# UNIVERSITI TEKNIKAL MALAYSIA MELAKA

## FAKULTI TEKNOLOGI MAKLUMAT DAN KOMUNIKASI

# W O R K S H O P  1

# R E P O R T

| | |
|---|---|
| **NAME** | **MUHAMMAD RIDHWAN BIN RAZALEE** |
| **MATRIC NUMBER** | **B031910197** |
| **COURSE** | **BACHELOR OF COMPUTER SCIENCE (SOFTWARE DEVELOPMENT)** |
| **PROJECT TITLE** | **SNACK FOOD MANAGEMENT SYSTEM** |
| **SUPERVISOR NAME** | **PROFESOR DR. MOHD KHANAPI BIN ABD GHANI** |
| **EVALUATOR NAME** | **DR. SATRYA FAJRI PRATAMA** |

# Table of Contents

# CHAPTER 1
# INTRODUCTION

## 1.1 Introduction

As this project title is Snack Food Management System and it will be applying to Kiosk System for ordering. This project may help in managing many kinds of services such as purchasing order, update stock list, billing, and sale report. In the food business, efficient service may result in outstanding customer service and a customer would likely to return in the future.

## 1.2 Problem Statement

Nowadays, with the growth of technologies, the world has been transformed. There are so many technologies that have been found to make human life easier. Meanwhile, some of the systems are still not upgraded yet. Some food business in Malaysia still uses conventional approach such as pen and paper techniques. There are lots of shortages in this system such as;

i. The conventional approach does not organize stock and order well.
ii. The order may be skipped or not structured.
iii. The item customer needed is hard to find.
iv. The conventional approach will take time to take the order and organize order.

## 1.3 Background of Project

In this system the administrator will be able to register their account to access the system and login to use this system. Moreover, the administrator will be able to view stock list, add new product, delete existing product and update stock. Also, there is customer menu order which will help in purchasing or buying product from the Kiosk System. After that, the customer will need to bring the receipt order to administrator to get their item. Next, there is sale report menu for administrator or manager to view their shop sale by product ID, product name, product highest sold, and by date.

## 1.4 Objectives

1. To develop and design a system to help food business such as snack food retail store.

2. To assess in managing inventory stock, order and sale report.

3. To make recommendations on improving management services.

## 1.5 Scopes

i. The system will be built by using Visual Studio with the programming language C++.

ii. MySQL will be the database to store data in this development.

iii. Module to be developed:

- Access Module
  - Admin will be able to register their account.
  - Admin will be able to login to their account and the system.
  - Admin will be able to update their password.
  - Admin will be able to update their phone number.
- Inventory Stock Module
  - Admin will be able to view existing stock product list.
  - Admin will be able to add new product.
  - Admin will be able delete existing product.
  - Admin will be able to update product stock.
- Order Module
  - Admin will be able to search receipt number.
  - Customer will be able to search product by name.
  - Customer will be able to purchase multiple order.
  - Customer will be able to view their order.
  - Customer will be able to delete unwanted order.
  - Customer will be able to cancel order.
  - The system will provide customer with receipt after payment.
- Sale Report Module
  - Admin will be able to view sale record.
  - Admin will be able to view sale by product ID.
  - Admin will be able to view sale by product name.
  - Admin will be able to view sale by highest product sold.
  - Admin will be able to view sale by date.

iv. Target User:

- Snack Shop
- Individual Food Business
- Food Truck Business

# CHAPTER 2
# ANALYSIS OF PROBLEM

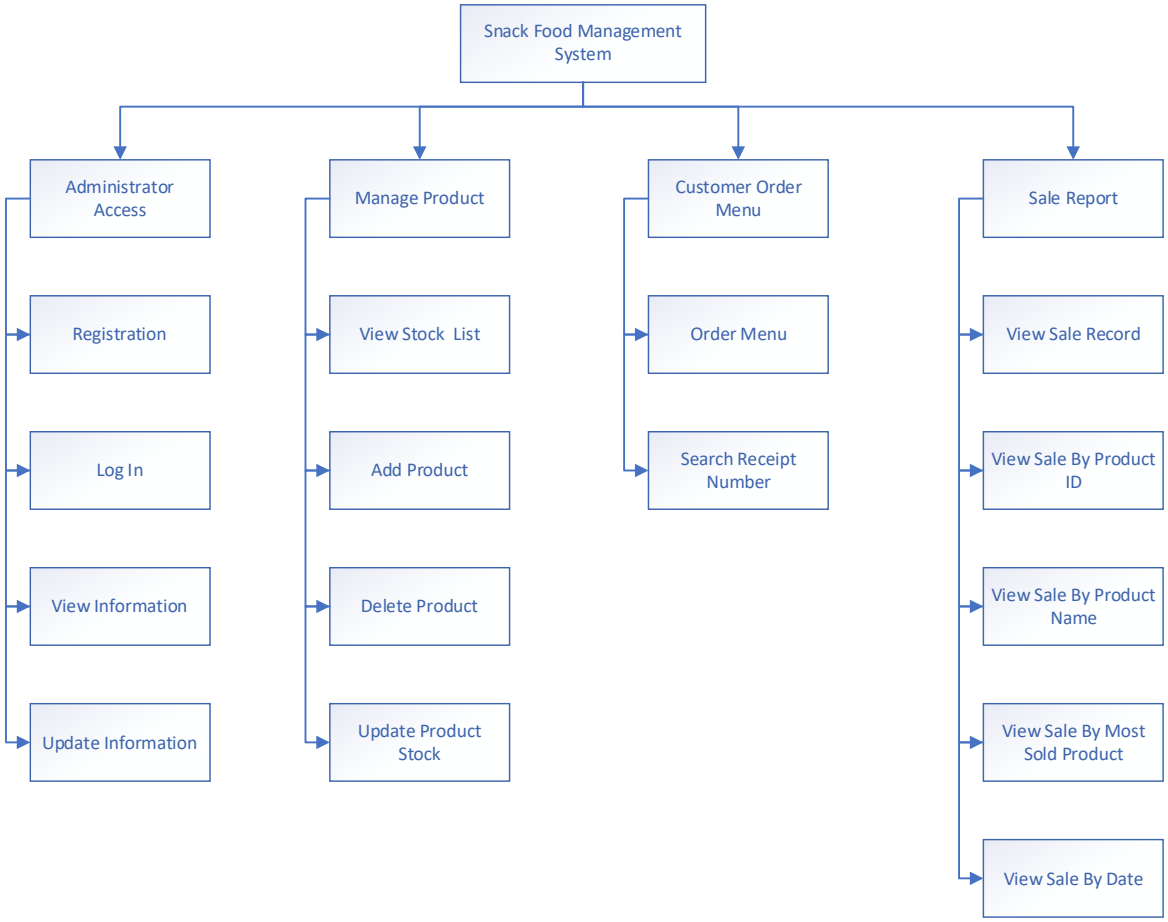
## 2.1 Problem Description

Nowadays, some food business in Malaysia still uses conventional approach such as pen and paper to update their stock, managing customer order, and to record their sale especially business such as retail store, individual business and etc. Moreover, conventional approach process quite slow because the customer will need to search for their item in the shop store to find what they desired to buy. This will take time as the customer need to search for their item. Therefore, for a bigger shop store the business owner can use this system where it will apply to kiosk system to help in finding their item and ordering their item. Also, many foods business existing system cover only on cashier part and billing part. This system will assess in inventory stock, purchasing order, and sale record.

## 2.2 Problem Decomposition

| No | Problem Decomposition | Solution |
|---|---|---|
| 1 | Conventional approach does not organize stock and order well. | The system will assess management in organize order, organize inventory stock and billing. |
| 2 | Conventional approach may make a mistake such as delivering different item to customer, forget to stock up inventory and incorrect balance. | The system may help in calculating the total payment and ordering. |
| 3 | Billing, Sales report could be damage easily | Report data will be saved to the system |

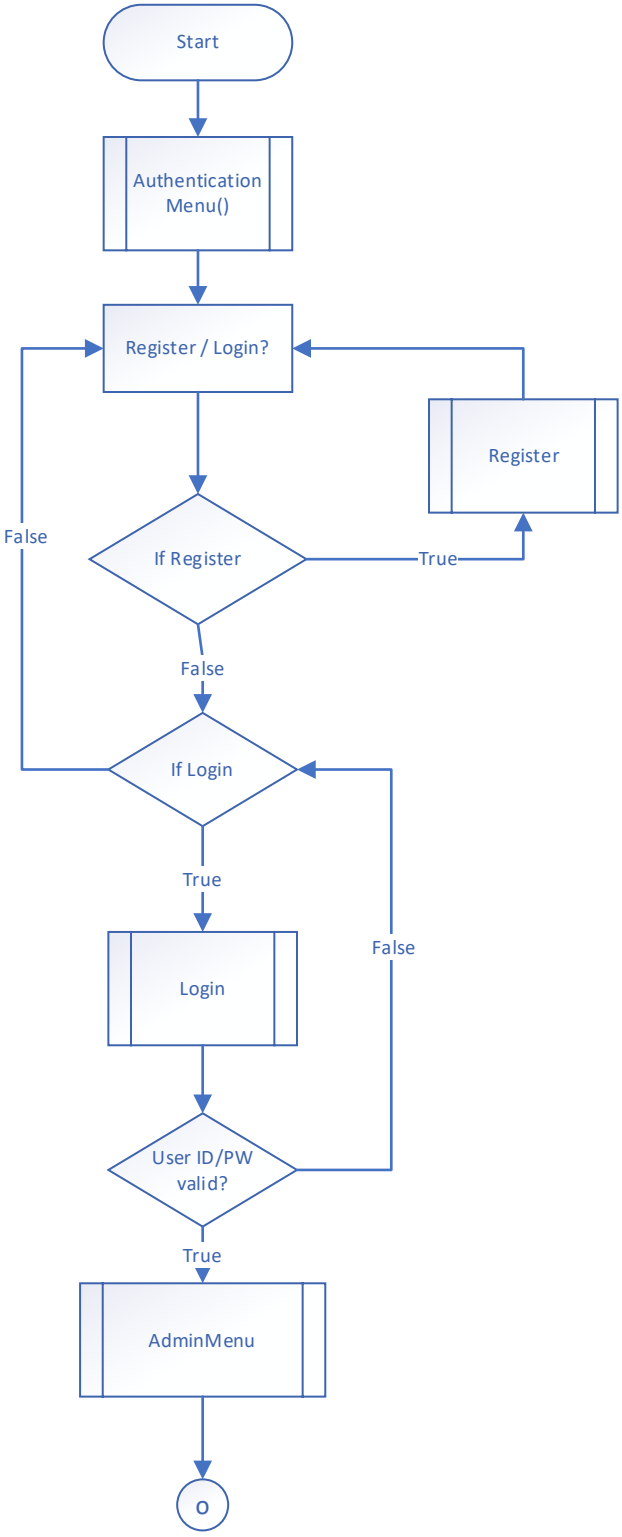
## 2.3 Structured Chart



# CHAPTER 3
# DESIGN

## 3.1 Flowchart

### 3.1.0 Overview of Main Flowchart



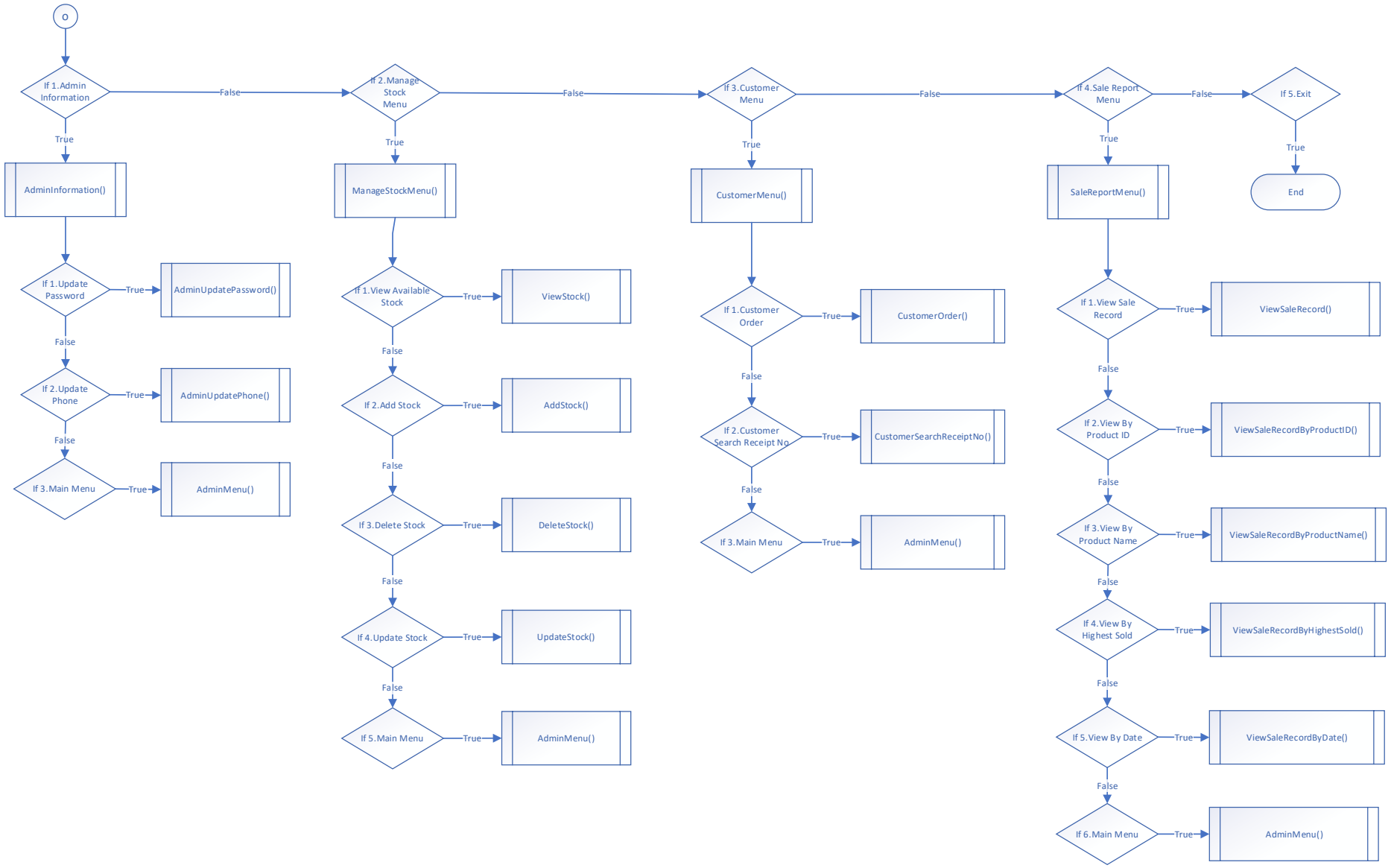



Figure 1: Flowchart Snack Food Management System

### 3.1.1 Main

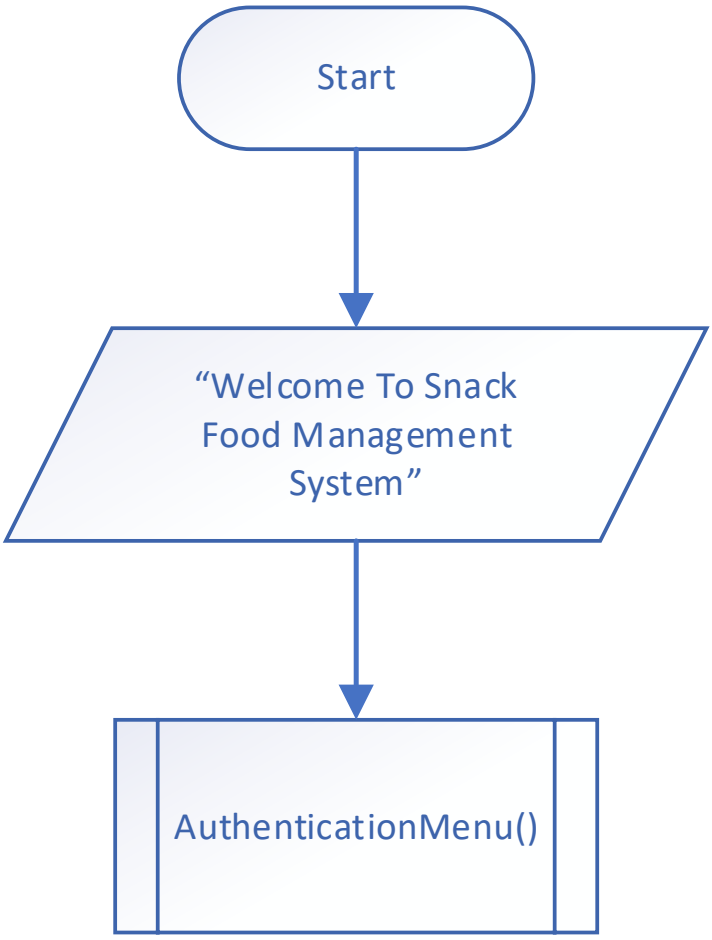



*Figure 2: FlowChart Main*

### 3.1.2 Authentication Menu

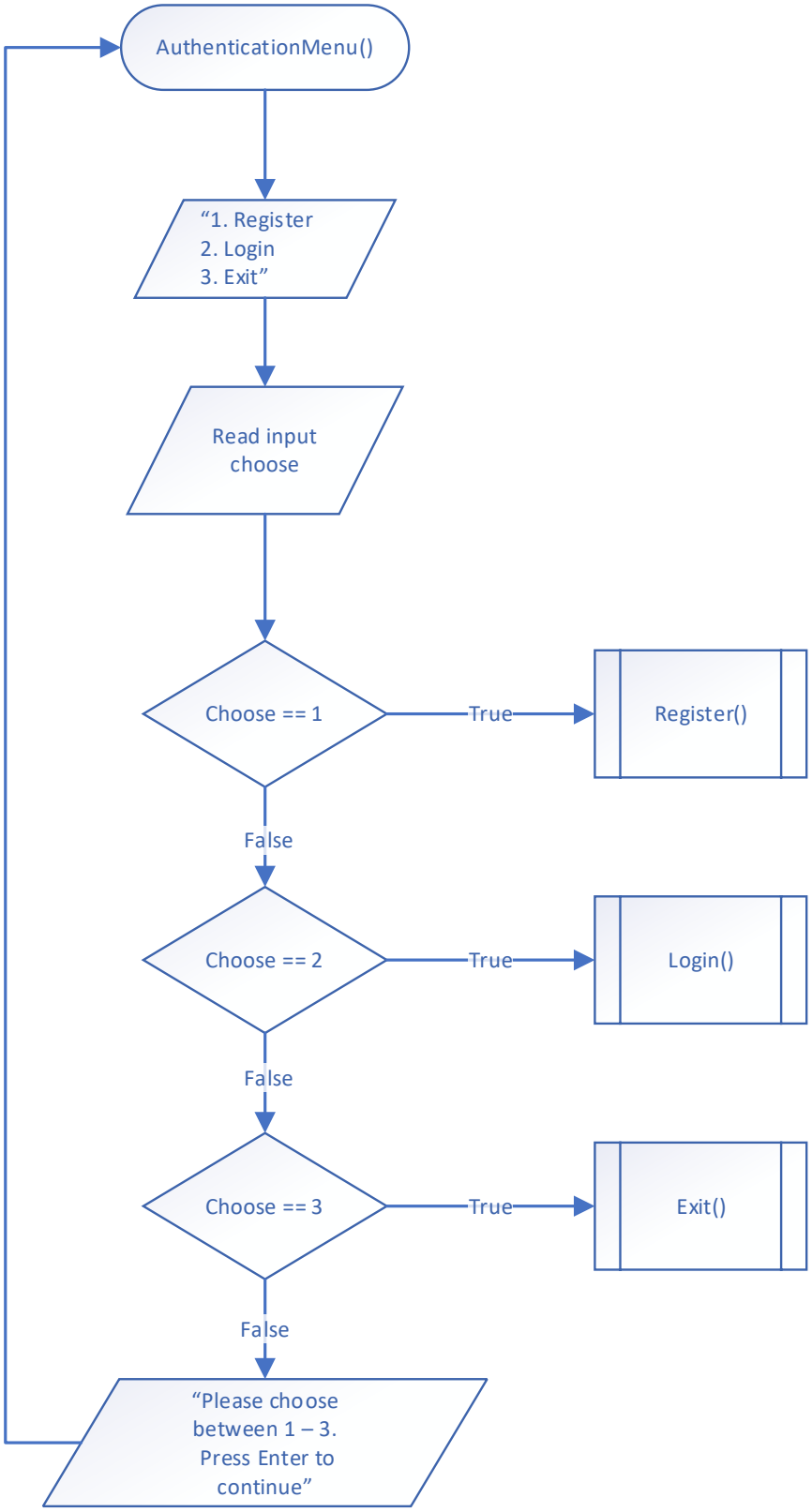



*Figure 3: Flowchart Authentication Menu*

### 3.1.3 Register

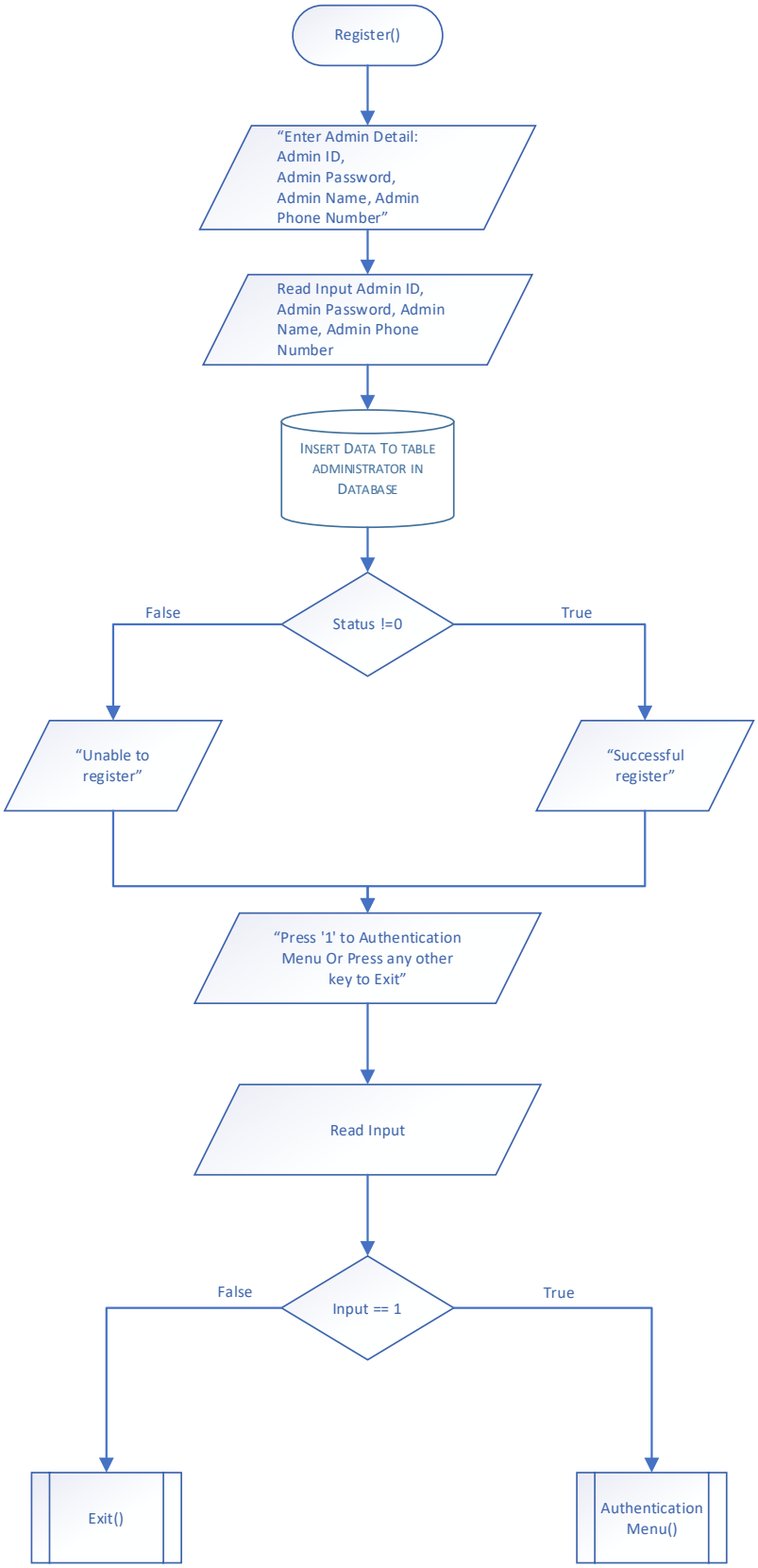



*Figure 4: Flowchart Register*

### 3.1.4 Login

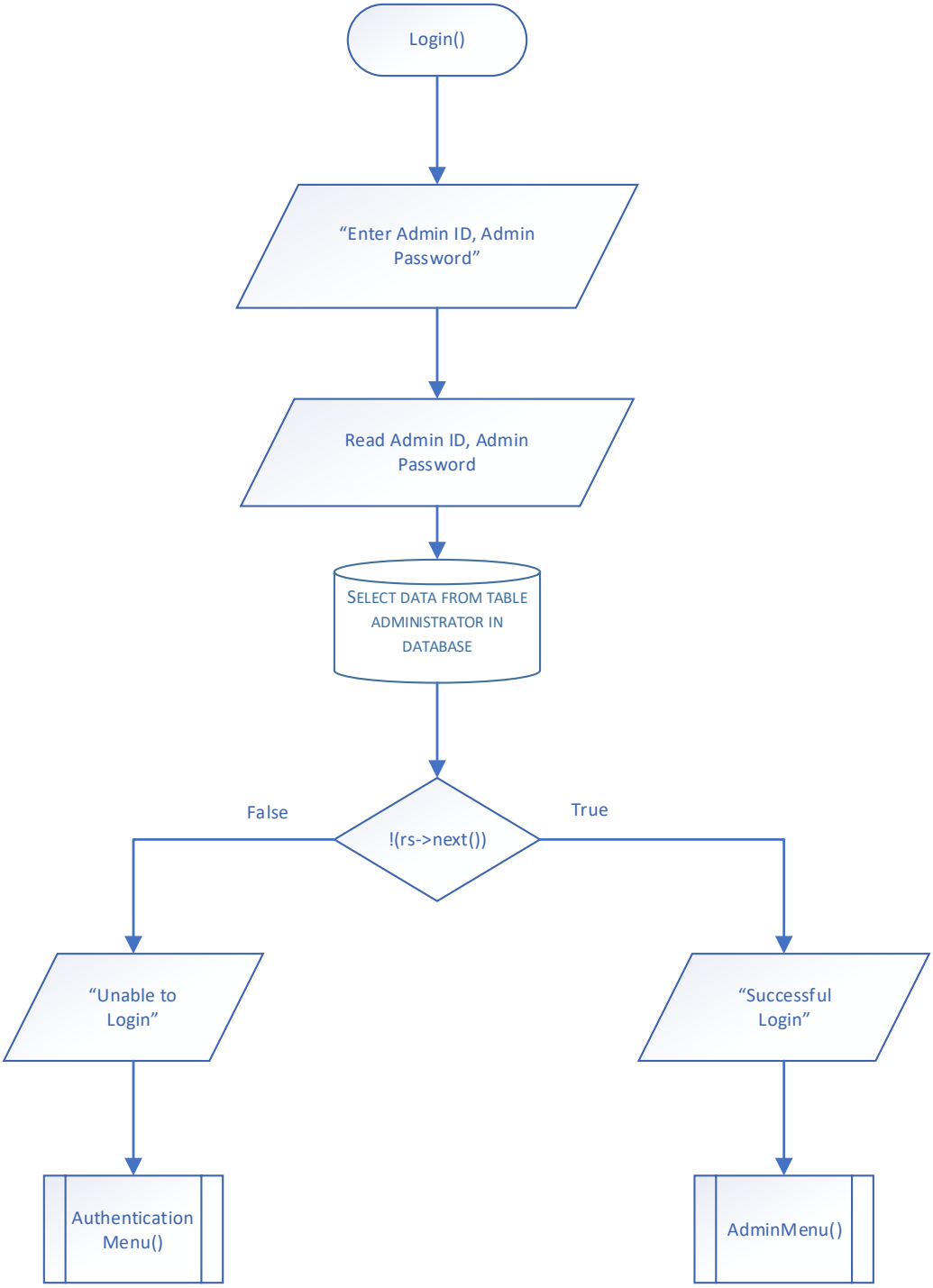



*Figure 5: Flowchart Login*

### 3.1.5 Admin Menu

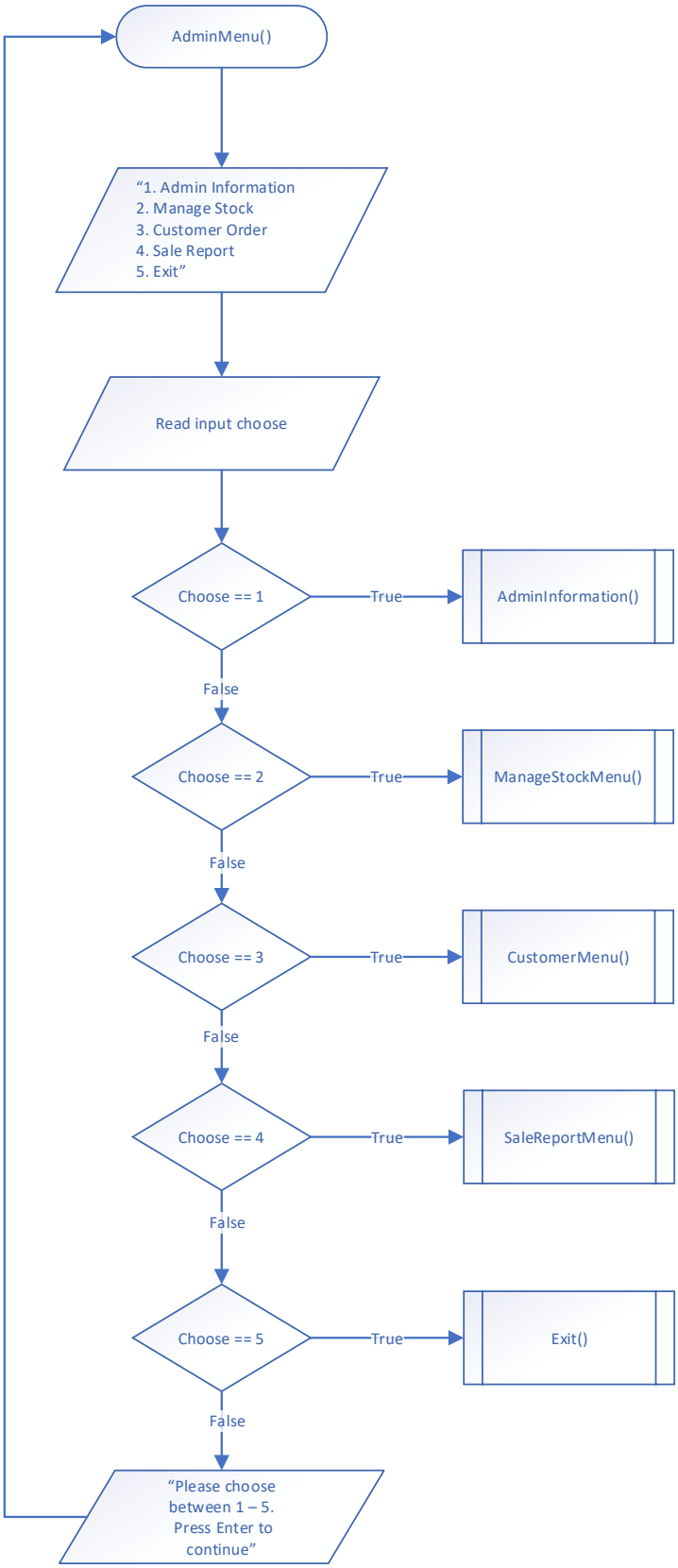



*Figure 6: Flowchart Admin Menu*

### 3.1.6 Admin Information

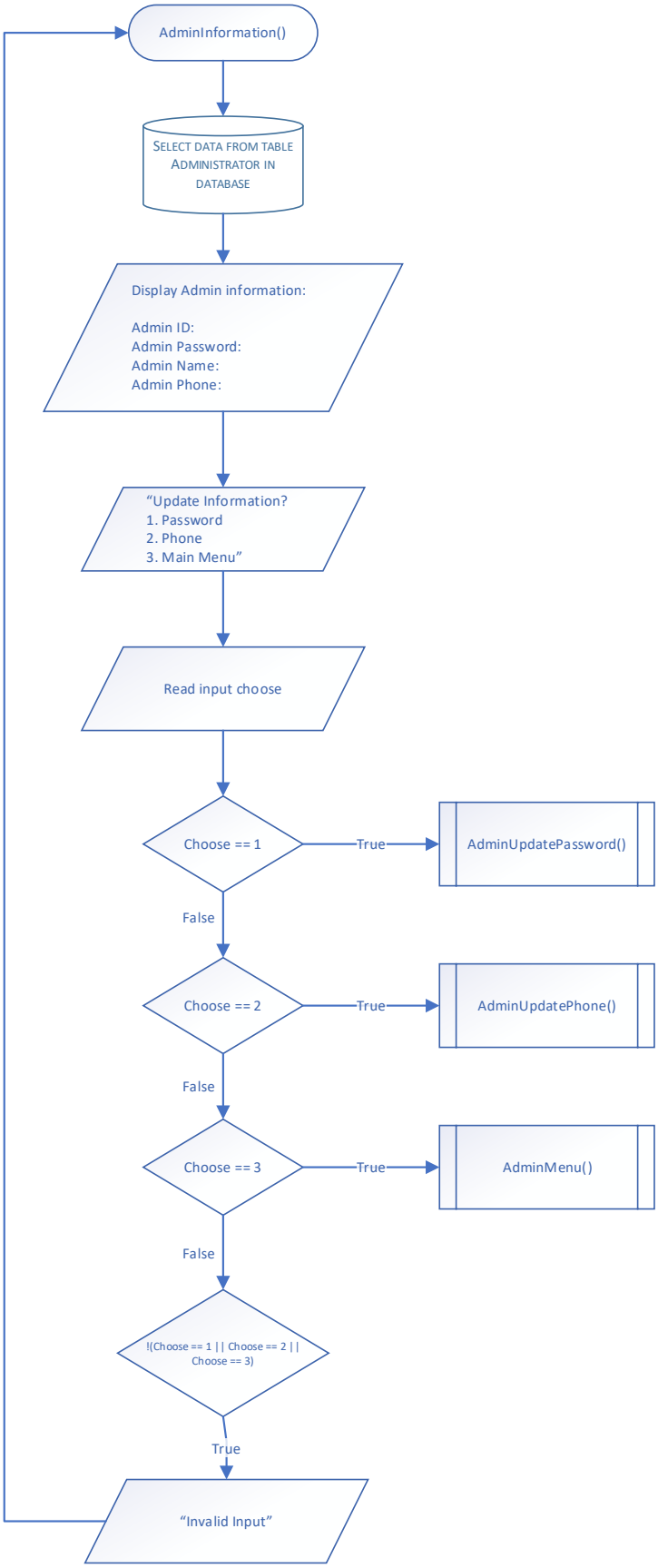



*Figure 7: Flowchart Admin Information*

### 3.1.7 Admin Update Password

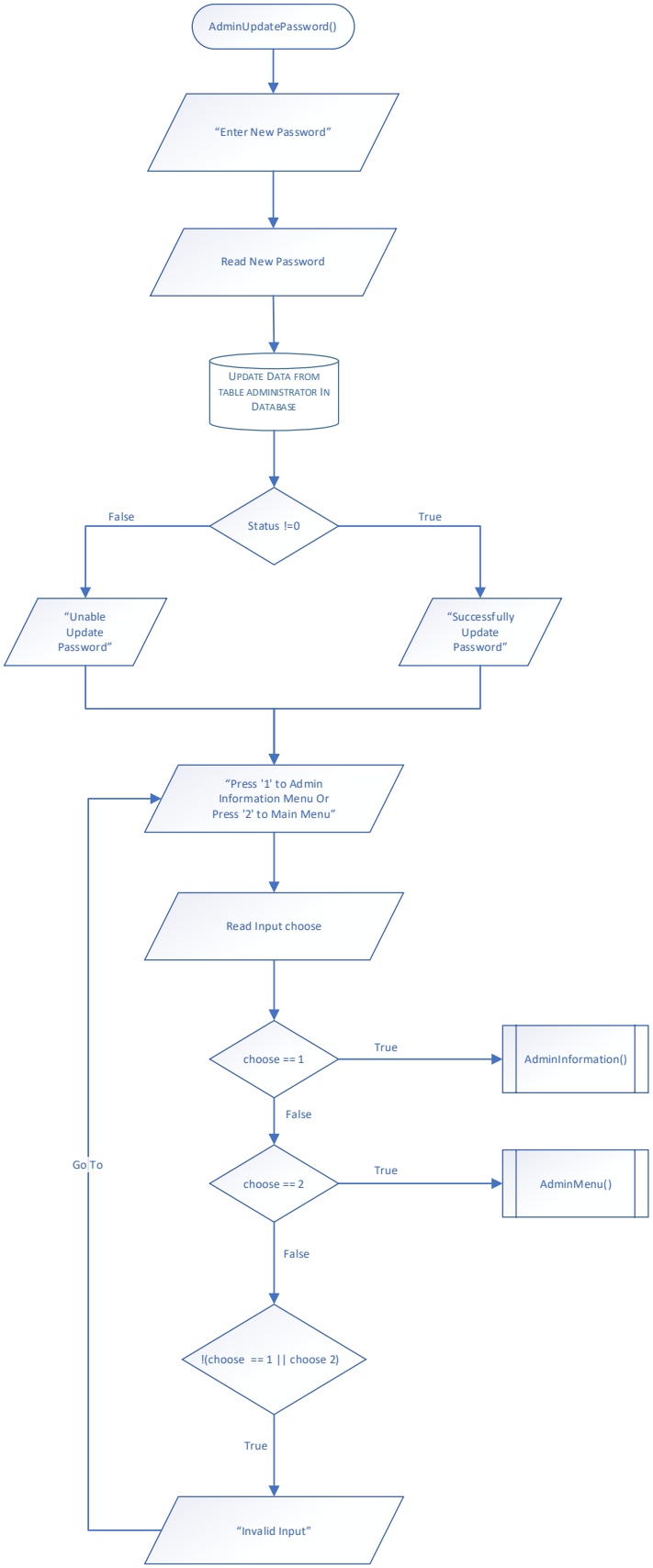



*Figure 8: Flowchart Admin Update Password*

### 3.1.8 Admin Update Phone

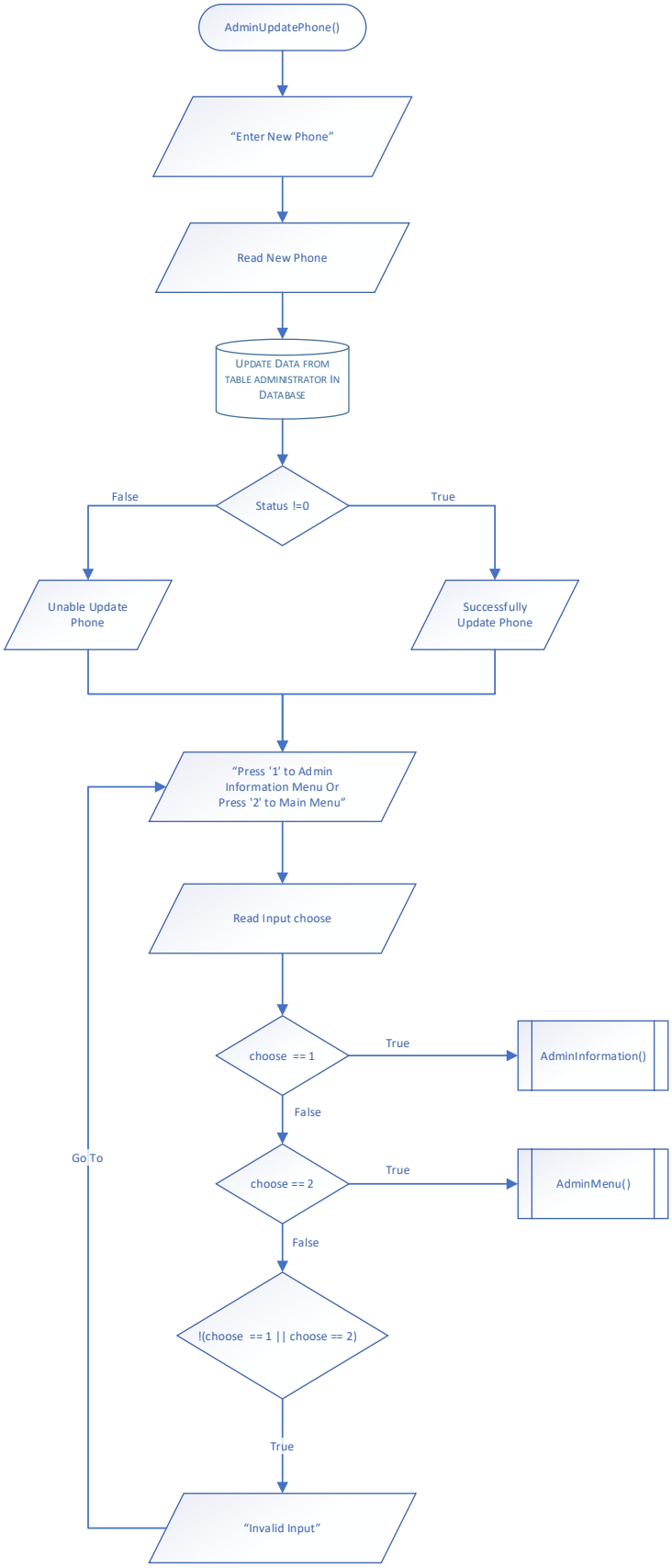



*Figure 9: Flowchart Admin Update Phone*

### 3.1.9 Manage Stock Menu

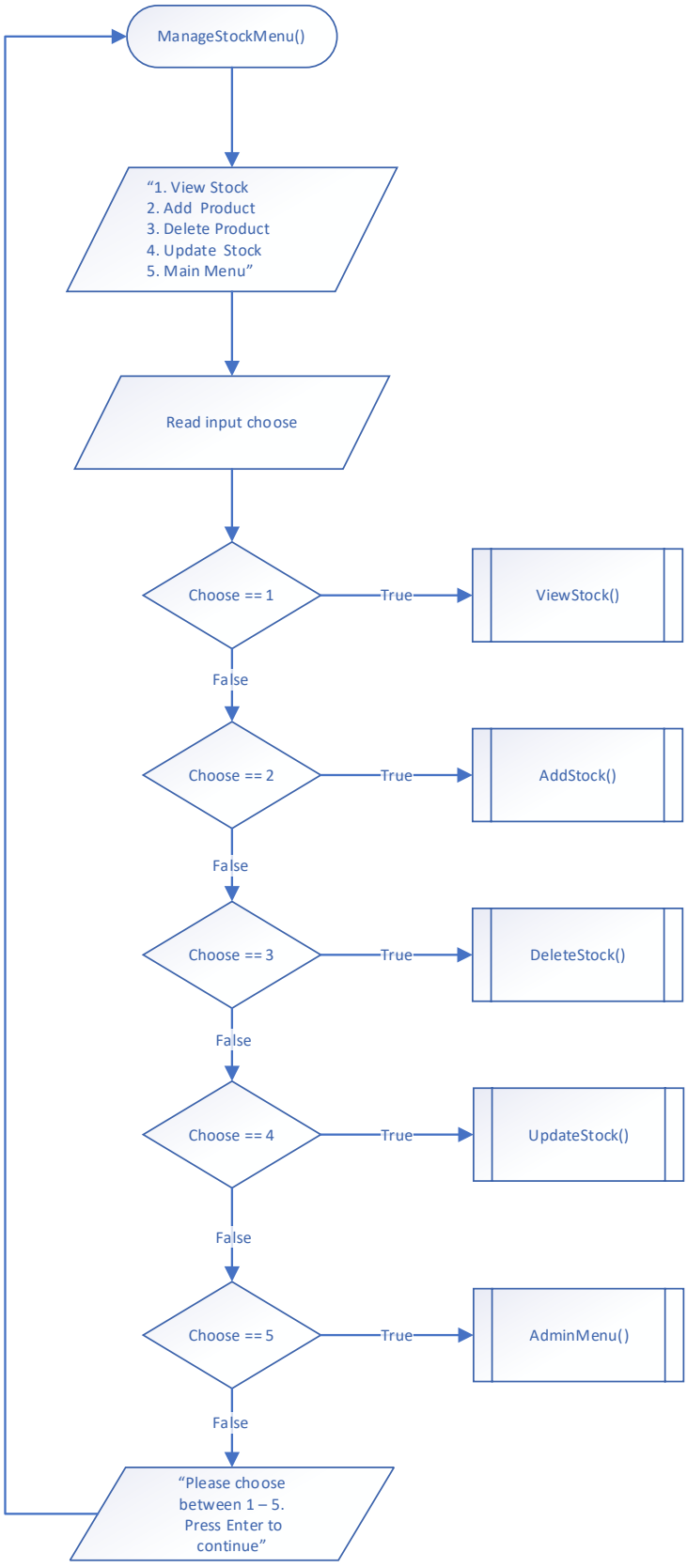



*Figure 10: Flowchart Manage Stock Menu*

### 3.1.10 View Stock

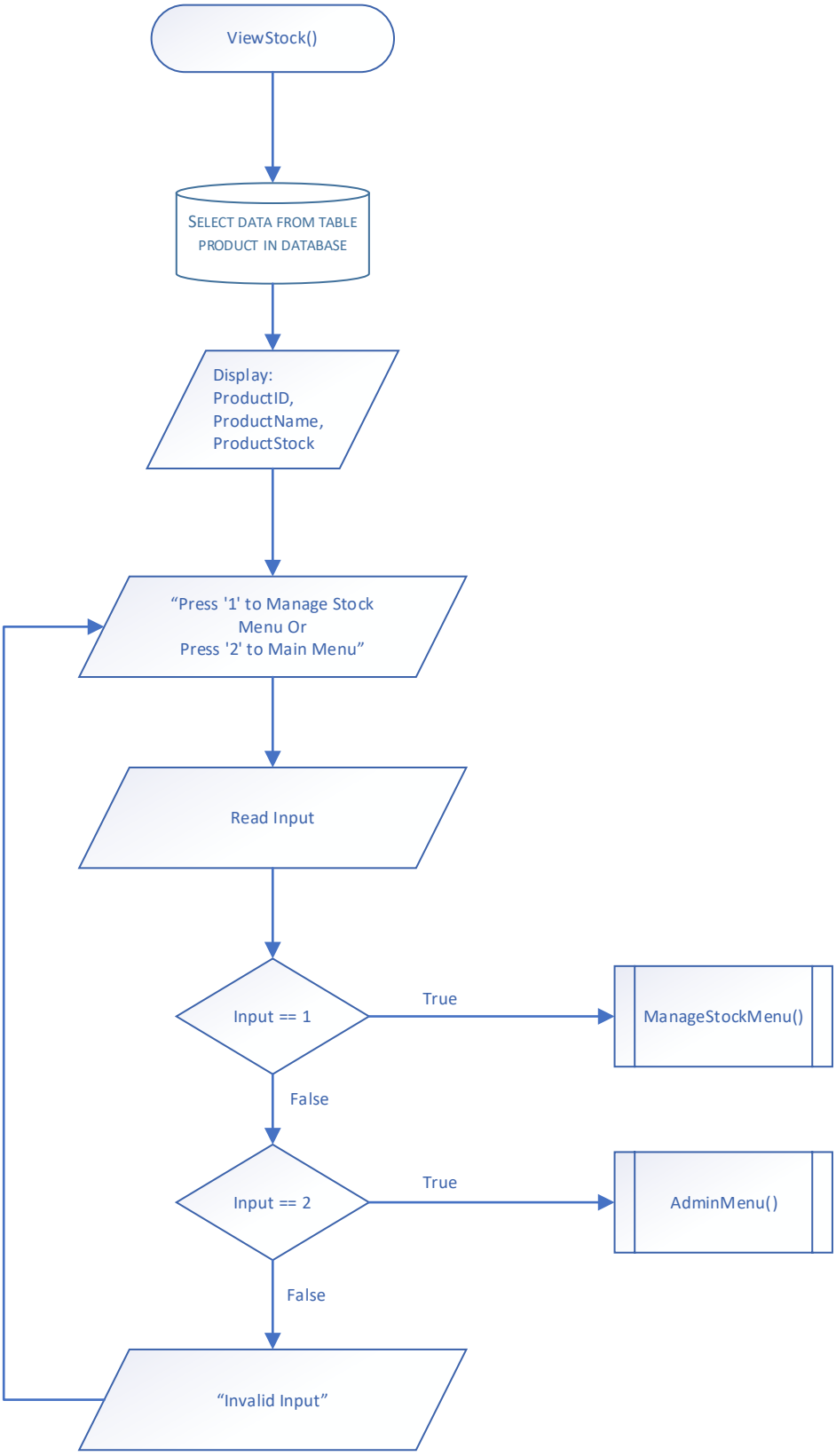



*Figure 11: Flowchart View Stock*

### 3.1.11 Add Stock

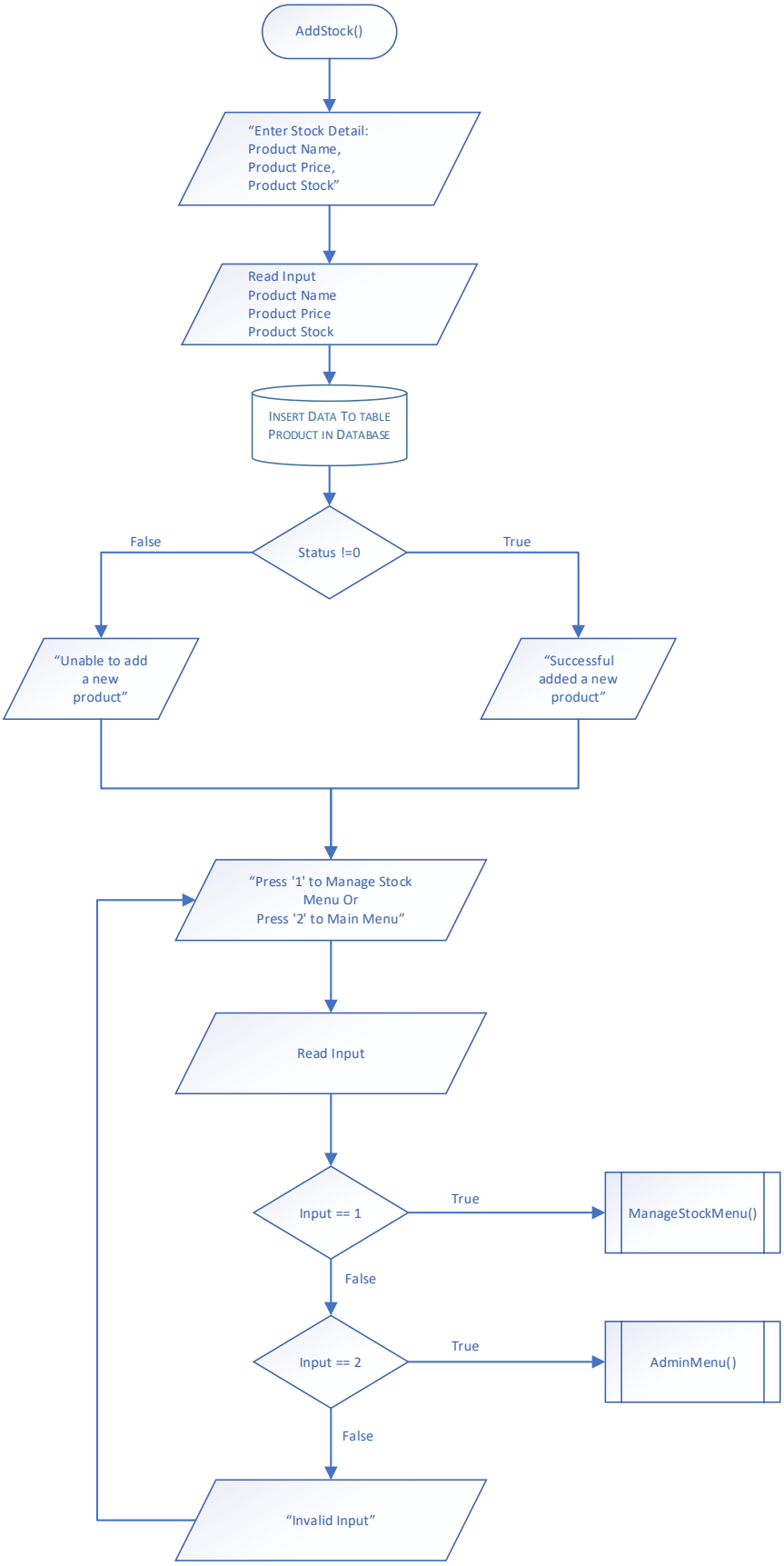



*Figure 12: Flowchart Add Stock*

### 3.1.12 Delete Stock

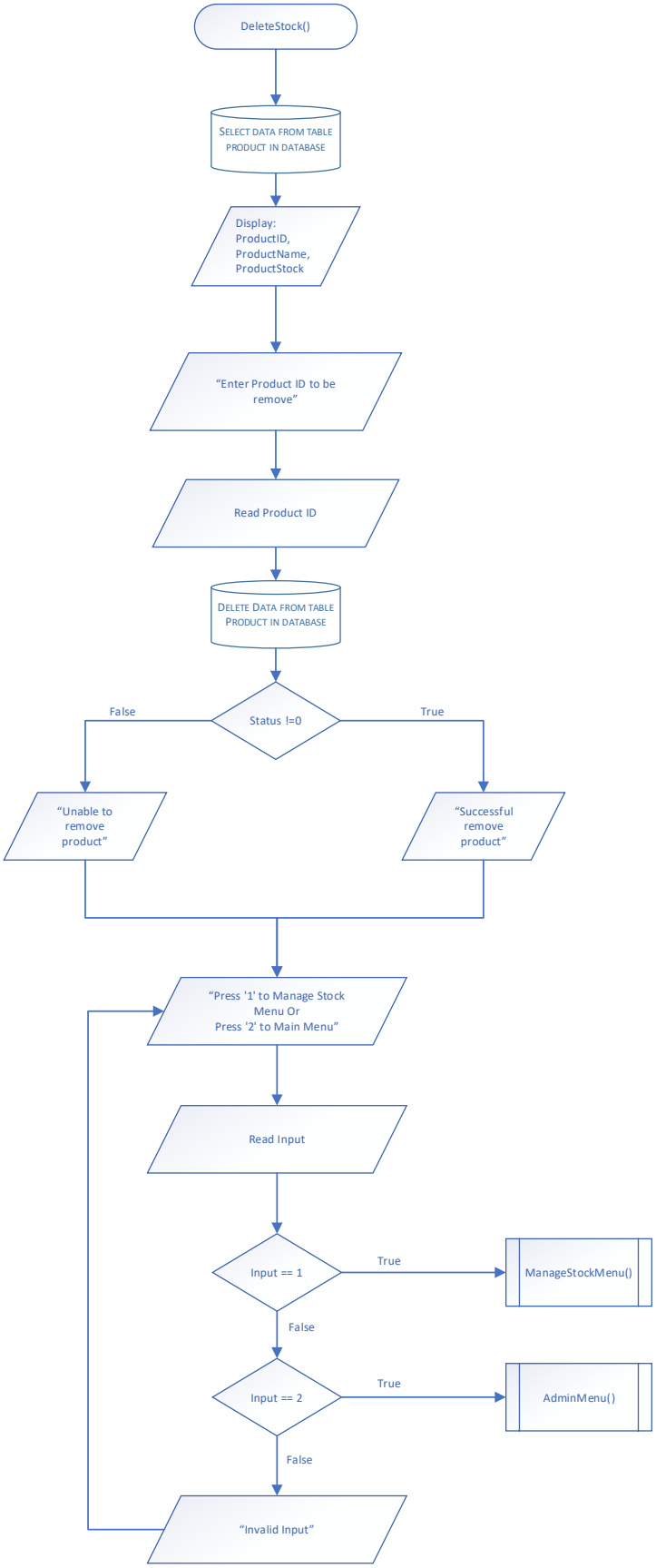
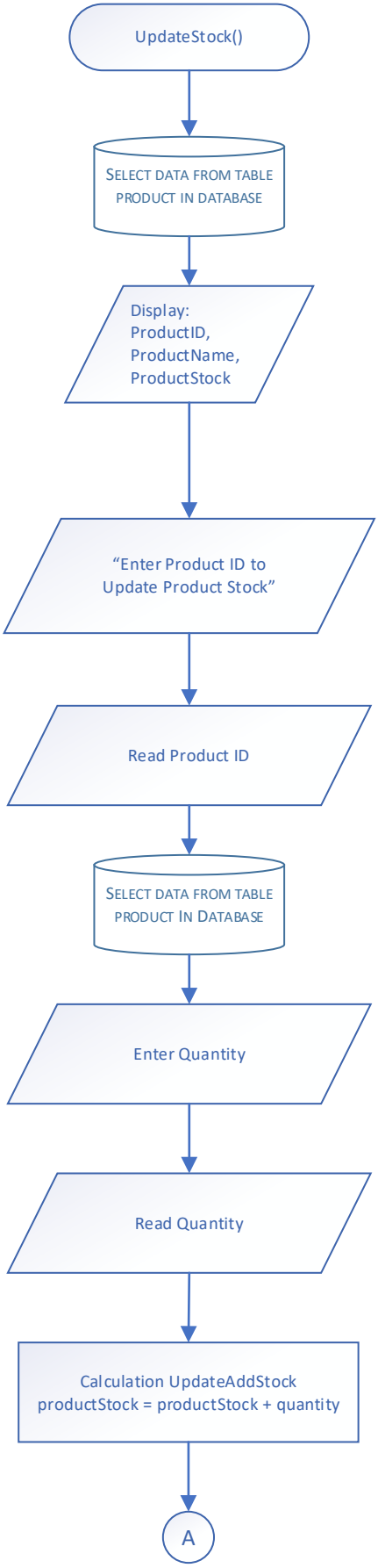



*Figure 13: Flowchart Delete Stock*

### 3.1.13 Update Stock



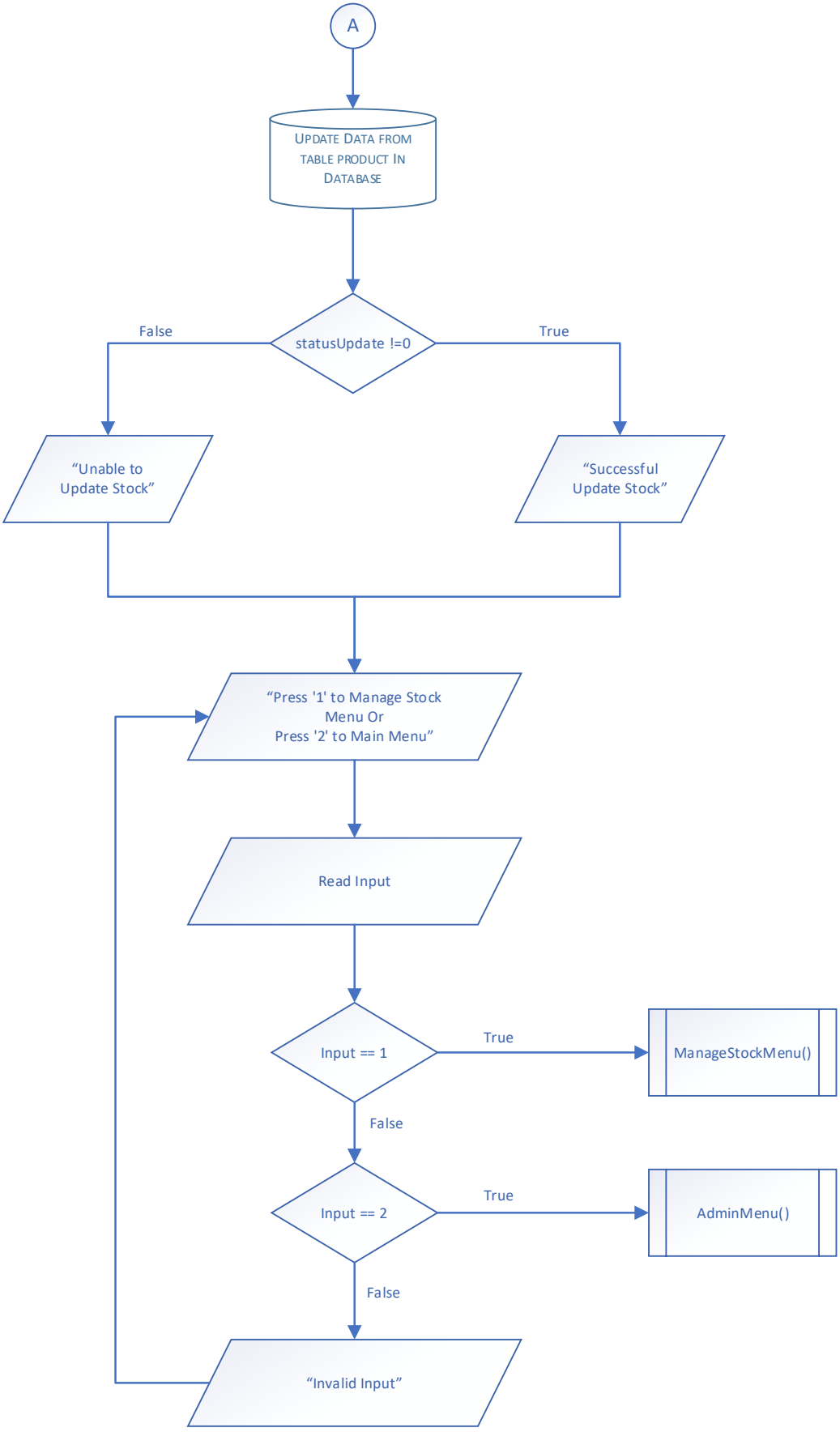



*Figure 14: Flowchart Update Stock*

### 3.1.14 Customer Menu

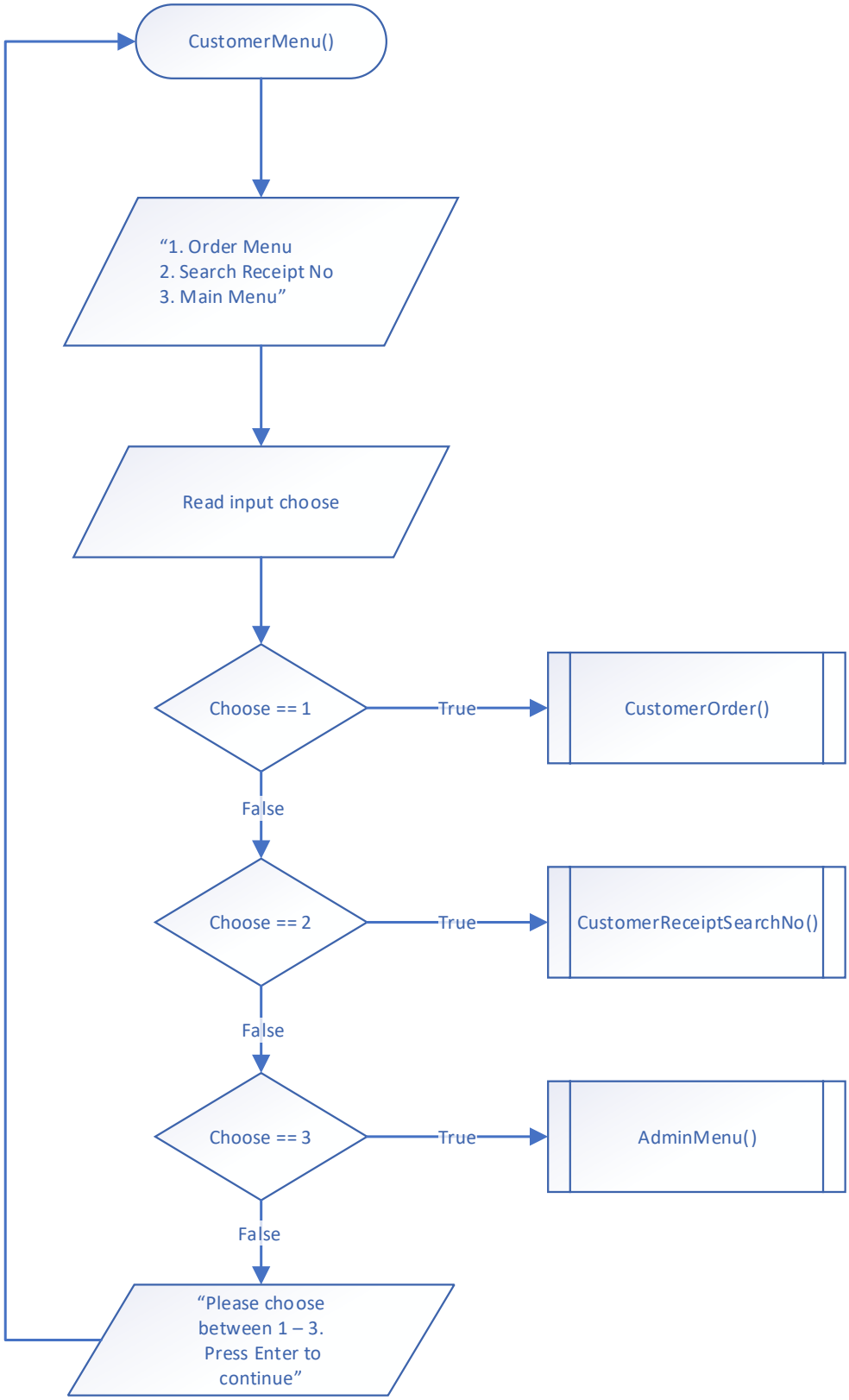
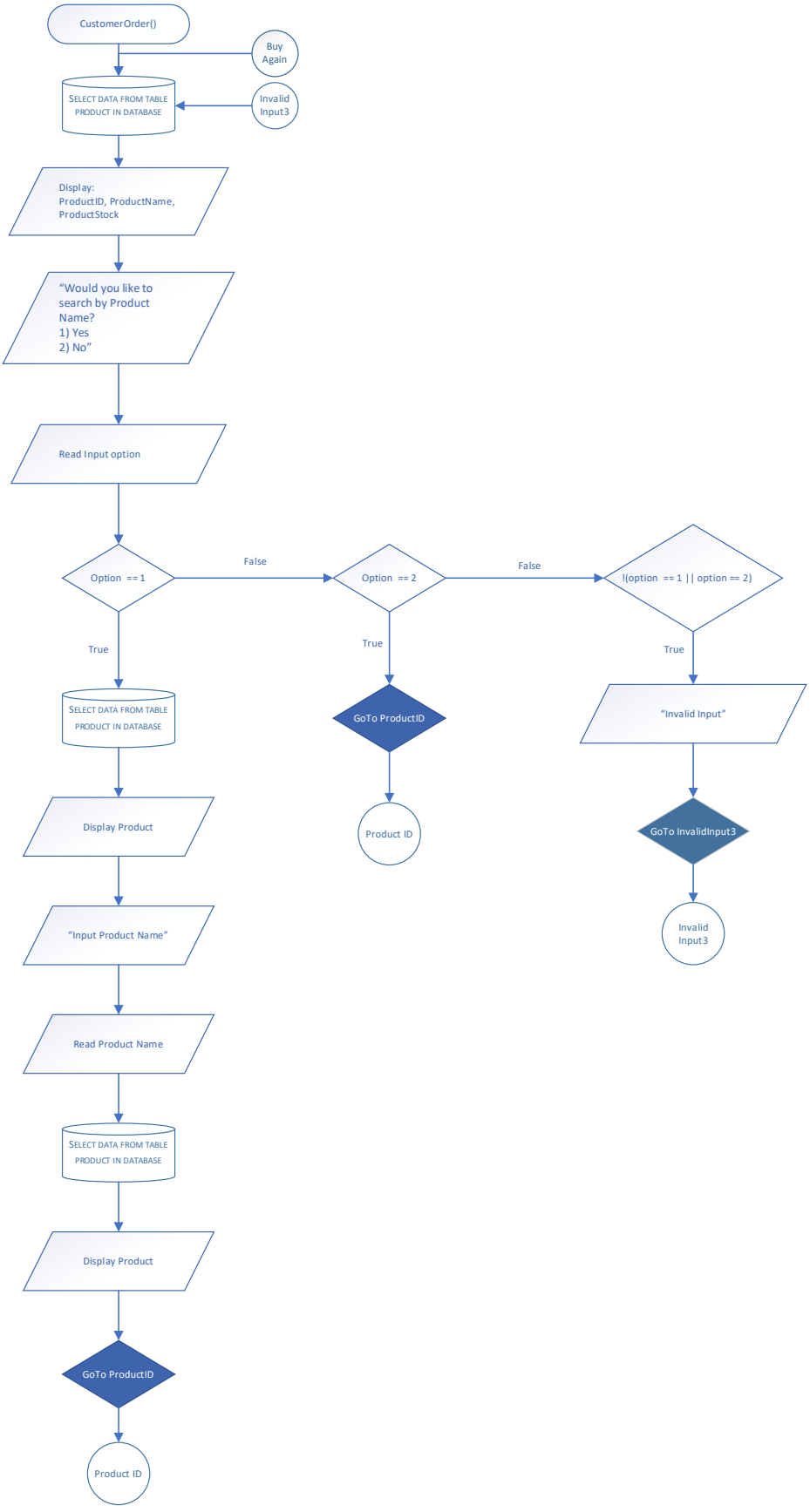
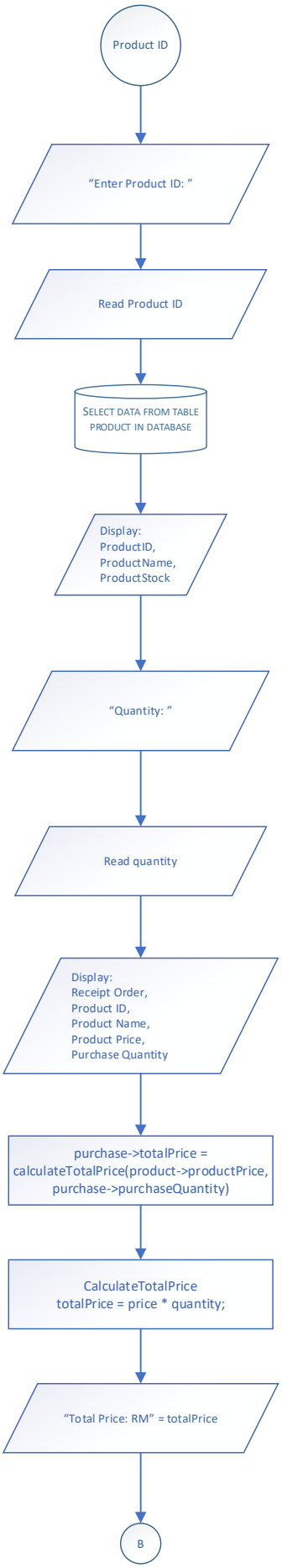
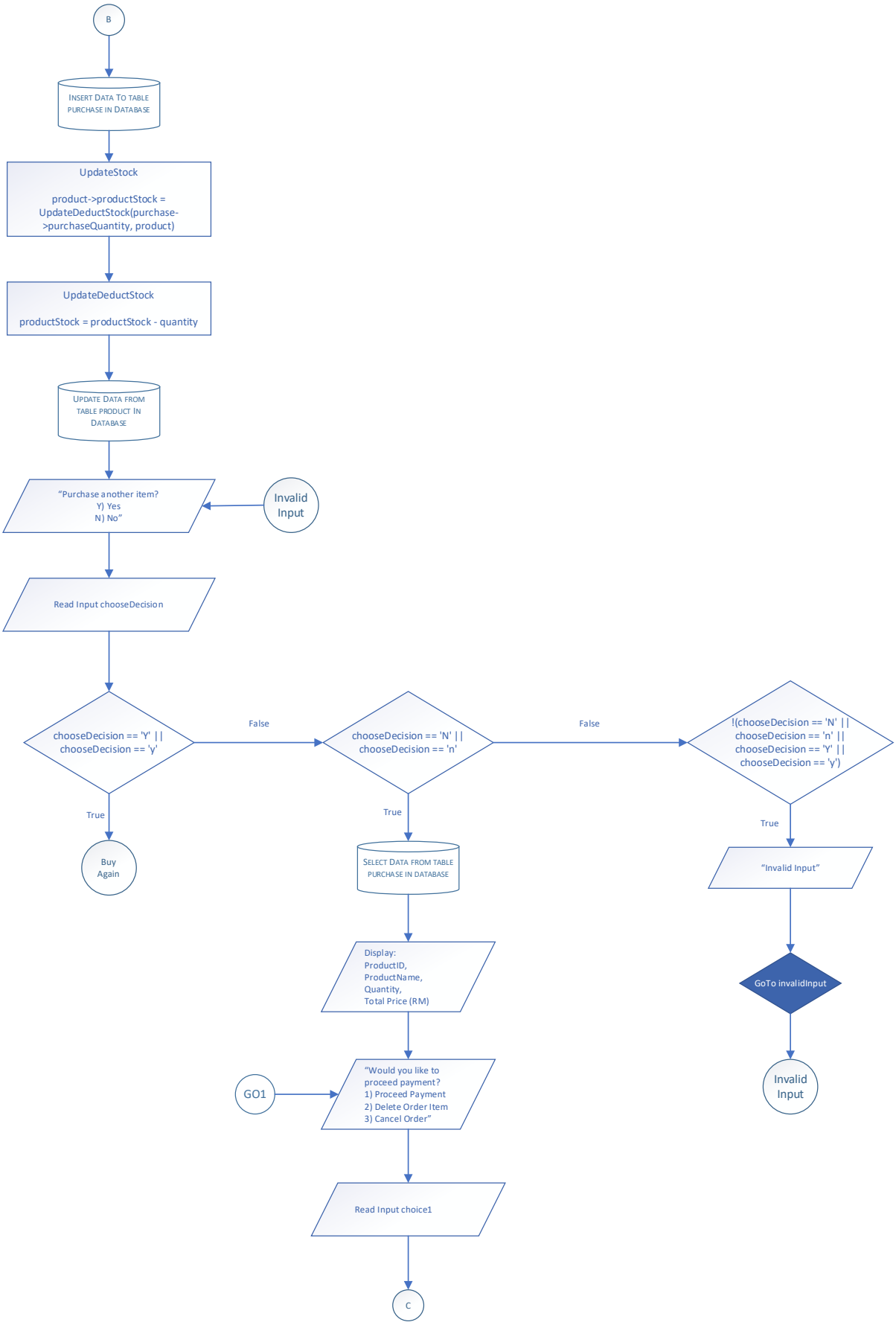


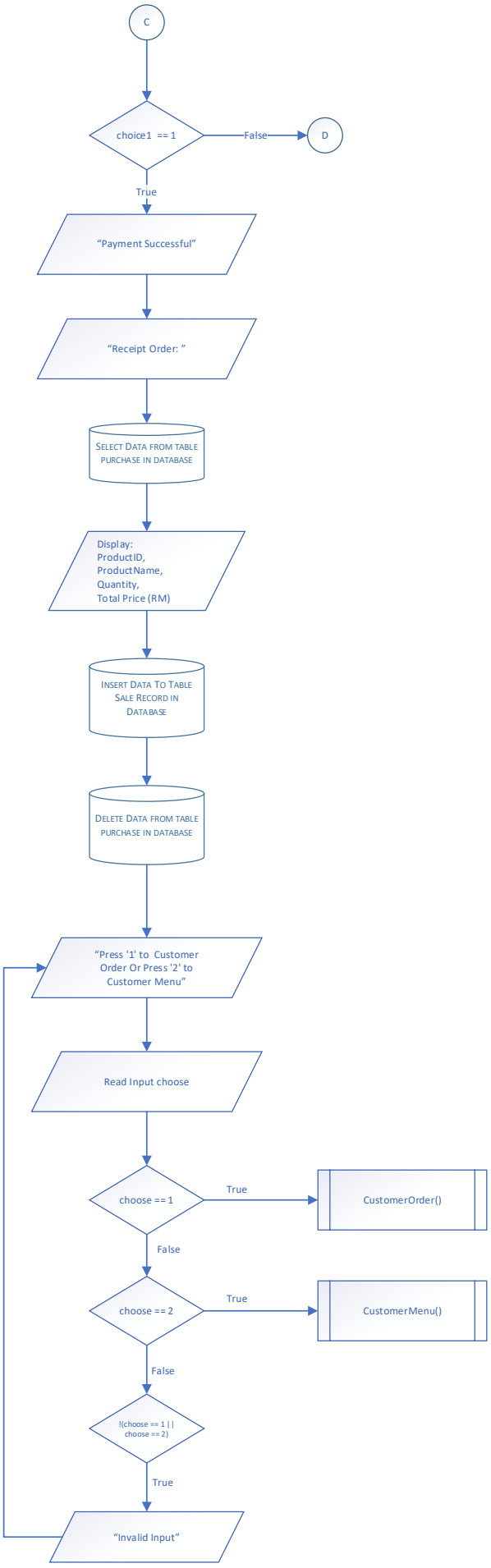

*Figure 15: Flowchart Customer Menu*

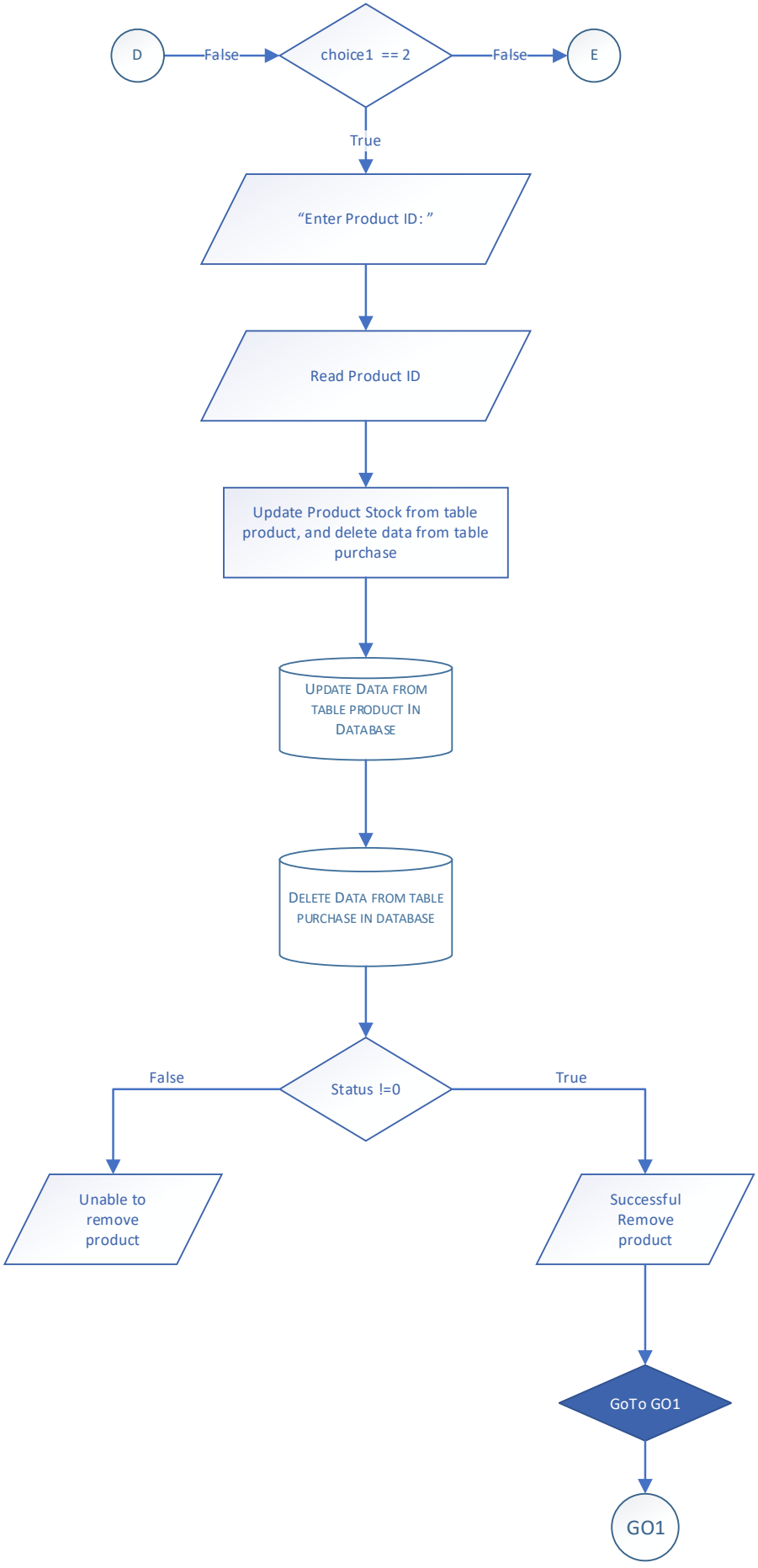
### 3.1.15 Customer Order




Product ID
“Enter Product ID: ”
Read Product ID
SELECT DATA FROM TABLE PRODUCT IN DATABASE
Display:
ProductID,
ProductName,
ProductStock
“Quantity: ”
Read quantity
Display:
Receipt Order,
Product ID,
Product Name,
Product Price,
Purchase Quantity
purchase->totalPrice =
calculateTotalPrice(product->productPrice,
purchase->purchaseQuantity)
CalculateTotalPrice
totalPrice = price * quantity;
“Total Price: RM” = totalPrice
B


B
INSERT DATA TO TABLE PURCHASE IN DATABASE
UpdateStock
product->productStock = UpdateDeductStock(purchase->purchaseQuantity, product)
UpdateDeductStock
productStock = productStock - quantity
UPDATE DATA FROM TABLE PRODUCT IN DATABASE
"Purchase another item?
Y) Yes
N) No"
Invalid Input
Read Input chooseDecision
chooseDecision == 'Y' || chooseDecision == 'y'
False
chooseDecision == 'N' || chooseDecision == 'n'
False
!(chooseDecision == 'N' || chooseDecision == 'n' || chooseDecision == 'Y' || chooseDecision == 'y')
True
Buy Again
True
SELECT DATA FROM TABLE PURCHASE IN DATABASE
True
"Invalid Input"
Display:
ProductID,
ProductName,
Quantity,
Total Price (RM)
GoTo invalidInput
GO1
"Would you like to proceed payment?
1) Proceed Payment
2) Delete Order Item
3) Cancel Order"
Invalid Input
Read Input choice1
C


C
choice1 == 1
False
D
True
"Payment Successful"
"Receipt Order: "
SELECT DATA FROM TABLE PURCHASE IN DATABASE
Display:
ProductID,
ProductName,
Quantity,
Total Price (RM)
INSERT DATA TO TABLE SALE RECORD IN DATABASE
DELETE DATA FROM TABLE PURCHASE IN DATABASE
"Press '1' to Customer Order Or Press '2' to Customer Menu"
Read Input choose
choose == 1
True
CustomerOrder()
False
choose == 2
True
CustomerMenu()
False
!(choose == 1 | | choose == 2)
True
"Invalid Input"


D
False
choice1 == 2
False
E
True
"Enter Product ID: "
Read Product ID
Update Product Stock from table product, and delete data from table purchase
UPDATE DATA FROM TABLE PRODUCT IN DATABASE
DELETE DATA FROM TABLE PURCHASE IN DATABASE
False
Status !=0
True
Unable to remove product
Successful Remove product
GoTo GO1
GO1

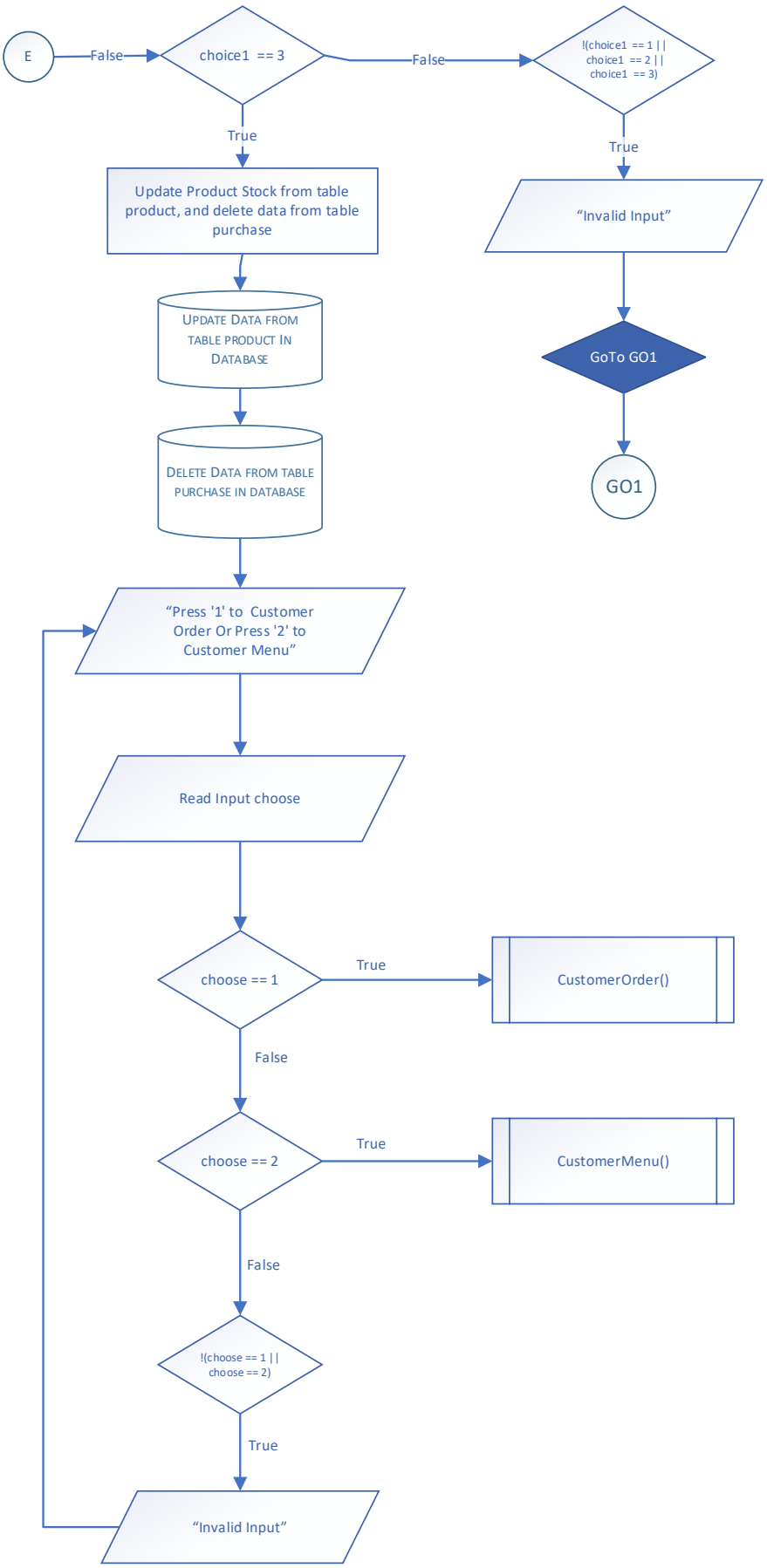



*Figure 16: Flowchart Customer Order*

### 3.1.16 Customer Search Receipt No

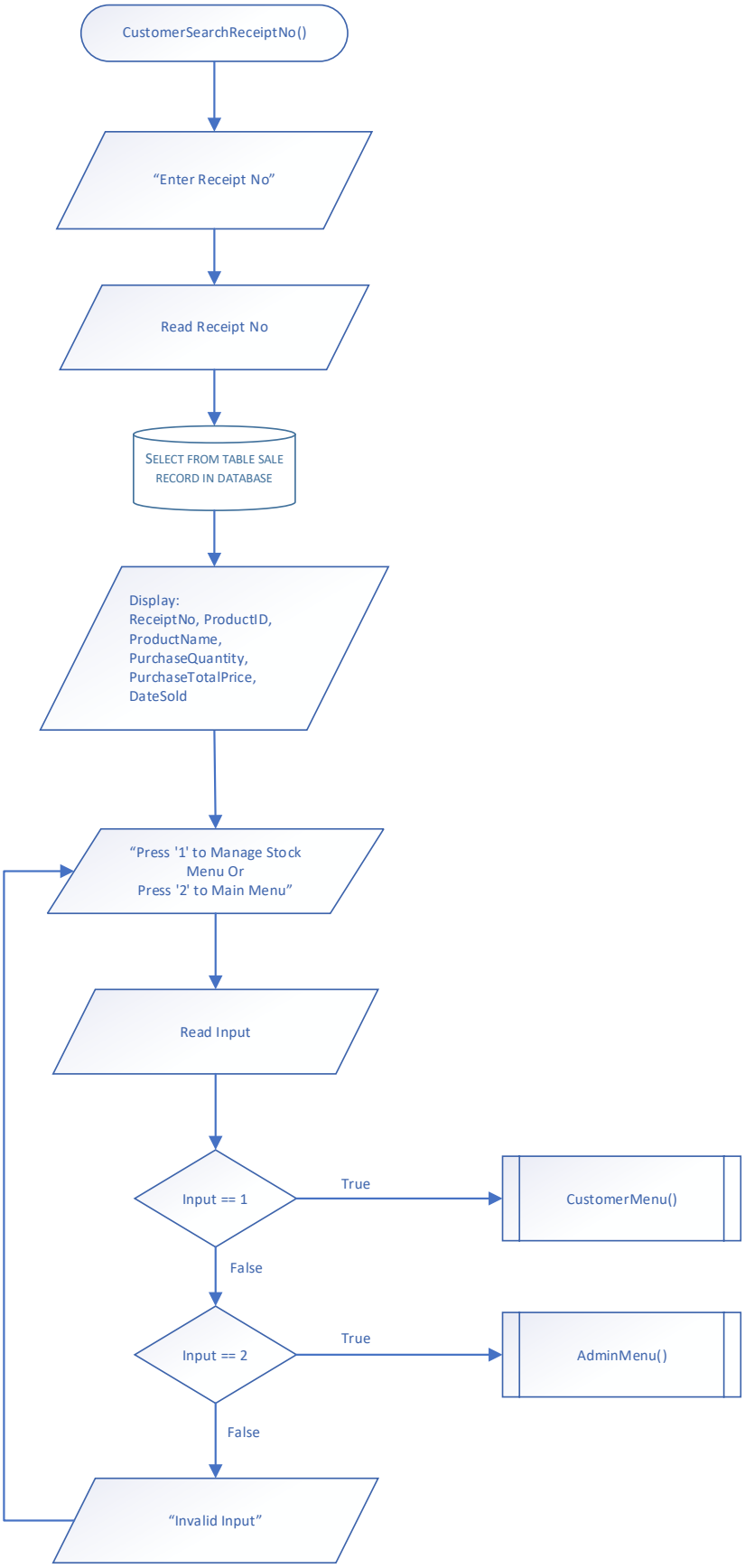



*Figure 17: Flowchart Search Receipt No*

### 3.1.17 Sale Report Menu

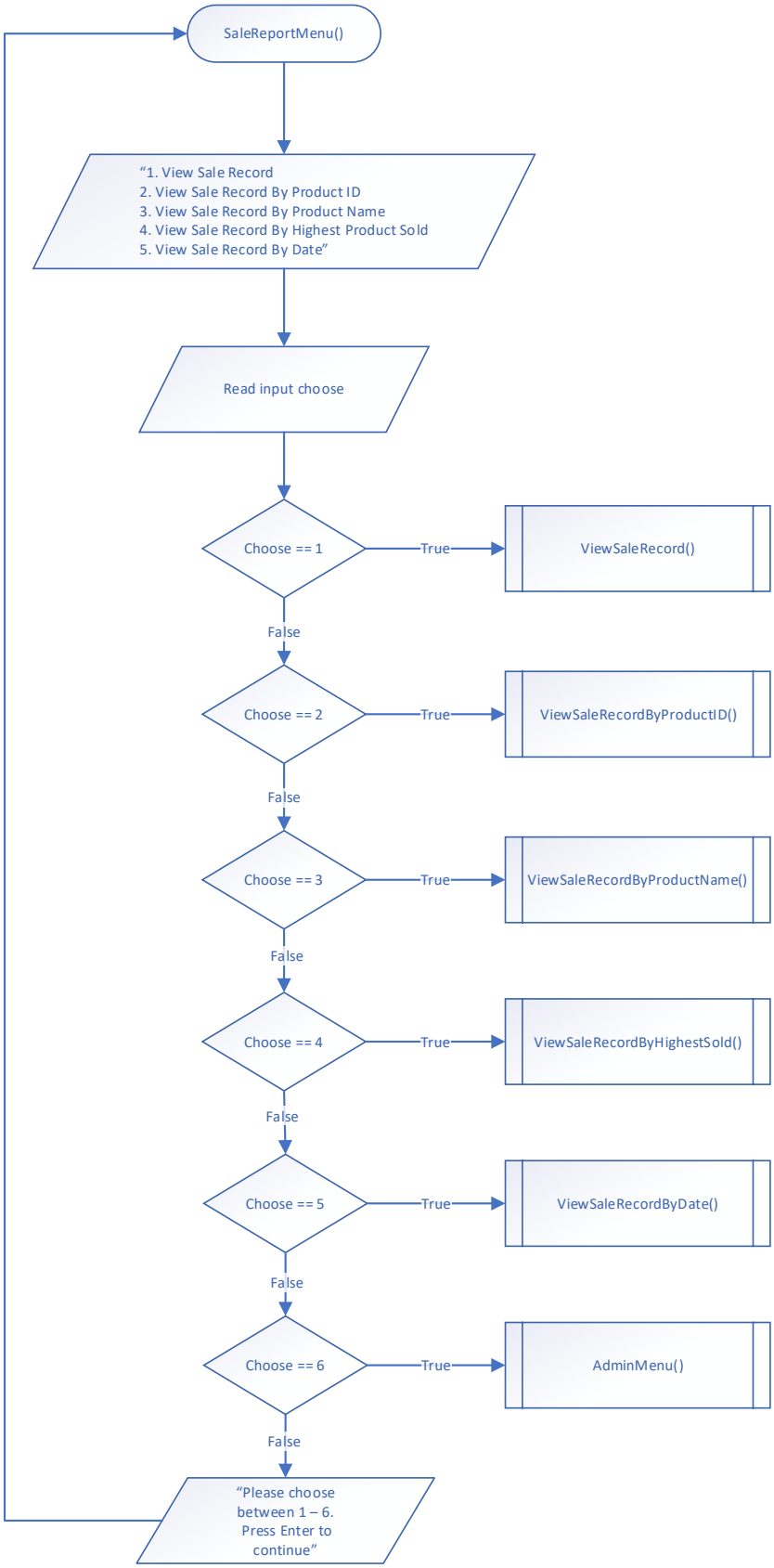



*Figure 18: Flowchart Sale Report Menu*

### 3.1.18 View Sale Record

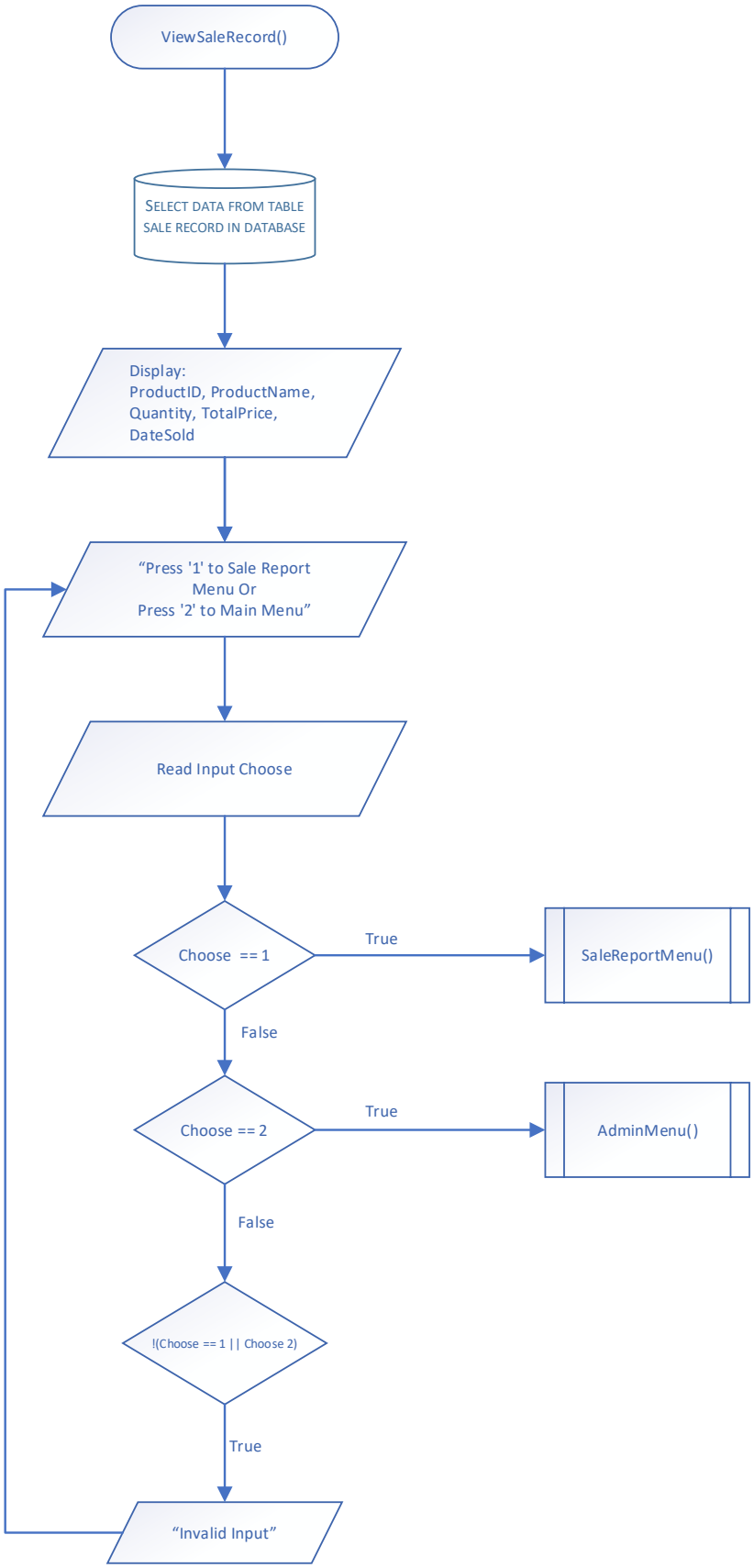



*Figure 19: Flowchart View Sale Record*

### 3.1.19 View Sale Record by Product ID

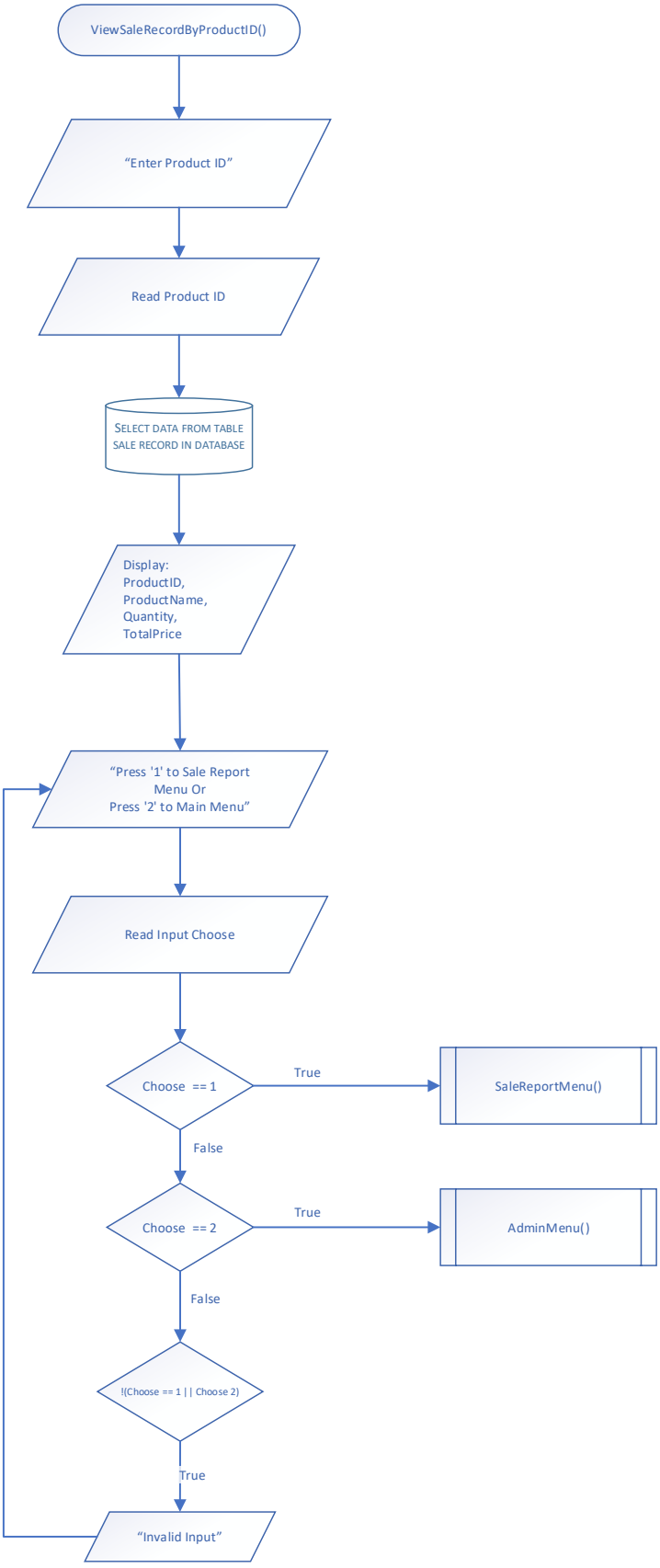



*Figure 20: Flowchart View Sale Record by Product ID*

### 3.1.20 View Sale Record by Product Name

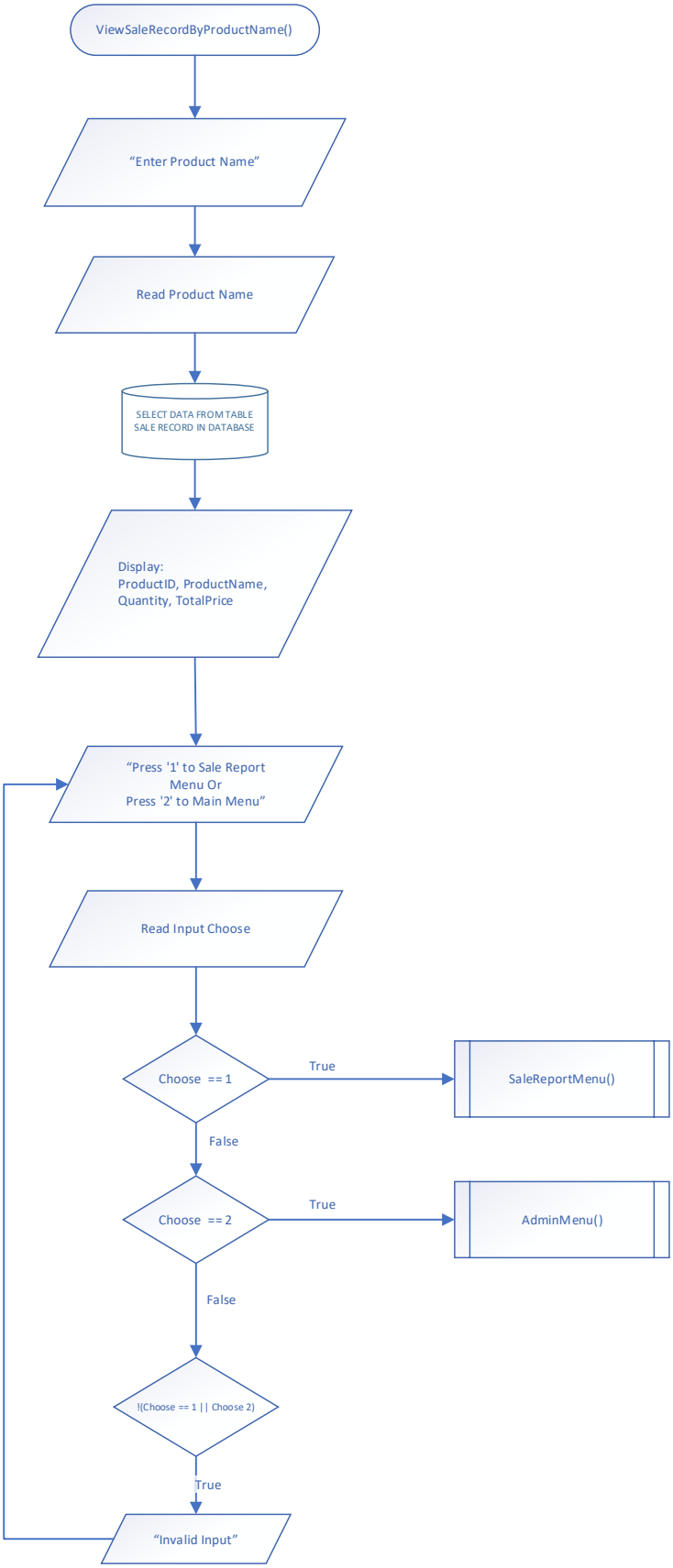



*Figure 21: Flowchart View Sale Record by Product Nam*

### 3.1.21 View Sale Record by Highest Sold

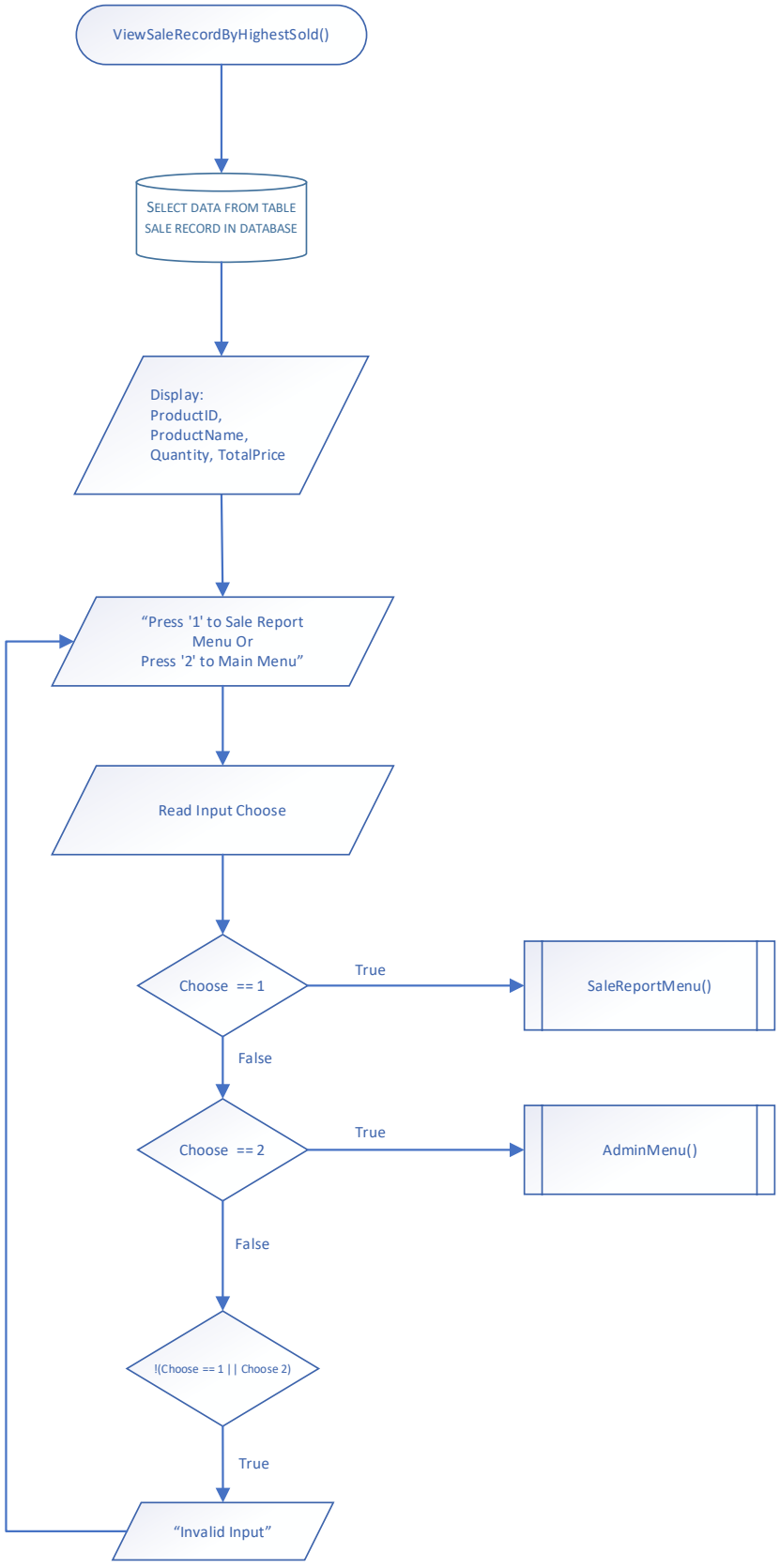



*Figure 22: Flowchart View Sale Record by Highest Sold*

### 3.1.22 View Sale Record by Date

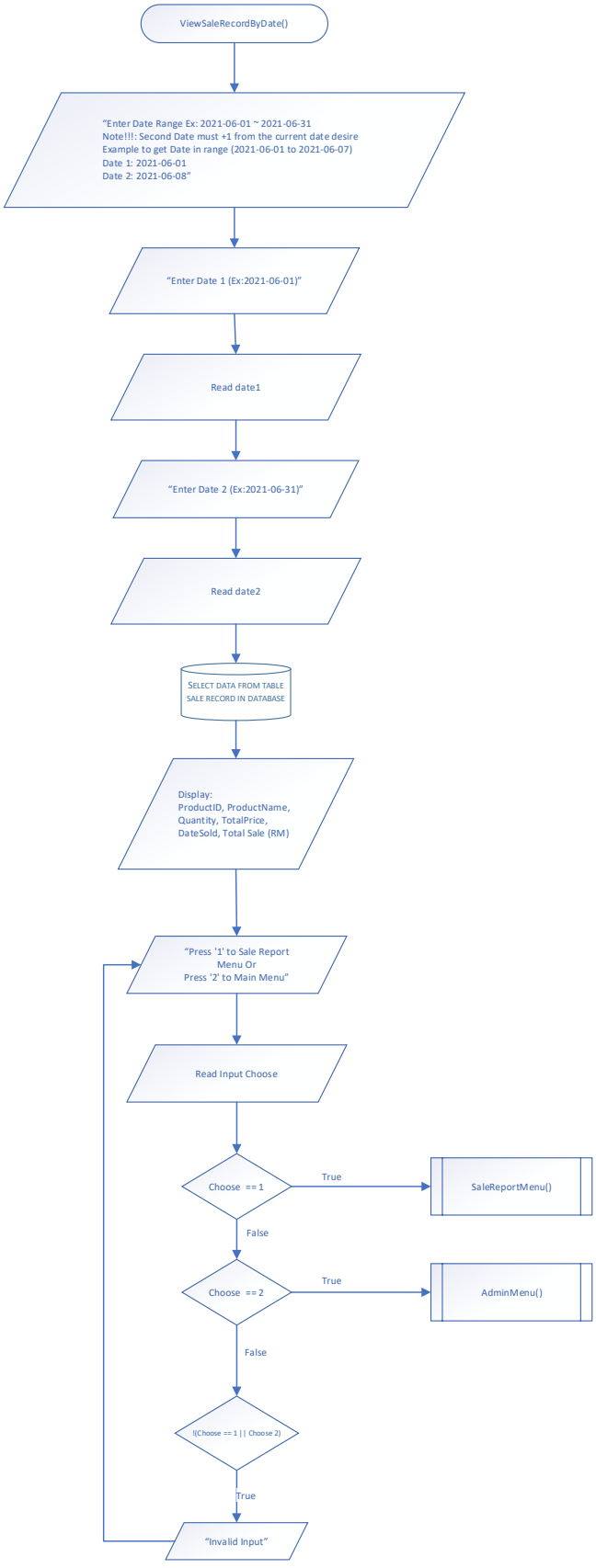



*Figure 23: Flowchart View Sale Record by Date*

### 3.2 Entity Relation Diagram (ERD)

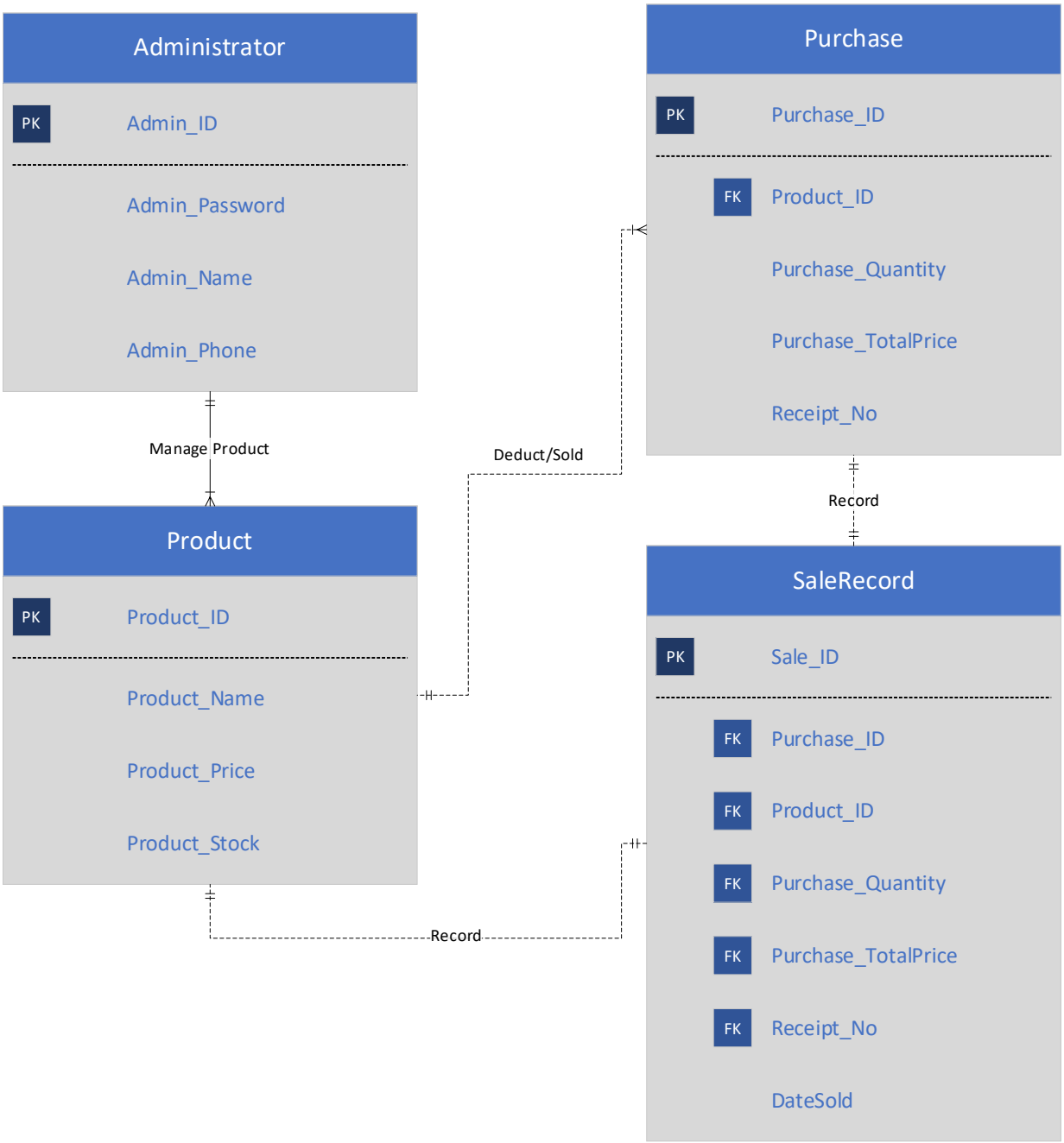



*Figure 24: ERD Snack Food Management System*

### 3.3 Data Dictionary

| Table | Product | | |
|---|---|---|---|
| **Name** | **Type** | **Constraint** | **Description** |
| Product_ID | INTEGER | Primary Key | Product ID |
| Product_Name | VARCHAR(30) | Not Null | Product Name |
| Product_Price | DOUBLE | Not Null | Product Price |
| Product_Stock | INTEGER | Not Null | Product Stock |

*Table 1: Data Dictionary for Table Product*

| Table | Administrator | | |
|---|---|---|---|
| **Name** | **Type** | **Constraint** | **Description** |
| Admin_ID | VARCHAR(15) | Primary Key | Administrator ID |
| Admin_Password | VARCHAR(15) | Not Null | Administrator Password |
| Admin_Name | VARCHAR(30) | Not Null | Administrator Name |
| Admin_Phone | CHAR(12) | Not Null, Unique | Administrator Phone |

*Table 2: Data Dictionary for Table Administrator*

| Table | Purchase | | |
|---|---|---|---|
| **Name** | **Type** | **Constraint** | **Description** |
| Purchase_ID | INTEGER | Primary Key | Purchase ID |
| Product_ID | INTEGER | Not Null | Product ID |
| Purchase_Quantity | INTEGER | Not Null | Quantity Purchase |
| Purchase_TotalPrice | DOUBLE | Not Null | Total Price Purchase |
| Receipt_No | INTEGER | Not Null | Receipt Number |

*Table 3: Data Dictionary for Table Purchase*

| Table | SaleRecord | | |
|---|---|---|---|
| **Name** | **Type** | **Constraint** | **Description** |
| Sale_ID | INTEGER | Primary Key | Sale ID |
| Purchase_ID | INTEGER | Not Null | Record of Purchase ID |
| Product_ID | INTEGER | Not Null | Record of Product ID |
| Purchase_Quantity | INTEGER | Not Null | Record of Quantity Purchase |
| Purchase_TotalPrice | DOUBLE | Not Null | Record of Total Price Purchase |
| Receipt_No | INTEGER | Not Null | Record of Receipt Number |
| DateSold | DATETIME | Not Null, Default, Current_TimeStamp() | Date of the product sold |

*Table 4: Data Dictionary for Table Sale Record*

### 3.4 Interface Design

This section is the interface design for the project for administrator, product , customer order menu, and sale report.

#### 3.4.1 Administrator

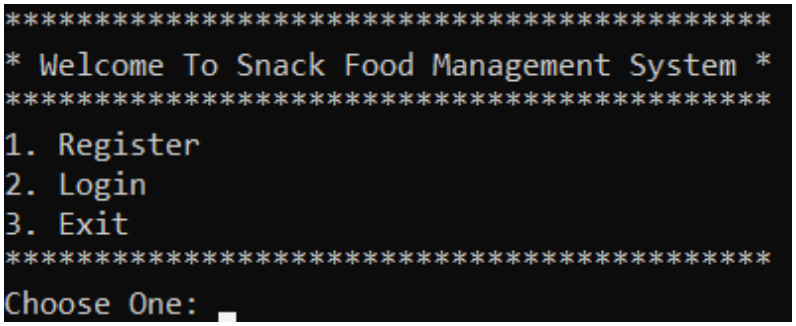



*Figure 25: Authentication Menu*

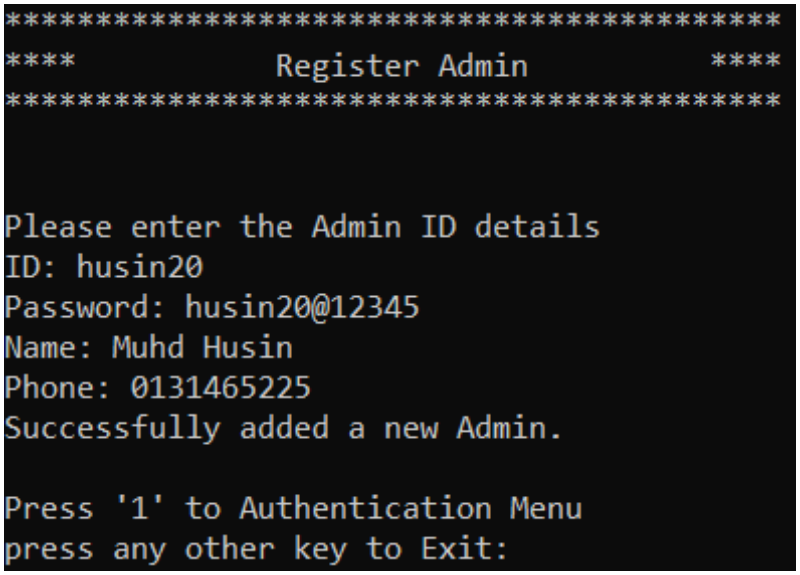



*Figure 26: Register Menu*

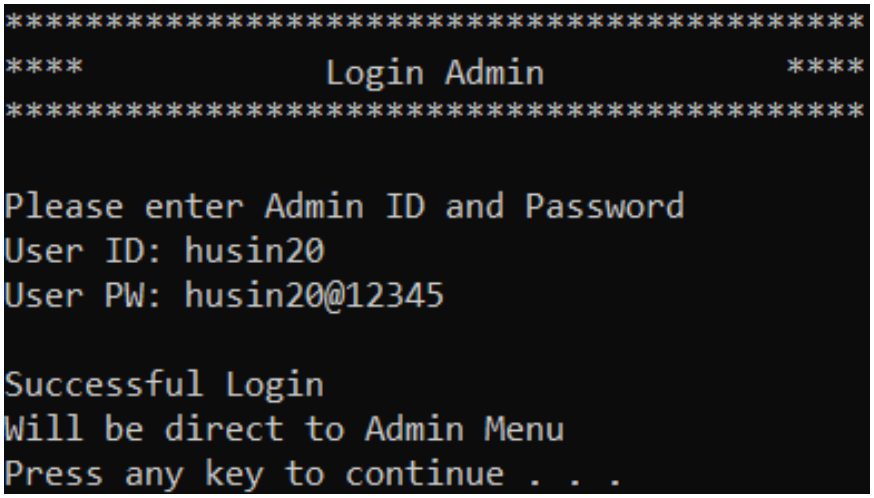



*Figure 27: Login Menu*

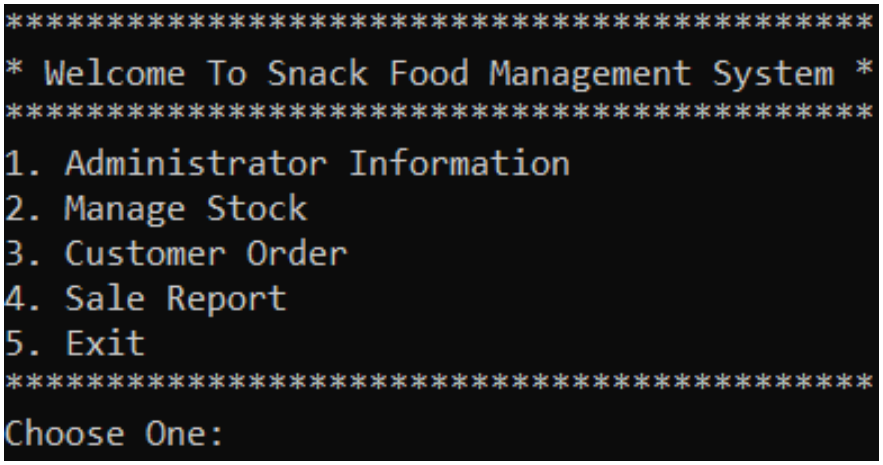



*Figure 28: Main Menu*

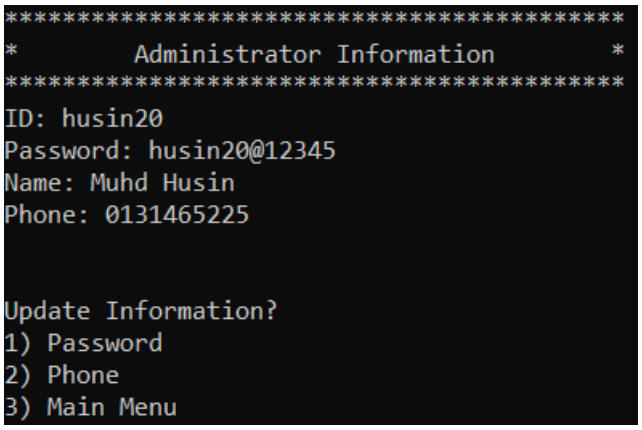



*Figure 29: Administrator Information Menu*

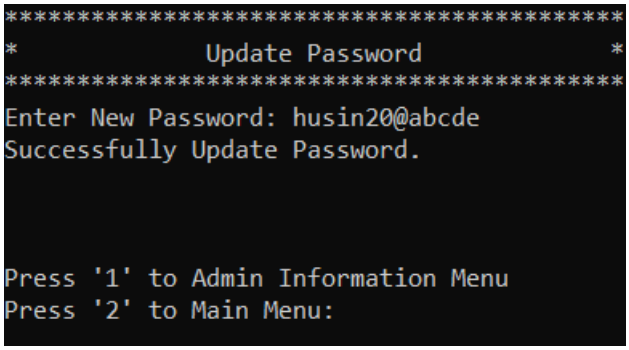



*Figure 30: Admin Update Password*

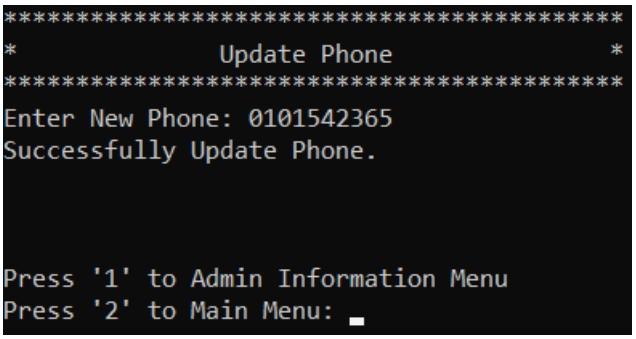



*Figure 31: Admin Update Phone Number*

### 3.4.2 Product

```
*********************************************
****          Manage Product             ****
*********************************************
1. View Stock
2. Add Product
3. Delete Product
4. Update Stock
5. Main Menu
*********************************************
Choose One:
```

*Figure 32: Manage Product Menu*

```
*****************************************************************
****                      View Stock                         ****
*****************************************************************
-----------------------------------------------------------------
|No| ID |              Name                  | Price | Stock |
-----------------------------------------------------------------
|1.|1000|       Kerepek Ubi Pedas (500g)     |  10   |   17  |
|2.|1001|       Kerepek Ubi Pedas (1000g)    |  20   |   26  |
|3.|1002|       Kerepek Ubi Jagung (500g)    |  10   |   19  |
|4.|1003|       Kerepek Ubi Jagung (1000g)   |  20   |   37  |
|5.|1006|       Kerepek Ubi Masin (500g)     |  10   |   20  |
|6.|1007|       Kerepek Ubi Masin (1000g)    |  20   |   20  |
|7.|1008|       Kerepek Ubi BBQ   (500g)     |  10   |   20  |
|8.|1009|       Kerepek Ubi BBQ   (1000g)    |  20   |   20  |
|9.|1010|       Kerepek Ubi Cheese (500g)    |  12   |   20  |
|10.|1011|      Kerepek Ubi Cheese (1000g)   |  12   |   20  |
|11.|1012|      Kerepek Ubi Racik (500g)     |  10   |   20  |
|12.|1013|      Kerepek Ubi Racik (1000g)    |  20   |   20  |
-----------------------------------------------------------------



Press '1' to Manage Stock Menu
Press '2' to Main Menu:
```

*Figure 33: View Stock Menu*

```
*********************************************
****              Add Stock                ****
*********************************************
Please enter the Stock details
Product Name: Kerepek Ubi Black Paper (500g)
Product Price: 8.00
Product Stock: 20
Successfully added a new product.

Press '1' to Manage Stock Menu
Press '2' to Main Menu:
```

*Figure 34: Add Stock Menu*

```
************************************************************
****                    Remove Stock                    ****
************************************************************
------------------------------------------------------------
|No| ID |              Name                  | Price | Stock |
------------------------------------------------------------
|1.|1000|       Kerepek Ubi Pedas (500g)     |  10   |   17  |
|2.|1001|       Kerepek Ubi Pedas (1000g)    |  20   |   26  |
|3.|1002|       Kerepek Ubi Jagung (500g)    |  10   |   19  |
|4.|1003|       Kerepek Ubi Jagung (1000g)   |  20   |   37  |
|5.|1006|       Kerepek Ubi Masin (500g)     |  10   |   20  |
|6.|1007|       Kerepek Ubi Masin (1000g)    |  20   |   20  |
|7.|1008|       Kerepek Ubi BBQ   (500g)     |  10   |   20  |
|8.|1009|       Kerepek Ubi BBQ   (1000g)    |  20   |   20  |
|9.|1010|       Kerepek Ubi Cheese (500g)    |  12   |   20  |
|10.|1011|      Kerepek Ubi Cheese (1000g)   |  12   |   20  |
|11.|1012|      Kerepek Ubi Racik (500g)     |  10   |   20  |
|12.|1013|      Kerepek Ubi Racik (1000g)    |  20   |   20  |
|13.|1014|      Kerepek Ubi Black Paper (500g)|  8    |   20  |
------------------------------------------------------------

Please Enter Product ID to be remove: 1014
Successfully remove product.

Press '1' to Manage Stock Menu
Press '2' to Main Menu:
```

*Figure 35: Delete Stock Menu*

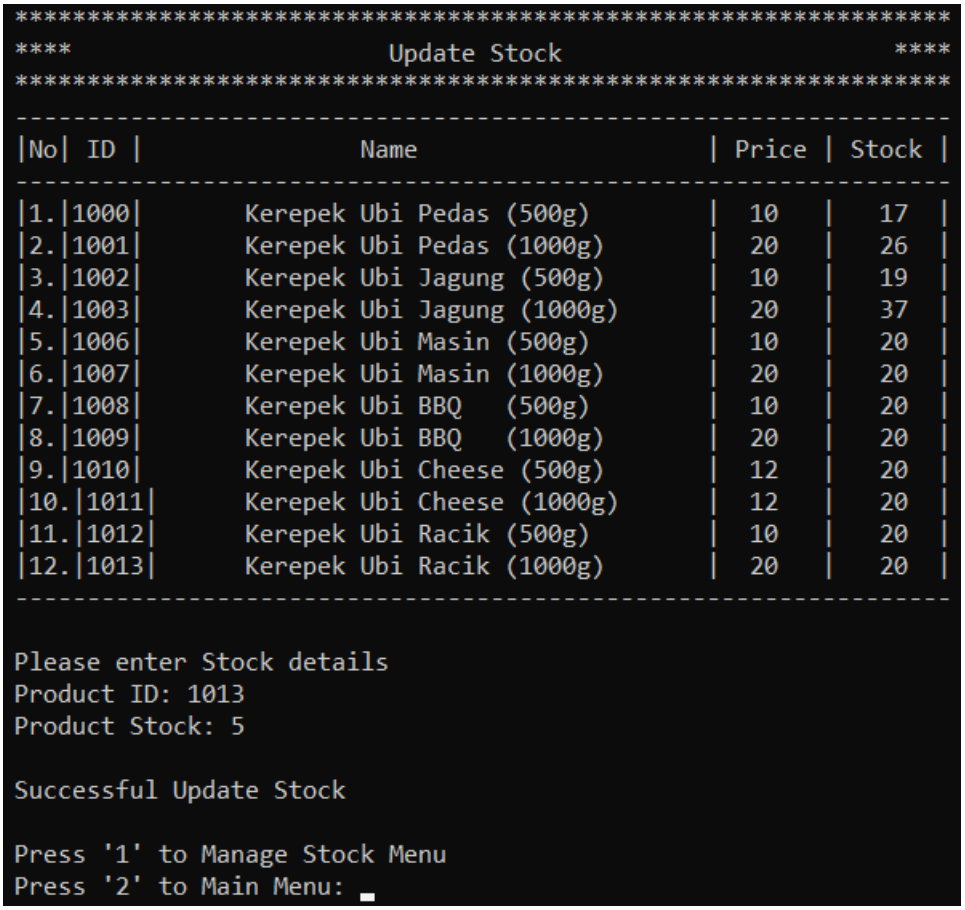



*Figure 36: Updating Stock*

### 3.4.3 Customer Order Menu

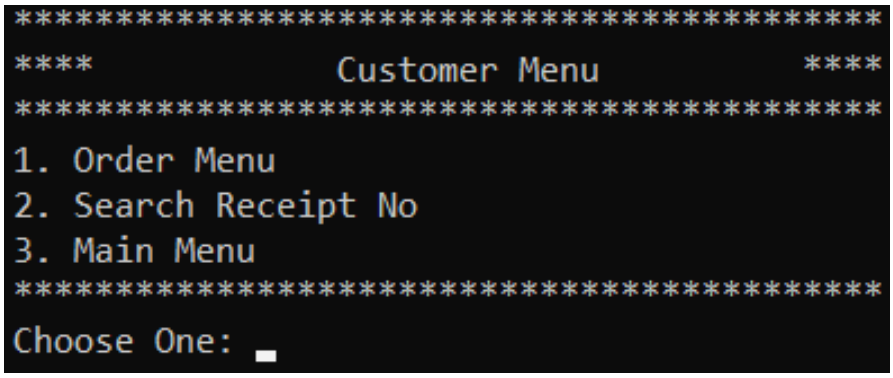



*Figure 37: Customer Menu*

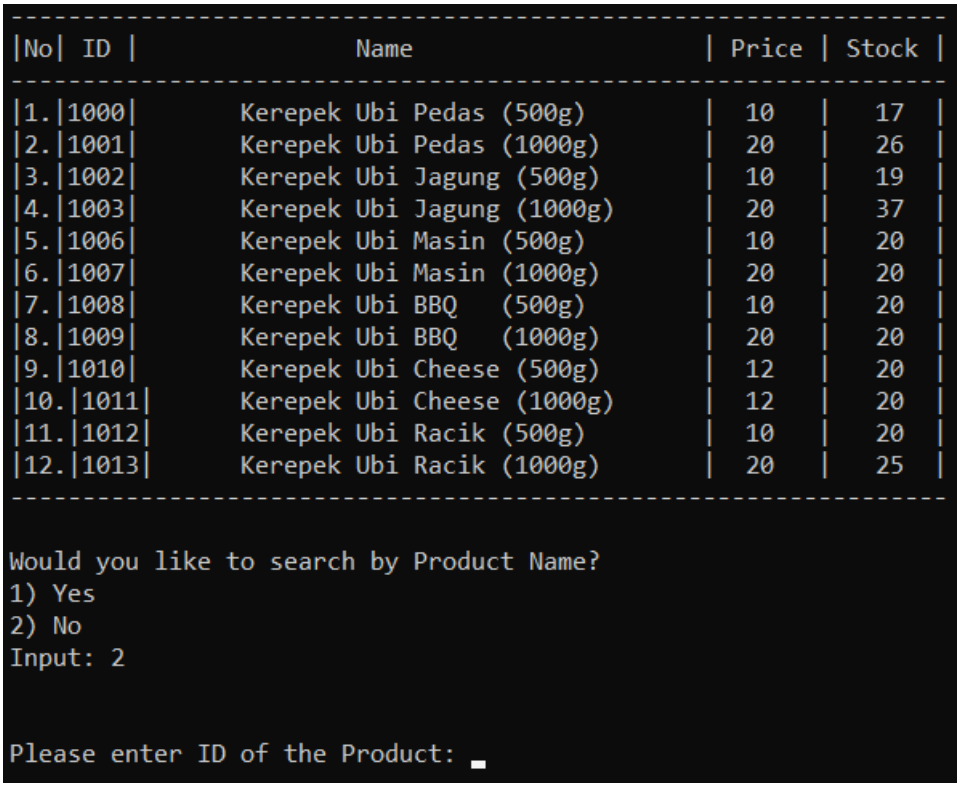



*Figure 38: Customer Order Menu*

```
-------------------------------------------------------------------------
| Product ID |           Product Name          | Quantity | Total Price|
-------------------------------------------------------------------------
|   1000     |   Kerepek Ubi Pedas (500g)      |     2    |     20     |
|   1002     |   Kerepek Ubi Jagung (500g)     |     3    |     30     |
|   1008     |   Kerepek Ubi BBQ   (500g)      |     4    |     40     |
-------------------------------------------------------------------------
                                                Total:   RM90
-------------------------------------------------------------------------
Would you like to proceed payment?
1) Proceed Payment
2) Delete Order Item
3) Cancel Order

Enter Input:
```

*Figure 39: After Done Order*

```
-------------------------------------------------------------------------
 Receipt Order: 3172                          Sun Jun 20 17:51:50 2021
-------------------------------------------------------------------------
| Product ID |           Product Name          | Quantity | Total Price|
-------------------------------------------------------------------------
|   1000     |   Kerepek Ubi Pedas (500g)      |     2    |     20     |
|   1002     |   Kerepek Ubi Jagung (500g)     |     3    |     30     |
|   1008     |   Kerepek Ubi BBQ   (500g)      |     4    |     40     |
-------------------------------------------------------------------------
                                                Total:   RM90
-------------------------------------------------------------------------


Press '1' to  Customer Order
Press '2' to Customer Menu:
```

*Figure 40: Customer Receipt*

```
*************************************************************************************************************************************
****                                          Customer Receipt Order                                                           ****
*************************************************************************************************************************************
Please input Receipt No: 3172
-------------------------------------------------------------------------------------------------------------------------------------
|  Receipt No  | Product ID |          Product Name          | Product Price |         Quantity          | Total Price (RM) |      Date Sold      |
-------------------------------------------------------------------------------------------------------------------------------------
|     3172     |    1000    |    Kerepek Ubi Pedas (500g)    |      10       |             2             |        20        | 2021-06-20 17:51:50 |
|     3172     |    1002    |    Kerepek Ubi Jagung (500g)   |      10       |             3             |        30        | 2021-06-20 17:51:50 |
|     3172     |    1008    |    Kerepek Ubi BBQ   (500g)    |      10       |             4             |        40        | 2021-06-20 17:51:50 |
-------------------------------------------------------------------------------------------------------------------------------------
                                                Total Price: RM90
-------------------------------------------------------------------------------------------------------------------------------------


Press '1' to  Customer Menu
Press '2' to Main Menu:
```

*Figure 41: Search Receipt Order*

### 3.4.4 Sale Report

```
*****************************************
****            Sale Report          ****
*****************************************
1. View Sale Record
2. View Sale Record By Product ID
3. View Sale Record By Product Name
4. View Sale Record By Highest Product Sold
5. View Sale Record By Date
6. Main Menu
*****************************************
Choose One: _
```

*Figure 42: Sale Report Menu*

```
*****************************************************************************************************
****                                        Sale Record                                          ****
*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
| Product ID |         Product Name          | Product Price |    Quantity    | Total Price |      Date Sold      |
-----------------------------------------------------------------------------------------------------
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        2       |     20      | 2021-06-04 12:31:45 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        2       |     40      | 2021-06-04 12:31:45 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-04 12:31:45 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-04 12:32:55 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-04 12:32:55 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        1       |     20      | 2021-06-04 12:32:55 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-04 18:57:33 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-04 18:57:33 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-04 18:57:33 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-04 21:42:31 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-04 21:42:31 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-04 21:43:13 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        1       |     20      | 2021-06-04 21:43:13 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        2       |     40      | 2021-06-05 17:32:04 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        2       |     40      | 2021-06-05 17:52:24 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        2       |     20      | 2021-06-05 22:20:07 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        2       |     40      | 2021-06-05 22:20:07 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-05 22:20:07 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-05 22:46:56 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-05 22:46:56 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-05 22:46:56 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-06 01:02:42 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-06 01:02:42 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        1       |     20      | 2021-06-06 16:38:30 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-06 18:17:46 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        1       |     20      | 2021-06-06 18:17:46 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-06 18:25:15 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-09 10:03:44 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-09 10:03:44 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-09 10:03:44 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-09 10:03:44 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-09 12:01:38 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        3       |     60      | 2021-06-09 12:01:38 |
|   1003     |    Kerepek Ubi Jagung (1000g) |      20       |        2       |     40      | 2021-06-09 12:01:38 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-09 12:02:44 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-09 13:23:07 |
|   1001     |    Kerepek Ubi Pedas (1000g)  |      20       |        1       |     20      | 2021-06-09 13:23:07 |
|   1002     |    Kerepek Ubi Jagung (500g)  |      10       |        1       |     10      | 2021-06-09 14:15:28 |
|   1000     |    Kerepek Ubi Pedas (500g)   |      10       |        1       |     10      | 2021-06-09 14:15:28 |
-----------------------------------------------------------------------------------------------------
                                        Total Sale: RM730
-----------------------------------------------------------------------------------------------------
```

*Figure 43: View Sale Record*

```
**********************************************************************************************************************
****                                                Sale Record                                                       ****
**********************************************************************************************************************
-------------------------------------------------------------
|No| ID |              Name                  | Price | Stock |
-------------------------------------------------------------
|1.|1000|       Kerepek Ubi Pedas (500g)     |  10   |   5   |
|2.|1001|       Kerepek Ubi Pedas (1000g)    |  20   |   5   |
|3.|1002|       Kerepek Ubi Jagung (500g)    |  10   |   4   |
|4.|1003|       Kerepek Ubi Jagung (1000g)   |  20   |   6   |
-------------------------------------------------------------
Enter Product ID: 1000
-----------------------------------------------------------------------------------------------------------------
| Product ID |          Product Name           | Product Price |    Total Quantity Sold      | Total Price (RM) |
-----------------------------------------------------------------------------------------------------------------
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |             14              |       140        |
-----------------------------------------------------------------------------------------------------------------


Press '1' to  Sale Report Menu
Press '2' to Main Menu:
```

*Figure 44: View Sale Record by Product ID*

```
**********************************************************************************************************************
****                                                Sale Record                                                       ****
**********************************************************************************************************************
-------------------------------------------------------------
|No| ID |              Name                  | Price | Stock |
-------------------------------------------------------------
|1.|1000|       Kerepek Ubi Pedas (500g)     |  10   |   5   |
|2.|1001|       Kerepek Ubi Pedas (1000g)    |  20   |   5   |
|3.|1002|       Kerepek Ubi Jagung (500g)    |  10   |   4   |
|4.|1003|       Kerepek Ubi Jagung (1000g)   |  20   |   6   |
-------------------------------------------------------------
Enter Product Name: Pedas
-----------------------------------------------------------------------------------------------------------------
| Product ID |          Product Name           | Product Price |    Total Quantity Sold      | Total Price (RM) |
-----------------------------------------------------------------------------------------------------------------
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |             14              |       140        |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |             17              |       340        |
-----------------------------------------------------------------------------------------------------------------

Successful Search


Press '1' to  Sale Report Menu
Press '2' to Main Menu:
```

*Figure 45: View Sale By Product Name*

```
**********************************************************************************************************************
****                                                Sale Record                                                       ****
**********************************************************************************************************************
-----------------------------------------------------------------------------------------------------------------
| Product ID |          Product Name           | Product Price |    Total Quantity Sold      | Total Price (RM) |
-----------------------------------------------------------------------------------------------------------------
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |             17              |       340        |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |             8               |       160        |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |             14              |       140        |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |             9               |       90         |
-----------------------------------------------------------------------------------------------------------------

Successful Search


Press '1' to  Sale Report Menu
Press '2' to Main Menu:
```

*Figure 46: View Sale Recrod By Highest Product Sold*

```
**********************************************************************************************************************************************
****                                                    Sale Record                                                                       ****
**********************************************************************************************************************************************

Enter Date Range Ex: 2021-06-01 ~ 2021-06-31

Note!!!: Second Date must +1 from the current date desire.

Example to get Date in range (2021-06-01 to 2021-06-07)
Date 1: 2021-06-01
Date 2: 2021-06-08


Enter Date 1 (Ex:2021-06-01): 2021-06-01
Enter Date 2 (Ex:2021-06-31): 2021-06-08
----------------------------------------------------------------------------------------------------------------------------------------------
| Product ID |          Product Name           | Product Price |    Total Quantity Sold      | Total Price (RM) |      Date Sold      |
----------------------------------------------------------------------------------------------------------------------------------------------
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              2              |        20        | 2021-06-01 12:31:45 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              2              |        40        | 2021-06-02 12:31:45 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-03 12:31:45 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              1              |        10        | 2021-06-04 12:32:55 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-04 12:32:55 |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |              1              |        20        | 2021-06-04 12:32:55 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              1              |        10        | 2021-06-04 18:57:33 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              1              |        20        | 2021-06-04 18:57:33 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-04 18:57:33 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              1              |        10        | 2021-06-04 21:42:31 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              1              |        20        | 2021-06-04 21:42:31 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-04 21:43:13 |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |              1              |        20        | 2021-06-04 21:43:13 |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |              2              |        40        | 2021-06-05 17:32:04 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              2              |        40        | 2021-06-05 17:52:24 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              2              |        20        | 2021-06-05 22:20:07 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              2              |        40        | 2021-06-05 22:20:07 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-05 22:20:07 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              1              |        10        | 2021-06-05 22:46:56 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              1              |        20        | 2021-06-05 22:46:56 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-05 22:46:56 |
|   1000     |     Kerepek Ubi Pedas (500g)    |      10       |              1              |        10        | 2021-06-06 01:02:42 |
|   1001     |     Kerepek Ubi Pedas (1000g)   |      20       |              1              |        20        | 2021-06-06 01:02:42 |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |              1              |        20        | 2021-06-07 16:38:30 |
|   1003     |     Kerepek Ubi Jagung (1000g)  |      20       |              1              |        20        | 2021-06-07 18:17:46 |
|   1002     |     Kerepek Ubi Jagung (500g)   |      10       |              1              |        10        | 2021-06-07 18:25:15 |
----------------------------------------------------------------------------------------------------------------------------------------------
                                                Total Sale: RM480
----------------------------------------------------------------------------------------------------------------------------------------------
```

*Figure 47: View Sale Record By Date*

# CHAPTER 4

# IMPLEMENTATION

## 4.1 Programming Technique

This section is for code snippet of programming technique used on this project. Some of it are function, selection, repetition, class and error handling.

### 4.1.1 Function

```
void AuthenticationMenu();
void Register();
void Login();
```

```
void AuthenticationMenu()
{
    int chooseOneFromMenu = 0;
    char exitConfirmation;

    system("cls");
    cout << "\t\t\t\t****************************************" << endl;
    cout << "\t\t\t\t* Welcome To Snack Food Management System *" << endl;
    cout << "\t\t\t\t****************************************" << endl;
    cout << "\t\t\t\t1. Register" << endl;
    cout << "\t\t\t\t2. Login" << endl;
    cout << "\t\t\t\t3. Exit " << endl;
    cout << "\t\t\t\t****************************************" << endl;

    cout << "\t\t\t\tChoose One: ";
    cin >> chooseOneFromMenu;
```

*Figure 48: Programming Technique Function*

### 4.1.2 Selection

```
void CustomerMenu()
{
    int chooseOneFromMenu = 0;

    system("cls");
    cout << "\t\t\t\t*****************************************" << endl;
    cout << "\t\t\t\t****           Customer Menu          ****" << endl;
    cout << "\t\t\t\t*****************************************" << endl;
    cout << "\t\t\t\t1. Order Menu" << endl;
    cout << "\t\t\t\t2. Search Receipt No" << endl;
    cout << "\t\t\t\t3. Main Menu " << endl;
    cout << "\t\t\t\t*****************************************" << endl;

    cout << "\t\t\t\tChoose One: ";
    cin >> chooseOneFromMenu;

    switch (chooseOneFromMenu)
    {
    case 1:
        CustomerOrder();
        break;
    case 2:
        CustomerSearchReceiptNo();
        break;
    case 3:
        AdminMenu(adminID);
        break;
    default:
        cout << "\t\t\t\tPlease choose between 1 - 2. Press Enter To Continue...";
        system("cls");
        CustomerMenu();
        break;
    }
}
```

*Figure 49: Selection - Switch Case*

```
cout << "\n\t\t\t\tPress '1' to Manage Stock Menu";
cout << "\n\t\t\t\tPress '2' to Main Menu: ";

cin >> choose;
if (choose == 1)
{
    ManageStockMenu();
}
else if (choose == 2)
{
    AdminMenu(adminID);
}
else if (!(choose == 2 || choose == 1))
{
system("cls");
cout << "\t\t\t\tInvalid Input" << endl;
system("pause");
UpdateStock();
}
```

*Figure 50: Selection - If Else If*

### 4.1.3 Repetition

```
int i = 1;
cout << "\t\t\t\t" << "-------------------------------------------------------------------" << endl;
cout << "\t\t\t\t" << "|" << "No" << "|" << " ID " << "|" << "\t\tName" << "\t\t\t| Price |" << " Stock |" << endl;
cout << "\t\t\t\t" << "-------------------------------------------------------------------" << endl;
while (rs->next())
{
    product = new Product();

    product->productID = rs->getInt(1);
    product->productName = rs->getString(2);
    product->productPrice = rs->getDouble(3);
    product->productStock = rs->getInt(4);

    cout << "\t\t\t\t" <<"|" << i << ".";
    cout << "|" << product->productID << "|";
    cout << "\t" << product->productName;
    cout << "\t" << "|  " << product->productPrice << "\t|";
    cout << "    " << product->productStock << "\t|";
    cout << endl;
    i++;
}

cout << "\t\t\t\t" << "-------------------------------------------------------------------";
```

*Figure 51: Repetition - While*

### 4.1.4 Class

```
class Purchase
{
public:
    string productName;
    int productStock,productID,purchaseID,purchaseQuantity,receiptNo;
    double productPrice,totalPrice;

    string dateSold;
};
```

*Figure 52: Programming Technique Class*

### 4.1.5 Error Handling

```
int status = productManager.insertProduct(product);

if (status != 0)
    cout << "\t\t\t\tSuccessfully added a new product." << endl;
else
    cout << "\t\t\t\tUnable to add a new product." << endl;

cout << "\n\t\t\t\tPress '1' to Manage Stock Menu";
cout << "\n\t\t\t\tPress '2' to Main Menu: ";

cin >> choose;
if (choose == 1)
{
    ManageStockMenu();
}
else if (choose == 2)
{
    AdminMenu(adminID);
}
else if (!(choose == 2 || choose == 1))
{
system("cls");
cout << "\t\t\t\tInvalid Input" << endl;
system("pause");
AddStock();
}
```

*Figure 53: Error Handling*

## 4.2 Database Implementation

This section is for code snippet that used in this project for database implementation such as establish database connection, query insert, select, update, and delete.

### 4.2.1 Database Connection

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

    connection->setSchema("dbsnackfoodsystem");
}

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

    delete connection;
}

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

*Figure 54: Database Connection*

### 4.2.2 Insert into Database

```
int ProductManager::insertProduct(Product* product)
{
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("INSERT INTO PRODUCT (PRODUCT_NAME, PRODUCT_PRICE, PRODUCT_STOCK) VALUES(?, ?, ?)");

    ps->setString(1, product->productName);
    ps->setDouble(2, product->productPrice);
    ps->setInt(3, product->productStock);


    int status = ps->executeUpdate();
    delete ps;

    return status;
}
```

*Figure 55: Query Insert into Database*

### 4.2.3 Select from Database

```
Product* ProductManager::selectProduct(int productID)
{
    Product* product = NULL;
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("Select * From PRODUCT WHERE PRODUCT_ID = ?");

    ps->setInt(1, productID);

    ResultSet* rs = ps->executeQuery();

    if (rs->next())
    {
        product = new Product();
        product->productID = rs->getInt(1);
        product->productName = rs->getString(2);
        product->productPrice = rs->getDouble(3);
        product->productStock = rs->getInt(4);
    }

    delete rs;
    delete ps;

    return product;
}
```

*Figure 56: Query Select from Database*

### 4.2.4 Update to Database

```
int ProductManager::updateProduct(Product* product)
{
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("UPDATE PRODUCT SET PRODUCT_STOCK = ? where PRODUCT_ID = ?");

    ps->setInt(1, product->productStock);
    ps->setInt(2, product->productID);

    int statusUpdate = ps->executeUpdate();

    if (statusUpdate != 0)
    {
        cout << "\n\t\t\t\tSuccessful Update Stock" << endl;
    }
    else
    {
        cout << "\n\t\t\t\tUnable to Update Stock" << endl;
    }

    delete ps;
    return statusUpdate;

}
```

*Figure 57: Query Update Database*

### 4.2.5 Delete From Database

```
int ProductManager::deleteProduct(Product* product)
{

    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("DELETE FROM PRODUCT WHERE PRODUCT_ID = ?");

    ps->setInt(1, product->productID);

    int status = ps->executeUpdate();
    delete ps;

    return status;

}
```

*Figure 58: Query Delete from Database*

### 4.3 Security Implementation

This section is for the code snippet of the project for security implementation in the project such as login security.

#### 4.3.1 Login Security

```
void Login()
{
    system("cls");
    cout << "\t\t\t\t****************************************" << endl;
    cout << "\t\t\t\t****            Login Admin           ****" << endl;
    cout << "\t\t\t\t****************************************" << endl;
    cout << "\n\t\t\t\tPlease enter Admin ID and Password" << endl;

    cout << "\t\t\t\tUser ID: ";
    cin.ignore(1, '\n');
    getline(cin, administrator->adminID);

    cout << "\t\t\t\tUser PW: ";
    getline(cin, administrator->adminPassword);

    if (administratorManager.loginAdmin(administrator))
    {
        cout << "\n\t\t\t\tSuccessful Login" << endl;
        system("pause");

    }
    else
    {
        system("cls");
        cout << "\n\t\t\t\tUnsuccessful Login" << endl;
        AuthenticationMenu();
    }
    AdminMenu(administrator->adminID);

}
```

*Figure 59: Code Snippet for Login*

```
bool AdministratorManager::loginAdmin(Administrator* admin)
{
    DatabaseConnection dbConn;
    PreparedStatement* ps = dbConn.prepareStatement("SELECT * FROM Administrator WHERE admin_ID = ? AND admin_Password = ?");

    ps->setString(1, admin->adminID);
    ps->setString(2, admin->adminPassword);

    ResultSet* rs = ps->executeQuery();

    if (!(rs->next()))
    {
        cout << "\n\t\t\t\tWrong ID or Password. Please try again." << endl;
        system("pause");
        system("cls");
        return false;
    }
    delete rs;
    delete ps;

    return true;
}
```

*Figure 60: Code Snippet for Function Definition Login*

# CHAPTER 5
# CONCLUSION

## 5.1 Further Enhancements

For future enhancements, this system may have rating function for customer to rating the product they buy. This also will help business owner to stock or buy the product that have highest rating from customer and avoid to restock the product that have low rating from customer. Thus, will improve business owner profit gain from their business.

Beside that, this system may be able to apply to online ordering to booking or buy many or largest product from the shop where the customer will need to register their account and then they will need to login to their account to order online and then the customer need to choose which date will they take the product from the shop.

## 5.2 Conclusion

In conclusion, this system may help business owner to manage their worker where they will need to register account for non-existing worker and for existing worker, they just need to login to use the system. Next, the system will help in managing shop stock such as add new product to current existing stock, delete existing product, and update new arrival stock. After that, the system will have order menu to help customer to order item and to looking for the item they desired. The system will record the sale to database and will display the sale to business owner.

## REFERENCE


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

Zeratul Izzah Mohd Yusoh, Intan Ermahani A. Jalil, Mohd Sanusi Azmi, Nuridawati Mustafa, Sarni Suhaila Raim, Ummi Raba'ah Hashim (2020). *Data Structure And Algorithm*. Durian Tunggal, Melaka: FTMK, Universiti Teknikal Malaysia Melaka

*C++ tutorial*. (n.d.). www.javatpoint.com. https://www.javatpoint.com/cpp-tutorial

*MySQL tutorial*. (n.d.). www.javatpoint.com. https://www.javatpoint.com/mysql-tutorial

*84 food business ideas to consider for your next venture*. (2020, May 1). Small Business Trends. https://smallbiztrends.com/2019/12/food-business-ideas.html

Kim-Soon, Ng & Ahmad, Abd Rahman & Yunus, Nur & Hasaballah, Abdel Hafiez. (2018). *Snack Food Packaging Features and Consumer Repeat Purchase Intent*. Advanced Science Letters. 24. 3161-3165. 10.1166/asl.2018.11336.

Doyle, Barry & Bell, Art & Smith, Dayle. (2011). *Specialty Food and Beverage: A Case Study of Small Business Management*. Journal of Business Case Studies (JBCS).