

## **Acknowledgment**

Praise and thank to the Almighty for the blessings and guidance to make myself possible to complete my report that can finally be accomplished as it should be. I would like to extent my special thanks to everyone who gave the opportunity to finish this report. The assistance provided by assessment supervisor Mr Danny Hamshananth was appreciable. Encouragement and suggestions, helped me to do the assessment and writing this report. Invested his full effort in guiding me to accomplishing the goal. I am highly indebted to ICBT for their guidance and constant supervision as well as for providing necessary information regarding the report & also for their support in completing the report in campus premises.

## List of contents

### Contents

Programming Methodologies.....	1
1. Procedural programming .....	1
2. Object-oriented Programming .....	2
3. Functional Programming .....	2
4. Logical programming.....	3
Control Structures .....	4
1. Sequential Control Structures .....	4
2. Selection Control Structures .....	4
• IF Statement .....	5
• IF-Else Statement .....	6
• Nested IF-Else Statement .....	6
• IF-Else Ladder/Else-IF Clause .....	7
• Switch Statements or Control Statements .....	8
3. Repetition / Loop Control Structure .....	8
1. indeterminate loops.....	8
2. determinate loops.....	10
Modularization .....	11
Introduction.....	12
Introduction.....	12
Structure of dissertation .....	12
Software Requirement Specification .....	13
Introduction.....	13

Motivation of the project .....	13
Objectives of the system .....	14
Scope of the propose system.....	14
Overall Description .....	14
User Needs .....	15
External Interface Requirements.....	15
Functional Requirements .....	15
Hardware Requirements.....	15
Software Requirements .....	16
Nonfunctional Requirements .....	16
Design of solution .....	17
Introduction.....	17
Process flow diagrams and maps .....	17
Flow chart .....	17
User Interface Design for the System .....	62
• Main menu.....	62
• Login .....	63
• Login success .....	63
• Manage login (user) .....	64
• Manage login (admin) .....	65
• Sign up.....	67
• Manage services & products .....	67
• Available Services (guest).....	71
• Available Services (user and admin).....	71

• Available Products (guest) .....	74
• Available Products (user and admin) .....	74
• Cart (user and admin).....	77
Development .....	78
Introduction.....	78
Development environment.....	78
System Development Tools and Technologies .....	78
DEV C++ .....	78
VS Code.....	78
TDM-GCC Compiler .....	78
C++ V.11 .....	79
Hierarchical Function Structure .....	79
Code Structure .....	80
Main requirements .....	80
Other requirements.....	88
Validation.....	118
Evaluation .....	121
Test plan – IEEE 829 .....	121
Test Plan Identifier.....	121
References.....	121
Introduction.....	121
Test Items.....	121
Software Risk Issues .....	122
Features to be tested.....	122

Features not to be Tested .....	122
Approach.....	122
Item Pass/Fail Criteria.....	123
Suspension Criteria and Resumption Requirements.....	123
Test Deliverables .....	123
Remaining Test Tasks.....	123
Environmental Needs.....	124
Staffing and Training Needs .....	124
Responsibilities .....	124
Schedule.....	124
Planning Risks and Contingencies.....	124
Approvals .....	124
Test case.....	125
1.    Login.....	125
2.    Logout.....	128
3.    Menu.....	129
4.    Signup.....	131
5.    Program exit .....	133
6.    List service.....	133
7.    Search service .....	134
8.    Buy service .....	135
9.    List product.....	136
10.    Search product .....	137
11.    Buy product .....	138

12.	Add service .....	139
13.	Delete service .....	140
14.	Update service .....	141
15.	Add product .....	142
16.	Delete product.....	142
17.	Update product .....	143
18.	Change password.....	144
19.	Delete account .....	145
20.	List user accounts .....	146
21.	Cart .....	147
	Test summery report .....	150
	User Accepting Testing (UAT).....	151
	Sample questionnaire .....	152
	User feedback.....	157
	Test conclusion .....	162
	Future Recommendation.....	162
	References.....	<b>Error! Bookmark not defined.</b>
	Appendix.....	<b>Error! Bookmark not defined.</b>
	questionnaire .....	<b>Error! Bookmark not defined.</b>

## List of figures

Figure 1 Procedural programming .....	1
Figure 2 object orientated programming.....	2
Figure 3 Functional Programming .....	3
Figure 4 Sequential control structure .....	4
Figure 5 1. Selection Control Structures.....	5
Figure 6 IF statement .....	6
Figure 7 IF-Else Statement .....	6
Figure 8 3. Nested IF-Else Statement .....	7
Figure 9 IF-Else Ladder/Else-IF Clause .....	7
Figure 10 Switch Statements or Control Statements .....	8
Figure 11 WHILE loop .....	9
Figure 12 DO-WHILE loop.....	10
Figure 13 FOR loop .....	10
Figure 14 Modularization .....	11
Figure 15 Flow chart Program starts.....	18
Figure 16 Flow chart main.....	18
Figure 17 Flow chart resetCartSystem.....	19
Figure 18 Flow chart menu .....	20
Figure 19 Flow chart Header .....	21
Figure 20 Flow chart guestmenu.....	22
Figure 21 Flow chart guestSelection.....	23
Figure 22 Flow chart usermenu .....	24
Figure 23 Flow chart userSelection .....	25

Figure 24 Flow chart adminmenu .....	26
Figure 25 Flow chart adminSelection .....	27
Figure 26 Flow chart Manage_user_accounts .....	28
Figure 27 Flow chart display_user_accounts.....	29
Figure 28 Flow chart deleteUserML.....	30
Figure 29 Flow chart availableService .....	31
Figure 30 Flow chart Available_Service_menu .....	32
Figure 31 Flow chart Available_Service_Selection .....	33
Figure 32 Flow chart availableProduct.....	34
Figure 33 Flow chart Available_Product_menu .....	35
Figure 34 Flow chart Available_Product_Selection .....	36
Figure 35 Flow chart List_Items.....	37
Figure 36 Flow chart buy_Item.....	38
Figure 37 Flow chart.....	39
Figure 38 Flow chart login.....	40
Figure 39 Flow chart signup .....	41
Figure 40 Flow chart logout.....	42
Figure 41 Flow chart admin_login_menu.....	42
Figure 42 Flow chart Manage_login.....	43
Figure 43 Flow chart admin_login_Selection.....	44
Figure 44 Flow chart user_login_menu .....	45
Figure 45 Flow chart Manage_Services_Proucts_menu.....	45
Figure 46 Flow chart user_login_Selection .....	46
Figure 47 Flow chart changePassword .....	47

Figure 48 Flow chart deleteAccount.....	48
Figure 49 Flow chart Manage_Services_Proucts .....	49
Figure 50 Flow chart Manage_Services_Proucts_Selection.....	50
Figure 51 Flow chart Add_Services .....	51
Figure 52 Flow chart Delete_Services .....	52
Figure 53 Flow chart Update_Services.....	53
Figure 54 Flow chart Add_Products .....	54
Figure 55 Flow chart Delete_Products .....	55
Figure 56 Flow chart Update_Products .....	56
Figure 57 Flow chart cart.....	57
Figure 58 Flow chart find_price_delete_cart.....	58
Figure 59 Flow chart bill.....	59
Figure 60 Flow chart money_total .....	60
Figure 61 Flow chart delete_S_P .....	61
Figure 62 main menu of guest .....	62
Figure 63 main menu of user .....	62
Figure 64 main menu of admin .....	63
Figure 65 login success .....	63
Figure 66 login fail.....	64
Figure 67 manage login of user .....	64
Figure 68 manage login of admin .....	65
Figure 69 change password.....	65
Figure 70 manage user accounts of admin.....	66
Figure 71 delete account .....	66

Figure 72 sign up .....	67
Figure 73 manage services & products.....	67
Figure 74 add services .....	68
Figure 75 delete services.....	68
Figure 76 update services.....	69
Figure 77 add products.....	69
Figure 78 delete products .....	70
Figure 79 update products.....	70
Figure 80 available services of guest .....	71
Figure 81 available services of user and admin .....	71
Figure 82 list services .....	72
Figure 83 search services .....	73
Figure 84 buy service of user and admin .....	73
Figure 85 available products of guest .....	74
Figure 86 available Products of user and admin .....	74
Figure 87 list products .....	75
Figure 88 search products .....	76
Figure 89 buy product of user and admin .....	76
Figure 90 cart of user and admin .....	77
Figure 91 Basic hierarchical Function Structure .....	79
Figure 92 add products.....	80
Figure 93 delete products.....	81
Figure 94 update products.....	82
Figure 95 add services .....	83

Figure 96 delete services.....	84
Figure 97 update services.....	85
Figure 98 list available services and products .....	86
Figure 99 search specific product or service.....	87
Figure 100 user login .....	88
Figure 101 logout.....	89
Figure 102 exit .....	89
Figure 103 headers.....	90
Figure 104 function prototyping .....	92
Figure 105 global Variable .....	92
Figure 106 main .....	92
Figure 107 reset cart system .....	93
Figure 108 header .....	93
Figure 109 money total .....	94
Figure 110 menu .....	95
Figure 111 guest menu.....	96
Figure 112 guest selection .....	96
Figure 113 available service .....	97
Figure 114 available service menu .....	97
Figure 115 available service selection .....	98
Figure 116 buy item .....	100
Figure 117 available product .....	100
Figure 118 available product menu.....	101
Figure 119 available product selection .....	101

Figure 120 admin menu .....	102
Figure 121 admin selection.....	102
Figure 122 manage services products .....	103
Figure 123 manage services products menu .....	104
Figure 124 manage services products selection.....	104
Figure 125 core function of delete service and product.....	105
Figure 126 user menu.....	106
Figure 127 user selection .....	106
Figure 128 sign up .....	107
Figure 129 manage login .....	108
Figure 130 admin login menu .....	108
Figure 131 admin login selection.....	109
Figure 132 user login menu .....	110
Figure 133 user login selection .....	110
Figure 134 change password.....	111
Figure 135 delete account .....	112
Figure 136 delete user in master login.....	113
Figure 137 manage user accounts .....	114
Figure 138 display user accounts.....	115
Figure 139 cart .....	116
Figure 140 bill.....	117
Figure 141 find the price of item in the cart to delete.....	117
Figure 142 UAT sample questionnaire page 1 .....	152
Figure 143 UAT sample questionnaire page 2 .....	153

Figure 144 UAT sample questionnaire page 3 .....	154
Figure 145 UAT sample questionnaire page 4 .....	155
Figure 146 UAT sample questionnaire page 5 .....	156
Figure 147 UAT user feedback page 1 .....	157
Figure 148 UAT user feedback page 2 .....	158
Figure 149 UAT user feedback page 3 .....	159
Figure 150 UAT user feedback page 4 .....	160
Figure 151 UAT user feedback page 5 .....	161

## List of tables

Table 1 Implantation environment.....	78
Table 2 Remaining test task.....	124
Table 3 Test case Login .....	125
Table 4 Test case login - admin results.....	126
Table 5 Test case login - user results .....	127
Table 6 Test case logout .....	128
Table 7 Test case logout - admin results.....	128
Table 8 Test case logout - user results .....	128
Table 9 Test case menu.....	129
Table 10 Test case menu - admin .....	129
Table 11 Test case menu - user.....	130
Table 12 Test case menu - guest .....	130
Table 13 Test case signup .....	131
Table 14 Test case signup - admin.....	132
Table 15 Test case signup - user .....	132
Table 16 Test case program exit .....	133
Table 17 Test case program exit result .....	133
Table 18 Test case list service .....	133
Table 19 Test case list view result .....	134
Table 20 Test case search service .....	134
Table 21 Test case search service results.....	135
Table 22 Test case buy service .....	135
Table 23 Test case buy service result .....	136

Table 24 Test case list product.....	136
Table 25 Test case list product result.....	137
Table 26 Test case search product .....	137
Table 27 Test case search product results.....	138
Table 28 Test case buy product .....	138
Table 29 Test case buy product result.....	139
Table 30 Test case add service.....	139
Table 31 Test case add service result.....	139
Table 32 Test case Delete service.....	140
Table 33 Test case Delete service results .....	140
Table 34 Test case Update service.....	141
Table 35 Test case Update service result.....	141
Table 36 Test case add product.....	142
Table 37 Test case add product.....	142
Table 38 Test case delete product.....	142
Table 39 Test case delete product.....	143
Table 40 Test case update product.....	143
Table 41 Test case update product results .....	144
Table 42 Test case change password .....	144
Table 43 Test case change password result .....	145
Table 44 Test case delete account.....	145
Table 45 Test case delete account results .....	145
Table 46 Test case list user accounts .....	146
Table 47 Test case list user accounts results.....	146

Table 48 Test case cart.....	147
Table 49 Test case cart results .....	149
Table 50 Test summery report .....	150

## **Programming Methodologies**

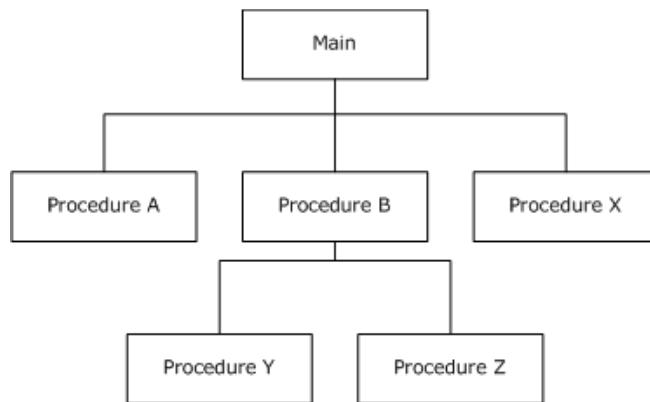
When systems are designed to resolve challenges in everyday life, they appear to be enormous and complicated. The approach to analysis, preparation and management of the production phase of such complex problems is the programming methodology. There are 3 types in programming methodologies. Those are:

1. Procedural Programming
2. Object-oriented Programming
3. Functional Programming
4. Logical Programming

(Programming Methodologies - Introduction - Tutorialspoint, 2021)

### **1. Procedural programming**

Procedural programming can be described as a programming model based on the principle of calling procedure derived from structured programming. Procedures, also referred to as routines, substrates or functions, are essentially a sequence of calculations to be performed. During the execution of a program, any other operation, or other processes or itself, may be called at any time.



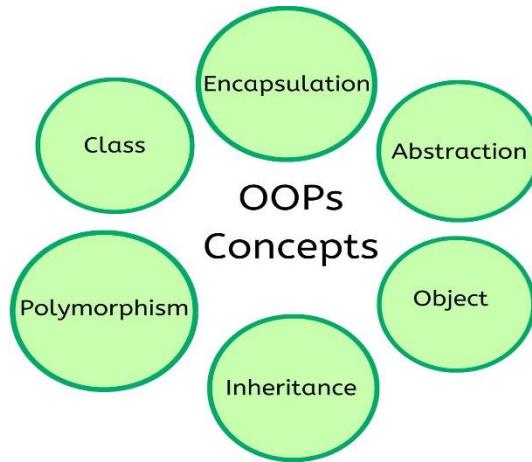
*Figure 1 Procedural programming*

Example: For developing calculator program, addition, subtraction, multiplication, divisions are will develop as separate procedures and call it into main module when user needs.

(Differences between Procedural and Object-Oriented Programming - GeeksforGeeks, 2021)

## 2. Object-oriented Programming

Object-oriented programming (OOP) is a programming framework based on the class and the object principle. It is used to organize a software program in plain, reusable code sheets that serve to construct individual instances of objects (usually known as classes).



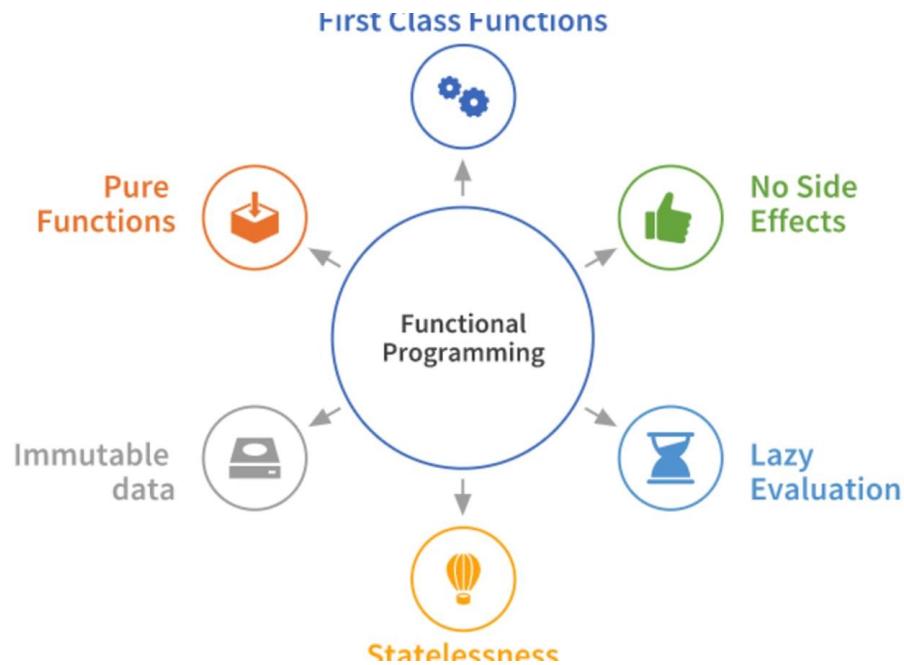
*Figure 2 object orientated programming*

Example: We built a car class to include all the features that a car needs, colour, brand, and model. Then we build an instance of an object class vehicle, myCar, to display my particular car. The value of the properties described in the class may be set to represent my vehicle, without affecting the class template or other objects. This class can be reused to display any number of vehicles.

(What is Object Oriented Programming? OOP Explained in Depth, 2021)

## 3. Functional Programming

The method of software building, by pure functioning, avoids share status, mutable data and side-effects, is functional programming (often abbreviated FP). The declarative rather than the obligatory functional programming, and pure functionality is the device state. Contrast to object-oriented programming, which normally shares the application state and place it in an object using methods.



*Figure 3 Functional Programming*

Examples: Functional units such as employee data maintenance, simple payroll calculations, gross compensation estimate, leave processing and loan refund processing may provide a payroll processing.

(Master the JavaScript Interview: What is Functional Programming?, 2021)

#### 4. Logical programming

Logical programming is a theory of programming focused primarily on formal logic. Every program that is written in a language of logical programming is a logical collection of phrases representing facts and rules regarding a problem field.

Example: Users have very defined functions in a school management system including instructor, professor, laboratory assistant, supervisor, academic officer, etc. Thus, according to functions of consumer the software can be split in units. Each user can have various interfaces, permissions, etc.

(Logic programming - Wikipedia, 2021)

# Control Structures

Control structures take program flow decisions based on parameters defined. It must be defined prior to coding, which are very essential elements of any program. Analysts and programmers use algorithms and pseudocodes to define the control structures that are required.

There are the three kinds of control systems.

1. Sequential Structures
2. Selection Structures
3. Repetition / Loop Control Structures

(Control Structures, 2021)

## 1. Sequential Control Structures

"Sequence control structure" means the line-by-line execution of statements in the same order in which they are executed in the program. For example, they may perform a number of read or write operations, arithmetic procedures or variable assignments.

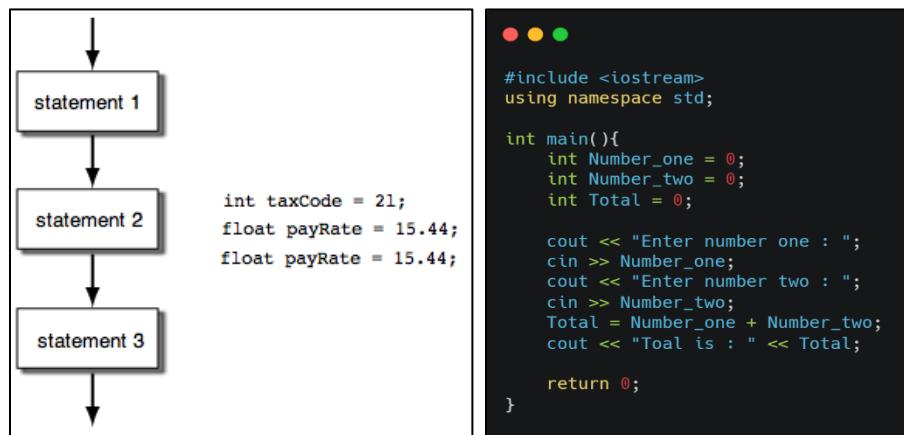


Figure 4 Sequential control structure

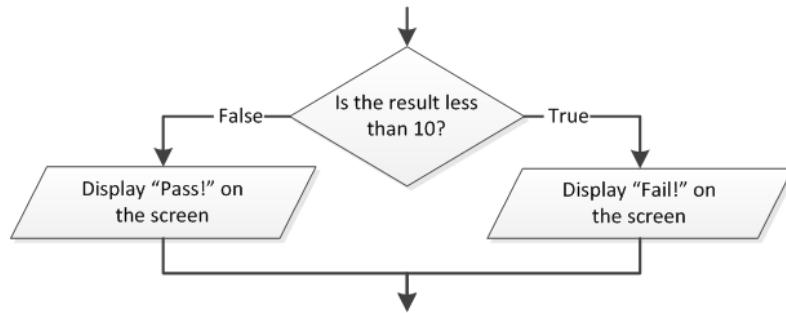
(What is the Sequence Control Structure?, 2021)

## 2. Selection Control Structures

In order to control sequences, statements in the same order in which they appear in the program are performed sequentially. Although you hardly want to execute the declarations successively in serious programming. You also want to execute a block of

statements in one situation and a completely opposite block of statements in another situation. A decision-making structure should take steps in this respect!

The decision Control structure considers a Boolean term or a series of Boolean phrases and then determines which block of statements to implement.



*Figure 5 1. Selection Control Structures*

(What is a Decision Control Structure?, 2021)

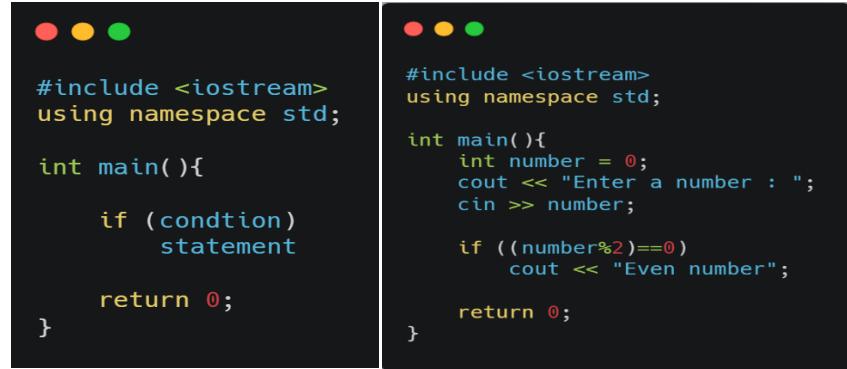
There are 5 kinds in Selection Control Structure

1. IF Statement
2. IF-Else Statement
3. Nested IF-Else Statement
4. IF-Else Ladder/Else-IF Clause
5. Switch Statements or Control Statements

(The Decision Control Structure - C Programming, 2021)

- **IF Statement**

IF Statement checks a given condition and if returns a TRUE value then code block of true will executed



```

# include <iostream>
using namespace std;

int main(){
    if (condtion)
        statement
    return 0;
}

#include <iostream>
using namespace std;

int main(){
    int number = 0;
    cout << "Enter a number : ";
    cin >> number;

    if ((number%2)==0)
        cout << "Even number";
    return 0;
}

```

Figure 6 IF statement

- **IF-Else Statement**

IF-Else Statement checks a given condition and if returns a TRUE value then code block of true will executed else code block of false will executed



```

#include <iostream>
using namespace std;

int main(){
    if (condtion)
        statement
    else
        statement
    return 0;
}

#include <iostream>
using namespace std;

int main(){
    int age = 0;
    cout << "Enter your age : ";
    cin >> age;
    if (age<18)
        cout << "You are child";
    else
        cout << "You are adult";
    return 0;
}

```

Figure 7 IF-Else Statement

- **Nested IF-Else Statement**

When a IF-Else statement contain one or more IF-Else statement/s inside in it then that's known as Nested IF-Else

```

● ● ●
#include <iostream>
using namespace std;

int main(){
    if (condition){
        statement
        if(condition)
            statement
        else
            statement
    }else{
        statement
        if(condition)
            statement
        else
            statement
    }
    return 0;
}

● ● ●
#include <iostream>
using namespace std;

int main(){
    int Number = 0;
    cout << "Enter a number : ";
    cin >> Number;
    if (Number<10){
        cout << "Your number is below 10";
        if((Number%2)==0)
            cout << " and even number";
        else
            cout << " and odd number";
    }else{
        cout << "Your number is above 10";
        if((Number%2)==0)
            cout << " and even number";
        else
            cout << " and odd number";
    }
    return 0;
}

```

Figure 8 3. Nested IF-Else Statement

- **IF-Else Ladder/Else-IF Clause**

IF-Else Ladder/Else-IF Clause is the other approach is the most common way to make a multi-way decision.

```

● ● ●
#include <iostream>
using namespace std;

int main(){
    if (condition)
        statement
    else if (condition)
        statement
    else if (condition)
        statement
    else
        statement

    return 0;
}

● ● ●
#include <iostream>
using namespace std;

int main(){
    int Number = 0;
    cout << "Enter a number : ";
    cin >> Number;
    if (Number<10)
        cout << "Your number is below 10";
    else if (Number<20)
        cout << "Your number is between 10-20";
    else if (Number<30)
        cout << "Your number is between 20-30";
    else
        cout << "Your number is above 30";

    return 0;
}

```

Figure 9 IF-Else Ladder/Else-IF Clause

- **Switch Statements or Control Statements**

The switch statement is a multi-way judgment that checks whether an expression fits one of a certain number of constant integer values.

The image shows two side-by-side screenshots of a terminal window. Both screenshots have three colored dots (red, yellow, green) at the top. The left screenshot contains the following C++ code:

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

int main(){
    switch(expression){
        case constant-expression:
            statement;
            break;
        case constant-expression:
            statement;
            break;
        case constant-expression:
            statement;
            break;
        case constant-expression:
            statement;
            break;
        default:
            statement;
    }
    return 0;
}
```

The right screenshot contains a similar C++ code snippet:

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

int main(){
    int Number = 0;
    cout << "Enter a number between 1-4 : ";
    cin >> Number;
    switch(Number){
        case 1:
            cout << "Number 1 called ";
            break;
        case 2:
            cout << "Number 2 called ";
            break;
        case 3:
            cout << "Number 3 called ";
            break;
        case 4:
            cout << "Number 4 called ";
            break;
        default:
            cout << "Error";
    }
    return 0;
}
```

Figure 10 Switch Statements or Control Statements

### 3. Repetition / Loop Control Structure

Repetitive control systems are groupings of codes, also known as iterative structures, structured to repeated a number of connected statements. This repeat may be repeated zero or more times before any control value or condition stops the repetition. Generally, there are two loops:

1. indeterminate loops
2. determinate loops

(Intro to Repetition Structures, 2021)

#### 1. indeterminate loops

A loop structure is indeterminate as long as a condition assesses a certain boolean value. Unlike the programmer knows precisely how many time iterations are needed, indeterminate loops can be used.

here are two types of indeterminate loops:

1. pre-test loops

## 2. post-test loops

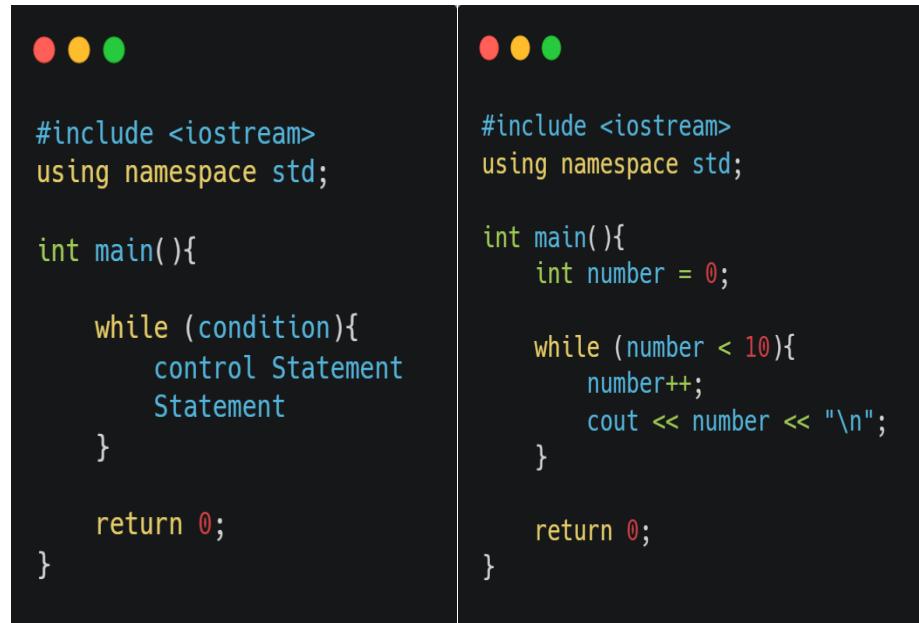
(Intro to Repetition Structures, 2021)

### 1. Pre-test loops

In pre-test loops, the loop body will execute after the loop condition is checked. Thus, for the probability for the body execution can be zero. Best example for this is WHILE loop

#### WHILE loop

The while loop is running through a code block, as long as a certain condition is valid.



The image shows a code editor with two side-by-side snippets of C++ code. Both snippets begin with '#include <iostream>' and 'using namespace std;'. The left snippet shows a general template for a while loop:

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

int main(){
    while (condition){
        control Statement
        Statement
    }
    return 0;
}
```

The right snippet shows a specific example of a while loop that prints numbers from 0 to 9:

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

int main(){
    int number = 0;

    while (number < 10){
        number++;
        cout << number << "\n";
    }

    return 0;
}
```

Figure 11 WHILE loop

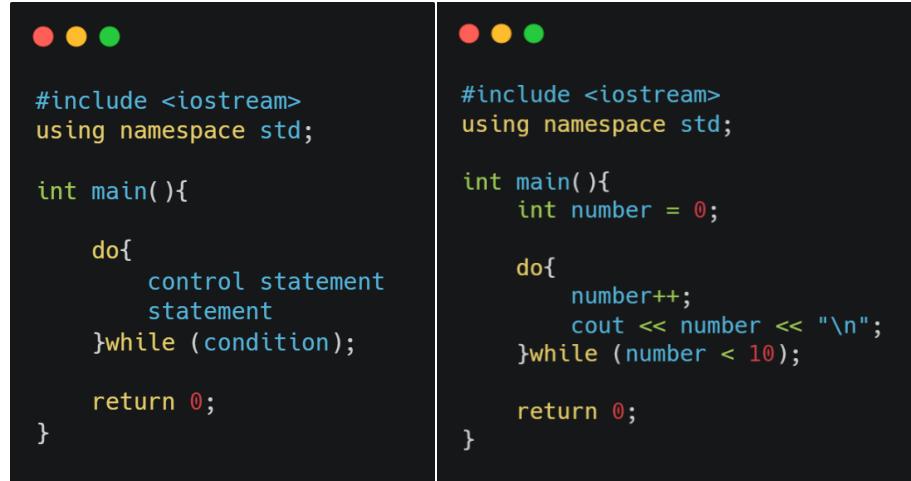
(JavaScript while Loop, 2021)

## 2. post-test loops

In post-test loops, the loop body execute first and then loop condition will be checked. Thus, the probability of loop body execution is minimum one. Best example for this is DO-WHILE loop

#### DO-WHILE loop

The loop is identical to the while loop with a significant exception. At least once is the body of the do-while loop performed.



```

● ● ●

#include <iostream>
using namespace std;

int main(){
    do{
        control statement
        statement
    }while (condition);

    return 0;
}

● ● ●

#include <iostream>
using namespace std;

int main(){
    int number = 0;

    do{
        number++;
        cout << number << "\n";
    }while (number < 10);

    return 0;
}

```

Figure 12 DO-WHILE loop

## 2. determinate loops

In determinate loops can be predict that how many times its going to loop through it. Best example for determinate loop is FOR loop

### FOR loop

There are two part of a for-loop: one header with the iteration and one body per iteration that is executed once. The header always declares an explicit loop variable, which enables the body to know which is performed. it generally used when before joining the loop the number of iterations is specified.

(For loop - Wikipedia, 2021)



```

● ● ●

#include <iostream>
using namespace std;

int main(){
    for(initialization ; condition; iteration){
        body
    }

    return 0;
}

● ● ●

#include <iostream>
using namespace std;

int main(){
    for(int i = 1; i <= 10; i++){
        cout << i << "\n";
    }

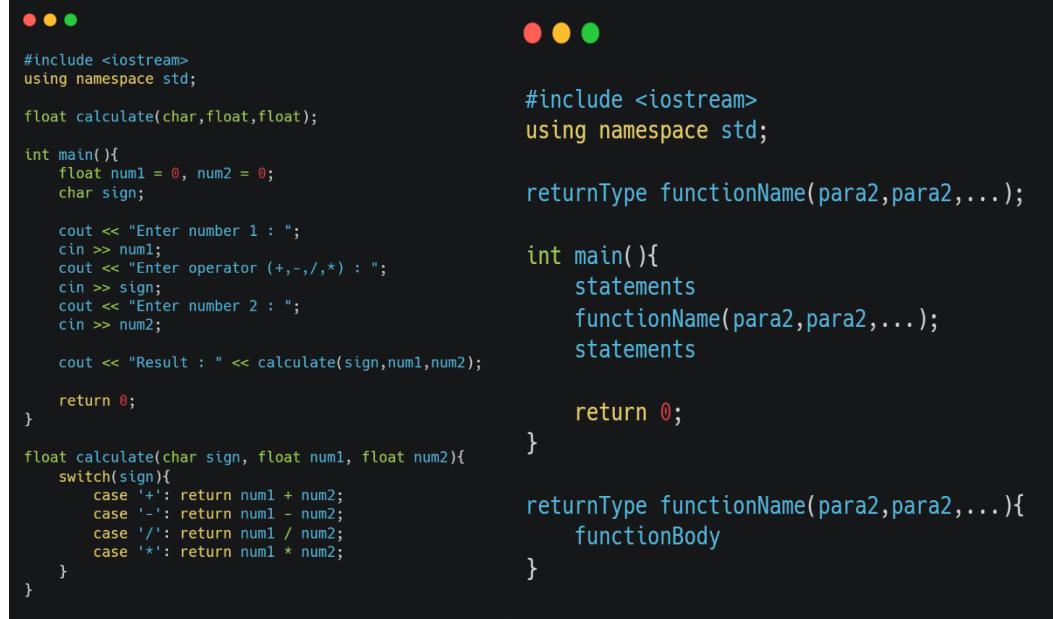
    return 0;
}

```

Figure 13 FOR loop

## Modularization

Modular programming is a software design process which emphasizes the division of the functionality of a program into interchangeable independent modules in such a way that each contains everything necessary to perform just one aspect.



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

float calculate(char, float, float);

int main(){
    float num1 = 0, num2 = 0;
    char sign;

    cout << "Enter number 1 : ";
    cin >> num1;
    cout << "Enter operator (+,-,/,*) : ";
    cin >> sign;
    cout << "Enter number 2 : ";
    cin >> num2;

    cout << "Result : " << calculate(sign,num1,num2);

    return 0;
}

float calculate(char sign, float num1, float num2){
    switch(sign){
        case '+': return num1 + num2;
        case '-': return num1 - num2;
        case '/': return num1 / num2;
        case '*': return num1 * num2;
    }
}
```

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

returnType functionName(para2,para2,...);

int main(){
    statements
    functionName(para2,para2,...);
    statements
}

return 0;
}

returnType functionName(para2,para2,...){
    functionBody
}
```

Figure 14 Modularization

(Modular programming - Wikipedia, 2021)

# **Introduction**

## **Introduction**

In our current mostly every human is being related with vehicles, that can be a manufacturing, distributing, using and repairing. In Sri Lanka, Best Auto Mart is a company which located in the Colombo city. in nowadays it's using manual system for manage all works such as accounting process, managing process, customer ordering process and so on. Since it be a manual system, it facing lots of struggles such as billing process when customer traffic is high, manual counting for inventory items is not accurate, difficulties in listing available service and products to the customer and so on. So now Best Auto Mart company is decided to computerized all the company works

## **Structure of dissertation**

This report is detailly describing about all the process that made for computerized best auto mart company works under five topics. Those are software requirement specification, design of solution, development, evaluation and conclusion

### **Software requirement specification**

In this section, it has 3 subroutines, firstly introduction. In this part introduction of the Best Auto Mart Company, motivation of the project, objectives of the project and scope of the proposing system are detailly explained. Second part is overall description. In this part overall description of proposing system and user roles and their needs are detailly explained. Finally in third part external interface requirements are detailly explained under the category of functional requirements, hardware requirements, software requirements and non-functional requirements.

### **Design of solution**

In this section firstly introduction of designing is given and after process flow diagrams and maps is explained. Then entire system's flow chart is attached. Finally entire system's user interface design is attached and explained.

## Development

In this section firstly introduction of development is given and after development environment and system development tools and technologies used for this project is clearly mentioned. After that, hierarchical function and code structure are attached. Finally, validation process of system is clearly mentioned.

## Evaluation

In this section test plan is explained clearly. After test cases are attached with screenshots of actual result. Ten test summary report is given. Finally, user accepting testing is clearly explained with UAT questionnaire.

## Conclusion

In this section firstly introduction of conclusion of this project is explained. And after critical assessment of the project and future recommendations are clearly discussed.

# **Software Requirement Specification**

A software specifications specification (SRS) is a concise overview of the functional and non-functional requirements of a software system to be built. The SRS is focused on the customer-contractor arrangement. It could include use cases for how the user will interact with the software system.

(Jain, Jain, Testing and Soma, 2021)

## **Introduction**

### **Motivation of the project**

Recently the Best Auto Mart (BAT) commenced an analysis regarding to company efficiency status. In now days BAT receiving high customer traffic which leads to many difficulties

- Difficulties in view available products and services
- Difficulties in Billing process
- Difficulties in inventory management

Therefore, the need of computerized system for BAT company is evolved to reduce all difficulties.

## **Objectives of the system**

Main objectives of the computerized BAT system are:

- View all the available service and products within short time
- Create invoices automatically
- Search and find the services and products in short time

## **Scope of the propose system**

Scope of the computerized BAT system is, reducing workloads, work's time complexity and able to manage all the customers easily.

## **Overall Description**

Computerized BAT system software is having 3 user roles and those are in different permission levels according to their work descriptions.

Fist user roles is guest, in this type of user can view available products and services and also can search services and products. Guest user can create a new account in the system via signup process and able to become second level user

Second user role is user (Customer), in this type of user can view available products and services and also can search services and products. Additionally, they can add services and products to the cart for buying process. Also, they can remove services and products from cart. This type user can change their account password and also able to delete their account permanently from the system.

Third user role is admin, in this type of user can view available products and services and also can search services and products. Admin user can add or remove or update the both services and products in the system. They can add services and products to the cart for buying process. Also, they can remove services and products from cart. This type user can change their account password and also able to delete their account permanently from the system. Finally, admins can view all user role's username and password and able to delete the account or change the password of all accounts.

## **User Needs**

User is need to have a knowledge in modern technology devices and handling methods. Also, user need to have a fully knowledge in handling BAT computerized system software for their works.

## **External Interface Requirements**

This section shall describe the interface requirements for the Best Auto Mart system. They specify the way the user shall interact with the system as well as define the necessary hardware interfaces and communication interfaces required by the software to store and retrieve data.

## **Functional Requirements**

Main requirements

- Manage available services and products (add, delete, update)
- List available services and products.
- Search specific product or service.

Other requirements are

- User login.
- Logout.
- Exit

## **Hardware Requirements**

1. Computer
  - a. 512MB RAM
  - b. Dual core processor
  - c. Monitor
  - d. Keyboard
  - e. 10gb HDD

2. Printer

### **Software Requirements**

1. Windows 7 or higher
2. Printer driver

### **Nonfunctional Requirements**

1. User friendly interface

BAT system's interface has navigation bar which help users to understand the software flow. And also, it has total price of purchased items is show on right top corner

2. Security

all asking prompts are validated to prompt needs

## **Design of solution**

### **Introduction**

Systems design is the process to define systems elements for a system based on the defined specifications such as the modules, architecture, components and their interfaces and data. It is the process of identifying, creating and designing systems that meet a company or organization's particular needs and requirements.

### **Process flow diagrams and maps**

Process flow diagrams (PFD) are a graphical method for the description and sequence of a process. A PFD supports method design brainstorming and collaboration. A full list of tasks including the process under examination is required for the PFMEA process. The team will determine the level of detail. It takes time to provide more information, but it decreases the chance of failure modes being missing.

(Process Flow Diagram - an overview | ScienceDirect Topics, 2021)

### **Flow chart**

A flow chart is a sequence-linked image of the individual process phases. It is a common tool that can be used to define many processes for a wide range of purposes. It is one of the seven fundamental tools of process analysis.

(What is a Flowchart? Process Flow Diagrams & Maps | ASQ, 2021)

All flow chart is deigned in VISIO 2019 software. You can clearly view all the flow chart designs in below link.

<https://drive.google.com/file/d/1gA-W3pXHZG31t5hn2VI7QeE6JngTnvtK/view>

1. program starts

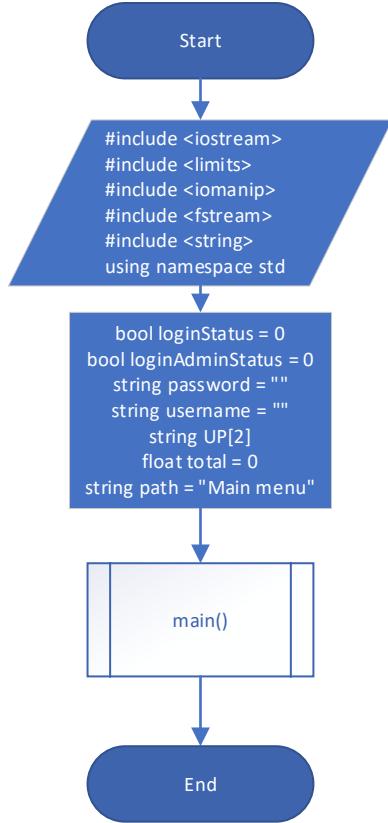


Figure 15 Flow chart Program starts

2. main

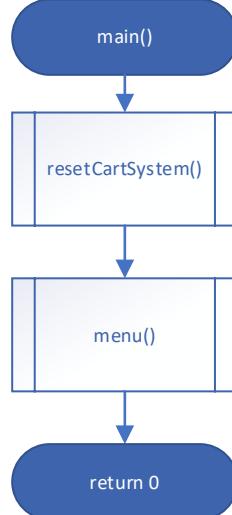


Figure 16 Flow chart main

### 3. resetCartSystem

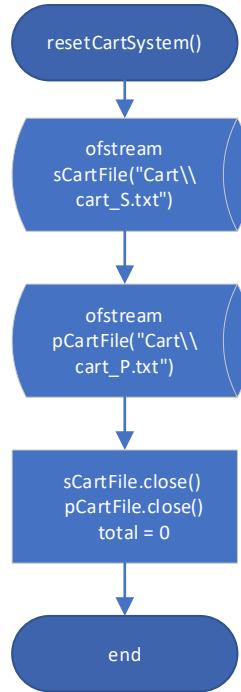


Figure 17 Flow chart `resetCartSystem`

#### 4. menu

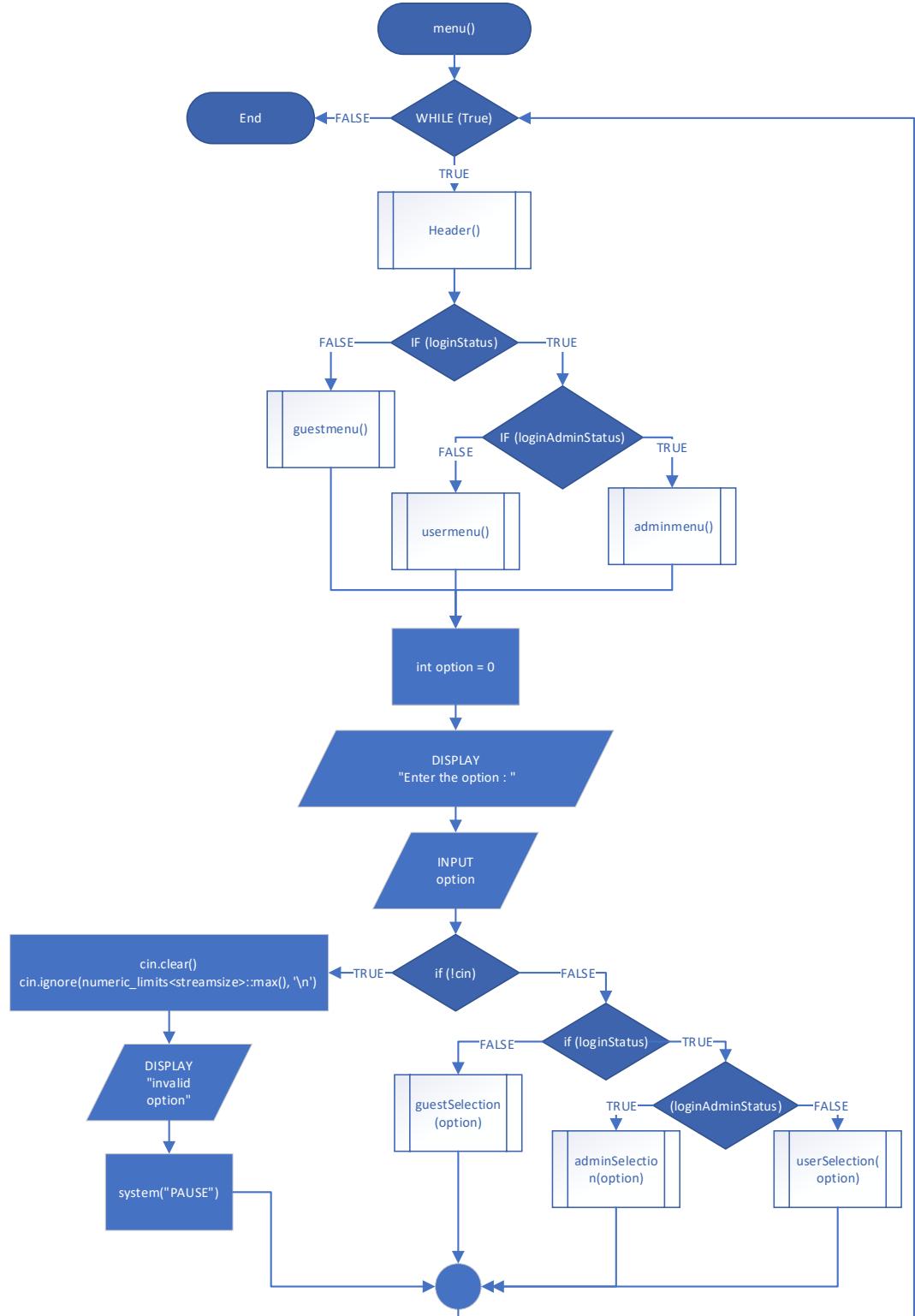


Figure 18 Flow chart menu

## 5. Header

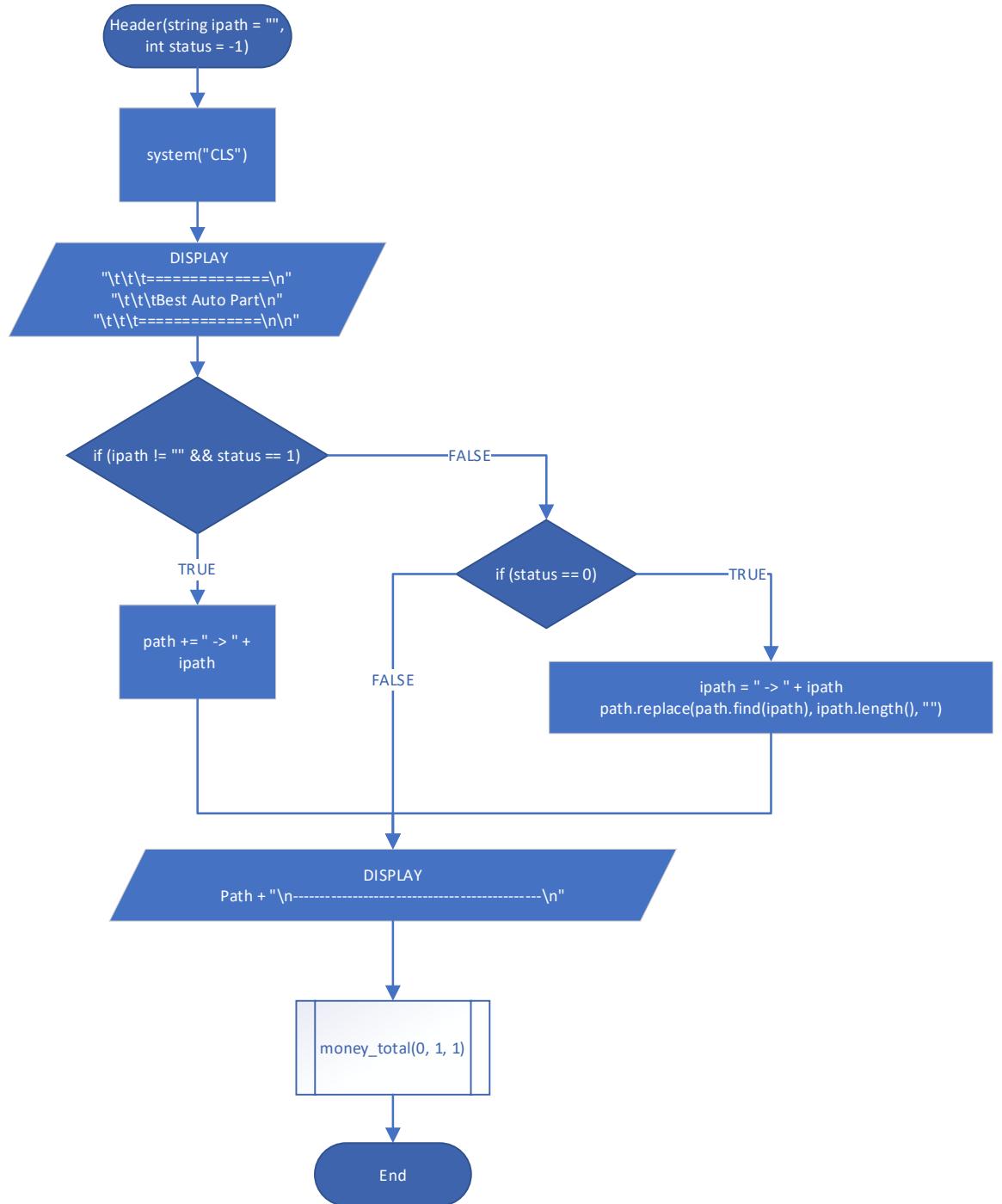


Figure 19 Flow chart Header

## 6. guestmenu

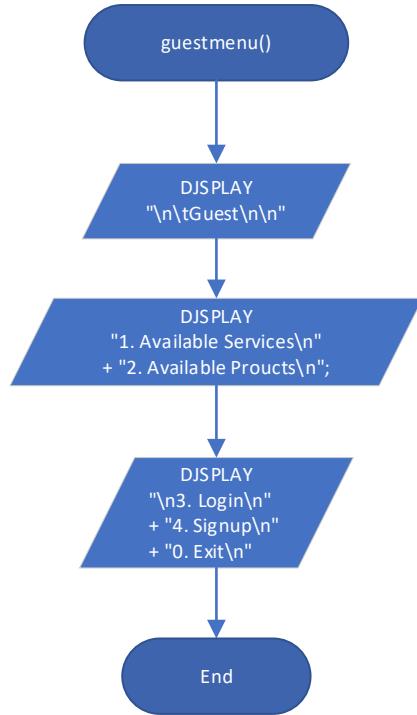


Figure 20 Flow chart guestmenu

## 7. guestSelection

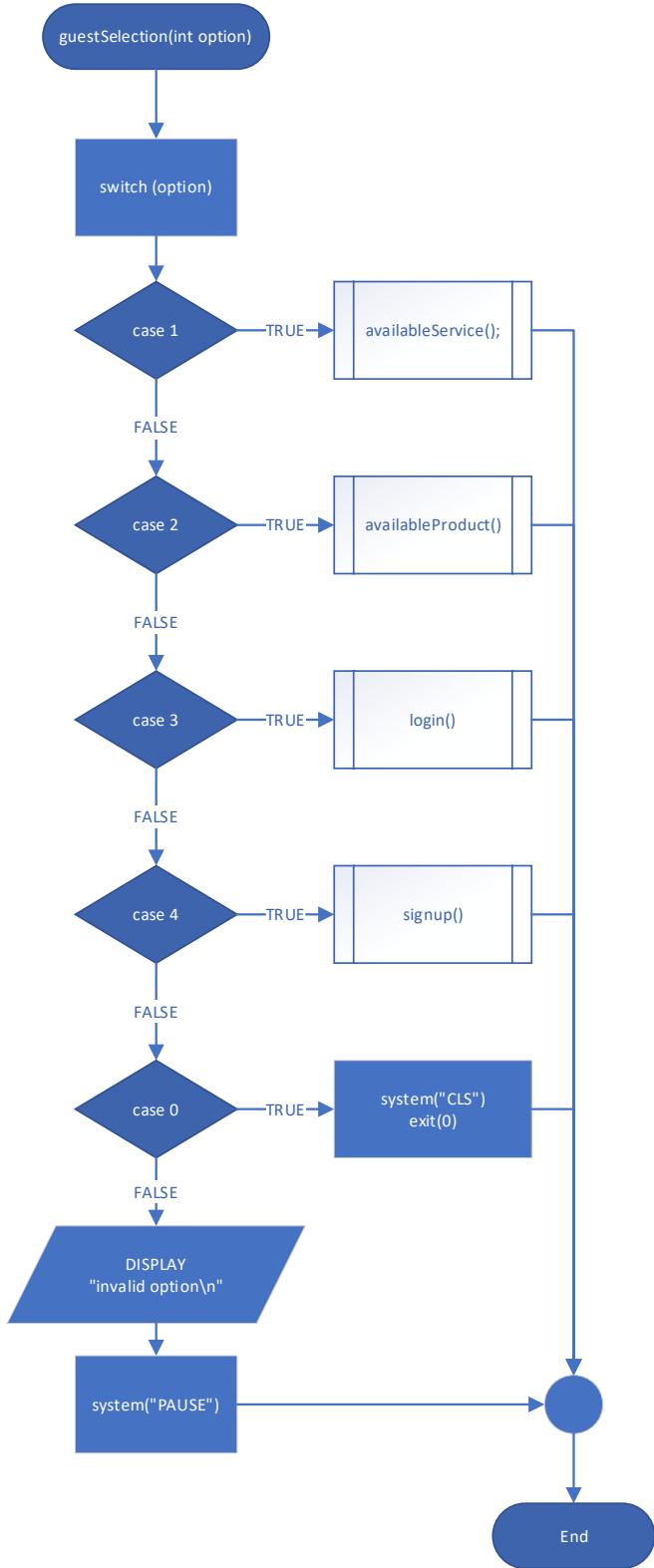


Figure 21 Flow chart guestSelection

## 8. usermenu

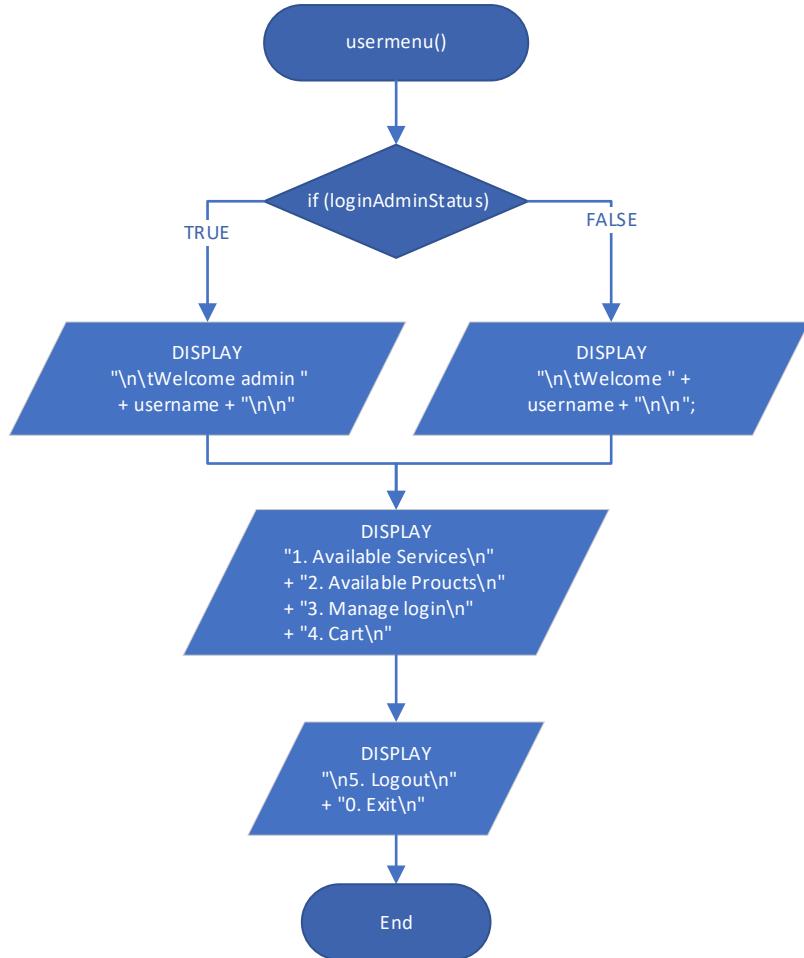


Figure 22 Flow chart usermenu

## 9. userSelection

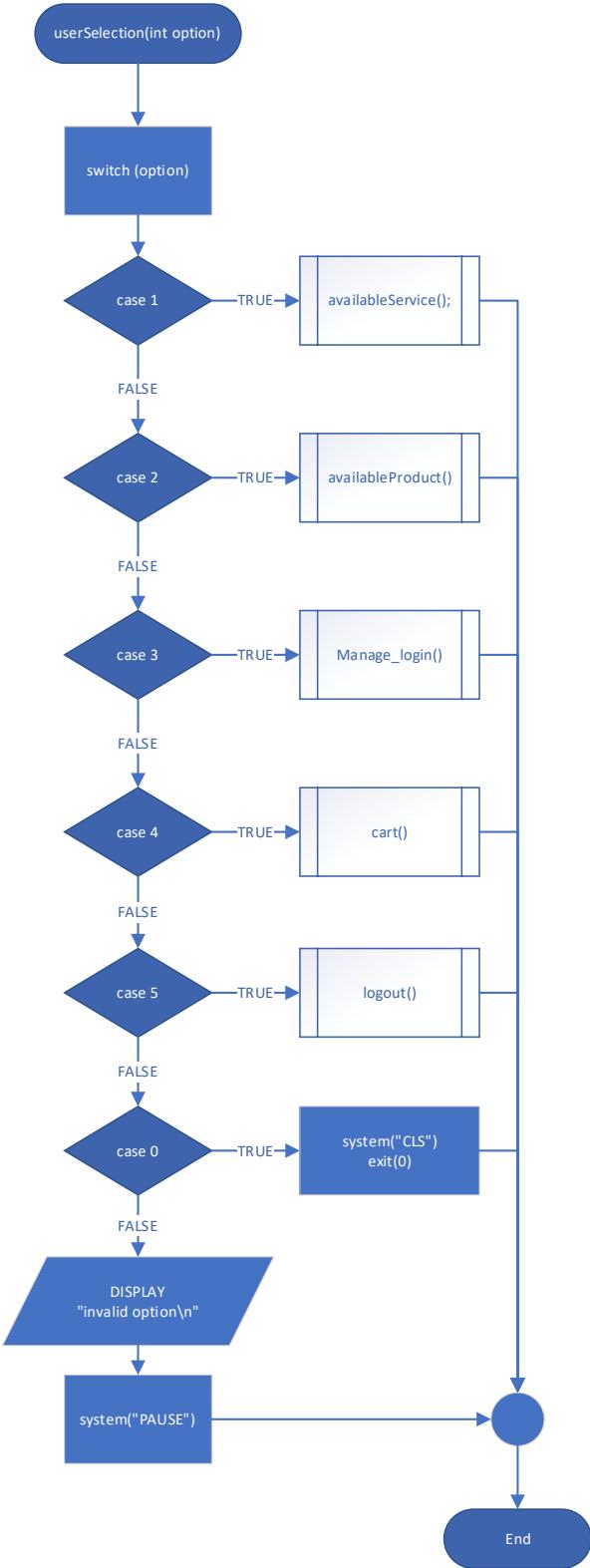


Figure 23 Flow chart userSelection

## 10. adminmenu

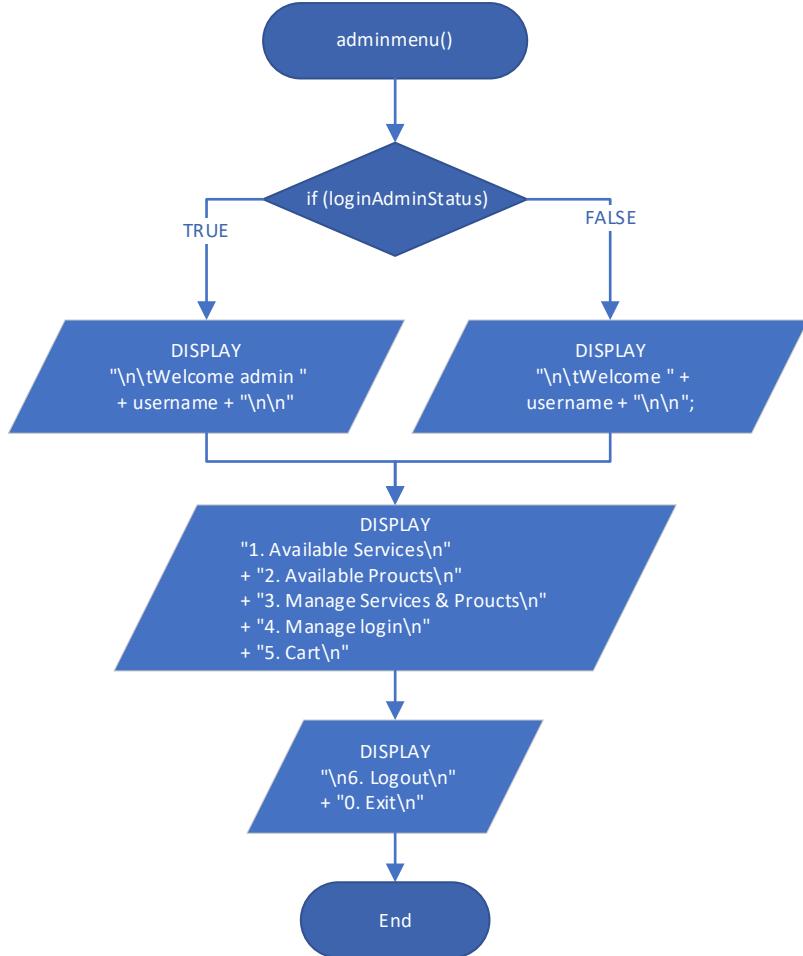


Figure 24 Flow chart adminmenu

## 11. adminSelection

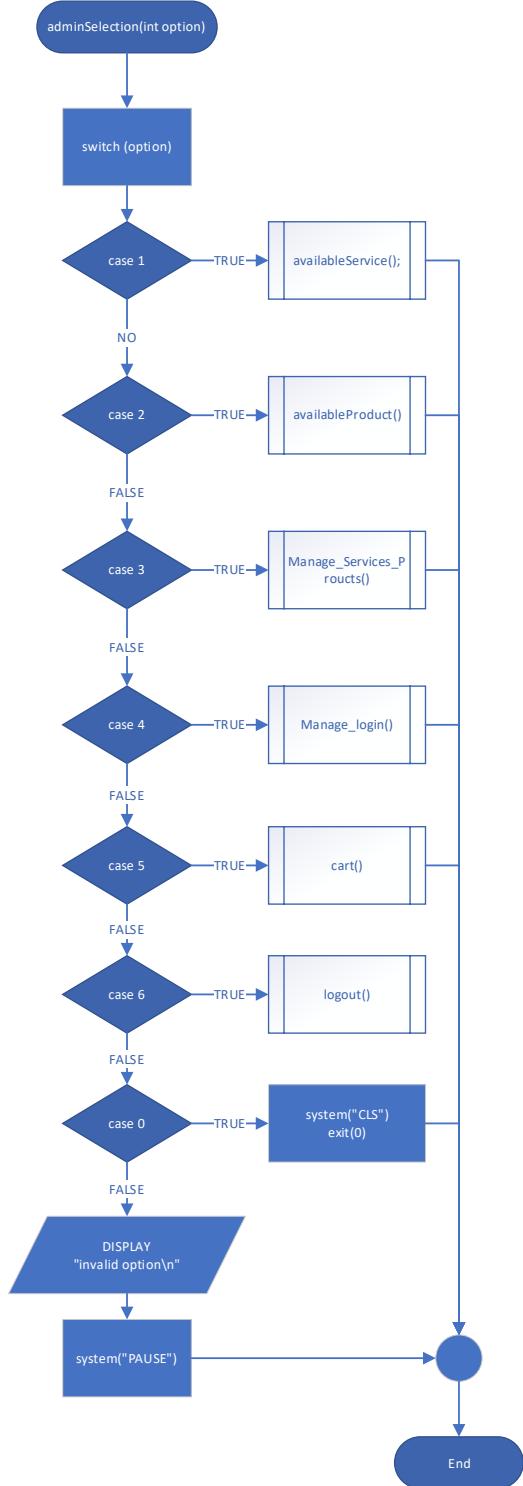


Figure 25 Flow chart adminSelection

## 12. Manage\_user\_accounts

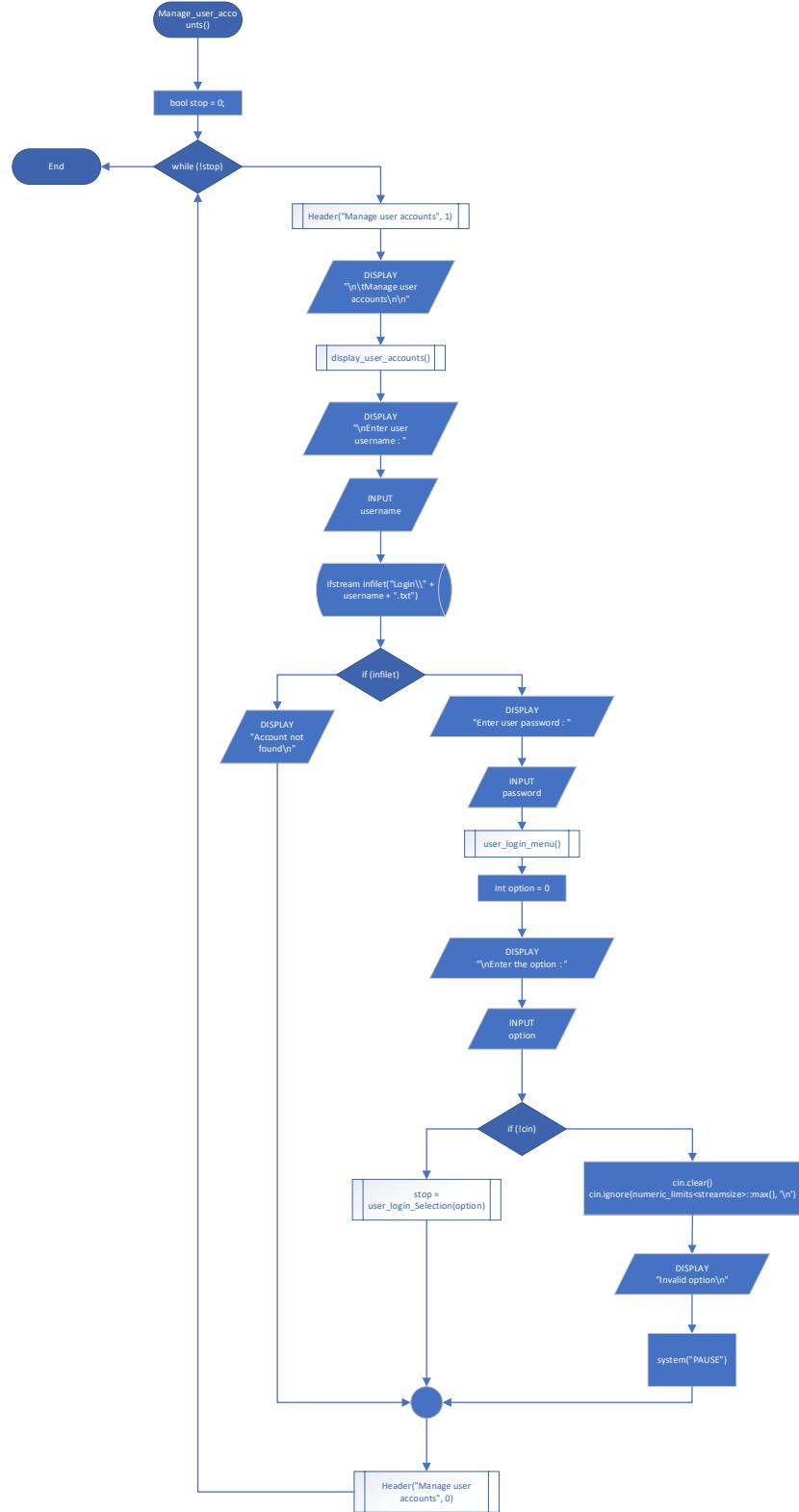


Figure 26 Flow chart Manage\_user\_accounts

### 13. display\_user\_accounts

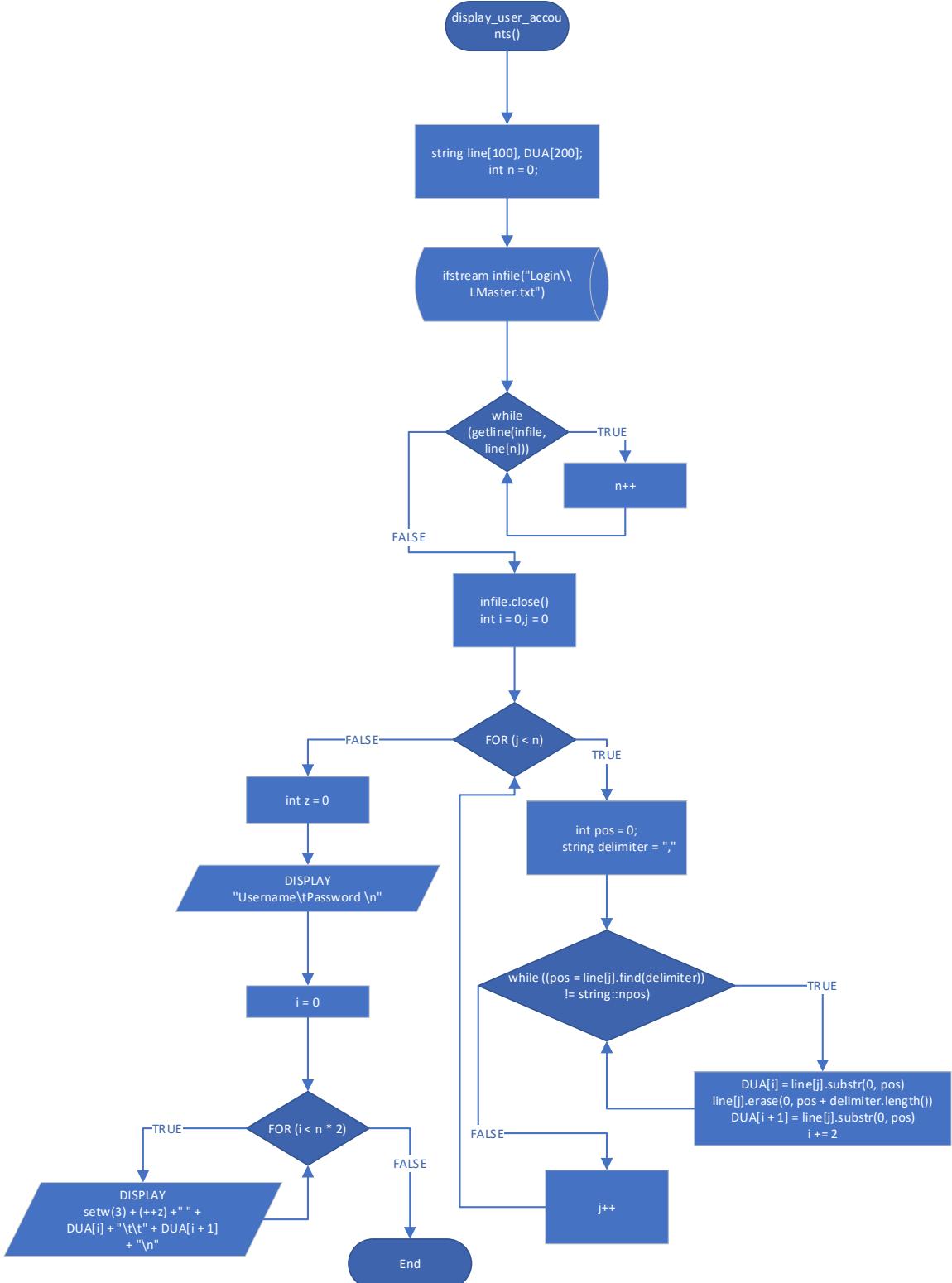


Figure 27 Flow chart display\_user\_accounts

#### 14. deleteUserML

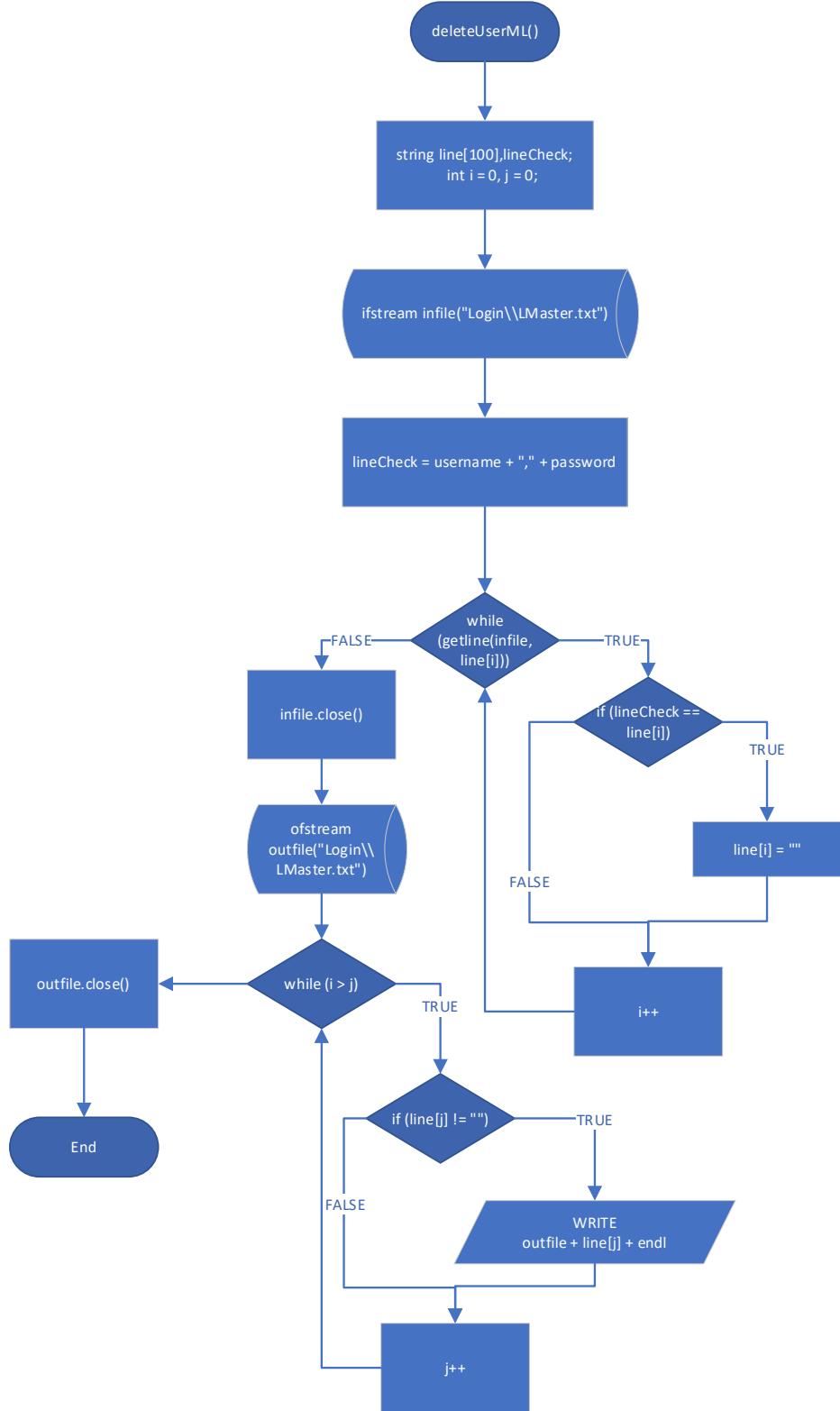


Figure 28 Flow chart deleteUserML

## 15. availableService

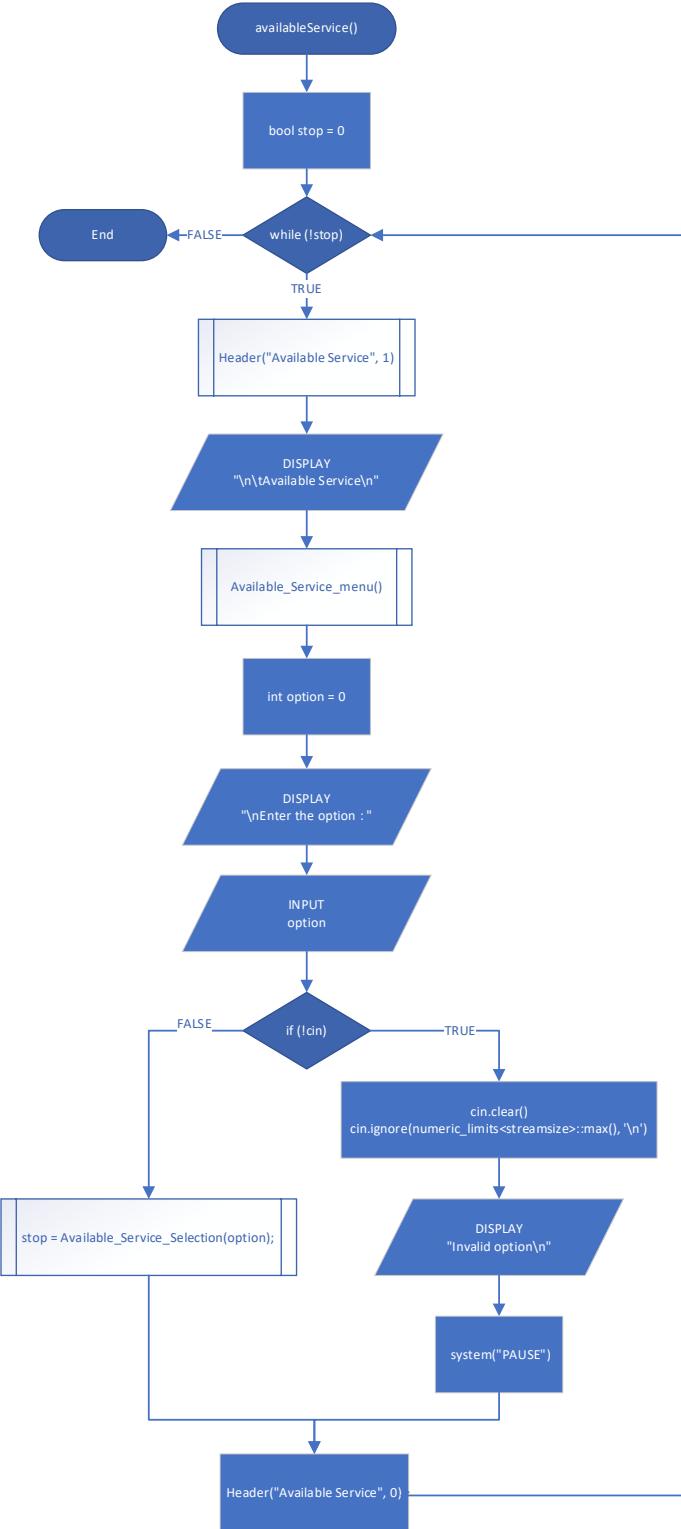


Figure 29 Flow chart availableService

## 16. Available\_Service\_menu

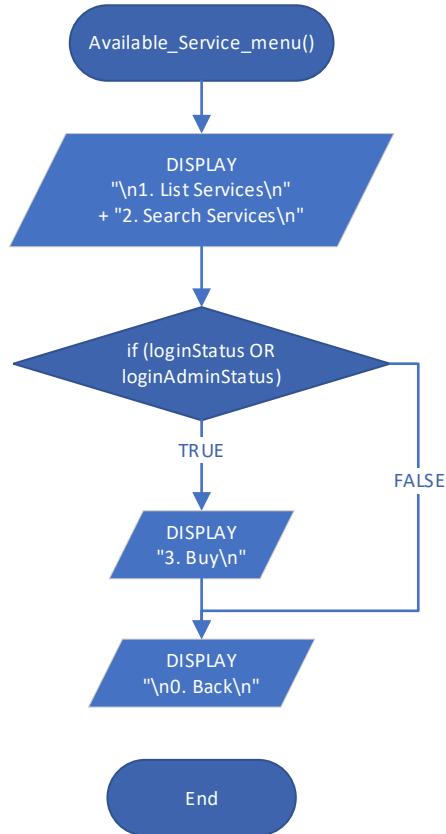


Figure 30 Flow chart Available\_Service\_menu

## 17. Available\_Service\_Selection

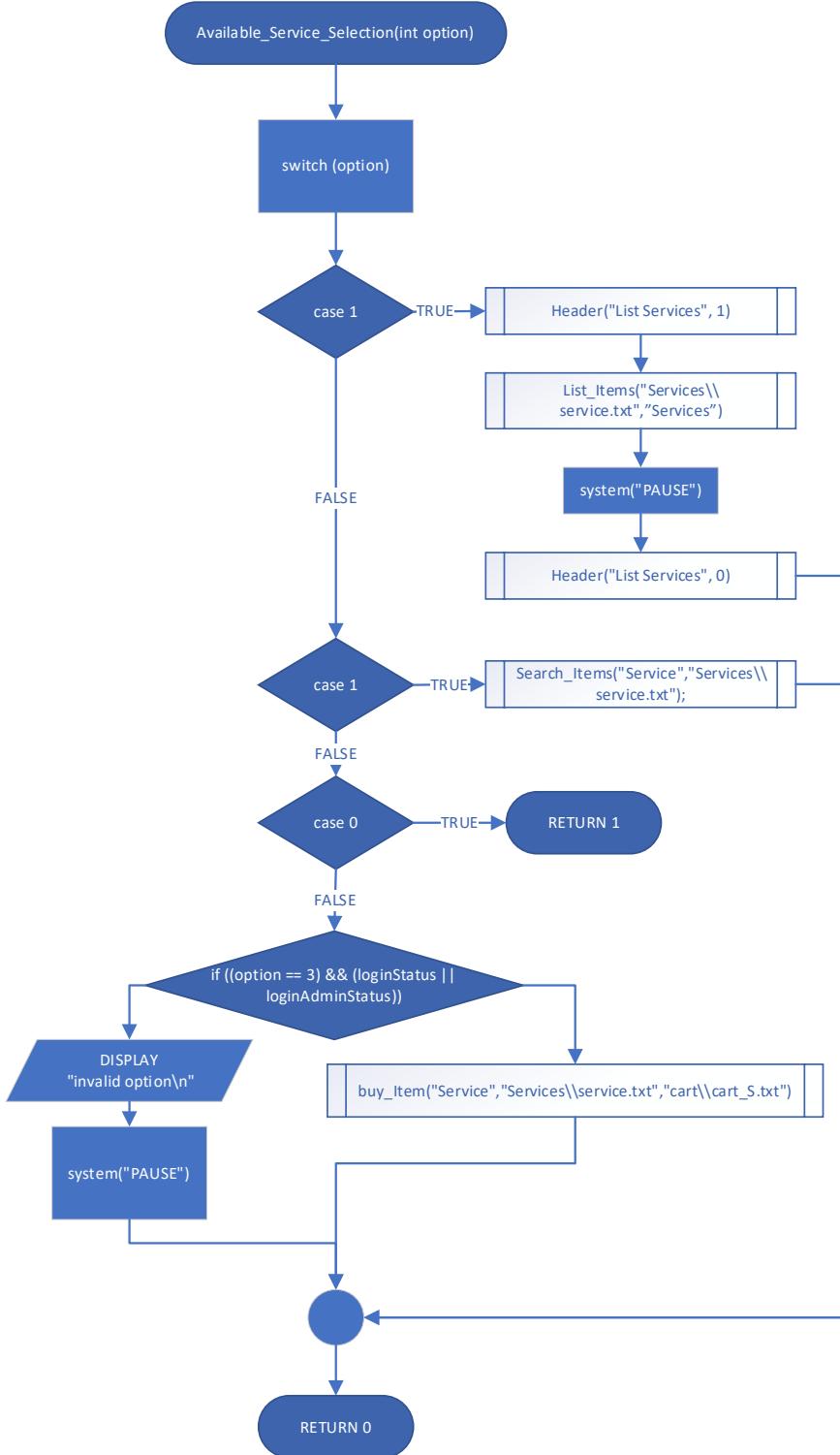


Figure 31 Flow chart Available\_Service\_Selection

## 18. availableProduct

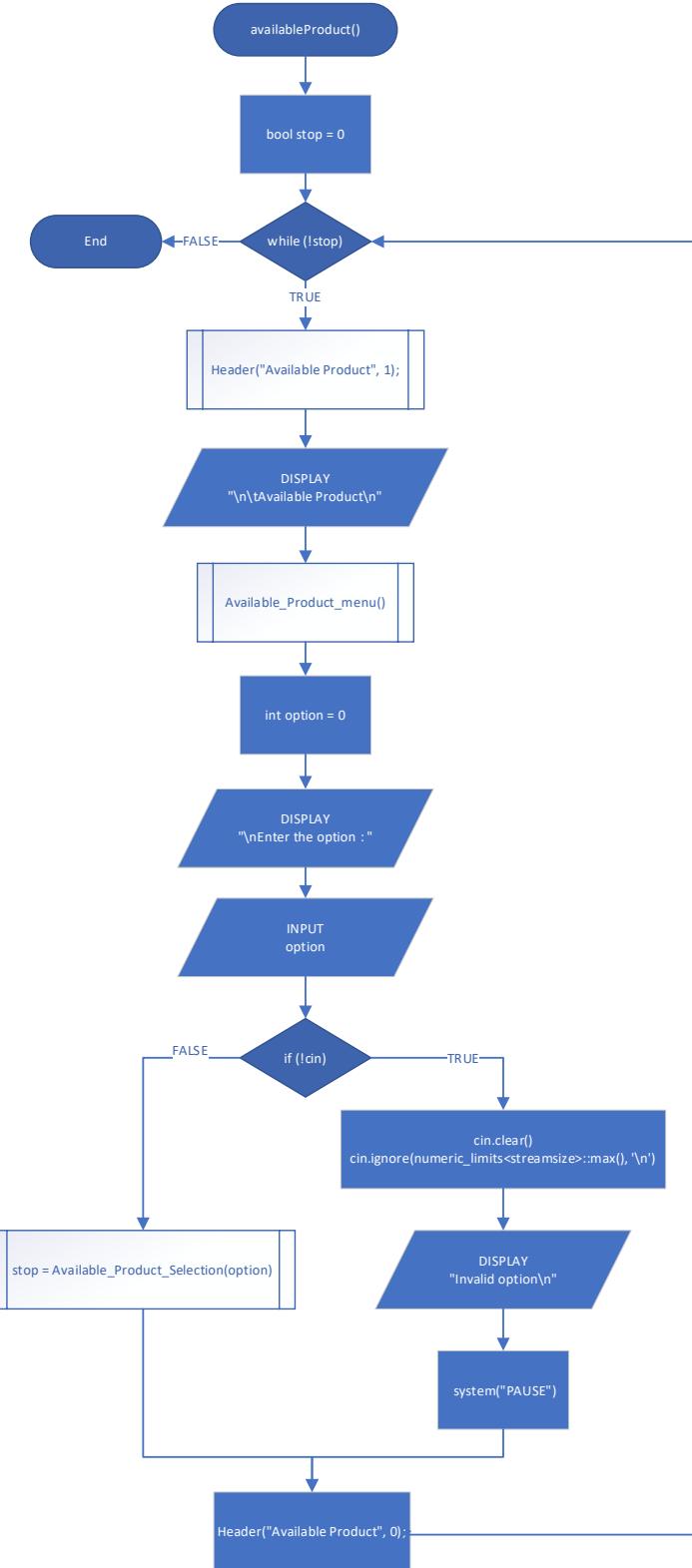


Figure 32 Flow chart availableProduct

## 19. Available\_Product\_menu

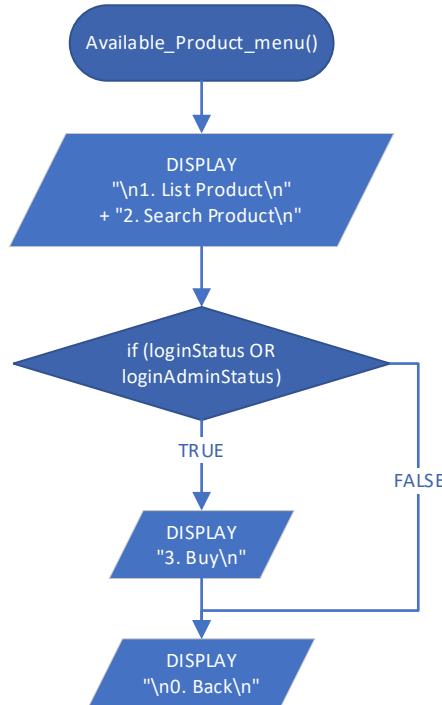


Figure 33 Flow chart Available\_Product\_menu

## 20. Available\_Product\_Selection

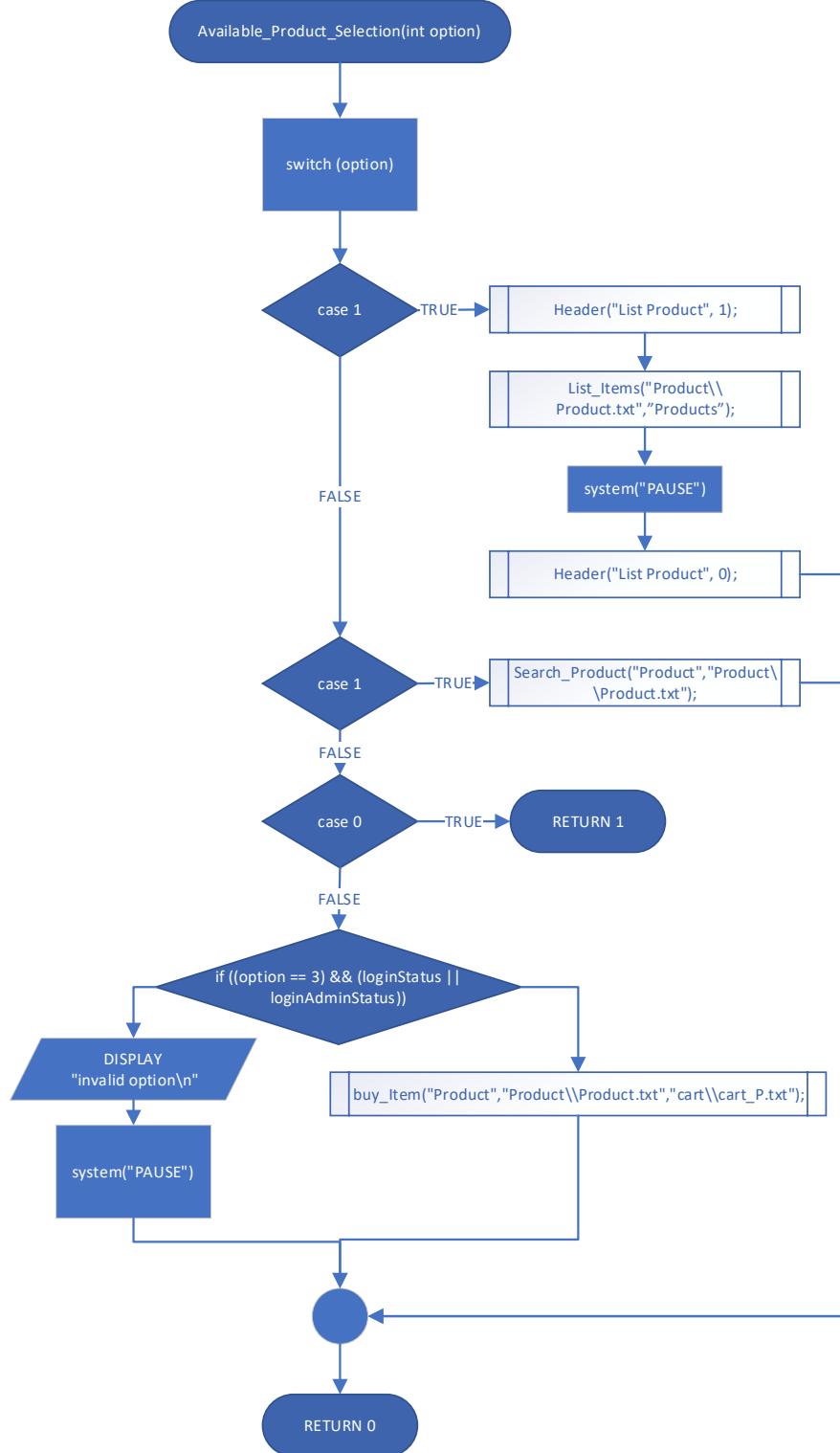
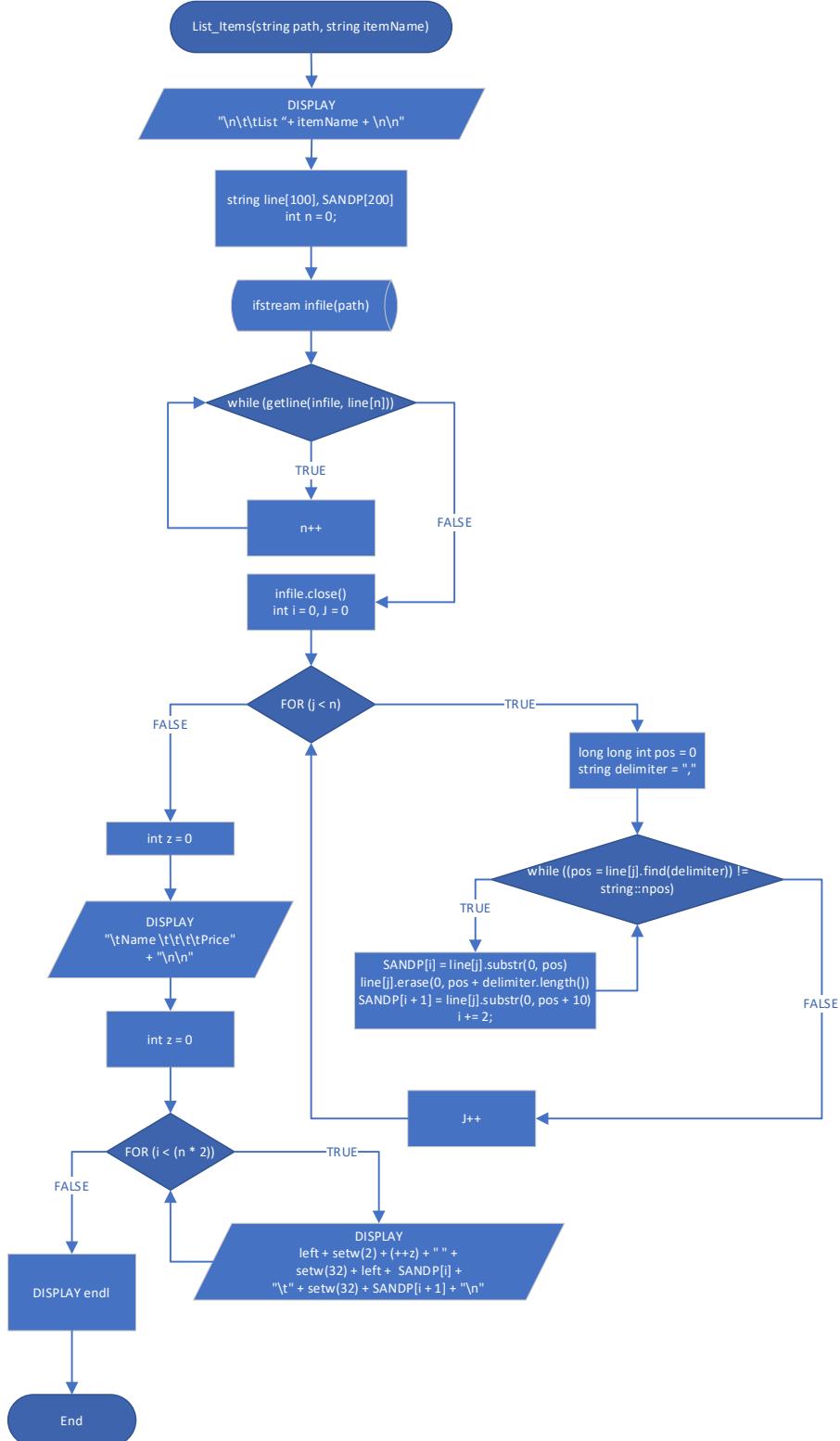


Figure 34 Flow chart Available\_Product\_Selection

## 21. List\_Items



*Figure 35 Flow chart List\_Items*

## 22. buy\_Item

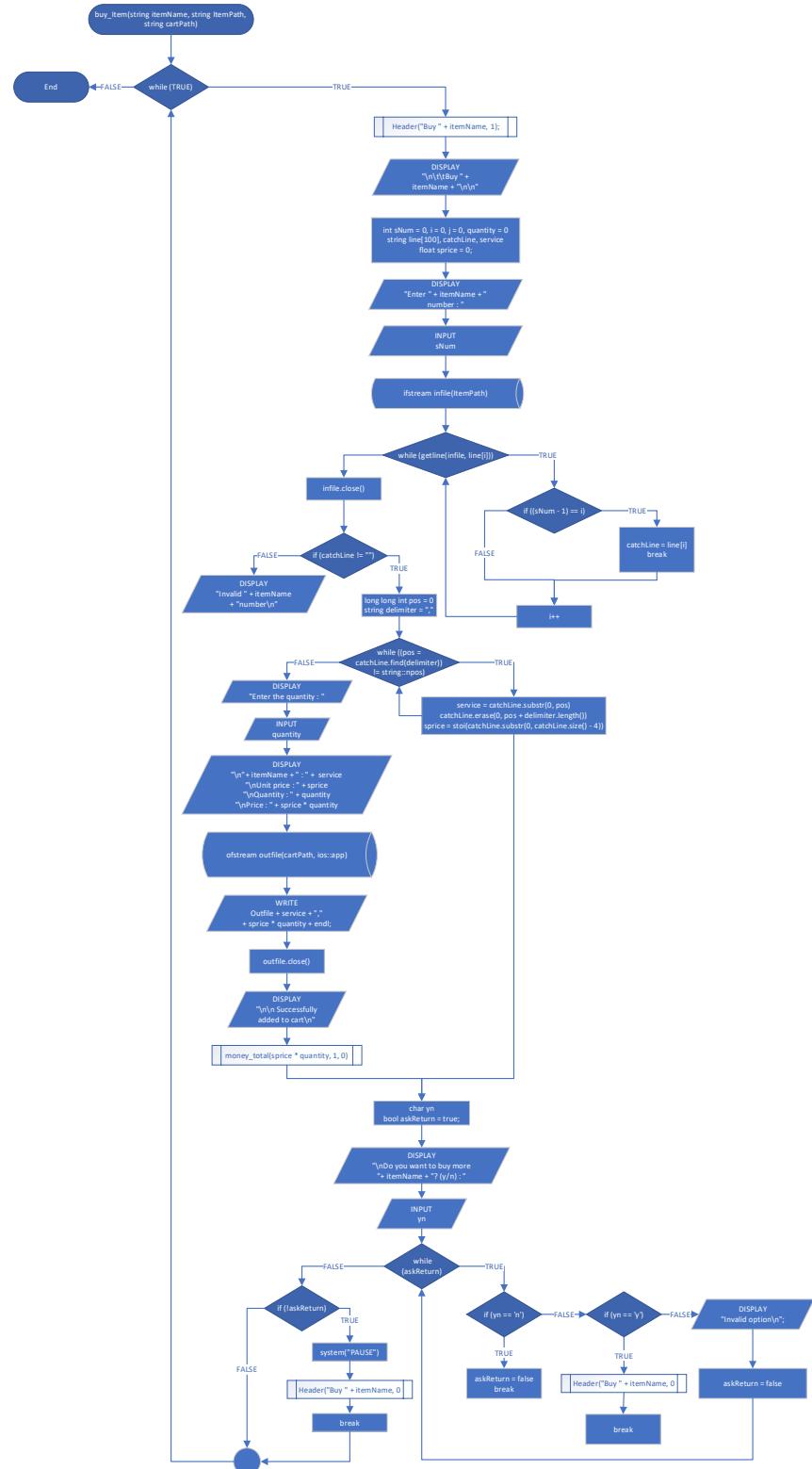


Figure 36 Flow chart buy\_Item

## 23. Search\_Items

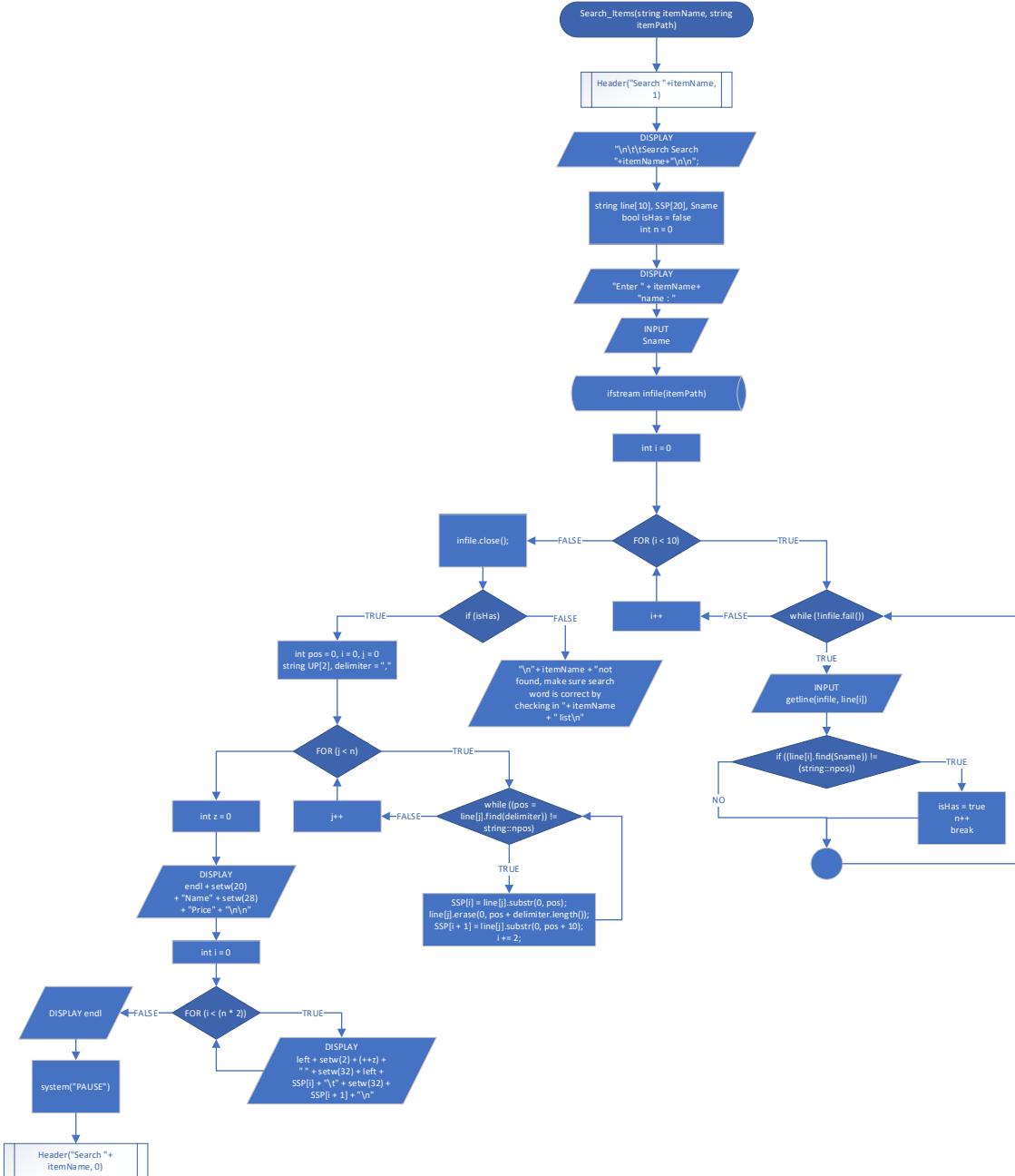


Figure 37 Flow chart

## 24. login

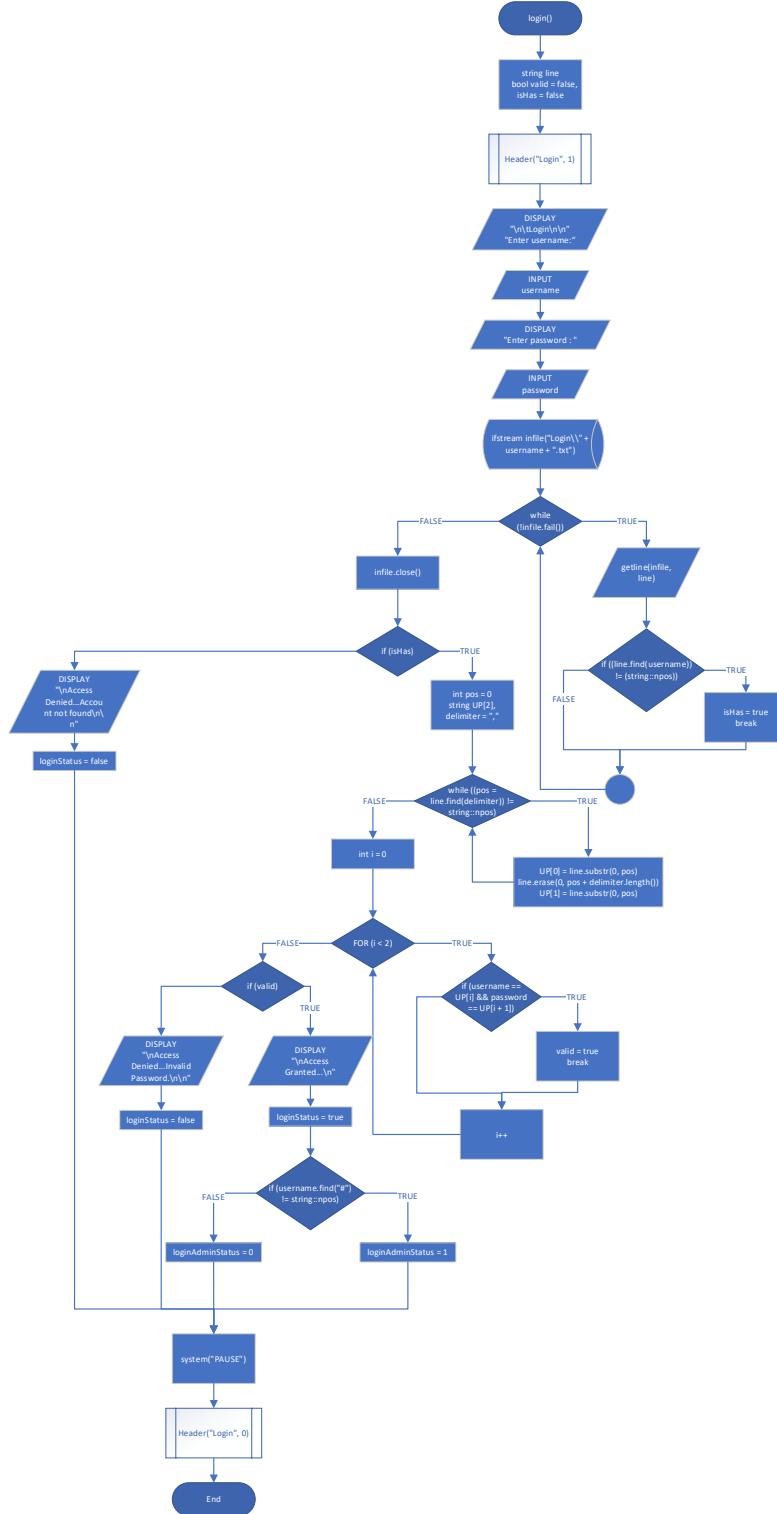


Figure 38 Flow chart login

## 25. signup

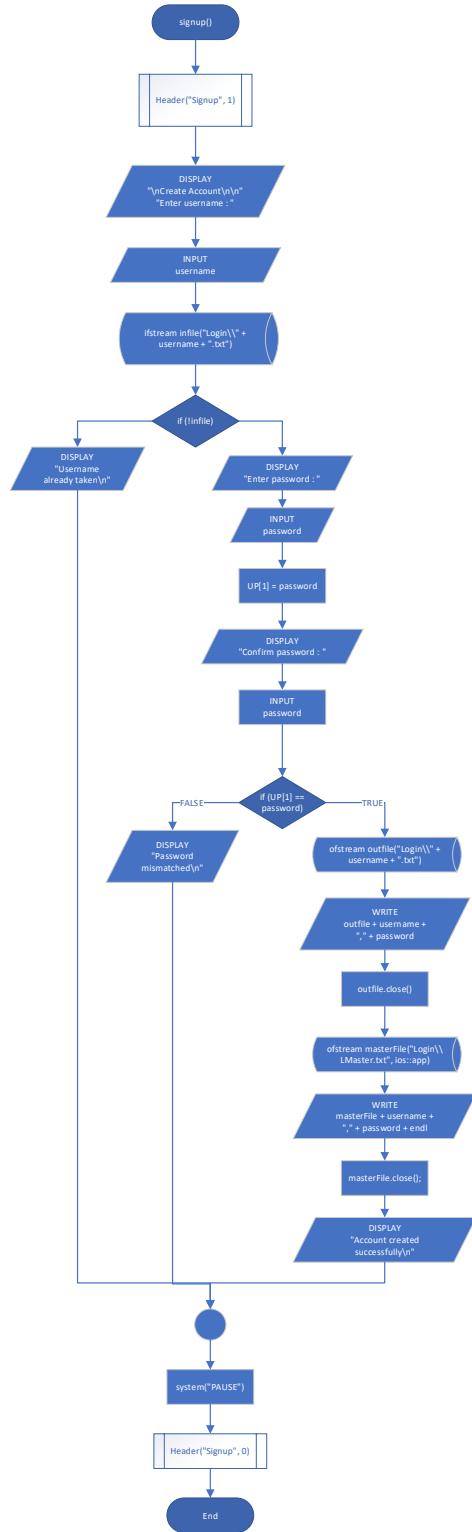


Figure 39 Flow chart signup

## 26. logout

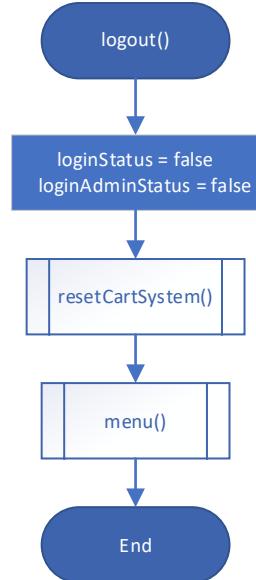


Figure 40 Flow chart logout

## 27. admin\_login\_menu

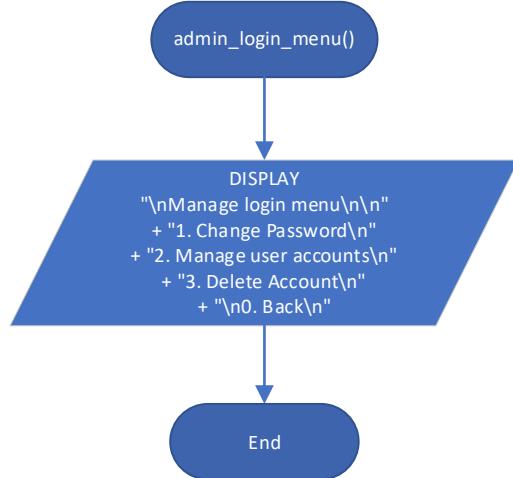


Figure 41 Flow chart admin\_login\_menu

## 28. Manage\_login

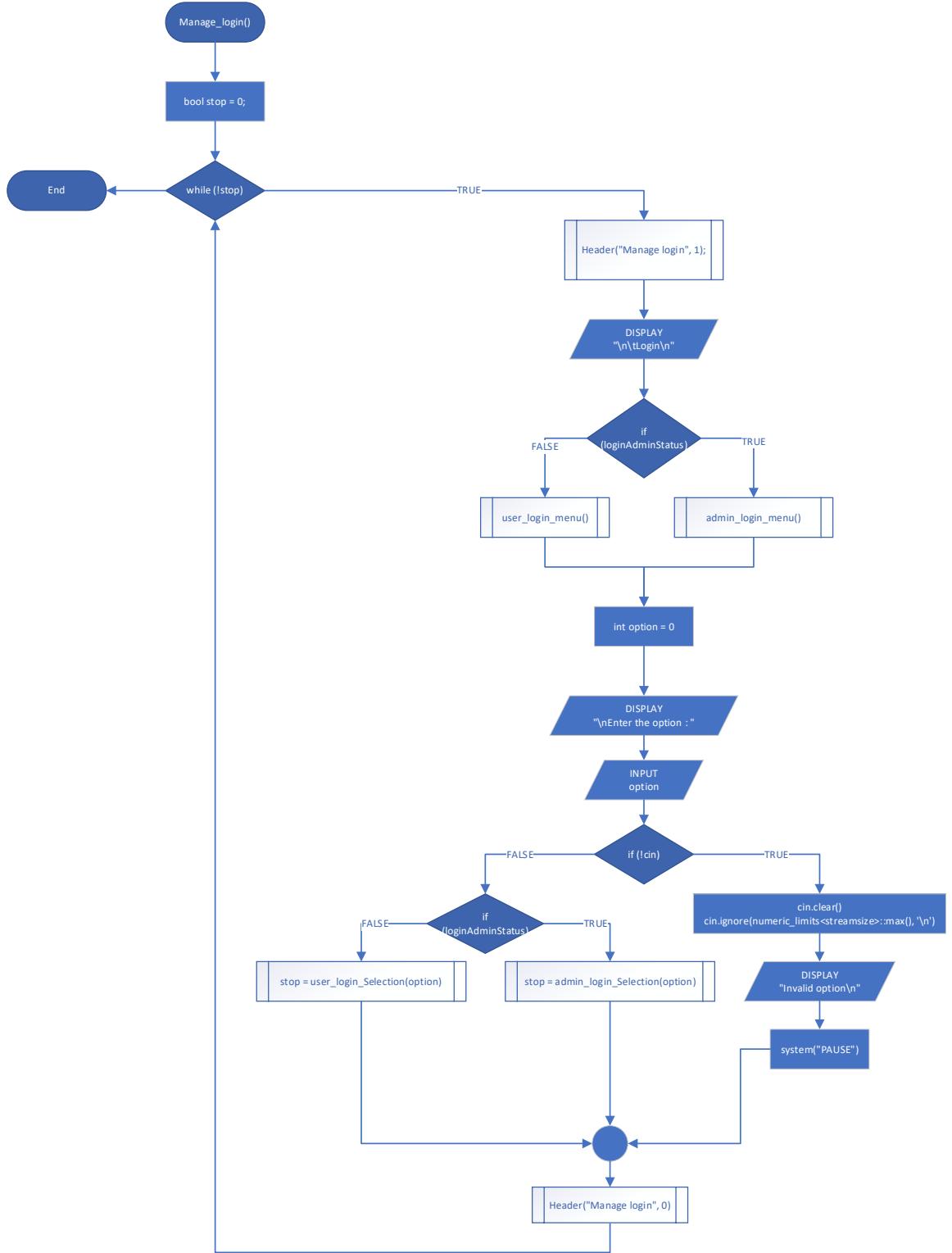


Figure 42 Flow chart Manage\_login

## 29. admin\_login\_Selection

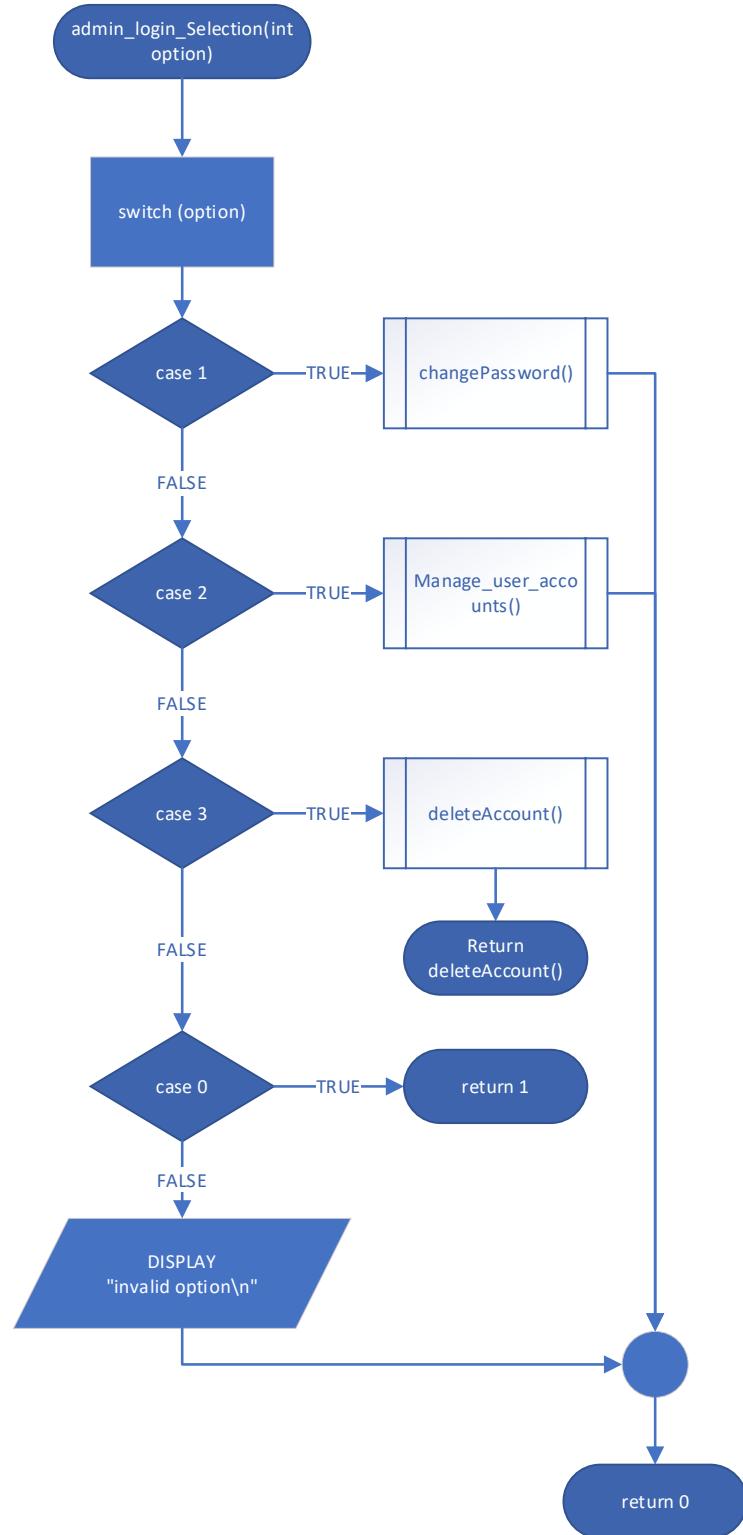


Figure 43 Flow chart admin\_login\_Selection

### 30. user\_login\_menu

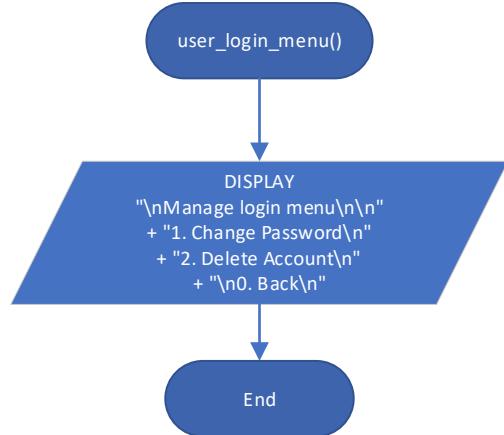


Figure 44 Flow chart user\_login\_menu

### 31. Manage\_Services\_Proucts\_menu

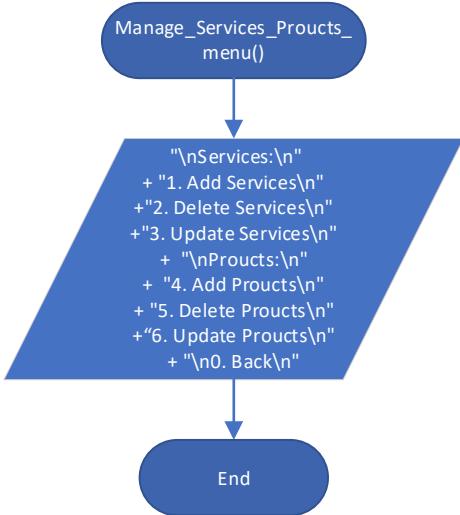


Figure 45 Flow chart Manage\_Services\_Proucts\_menu

### 32. user\_login\_Selection

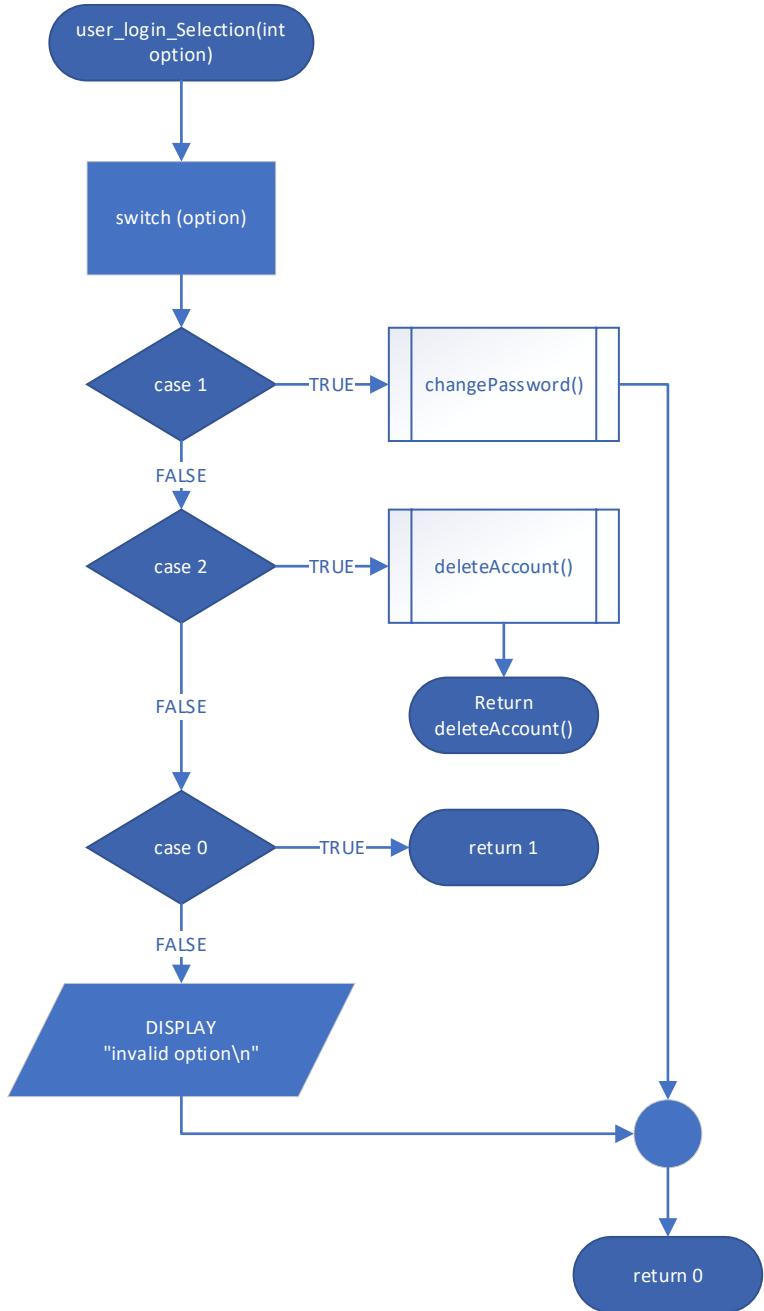


Figure 46 Flow chart user\_login\_Selection

### 33. changePassword

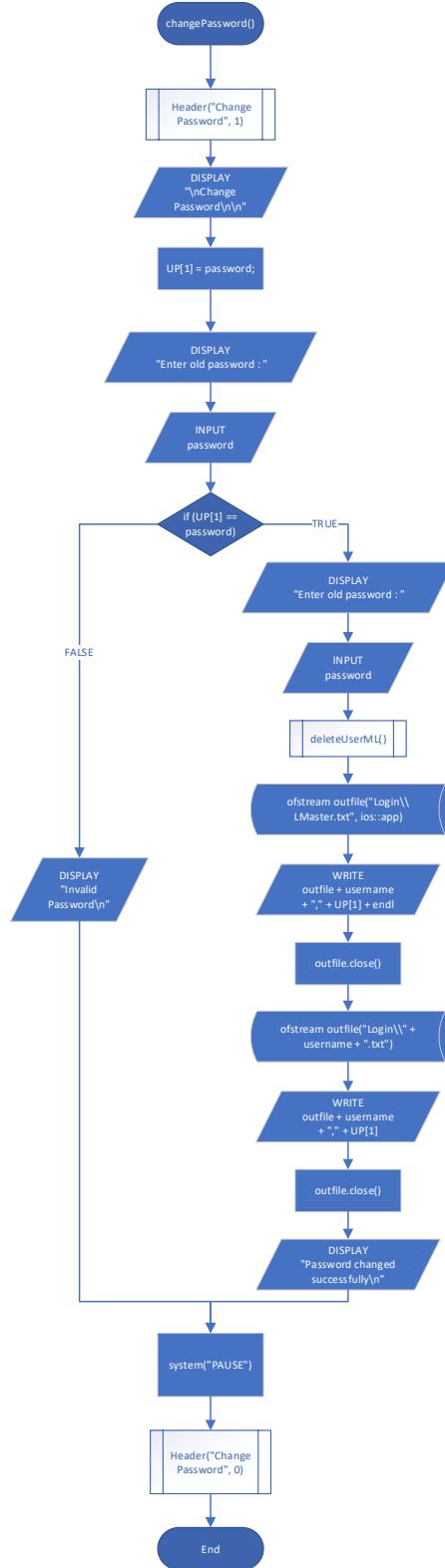


Figure 47 Flow chart changePassword

### 34. deleteAccount

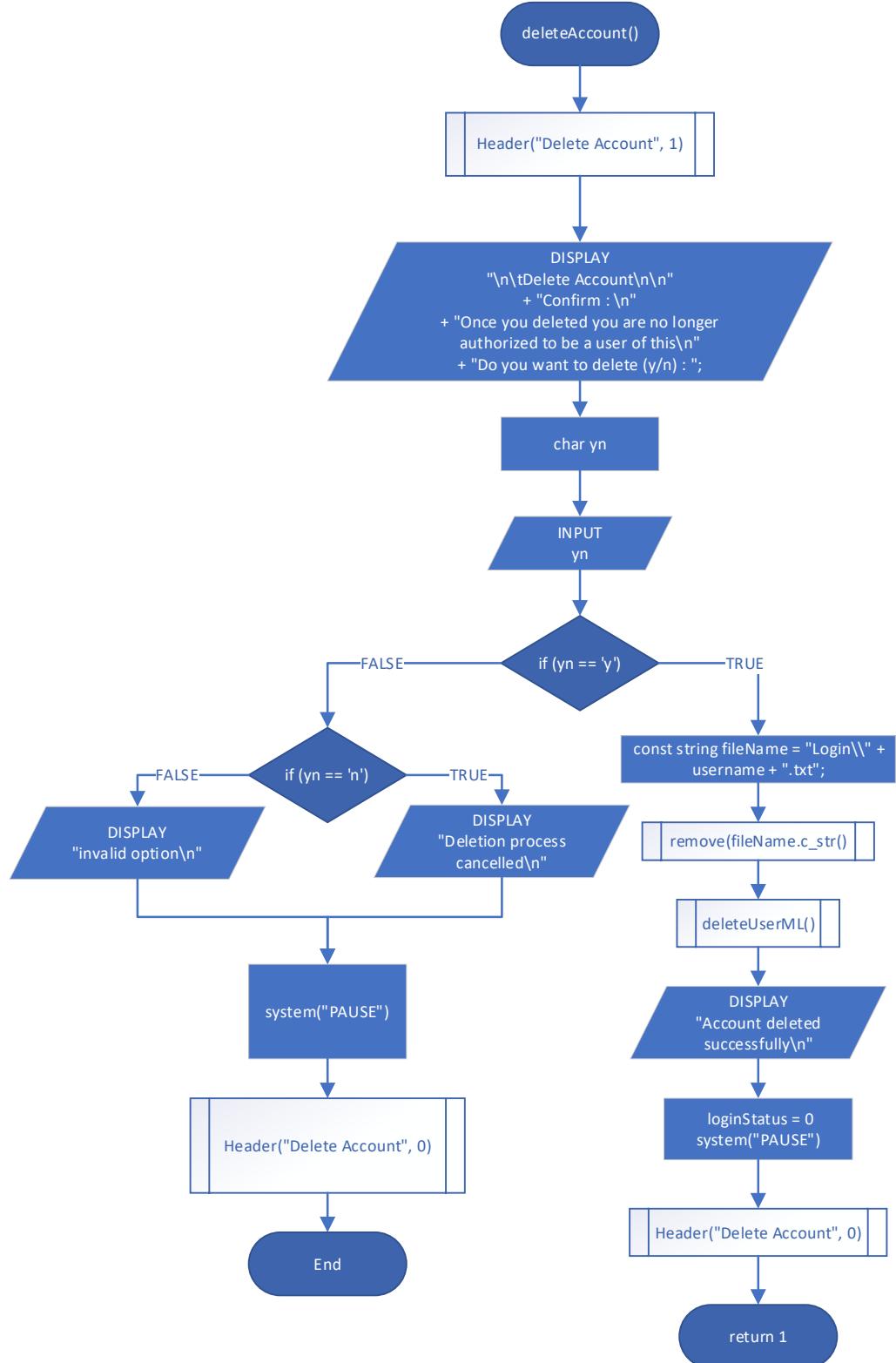


Figure 48 Flow chart deleteAccount

### 35. Manage\_Services\_Products

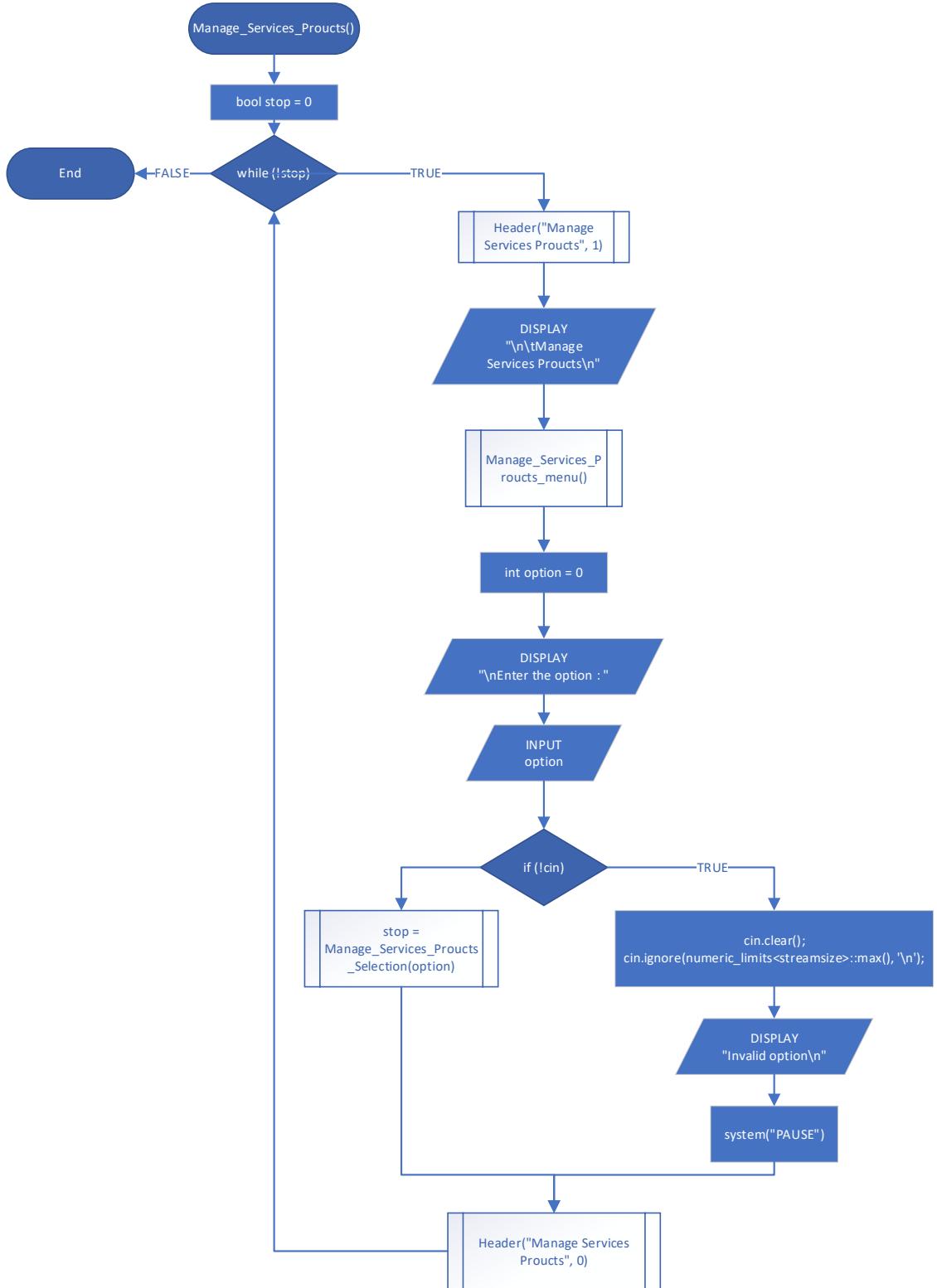


Figure 49 Flow chart Manage\_Services\_Products

### 36. Manage\_Services\_Products\_Selection

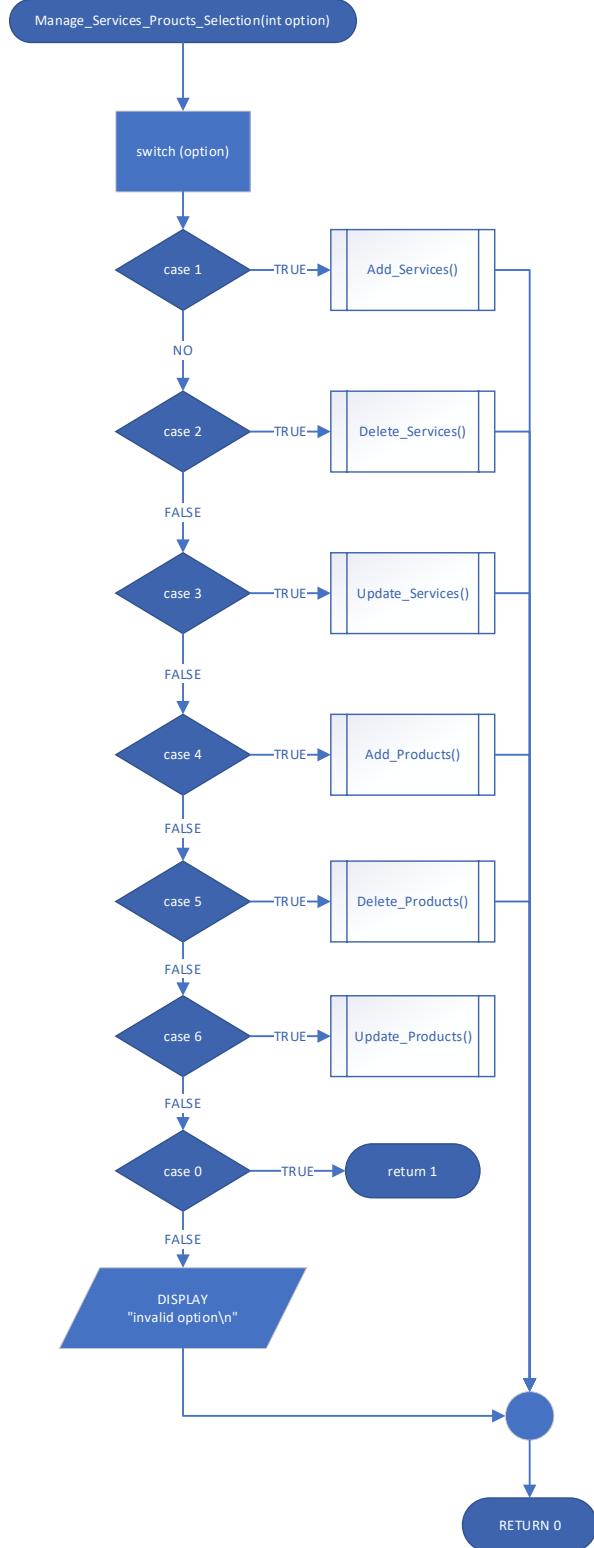


Figure 50 Flow chart Manage\_Services\_Products\_Selection

### 37. Add\_Services

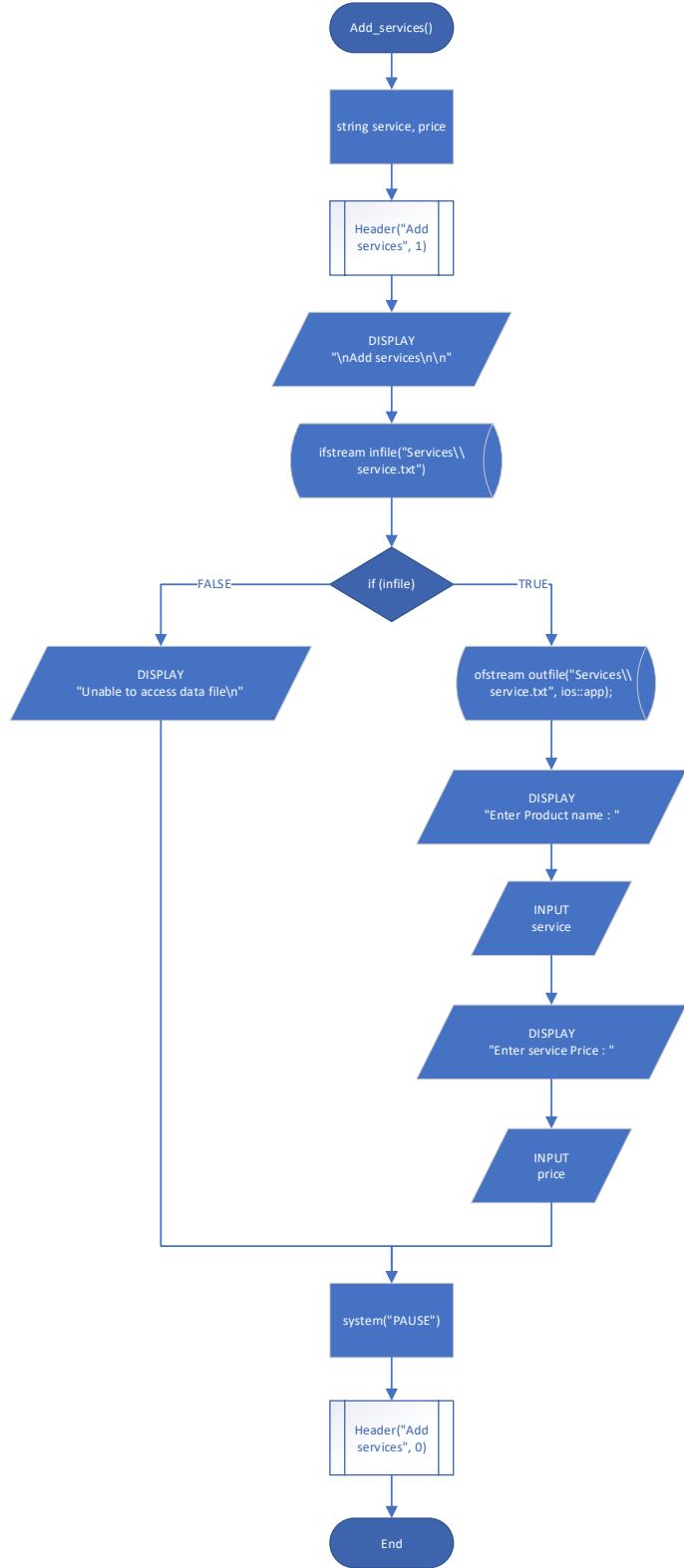


Figure 51 Flow chart Add\_Services

## 38. Delete\_Services

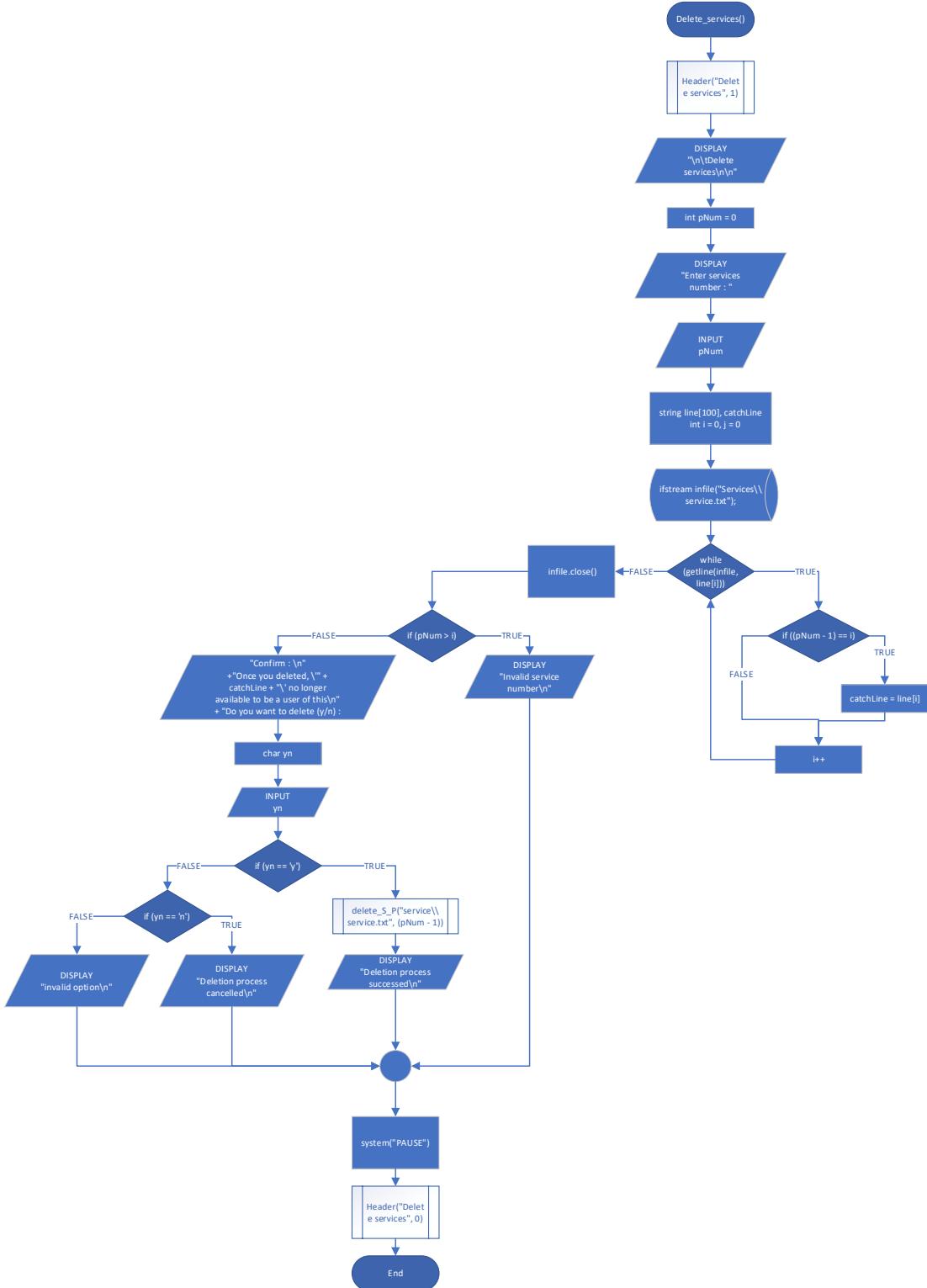


Figure 52 Flow chart Delete\_Services

### 39. Update\_Services

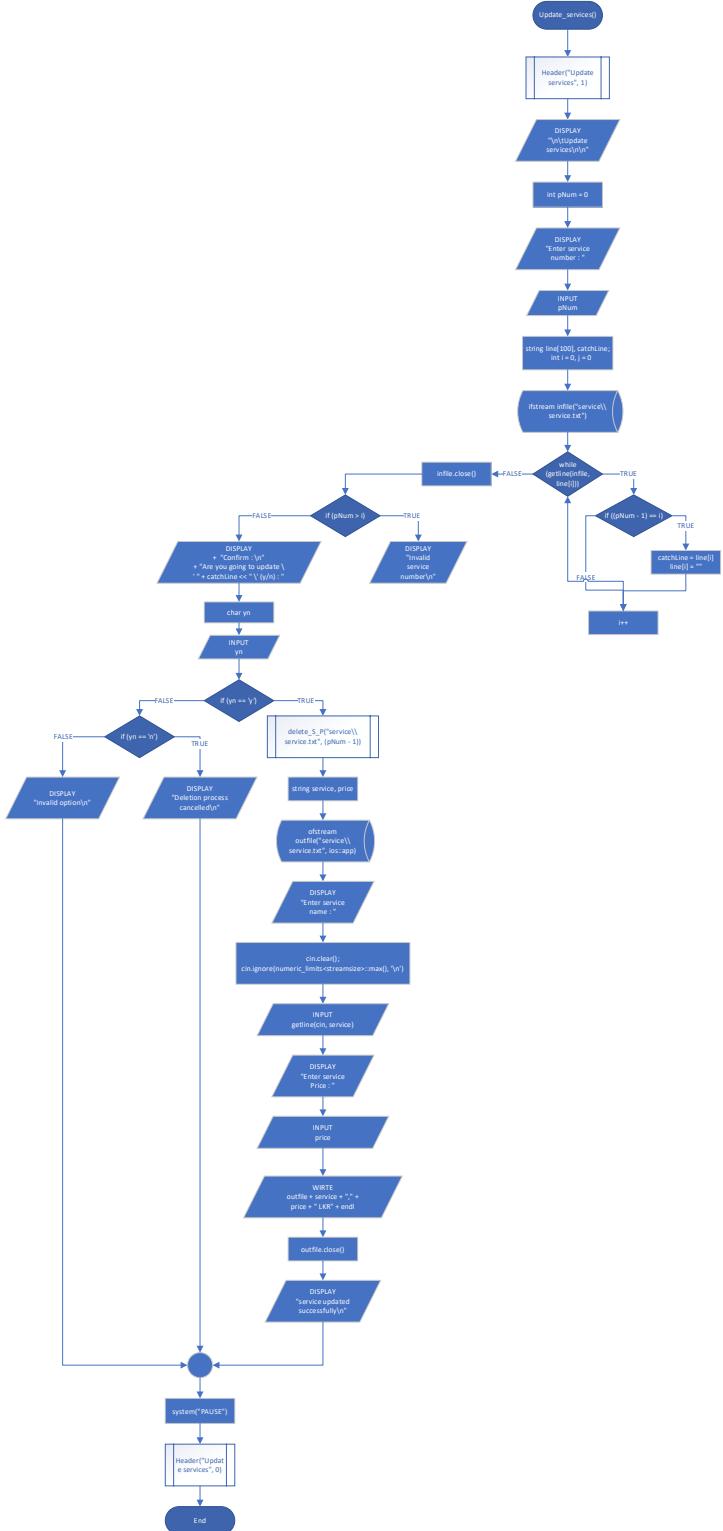


Figure 53 Flow chart Update\_Services

#### 40. Add\_Products

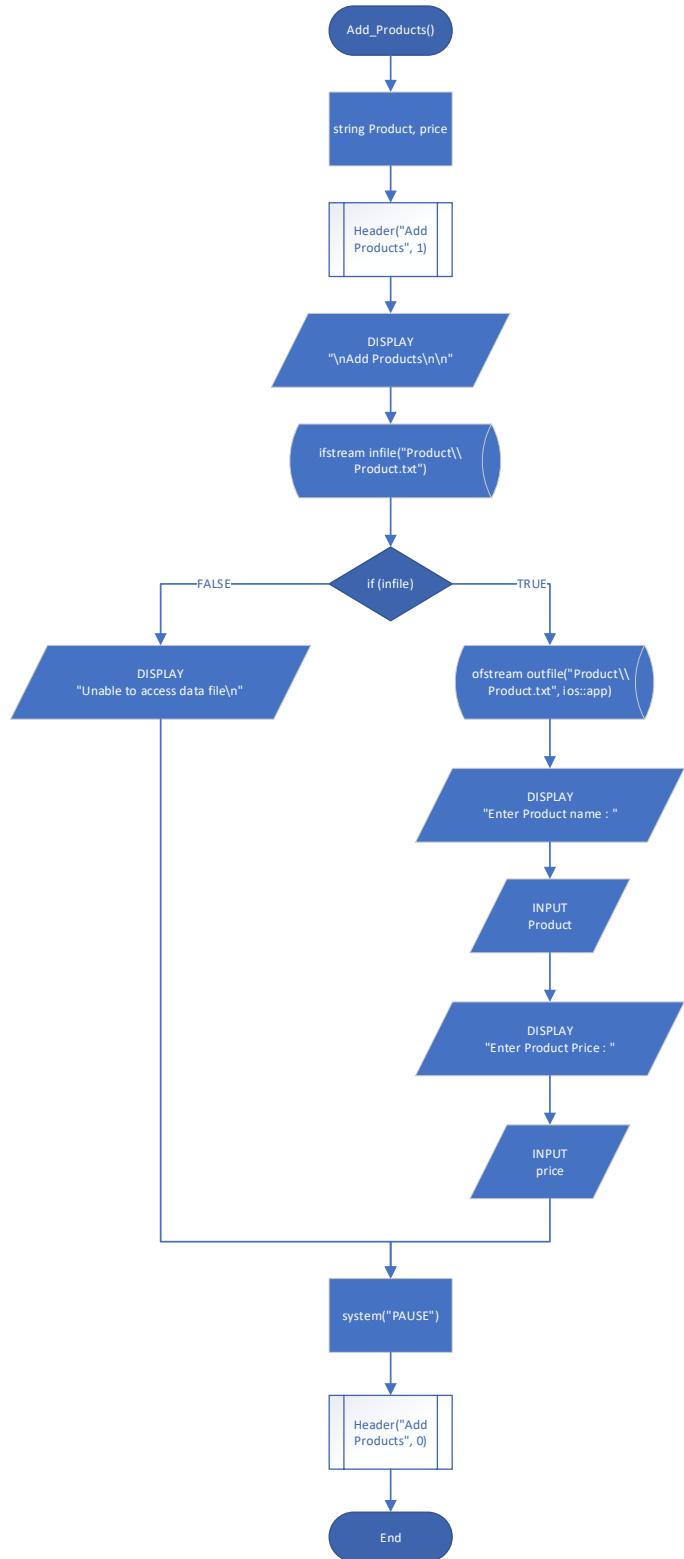


Figure 54 Flow chart Add\_Products

## 41. Delete\_Products

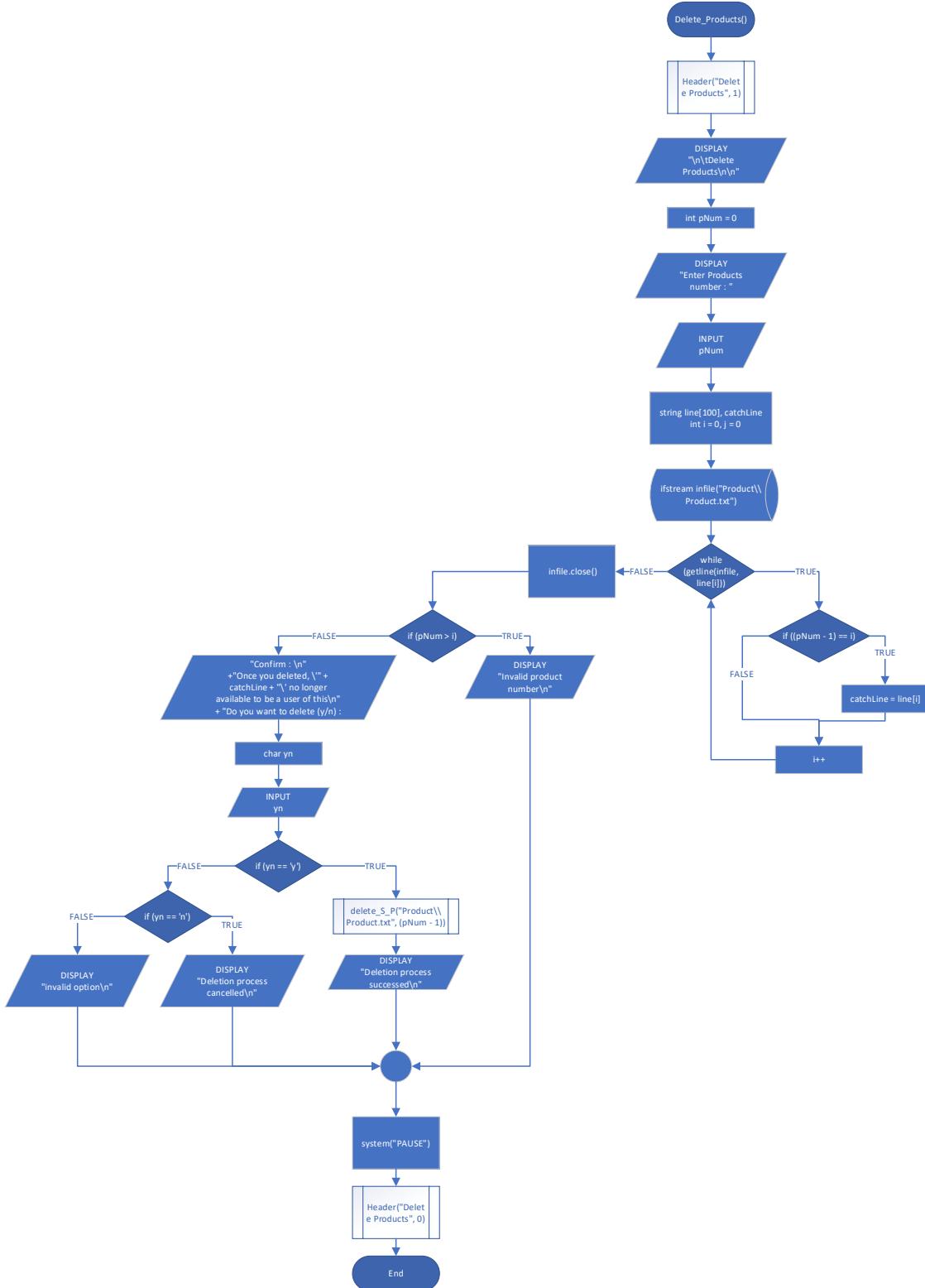


Figure 55 Flow chart Delete\_Products

## 42. Update\_Products

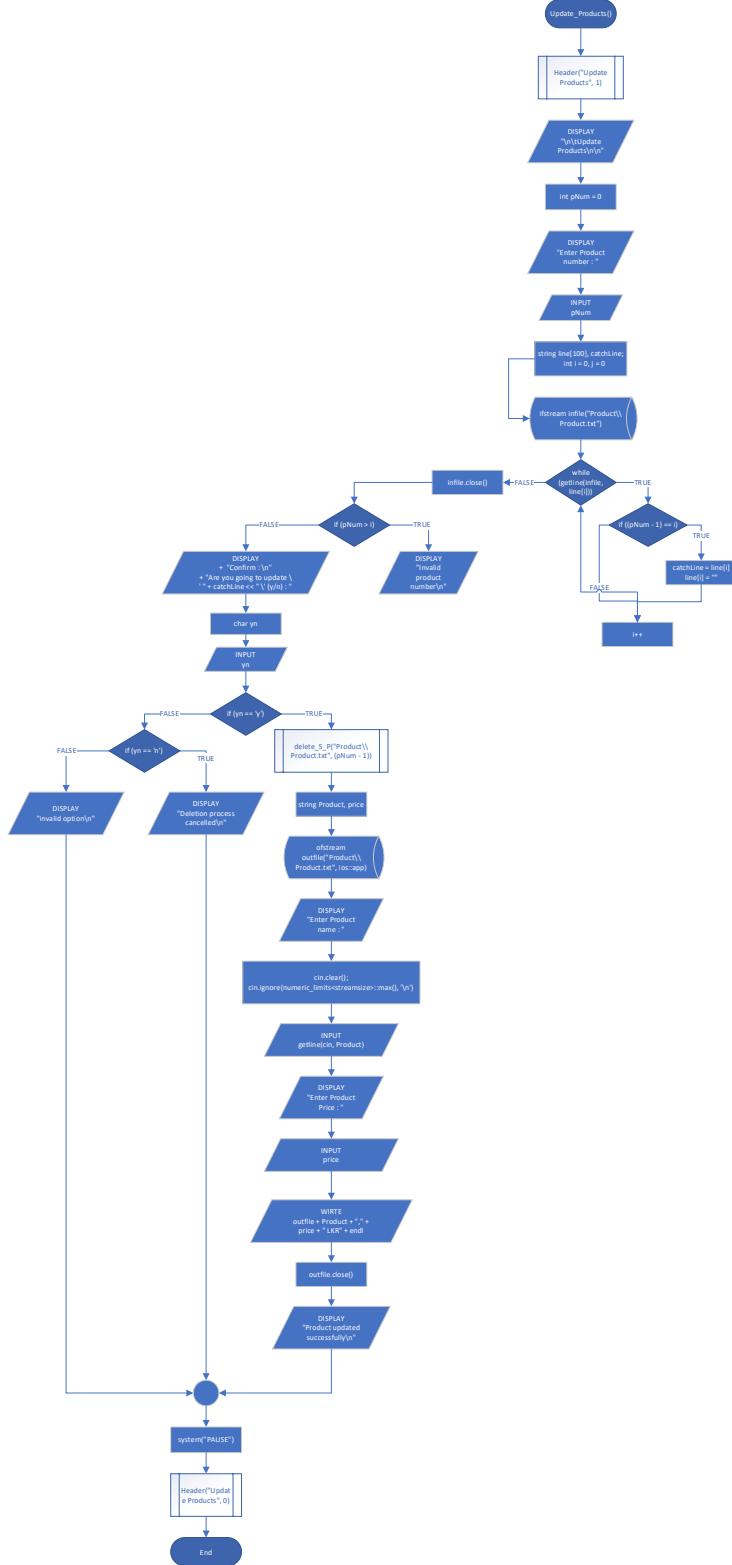


Figure 56 Flow chart Update\_Products

### 43. cart

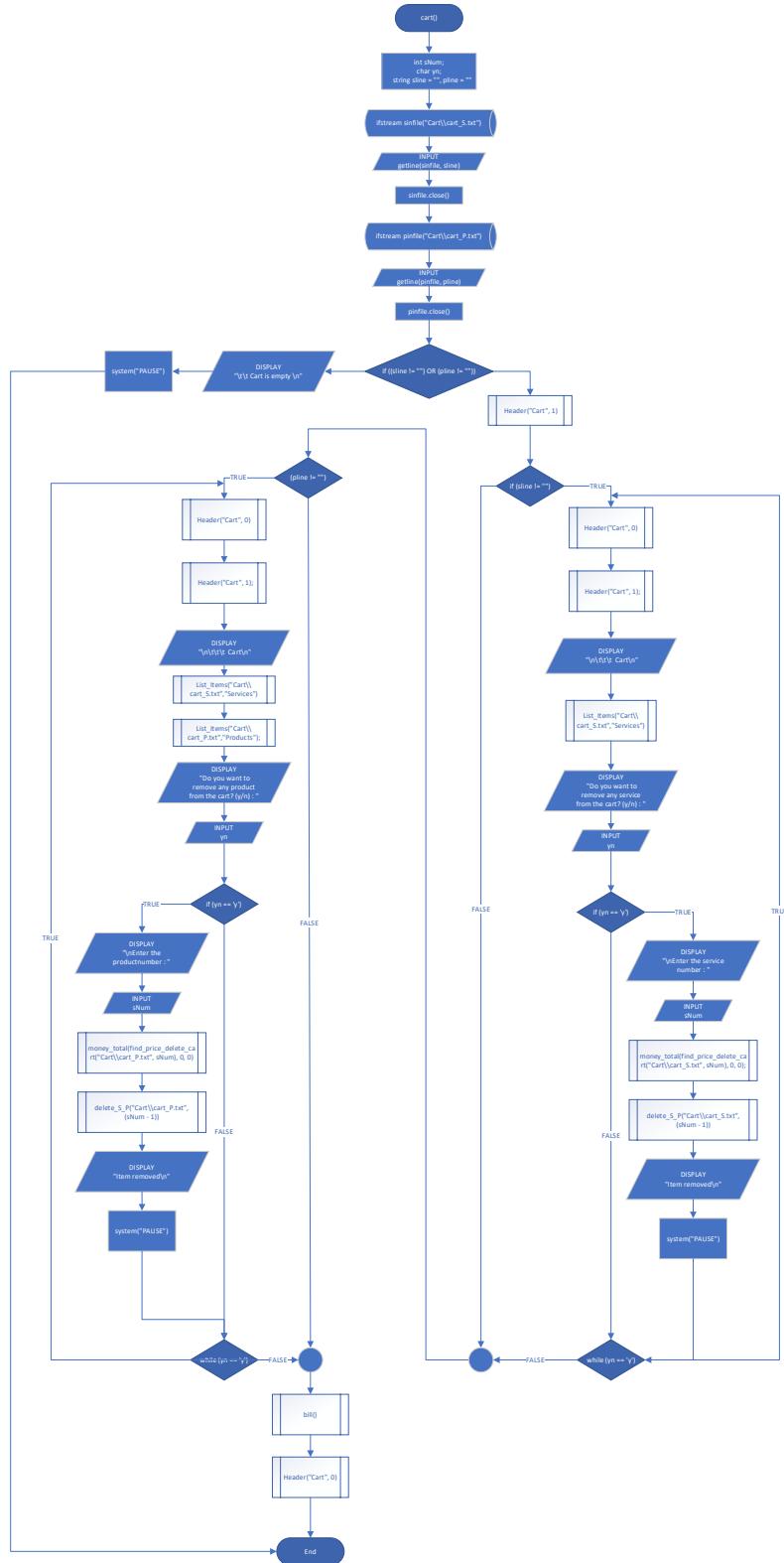


Figure 57 Flow chart cart

#### 44. find\_price\_delete\_cart

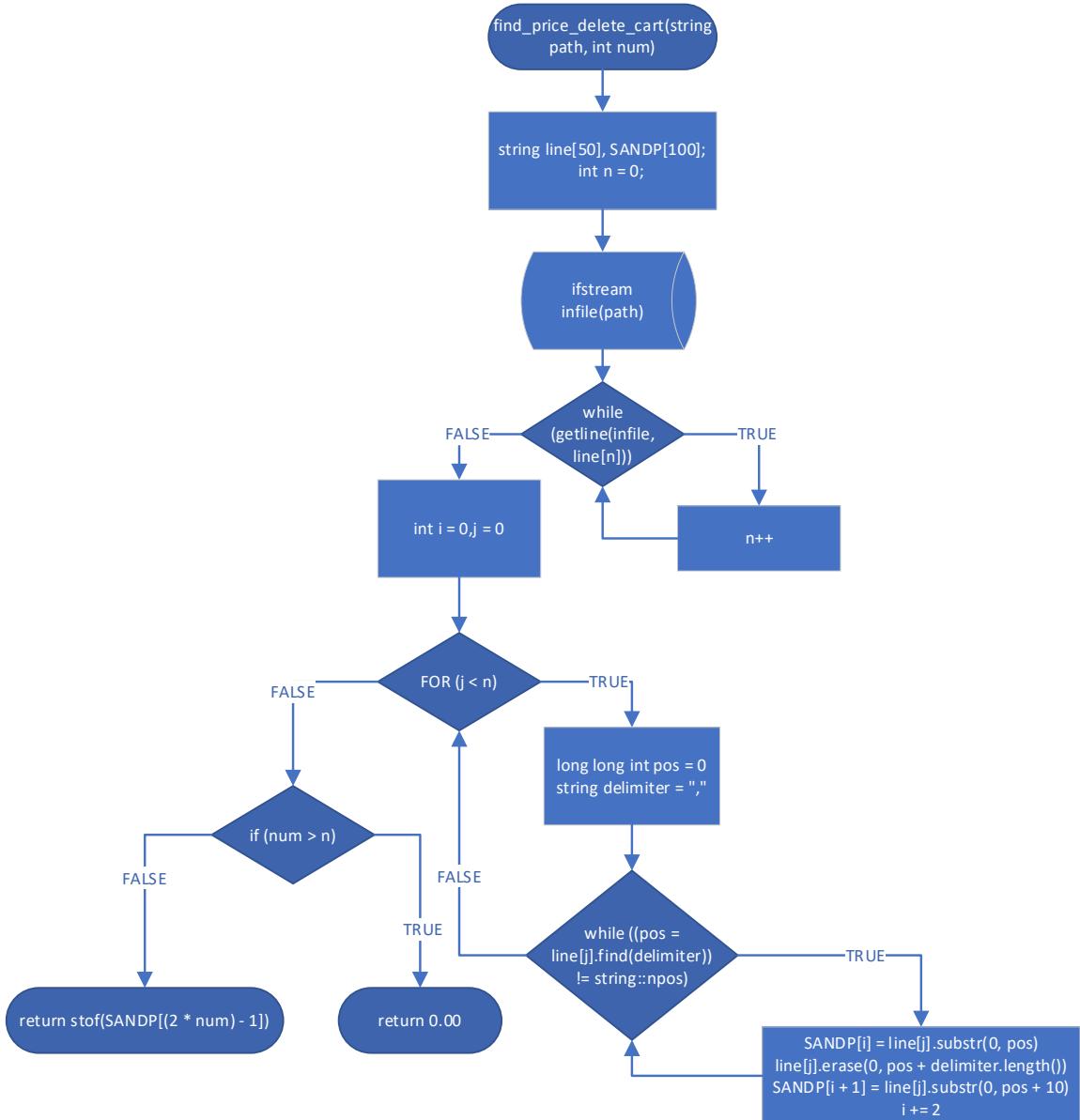


Figure 58 Flow chart `find_price_delete_cart`

#### 45. bill

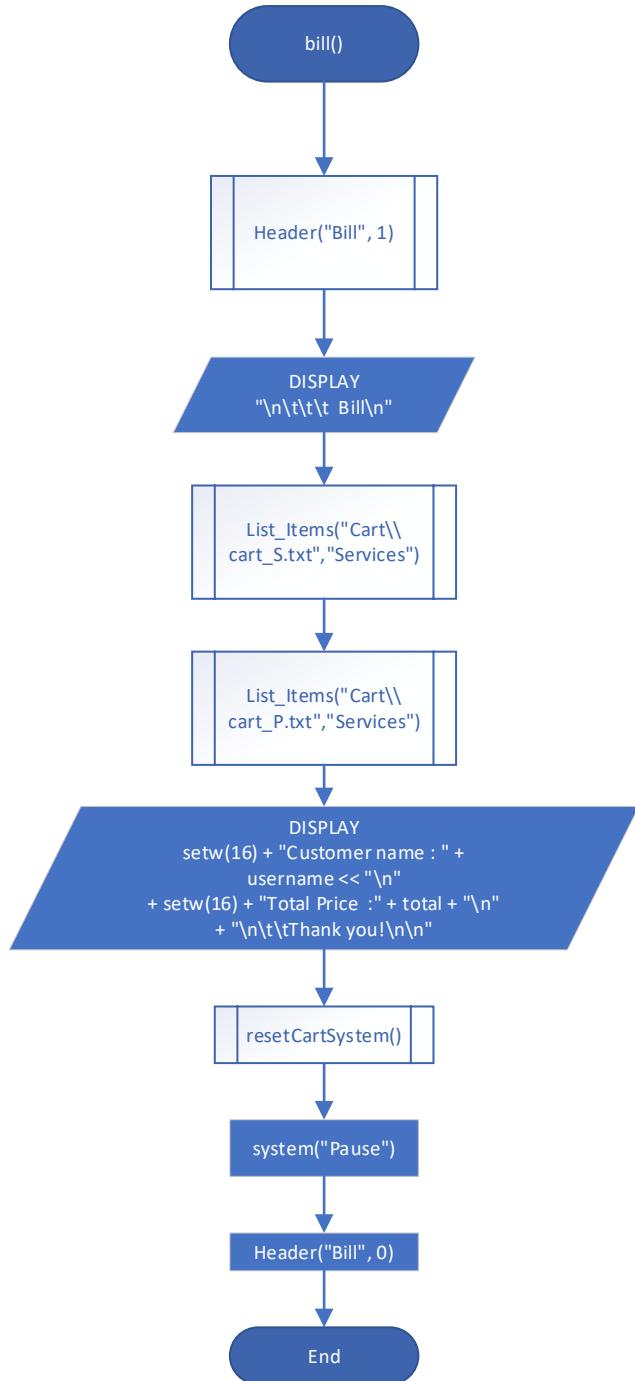


Figure 59 Flow chart bill

#### 46. money\_total

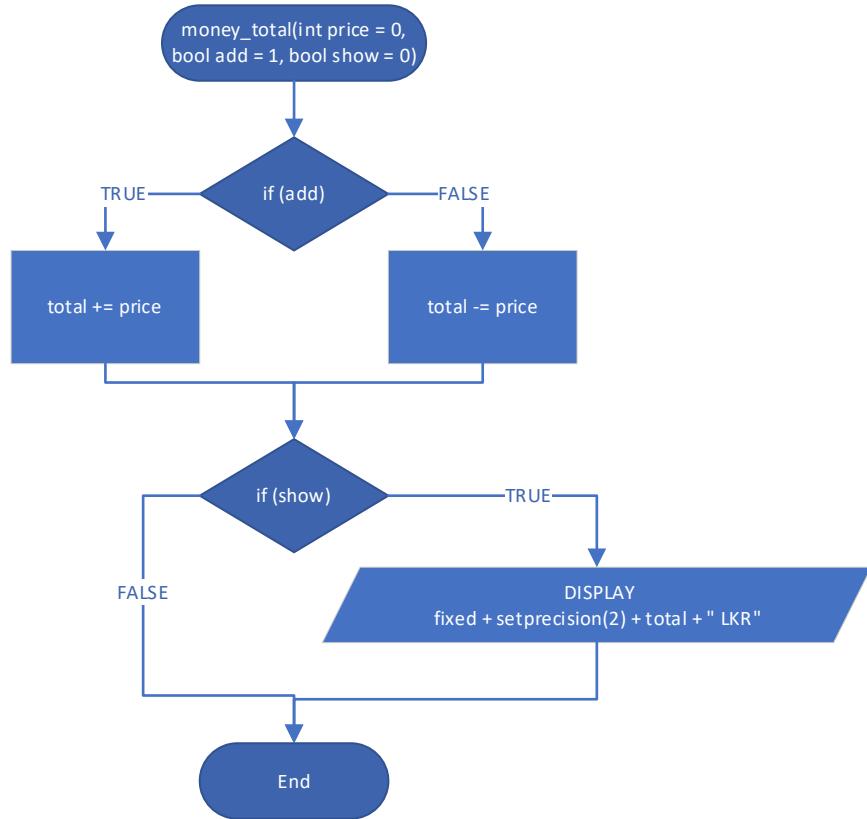


Figure 60 Flow chart money\_total

#### 47. delete\_S\_P

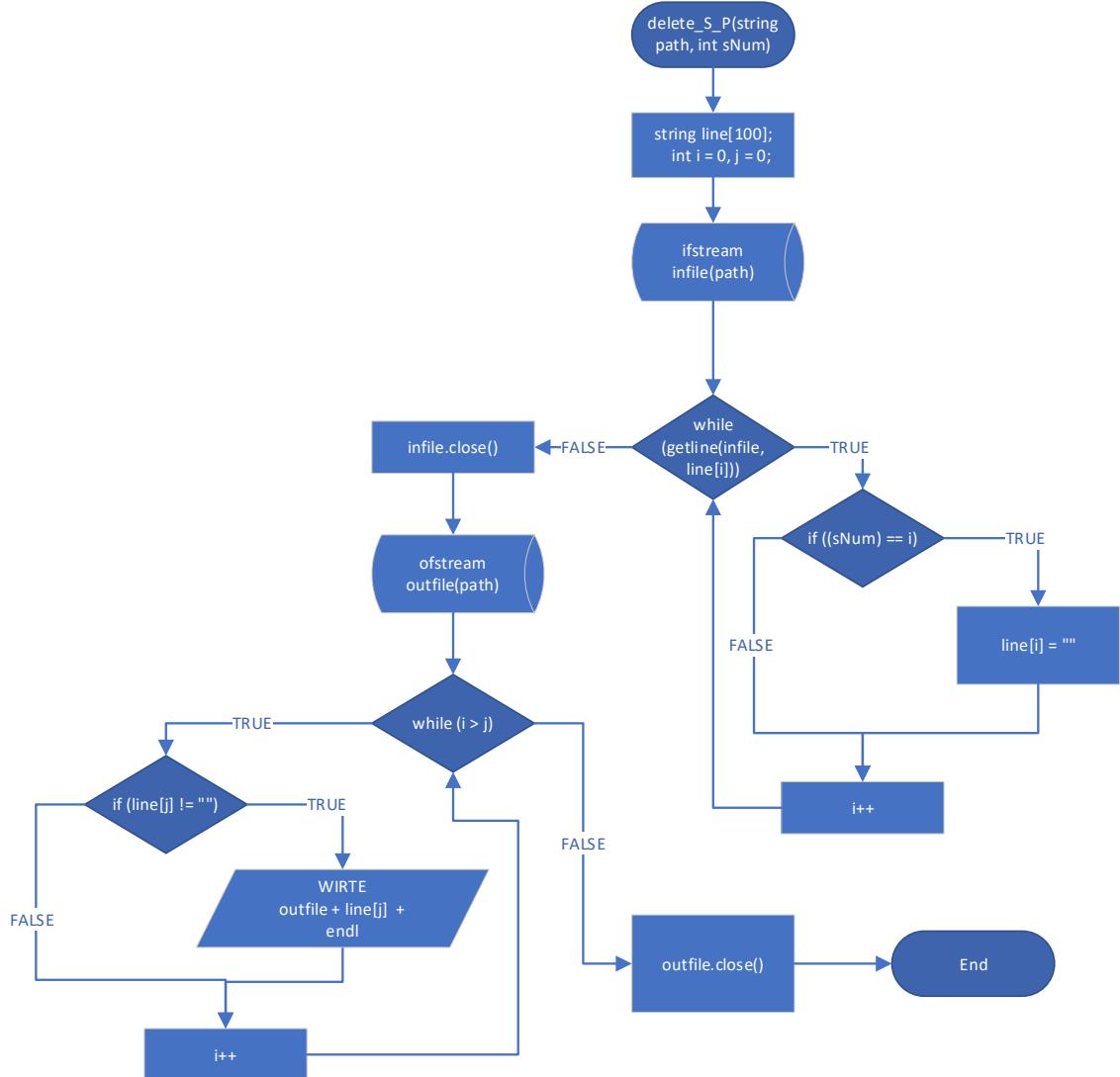


Figure 61 Flow chart delete\_S\_P

## User Interface Design for the System

- Main menu

- Guest

```
=====
Best Auto Part
=====

Main menu
-----
          0.00 LKR

Guest

1. Available Services
2. Available products

3. Login
4. Signup
0. Exit

Enter the option :
```

Figure 62 main menu of guest

- User

```
=====
Best Auto Part
=====

Main menu
-----
          0.00 LKR

Welcome Jiva

1. Available Services
2. Available products
3. Manage login
4. Cart

5. Logout
0. Exit

Enter the option :
```

Figure 63 main menu of user

- Admin

```
=====
Best Auto Part
=====

Main menu
-----
0.00 LKR

Welcome admin #Suresh

1. Available Services
2. Available products
3. Manage services & products
4. Manage login
5. Cart

6. Logout
0. Exit

Enter the option :
```

Figure 64 main menu of admin

- Login

- Login success

```
=====
Best Auto Part
=====

Main menu -> Login
-----
0.00 LKR

Login

Enter username : Jiva
Enter password : 5580

Access Granted...
Press any key to continue . . .
```

Figure 65 login success

- **Login fail**

```
=====
Best Auto Part
=====

Main menu -> Login
-----
0.00 LKR

Login

Enter username : Ravi
Enter password : a12c

Access Denied...Account not found

Press any key to continue . . .
```

Figure 66 login fail

- **Manage login (user)**

```
=====
Best Auto Part
=====

Main menu -> Manage login
-----
0.00 LKR

Login

Manage login menu

1. Change Password
2. Delete Account

0. Back

Enter the option :
```

Figure 67 manage login of user

- **Manage login (admin)**

```
=====
Best Auto Part
=====

Main menu -> Manage login
-----
0.00 LKR

Login

Manage login menu

1. Change Password
2. Manage user accounts
3. Delete Account

0. Back

Enter the option :
```

Figure 68 manage login of admin

- **Change password**

```
=====
Best Auto Part
=====

Main menu -> Manage login -> Change Password
-----
0.00 LKR

Change Password

Enter old password : 5580
Enter new password : 2121
Password changed successfully
Press any key to continue . . .
```

Figure 69 change password

- **Manage user accounts (admin)**

```
=====
Best Auto Part
=====

Main menu -> Manage login -> Manage user accounts
-----
                                         0.00 LKR

        Manage user accounts

        Username      Password
1  #Suresh          78780
2  #Rizni           88
3  Raja             ac2c
4  Jiva              2121

Enter user username : Raja
Enter user password : ac2c

Manage login menu

1. Change Password
2. Delete Account

0. Back

Enter the option :
```

Figure 70 manage user accounts of admin

- **Delete account**

```
=====
Best Auto Part
=====

Main menu -> Manage login -> Delete Account
-----
                                         0.00 LKR

        Delete Account

Confirm :
    Once you deleted you are no longer authorized to be a user of this
    Do you want to delete (y/n) : y
Account deleted successfully
Press any key to continue . . .


```

Figure 71 delete account

- **Sign up**

```
=====
Best Auto Part
=====

Main menu -> Signup                                         0.00 LKR

Create Account

Enter username : Raja
Enter password : ac2c56v
Confirm password : ac2c56v
Account created successfully
Press any key to continue . . .
```

*Figure 72 sign up*

- **Manage services & products**

```
=====
Best Auto Part
=====

Main menu -> Manage services & products                         0.00 LKR

Manage services & products

Services:
1. Add Services
2. Delete Services
3. Update Services

products:
4. Add products
5. Delete products
6. Update products

0. Back

Enter the option :
```

*Figure 73 manage services & products*

- **Services:**

- a. Add services

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Add Services
-----
0.00 LKR

Add Services

Enter service name : Break repair
Enter service Price : 400
Service added successfully
Press any key to continue . . .
```

*Figure 74 add services*

- b. Delete services

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Delete Services
-----
0.00 LKR

Delete Services

Enter service number : 17
Confirm :
    Once you deleted, 'Check suspension system,650 LKR' no longer available to be a user of this
    Do you want to delete (y/n) : y
Deletion process successed
Press any key to continue . . .
```

*Figure 75 delete services*

### c. Update services

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Update Services
-----
                                         0.00 LKR
Update Services

Enter service number : 15
Confirm :
    Are you going to update ' Tire change,800 LKR ' (y/n) : y
Enter service name : Tire change
Enter service Price : 850
Service updated successfully
Press any key to continue . . .
```

Figure 76 update services

- **Products:**

- a. Add products

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Add Products
-----
                                         0.00 LKR
Add Products

Enter Product name : break oil
Enter Product Price : 540
Product added successfully
Press any key to continue . . .
```

Figure 77 add products

## b. Delete products

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Delete Products
-----
0.00 LKR

Delete Products

Enter Products number : 17
Confirm :
    Once you deleted, 'clutch plate,17200 LKR' no longer available to be a user of this
        Do you want to delete (y/n) : y
Deletion process successed
Press any key to continue . . .
```

Figure 78 delete products

## c. Update products

```
=====
Best Auto Part
=====

Main menu -> Manage services & products -> Update Products
-----
0.00 LKR

Update Products

Enter Product number : 18
Confirm :
    Are you going to update ' tire,23450 LKR ' (y/n) : y
Enter Product name : Tire
Enter Product Price : 25000
Product updated successfully
Press any key to continue . . .
```

Figure 79 update products

- **Available Services (guest)**

```
=====
Best Auto Part
=====

Main menu -> Available Service
-----
0.00 LKR

Available Service

1. List Services
2. Search Services

0. Back

Enter the option :
```

*Figure 80 available services of guest*

- **Available Services (user and admin)**

```
=====
Best Auto Part
=====

Main menu -> Available Service
-----
0.00 LKR

Available Service

1. List Services
2. Search Services
3. Buy

0. Back

Enter the option :
```

*Figure 81 available services of user and admin*

- **List Services**

List Services		0.00 LKR
	Name	Price
1	Full engine repair	32500 LKR (3-month warranty)
2	Air Filter	950 LKR
3	Oil change	4550 LKR (5 liters)
4	Wheel Alignment	1200 LKR
5	Battery Check	2800 LKR
6	Check tire air pressure and fill	100 LKR (N2/Tire)
7	Body wash	2500 LKR
8	Check light system	650 LKR
9	Clean power steering pump	2200 LKR
10	Clean break system	1200 LKR
11	Clean cooling system	3200 LKR
12	Change clutch plate (Manual)	12250 LKR
13	Emission test	750 LKR
14	Vehicle polish	6700 LKR
15	Tire change	800 LKR
16	Rear excel bearing	1300 LKR
17	Check suspension system	650 LKR

*Figure 82 list services*

- **Search Services**

```
=====
Best Auto Part
=====

Main menu -> Available Service -> Search Service
-----
                                         0.00 LKR
          Search Service

Enter Service name : Air Filter

Name           Price
1  Air Filter           950 LKR

Press any key to continue . . .
```

Figure 83 search services

- **Buy (user and admin)**

```
=====
Best Auto Part
=====

Main menu -> Available Service -> Buy Service
-----
                                         0.00 LKR
          Buy Service

Enter Service number : 2
Enter the quantity : 3

Service      : Air Filter
Unit price   : 950.00
Quantity     : 3
Price        : 2850.00

Successfully added to cart

Do you want to buy more Service? (y/n) :
```

Figure 84 buy service of user and admin

- **Available Products (guest)**

```
=====
Best Auto Part
=====

Main menu -> Available Product
-----
                                         0.00 LKR

Available Product

1. List Product
2. Search Product

0. Back

Enter the option : Invalid option
Press any key to continue . . .
```

Figure 85 available products of guest

- **Available Products (user and admin)**

```
=====
Best Auto Part
=====

Main menu -> Available Product
-----
                                         2850.00 LKR

Available Product

1. List Product
2. Search Product
3. Buy

0. Back

Enter the option :
```

Figure 86 available Products of user and admin

- List Product

List Products		0.00 LKR
	Name	Price
1	engine oil	5850 LKR
2	battery water	80 LKR (per liter)
3	headlight	2000 LKR
4	signal light	700 LKR
5	break light	900 LKR
6	power steering oil	1150 LKR
7	power steering seal full set	1300 LKR
8	break oil	520 LKR
9	break shoe	2850 LKR
10	break washer	800 LKR
11	master pump busher set	2300 LKR
12	disk plate	1500 LKR
13	break switch	750 LKR
14	abs sensor	3200 LKR
15	speedometer sensor	2950 LKR
16	rpm sensor	4200 LKR
17	clutch plate	17200 LKR
18	gladiator cap	1100 LKR
19	tire	23450 LKR
20	alloy wheel	22700 LKR
21	bearing	8000 LKR
22	oil filter	1950 LKR
23	air filter	3300 LKR

Press any key to continue . . .

Figure 87 list products

- Search Product

```
=====
Best Auto Part
=====

Main menu -> Available Product -> Search Product
-----
                                         0.00 LKR
          Search Product

Enter Product name : engine oil

Name           Price
1   engine oil           5850 LKR

Press any key to continue . . .

```

*Figure 88 search products*

- Buy (user and admin)

```
=====
Best Auto Part
=====

Main menu -> Available Product -> Buy Product
-----
                                         0.00 LKR
          Buy Product

Enter Product number : 4
Enter the quantity : 2

Product      : signal light
Unit price   : 700.00
Quantity     : 2
Price        : 1400.00

Successfully added to cart

Do you want to buy more Product? (y/n) :

```

*Figure 89 buy product of user and admin*

- **Cart (user and admin)**

```
=====
Best Auto Part
=====

Main menu -> Cart -> Bill
-----
                                         44620.00 LKR
        Bill

        List Services

        Name          Price
1 Full engine repair      32500
2 Wheel Alignment         1200
3 Battery Check           2800

        List Services

        Name          Price
1 battery water            320
2 power steering seal full set    7800

Customer name : Jiva
Total Price : 44620.00

        Thank you!

Press any key to continue . . .
```

*Figure 90 cart of user and admin*

# **Development**

## **Introduction**

In order to connect clients' needs with the relevant technical solutions, software engineers engage in the software development life cycle. Thus, processes to provide specific functions are systematically created.

(Software ‘Developer’ or ‘Engineer’: What’s the Difference? — SheCanCode, 2021)

## **Development environment**

The Hardware and Software configurations that were used for system development have been categorized in the Table 1

Hardware specification	Software specification
System Model - Aspire A515-54G	Microsoft Windows 10 Home Single Language
RAM 8.00 GB	VS Code V.1.55.2(user setup)
1TB HDD + 256 SSD	Dev C++ V.5.11
Nvidia MX-250 2GB	TDM-GCC Compiler V.4.9.2
	C++ V.11

*Table 1 Implantation environment*

## **System Development Tools and Technologies**

### **DEV C++**

Dev C++ IDE V.5.11 is used to develop entire program in C++.

### **VS Code**

VS Code IDE V.1.55.2(user setup) is used to test entire program in C++.

### **TDM-GCC Compiler**

TDM-GCC Compiler V.4.9.2 is used for compiling codes and execute the program.

## C++ V.11

C++ V.11 programming language is used to code entire program

### Hierarchical Function Structure

Hierarchical Function Structure is used to indicate the function groups which helps to understand the code structure.

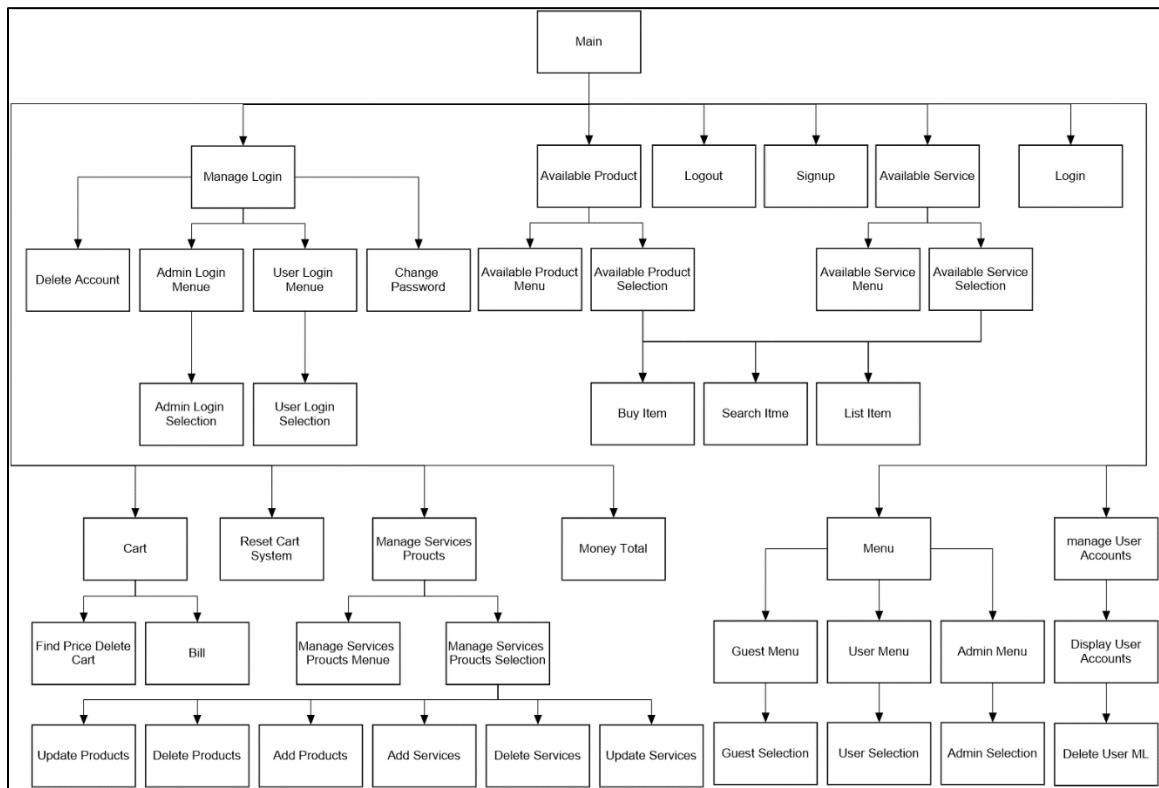


Figure 91 Basic hierarchical Function Structure

## Code Structure

This section describes all major code segments that have been implemented during the Implementation phase to make the intended functionalities of the project a workable and productive solution. Visit below link for all code

[https://drive.google.com/drive/folders/1LB9X\\_2W0LRUbmnOuZNUEfG52zfzMzbEM?usp=sharing](https://drive.google.com/drive/folders/1LB9X_2W0LRUbmnOuZNUEfG52zfzMzbEM?usp=sharing)

## Main requirements

- Manage available services and products (add, delete, update)
  - Products
    - Add

```
void Add_Products()
{
    string Product, price;
    Header("Add Products", 1);
    cout << "\nAdd Products\n\n";

    ifstream infile("Product\\Product.txt");

    if (infile)
    {

        ofstream outfile("Product\\Product.txt", ios::app);

        cout << "Enter Product name : ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        getline(cin,Product);

        cout << "Enter Product Price : ";
        cin >> price;

        outfile << Product << "," << price << " LKR" << endl;
        outfile.close();
        cout << "Product added successfully\n";
    }
    else
    {
        cout << "Unable to access data file\n";
    }

    system("PAUSE");
    Header("Add Products", 0);
}
```

Figure 92 add products

- Delete

```

void Delete_Products()
{
    Header("Delete Products", 1);

    cout << "\n\tDelete Products\n\n";

    int pNum = 0;
    cout << "Enter Products number : ";
    cin >> pNum;

    string line[100], catchLine;
    int i = 0, j = 0;
    ifstream infile("Product\\Product.txt");

    while (getline(infile, line[i]))
    {
        if ((pNum - 1) == i)
        {
            catchLine = line[i];
        }
        i++;
    }
    infile.close();

    if (pNum > i)
    {
        cout << "Invalid product number\n";
    }
    else
    {

        cout << "Confirm : \n"
            << " Once you deleted, \' " << catchLine << "' no longer available to be a user of this\n"
            << " Do you want to delete (y/n) : ";

        char yn;
        cin >> yn;

        if (yn == 'y')
        {
            {
                delete _S_P("Product\\Product.txt", (pNum - 1));
            }
            cout << "Deletion process successed\n";
        }
        else if (yn == 'n')
        {
            cout << "Deletion process cancelled\n";
        }
        else
        {
            cout << "invalid option\n";
        }
    }

    system("PAUSE");
    Header("Delete Products", 0);
}

```

Figure 93 delete products

- Update

```

    void Update_Products()
    {
        Header("Update Products", 1);

        cout << "\n\tUpdate Products\n\n";

        int pNum = 0;
        cout << "Enter Product number : ";
        cin >> pNum;

        string line[100], catchLine;
        int i = 0, j = 0;
        ifstream infile("Product\\Product.txt");

        while (getline(infile, line[i]))
        {
            if ((pNum - 1) == i)
            {
                catchLine = line[i];
                line[i] = "";
            }
            i++;
        }
        infile.close();

        if (pNum > i)
        {
            cout << "Invalid product number\n";
        }
        else
        {
            cout << "Confirm : \n"
                << " Are you going to update \' " << catchLine << " \' (y/n) : ";

            char yn;
            cin >> yn;

            if (yn == 'y')
            {
                delete_S_P("Product\\Product.txt", (pNum - 1));

                string Product, price;
                ofstream outfile("Product\\Product.txt", ios::app);
                ;

                cout << "Enter Product name : ";
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                getline(cin, Product);

                cout << "Enter Product Price : ";
                cin >> price;

                outfile << Product << "," << price << " LKR" << endl;
                outfile.close();
                cout << "Product updated successfully\n";
            }
            else if (yn == 'n')
            {
                cout << "Deletion process cancelled\n";
            }
            else
            {
                cout << "invalid option\n";
            }
        }

        system("PAUSE");
        Header("Update Products", 0);
    }
}

```

Figure 94 update products

- Services

- Add

```
void Add_Services()
{
    string service, price;
    Header("Add Services", 1);
    cout << "\nAdd Services\n\n";

    ifstream infile("Services\\service.txt");

    if (infile)
    {
        ofstream outfile("Services\\service.txt", ios::app);

        cout << "Enter service name : ";
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        getline(cin, service);

        cout << "Enter service Price : ";
        cin >> price;

        outfile << service << "," << price << " LKR" << endl;
        outfile.close();
        cout << "Service added successfully\n";
    }
    else
    {
        cout << "Unable to access data file\n";
    }

    system("PAUSE");
    Header("Add Services", 0);
}
```

Figure 95 add services

- Delete

```

    ~ void Delete_Services()
    {
        Header("Delete Services", 1);

        cout << "\n\tDelete Services\n\n";

        int sNum = 0;
        cout << "Enter service number : ";
        cin >> sNum;

        string line[100], catchLine;
        int i = 0, j = 0;
        ifstream infile("Services\\service.txt");

        while (getline(infile, line[i]))
        {
            if ((sNum - 1) == i)
            {
                catchLine = line[i];
            }
            i++;
        }
        infile.close();

        if (sNum > i)
        {
            cout << "invalid number\n";
        }
        else
        {
            cout << "Confirm : \n"
                << " Once you deleted, \" " << catchLine << "\' no longer available to be a user of this\n"
                << " Do you want to delete (y/n) : ";

            char yn;
            cin >> yn;

            if (yn == 'y')
            {
                {
                    delete_S_P("Services\\service.txt", (sNum - 1));
                }
                cout << "Deletion process successed\n";
            }
            else if (yn == 'n')
            {
                cout << "Deletion process cancelled\n";
            }
            else
            {
                cout << "invalid option\n";
            }
        }

        system("PAUSE");
        Header("Delete Services", 0);
    }
}

```

Figure 96 delete services

- Update

```

void Update_Services()
{
    Header("Update Services", 1);

    cout << "\n\tUpdate Services\n\n";

    int sNum = 0;
    cout << "Enter service number : ";
    cin >> sNum;

    string line[100], catchLine;
    int i = 0, j = 0;
    ifstream infile("Services\\service.txt");

    while (getline(infile, line[i]))
    {

        if ((sNum - 1) == i)
        {
            catchLine = line[i];
        }
        i++;
    }
    infile.close();

    if (sNum > i)
    {
        cout << "Invalid service number\n";
    }
    else
    {

        cout << "Confirm : \n"
        << " Are you going to update \' " << catchLine << " \' (y/n) : ";

        char yn;
        cin >> yn;

        if (yn == 'y')
        {
            {
                delete_S_P("Services\\service.txt", (sNum - 1));

                string service, price;
                ofstream outfile("Services\\service.txt", ios::app);
                ;

                cout << "Enter service name : ";
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                getline(cin, service);

                cout << "Enter service Price : ";
                cin >> price;

                outfile << service << "," << price << " LKR" << endl;
                outfile.close();
                cout << "Service updated successfully\n";
            }
        }
        else if (yn == 'n')
        {
            cout << "Deletion process cancelled\n";
        }
        else
        {
            cout << "invalid option\n";
        }
    }

    system("PAUSE");
    Header("Update Services", 0);
}

```

Figure 97 update services

- List available services and products.

```

void List_Items(string path, string itemName)
{
    cout << "\n\t\tList " << itemName << "\n\n";
    string line[100], SANDP[200];
    int n = 0;

    ifstream infile(path);

    while (getline(infile, line[n]))
    {
        n++;
    }
    infile.close();

    int i = 0;
    for (int j = 0; j < n; j++)
    {
        long long int pos = 0;
        string delimiter = ",";
        while ((pos = line[j].find(delimiter)) != string::npos)
        {
            SANDP[i] = line[j].substr(0, pos);
            line[j].erase(0, pos + delimiter.length());
            SANDP[i + 1] = line[j].substr(0, pos + 10);
            i += 2;
        }
    }

    int z = 0;
    cout << "\tName \t\t\tPrice"
    | << "\n\n";
    for (int i = 0; i < (n * 2); i += 2)
    {
        cout << left << setw(2) << ++z << " " << setw(32) << left << SANDP[i] << "\t" << setw(32) << SANDP[i + 1] << "\n";
    }

    cout << endl;
}

```

*Figure 98 list available services and products*

- Search specific product or service.

```

void Search_Items(string itemName, string itemPath)
{
    Header("Search "+itemName, 1);

    cout << "\n\t\t Search "+itemName+"\n\n";

    string line[10], SSP[20], Sname;
    bool isHas = false;
    int n = 0;

    cout << "Enter " << itemName << " name : ";
    cin >> Sname;

    ifstream infile(itemPath);
    for (int i = 0; i < 10; i++)
    {
        while (!infile.fail())
        {
            getline(infile, line[i]);
            if ((line[i].find(Sname)) != (string::npos))
            {
                isHas = true;
                n++;
                break;
            }
        }
    }
    infile.close();

    if (isHas)
    {
        int pos = 0, i = 0;
        string UP[2], delimiter = ",";
        for (int j = 0; j < n; j++)
        {
            while ((pos = line[j].find(delimiter)) != string::npos)
            {
                SSP[i] = line[j].substr(0, pos);
                line[j].erase(0, pos + delimiter.length());
                SSP[i + 1] = line[j].substr(0, pos + 10);
                i += 2;
            }
        }

        int z = 0;
        cout << endl
            << setw(20) << "Name" << setw(28) << "Price"
            << "\n\n";
        for (int i = 0; i < (n * 2); i += 2)
        {
            cout << left << setw(2) << ++z << " " << setw(32) << left << SSP[i] << "\t" << setw(32) << SSP[i + 1] << "\n";
        }
    }
    else
    {
        cout << "\n"<< itemName << "not found, make sure search word is correct by checking in "<< itemName << " list\n";
    }

    cout << endl;
    system("PAUSE");
    Header("Search "+itemName, 0);
}

```

*Figure 99 search specific product or service*

## Other requirements

- User login

```
void login()
{
    string line;
    bool valid = false, isHas = false;

    Header("Login", 1);

    cout << "\n\tLogin\n\n";

    cout << "Enter username : ";
    cin >> username;

    cout << "Enter password : ";
    cin >> password;

    ifstream infile("Login\\" + username + ".txt");
    while (!infile.fail())
    {
        getline(infile, line);
        if ((line.find(username)) != string::npos)
        {
            isHas = true;
            break;
        }
    }
    infile.close();

    if (isHas)
    {
        int pos = 0;
        string UP[2], delimiter = ",";
        while ((pos = line.find(delimiter)) != string::npos)
        {
            UP[0] = line.substr(0, pos);
            line.erase(0, pos + delimiter.length());
            UP[1] = line.substr(0, pos);
        }

        for (int i = 0; i < 2; i++)
        {
            if (username == UP[i] && password == UP[i + 1])
            {
                valid = true;
                break;
            }
        }
    }

    if (valid)
    {
        cout << "\nAccess Granted...\n";
        loginStatus = true;
        if (username.find("#") != string::npos)
        {
            loginAdminStatus = 1;
        }
        else
        {
            loginAdminStatus = 0;
        }
    }
    else
    {
        cout << "\nAccess Denied...Invalid Password.\n\n";
        loginStatus = false;
    }
}
else
{
    cout << "\nAccess Denied...Account not found\n\n";
    system("PAUSE");
    Header("Login", 0);
}
```

Figure 100 user login

- Logout

```
void logout()
{
    loginStatus = false;
    loginAdminStatus = false;
    resetCartSystem();
    menu();
}
```

Figure 101 logout

- Exit

```
case 0:
    system("CLS");
    exit(0);
    break;
```

Figure 102 exit

- Headers

```
//headers
#include <iostream>
#include <limits>
#include <iomanip>
#include <fstream>
#include <string>
using namespace std;
```

*Figure 103 headers*

- Function prototyping

```
//Function Prototyping
void Header(string, int);
void menu();
void guestmenu();
void guestSelection(int);
void usermenu();
void userSelection(int);
void adminmenu();
void adminSelection(int);
void Manage_user_accounts();
void display_user_accounts();
void deleteUserML();
void availableService();
void Available_Service_menu();
bool Available_Service_Selection(int);
void availableProduct();
void Available_Product_menu();
bool Available_Product_Selection(int);
void List_Items(string,string);
void buy_Item(string,string,string);
void Search_Items(string,string);
void login();
void signup();
void logout();
void Manage_login();
void admin_login_menu();
bool admin_login_Selection(int);
void user_login_menu();
bool user_login_Selection(int);
void changePassword();
bool deleteAccount();
void Manage_Services_products();
void Manage_Services_products_menu();
bool Manage_Services_products_Selection(int);
void Add_Services();
void Delete_Services();
void Update_Services();
void Add_Products();
void Delete_Products();
void Update_Products();
void delete_S_P(string,int);
void cart();
float find_price_delete_cart(string, int);
void bill();
void money_total(int, bool, bool);
void resetCartSystem();
```

Figure 104 function prototyping

- Global Variable

```
//Global Variable
bool loginStatus = 0;
bool loginAdminStatus = 0;
string password = "";
string username = "";
string UP[2];
string path = "Main menu";
float total = 0;
```

Figure 105 global Variable

- Main

```
int main()
{
    resetCartSystem();
    menu();
    return 0;
}
```

Figure 106 main

- Reset cart system

```

    < void resetCartSystem()
    {
        ofstream sCartFile("Cart\\cart_S.txt");
        ofstream pCartFile("Cart\\cart_P.txt");
        sCartFile.close();
        pCartFile.close();
        total = 0;
    }
}

```

Figure 107 reset cart system

- Header

```

void Header(string ipath = "", int status = -1)
{
    system("CLS");
    cout << "\t\t=====\n"
        << "\t\tBest Auto Part\n"
        << "\t\t=====\n\n";

    if (ipath != "" && status == 1)
    {
        path += " -> " + ipath;
    }
    else if (status == 0)
    {
        ipath = " -> " + ipath;
        path.replace(path.find(ipath), ipath.length(), "");
    }

    cout << path << "\n-----\n";
    money_total(0, 1, 1);
}

```

Figure 108 header

- Money total

*Figure 109 money total*

- Menu

```
▽ void menu()
{
    ▽ while (true)
    {
        Header();

        if (loginStatus)
            (loginAdminStatus) ? adminmenu() : usermenu();
        else
            guestmenu();

        int option = 0;
        cout << "\nEnter the option : ";
        cin >> option;
        if (!cin)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "invalid option\n";
            system("PAUSE");
        }
        else
        {
            if (loginStatus)
                (loginAdminStatus) ? adminSelection(option) : userSelection(option);
            else
                guestSelection(option);
        }
    }
}
```

Figure 110 menu

- Guest menu

```

< void guestmenu()
{
    cout << "\n\tGuest\n\n";
    cout << "1. Available Services\n"
        |<< "2. Available products\n";
    cout << "\n3. Login\n"
        |<< "4. Signup\n"
        |<< "0. Exit\n";
}

```

*Figure 111 guest menu*

- Guest selection

```

< void guestSelection(int option)
{
    switch (option)
    {
        case 1:
            availableService();
            break;
        case 2:
            availableProduct();
            break;
        case 3:
            login();
            break;
        case 4:
            signup();
            break;
        case 0:
            system("CLS");
            exit(0);
            break;
        default:
            cout << "invalid option\n"
                |<< endl;
            system("PAUSE");
    }
}

```

*Figure 112 guest selection*

- Available service

```

void availableService()
{
    bool stop = 0;
    while (!stop)
    {
        Header("Available Service", 1);
        cout << "\n\tAvailable Service\n";

        Available_Service_menu();

        int option = 0;
        cout << "\nEnter the option : ";
        cin >> option;
        if (!cin)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Invalid option\n";
            system("PAUSE");
        }
        else
        {
            stop = Available_Service_Selection(option);
        }

        Header("Available Service", 0);
    }
}

```

Figure 113 available service

- Available service menu

```

void Available_Service_menu()
{
    cout << "\n1. List Services\n"
        | << "2. Search Services\n";

    if (loginStatus || loginAdminStatus)
    {
        cout << "3. Buy\n";
    }

    cout << "\n0. Back\n";
}

```

Figure 114 available service menu

- Available service selection

```

bool Available_Service_Selection(int option)
{
    switch (option)
    {
        case 1:
            Header("List Services", 1);
            List_Items("Services\\service.txt", "Services");
            system("PAUSE");
            Header("List Services", 0);
            break;
        case 2:
            Search_Items("Service", "Services\\service.txt");
            break;
        case 0:
            return 1;
            break;
        default:
            if ((option == 3) && (loginStatus || loginAdminStatus))
            {
                buy_Item("Service", "Services\\service.txt", "cart\\cart_S.txt");
            }
            else
            {
                cout << "invalid option\n";
                system("PAUSE");
            }
    }
    return 0;
}

```

*Figure 115 available service selection*

- Buy item

```

void buy_item(string itemName, string itemPath, string cartPath)
{
    while (1)
    {
        Header("Buy " + itemName, 1);

        cout << "\n\n\tBuy " << itemName << "\n\n";

        int shNum = 0, i = 0, j = 0, quantity = 0;
        string line[100], catchline, service;
        float sprice = 0;

        cout << "Enter " << itemName << " number : ";
        cin >> shNum;

        ifstream inFile(itemPath);

        while (getline(inFile, line[i]))
        {
            if ((shNum - 1) == i)
            {
                catchLine = line[i];
                break;
            }
            i++;
        }
        inFile.close();

        if (catchLine != "")
        {
            long long int pos = 0;
            string delimiter = ",";
            while ((pos = catchLine.find(delimiter)) != string::npos)
            {
                service = catchLine.substr(0, pos);
                catchLine.erase(0, pos + delimiter.length());
                sprice = stof(catchLine.substr(0, catchLine.size() - 4));
            }

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

            cout << "\n" << itemName << " : " << service
            << "\nUnit price : " << sprice
            << "\nQuantity : " << quantity
            << "\nPrice : " << sprice * quantity;

            {
                ofstream outfile(cartPath, ios::app);
                outfile << service << "," << sprice * quantity << endl;
                outfile.close();
            }

            cout << "\n\n Successfully added to cart\n";
            money_total(sprice * quantity, 1, 0);
        }
        else
        {
            cout << "Invalid " << itemName << " number\n";
        }
    }

    char yn;
    bool askReturn = true;
    cout << "\nDo you want to buy more " << itemName << "? (y/n) : ";
    cin >> yn;

    while (askReturn)
    {
        if (yn == 'n')
        {
            askReturn = false;
            break;
        }
        else if (yn == 'y')
        {
            Header("Buy " + itemName, 0);
            break;
        }
        else
        {
            cout << "Invalid option\n";
            askReturn = false;
        }
    }

    if (!askReturn)
    {
        system("PAUSE");
        Header("Buy " + itemName, 0);
        break;
    }
}
}

```

Figure 116 buy item

- Available product

```
void availableProduct()
{
    bool stop = 0;
    while (!stop)
    {
        Header("Available Product", 1);
        cout << "\n\tAvailable Product\n";

        Available_Product_menu();

        int option = 0;
        cout << "\nEnter the option : ";
        cin >> option;
        if (!cin)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Invalid option\n";
            system("PAUSE");
        }
        else
        {
            stop = Available_Product_Selection(option);
        }
        Header("Available Product", 0);
    }
}
```

Figure 117 available product

- Available product menu

```

void Available_Product_menu()
{
    cout << "\n1. List Product\n"
        | << "2. Search Product\n";

    if (loginStatus || loginAdminStatus)
    {
        cout << "3. Buy\n";
    }

    cout << "\n0. Back\n";
}

```

*Figure 118 available product menu*

- Available product selection

```

bool Available_Product_Selection(int option)
{
    switch (option)
    {
    case 1:
        Header("List Product", 1);
        List_Items("Product\\Product.txt","Products");
        system("PAUSE");
        Header("List Product", 0);
        break;
    case 2:
        Search_Items("Product","Product\\Product.txt");
        break;
    case 0:
        return 1;
        break;
    default:
        if ((option == 3) && (loginStatus || loginAdminStatus))
        {
            buy_Item("Product","Product\\Product.txt","cart\\cart_P.txt");
        }
        else
        {
            cout << "invalid option\n";
            system("PAUSE");
        }
    }
    return 0;
}

```

*Figure 119 available product selection*

- Admin menu

```
void adminmenu()
{
    if (loginAdminStatus)
    {
        cout << "\n\tWelcome admin " << username << "\n\n";
    }
    else
    {
        cout << "\n\tWelcome " << username << "\n\n";
    }

    cout << "1. Available Services\n"
        << "2. Available products\n"
        << "3. Manage services & products\n"
        << "4. Manage login\n"
        << "5. Cart\n";

    cout << "\n6. Logout\n"
        << "0. Exit\n";
}
```

Figure 120 admin menu

- Admin selection

```
void adminSelection(int option)
{
    switch (option)
    {
        case 1:
            availableService();
            break;
        case 2:
            availableProduct();
            break;
        case 3:
            Manage_Services_products();
            break;
        case 4:
            Manage_login();
            break;
        case 5:
            cart();
            break;
        case 6:
            logout();
            break;
        case 0:
            system("CLS");
            exit(0);
            break;
        default:
            cout << "invalid option\n";
            system("PAUSE");
    }
}
```

Figure 121 admin selection

- Manage services products

```

void Manage_Services_products()
{
    bool stop = 0;
    while (!stop)
    {
        Header("Manage services & products", 1);
        cout << "\n\tManage services & products\n";

        Manage_Services_products_menu();

        int option = 0;
        cout << "\nEnter the option : ";
        cin >> option;
        if (!cin)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Invalid option\n";
            system("PAUSE");
        }
        else
        {
            stop = Manage_Services_products_Selection(option);
        }

        Header("Manage services & products", 0);
    }
}

```

Figure 122 manage services products

- Manage services products menu

```

void Manage_Services_products_menu()
{
    cout << "\nServices:\n"
        << "    1. Add Services\n"
        << "    2. Delete Services\n"
        << "    3. Update Services\n";

    cout << "\nproducts:\n"
        << "    4. Add products\n"
        << "    5. Delete products\n"
        << "    6. Update products\n";

    cout << "\n0. Back\n";
}

```

*Figure 123 manage services products menu*

- Manage services products selection

```

bool Manage_Services_products_Selection(int option)
{
    switch (option)
    {
    case 1:
        Add_Services();
        break;
    case 2:
        Delete_Services();
        break;
    case 3:
        Update_Services();
        break;
    case 4:
        Add_Products();
        break;
    case 5:
        Delete_Products();
        break;
    case 6:
        Update_Products();
        break;
    case 0:
        return 1;
        break;
    default:
        cout << "invalid option\n";
        system("PAUSE");
    }
    return 0;
}

```

*Figure 124 manage services products selection*

- Core function of delete service and product

```

void delete_S_P(string path, int sNum)
{
    string line[100];
    int i = 0, j = 0;
    ifstream infile(path);

    while (getline(infile, line[i]))
    {
        if ((sNum) == i)
        {
            line[i] = "";
        }
        i++;
    }
    infile.close();

    ofstream outfile(path);
    ;
    while (i > j)
    {
        if (line[j] != "")
        {
            outfile << line[j] << endl;
        }
        j++;
    }
    outfile.close();
}

```

Figure 125 core function of delete service and product

- User menu

```

void usermenu()
{
    if (loginAdminStatus)
    {
        cout << "\n\tWelcome admin " << username << "\n\n";
    }
    else
    {
        cout << "\n\tWelcome " << username << "\n\n";
    }

    cout << "1. Available Services\n"
        << "2. Available products\n"
        << "3. Manage login\n"
        << "4. Cart\n";

    cout << "\n5. Logout\n"
        << "0. Exit\n";
}

```

*Figure 126 user menu*

- User selection

```

void userSelection(int option)
{
    switch (option)
    {
        case 1:
            availableService();
            break;
        case 2:
            availableProduct();
            break;
        case 3:
            Manage_login();
            break;
        case 4:
            cart();
            break;
        case 5:
            logout();
            break;
        case 0:
            system("CLS");
            exit(0);
            break;
        default:
            cout << "invalid option\n";
            system("PAUSE");
    }
}

```

*Figure 127 user selection*

- Sign up

```

void signup()
{
    Header("Signup", 1);
    cout << "\nCreate Account\n\n";

    cout << "Enter username : ";
    cin >> username;

    ifstream infile("Login\\" + username + ".txt");

    if (!infile)
    {
        cout << "Enter password : ";
        cin >> password;
        UP[1] = password;
        cout << "Confirm password : ";
        cin >> password;

        if (UP[1] == password)
        {

            ofstream outfile("Login\\" + username + ".txt");
            outfile << username << "," << password;
            outfile.close();

            ofstream masterFile("Login\\LMaster.txt", ios::app);
            masterFile << username << "," << password << endl;
            masterFile.close();

            cout << "Account created successfully\n";
        }
        else
        {
            cout << "Password mismatched\n";
        }
    }
    else
    {
        cout << "Username already taken\n";
    }

    system("PAUSE");
    Header("Signup", 0);
}

```

Figure 128 sign up

- Manage login

```

void Manage_login()
{
    bool stop = 0;
    while (!stop)
    {
        Header("Manage login", 1);
        cout << "\n\tLogin\n";

        if (loginAdminStatus)
            admin_login_menu();
        else
            user_login_menu();

        int option = 0;
        cout << "\nEnter the option : ";
        cin >> option;
        if (!cin)
        {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Invalid option\n";
            system("PAUSE");
        }
        else
        {
            if (loginAdminStatus)
            {
                stop = admin_login_Selection(option);
            }
            else
                stop = user_login_Selection(option);
        }

        Header("Manage login", 0);
    }
}

```

Figure 129 manage login

- Admin login menu

```

void admin_login_menu()
{
    cout << "\nManage login menu\n\n";

    cout << "1. Change Password\n"
        << "2. Manage user accounts\n"
        << "3. Delete Account\n";

    cout << "\n0. Back\n";
}

```

Figure 130 admin login menu

- Admin login selection

```
bool admin_login_Selection(int option)
{
    switch (option)
    {
        case 1:
            changePassword();
            break;
        case 2:
            Manage_user_accounts();
            break;
        case 3:
            return deleteAccount();
            break;
        case 0:
            return 1;
            break;
        default:
            cout << "invalid option\n";
            system("PAUSE");
    }
    return 0;
}
```

Figure 131 admin login selection

- User login menu

```
void user_login_menu()
{
    cout << "\nManage login menu\n\n";

    cout << "1. Change Password\n"
        | << "2. Delete Account\n";

    cout << "\n0. Back\n";
}
```

Figure 132 user login menu

- User login selection

```
bool user_login_Selection(int option)
{
    switch (option)
    {
        case 1:
            changePassword();
            break;
        case 2:
            return deleteAccount();
            break;
        case 0:
            return 1;
            break;
        default:
            cout << "invalid option\n";
    }
    return 0;
}
```

Figure 133 user login selection

- Change password

```

void changePassword()
{
    Header("Change Password", 1);

    cout << "\nChange Password\n\n";
    UP[1] = password;
    cout << "Enter old password : ";
    cin >> password;

    if (UP[1] == password)
    {
        cout << "Enter new password : ";
        cin >> UP[1];
        {
            deleteUserML();
            ofstream outfile("Login\\LMaster.txt", ios::app);
            outfile << username << "," << UP[1] << endl;
            outfile.close();
        }
        ofstream outfile("Login\\" + username + ".txt");
        outfile << username << "," << UP[1];
        outfile.close();

        cout << "Password changed successfully\n";
    }
    else
    {
        cout << "Invalid Password\n";
    }
    system("PAUSE");
    Header("Change Password", 0);
}

```

Figure 134 change password

- Delete account

```
✓ bool deleteAccount()
{
    Header("Delete Account", 1);

    cout << "\n\tDelete Account\n\n";

    cout << "Confirm : \n"
        << "    Once you deleted you are no longer authorized to be a user of this\n"
        << "    Do you want to delete (y/n) : ";

    char yn;
    cin >> yn;

    if (yn == 'y')
    {
        const string fileName = "Login\\" + username + ".txt";
        remove(fileName.c_str());
        deleteUserML();
        cout << "Account deleted successfully\n";

        loginStatus = 0;
        system("PAUSE");
        Header("Delete Account", 0);
        return 1;
    }
    else if (yn == 'n')
    {
        cout << "Deletion process cancelled\n";
    }
    else
    {
        cout << "invalid option\n";
    }

    system("PAUSE");
    Header("Delete Account", 0);
    return 0;
}
```

Figure 135 delete account

- Delete user in master login

```

    < void deleteUserML()
    {
        string line[100],lineCheck;
        int i = 0, j = 0;
        ifstream infile("Login\\LMaster.txt");

        lineCheck = username + "," + password;

        while (getline(infile, line[i]))
        {
            if (lineCheck == line[i])
            {
                line[i] = "";
            }
            i++;
        }
        infile.close();

        ofstream outfile("Login\\LMaster.txt");
        while (i > j)
        {
            if (line[j] != "")
            {
                outfile << line[j] << endl;
            }
            j++;
        }
        outfile.close();
    }

```

Figure 136 delete user in master login

- Manage user accounts

```
void Manage_user_accounts()
{
    bool stop = 0;
    while (!stop)
    {
        Header("Manage user accounts", 1);

        cout << "\n\tManage user accounts\n\n";

        display_user_accounts();

        cout << "\nEnter user username : ";
        cin >> username;

        ifstream infilet("Login\\" + username + ".txt");

        if (infilet)
        {
            cout << "Enter user password : ";
            cin >> password;

            user_login_menu();

            int option = 0;
            cout << "\nEnter the option : ";
            cin >> option;
            if (!cin)
            {
                cin.clear();
                cin.ignore(numeric_limits<streamsize>::max(), '\n');
                cout << "Invalid option\n";
                system("PAUSE");
            }
            else
            {
                stop = user_login_Selection(option);
            }
        }
        else
        {
            cout << "Account not found\n";
        }

        Header("Manage user accounts", 0);
    }
}
```

Figure 137 manage user accounts

- Display user accounts

```

void display_user_accounts()
{
    string line[100], DUA[200];
    int n = 0;

    ifstream infile("Login\\LMaster.txt");

    while (getline(infile, line[n]))
    {
        n++;
    }
    infile.close();

    int i = 0;
    for (int j = 0; j < n; j++)
    {
        int pos = 0;
        string delimiter = ",";
        while ((pos = line[j].find(delimiter)) != string::npos)
        {
            DUA[i] = line[j].substr(0, pos);
            line[j].erase(0, pos + delimiter.length());
            DUA[i + 1] = line[j].substr(0, pos);
            i += 2;
        }
    }

    int z = 0;
    cout << "    Username\tPassword \n";
    for (int i = 0; i < n * 2; i += 2)
    {
        cout << setw(3) << ++z << " " << DUA[i] << "\t\t" << DUA[i + 1] << "\n";
    }
}

```

*Figure 138 display user accounts*

- Cart

```

void cart()
{
    int sNum;
    char yn;

    string sline = "", pline = "";

    ifstream sinfile("Cart\\cart_S.txt");
    getline(sinfile, sline);
    sinfile.close();

    ifstream pinfile("Cart\\cart_P.txt");
    getline(pinfile, pline);
    pinfile.close();

    if ((sline != "") || (pline != ""))
    {
        Header("Cart", 1);
        if (sline != "")
        {
            do
            {
                Header("Cart", 0);
                Header("Cart", 1);

                cout << "\n\t\t\t Cart\n";

                List_Items("Cart\\cart_S.txt","Services");

                cout << "Do you want to remove any service from the cart? (y/n) : ";
                cin >> yn;

                if (yn == 'y')
                {
                    cout << "\nEnter the service number : ";
                    cin >> sNum;
                    money_total(find_price_delete_cart("Cart\\cart_S.txt", sNum), 0, 0);
                    delete_S_P("Cart\\cart_S.txt", (sNum - 1));
                    cout << "Item removed\n";
                    system("PAUSE");
                }
            } while (yn == 'y');
        }

        if (pline != "")
        {
            do
            {
                Header("Cart", 0);
                Header("Cart", 1);
                cout << "\n\t\t\t Cart\n";
                List_Items("Cart\\cart_S.txt","Services");

                List_Items("Cart\\cart_P.txt","Products");

                cout << "Do you want to remove any product from the cart? (y/n) : ";
                cin >> yn;

                if (yn == 'y')
                {
                    cout << "\nEnter the product number : ";
                    cin >> sNum;
                    money_total(find_price_delete_cart("Cart\\cart_P.txt", sNum), 0, 0);
                    delete_S_P("Cart\\cart_P.txt", (sNum - 1));
                    cout << "Item removed\n";
                    system("PAUSE");
                }
            } while (yn == 'y');
        }

        bill();
        Header("Cart", 0);
    }
    else
    {
        cout << "\t\t Cart is empty \n";
        system("PAUSE");
    }
}

```

Figure 139 cart

- Bill

```

    < void bill()
    {
        Header("Bill", 1);

        cout << "\n\t\t\t Bill\n";

        List_Items("Cart\cart_S.txt","Services");
        List_Items("Cart\cart_P.txt","Services");

        cout << setw(16) << "Customer name : " << username << "\n";
        cout << setw(16) << "Total Price  :" << total << "\n";

        cout << "\n\t\tThank you!\n\n";

        resetCartSystem();

        system("Pause");
        Header("Bill", 0);
    }

```

Figure 140 bill

- Find the price of item in the cart to delete

```

    < float find_price_delete_cart(string path, int num)
    {
        string line[50], SANDP[100];
        int n = 0;

        ifstream infile(path);

        while (getline(infile, line[n]))
        {
            n++;
        }
        infile.close();

        int i = 0;
        for (int j = 0; j < n; j++)
        {
            long long int pos = 0;
            string delimiter = ",";
            while ((pos = line[j].find(delimiter)) != string::npos)
            [
                SANDP[i] = line[j].substr(0, pos);
                line[j].erase(0, pos + delimiter.length());
                SANDP[i + 1] = line[j].substr(0, pos + 10);
                i += 2;
            ]
        }
        if (num > n)
            return 0.00;
        else
            return stof(SANDP[(2 * num) - 1]);
    }

```

Figure 141 find the price of item in the cart to delete

## **Validation**

- Page - Main menu

In the main menu, user need to pick an option from displaying menu list. Otherwise, it will throw “Invalid option” message in the display
- Page – Available service

In the Available service menu, user need to pick an option from displaying menu list. Otherwise, it will throw “Invalid option” message in the display
- Page – Search service

In the search service page, user need to enter correct service name or part of name for searching. Otherwise, it will throw “Service not found, make sure search word is correct by checking in Service list” message in the display
- Page – Buy service

In the buy service page, user need to enter correct service number for select service to buy. Otherwise, it will throw “Invalid Service number” message in the display

For answering “Do you want to buy more Service? (y/n) : ” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display
- Page – Available product

In the Available product menu, user need to pick an option from displaying menu list. Otherwise, it will throw “Invalid option” message in the display
- Page – Search product

In the search product page, user need to enter correct product name or part of name for searching. Otherwise, it will throw “Product not found, make sure search word is correct by checking in Product list” message in the display
- Page – Buy product

In the buy product page, user need to enter correct product number for select product to buy. Otherwise, it will throw “Invalid Product number” message in the display

For answering “Do you want to buy more Product? (y/n) : ” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display

- Page - Manage Services & Products

In the Available product menu, user need to pick an option from displaying menu list. Otherwise, it will throw “Invalid option” message in the display

- Page – Delete service

In the delete service, user need to enter correct service number for select service to delete. Otherwise, it will throw “Invalid number” message in the display

- Page – Delete product

In the delete product page, user need to enter correct product number for select product to delete. Otherwise, it will throw “Invalid number” message in the display

- Page – Update services

In the Update Services page, user need to enter correct service number for select service to update. Otherwise, it will throw “Invalid service number” message in the display

For answering “Are you going to update ' <Name>, <Price> LKR ' (y/n) : ” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display

- Page – Update product

In the Update product page, user need to enter correct product number for select product to update. Otherwise, it will throw “Invalid product number” message in the display

For answering “Are you going to update ' <Name>, <Price> LKR ' (y/n) : ” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display

- Page - Manage login

In the Manage login menu, user need to pick an option from displaying menu list. Otherwise, it will throw “Invalid option” message in the display

- Page - Change password

In the change password page, user need to enter correct old password first. Otherwise, it will throw “Invalid password” message in the display

- Page - Delete account

In the delete account page, for answering “Once you deleted you are no longer authorized to be a user of this system. Do you want to delete (y/n) :” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display

- Page - Manage user accounts

In the Manage user accounts page, user need to enter correct username from list which that need to manage. Otherwise, it won’t accept any data.

- Page - Cart

In the cart page, for answering “Do you want to remove any service from the cart? (y/n) :” and “Do you want to remove any product from the cart? (y/n) :” this question, user need to type ‘y’ or ‘n’. Otherwise, it will throw “Invalid option” message in the display

## **Evaluation**

### **Test plan – IEEE 829**

#### **Test Plan Identifier**

BAP\_001

#### **References**

Software Requirement Specification

#### **Introduction**

This is the master plan for the best auto mart software. In this plan will addressed those are related to vehicle repairing process of best auto mart. The main aim of this plan is ensuring that new computerized system is giving the same level out of current manual system.

This project consists of three level testing plans. Those are unit testing, integrated/system testing and user acceptance testing

The time allocation for this testing phase is 2 months. The user acceptance testing will contain 1 month after installation process on best auto part company

#### **Test Items**

The test items are:

- a. Navigation header
- b. Program exit
- c. Login
- d. Logout
- e. Menu
- f. Signup
- g. Program exit
- h. List service
- i. Search service
- j. Buy service
- k. List product
- l. Search product
- m. Buy product
- n. Add service
- o. Delete service
- p. update service
- q. Add product
- r. Delete product
- s. Update product
- t. Change password
- u. Delete account
- v. List user accounts
- w. Cart

## **Software Risk Issues**

Safety and access to the database must be specified and verified.

## **Features to be tested**

- |                    |                        |
|--------------------|------------------------|
| 1. Login           | 12. Add service        |
| 2. Logout          | 13. Delete service     |
| 3. Menu            | 14. update service     |
| 4. Signup          | 15. Add product        |
| 5. Program exit    | 16. Delete product     |
| 6. List service    | 17. update product     |
| 7. Search service  | 18. Change password    |
| 8. Buy service     | 19. Delete account     |
| 9. List product    | 20. List user accounts |
| 10. Search product | 21. cart               |
| 11. Buy product    |                        |

## **Features not to be Tested**

1. Navigation header  
It's low-level risk and non-functional area
2. Program exit  
Exit program usually possible and it's handled by operating system

## **Approach**

A test will include the Unit, System/Integration (combined) and Acceptance tests for the BEST AUTO MART project. It is hoped that for system/integration tests there would be at least one full-time independent test worker. However, it is possible to do most research by the developer with budget restrictions and time limits.

**Unit testing:** The developer will perform UNIT testing, and the programmer must have proof of unit testing (test case list, sample output, data printouts, defect information).

**Integration testing:** integration/system tests are carried out by programmer

User acceptance testing. This project does not have specialized test tools available. After all critical errors have been repaired, programs come into the system/integration

test. A software will have up to two major faults until the testing of the program is impeded.

**User acceptance testing:** Acceptance The actual end-users with the developer team leader will conduct testing. The acceptance test will take place for a period of one month after the completion of the System/Integration testing in conjunction with the current manual system operation.

### **Item Pass/Fail Criteria**

The program should operate as required to achieve all its main features, and the pass percentage of test cases must exceed 95% and no essential bugs must occur.

### **Suspension Criteria and Resumption Requirements**

If all of the main features are not working or system login problems, the test should be suspended.

### **Test Deliverables**

1. Plan for Acceptance Testing
2. Test plan System/Integration
3. Documentation of the unit test schedules and turnover
4. Prototypes for screens
5. Mock-up's study
6. Reports and summaries of defect/incident
7. Turnover statistics and test records

### **Remaining Test Tasks**

<b>TASK</b>	<b>Assigned To</b>	<b>Status</b>
Plan for Acceptance Testing	Tester	100%
Test plan System/Integration	Tester	100%
Documentation of the unit test schedules and turnover	Tester	100%

Reports and summaries of defect/incident	Tester	100%
--	--------	------

*Table 2 Remaining test task*

### **Environmental Needs**

8. Laptop: intel i5 processor, 10<sup>th</sup> generation, 8gb ram, 2gb nvidia graphics, 100gb disk space
9. Software: VS Code, notepad, chrome, MS word, MS excel
10. Internet access

### **Staffing and Training Needs**

It is preferred that there will be at least one (1) full time tester allocated to the project for the system/integration and acceptance testing phases of the project. This will entail appointment of an individual part time at the beginning of the project to assist in assessments etc... And they will be assigned full time about two months into the project.

### **Responsibilities**

Test plan, Preparation and execution of tests should be carried out by tester.

### **Schedule**

Start date : 01.03.2021

End date : 30.04.2021

Perform test execution of perform test – man-hours 120,

Reporting the test – man-hours 30

### **Planning Risks and Contingencies**

The expense can exceed in the event of an erroneous budget estimate. Contingency Plan – Set the scope before test tasks begin and be aware of the project planning and also constantly monitor budgetary estimates.

### **Approvals**

The project manager should negotiate on project completion and decide the next steps

(Test plan- IEEE829, 2021)

## Test case

A test case for software engineering is a specified input, running conditions, test procedures and anticipated outcomes defining a single test to be performed to accomplish a certain purpose for software testing, such as the exercise of a certain curriculum course or verification of compliance with a particular requirement.

(Test case - Wikipedia, 2021)

### 1. Login

SCREEN NAME	Login page				Best Auto Mart					
PREPARED BY	MN Rizni Mohamed									
DATE	01-03-2021									
SCENARIO ID	LP_001	SCENARIO DESCRIPTION			Test login processing function for admin					
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION				
1	LAT_001	Test login response on correct username but wrong password	Must install Best Auto Mart software. Have login database	username: #asd password: ad	Access Denied. Invalid Password.	-				
2	LAT_002	Test login response on wrong username but correct password fields		username:#asdd password:s	Access Denied. Account not found					
3	LAT_003	Test login response on wrong username but wrong password fields		username:#asdd password:ss	Access Denied. Account not found					
4	LAT_004	Test login response on correct username and correct password		username: password:	Access Granted					
SCENARIO ID	LP_002	SCENARIO DESCRIPTION			Test login processing function for user					
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION				
1	LUT_001	Test login response on correct username but wrong password	Must install Best Auto Mart software. Have login database	username: 1 password: 123	Access Denied. Invalid Password.	-				
2	LUT_002	Test login response on wrong username but correct password		username:123 password:2	Access Denied. Account not found					
3	LUT_003	Test login response on wrong username and wrong password		username:123 password:123	Access Denied. Account not found					
4	LUT_004	Test login response on correct username but correct password		username:1 password:2	Access Granted					

Table 3 Test case Login

No	Test case id	Actual result	Status	Defect id
1	LAT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR Enter username : #asd Enter password : ad Access Denied...Invalid Password.  Press any key to continue . . . ■</pre>	Pass	-
2	LAT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR Enter username : #asdd Enter password : s Access Denied...Account not found  Press any key to continue . . . ■</pre>	Pass	-
3	LAT_003	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR Enter username : #asdd Enter password : ss Access Denied...Account not found  Press any key to continue . . . ■</pre>	Pass	
4	LAT_004	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR Enter username : #asd Enter password : s Access Granted... Press any key to continue . . . ■</pre>	Pass	

Table 4 Test case login - admin results

No	Test case id	Actual result	Status	Defect id
1	LUT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR  Enter username : 1 Enter password : 123  Access Denied...Invalid Password.  Press any key to continue . . . ■</pre>	Pass	-
2	LUT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR  Enter username : 123 Enter password : 2  Access Denied...Account not found  Press any key to continue . . . ■</pre>	Pass	-
3	LUT_003	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR  Enter username : 123 Enter password : 123  Access Denied...Account not found  Press any key to continue . . . ■</pre>	Pass	
4	LUT_004	<pre>===== Best Auto Part =====  Main Menue -&gt; Login -----           Login                               0.00 LKR  Enter username : 1 Enter password : 2  Access Granted... Press any key to continue . . . ■</pre>	Pass	

Table 5 Test case login - user results

## 2. Logout

SCREEN NAME	Logout page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	01-03-2021						
SCENARIO ID	LO_001	SCENARIO DESCRIPTION			Test logout processing function for admin		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	LOAT_001	Test logout response on menu page	Must install Best Auto Mart software and need to be login	Select 6th option in admin menu	Direct to guest menu page	-	
SCENARIO ID	LO_002	SCENARIO DESCRIPTION			Test logout processing function for user		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	LOUT_001	Test logout response on menu page	Must install Best Auto Mart software and need to be login	Select 5th option in admin menu	Direct to guest menu page	-	

Table 6 Test case logout

No	Test case id	Actual result	Status	Defect id
1	LOAT_001	<pre>===== Best Auto Part =====  Main Menue ----- Guest   0.00 LKR  1. Available Services 2. Available Proucts  3. Login 4. Signup 0. Exit  Enter the option :  </pre>	Pass	-

Table 7 Test case logout - admin results

No	Test case id	Actual result	Status	Defect id
1	LOUT_001	<pre>===== Best Auto Part =====  Main Menue ----- Guest   0.00 LKR  1. Available Services 2. Available Proucts  3. Login 4. Signup 0. Exit  Enter the option :  </pre>	Pass	-

Table 8 Test case logout - user results

### 3. Menu

SCREEN NAME	Menu page				Best Auto Mart	
PREPARED BY	MN Rizni Mohamed					
DATE	02-03-2021					
SCENARIO ID	M_001	SCENARIO DESCRIPTION			Test menu page displaying for admin	
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	MAT_001	Test admin menu response on after admin login	Must install Best Auto Mart software and need admin to be login	-	1. Available Services 2. Available Products 3. Manage Services & Products 4. Manage login 5. Cart 6. Logout 0. Exit Enter the option :	-
SCENARIO ID	M_002	SCENARIO DESCRIPTION			Test menu page displaying for user	
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	MUT_001	Test user menu response on after user login	Must install Best Auto Mart software and need user to be login	-	1. Available Services 2. Available Products 3. Manage login 4. Cart 5. Logout 0. Exit Enter the option :	-
SCENARIO ID	M_003	SCENARIO DESCRIPTION			Test menu page displaying for guest	
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	MGT_001	Test guest menu response on program started	Must install Best Auto Mart software	-	1. Available Services 2. Available Products 3. Login 4. Signup 0. Exit Enter the option :	-

Table 9 Test case menu

No	Test case id	Actual result	Status	Defect id
1	MAT_001	<pre>===== Best Auto Part =====  Main Menue ----- Welcome admin #asd                                     0.00 LKR  1. Available Services 2. Available Products 3. Manage Services &amp; Products 4. Manage login 5. Cart 6. Logout 0. Exit  Enter the option :  </pre>	Pass	-

Table 10 Test case menu - admin

No	Test case id	Actual result	Status	Defect id
1	MUT_001	<pre> ===== Best Auto Part =====  Main Menue ----- Welcome Rizni                               0.00 LKR  1. Available Services 2. Available Proucts 3. Manage login 4. Cart  5. Logout 0. Exit  Enter the option : </pre>	Pass	-

Table 11 Test case menu - user

No	Test case id	Actual result	Status	Defect id
1	MGT_001	<pre> ===== Best Auto Part =====  Main Menue ----- Guest   0.00 LKR  1. Available Services 2. Available Proucts  3. Login 4. Signup 0. Exit  Enter the option : </pre>	Pass	-

Table 12 Test case menu - guest

#### 4. Signup

SCREEN NAME	Signin page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	02-03-2021						
SCENARIO ID	SI_001	SCENARIO DESCRIPTION			Test signin processing function for admin		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	SIAT_001	Test signin response on correct username but wrong confirm password	Must install Best Auto Mart software. Have login database	username:#qwee password: ad con.pass:add	Password mismatched	direct to guest page	
2	SIAT_002	Test signin response on already existing username		username:#asd password: con.pass:	Username already taken		
3	SIAT_003	Test signin response on correct username and correct confirm password		username:#asdd password: add con.pass:add	Account created successfully		
SCENARIO ID	SI_002	SCENARIO DESCRIPTION			Test signin processing function for user		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	SIUT_001	Test signin response on correct username but wrong confirm password	Must install Best Auto Mart software. Have login database	username:qwee password: ad con.pass:add	Password mismatched	direct to guest page	
2	SIUT_002	Test signin response on already existing username		username:1 password: con.pass:	Username already taken		
3	SIUT_003	Test signin response on correct username and correct confirm password		username:asdd password: add con.pass:add	Account created successfully		

Table 13 Test case signup

No	Test case id	Actual result	Status	Defect id
1	SIAT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Signup ----- Create Account   0.00 LKR  Enter username : #qwee Enter password : ad Confirm password : add Password mismatched Press any key to continue . . . ■</pre>	Pass	-
2	SIAT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Signup ----- Create Account   0.00 LKR  Enter username : #asd Username already taken Press any key to continue . . . ■</pre>	Pass	-
3	SIAT_003		Pass	

```

=====
Best Auto Part
=====

Main Menue -> Signup
-----
Create Account           0.00 LKR

Enter username : #asdd
Enter password : add
Confirm password : add
Account created successfully
Press any key to continue . . .

```

Table 14 Test case signup - admin

No	Test case id	Actual result	Status	Defect id
1	SIUT _001	<pre> ===== Best Auto Part =====  Main Menue -&gt; Signup ----- Create Account           0.00 LKR  Enter username : qwee Enter password : ad Confirm password : add Password mismatched Press any key to continue . . . </pre>	Pass	-
2	SIUT _002	<pre> ----- =====  Best Auto Part =====  Main Menue -&gt; Signup ----- Create Account           0.00 LKR  Enter username : 1 Username already taken Press any key to continue . . . </pre>	Pass	-
3	SIUT _003	<pre> ----- =====  Best Auto Part =====  Main Menue -&gt; Signup ----- Create Account           0.00 LKR  Enter username : asdd Enter password : add Confirm password : add Account created successfully Press any key to continue . . . </pre>	Pass	

Table 15 Test case signup - user

## 5. Program exit

SCREEN NAME	Exit page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	03-03-2021						
SCENARIO ID	Ex_001	SCENARIO DESCRIPTION			Test exit processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	ET_001	Test exit on menu page	Must install Best Auto Mart software	Select 0th option in admin menu	direct to system terminal	-	

Table 16 Test case program exit

No	Test case id	Actual result	Status	Defect id
1	ET_001	PS C:\Users\user\Desktop\c++ VS code> █	Pass	-

Table 17 Test case program exit result

## 6. List service

SCREEN NAME	Service list view page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	03-03-2021						
SCENARIO ID	SLV_001	SCENARIO DESCRIPTION			Test list service view processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	SLVT_001	Test service list view on available service page	Must install Best Auto Mart software. Need services data store	Select 1st option in available service page	List all services	-	

Table 18 Test case list service

No	Test case id	Actual result	Status	Defect id
1	LVT_001	<pre> ===== Best Auto Part =====  Main Menue -&gt; Available Service -&gt; List Services ----- List Services   0.00 LKR  Name  Price 1 Full engine repair      32500 LKR (3-month warranty) 2 Air Filter                950 LKR 3 Oil change                 4550 LKR (5 liters) 4 Wheel Alignment            1200 LKR 5 Battery Check              2800 LKR 6 Check tire air pressure and fill    100 LKR (N2/Tire) 7 Body wash                  2500 LKR 8 Check light system          650 LKR 9 Clean power steering pump    2200 LKR 10 Clean break system         1200 LKR 11 Clean cooling system       3200 LKR 12 Change clutch plate (Manual) 12250 LKR 13 Emission test              750 LKR 14 Vehicle polish             6700 LKR 15 Tire change                 800 LKR 16 Rear excel bearing          1300 LKR 17 Check suspension system     650 LKR 18 bvn                         bvn LKR  Press any key to continue . . . </pre>	Pass	-

Table 19 Test case list view result

## 7. Search service

SCREEN NAME	Search services page			Best Auto Mart	
PREPARED BY	MN Rizni Mohamed				
DATE	03-03-2021				
SCENARIO ID	SS_001	SCENARIO DESCRIPTION	Test search services processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
1	SST_001	Test searching for services by services words on available service page with right key word	Must install Best Auto Mart software. Need services data store	Enter service name : Check	1 Battery Check 2800 LKR 2 Check tire air pressure and fill 100 LKR (N2/Tire) 3 Check light system 650 LKR 4 Check suspension system 650 LKR
2	SST_002	Test searching for services by services words on available service page with wrong key word	Must install Best Auto Mart software. Need services data store	Enter service name : asdf	Service not found, make sure search word is correct by checking in service list

Table 20 Test case search service

No	Test case id	Actual result	Status	Defect id
1	SST_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Service -&gt; Search Services ----- Search Services   0.00 LKR  Enter service name : Check  Name          Price 1 Battery check      2800 LKR 2 Check tire air pressure and fill    100 LKR (N2/Tire) 3 Check light system     650 LKR 4 Check suspension system    650 LKR  Press any key to continue . . . ■</pre>	Pass	-
2	SST_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Service -&gt; Search Services ----- Search Services   3640.00 LKR  Enter service name : asdf  Service not found, make sure search word is correct by checking in service list  Press any key to continue . . . ■</pre>	Pass	-

Table 21 Test case search service results

## 8. Buy service

SCREEN NAME	Buy services page			Best Auto Mart		
PREPARED BY	MN Rizni Mohamed					
DATE	03-03-2021					
SCENARIO ID	BS_001	SCENARIO DESCRIPTION	Test buy services processing function			
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	BST_001	Test buying for services by services ID on available service page with correct id number	Must install Best Auto Mart software. Need services data store. Admin has to be login in	Enter service number : 8  Quantity: 2	Service : Check light system Unit price : 650.00 Quantity : 2 Price : 1300.00  Successfully added to cart	-
2	BST_002	Test buying for services by services ID on available service page with wrong id number	Must install Best Auto Mart software. Need services data store. Admin has to be login in	Enter service number : 20  Quantity:	Invalid service number	-

Table 22 Test case buy service

No	Test case id	Actual result	Status	Defect id
1	BST_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Service -&gt; Buy Service ----- Buy Service   0.00 LKR  Enter service number : 8 Enter the quantity : 2  Service      : check light system Unit price   : 650.00 Quantity     : 2 Price        : 1300.00  Successfully added to cart  Do you want to buy more services? (y/n) : y Press any key to continue . . . █</pre>	Pass	-
2	BST_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Service -&gt; Buy Service ----- Buy Service   0.00 LKR  Enter service number : 20 Invalid service number  Do you want to buy more services? (y/n) : █</pre>	Pass	-

Table 23 Test case buy service result

## 9. List product

SCREEN NAME	product list view page				Best Auto Mart	
PREPARED BY	MN Rizni Mohamed					
DATE	04-03-2021					
SCENARIO ID	PLV_001	SCENARIO DESCRIPTION		Test list product view processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	PLVT_001	Test product list view on available product page	Must install Best Auto Mart software. Need products data store	Select 1st option in available product page	List all products	-

Table 24 Test case list product

No	Test case id	Actual result	Status	Defect id
1	PLVT _001	<pre> ===== Best Auto Part =====  Main Menue -&gt; Available Product -&gt; List Product ----- List Product   2600.00 LKR        Name          Price 1 engine oil           5850 LKR 2 battery water        80 LKR (per liter) 3 headlight            2000 LKR 4 signal light         700 LKR 5 break light          900 LKR 6 power steering oli   1150 LKR 7 power steering seal full set 1300 LKR 8 break oil             520 LKR 9 break shoe            2850 LKR 10 break washer         800 LKR 11 master pump busher set 2300 LKR 12 disk plate          1500 LKR 13 break switch         750 LKR 14 abs sensor           3200 LKR 15 speedometer sensor   2950 LKR 16 rpm sensor           4200 LKR 17 clutch plate         17200 LKR 18 gladiator cap       1100 LKR 19 tire                  23450 LKR 20 alloy wheel          22700 LKR 21 bearing              8000 LKR 22 oil filter            1950 LKR 23 air filter            3300 LKR 24 zxc                  ZXC LKR  Press any key to continue . . . </pre>	Pass	-

Table 25 Test case list product result

## 10. Search product

SCREEN NAME	Search products page			Best Auto Mart	
PREPARED BY	MN Rizni Mohamed				
DATE	04-03-2021				
SCENARIO ID	SP_001	SCENARIO DESCRIPTION	Test search products processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT
1	SPT_001	Test searching for products by products words on available product page with right key word	Must install Best Auto Mart software. Need products data store	Enter product name : break	1 break light 900 LKR 2 break oil 520 LKR 3 break shoe 2850 LKR 4 break washer 800 LKR 5 break switch 750 LKR
2	SPT_002	Test searching for products by products words on available product page with wrong key word	Must install Best Auto Mart software. Need products data store	Enter product name : asdd	Product not found, make sure search word is correct by checking in service list

Table 26 Test case search product

No	Test case id	Actual result	Status	Defect id
1	SPT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Product -&gt; Search Product ----- Search Product   2600.00 LKR  Enter Product name : break Name          Price 1  break light      900 LKR 2  break oil        520 LKR 3  break shoe       2850 LKR 4  break washer     800 LKR 5  break switch    750 LKR  Press any key to continue . . . </pre>	Pass	-
2	SPT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Product -&gt; Search Product ----- Search Product   0.00 LKR  Enter Product name : asdd Product not found, make sure search word is correct by checking in service list Press any key to continue . . . </pre>	Pass	-

Table 27 Test case search product results

## 11. Buy product

SCREEN NAME	Buy products page				Best Auto Mart					
PREPARED BY	MN Rizni Mohamed									
DATE	04-03-2021									
SCENARIO ID	BP_001	SCENARIO DESCRIPTION		Test buy products processing function						
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION				
1	BPT_001	Test buying for products by products ID on available product page with correct id	Must install Best Auto Mart software. Need products data store. Admin has to be login in	Enter product number : 8  Quantity: 2	Enter Product number : 8 Enter the quantity : 2  Product : break oil Unit price : 520.00 Quantity : 2 Price : 1040.00  Successfully added to cart	-				
2	BPT_002	Test buying for products by products ID on available product page with wrong id	Must install Best Auto Mart software. Need products data store. Admin has to be login in	Enter product number : 30  Quantity:	Invalid Product number	-				

Table 28 Test case buy product

No	Test case id	Actual result	Status	Defect id
1	BPT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Product -&gt; Buy Product ----- Buy Product   2600.00 LKR  Enter Product number : 8 Enter the quantity : 2  Product      : break oil Unit price   : 520.00 Quantity     : 2 Price        : 1040.00  Successfully added to cart  Do you want to buy more Products? (y/n) : █</pre>	Pass	-
2	BPT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Available Product -&gt; Buy Product ----- Buy Product   0.00 LKR  Enter Product number : 30 Invalid Product number  Do you want to buy more Products? (y/n) : █</pre>	Pass	-

Table 29 Test case buy product result

## 12. Add service

SCREEN NAME		Add services page		Best Auto Mart		
PREPARED BY		MN Rizni Mohamed				
DATE		05-03-2021				
SCENARIO ID	AS_001	SCENARIO DESCRIPTION	Test add service processing function			
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	AST_001	Test add services response on add service page.	Must install Best Auto Mart software. Need service data store. Admin has to be login in	Enter service number : item Quantity: 200	Enter service name : item Enter service Price : 200 Service added successfully	-

Table 30 Test case add service

No	Test case id	Actual result	Status	Defect id
1	AST_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Procuts -&gt; Add Services ----- Add Services   3640.00 LKR  Enter service name : item Enter service Price : 200 Service added successfully Press any key to continue . . . █</pre>	Pass	-

Table 31 Test case add service result

### 13. Delete service

SCREEN NAME	Delete services page			Best Auto Mart		
PREPARED BY	MN Rizni Mohamed					
DATE	05-03-2021					
SCENARIO ID	DS_001	SCENARIO DESCRIPTION			Test delete service processing function	
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	DST_001	Test add services response on add service page. Enter correct service number	Must install Best Auto Mart software. Need service data store. Admin has to be login in	Enter service number : 19	Deletion process successed	-
2	DST_002	Test add services response on add service page. Enter wrong service number	Must install Best Auto Mart software. Need service data store. Admin has to be login in	Enter service number : 25	Invalid number	-

Table 32 Test case Delete service

No	Test case id	Actual result	Status	Defect id
1	DST_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Products -&gt; Delete Services ----- Delete Services ----- 0.00 LKR  Enter service number : 19 Confirm : Once you deleted, 'item,200 LKR' no longer available to be a user of this Do you want to delete (y/n) : y Deletion process successed Press any key to continue . . .</pre>	Pass	-
2	DST_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Products -&gt; Delete Services ----- Delete Services ----- 0.00 LKR  Enter service number : 25 invalid number Press any key to continue . . .</pre>	Pass	-

Table 33 Test case Delete service results

## 14. Update service

SCREEN NAME	Update services page			Best Auto Mart		
PREPARED BY	MN Rizni Mohamed					
DATE	05-03-2021					
SCENARIO ID	US_001	SCENARIO DESCRIPTION			Test update service processing function	
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	UST_001	Test update services response on add service page. Enter correct service number	Must install Best Auto Mart software. Need service data store. Admin has to be login in	Enter service number : 19 Enter service name : Item 2 Enter service price : 500	Service updated successfully	-
2	UST_002	Test add services response on add service page. Enter wrong service number	Must install Best Auto Mart software. Need service data store. Admin has to be login in	Enter service number : 25	Invalid service number	-

Table 34 Test case Update service

No	Test case id	Actual result	Status	Defect id
1	UST_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Products -&gt; Update Services ----- Update Services   0.00 LKR Enter service number : 19 Confirm : Are you going to update ' item 1,300 LKR ' (y/n) : y Enter service name : item 2 Enter service Price : 500 Service updated successfully Press any key to continue . . . [</pre>	Pass	-
2	UST_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Products -&gt; Update Services ----- Update Services   0.00 LKR Enter service number : 25 Invalid service number Press any key to continue . . . [</pre>	Pass	-

Table 35 Test case Update service result

## 15. Add product

SCREEN NAME	Add products page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	06-03-2021						
SCENARIO ID	AP_001	SCENARIO DESCRIPTION	Test add product processing function				
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	APT_001	Test add products response on add product page.	Must install Best Auto Mart software. Need product data store. Admin has to be login in	Enter product number : item	Enter product name : item	-	
				Quantity: 200	Enter product Price : 200 product added successfully		

Table 36 Test case add product

No	Test case id	Actual result	Status	Defect id
1	APT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Proucts -&gt; Add Products ----- Add Products   0.00 LKR  Enter Product name : item Enter Product Price : 200 Product added successfully Press any key to continue . . .</pre>	Pass	-

Table 37 Test case add product

## 16. Delete product

SCREEN NAME	Delete products page				Best Auto Mart		
PREPARED BY	MN Rizni Mohamed						
DATE	05-03-2021						
SCENARIO ID	DP_001	SCENARIO DESCRIPTION	Test delete product processing function				
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION	
1	DPT_001	Test add products response on add product page. Enter correct product number	Must install Best Auto Mart software. Need product data store. Admin has to be login in	Enter product number : 25	Deletion process successed	-	
				Enter product number : 30	Invalid number		

Table 38 Test case delete product

No	Test case id	Actual result	Status	Defect id
1	DPT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Proucts -&gt; Delete Products ----- Delete Products   0.00 LKR  Enter Products number : 25 Confirm : Once you deleted, 'item,500 LKR' no longer available to be a user o f this Do you want to delete (y/n) : y Deletion process successed Press any key to continue . . . ■</pre>	Pass	-
2	DPT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Proucts -&gt; Delete Products ----- Delete Products   0.00 LKR  Enter Products number : 30 Invalid product number Press any key to continue . . . ■</pre>	Pass	-

Table 39 Test case delete product

## 17. Update product

SCREEN NAME		Update products page		Best Auto Mart		
PREPARED BY		MN Rizni Mohamed				
DATE		05-03-2021				
SCENARIO ID	UP_001	SCENARIO DESCRIPTION		Test update product processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	UST_001	Test update products response on add product page. Enter correct product number	Must install Best Auto Mart software. Need product data store. Admin has to be login in	Enter product number : 25 Enter product name : Item Enter product price : 500	product updated successfully	-
2	UST_002	Test add products response on add product page. Enter wrong product number	Must install Best Auto Mart software. Need product data store. Admin has to be login in	Enter product number : 30	Invalid product number	-

Table 40 Test case update product

No	Test case id	Actual result	Status	Defect id
1	UPT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Proucts -&gt; Update Products ----- Update Products   0.00 LKR  Enter Product number : 25 Confirm :     Are you going to update ' ,item LKR ' (y/n) : y Enter Product name : item Enter Product Price : 500 Product updated successfully Press any key to continue . . . ■</pre>	Pass	-
2	UPT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage Services Proucts -&gt; Update Products ----- Update Products   0.00 LKR  Enter Product number : 30 Invalid product number Press any key to continue . . . ■</pre>	Pass	-

Table 41 Test case update product results

## 18. Change password

SCREEN NAME	change password page			Best Auto Mart	
PREPARED BY	MN Rizni Mohamed				
DATE	06-03-2021				
SCENARIO ID	CP_001	SCENARIO DESCRIPTION	Test change password processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	
1	CPT_001	Test change account password response on manage account page. Enter correct old password	Must install Best Auto Mart software. Need login data store. Admin or user has to be login in	Enter old password : v Enter new password : m	
2	CPT_002	Test change account password response on manage account page. Enter wrong old password	Must install Best Auto Mart software. Need login data store. Admin or user has to be login in	Enter old password : c Enter new password :	

Table 42 Test case change password

No	Test case id	Actual result	Status	Defect id
1	CPT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage login -&gt; Change Password ----- 0.00 LKR  Change Password  Enter old password : v Enter new password : m Password changed successfully Press any key to continue . . . ■</pre>	Pass	-
2	CPT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage login -&gt; Change Password ----- 0.00 LKR  Change Password  Enter old password : c Invalid Password Press any key to continue . . . ■</pre>	Pass	-

Table 43 Test case change password result

## 19. Delete account

SCREEN NAME	Delete account page			Best Auto Mart		
PREPARED BY	MN Rizni Mohamed					
DATE	06-03-2021					
SCENARIO ID	DA_001	SCENARIO DESCRIPTION	Test Delete account processing function			
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	DAT_001	Test delete account response on manage account page.	Must install Best Auto Mart software. Need login data store. Admin or user has to be login in	Do you want to delete (y/n) : y	Account deleted successfully	Redirect to guest page

Table 44 Test case delete account

No	Test case id	Actual result	Status	Defect id
1	DAT_001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage login -&gt; Delete Account ----- 0.00 LKR  Delete Account  Confirm : Once you deleted you are no longer authorized to be a user of this Do you want to delete (y/n) : y Account deleted successfully Press any key to continue . . . ■</pre>	Pass	-

Table 45 Test case delete account results

## 20. List user accounts

SCREEN NAME	List user account page				Best Auto Mart					
PREPARED BY	MN Rizni Mohamed									
DATE	06-03-2021									
SCENARIO ID	LUA_001	SCENARIO DESCRIPTION			Test List user account processing function					
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION				
1	LUT_001	Test List user account response on manage account page.	Must install Best Auto Mart software. Need login data store. Admin has to be login in	select option 2	List all user accounts	-				

Table 46 Test case list user accounts

No	Test case id	Actual result	Status	Defect id
1	LUT _001	<pre>===== Best Auto Part =====  Main Menue -&gt; Manage login -&gt; Manage user accounts ----- 0.00 LKR  Manage user accounts  Username    Password 1 d          e 2 #asd      s 3 2          3 4 a          i 5 1          2 6 5          4 7 g          5 8 #z          x 9 #ad         a 10 Rizni     88 11 #asdd    add 12 asdd     add 13 c          m  Enter user username : </pre>	Pass	-

Table 47 Test case list user accounts results

## 21. Cart

SCREEN NAME	Cart page			Best Auto Mart		
PREPARED BY	MN Rizni Mohamed					
DATE	07-03-2021					
SCENARIO ID	C_001	SCENARIO DESCRIPTION		Test cart processing function		
NO	TEST CASE ID	TEST CASE DESCRIPTION	PRECONDITION	TEST DATA	EXPECTED RESULT	POSTCONDITION
1	CT_001	Test Cart response on main menu page. No items in cart	Must install Best Auto Mart software. Need cart data store. Admin or user has to be login in	select option 5	Cart is empty	-
2	CT_002	Test Cart response on main menu page. Only services. Remove item from card	Must install Best Auto Mart software. Need cart data store. Admin or user has to be login in	Do you want to remove any service from the cart? (y/n) : y	Item removed	-
				Enter service number:1		
3	CT_003	Test Cart response on main menu page. Only services. Didn't remove item from card	Must install Best Auto Mart software. Need cart data store. Admin or user has to be login in	Do you want to remove any service from the cart? (y/n) : n	Show bill	-
5	CT_005	Test Cart response on main menu page. Only product Remove item from card	Must install Best Auto Mart software. Need cart data store. Admin or user has to be login in	Do you want to remove any service from the cart? (y/n) : y	Item removed	-
				Enter service number:2		
6	CT_006	Test Cart response on main menu page. both product and service in the card and did not remove any item from card	Must install Best Auto Mart software. Need cart data store. Admin or user has to be login in	Do you want to remove any service from the cart? (y/n) : n	Show bill	-

Table 48 Test case cart

No	Test case id	Actual result	Status	Defect id
1	CT_001	<pre>===== Best Auto Part =====  Main Menu ----- Welcome admin d   0.00 LKR  1. Available Services 2. Available Products 3. Manage Services &amp; Products 4. Manage login 5. Cart  6. Logout 0. Exit  Enter the option : 5                                     Cart is empty Press any key to continue . . . ■</pre>	Pass	-

2	CT_002	<pre>===== Best Auto Part =====  Main Menue -&gt; Cart ----- Cart List Services Name          Price 1 Clean power steering pump    8800 2 Battery Check             19600  Do you want to remove any service from the cart? (y/n) : y Enter the service number : 1 Item removed Press any key to continue . . . ■</pre>	Pass	-
3	CT_003	<pre>===== Best Auto Part =====  Main Menue -&gt; Cart -&gt; Bill ----- Bill List Services Name          Price 1 Battery Check    5600 2 Air Filter        950  List Product Name          Price  Customer name : d Total Price   : 6550.00  Thank you! Press any key to continue . . . ■</pre>	Pass	-
4	CT_004	<pre>===== Best Auto Part =====  Main Menue -&gt; Cart ----- Cart List Services Name          Price List Product Name          Price 1 signal light      7000  Do you want to remove any product from the cart? (y/n) : y Enter the product number : 1 Item removed Press any key to continue . . . ■</pre>	Pass	-

5	CT_005	<pre>===== Best Auto Part =====  Main Menue -&gt; Cart -&gt; Bill ----- Bill   6650.00 LKR  List Services  Name          Price List Product  Name          Price 1 break light      900 2 power steering oli  5750  Customer name : #asd Total Price   : 6650.00  Thank you!  Press any key to continue . . . ■</pre>	Pass	-
6	CT_006	<pre>===== Best Auto Part =====  Main Menue -&gt; Cart -&gt; Bill ----- Bill   30000.00 LKR  List Services  Name          Price 1 clean break system    12000  List Product  Name          Price 1 break light       18000  Customer name : #asd Total Price   : 30000.00  Thank you!  Press any key to continue . . . ■</pre>	Pass	-

Table 49 Test case cart results

## Test summary report

Test Summary Report (TSR) is an essential deliverable that is generated at the conclusion of or after testing. But a consolidated update on the testing carried out on the project so far is included in the test summary report.

(Test summery report - softwaretestinghelp, 2021)

Test summery report					
Test cycle : System/Intergration					
Executed	Passed		21		
	Failed		0		
	Total test excuted (Passed + failed)		21		
Pending			0		
In progress			0		
Blocked			0		
(Sub total) Test planned			21		
(Pending + In progress + Blocked + Test executed)			21		
Functions	Description	% TCs execututed	% TCs passed	% TCs pending	priority
1. Login	Login user and admin	100	100	0	High
2. Logout	Logout user and admin	100	100	0	High
3. Menu	Different menu for user, admin and guest	100	100	0	High
4. Signup	signup for new admin and user	100	100	0	High
5. Program exit	exiting the program	100	100	0	High
6. List service	list all available service	100	100	0	Medium
7. Search service	search service from available service	100	100	0	High
8. Buy service	buy service from available	100	100	0	High
9. List product	list all available products	100	100	0	Medium
10. Search product	search service from available products	100	100	0	High
11. Buy product	buy product from available	100	100	0	High
12. Add service	add new service to service data file	100	100	0	High
13. Delete service	delete a service from service data file	100	100	0	High
14. update service	update a service of service data file	100	100	0	High
15. Add product	add new product to service data file	100	100	0	High
16. Delete product	delete a product from service data file	100	100	0	High
17. update product	update a product of service data file	100	100	0	High
18. Change password	change user and admin password	100	100	0	High
19. Delete account	delete user and admin account	100	100	0	High
20. List user accounts	list all user accounts with username & password	100	100	0	Medium
21. cart	List cart items, remove item, billing	100	100	0	High

Table 50 Test summery report

## **User Accepting Testing (UAT)**

After completing the implementation work, the system was installed at the clients' location in order to go through the user acceptance testing process. The objective was to perform user acceptance testing with the help from potential users during a certain period of time. The user who has administrative privileges completed end to end testing of the system and some system defects were identified during the testing. Those identified defects were fixed and the fixed issues were successfully got verified by administrative user again. The billing functions are tested by the potential users from accounts department of BAM. The functionalities dedicated for users' level were successfully tested by few users from the BAM company.

At the final stage of the user acceptance testing, the comments and feedback about the new system were collected from users by presenting an evaluation questionnaire form. Some feedback comments received from a user.

## Sample questionnaire

<b>User Acceptance Testing (UAT)</b>					
<b>Project Name</b>	Best Auto Mart				
<b>Testing Start Date / Time</b>	04-04-2021				
<b>Testing End Date / Time</b>	04-04-2.21				
<b>Name of Tester</b>	MN Rizni Mohamed				
<b>Name of User</b>					
Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
<b>Role : Guest</b>					
1	View service list	Open software -> 1 available service -> 1 List service	List services		
2	view product list	Open software -> 2 available product -> 1 List product	List products		
3	search service	Open software -> 1 available service -> 2 search service -> enter service key word	Search services		
4	search product	Open software -> 2 available product -> 2 search product -> enter service key word	Search product		
5	Signup	Open software -> 4 signup -> enter username, password, confirm password	Account created		
<b>Role : User</b>					
1	View service list	Open software -> login -> 1 available service -> 1 List service	List services		
2	view product list	Open software -> login -> 2 available product -> 1 List product	List products		

Figure 142 UAT sample questionnaire page 1

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
3	search service	Open software -> login -> 1 available service -> 2 search service -> enter service key word	Search services		
4	search product	Open software -> login -> 2 available product -> 2 search product -> enter service key word	Search product		
5	Buy product	Open software -> login -> 2 available product -> 3 buy -> enter product id -> enter quantity	add product to cart		
6	Buy service	Open software -> login -> 1 available service -> 3 buy -> enter service id -> enter quaintly	add service to cart		
7	cart management and billing	open software -> login -> 4 cart -> if item need to be remove then enter y -> enter item number on cart list -> if item no need to be remove then enter n	List all final items (product, service), remove item from final cart, view bill invoice		
8	change password	open software -> login -> 3 manage login -> 1 change password -> old password -> new password	password change		
9	delete account	open software -> login -> 3 manage login -> 2 delete account -> enter y	account deleted		
<b>Role : Admin</b>					
1	View service list	Open software -> login -> 1 available service -> 1 List service	List services		

Figure 143 UAT sample questionnaire page 2

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
2	view product list	Open software -> login -> 2 available product -> 1 List product	List products		
3	search service	Open software -> login -> 1 available service -> 2 search service -> enter service key word	Search services		
4	search product	Open software -> login -> 2 available product -> 2 search product -> enter service key word	Search product		
5	Buy product	Open software -> login -> 2 available product -> 3 buy -> enter product id -> enter quantity	add product to cart		
6	Buy service	Open software -> login -> 1 available service -> 3 buy -> enter service id -> enter quantity	add service to cart		
7	cart management and billing	open software -> login -> 5 cart -> if item need to be remove then enter y -> enter item number on cart list -> if item no need to be remove then enter n	List all final items (product, service), remove item from final cart, view bill invoice		
8	change password	open software -> login -> 4 manage login -> 1 change password -> old password -> new password	password change		
9	delete account	open software -> login -> 4 manage login -> 3 delete account -> enter y	account deleted		

Figure 144 UAT sample questionnaire page 3

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
10	Add Service	open software -> login -> 3 manage service & product -> 1 add service -> enter service name -> enter price	Service added to service file		
11	Delete service	open software -> login -> 3 manage service & product -> 2 delete service -> enter service number -> confirm y	service deleted to service file		
12	Update service	open software -> login -> 3 manage service & product -> 3 update service -> enter service number -> enter service name -> enter service price	service updated to service file		
13	Add product	open software -> login -> 3 manage product & product -> 1 add product -> enter product name -> enter price	product added to service file		
14	Delete product	open software -> login -> 3 manage product & product -> 2 delete product -> enter product number -> confirm y	product deleted to service file		
15	Update product	open software -> login -> 3 manage product & product -> 3 update product -> enter product number -> enter product name -> enter product price	product updated to product file		

Figure 145 UAT sample questionnaire page 4

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
16	delete user account	open software -> login -> 4 manage login -> 2 manage user accounts -> enter target username -> enter target username's password -> 2 delete account -> confirm y	list all user name with password, delete target user account		
17	change user account's password	open software -> login -> 4 manage login -> 2 manage user accounts -> enter target username -> enter target username's password -> 1 change password -> enter old password -> enter new password	list all user name with password, change target user accounts password		
General Questions/Comments					Signature
<p>.....</p>					

*Figure 146 UAT sample questionnaire page 5*

## User feedback

User Acceptance Testing (UAT)					
Project Name		Best Auto Mart			
Testing Start Date / Time		04-04-2021			
Testing End Date / Time		04-04-21			
Name of Tester		MN Rizni Mohamed			
Name of User		MN Raiza Mohamed			
Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
<b>Role : Guest</b>					
1	View service list	Open software -> 1 available service -> 1 List service	List services	Pass	Prices are not in proper order.
2	view product list	Open software -> 2 available product -> 1 List product	List products	Pass	satisfied
3	search service	Open software -> 1 available service -> 2 search service -> enter service key word	Search services	Pass	good
4	search product	Open software -> 2 available product -> 2 search product -> enter service key word	Search product	Pass	good
5	Signup	Open software -> 4 signup -> enter username, password, confirm password	Account created	Pass	good
<b>Role : User</b>					
1	View service list	Open software -> login -> 1 available service -> 1 List service	List services	Pass	Prices have to be in proper order
2	view product list	Open software -> login -> 2 available product -> 1 List product	List products	Pass	good

Figure 147 UAT user feedback page 1

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
3	search service	Open software -> login -> 1 available service -> 2 search service -> enter service key word	Search services	Pass	good
4	search product	Open software -> login -> 2 available product -> 2 search product -> enter service key word	Search product	Pass	price title is not in the proper alignment
5	Buy product	Open software -> login -> 2 available product -> 3 buy -> enter product id -> enter quantity	add product to cart	Pass	very good.
6	Buy service	Open software -> login -> 1 available service -> 3 buy -> enter service id -> enter quantity	add service to cart	Pass	satisfied
7	cart management and billing	open software -> login -> 4 cart -> if item need to be remove then enter y -> enter item number on cart list -> if item no need to be remove then enter n	List all final items (product,service), remove item from final cart, view bill invoice	Pass	good
8	change password	open software -> login -> 3 manage login -> 1 change password -> old password -> new password	password change	Pass	password characters not hiding
9	delete account	open software -> login -> 3 manage login -> 2 delete account -> enter y	account deleted	Pass	satisfied
<b>Role : Admin</b>					

Figure 148 UAT user feedback page 2

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
1	View service list	Open software -> login -> 1 available service -> 1 List service	List services	Pass	satisfied
2	view product list	Open software -> login -> 2 available product -> 1 List product	List products	Pass	satisfied
3	search service	Open software -> login -> 1 available service -> 2 search service -> enter service key word	Search services	Pass	price title is not in the proper alignment
4	search product	Open software -> login -> 2 available product -> 2 search product -> enter service key word	Search product	Pass	price title is not in the proper alignment
5	Buy product	Open software -> login -> 2 available product -> 3 buy -> enter product id -> enter quantity	add product to cart	Pass	satisfied
6	Buy service	Open software -> login -> 1 available service -> 3 buy -> enter service id -> enter quantity	add service to cart	Pass	satisfied
7	cart management and billing	open software -> login -> 5 cart -> if item need to be remove then enter y -> enter item number on cart list -> if item no need to be remove then enter n	List all final items (product,service) , remove item from final cart, view bill invoice	Pass	satisfied.

Figure 149 UAT user feedback page 3

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
8	change password	open software -> login -> 4 manage login -> 1 change password -> old password -> new password	password change	Pass	password characters are visible.
9	delete account	open software -> login -> 4 manage login -> 3 delete account -> enter y	account deleted	Pass	satisfied
10	Add Service	open software -> login -> 3 manage service & product -> 1 add service -> enter service name -> enter price	Service added to serivce file	Pass	good
11	Delete service	open software -> login -> 3 manage service & product -> 2 delete service -> enter service number -> confirm y	service deleted to serivce file	Pass	good
12	Update service	open software -> login -> 3 manage service & product -> 3 update service -> enter service number -> enter service name -> enter service price	service updated to service file	Pass	satisfied
13	Add prodcut	open software -> login -> 3 manage product & product -> 1 add product -> enter product name -> enter price	product added to serivce file	Pass	good
14	Delete product	open software -> login -> 3 manage product & product -> 2 delete product -> enter product number -> confirm y	product deleted to serivce file	Pass	satisfied

Figure 150 UAT user feedback page 4

Test #	Description of tasks	Steps To Execute	Expected Results	Pass/ Fail	Defect/Comments/Additions
15	Update product	open software -> login -> 3 manage product & product -> 3 update product -> enter product number -> enter product name -> enter product price	product updated to product file	Pass	satisfied
16	delete user account	open software -> login -> 4 manage login -> 2 manage user accounts -> enter target username -> enter target username's password -> 2 delete account -> confirm y	list all user name with password, delete target user account	Pass	good
17	change user account's password	open software -> login -> 4 manage login -> 2 manage user accounts -> enter target username -> enter target username's password -> 1 change password -> enter old password -> enter new password	list all user name with password, change target user accounts password	Pass	password characters are visible but its good
General Questions/Comments					Signature
I am truly satisfied with the software and 100% recommended it to any company which want to computerize their products and services.					

Figure 151 UAT user feedback page 5

## **Test conclusion**

Test cases and test summery reports are depicting that all the functionalities are meet the requirements successfully according to test exit criteria. Although, the user feedbacks say there two non-functionalities are need to be modified. Those are mentioned below

- In View service list, search product, search service processes, the tile of service/product and price are not in proper alignment
- When typing password, characters are visible

Since above bugs are non-functional requirements, this system can go live to the company for managing their works and these bugs will fixes in next version release

## **Future Recommendation**

Following are some of the suggestions that could be added as enhancements to the existing system to make the system a much-improved product in future.

- Make proper alignments in View service list, search product, search service processes
- Make password characters are hidden when typing it
- Give GUI interface to this program by QT framework
- Convert all local database into server-based database like MYSQL, SQL, Firebase
- Make live this software on server to give remote access
- Give reset account password way to users like OTP verification