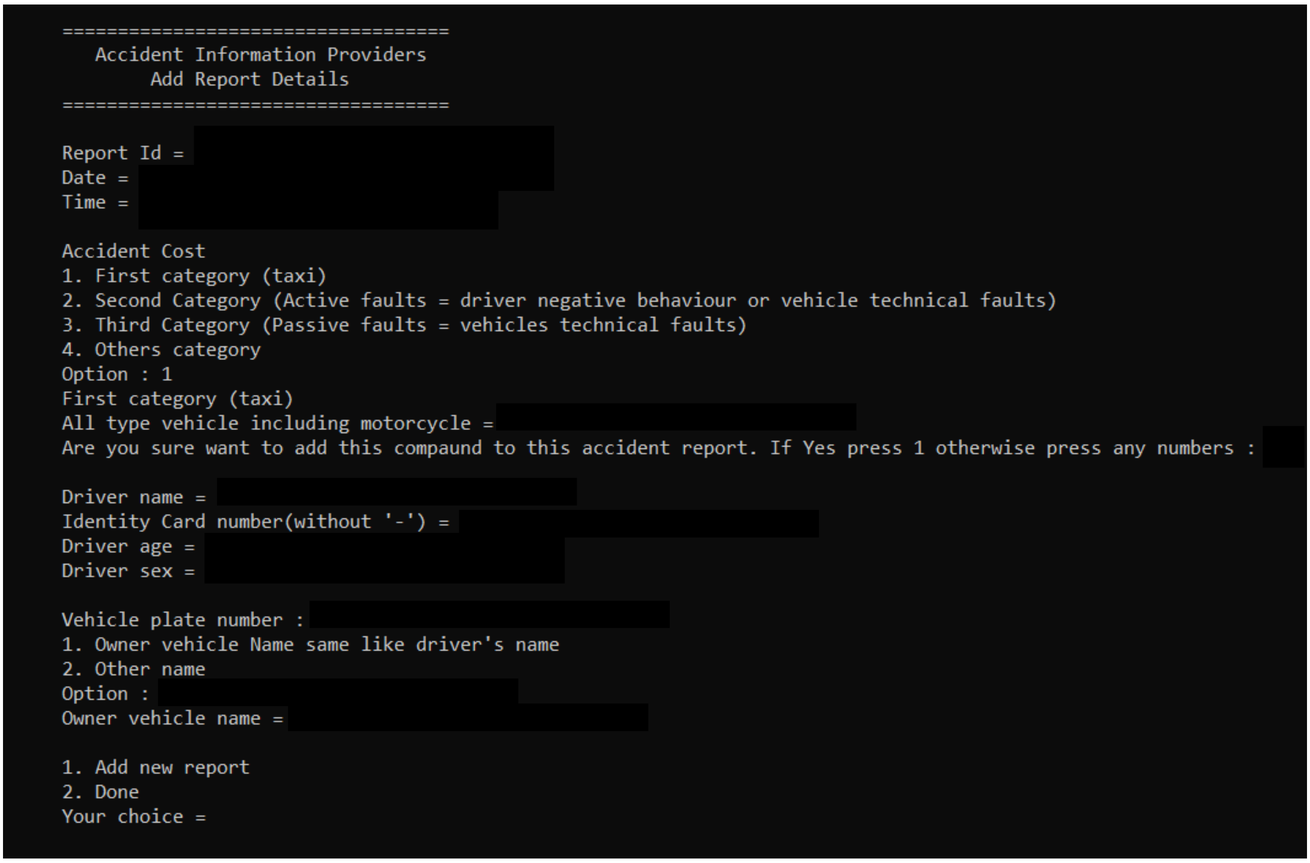
4.3.6 Add Report Details page



```
===================================
   Accident Information Providers
        Add Report Details
===================================

Report Id = [redacted]
Date = [redacted]
Time =

Accident Cost
1. First category (taxi)
2. Second Category (Active faults = driver negative behaviour or vehicle technical faults)
3. Third Category (Passive faults = vehicles technical faults)
4. Others category
Option :
```

Figure 4.12: Output add new report page

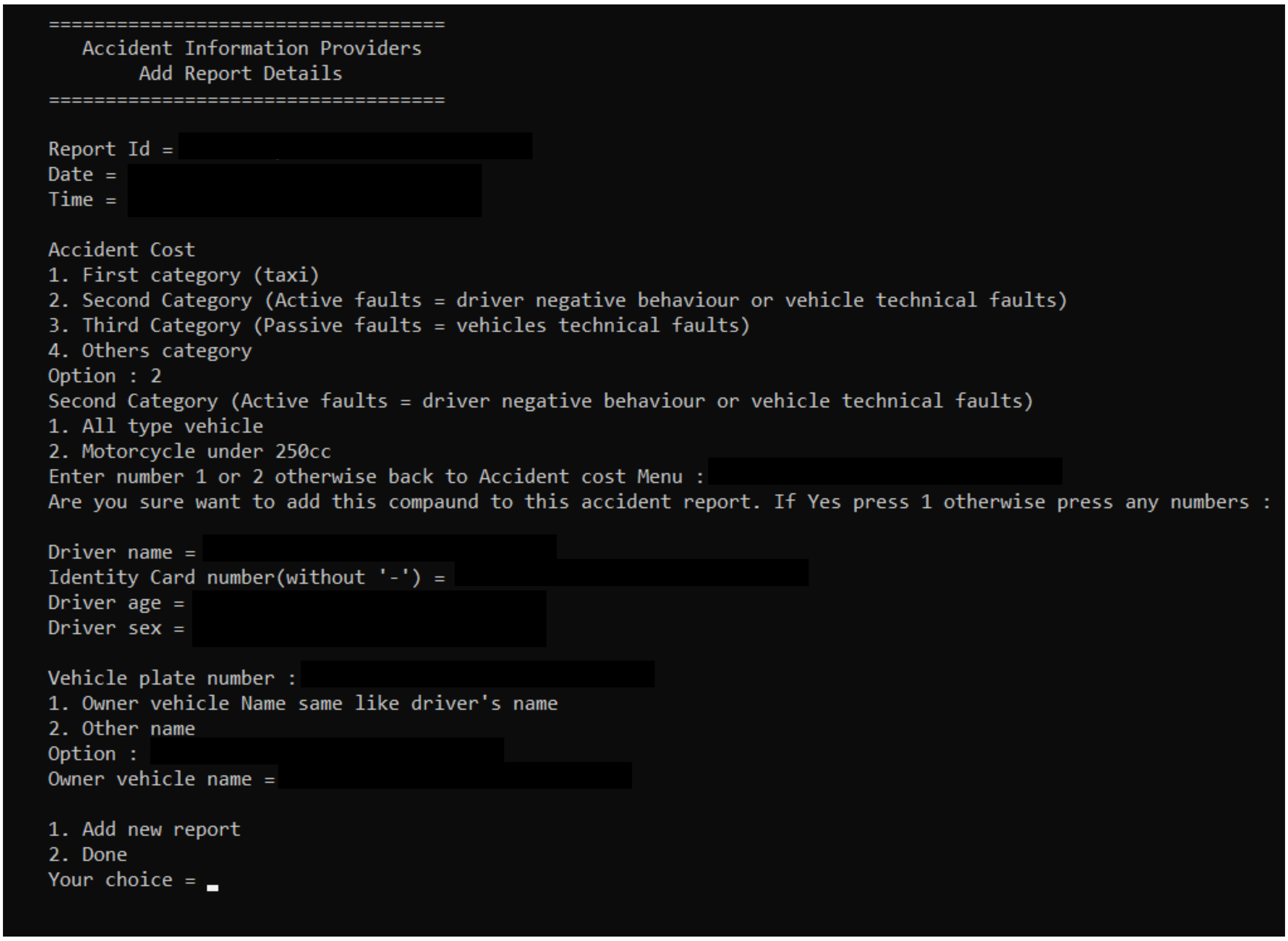
[Figure 4.12 is adding new report]



```
===================================
   Accident Information Providers
        Add Report Details
===================================

Report Id = [redacted]
Date =
Time = [redacted]

Accident Cost
1. First category (taxi)
2. Second Category (Active faults = driver negative behaviour or vehicle technical faults)
3. Third Category (Passive faults = vehicles technical faults)
4. Others category
Option : 1
First category (taxi)
All type vehicle including motorcycle = [redacted]
Are you sure want to add this compaund to this accident report. If Yes press 1 otherwise press any numbers : [redacted]

Driver name = [redacted]
Identity Card number(without '-') = [redacted]
Driver age = [redacted]
Driver sex =

Vehicle plate number : [redacted]
1. Owner vehicle Name same like driver's name
2. Other name
Option : [redacted]
Owner vehicle name = [redacted]

1. Add new report
2. Done
Your choice =
```

Figure 4.13: Output add new report, driver, and vehicle by choosing first cost details page

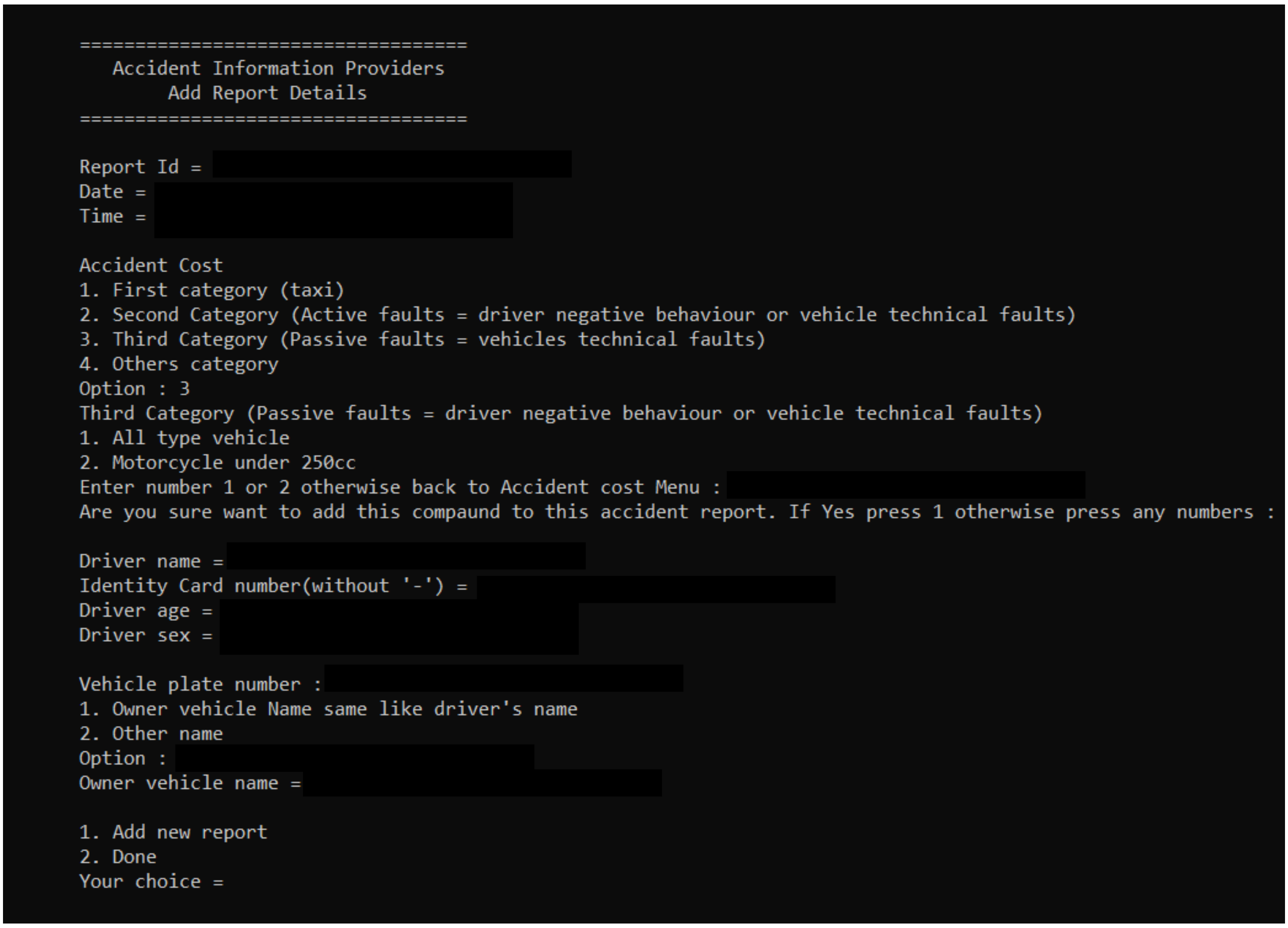
[Figure 4.13 is adding new report, driver, and vehicle for first cost detail]



```
===================================
   Accident Information Providers
        Add Report Details
===================================

Report Id = [redacted]
Date = [redacted]
Time =

Accident Cost
1. First category (taxi)
2. Second Category (Active faults = driver negative behaviour or vehicle technical faults)
3. Third Category (Passive faults = vehicles technical faults)
4. Others category
Option : 2
Second Category (Active faults = driver negative behaviour or vehicle technical faults)
1. All type vehicle
2. Motorcycle under 250cc
Enter number 1 or 2 otherwise back to Accident cost Menu : [redacted]
Are you sure want to add this compaund to this accident report. If Yes press 1 otherwise press any numbers :

Driver name = [redacted]
Identity Card number(without '-') = [redacted]
Driver age = [redacted]
Driver sex =

Vehicle plate number : [redacted]
1. Owner vehicle Name same like driver's name
2. Other name
Option : [redacted]
Owner vehicle name = [redacted]

1. Add new report
2. Done
Your choice =
```

Figure 4.14: Output add new report, driver, and vehicle by choosing second cost details page

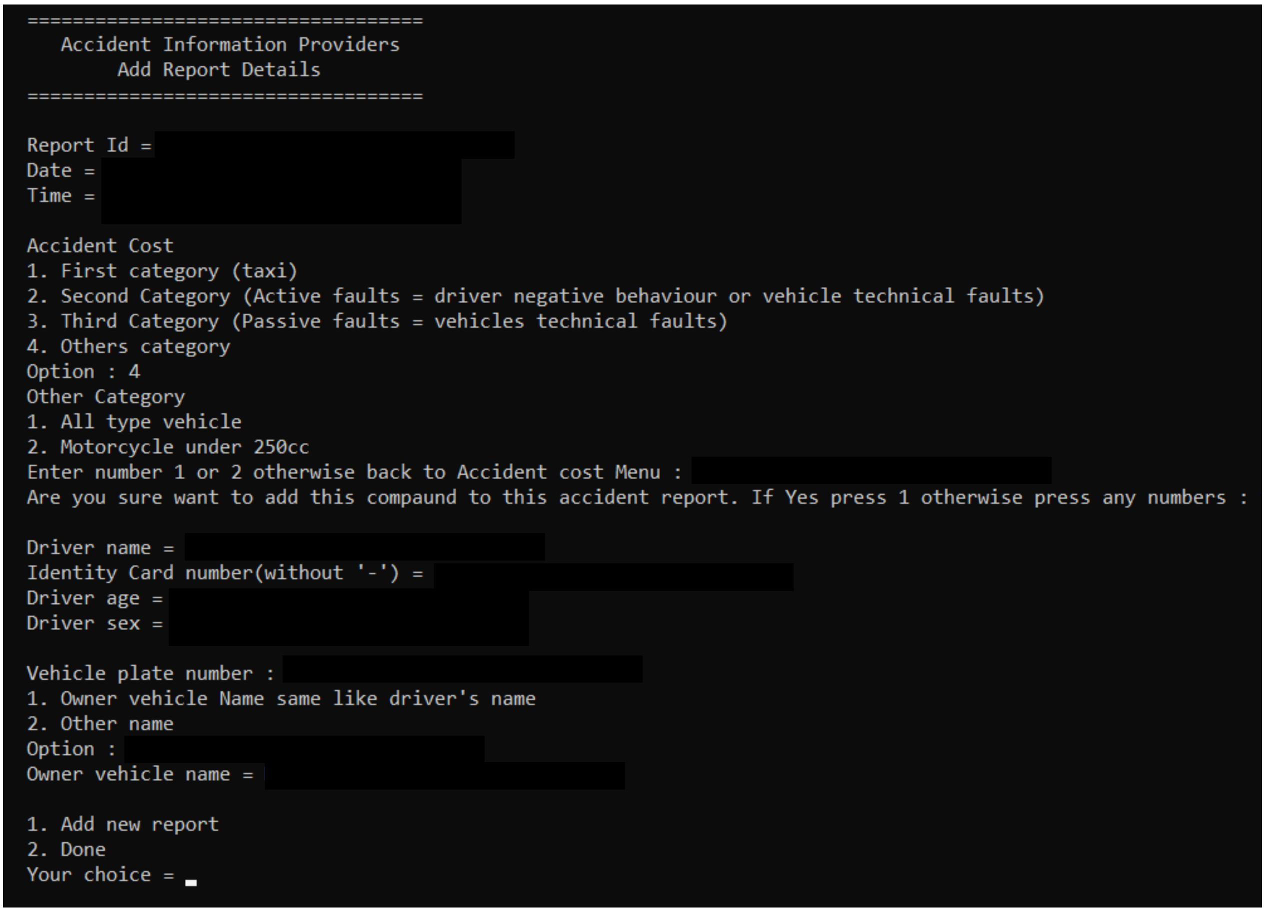
[Figure 4.14 is adding new report, driver, and vehicle for second cost detail]



```
===================================
   Accident Information Providers
        Add Report Details
===================================

Report Id = [redacted]
Date = [redacted]
Time =

Accident Cost
1. First category (taxi)
2. Second Category (Active faults = driver negative behaviour or vehicle technical faults)
3. Third Category (Passive faults = vehicles technical faults)
4. Others category
Option : 3
Third Category (Passive faults = driver negative behaviour or vehicle technical faults)
1. All type vehicle
2. Motorcycle under 250cc
Enter number 1 or 2 otherwise back to Accident cost Menu : [redacted]
Are you sure want to add this compaund to this accident report. If Yes press 1 otherwise press any numbers :

Driver name = [redacted]
Identity Card number(without '-') = [redacted]
Driver age = [redacted]
Driver sex =

Vehicle plate number : [redacted]
1. Owner vehicle Name same like driver's name
2. Other name
Option : [redacted]
Owner vehicle name = [redacted]

1. Add new report
2. Done
Your choice =
```

Figure 4.15: Output add new report, driver, and vehicle by choosing third cost details page

[Figure 4.15 is adding new report, driver, and vehicle for third cost detail]

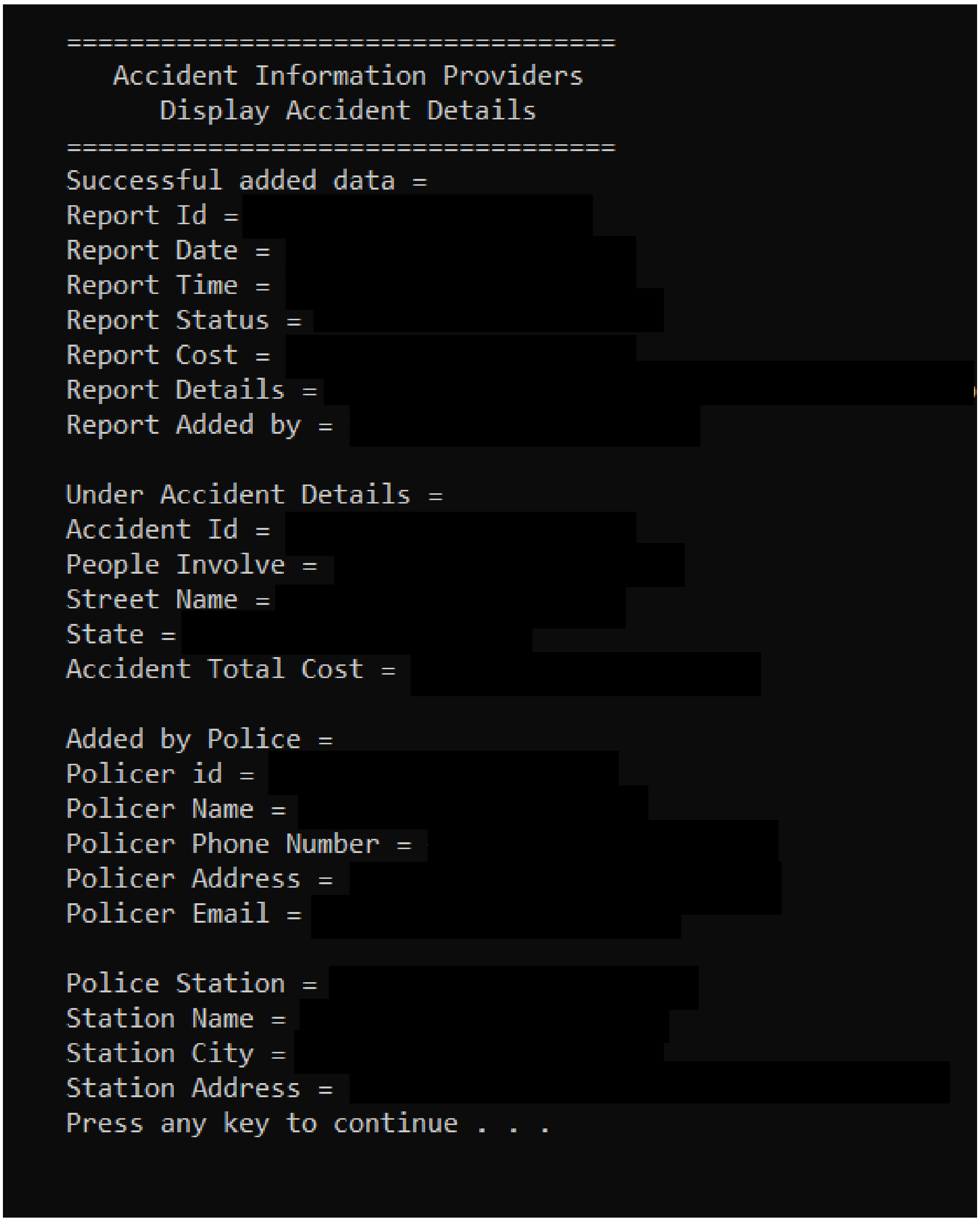


```
===================================
   Accident Information Providers
        Add Report Details
===================================

Report Id = [redacted]
Date = [redacted]
Time =

Accident Cost
1. First category (taxi)
2. Second Category (Active faults = driver negative behaviour or vehicle technical faults)
3. Third Category (Passive faults = vehicles technical faults)
4. Others category
Option : 4
Other Category
1. All type vehicle
2. Motorcycle under 250cc
Enter number 1 or 2 otherwise back to Accident cost Menu : [redacted]
Are you sure want to add this compaund to this accident report. If Yes press 1 otherwise press any numbers :

Driver name = [redacted]
Identity Card number(without '-') = [redacted]
Driver age = [redacted]
Driver sex =

Vehicle plate number : [redacted]
1. Owner vehicle Name same like driver's name
2. Other name
Option : [redacted]
Owner vehicle name = [redacted]

1. Add new report
2. Done
Your choice = _
```

Figure 4.16: Output add new report, driver, and vehicle by choosing fourth cost details page

[Figure 4.16 is adding new report, driver, and vehicle for fourth cost detail]

4.3.7 Viewing details after adding into database page

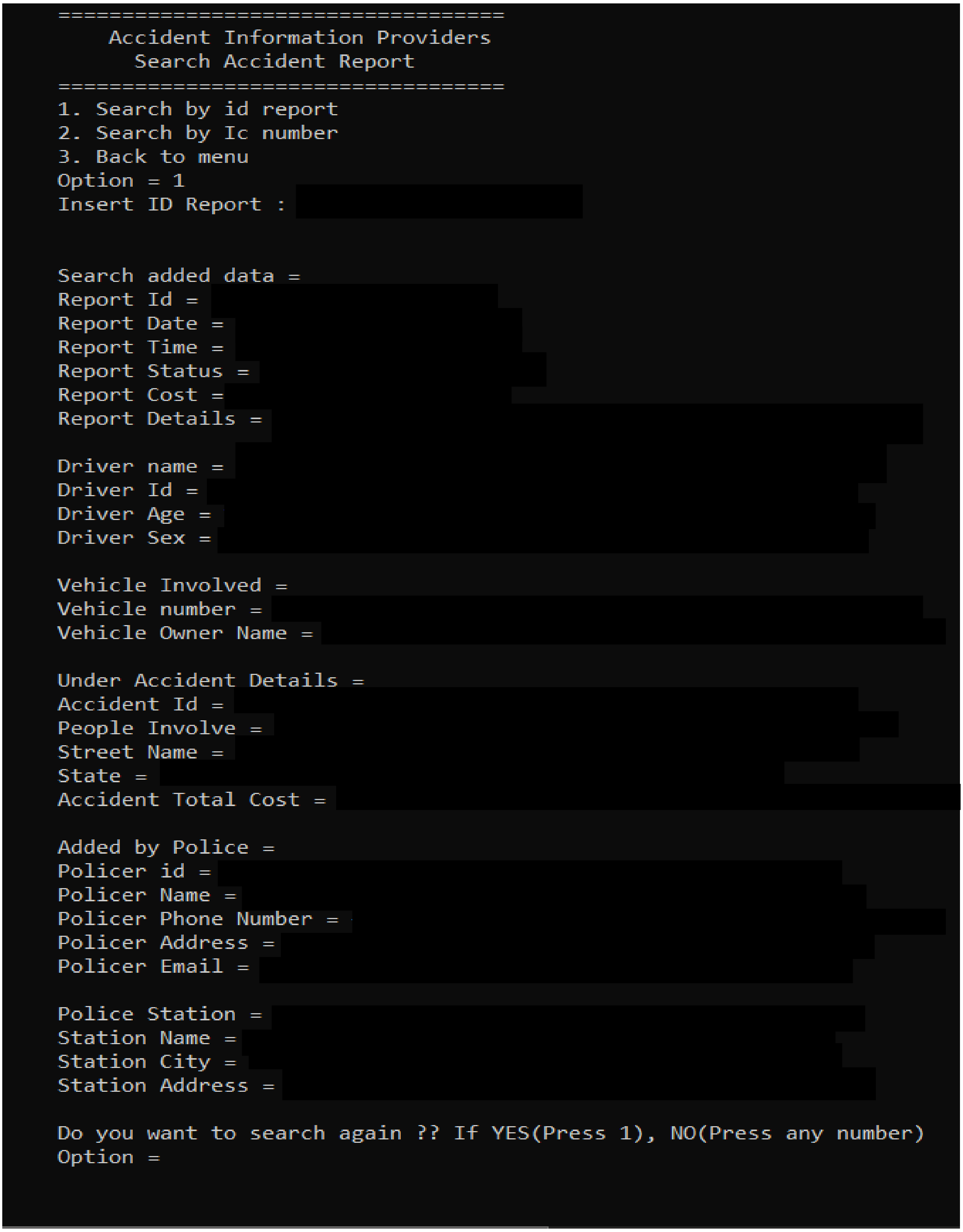


```
===================================
   Accident Information Providers
      Display Accident Details
===================================
Successful added data =
Report Id = [redacted]
Report Date = [redacted]
Report Time = [redacted]
Report Status = [redacted]
Report Cost = [redacted]
Report Details = [redacted]
Report Added by = [redacted]

Under Accident Details =
Accident Id = [redacted]
People Involve = [redacted]
Street Name = [redacted]
State = [redacted]
Accident Total Cost = [redacted]

Added by Police =
Policer id = [redacted]
Policer Name = [redacted]
Policer Phone Number = [redacted]
Policer Address = [redacted]
Policer Email = [redacted]

Police Station = [redacted]
Station Name = [redacted]
Station City = [redacted]
Station Address = [redacted]
Press any key to continue . . .
```

Figure 4.17: Output viewing details after added page

[Figure 4.17 is display data after adding details. So, user can check again the info that already insert.]

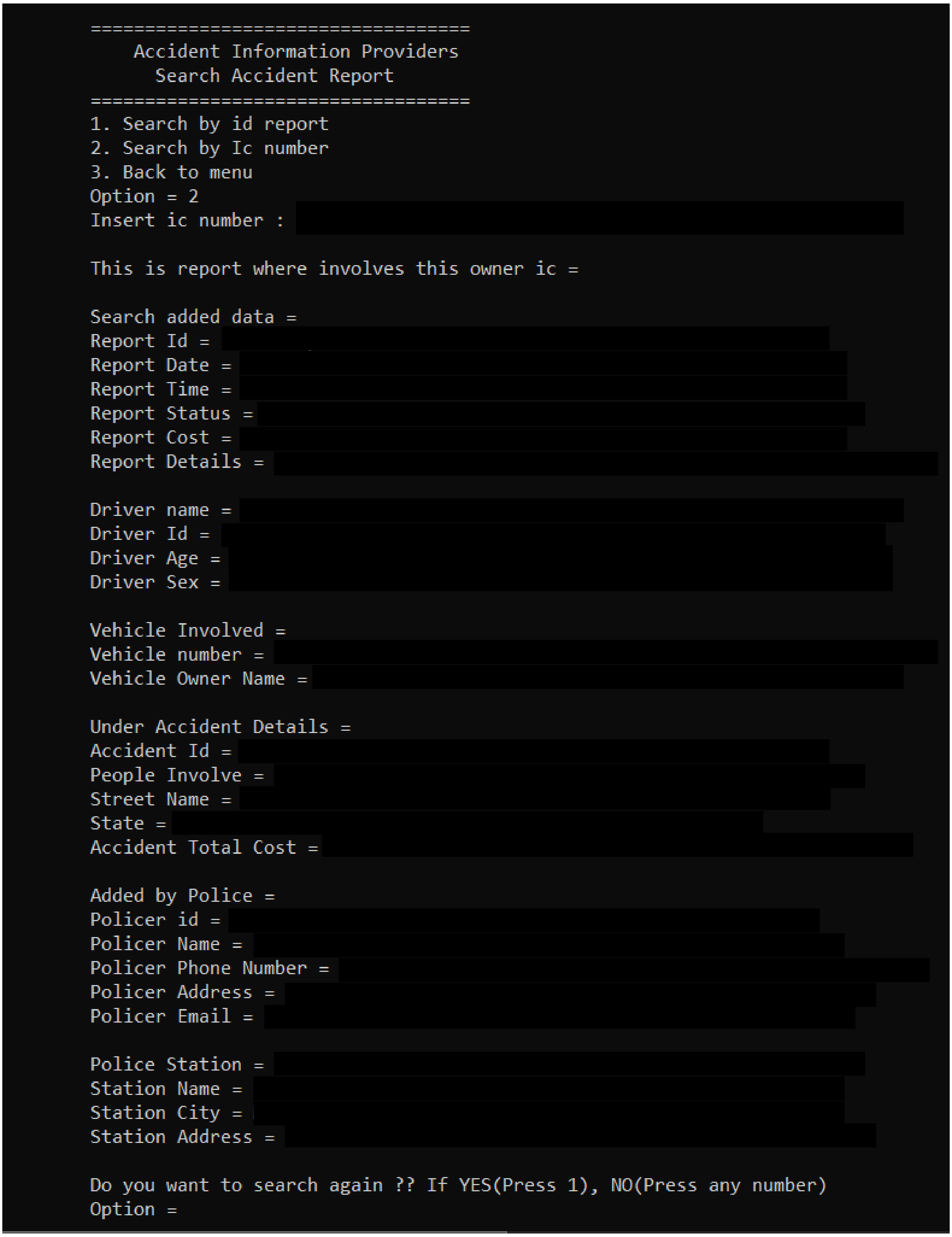
4.3.8 Search Report



```
===================================
    Accident Information Providers
      Search Accident Report
===================================
1. Search by id report
2. Search by Ic number
3. Back to menu
Option = 1
Insert ID Report : [redacted]


Search added data =
Report Id = [redacted]
Report Date = [redacted]
Report Time = [redacted]
Report Status = [redacted]
Report Cost = [redacted]
Report Details = [redacted]

Driver name = [redacted]
Driver Id = [redacted]
Driver Age = [redacted]
Driver Sex = [redacted]

Vehicle Involved =
Vehicle number = [redacted]
Vehicle Owner Name = [redacted]

Under Accident Details =
Accident Id = [redacted]
People Involve = [redacted]
Street Name = [redacted]
State = [redacted]
Accident Total Cost = [redacted]

Added by Police =
Policer id = [redacted]
Policer Name = [redacted]
Policer Phone Number = [redacted]
Policer Address = [redacted]
Policer Email = [redacted]

Police Station = [redacted]
Station Name = [redacted]
Station City = [redacted]
Station Address = [redacted]

Do you want to search again ?? If YES(Press 1), NO(Press any number)
Option =
```

Figure 4.18: Search Report by id report

[Figure 4.18 is the search report by id report.]

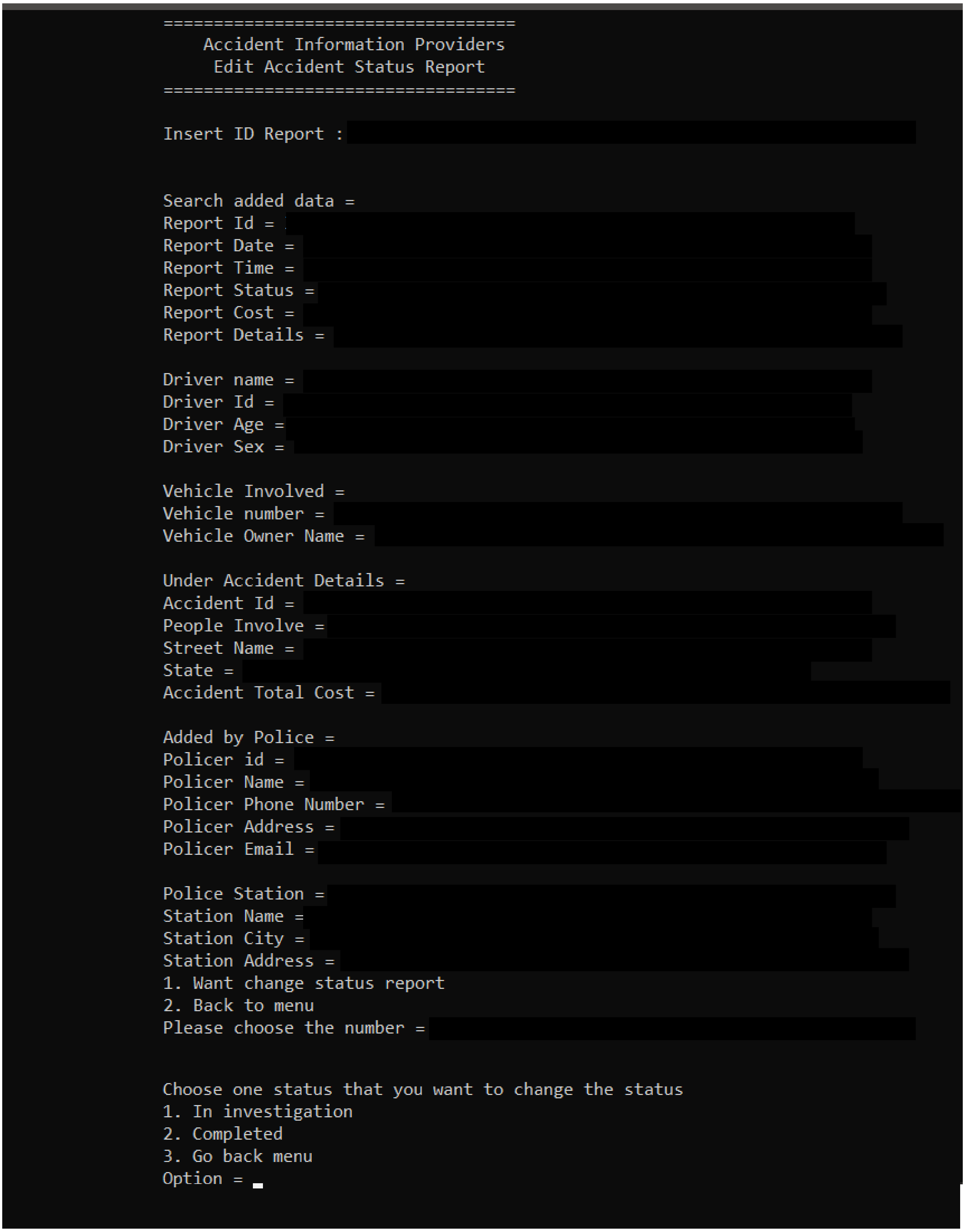


```
===================================
    Accident Information Providers
      Search Accident Report
===================================
1. Search by id report
2. Search by Ic number
3. Back to menu
Option = 2
Insert ic number : [redacted]

This is report where involves this owner ic =

Search added data =
Report Id = [redacted]
Report Date = [redacted]
Report Time = [redacted]
Report Status = [redacted]
Report Cost = [redacted]
Report Details = [redacted]

Driver name = [redacted]
Driver Id = [redacted]
Driver Age = [redacted]
Driver Sex = [redacted]

Vehicle Involved =
Vehicle number = [redacted]
Vehicle Owner Name = [redacted]

Under Accident Details =
Accident Id = [redacted]
People Involve = [redacted]
Street Name = [redacted]
State = [redacted]
Accident Total Cost = [redacted]

Added by Police =
Policer id = [redacted]
Policer Name = [redacted]
Policer Phone Number = [redacted]
Policer Address = [redacted]
Policer Email = [redacted]

Police Station = [redacted]
Station Name = [redacted]
Station City = [redacted]
Station Address = [redacted]

Do you want to search again ?? If YES(Press 1), NO(Press any number)
Option =
```

Figure 4.19: Search Report by ic number

[Figure 4.19 is the search report by ic number.]

4.3.9 Edit Status Report page

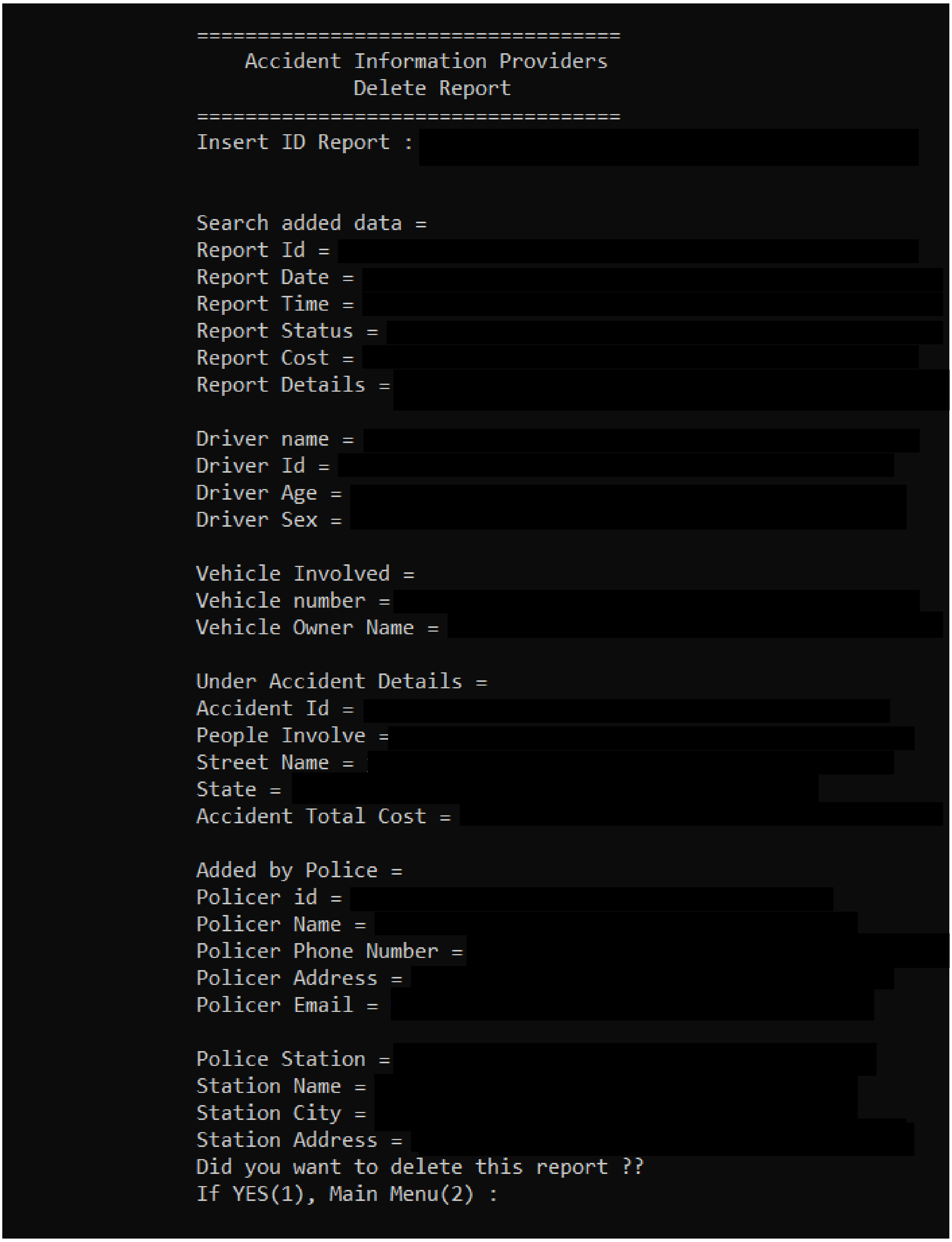


```
===================================
    Accident Information Providers
     Edit Accident Status Report
===================================

Insert ID Report : [redacted]


Search added data =
Report Id = [redacted]
Report Date = [redacted]
Report Time = [redacted]
Report Status = [redacted]
Report Cost = [redacted]
Report Details = [redacted]

Driver name = [redacted]
Driver Id = [redacted]
Driver Age = [redacted]
Driver Sex = [redacted]

Vehicle Involved =
Vehicle number = [redacted]
Vehicle Owner Name = [redacted]

Under Accident Details =
Accident Id = [redacted]
People Involve = [redacted]
Street Name = [redacted]
State = [redacted]
Accident Total Cost = [redacted]

Added by Police =
Policer id = [redacted]
Policer Name = [redacted]
Policer Phone Number = [redacted]
Policer Address = [redacted]
Policer Email = [redacted]

Police Station = [redacted]
Station Name = [redacted]
Station City = [redacted]
Station Address = [redacted]
1. Want change status report
2. Back to menu
Please choose the number = [redacted]


Choose one status that you want to change the status
1. In investigation
2. Completed
3. Go back menu
Option = _
```

Figure 4.20: Output edit status report page

[Figure 4.20 is to edit status report]

4.3.11 Delete id report page

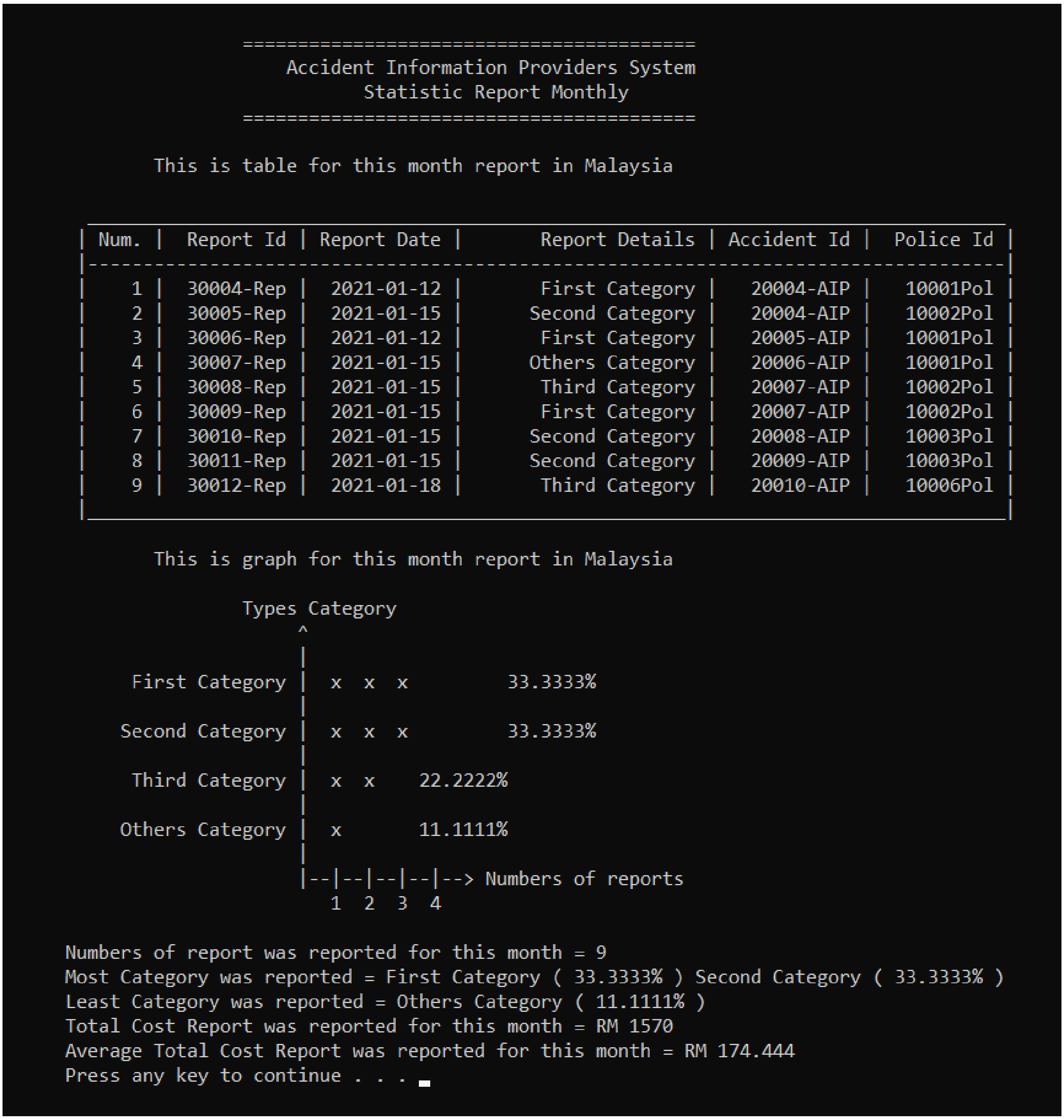


```
===================================
    Accident Information Providers
             Delete Report
===================================
Insert ID Report : [redacted]


Search added data =
Report Id = [redacted]
Report Date = [redacted]
Report Time = [redacted]
Report Status = [redacted]
Report Cost = [redacted]
Report Details = [redacted]

Driver name = [redacted]
Driver Id = [redacted]
Driver Age = [redacted]
Driver Sex = [redacted]

Vehicle Involved =
Vehicle number = [redacted]
Vehicle Owner Name = [redacted]

Under Accident Details =
Accident Id = [redacted]
People Involve = [redacted]
Street Name = [redacted]
State = [redacted]
Accident Total Cost = [redacted]

Added by Police =
Policer id = [redacted]
Policer Name = [redacted]
Policer Phone Number = [redacted]
Policer Address = [redacted]
Policer Email = [redacted]

Police Station = [redacted]
Station Name = [redacted]
Station City = [redacted]
Station Address = [redacted]
Did you want to delete this report ??
If YES(1), Main Menu(2) :
```

Figure 4.21: Output delete id report page

[Figure 4.21 is to delete report, vehicle and drivers details in database under the inserted id]

4.3.12 Statistics graph page

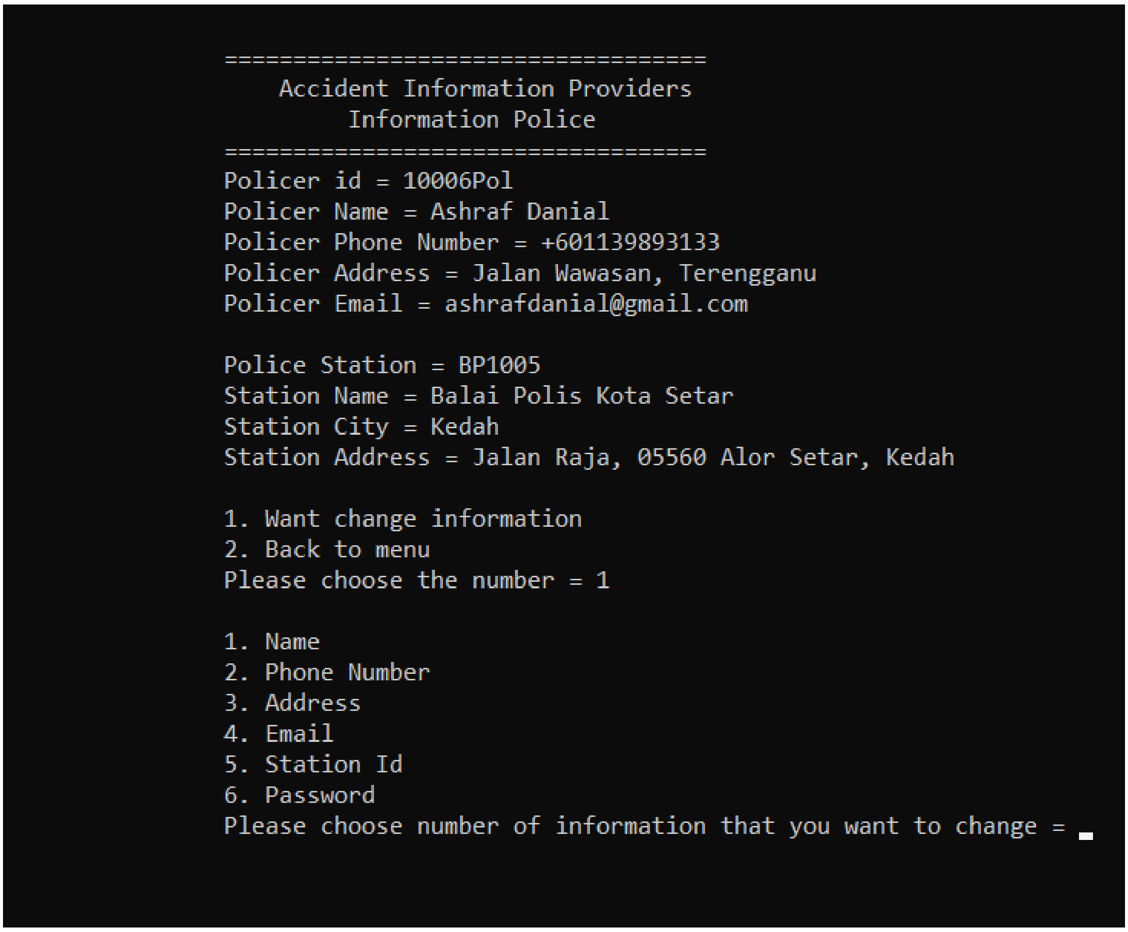


```
                     =========================================
                         Accident Information Providers System
                                Statistic Report Monthly
                     =========================================

             This is table for this month report in Malaysia

       ___________________________________________________________________________________
      | Num. |  Report Id | Report Date |       Report Details | Accident Id |  Police Id |
      |-----------------------------------------------------------------------------------|
      |    1 |  30004-Rep |  2021-01-12 |       First Category |   20004-AIP |   10001Pol |
      |    2 |  30005-Rep |  2021-01-15 |      Second Category |   20004-AIP |   10002Pol |
      |    3 |  30006-Rep |  2021-01-12 |       First Category |   20005-AIP |   10001Pol |
      |    4 |  30007-Rep |  2021-01-15 |      Others Category |   20006-AIP |   10001Pol |
      |    5 |  30008-Rep |  2021-01-15 |       Third Category |   20007-AIP |   10002Pol |
      |    6 |  30009-Rep |  2021-01-15 |       First Category |   20007-AIP |   10002Pol |
      |    7 |  30010-Rep |  2021-01-15 |      Second Category |   20008-AIP |   10003Pol |
      |    8 |  30011-Rep |  2021-01-15 |      Second Category |   20009-AIP |   10003Pol |
      |    9 |  30012-Rep |  2021-01-18 |       Third Category |   20010-AIP |   10006Pol |
      |___________________________________________________________________________________|

             This is graph for this month report in Malaysia

                     Types Category
                          ^
                          |
           First Category |  x  x  x         33.3333%
                          |
          Second Category |  x  x  x         33.3333%
                          |
           Third Category |  x  x    22.2222%
                          |
          Others Category |  x       11.1111%
                          |
                          |--|--|--|--|--> Numbers of reports
                             1  2  3  4

     Numbers of report was reported for this month = 9
     Most Category was reported = First Category ( 33.3333% ) Second Category ( 33.3333% )
     Least Category was reported = Others Category ( 11.1111% )
     Total Cost Report was reported for this month = RM 1570
     Average Total Cost Report was reported for this month = RM 174.444
     Press any key to continue . . .
```

Figure 4.22: Output statistics page

[Figure 4.22 is to view list monthly report and graph was reported in Malaysia. It will show list table and graph]

4.3.12 Information page

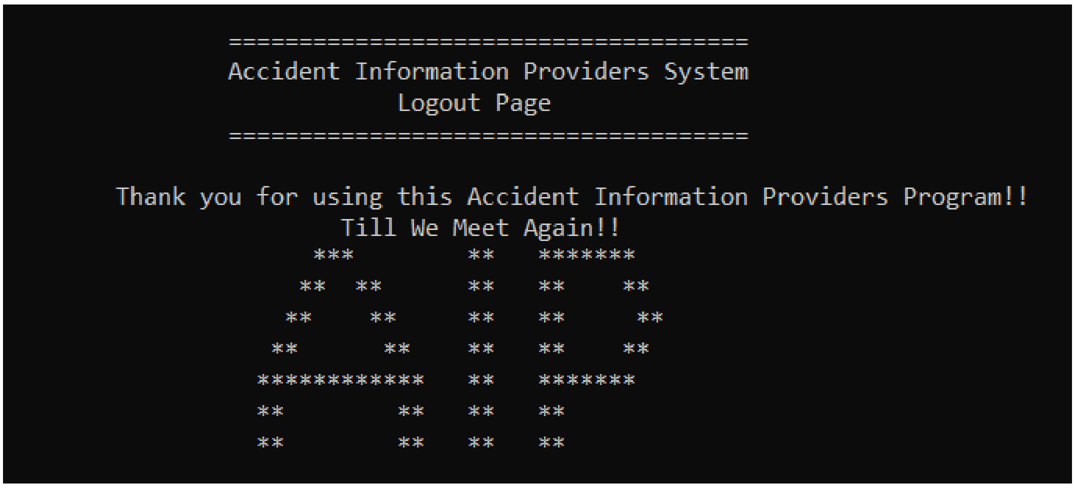


```
===================================
    Accident Information Providers
         Information Police
===================================
Policer id = 10006Pol
Policer Name = Ashraf Danial
Policer Phone Number = +601139893133
Policer Address = Jalan Wawasan, Terengganu
Policer Email = ashrafdanial@gmail.com

Police Station = BP1005
Station Name = Balai Polis Kota Setar
Station City = Kedah
Station Address = Jalan Raja, 05560 Alor Setar, Kedah

1. Want change information
2. Back to menu
Please choose the number = 1

1. Name
2. Phone Number
3. Address
4. Email
5. Station Id
6. Password
Please choose number of information that you want to change =
```

Figure 4.23: Output Information page

[Figure 4.23 is view profile police and can update if want to change information]

4.3.13 Exit or Logout Page



```
        =====================================
        Accident Information Providers System
                    Logout Page
        =====================================

Thank you for using this Accident Information Providers Program!!
                Till We Meet Again!!
              ***        **   *******
             **  **      **   **    **
            **    **     **   **     **
           **      **    **   **    **
          ************   **   *******
          **        **   **   **
          **        **   **   **
```

Figure 4.24: Output exit or logout page

[Figure 4.24 is user want to exit from program]

4.4 System Testing

Test name	Description	Test data	Expected Output	Result
Register	Insert new user data in table users	-name -address -numberPhone -password -re-enter pass -email -policeSationId	Message “Successfully Added Your Data in Database” appear	Success
Login	Select user data from table user based on correct user input	-policeId -password	Message “Successfully logged in as username” appear	Success
Add Accident, Report, Driver, Vehicle	Insert new accident data inside table accident, report, driver, vehicle	-numberOfPeople -streetName -stateName -accidentCost -DriverName -driverIc -vehicleOwner -vehicleNumPlate	Message “Press any key to continue”	Success
View accident	Select all accident, report, vehicle, driver data from table student	none	List of accident, report, vehicle, driver, police details displayed	Success
Search report	Search report for display report details	-reportId -icNumber	List of accident, report, vehicle, driver, police details displayed	Success
Edit Status report	Change status report and update in database	-reportId	List of accident, report, vehicle, driver, police details displayed	Success
Delete id report	Search id report then delete all data that use the same database	-reportId	Message “Successfully delete report id!!”	Success

Statistic Graph	Display number of reports was reported monthly in graph and table	None	Table and graph of reported report.	Success
Update User data	Update user data in table users	-name -phoneNum -address -email -stationId -passWord	Message “Successfully updated user data” appear	Success

CHAPTER 5: CONCLUSION

5.1 Project Summarization

The first chapter describes the introduction, problem statement, objective, scope and significant of study. This chapter is to identify the problem of the previous system and make an objective to build a new system. It also states what are the scope of the system such as the system users and modules. Lastly, it describes the importance of researching about this project.

The second chapter specifies the analysis of the system. This chapter studies thoroughly the problem statement and develops a new concept for the new system. It also investigates all part of the system to ensure all components of the system work flawlessly to accomplish its aim.

The third chapter studies the designing of the system. It describes the entity relationship diagram, data dictionary and interface design. In this chapter, it describes the functionality of the system through flowchart.

The fourth chapter examines the implementation and testing part of the system. This chapter will implement what I have design on the previous chapter. All of the design decisions will be guided by the flowchart that I have created using C++ codes. This chapter achieved the project's objectives, to develop a management system and to store student's data efficiently.

In conclusion, Accident Information Providers System is very useful and effective for the policer to make and manage the report applications and the data stored securely in the databases. It assists the process of searching, editing, deleting and display statistic and graph of report and update information.

5.2 Strength and Weaknesses

5.2.1 Strengths

The strengths of this system are:

i. Easy to find and get report details

For users, they can easily search report into system by logging in and search report and accident details. This will improve the lack of previous system which is by searching in the file room.

ii. Easy to update police information and register new police.

In this system, user have easy way to register police information once they officially being a police officer. They also easily could update their information if they moved to others officer or others information.

iii. Easily to know the numbers of accident in Malaysia

In this system, user can know the number of reports was report for current month at Malaysia. The process of displaying is very easy to see.

5.2.2 Weaknesses

The weaknesses of this system are:

i. Console application

The system is not built for multi-platform it is still under development and lack user interfaces.

ii. Does not have a manual guide

The system does not have a manual guide, it makes some first-time user hard to use.

5.3 Problem Solved

i. Create a platform for police to add a new report.

ii. Create a platform for police search, edit, and delete report.

iii. Replace the old system where it stores data in physical form

iv. Help the police to manage many reports was reported in Malaysia.

5.4 Suggestions for Improvement

The following point is the things that need to be improve in the future. The improvements based on problems or weakness in the system mentioned on previous sub chapter. As a developer, we need an improvement on our system to make it more valuable.

- Mobile Friendly
  - Firstly, a mobile app for this system can be developed. This will help the system to be portable.
- Add a System Guide
  - Secondly, a system guide can help the user to learn how to use the functions in the system. New users will have an easier time when using the system.

5.5 Conclusion

All the modules and functionality in this system are running as planned. That is the main criterion to evaluate whether this project is a success or not. There are strengths and weaknesses in the system. As a developer, fault and limitations is an encouragement to do better in the future. Therefore, we must learn from the mistake to gain more experience.

There are many challenges encountered when completing this project. One of the challenges is the requirement from supervisor and user always changes. The changes of the requirements were hard to make because all the process such as analyzing, implementing codes, and designing databases will be affected. The duration to complete the system is shorter compare to the requirement given that need to be done before the deadline.

During the development phase, time management is the most important part, by managing time properly, the system has been completed ahead of time. In the real industrial environment, punctuality is crucial.

Lastly, the system is easy to use and user-friendly. The flow of the process in the system is running smoothly. Input validations are working as intended. Entered information can easily be modified by the right user.

REFERENCES

1. GeeksForGeeks, (2020) *Geeksforgeeks.* Retrieved 2020, December 2, from https://www.geeksforgeeks.org/
2. w3schools.com, (n.d.) *Sql querries.* Retrieved 2020, November 28, from https://www.w3schools.com/sql/default.asp
3. cplusplus.com, (n.d.) *C++ language library*. Retrieved 2020, November 28, from http://www.cplusplus.com/