



## **Programming With Python (PWP) Group Assignment**

**Module Code: 032021-SAM**

**Name of members:**

**Teo Kai Yii (Leader) - TP058618**

**Tang Kar Lok - TP060733**

**Intake code: UCDF2005(1) ICT(SE)**

**Hand In Date: 22 April 2021**

**Hand Out Date: 18 June 2021**

**Lecture: MR. USMAN HASHMI**

## Table of Contents

<b>1.0 Introduction and Assumptions .....</b>	3
<b>2.0 Design of the program .....</b>	5
<b>2.1 Pseudocode .....</b>	5
<b>2.2 Flowchart .....</b>	53
<b>4.0 Additional Features Source Code with Explanation.....</b>	82
<b>5.0 Screenshots of sample input and output with explanation .....</b>	88
<b>6.0 Conclusion.....</b>	122
<b>7.0 Workload Matrix.....</b>	123
<b>8.0 References .....</b>	124

## 1.0 Introduction and Assumptions

This Python program is developed for SUPER CAR RENTAL SERVICES (SCRS) where customer to be able to book car rental online. This system aims to save customer's time as customer no longer need to book car rental physically. The python program is called ONLINE CAR RENTAL SYSTEM (OCRS). Design for Malaysia's fast-expanding online car rental services to allow customers to book rental cars virtually and to provide a modern and convenient services by saving both customers and the staffs time.

OCRS has a proper validation during user input. If user do not enter value and do not follow the format given, user will not be able to proceed further. For example, user need to provide a proper contact number, date of birth and password to register an account. Repeated user name will be rejected to keep the uniqueness of the access. Age restrictions will be implemented when user registering the account. In Malaysia, users must have a minimum age of 23 years old and maximum 65 years old to rent a car (Auto Europe, n.d.).

OCRS has an appropriate security feature. User will not be able to login and access features of OCRS without a pre-registered name and password; no matter the user is customer or admin. This will prevent cases of not knowing which customer made a booking to rent a car online. In addition, this security forbid unknown users to modify information like price and quantity.

OCRS has an option for users to exit anytime or when user finish accessing a feature. Some examples are users will be prompted whether they want to continue to access the same features, access other features or log out OCRS. OCRS is very flexible as user can exit the system anytime. Undeniably, OCRS will prompt the users once again if they want to exit OCRS to ensure user not exiting mistakenly.

OCRS stores booking information accurately. OCRS ensures customer is registered before customer starts to fill in booking information. Customer needs to select a car that is available from the available car list. Moreover, after selection of car, the car details will be shown to the customer again. The maximum days of car rental allowed for the customer is set to 28 days (Partridge, 2018). Furthermore, customer allow to place booking 1 years ahead. Customer needs to provide the car rental date and time while filling up booking information. Rented car can be collected from 8:00 to 22:59 daily include weekend and public holiday.

The additional features of OCRS are snacks and drinks ordering while booking a car. The snacks and drinks will be provided in car when customer collect the car. This is convenient and customers will enjoy having snacks and drinks in the car.

Snacks and drinks will be in charge by the F&B department. While customers order snacks and drinks, information of car rental date and time can be retrieve by the F&B department to prepare the snacks and drinks on time.

After customer enters all booking information, online payments need to be made to confirm the booking. If customer fails to pay the rental fees, booking will be rejected.

The details of bookings, payments, snacks and drinks order will be stored accurately in text file of booking and snack. In the OCRS, admin has a total of 13 features, such as modify car details, add cars to be rented out, and search customer bookings. Car ID and Snack ID cannot be modified and repeat to maintain its unique identity. OCRS can sort results for the users. OCRS provide four different view mode to the user which user can view customer bookings, customer payments, available cars, rented cars and etc. with the mode of “All”, “Price (Low-High)”, “Price (High-Low)”, and “Latest”. Furthermore, OCRS has the features to search for customer booking and payment which lets admin to monitor business activities. Lastly, car returned by customer after the car rental end will be returned to the OCRS by admin.

## 2.0 Design of the program

### 2.1 Pseudocode

```

FUNCTION Ask_Car_ID (variable_name, instruction)
BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        Open car text file as file_car
        IF LENGTH OF variable_name == 0 THEN
            DISPLAY" Car ID is required!"
        IF LENGTH OF variable_name != 3 THEN
            DISPLAY"Car ID should be in 3 character"
        ELIF variable_name exist IN file_car THEN
            DISPLAY "Car ID unavailable! Please choose another Car ID"
            Close Car text file
        ELSE
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

FUNCTION Ask\_Car\_Name (variable\_name, instruction)

```

BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        IF LENGTH OF variable_name == 0 THEN
            DISPLAY"Name required!"
        ELSE
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

FUNCTION Ask\_Car\_Description (variable\_name, instruction)

```

BEGIN
    DOWHILE (True)
        DISPLAY instruction

```

```

READ variable_name
IF LENGTH OF variable_name == 0 THEN
    DISPLAY" Description required!"
ELSE
    BREAK
END IF
ENDDO
RETURN variable_name
END

```

```

FUNCTION Ask_Car_Seat(variable_name,instruction)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY instruction
            READ variable_name
            IF variable_name NOT IN RANGE (2,11) THEN
                DISPLAY" Please enter approriate number of seats"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT:
            DISPLAY" Only numbers allowed!"
            CONTINUE
        END EXCEPT
    ENDDO
    RETURN variable_name
END

```

```

FUNCTION Ask_Car_Price(variable_name,instruction)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY instruction
            READ variable_name
            IF LENGTH OF variable_name == 0 THEN
                DISPLAY" Daily Price required!"
            ELIF variable_name < 100 THEN
                DISPLAY" Minimum RM100"
            END IF
        END TRY
    ENDDO
    RETURN variable_name
END

```

```

ELSE:
    BREAK
END IF
END TRY
EXCEPT:
    DISPLAY" Only whole numbers allowed!"
    CONTINUE
END EXCEPT
ENDDO
RETURN variable_name
END

```

```

FUNCTION Ask_Snack_Name (variable_name, instruction)
BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        IF LENGTH OF variable_name == 0 THEN
            DISPLAY "Snack name required!"
        ELSE
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

```

FUNCTION Ask_Snack_Price(variable_name,instruction)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY instruction
            READ variable_name
            IF LENGTH OF variable_name NOT IN RANGE (1,10) THEN
                DISPLAY "Price in RM1 to RM 9"
            ELSE:
                BREAK
            END IF
        END TRY
        EXCEPT:
            DISPLAY" Only whole numbers allowed!"

```

```

        CONTINUE
    END EXCEPT
ENDDO
RETURN variable_name
END

FUNCTION View_Mode(option, item)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY "Which mode would you like to view "+item+"?"
            READ option
            "1. All
            2.Price (high- Low)
            3.Price (low to high)
            4. Latest
            Option:"
            IF option NOT IN RANGE (1,5) THEN
                DISPLAY" Only option 1 to option 4 is allowed!"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT:
            DISPLAY" Invalid! Only number is allowed"
            CONTINUE
        END EXCEPT
    ENDDO
    RETURN option
END

```

```

FUNCTION Ask_Quantity(variable_name,instruction)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY instruction
            READ variable_name
            IF LENGTH OF variable_name < 1THEN
                DISPLAY "Minimum One quantity"
            ELSE:

```

```

        BREAK
    ENDIF
END TRY
EXCEPT:
    DISPLAY" Only whole numbers allowed!"
    CONTINUE
END EXCEPT
ENDDO
RETURN variable_name
END

```

```

FUNCTION Update_File(file name, list)
BEGIN
    Open file_name and clear the text file to insert lines
    i = 0
    DO WHILE (i < LENGTH OF list)
        detail = connect list[i] with a tab
        Insert line (detail + new line) to file_name
        i += 1
    ENDDO
    RETURN
END

```

```

FUNCTION Ask_index (file name, list, num)
BEGIN
    DOWHILE (True)
        index_num = 0
        TRY
            DISPLAY "Enter field number to be modify: "
            READ index_num
            IF index_num NOT IN RANGE (1, limit) THEN
                DISPLAY "Only option 1 to option" + num+
                "available"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT
            DISPLAY "Invalid ! Only number is allowed! "
            CONTINUE

```

```

        END EXCEPT
    END DO
    RETURN index_num
END

FUNCTION Get_File_Details(file, list)
BEGIN
    Define list
    Open file as file_detail to read data in file:
    FOR each IN file_detail
        Strip each element from beginning and end
        Separate each element with tab
        Insert each element in the end of the list
    END FOR
    RETURN list
END

```

```

FUNCTION User_exist()
BEGIN
    DO WHILE (True)
        Define user_details list
        user_details = get_file_details("user.txt",user_details)
        DISPLAY "Name:"
        READ name
        FOR each_customer IN user_details
            customer_name = each_customer[0]
            IF name == customer_name THEN
                name_check = True
                BREAK
            ELSE:
                name_check = False
            END IF
        END FOR
        IF LENGTH OF name == 0 THEN
            DISPLAY "Please enter your name"
        ELIF name_check THEN
            DISPLAY "User exist! Please continue"
            BREAK
        ELSE
            DISPLAY "User not found ! Please re-enter again !"
        END IF
    END WHILE

```

```

ENDDO
RETURN Name
END

```

```

FUNCTION Ascending(list,num)
BEGIN
  FOR j IN RANGE (0, LENGTH OF list- 1)
    swapped = False
    FOR I IN RANGE (0, LENGTH OF list- 1)
      IF int(list[i][num]) > int(list[i+1][num]) THEN
        swap = list[i]
        list[i] = list[i+1]
        list[i+1] = swap
        swapped = True
      ENDIF
    END FOR
    IF NOT swapped THEN
      BREAK
    END IF
  END FOR
  RETURN list
END

```

```

FUNCTION Descending(list,num)
BEGIN
  FOR j IN RANGE (0, LENGTH OF list -1)
    swapped = False
    FOR I IN RANGE (0, LENGTH OF list -1)
      IF int(list[i][num]) < int(list[i+1][num]) THEN
        swap = list[i]
        list[i] = list[i+1]
        list[i+1] = swap
        swapped = True
      ENDIF
    END FOR
    IF NOT swapped THEN
      BREAK
    END IF
  END FOR
  RETURN list
END

```

END

```

FUNCTION Latest(list)
BEGIN
    I = 0
    length = LENGTH OF list
    FOR I IN RANGE (0, length//2)
        swap= list
        list[i] = list[length-i-1]
        list[length-i-1] = swap
    END FOR
    RETURN list
END

```

FUNCTION day\_month()

```

BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY "Day(DD):"
            READ day
            IF day NOT IN RANGE (1,32)THEN
                DISPLAY" Please enter an appropriate day"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT:
            DISPLAY" Please enter according to the format given"
            CONTINUE
        END EXCEPT
    ENDDO
    DOWHILE (True)
        TRY:
            DISPLAY "Month (MM):"
            READ month
            IF month NOT IN RANGE (1,13) THEN
                DISPLAY" Please enter an appropriate month"
            ELSE:
                BREAK
            END IF

```

```

END TRY
EXCEPT:
    DISPLAY" Please enter according to the format given"
    CONTINUE
END EXCEPT
ENDDO
RETURN day, month

END

```

```

FUNCTION Ask_Rental_date()
BEGIN
    isValidate = False
    DOWHILE isValidate = False
        Day, month = day_month()
        DOWHILE (True)
            TRY:
                DISPLAY "Year(YYYY):"
                READ year
                IF LENGTH OF str(year) != 4 THEN
                    DISPLAY" Please enter an appropriate year"
                ELIF year NOT IN RANGE (1956, 1999)THEN
                    DISPLAY" Only 23 to 65 yrs old is allowed!"
                ELSE
                    BREAK
                END IF
            END TRY
            EXCEPT:
                DISPLAY" Please enter according to the format given"
                CONTINUE
            END EXCEPT
        ENDDO
        DOWHILE (True)
            TRY:
                DISPLAY "Hour(HH):"
                READ hour
                IF hour NOT IN RANGE (8,23) THEN
                    DISPLAY" Please enter an appropriate time in 24-hr
format.Only 8:00 to 22:59 is allowed"
                ELSE:
                    BREAK
            END TRY
    END DOWHILE
END FUNCTION

```

```

        END IF
    END TRY
    EXCEPT:
        DISPLAY" Please enter according to the format given"
        CONTINUE
    END EXCEPT
ENDDO
DOWHILE (True)
TRY:
    DISPLAY "Minutes(MM):"
    READ minutes
    IF minutes NOT IN RANGE (1,60) THEN
        DISPLAY" Please enter an appropriate time"
    ELSE
        BREAK
    END IF
END TRY
EXCEPT:
    DISPLAY" Please enter according to the format given"
    CONTINUE
END EXCEPT
ENDDO
start_date= str(year)+"-"+str(month)+"-"+str(day)+"-"+str(hour)
+":"+str(minutes)
    rental_start_date = 0
DOWHILE (True)
TRY:
    rental_start_date = VALIDATE DATE AND TIME
    isValidate = True
    BREAK
END TRY
EXCEPT ValueError:
    isValidate = False
    CONTINUE
END EXCEPT
IF NOT isValidate THEN
    THEN DISPLAY" Invalid Date or Time ! Please re-enter"
ENDDO
ENDDO
RETURN rental_start_date
END

```

```

FUNCTION Payment(total)
BEGIN
    DISPLAY "The total amount to be paid is RM", total
    payment_date = STORE TODAY'S DATE
    DISPLAY "Please choose a payment method:"
    payment_method = ["Credit Card", "Debit Card", "Online Banking"]
    FOR i IN RANGE (0,3):
        i += 1
        str(i)
        DISPLAY (i,"->",payment_method[i-1])
        IF i == 3 THEN
            BREAK
        ENDIF
    END FOR
    DOWHILE True
        TRY
            DISPLAY "Payment Method:"
            READ payment_method
            IF LENGTH OF str(payment_method) == 0 THEN
                DISPLAY "Payment Method required!"
            ELIF payment_method NOT IN RANGE (1,4)THEN
                DISPLAY "Please choose an available option!"
            ELIF payment_method IN RANGE (1, 4) THEN
                BREAK
            END IF
        END TRY
        EXCEPT
            DISPLAY "Invalid! Only number is allowed!"
            CONTINUE
        END EXCEPT
    ENDDO
    IF payment_method == 2 THEN
        payment_method = "Credit Card"
    ELIF payment_method == 3 THEN
        payment_method = "Debit Card"
    ELIF payment_method == 4 THEN
        payment_method = "Online Banking"
    ENDIF
    payment_status = "Paid"
    RETURN payment_status, payment_date, payment_method
END

```

```

FUNCTION Ask_Name(variable_name,instruction)
BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        Open user text file as file_user
        IF LENGTH OF variable_name == 0 THEN
            DISPLAY "Please enter your name! Name is required!"
        ELIF variable_name IN file_user THEN
            DISPLAY "Name unavailable! Please choose another name"
            Close user text file
        ELSE:
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

```

FUNCTION Ask_Gender(variable_name,instruction)
BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        IF LENGTH OF variable_name == 0 THEN
            DISPLAY "Please enter your gender"
        ELIF variable_name != "M" AND variable_name!="F" THEN
            DISPLAY "Provide gender with "M" as Male "F" as Female."
        ELSE:
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

```

FUNCTION Ask_Phone(variable_name,instruction)
BEGIN
    DOWHILE (True)
        TRY:
            DISPLAY instruction

```

```

READ variable_name
IF LENGTH OF variable_name != 9 THEN
    DISPLAY "Please enter your phone number in 9 digit"
ELSE:
    BREAK
END IF
END TRY
EXCEPT:
    DISPLAY "Please enter your phone number"
    CONTINUE
END EXCEPT
ENDDO
RETURN variable_name
END

```

```

FUNCTION Ask_Email(variable_name,instruction)
BEGIN
    DOWHILE (True)
        DISPLAY instruction
        READ variable_name
        IF "@" NOT IN variable_name THEN
            DISPLAY "Please enter an appropriate email address"
        ELIF "." NOT IN variable_name THEN
            DISPLAY "Please enter an appropriate email address"
        ELIF "mail" NOT IN variable_name THEN
            DISPLAY "Please enter an appropriate email address"
        ELSE:
            BREAK
        END IF
    ENDDO
    RETURN variable_name
END

```

```

FUNCTION Ask_Password(instruction1, instruction2)
BEGIN
    DOWHILE (True)
        DISPLAY instruction1
        READ password
        IF LENGTH OF password <8 THEN
            DISPLAY "Password should be at least 8 character"

```

```

ELSE
    DISPLAY instruction2
    READ confirm_pwd
    IF confirm_pwd != password THEN
        DISPLAY "Password does not match! Please re-enter your
password:"

    ELSE:
        BREAK
    ENDIF
END IF
ENDDO
RETURN confirm_pwd
END

```

```

FUNCTION Ask_Birthday ()
BEGIN
isValidate = False
DOWHILE isValidate = False
    day, month = day_month()
    DOWHILE (True)
        TRY:
            DISPLAY "Year(YYYY):"
            READ year
            IF LENGTH OF str(year) != 4 THEN
                DISPLAY "Please enter an appropriate year"
            ELIF year NOT IN RANGE (1956, 1999)THEN
                DISPLAY "Only 23 to 65 yrs old is allowed!"
            ELSE:
                BREAK
            END IF
        END TRY
        EXCEPT:
            DISPLAY "Please enter according to the format given"
            CONTINUE
        END EXCEPT
    ENDDO
    dob = str(year)+"-"+str(month)+"-"+str(day)
    birthday = 0
    DOWHILE (True)
        TRY:
            birthday = IDENTIFY VALIDATE DATE

```

```

        isValidate = True
        BREAK
    END TRY
    EXCEPT ValueError:
        isValidate = False
        CONTINUE
    END EXCEPT
    IF NOT isValidate THEN
        DISPLAY " Invalid Date ! Please re-enter"
    ENDIF
ENDDO
ENDDO
RETURN birthday
END

FUNCTION Log_Out()
BEGIN
    DO WHILE (True)
        option = 0
        TRY:
            DISPLAY ("Do you wish to log out")
            Please choose an option
            1: yes
            2: no
            option: "
            IF option NOT IN RANGE (1,3)
                THEN DISPLAY "Only option 1 and option 2 is available!"
            ELSE
                BREAK
            ENDIF
        END TRY
        EXCEPT:
            DISPLAY " Only number is allowed!"
            CONTINUE
        END EXCEPT
    ENDDO
    IF Option == 1 THEN
        DISPLAY "Thank you for coming! Hope to see you soon ! "
    ELIF Option == 2 THEN
        DISPLAY " Approaching to the welcome page ... "
        welcome()
    END IF

```

```

    RETURN
END

FUNCTION Who_Register()
BEGIN
    DO WHILE (True)
        option = 0
        TRY:
            DISPLAY ("What is your role?
Please choose an option
1: Customer
2: Admin
3: Exit
option: "
READ option
IF option NOT IN RANGE (1,4)
    THEN DISPLAY "Only option 1 to option 3 is available!"
ELSE
    BREAK
END TRY
EXCEPT:
    DISPLAY" Only number is allowed!"
    CONTINUE
END EXCEPT
ENDDO
IF who_register == 3 THEN
    log_out()
ENDIF
RETURN option
END

FUNCTION Welcome()
BEGIN
    DISPLAY "WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) !"
    DO WHILE (True)
        option = 0
        TRY:
            option = DISPLAY "Do you have an account?
Please choose an option"
            1: Yes
            2: No
            3: Exit

```

```

        Option: “
READ option
IF option NOT IN RANGE (1,4) THEN
    DISPLAY “Only option 1 to option 3 is available!”
ELSE
    BREAK
END TRY
EXCEPT:
    DISPLAY” Only number is allowed!”
    CONTINUE
END EXCEPT
ENDDO
IF OPTION == 1 THEN
    DOWHILE(True)
        TRY:
            option = 0
            option = DISPLAY “Do you want to login now?
                Please choose an option
                1: Login
                2: Exit
                Option:”
            READ option
            IF option NOT IN RANGE (1,3) THEN
                DISPLAY “Only option 1 and option 2 is available!”
            ELSE
                BREAK
            ENDIF
        END TRY
        EXCEPT:
            DISPLAY” Only number is allowed!”
            CONTINUE
        END EXCEPT
    ENDDO
    IF option == 1 THEN
        user_type = who_register()
        IF user_type == 1 THEN
            customer_login()
        ELIF user_type == 2 THEN
            admin_login()
        ENDIF
    ELSE:
        log_out()
    ENDIF

```

```

        ENDIF
ELIF option ==2 THEN
    DOWHILE(True)
        TRY:
            option = 0
            option = DISPLAY "register an account now?"
                Please choose an option
                1: Login
                2: View all cars available for rent
                3: Exit
                Option"
            IF option NOT IN RANGE (1,4) THEN
                DISPLAY "Only option 1 to option 3 is available!"
            ELSE
                BREAK
            ENDIF
        END TRY
        EXCEPT:
            DISPLAY" Only number is allowed!"
            CONTINUE
        END EXCEPT
    ENDDO
    IF option == 1 THEN
        user_type = who_register()
        IF user_type == 1 THEN
            customer_register()
        ELIF user_type == 2 THEN
            admin_register()
        ENDIF
    ELIF option == 2 THEN
        guest_available_car()
    ELIF option == 3 THEN
        log_out()
    ENDIF
    ELSE
        log_out()
    RETURN
END

```

```

FUNCTION Menu_Admin()
BEGIN
    DISPLAY "What would you like to do?"

```

```

DISPLAY " 1. Modify Personal Details"
DISPLAY " 2. Modify Car Details"
DISPLAY " 3. Modify Snacks & Drinks Details"
DISPLAY " 4. Add Cars To Be Rented Out"
DISPLAY " 5. Add Snacks & Drinks To Be Sold"
DISPLAY " 6. View Cars Rented Out"
DISPLAY " 7. View Cars Available For Rent"
DISPLAY " 8. View Customer Bookings"
DISPLAY " 9. View Customer Payment"
DISPLAY "10. View Available Snacks & Drinks"
DISPLAY "11. Search Customer Booking"
DISPLAY "12. Search Customer Payment"
DISPLAY "13. Return Rented Car"
DISPLAY "14. Exit"
DOWHILE(True)
    TRY:
        DISPLAY "Please choose an option
        option: "
        IF option NOT IN RANGE (1,15) THEN
            DISPLAY "Only option 1 to option 14 is available!"
        ELSE
            BREAK
        ENDIF
    END TRY
    EXCEPT:
        DISPLAY " Only number is allowed!"
        CONTINUE
    END EXCEPT
    READ option
    ENDDO
    IF option == 1 THEN
        modify_admin_details()
    ELIF option == 2 THEN
        modify_car()
    ELIF option == 3 THEN
        modify_snacks()
    ELIF option == 4 THEN
        insert_car()
    ELIF option ==5 THEN
        insert_snacks()
    ELIF option ==6 THEN
        admin_rented_car()

```

```

ELIF option ==7 THEN
    admin_available_car()
ELIF option ==8 THEN
    view_booking()
ELIF option ==9 THEN
    view_payment()
ELIF option ==10 THEN
    admin_snacks()
ELIF option ==11 THEN
    admin_search_booking()
ELIF option ==12 THEN
    search_payment()
ELIF option ==13 THEN
    return_car()
ELIF option ==14 THEN
    log_out()
END IF
RETURN
END

```

```

FUNCTION Admin_continue()
BEGIN
    answer = 1
    DOWHILE True
        TRY
            DISPLAY "Do you want to continue? Please choose an option"
            1: Yes
            2: View Admin Features Menu
            3: Exit
            Option:""
            READ answer
            IF answer NOT IN RANGE (1,4) THEN
                DISPLAY "Only option 1 to option 3"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT
            DISPLAY "Invalid! Only numbers allowed!"
            CONTINUE
        END EXCEPT

```

```

END DO
IF answer == 2 THEN
    menu_admin()
ELIF answer ==3 THEN
    log_out ()
RETURN answer
END

FUNCTION Menu_Customer()
BEGIN
    DISPLAY "What would you like to do?"
    DISPLAY " 1. Modify Personal Details"
    DISPLAY " 2. View Personal Rental History"
    DISPLAY " 3. View Details of Available Car"
    DISPLAY " 4. Place a Booking"
    DISPLAY " 5. Exit"
    DOWHILE(True)
        TRY:
            DISPLAY "Please choose an option
            option: "
            IF option NOT IN RANGE (1,6)
                THEN DISPLAY "Only option 1 to option 5 is available!"
            ELSE
                BREAK
            ENDIF
        END TRY
        EXCEPT:
            DISPLAY" Only number is allowed!"
            CONTINUE
        END EXCEPT
    ENDDO
    IF option == 1 THEN
        modify_customer_details()
    ELIF option == 2 THEN
        cus_search_booking()
    ELIF option == 3 THEN
        cus_available_car()
    ELIF option == 4 THEN
        Place_booking()
    ELIF option ==5 THEN
        log_out()
    END IF

```

```

    RETURN
END

FUNCTION Customer_continue()
BEGIN
    answer = 1
    DOWHILE True
        TRY
            DISPLAY "Do you want to continue? Please choose an option"
            1: Yes
            2: View Features Menu
            3: Exit
            Option:""
            READ answer
            IF answer NOT IN RANGE (1,4) THEN
                DISPLAY "Only option 1 to option 3"
            ELSE
                BREAK
            END IF
        END TRY
        EXCEPT
            DISPLAY "Invalid! Only numbers allowed!"
            CONTINUE
        END EXCEPT
    END DO
    IF answer == 2 THEN
        menu customer ()
    ELIF answer ==3 THEN
        log_out()
    RETURN answer
END

FUNCTION Show_snacks()
BEGIN
    option = 0
    option = view_mode(option,"snack and drinks")
    snack_details = []
    snack_details = get_file_details("snack.txt", snack_details)
    snack_details = [each_snack FOR each_snack IN snack_details IF LENGTH OF
    each_snack[1] > 3]
    IF option = 1 THEN
        FOR j IN RANGE (0, LENGTH OF list-1)

```

```

swapped = False
FOR I IN RANGE (0, LENGTH OF list -1)
    IF int(list[i][1:3]) > int(list[i+1][1:3]) THEN
        swap = list[i]
        list[i] = list[i+1]
        list[i+1]= swap
        swapped = True
    ENDIF
END FOR
IF NOT swapped THEN
    BREAK
END IF
END FOR
ELIF option = 2 THEN
    snack_details = ascending(snack_details,2)
ELIF option = 3 THEN
    snack_details = descending(snack_details,2)
ELIF option = 4 THEN
    snack_details = latest(snack_details)
DISPLAY drinks list
DISPLAY snacks list
RETURN
END

```

```

FUNCTION Admin_snacks()
BEGIN
    answer = 1
    DOWHILE answer = 1
        show_snacks ()
        answer = admin_continue ()
    RETURN
END

```

```

FUNCTION Customer_snacks ()
BEGIN
    answer = 1
    DOWHILE answer = 1
        show_snacks ()
        answer = customer_continue ()
    RETURN

```

END

FUNCTION Order\_snacks ()

BEGIN

DISPLAY "Snacks and Drinks will be provided in the car when you come to collect the car"

answer = 1

snack\_total, snack\_cost = 0, 0

DOWHILE(True):

TRY

DISPLAY "Do you need any snacks and drinks? Please choose an option"

1. Yes

2. No

Option:"

READ option

IF option NOT IN RANGE (1,3) THEN

DISPLAY "Only option 1 and 2 is allowed"

ELSE

BREAK

END TRY

EXCEPT

DISPLAY "Invalid! Only number is allowed!"

CONTINUE

END EXCEPT

ENDDO

DOWHILE answer == 1:

Define booking\_name, order\_snack\_id, order\_quantity, rental\_start\_date, index

booking\_name = user\_exist()

show\_snacks()

snack\_details = []

snack\_details = get\_file\_details("snack.txt", snack\_details)

DISPLAY "Which snack or drink would you like to order?"

DOWHILE(True)

DISPLAY "Enter Snack ID to order "

READ order\_snack\_id

not\_found = True

FOR all\_snack IN snack\_details:

index += 1

IF order\_snack\_id == all\_snack[0] THEN

DISPLAY ("Snack ID: ", all\_snack[0])

DISPLAY ("1: Name:", all\_snack[1])

```

        DISPLAY ("2: Price:",all_snack[2])
        DISPLAY ("3: Quantity:",all_snack[3])
        not_found = False
        BREAK
    ENDIF
ENDFOR
index = index - 1
IF not_found THEN
    DISPLAY "Snack or Drink is not found !"
ELSE:
    BREAK
ENDIF
ENDDO
DISPLAY "How many would you like to order ? "
order_quantity = ask_quantity(order_quantity,"Quantity: ")
FOR i IN RANGE (index, index+ 1):
    snack_cost = 0
    snack_cost = int(all_snack[2])*int(order_quantity)
    snack_total += snack_cost
ENDFOR
DISPLAY "When is your car rental start date and time ? "
rental_start_date = ask_rental_date()
rental_start_date = str(rental_start_date)
rental_start_date = rental_start_date[:16]
Open snack text file as file snack to insert values
Insert values to the file snack
Close file next
updated_list = []
Open snack text file to READ all data
FOR each_snack in file_snack:
    Strip each element from beginning and end
    Separate each element with tab
    Insert each element in the end of updated list
    IF order_snack_id == all_snack[0]
        DOWHILE(True):
            all_snack[3] = int(all_snack[3])-order_quantity
            all_snack[3] = str(all_snack[3])
            BREAK
    ENDDO
    Insert the list of all_snacks to updated list
ENDIF
updated_list.append(all_snack)

```

```

ENDFOR
update_file("snack.txt",updated_list)
DISPLAY "Snacks ordered successfully!"
DOWHILE(True):
    TRY:
        DISPLAY "Do you want to continue? Please choose an option
        1: Yes
        2: No
        Option: "
        READ answer
        IF answer NOT IN RANGE (1,3) THEN
            DISPLAY "Only option 1 and option 2 is available !"
        ELSE:
            BREAK
        ENDIF
    ENDTRY
    EXCEPT:
        DISPLAY "Only number is allowed!"
        CONTINUE
    END EXCEPT
END DO
IF answer == 2THEN
    BREAK
ENDIF
ENDDO
RETURN snack_total
END

FUNCTION View_booking()
BEGIN
    answer = 1
    DOWHILE answer == 1
        option = 0
        option = view_mode(option, "bookings")
        booking_details = []
        booking_details = get_file_details("booking.txt",booking_details)
        snack_details = []
        snack_details = get_file_details("snack.txt",snack_details)
        snack_details = [each_snack FOR each_snack IN snack_details IF LENGTH
OF each_snack[0] > 3]
        IF option == 1 THEN
            DISPLAY "View mode: All "

```

```

ELIF option == 2 THEN
    DISPLAY "View mode: Sort by Total Amount (Low to High) "
    booking_details = ascending(booking_details,7)
    snack_details = ascending(snack_details,2)
ELIF option == 3 THEN
    DISPLAY "View mode: Sort by Total Amount (High to Low)"
    booking_details = descending(booking_details,7)
    snack_details = descending(snack_details,2)
ELIF option == 4 THEN
    DISPLAY" View mode: Show Latest Bookings"
    booking_details = latest(booking_details)
    snack_details = latest(snack_details)
ENDIF
num = 0
FOR each_booking IN booking_details:
    DISPLAY booking_details according to index
    num += 1
    IF num == LENGTH OF booking_details THEN
        BREAK
END FOR
num = 0
FOR each_snack IN snack_details:
    DISPLAY snack details according to index
    num += 1
    IF num == LENGTH OF snack_details THEN
        BREAK
END FOR
answer = admin_continue()
ENDDO
RETURN
END

FUNCTION View_payment()
BEGIN
    answer = 1
    DOWHILE answer == 1
        option = 0
        option = view_mode(option, "payments")
        booking_details = []
        booking_details = get_file_details("booking.txt",booking_details)
        IF option == 1 THEN
            DISPLAY "View mode: All "

```

```

ELIF option == 2 THEN
    DISPLAY "View mode: Sort by Total Amount (Low to High)"
    booking_details = ascending(booking_details,7)
ELIF option == 3 THEN
    DISPLAY "View mode: Sort by Total Amount (High to Low)"
    booking_details = descending(booking_details,7)
ELIF option == 4 THEN
    DISPLAY" View mode: Show Latest Bookings"
    booking_details = latest(booking_details)
ENDIF
num = 0
FOR each_booking IN booking_details:
    DISPLAY booking_details according to index
    num += 1
    IF num == LENGTH OF booking_details THEN
        BREAK
    ENDIF
END FOR
answer = admin_continue()
ENDDO
RETURN
END

```

```

FUNCTION Available_car()
BEGIN
    Define option
    option = view_mode(option,"available cars for rent")
    Define car_details
    car_details = get_file_details("car.txt",car_details)
    car_details = [each_car FOR each_car IN car_details IF int(each_car[5]) > 0]
    IF option == 1THEN
        PASS
    ELIF option == 2 THEN
        car_details = ascending(car_details,4)
    ELIF option == 3 THEN
        car_details = descending(car_details,4)
    ELIF option == 4THEN
        car_details = latest(car_details)
    ENDIF
    FOR each car IN car_details
        DISPLAY (each_car[0]+\t+" "+each_car[3]+\t+each_car[4]+\t+
+each_car[1]+\t+each_car[2])

```

```

    num += 1
    IF num == LENGTH OF car_details THEN
        BREAK
    ENDIF
    END FOR
    RETURN
END

```

```

FUNCTION Customer_Available_Car()
BEGIN
    answer = 1
    DOWHILE answer = 1
        available_car ()
        answer = customer_continue ()
    RETURN
END

```

```

FUNCTION Admin_available_car()
BEGIN
    answer = 1
    DOWHILE answer = 1
        available_car ()
        answer = admin_continue()
    RETURN
END

```

```

FUNCTION Rented_car()
BEGIN
    Define option
    option = view_mode(option,"available cars for rent")
    Define car_details
    car_details = get_file_details("car.txt",car_details)
    car_details = [each_car FOR each_car IN car_details IF int(each_car[6]) > 0]
    IF option == 1 THEN
        PASS
    ELIF option == 2 THEN
        car_details = ascending(car_details,4)
    ELIF option == 3 THEN
        car_details = descending(car_details,4)
    ELIF option == 4 THEN
        car_details = latest(car_details)
    ENDIF

```

```

FOR each car IN car_details
    DISPLAY (each_car[0]+\t+" "+each_car[3]+\t+each_car[4]
    +"\t"+each_car[1]+\t+each_car[2])
    num += 1
    IF num == LENGTH OF car_details THEN
        BREAK
    ENDIF
END FOR
RETURN
END

```

```

FUNCTION Admin_rented_car()
BEGIN
    answer = 1
    DOWHILE answer = 1
        rented_car()
        answer = admin_continue()
    RETURN
END

```

```

FUNCTION Return_car()
BEGIN
    answer = 1
    DOWHILE answer = 1
        rented_car()
        Define update list, car details list
        car_details = get_file_details("car.txt",car_details)
        DISPLAY "Enter Car ID to return a rented car:"
        READ search_car
        not_found = True
        FOR all_car IN car_details
            IF search_car == all_car[0] THEN
                DISPLAY "Car ID: ", all_car[0]
                DISPLAY "1: Name:",all_car[1]
                DISPLAY "2: Description:",all_car[2]
                DISPLAY "3: Number Of Seats:",all_car[3],"seater"
                DISPLAY "4: Daily Price: RM",all_car[4]
                DISPLAY "5: Quantity:",all_car[5]
                all_car[5] = int(all_car[5])+1
                all_car[6] = int(all_car[6])-1
                all_car[5] = str(all_car[5])
                all_car[6] = str(all_car[6])

```

```

        not_found = False
    ENDIF
    Insert all car list into the updated list
ENDFOR
IF not_found THEN
    DISPLAY "Car is not found !"
    CONTINUE
ELSE
    update_file("car.txt",updated_list)
DISPLAY "Rented Car Return successfully!"
answer = admin_continue()
ENDDO
RETURN
END

FUNCTION modify_customer_details()
BEGIN
    answer = 1
    DOWHILE answer = 1
        modified_list = []
        Open user text file as file user to read
        DISPLAY" Enter your name to modify personal details:"
        READ search name
        not_found = True
        FOR each_customer IN file_user
            Split each customer with tab and strip from the beginning and
end
            Store each customer in all_customer
            IF all_customer[2] != "Admin" AND search_name =
all_customer[0] THEN
                DISPLAY "Name:",all_customer[0]
                DISPLAY "1: Gender:",all_customer[1]
                DISPLAY "2: Date Of Birthday:",all_customer[2]
                DISPLAY "3: Phone number: (+60)",all_customer[3]
                DISPLAY "4: Email Address:",all_customer[4]
                DISPLAY "5: Password:",all_customer[5]
                not_found = False
                DOWHILE (True)
                    index_num = 0
                    index_num = ask_index(index_num,6,5)
                    BREAK
                END DO

```

```

DOWHILE(True)
    DISPLAY ("Existing information in field
is: ", all_customer[index_num])
    IF index_num == 1 THEN
        all_customer[index_num] =
ask_gender(all_customer[index_num], "Enter new gender:")
        BREAK
    ELIF index_num == 2 THEN
        all_customer[index_num] =
ask_birthday()
        all_customer[index_num] =
str(all_customer[index_num])
        all_customer[index_num] =
all_customer[index_num][:10]
        BREAK
    ELIF index_num == 3 THEN
        all_customer[index_num] =
ask_phone(all_customer[index_num], "Enter new phone number (+60):")
        all_customer[index_num] =
str(all_customer[index_num])
        BREAK
    ELIF index_num == 4 THEN
        all_customer[index_num] = ask_email
(all_customer[index_num], "Enter new email address:")
        BREAK
    ELIF index_num == 5 THEN
        all_customer[index_num] =
ask_password("Enter new password:", "Confirm password: ")
        all_customer[index_num] =
all_customer[index_num]
        BREAK
    ENDIF
END DO
Insert all customer details to the modified list
END FOR
IF not_found THEN
    DISPLAY "Name is not found!"
    CONTINUE
ELSE
    update_file("user.txt",modified_list)
END IF
DISPLAY "Personal details successfully updated!"

```

```

        answer = customer_continue()
END DO
RETURN
END

FUNCTION modify_admin_details()
BEGIN
    answer = 1
    DOWHILE answer = 1
        modified_list = []
        Open user text file as file_user to read data
            DISPLAY" Enter your name to modify personal details:"
            READ search_name
            not_found = True
            FOR each_admin IN file_user
                Split each admin with tab and strip from the beginning and end
                Store each admin in all admin
                IF all_admin[2] == "Admin" AND search_name ==
all_admin[0] THEN
                DISPLAY "Name:",all_admin[0]
                DISPLAY "1: Gender:",all_admin[1]
                DISPLAY "2: Phone number: (+60)",all_admin[3]
                DISPLAY "3: Email Address:",all_admin[4]
                DISPLAY "4: Password:",all_admin[5]
                not_found = False
                DOWHILE (True)
                    TRY
                        DISPLAY "Enter field number to be
modify: "
                        READ index_num
                        IF index_num NOT in RANGE (1,5)
THEN
                        DISPLAY "Only option 1 to
option 4 available"
                        ELSE
                            IF index_num in RANGE (2,5)
THEN
                                index_num += 1
                                BREAK
                            END IF
                        END IF
                    END TRY

```

```

        EXCEPT
            DISPLAY "Invalid ! Only number is
allowed"
        CONTINUE
    END EXCEPT
END DO
DOWHILE(True)
    DISPLAY ("Existing information in field is: ",
all_admin[index_num])
    IF index_num == 1 THEN
        all_admin[index_num] =
ask_gender(all_admin[index_num], "Enter new gender:")
        BREAK
    ELIF index_num == 3 THEN
        all_admin[index_num] =
ask_phone(all_admin[index_num], "Enter new phone number (+60):")
        all_admin[index_num] =
str(all_admin[index_num])
        BREAK
    ELIF index_num == 4 THEN
        all_admin[index_num] = ask_email
(all_admin[index_num], "Enter new email address:")
        BREAK
    ELIF index_num == 5 THEN
        all_admin[index_num] =
ask_password("Enter new password:", "Confirm password: ")
        all_admin[index_num] =
all_admin[index_num]
        BREAK
    ENDIF
END DO
Insert all customer details to the modified list
END FOR
IF not_found THEN
    DISPLAY "Name is not found!"
    CONTINUE
ELSE
    update_file("user.txt",modified_list)
END IF
DISPLAY "Personal details successfully updated! "
answer = admin_continue()
END DO

```

```

    RETURN
END

FUNCTION Login()
BEGIN
    Define user_details
    user_details = get_file_details("user.txt",user_details)
    user_exist()
    RETURN user_details
END

```

```

FUNCTION Customer_login()
BEGIN
    user_details = login()
    DO WHILE (True)
        DISPLAY " Password:""
        READ login_password
        FOR each_customer IN user_details
            customer_password = each_customer[5]
            IF login_password == customer_password THEN
                password_check = True
                BREAK
            ELSE
                password_check = False
            ENDIF
        END FOR
        IF LENGTH OF login_password == 0 THEN
            DISPLAY "please enter your password"
        ELIF password_check THEN
            DISPLAY "Succesfully Logged in! Welcome back!"
            DISPLAY "Approaching Features Menu"
            menu_customer()
            BREAK
        ELSE:
            DISPLAY "Password wrong!"
        ENDIF
    ENDDO
    RETURN
END

```

```

FUNCTION Admin_login ()
BEGIN

```

```

user_details = login()
DO WHILE (True)
    Login_password = DISPLAY "Password:"
    FOR each_admin IN user_details
        admin_password = each_admin[5]
        IF login_password == admin_password THEN
            password_check = True
            BREAK
        ELSE
            password_check = False
        ENDIF
    END FOR
    IF LENGTH OF login_password == 0 THEN
        DISPLAY "please enter your password"
    ELIF password_check THEN
        DISPLAY "Successfully Logged in! Welcome back!"
        DISPLAY "Approaching Admin Features Menu"
        menu_admin()
        BREAK
    ELSE:
        DISPLAY "Password wrong!"
    ENDIF
ENDDO
RETURN
END

FUNCTION Admin_register()
BEGIN
    admin = "Admin"
    DO WHILE (True)
        Define name, gender, phone, email, comfirm_pwd
        Open file_user to insert line
        Close file_user
        name = ask_name(name, "Name: ")
        gender = ask_gender(gender, "Gender(M/F): ")
        phone = ask_phone(phone, "Phone number: (+60)")
        email = ask_email(email, "Email Address: ")
        confirm_pwd = ask_password("Password: ", "Confirm Password: ")
        Open user text file as file_user to insert line
        Insert values to file_user
        Close file_user
        DISPLAY " Registration successfully completed!"
    ENDIF
END

```

```

DISPLAY "Approaching login session ..."
admin_login()
BREAK
ENDDO
RETURN
END

FUNCTION Customer_register()
BEGIN
DO WHILE (True)
    Define name, gender, birthday, phone, email, comfirm_pwd
    Open user text file as file_user to insert line
    Close file_user
    name = ask_name(name,"Name: ")
    gender = ask_gender(gender,"Gender(M/F): ")
    DISPLAY ("Date Of Birth: ")
    birthday = ask_birthday()
    birthday = str(birthday)
    birthday = birthday [:10]
    phone = ask_phone(phone,"Phone number: (+60)")
    email = ask_email(email,"Email Address:")
    confirm_pwd = ask_password("Password: ","Confirm Password: ")
    Open user text file as file_user to insert line
    Insert values to file_user
    Close file_user
    DISPLAY " Registration successfully completed!"
    DISPLAY "Approaching login session ..."
    customer_login()
    BREAK
ENDDO
RETURN
END

FUNCTION Guest_Available_Car()
BEGIN
answer = 1
DO WHILE answer == 1
    available_car()
    DOWHILE(True)
        answer = 0
        TRY:
            DISPLAY "Do you want to register an account to place

```

booking?

```

        Please choose an option
        1: Continue to view all cars available for rent
        2: Register now
        3: Exit
        option: “
        IF option NOT IN RANGE (1,4) THEN
            DISPLAY “Only option 1 to option 3 is available!”
        ELSE
            BREAK
        ENDIF
    END TRY
    EXCEPT:
        DISPLAY” Only number is allowed!”
        CONTINUE
    END EXCEPT
    IF answer == 2 THEN
        customer_register()
    ELIF answer == 3 THEN
        log_out()
    ENDIF
ENDDO
RETURN
END

```

```

FUNCTION place_booking()
BEGIN
    answer = 1
    DOWHILE answer = 1
        Define booking_name, booking_car, booking_day, rental_start_date, index,
        car_details
        booking_name = user_exist()
        available_car()
        car_details = get_file_details("car.txt",car_details)
        DOWHILE True
            DISPLAY” Enter Car ID to book:”
            READ booking car
            not_found = True
            FOR all_car IN file_car
                IF search_car == all_car[0] THEN
                    DISPLAY "Car ID: ", all_car[0]

```

```

        DISPLAY "1: Name:",all_car[1]
        DISPLAY "2: Description:",all_car[2]
        DISPLAY "3: Number Of Seats:",all_car[3],"seater"
        DISPLAY "4: Daily Price: RM",all_car[4]
        DISPLAY "5: Quantity:",all_car[5]
        not_found = False
        BREAK
    ENDIF
END FOR
IF not_found THEN
    DISPLAY "Car is not found !"
ELSE
    BREAK
ENDIF
DISPLAY "How many days would you like to book ? "
DOWHILE True
TRY
    DISPLAY" Day:"
    READ booking day
    IF LENGTH OF str(booking_day) == 0 THEN
        DISPLAY "Please enter how many days would you like
to book"
    ELIF booking_day NOT IN RANGE (1, 29) THEN
        DISPLAY "Maximum 29 days are allowed!"
    ELSE
        BREAK
    ENDIF
END TRY
EXCEPT
    DISPLAY "only numbers allowed !"
    CONTINUE
END EXCEPT
ENDDO
FOR i IN RANGE (index, index+ 1):
    car_total = int(all_car[4])*int(booking_day)
ENDFOR
DISPLAY "When do you want to start your car rental ? Please provide date
and time"
rental_start_date = ask_rental_date()
rental_start_date = str(rental_start_date)
rental_start_date = rental_start_date[:16]
snack_total = order_snack()

```

```

payment_amount = car_total + snack_total
payment_status , payment_date, payment_method = payment(car_total)
Open booking text file as file_booking to insert values
Insert values to file_booking
Close file_booking
updated_list = []
FOR all_car in car_details:
    IF booking_car == all_car[0]:
        DOWHILE (True):
            all_car[5] = int(all_car[5])-1
            all_car[6] = int(all_car[6])+1
            all_car[5] = str(all_car[5])
            all_car[6] = str(all_car[6])
            BREAK
        ENDDO
    ENDIF
    Insert all car to the updated list
ENDFOR
update_file("car.txt",updated_list)
DISPLAY "Booking placed successfully!"
answer = customer_continue()
END DO
RETURN
END

```

```

FUNCTION Search_personal_history (name)
BEGIN
    search_name = user_exist()
    booking_details = []
    booking_details = get_file_details("booking.txt",booking_details)
    booking_details = [each_booking FOR each_booking IN booking_details IF
each_booking[0] == search_name]
    view_car = []
    FOR each_booking IN booking_details:
        Insert each_booking[1] into view car
ENDFOR
num = 0
FOR each_booking IN booking_details:
    IF each_booking[0] == search_name THEN
        DISPLAY booking details
        num += 1
    ELIF num == LENGTH OF booking_details THEN

```

```

        BREAK
    ENDIF
ENDFOR
car_details = []
car_details = get_file_details("car.txt",car_details)
num , i= 0 , 0
FOR each_car IN car_details:
    FOR i IN RANGE (LENGTH OF view_car)
        IF view_car[i] == each_car[0] THEN
            DISPLAY car details
            i += 1
        END IF
    END FOR
    num += 1
    IF num == LENGTH OF car_details THEN
        BREAK
    END IF
END FOR
snack_details = []
snack_details = get_file_details("snack.txt",snack_details)
snack_details = [each_snack FOR each_snack IN snack_details IF each_snack[0] ==
search_name]
view_snack = []
view_quantity = []
FOR each_snack IN snack_details:
    Insert each_booking[1] into view_car
    Insert each_booking[2] into view_quantity
END FOR
snack_details = []
snack_details = get_file_details("snack.txt",snack_details)
num, i = 0, 0
FOR each_snack IN snack_details:
    FOR i IN RANGE (LENGTH OF view_snack):
        IF view_snack[i] == each_snack[0] THEN
            DISPLAY snack details
            i += 1
        ENDIF
    ENDFOR
    num += 1
    IF num == LENGTH OF snack_details THEN
        BREAK
    END IF

```

```

ENDFOR
answer = customer_continue()
RETURN
END

FUNCTION Cus_search_booking():
    Answer = 1
    DOWHILE answer = 1:
        search_personal_history("PERSONAL RENTAL HISTORY SESSION
STARTED")
        answer = customer_continue()
    RETURN

FUNCTION Admin_search_booking():
    Answer = 1
    DOWHILE answer = 1:
        search_personal_history("SEARCH BOOKINGS SESSION STARTED")
        answer = admin_continue()
    RETURN

FUNCTION Insert_Car()
BEGIN
    answer = 1
    DOWHILE answer = 1
        Define car_id, name, description, seat, daily_price, quantity
        rented = 0
        Open car text file as file_car to insert value
        Close file_car
        car_id = ask_car_id(car_id,"Car ID: ")
        name = ask_car_name(name, "Name: ")
        description = ask_car_description(description,"Description: ")
        seat = ask_car_seat(seat, "Number Of Seats:")
        daily_price = ask_car_price(daily_price, "Daily Price: RM")
        quantity = ask_quantity(quantity, "Quantity: ")
        Open car text file as file_car to insert value
        Insert values to file_car
        Close file_car
        DISPLAY "Car information successfully updated!"
        answer = admin_continue()
    RETURN
END

FUNCTION modify_car()

```

```

BEGIN
    answer = 1
    DOWHILE answer = 1
        modified_list = []
        Open car file as file_car to read
            DISPLAY" Enter Car ID to modify car details:"
            READ search car
            not_found = True
            FOR each_car IN file_car
                Split each car with tab and strip from the beginning and end
                Store each car in all_car
                IF search_car == all_car[0] THEN
                    DISPLAY "Car ID: ", all_car[0]
                    DISPLAY "1: Name:",all_car[1]
                    DISPLAY "2: Description:",all_car[2]
                    DISPLAY "3: Number Of Seats:",all_car[3],"seater"
                    DISPLAY "4: Daily Price: RM",all_car[4]
                    DISPLAY "5: Quantity:",all_car[5]
                    not_found = False
                    DOWHILE (True)
                        index_num = 0
                        index_num = ask_index(index_num,6,5)
                        BREAK
                END DO
                DOWHILE(True)
                    DISPLAY ("Existing information in field is:
",all_car[index_num])
                    IF index_num == 1 THEN
                        all_car[index_num] =
                        ask_car_name(all_car[index_num], "Enter new name: ")
                        BREAK
                    ELIF index_num == 2 THEN
                        all_car[index_num] =
                        ask_car_description(all_car[index_num],"Enter new description: ")
                        BREAK
                    ELIF index_num == 3 THEN
                        all_car[index_num] =
                        ask_car_seat(int(all_car[index_num]),"Enter new number of seats: ")
                        all_car[index_num] =
                        str(all_car[index_num])
                        BREAK
                    ELIF index_num == 4 THEN

```

```

        all_car[index_num] =
ask_car_price(all_car[index_num], "Enter new daily price: RM")
        all_car[index_num] =
str(all_car[index_num])

        BREAK

        ELIF index_num == 5 THEN
            all_car[index_num] =
ask_quantity(all_car[index_num], "Enter new quantity: ")
            all_car[index_num] =
str(all_car[index_num])

        BREAK

        ENDIF

    END DO

    Insert all car details to the modified_list

    END FOR

    IF not_found THEN
        DISPLAY "Car is not found!"
        CONTINUE

    ELSE
        update_file("car.txt",modified_list)
    END IF

    DISPLAY "Car details successfully updated! "
    answer = admin_continue()

    END DO

    RETURN

END

FUNCTION Search_payment()
BEGIN
    answer = 1
    DOWHILE answer = 1
        DOWHILE (True)
            option = 0
            TRY
                DISPLAY "How would you like to search? Please choose an
option
                    1: Search by Name
                    2: Search by Date
                    Option: "
                READ option
                IF option NOT IN RANGE (1,3) THEN
                    DISPLAY "Only option 1 and option 2 is available!"

```

```

        ELSE:
            BREAK
        ENDIF
    ENDTRY
    EXCEPT:
        DISPLAY "Invalid! Only number is allowed!"
        CONTINUE
    END EXCEPT
ENDDO
user_details = []
user_details = get_file_details("user.txt",user_details)
booking_details = []
booking_details = get_file_details("booking.txt",booking_details)
IF option = 1 THEN
    search_name = user_exist()
    booking_details = [each_booking FOR each_booking IN
booking_details IF each_booking[0] == search_name]
ELIF option == 2 THEN
    DOWHILE True
        Search_date = 0
        DISPLAY "Enter a date to search(YYYY-MM-DD):"
        READ search_date
        FOR each_booking IN booking_details:
            Payment_date = each_booking [5]
            IF search_date == payment_date THEN
                date_check = True
                BREAK
            ELSE
                date check = False
            ENDIF
        ENDFOR
        IF LENGTH OF search_date = 0 THEN
            DISPLAY "please enter a date"
        ELIF date_check THEN
            DISPLAY "Date exist!"
        ELSE
            DISPLAY "Date not found!"
        ENDIF
    ENDDO
    booking_details = [each_booking FOR each_booking IN
booking_details IF each_booking[0] == search_name]
    FOR each_booking IN booking_details:

```

```

        DISPLAY booking details
        num += 1
        IF num = LENGTH OF booking_details THEN
            BREAK
        ENDIF
    END FOR
    answer = admin_continue()
    RETURN
END

FUNCTION Insert_Snacks()
BEGIN
    answer = 1
    DOWHILE answer = 1
        Define snack_id, snack_price, quantity, snack_details
        Open snack text file as file_snack to insert value
        Close file_snack
        DOWHILE True
            snack_details = get_file_details("snack.txt",snack_details)
        DISPLAY "Snack ID: "
            READ snack_id
            id_check = False
            FOR each_snack in snack_details:
                id_snack = each_snack[0]
                IF snack_id == id_snack THEN
                    id_check = True
                    BREAK
                ELSE
                    Id_check = False
                ENDIF
            END FOR
            IF LENGTH OF snack_id = 0 THEN
                DISPLAY "Snack ID required!"
            ELIF LENGTH OF snack_id != 3THEN
                DISPLAY" Snack ID should be in 3 character"
            ELIF NOT id_check THEN
                BREAK
            ELIF id_check THEN
                DISPLAY "Snack ID already exist"
            ENDIF
        ENDDO
        snack = ask_snack_name(snack, "Snack: ")
    END

```

```

        price = ask_snack_price(price,"Price: RM")
        quantity = ask_quantity(quantity,"Quantity: " )
        Open snack text file as file_snack to insert value
        Insert values to file_snack
        Close file_snack
        DISPLAY "Information successfully updated!"
        answer = admin_continue()

    ENDDO
    RETURN
END

FUNCTION modify_snacks()
BEGIN
    answer = 1
    DOWHILE answer = 1
        modified_list = []
        Open snack file as file_snack to READ
            DISPLAY" Enter Snack ID to modify snack details."
            READ search_snack
            not_found = True
            FOR each_snack IN file_snack
                Split each snack with tab and strip from the beginning and end
                Store each snack in all_snack
                IF search_snack == all_snack[0] THEN
                    DISPLAY "Snack ID: ", all_snack[0]
                    DISPLAY "1: Name:",all_snack[1]
                    DISPLAY "2: Price:",all_snack[2]
                    DISPLAY "3: Quantity:",all_snack[3]
                    not_found = False
                    DOWHILE (True)
                        index_num = 0
                        index_num = ask_index(index_num,4,3)
                        BREAK
                END DO
                DOWHILE(True)
                    DISPLAY ("Existing information in field is:
",all_snack[index_num])
                    IF index_num == 1 THEN
                        all_snack[index_num] =
                        ask_snack_name(all_snack[index_num], "Enter new name: ")
                        BREAK
                    ELIF index_num == 2 THEN

```

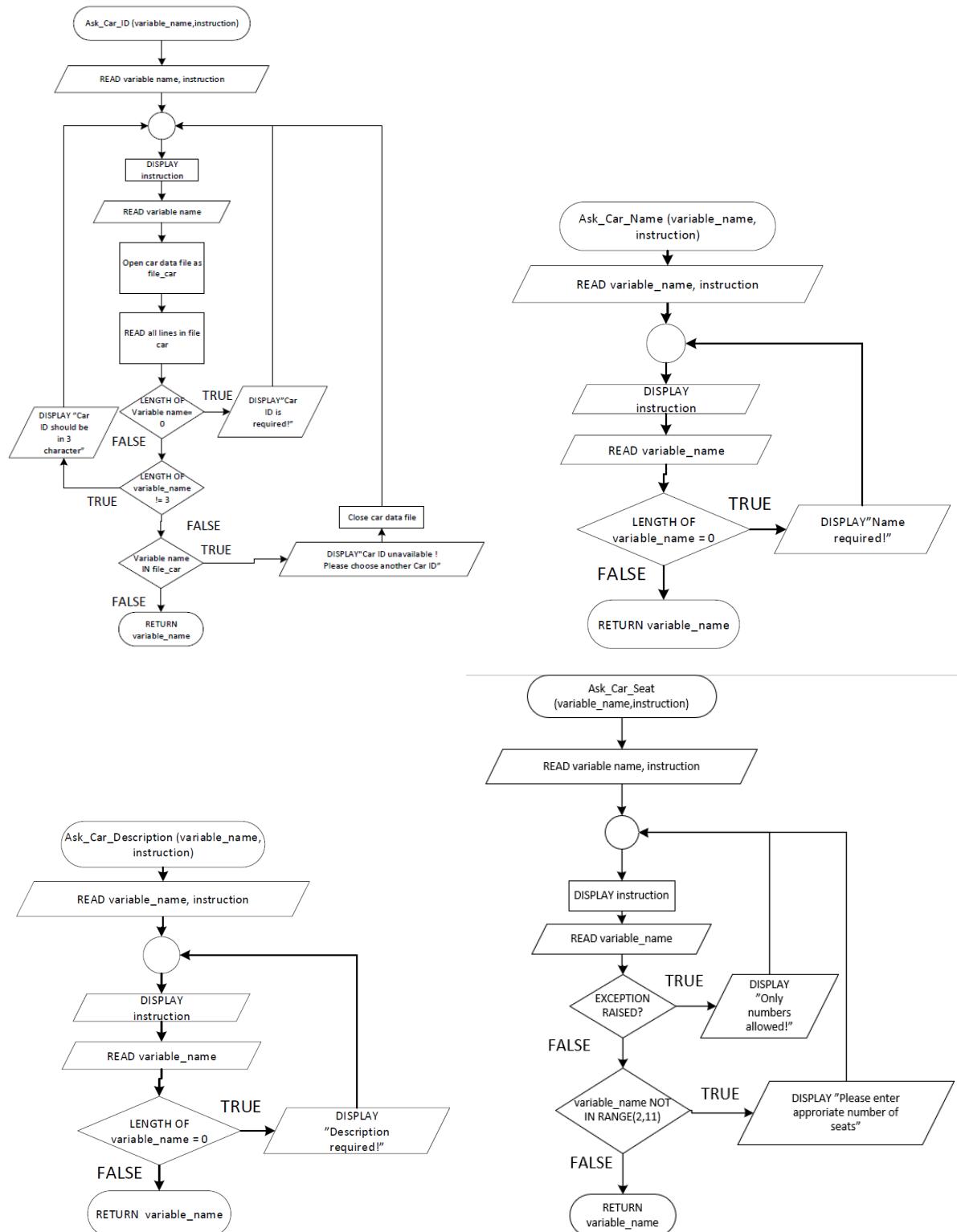
```

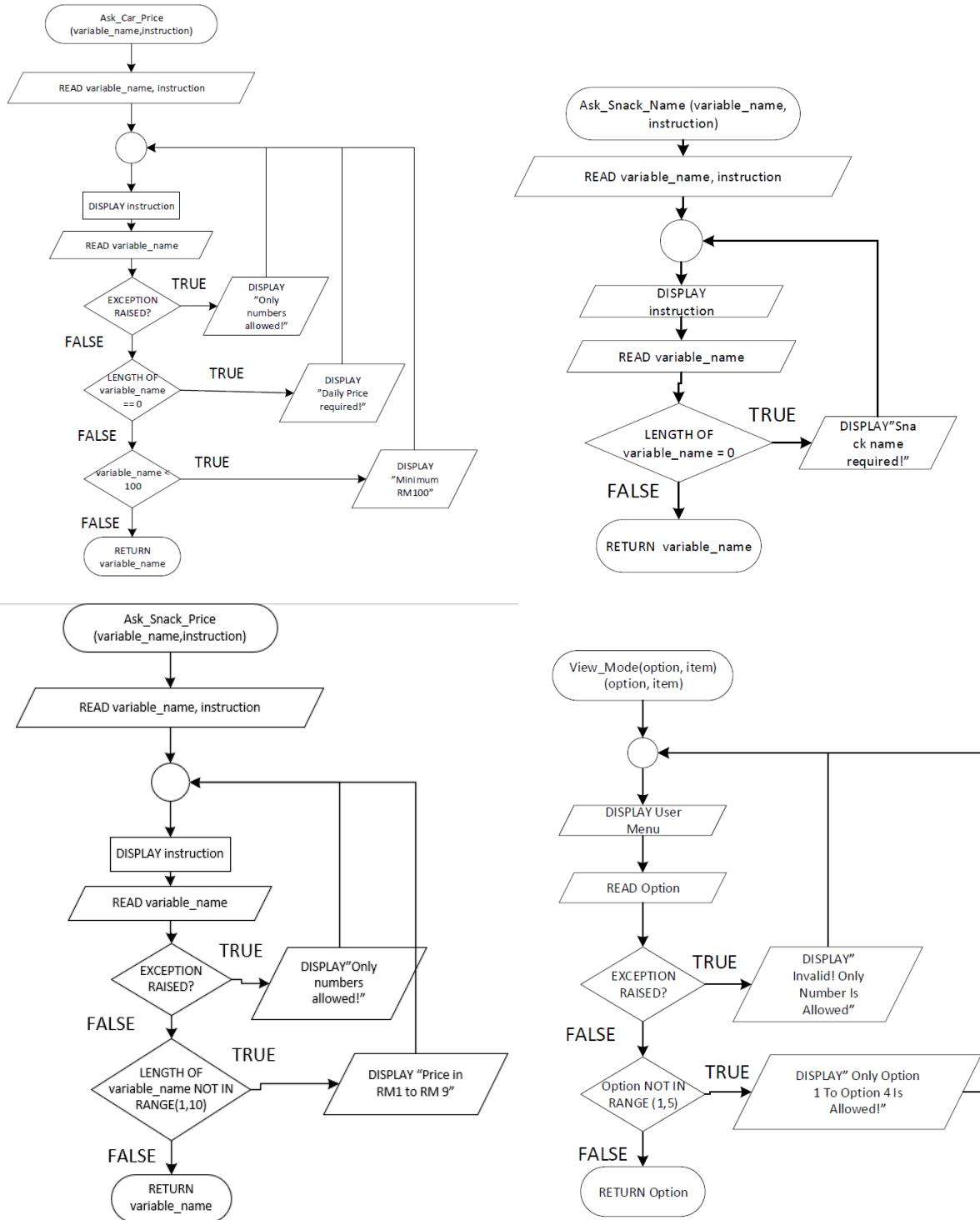
        all_snack[index_num] =
ask_snack_price(all_snack[index_num],"Enter new price: ")
        all_snack[index_num] =
str(all_snack[index_num])
        BREAK
    ELIF index_num == 3 THEN
        all_snack[index_num] =
ask_quantity(all_snack[index_num],"Enter new quantity:")
        all_snack[index_num] =
str(all_snack[index_num])
        BREAK
    ENDIF
END DO
Insert all snack details to the modified_list
END FOR
IF not_found THEN
    DISPLAY "Snack or Drink is not found!"
    CONTINUE
ELSE
    update_file("snack.txt",modified_list)
    END IF
    DISPLAY "Snacks & Drinks details successfully updated!"
    answer = admin_continue()
END DO
RETURN
END

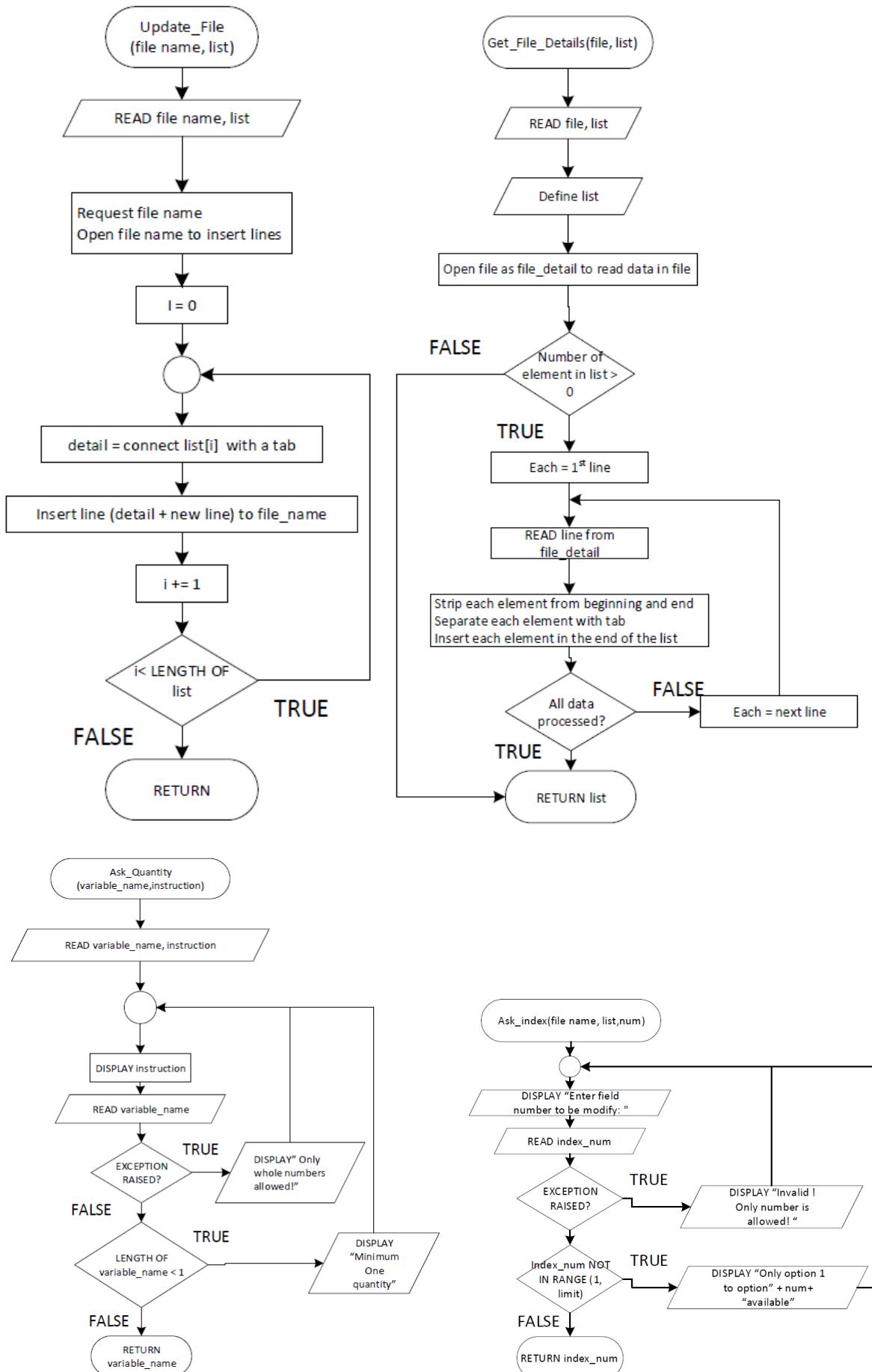
CALL Welcome() FUNCTION

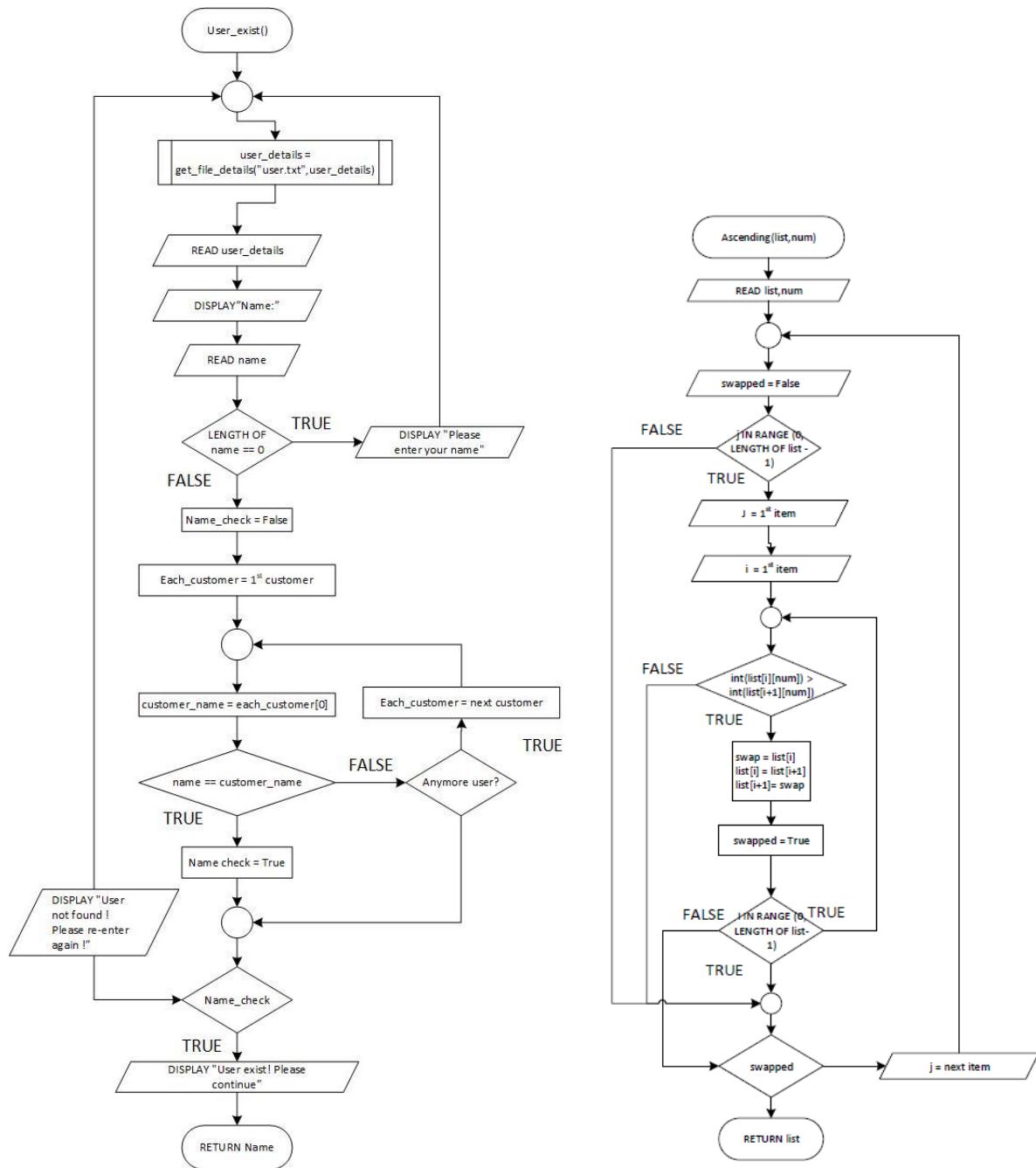
```

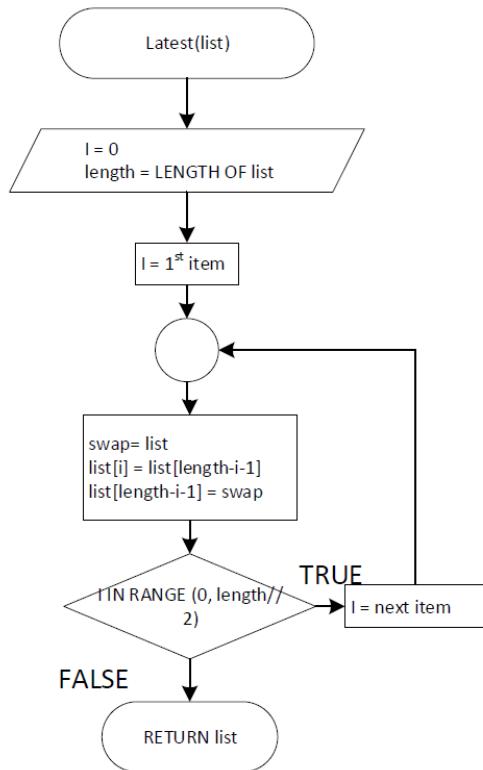
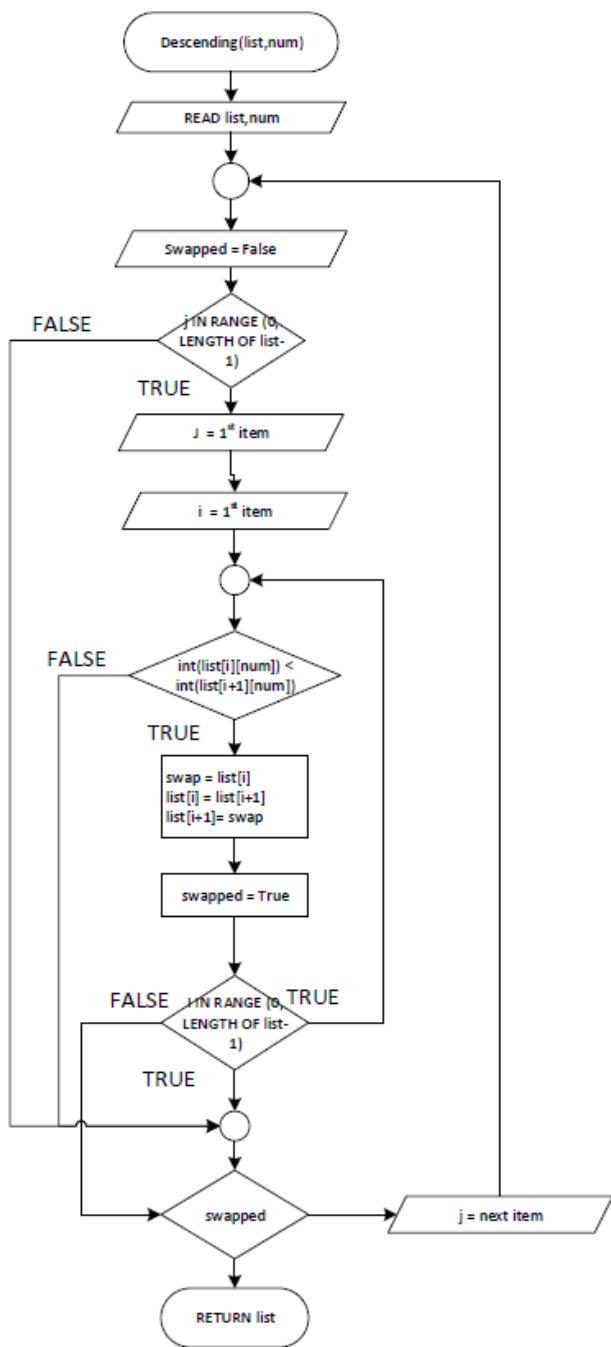
## 2.2 Flowchart

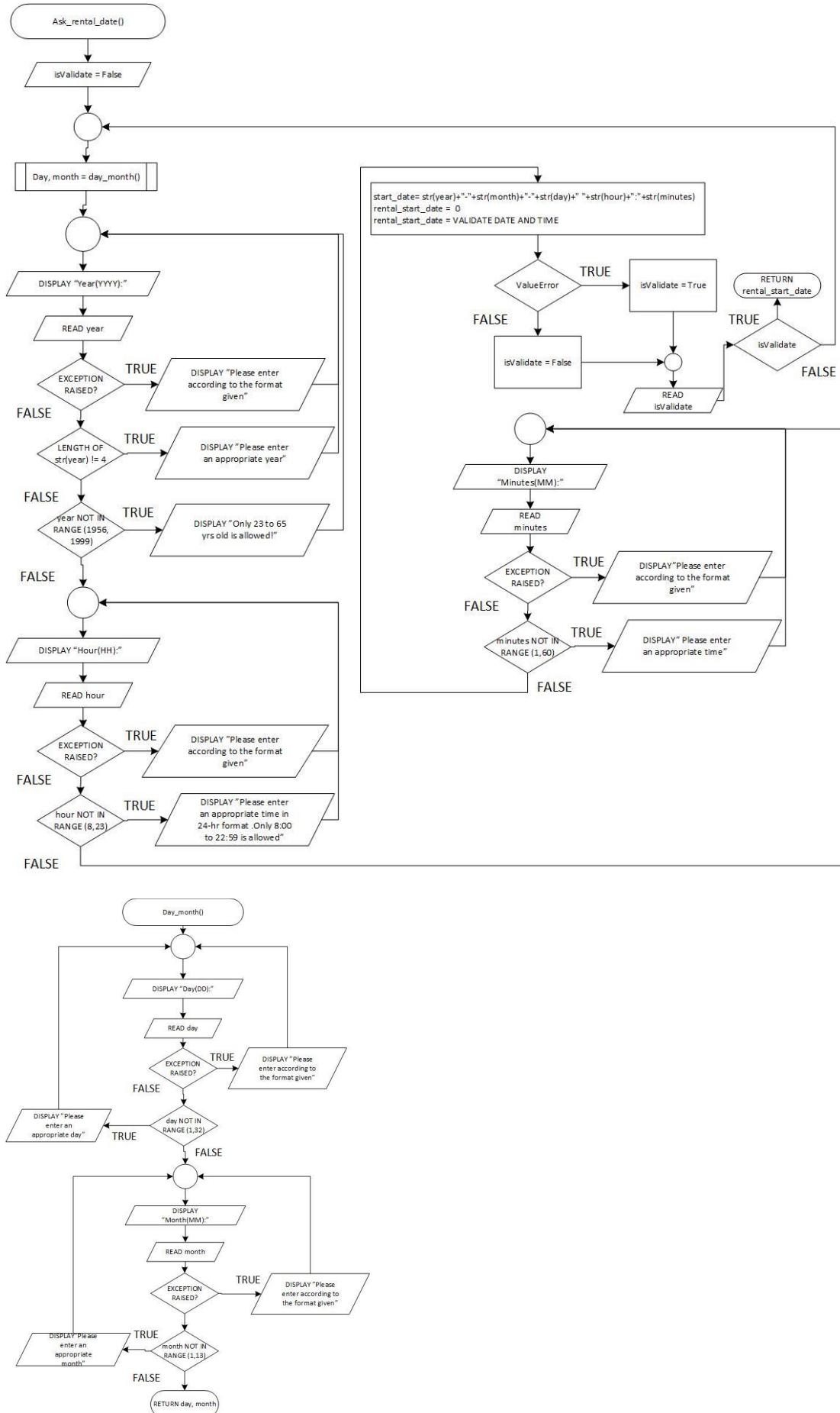


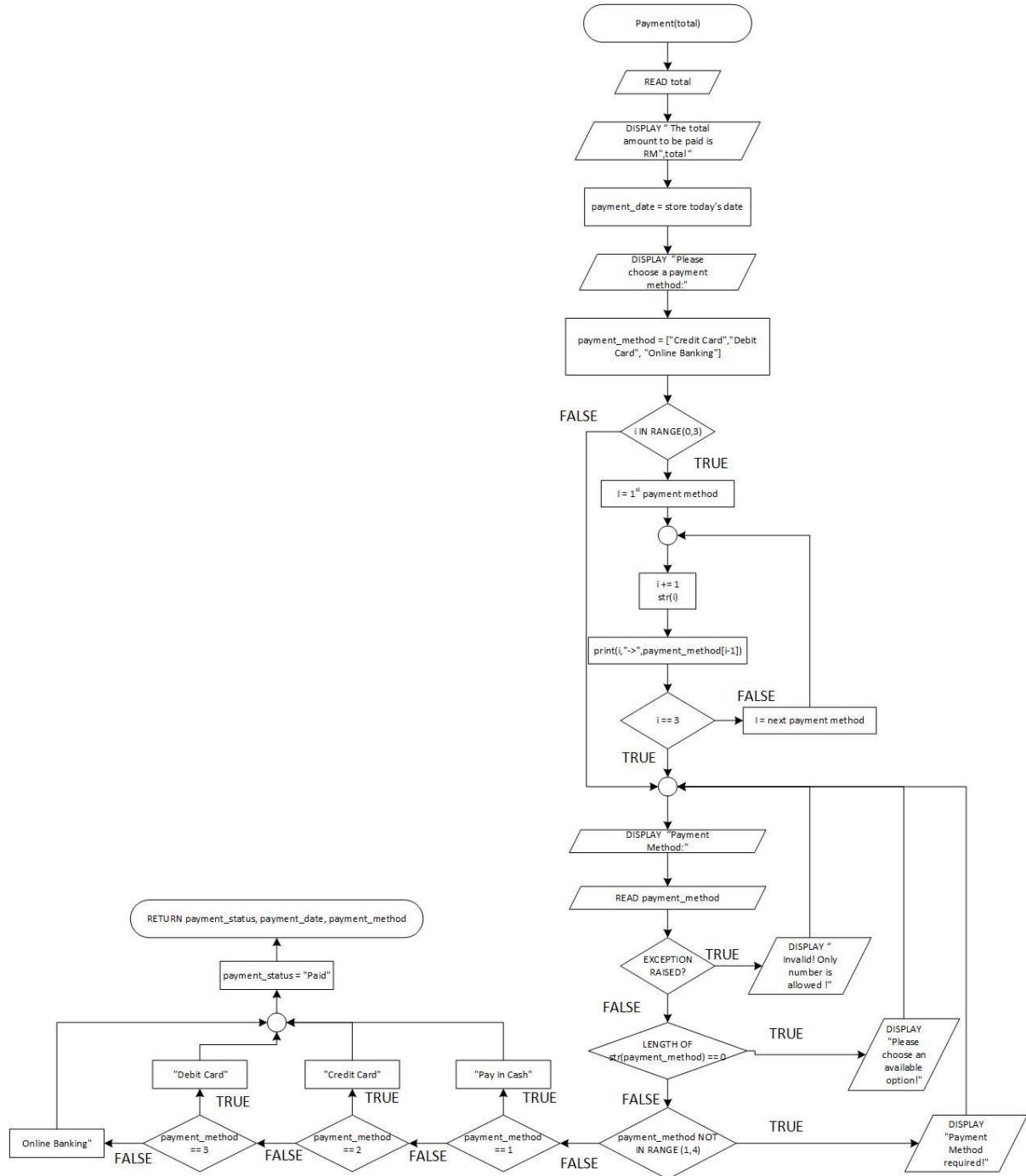


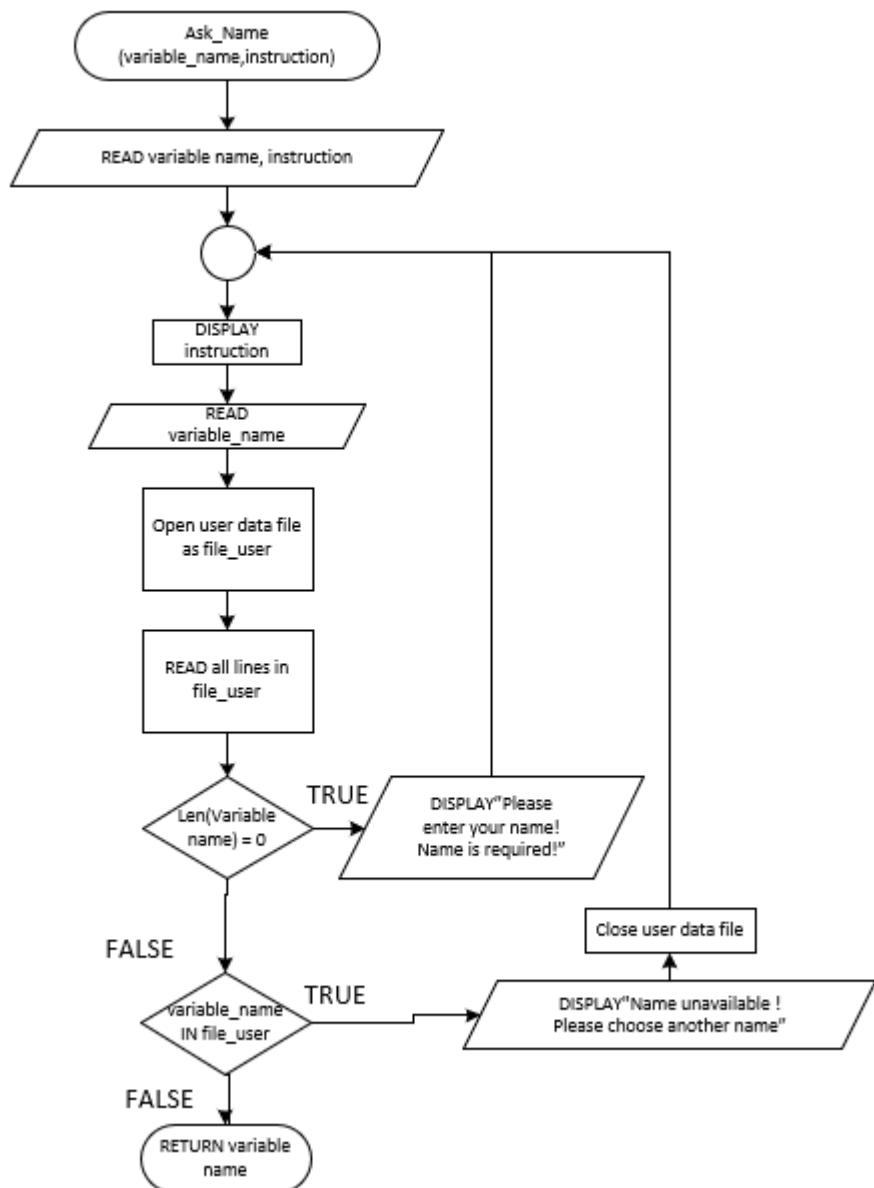


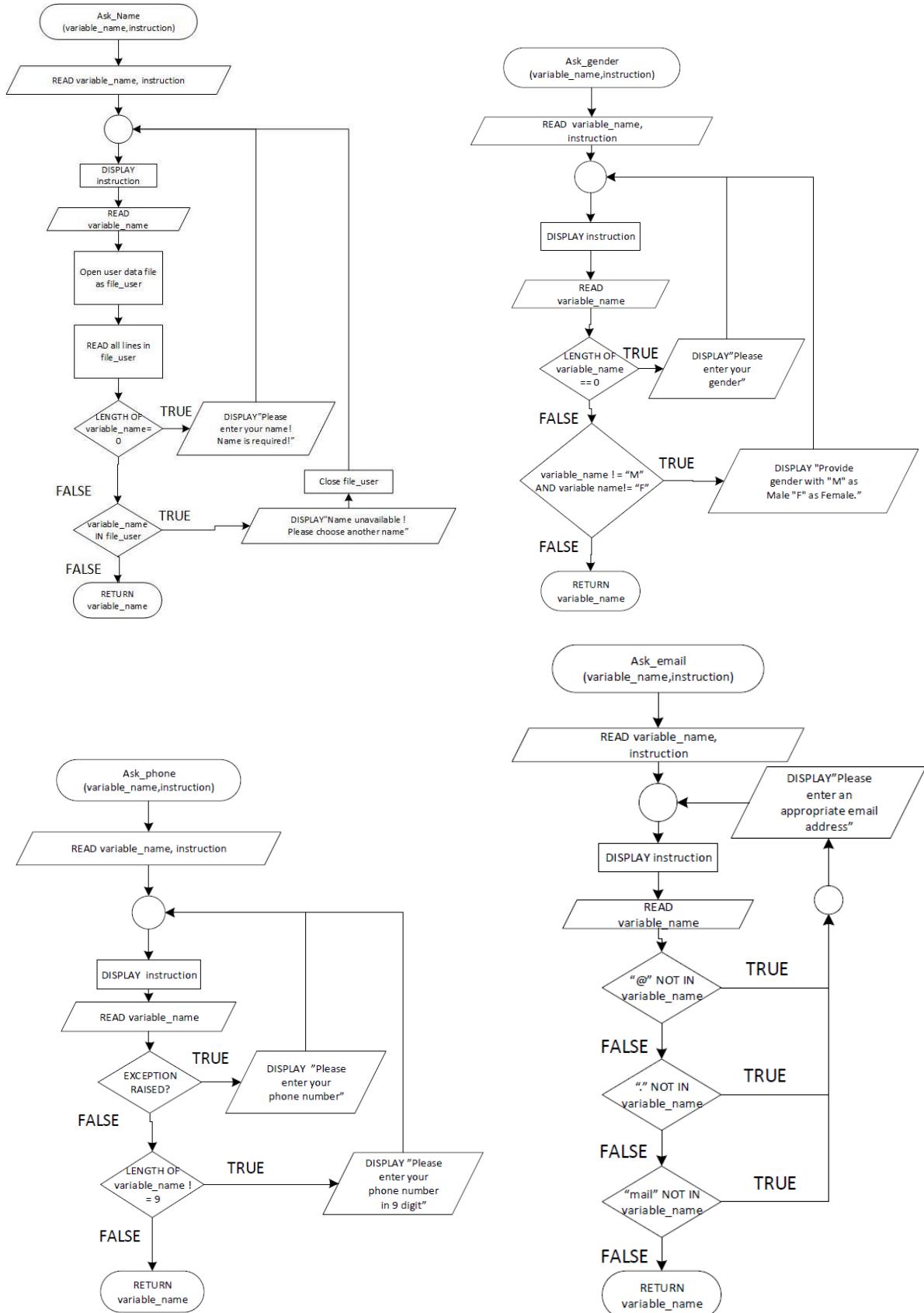


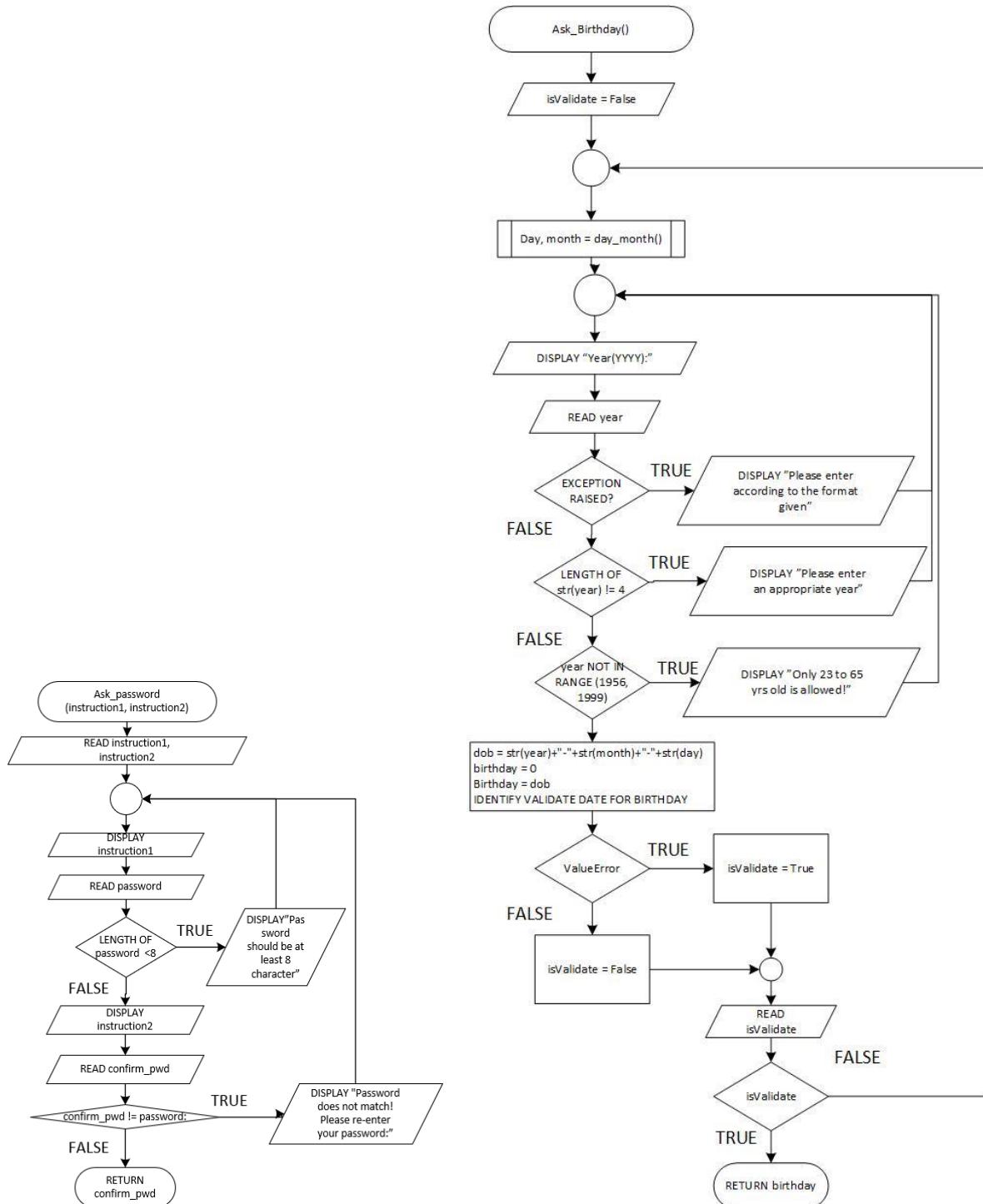


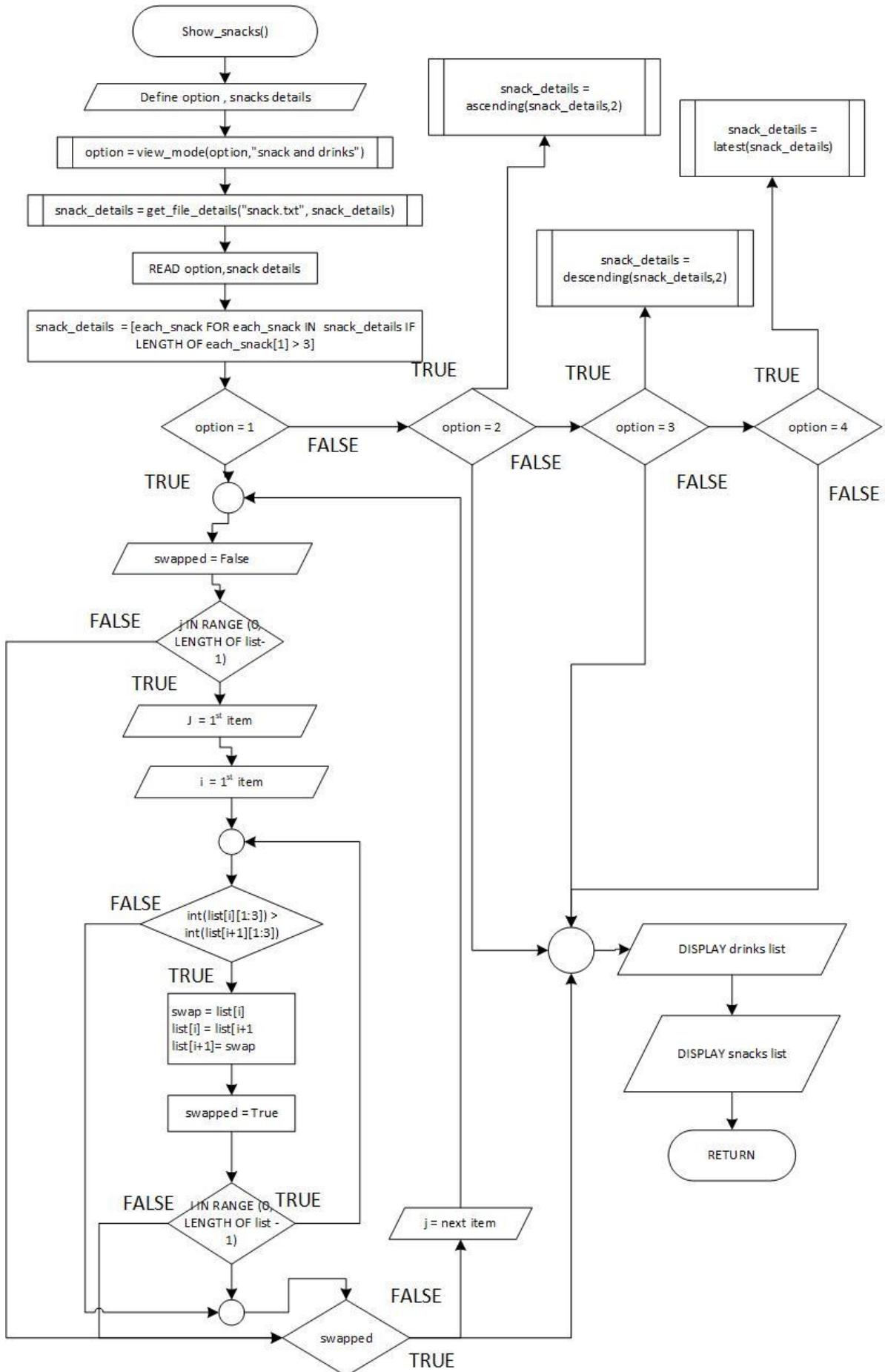


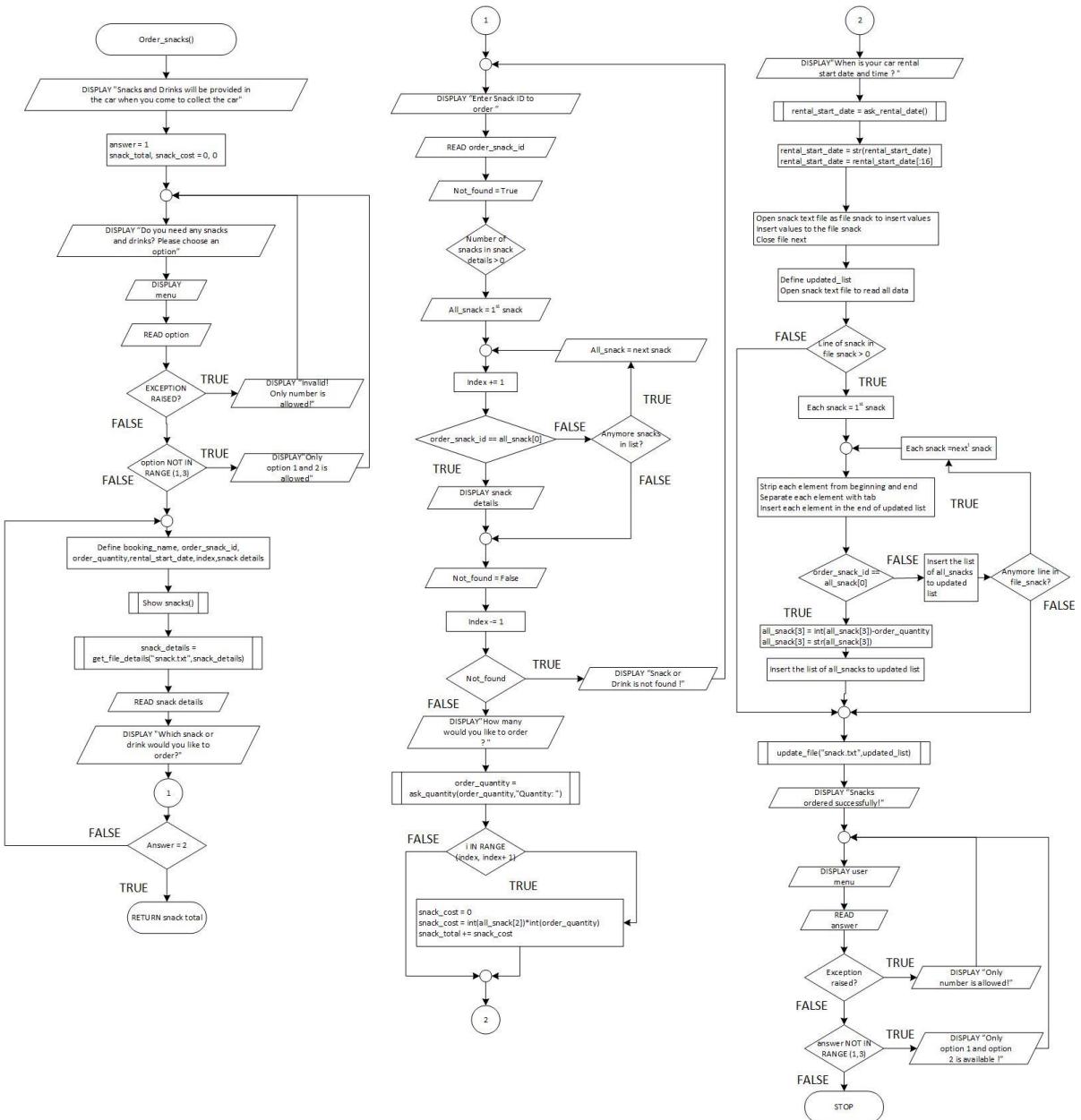


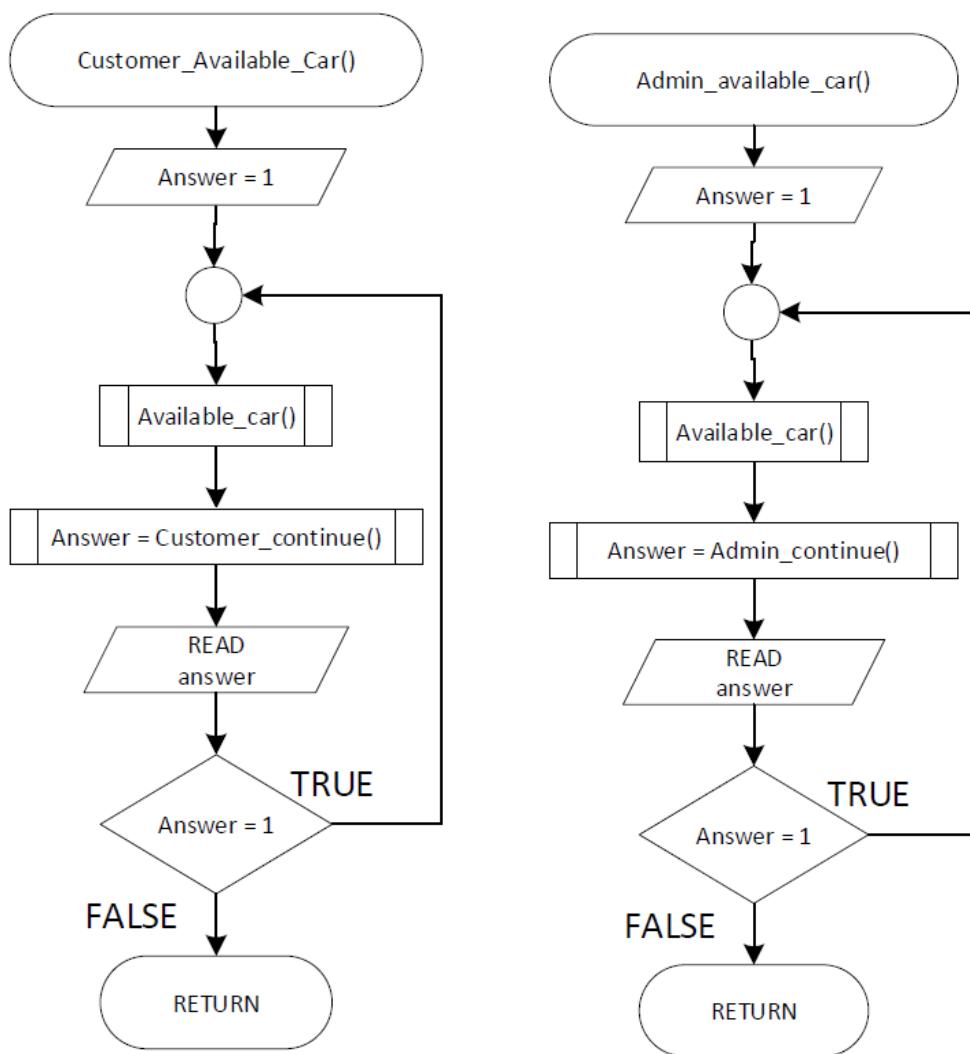
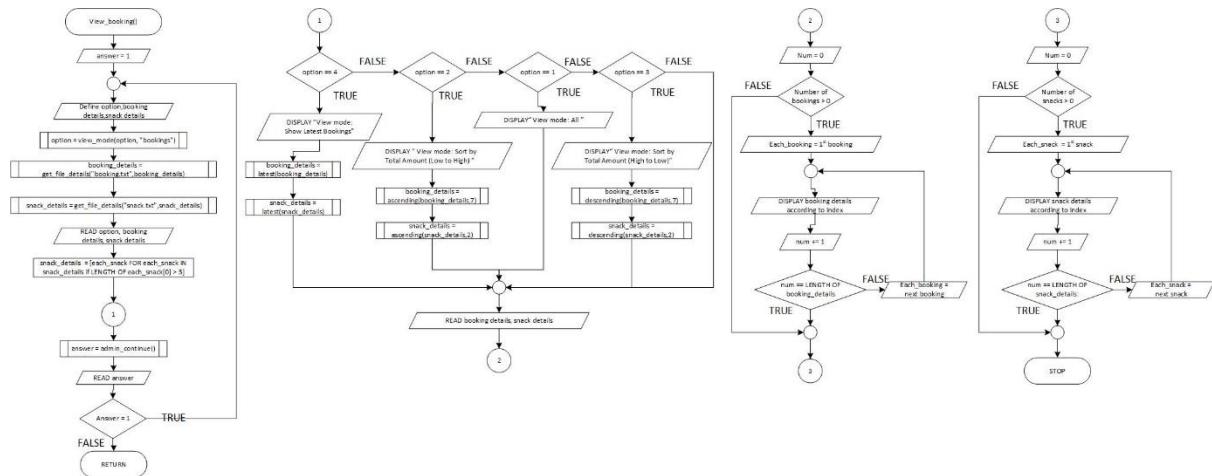


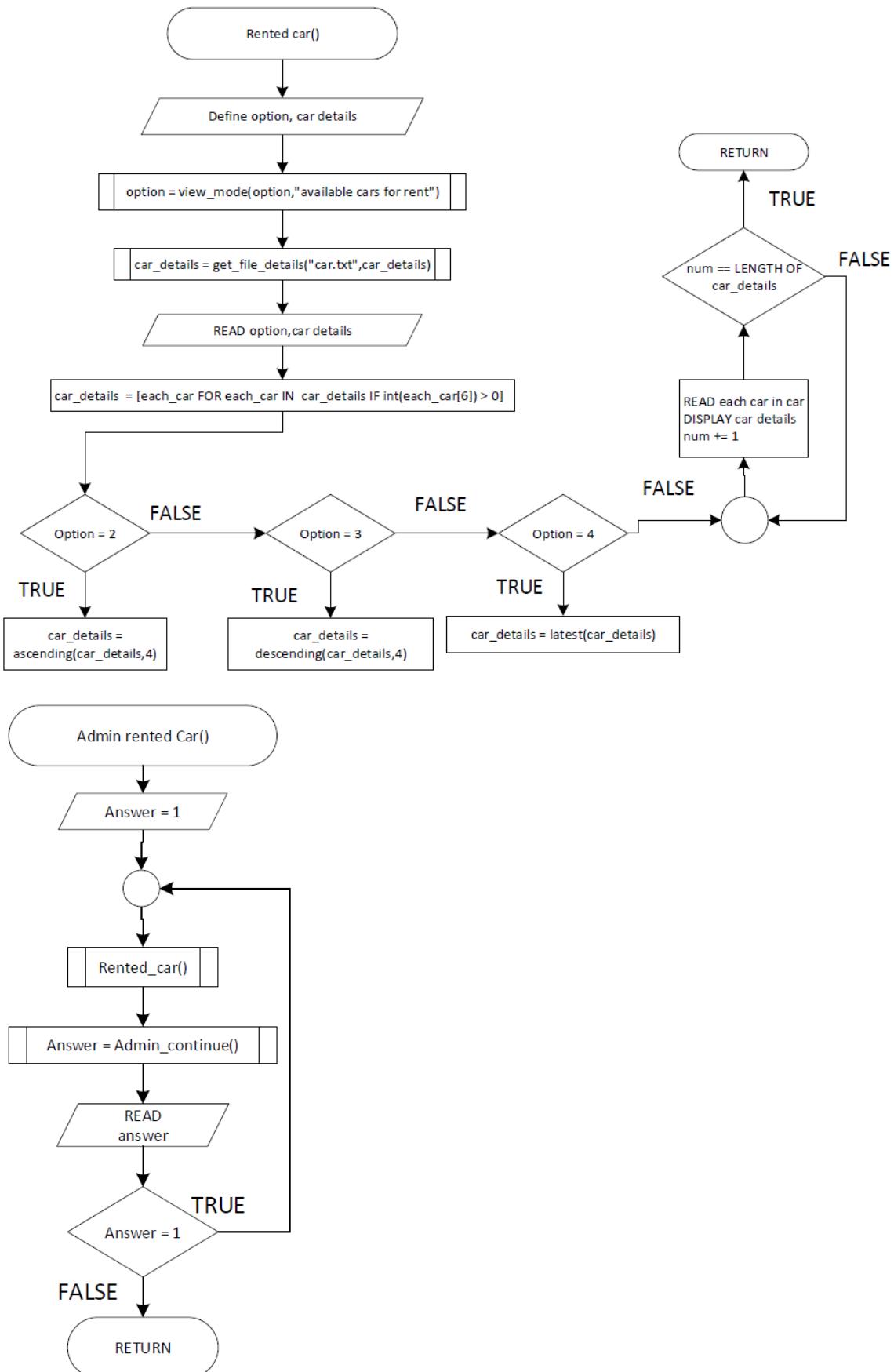


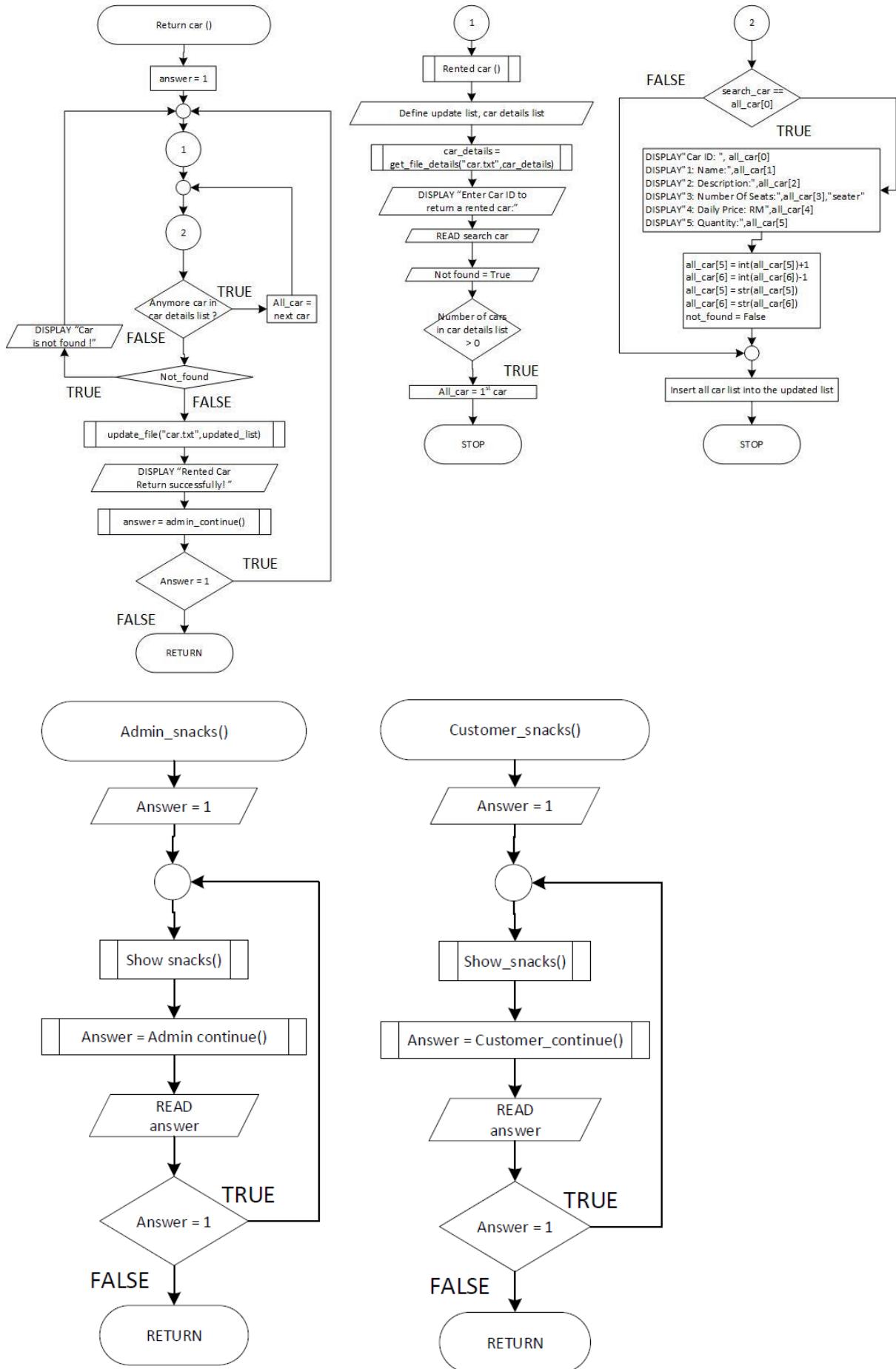


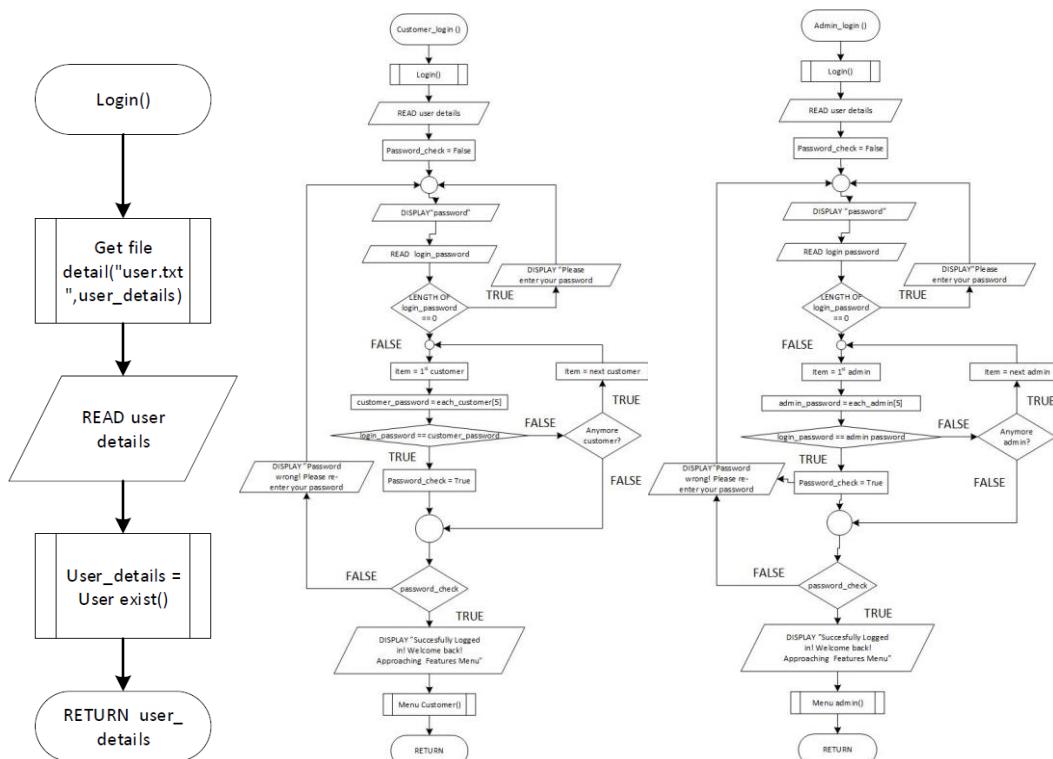
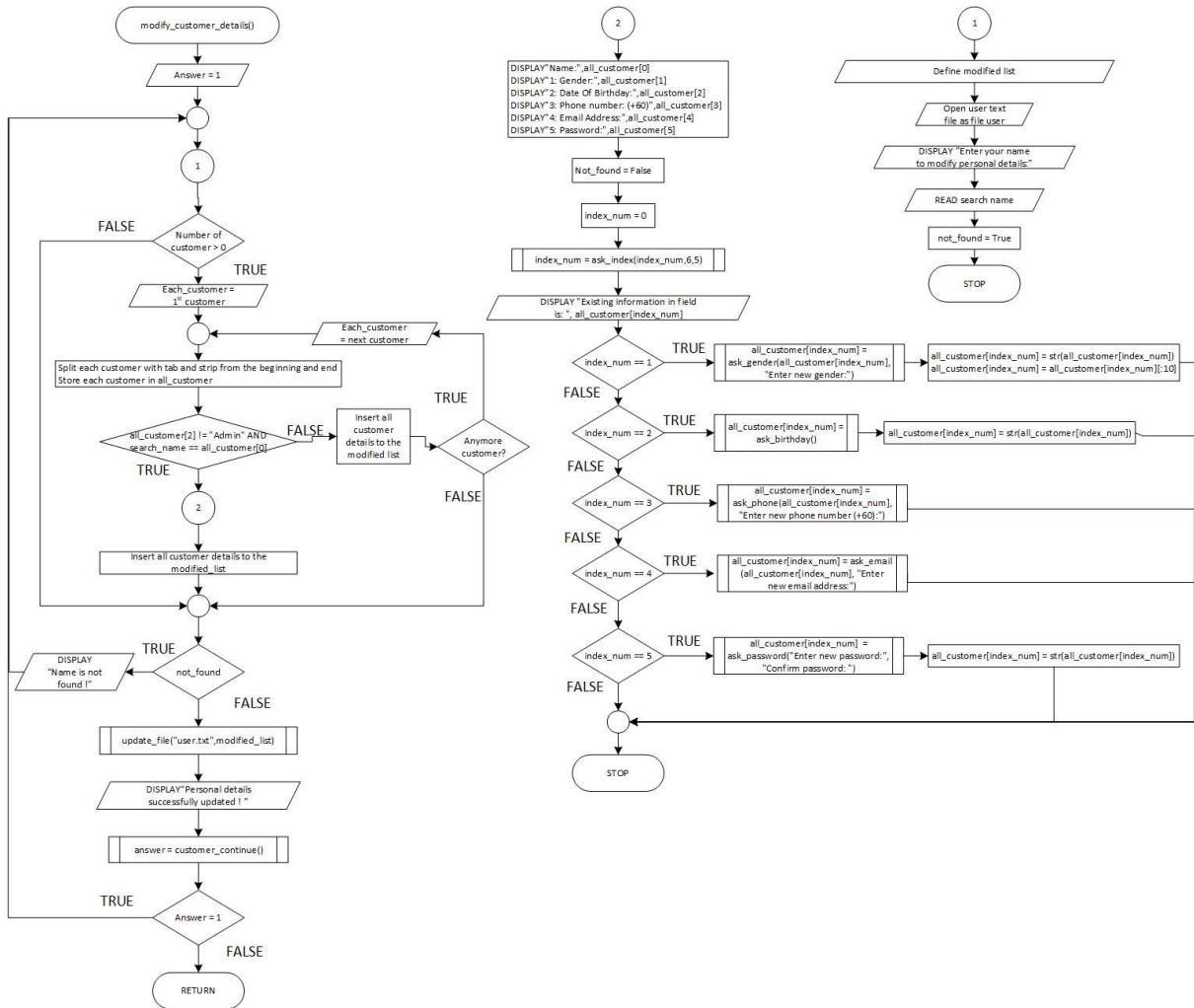


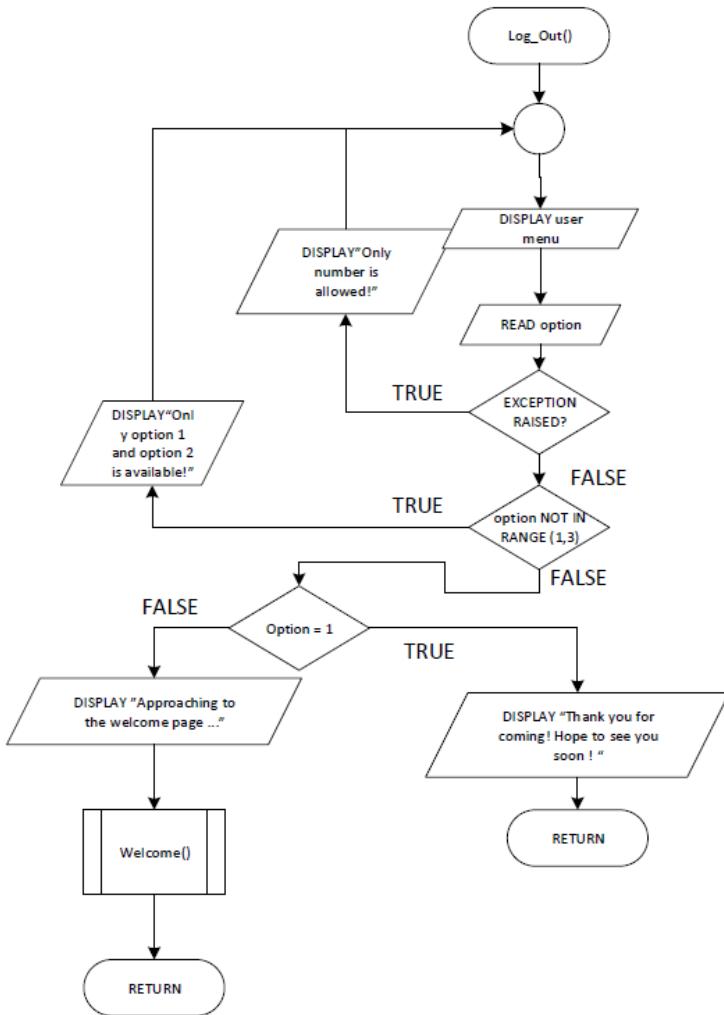
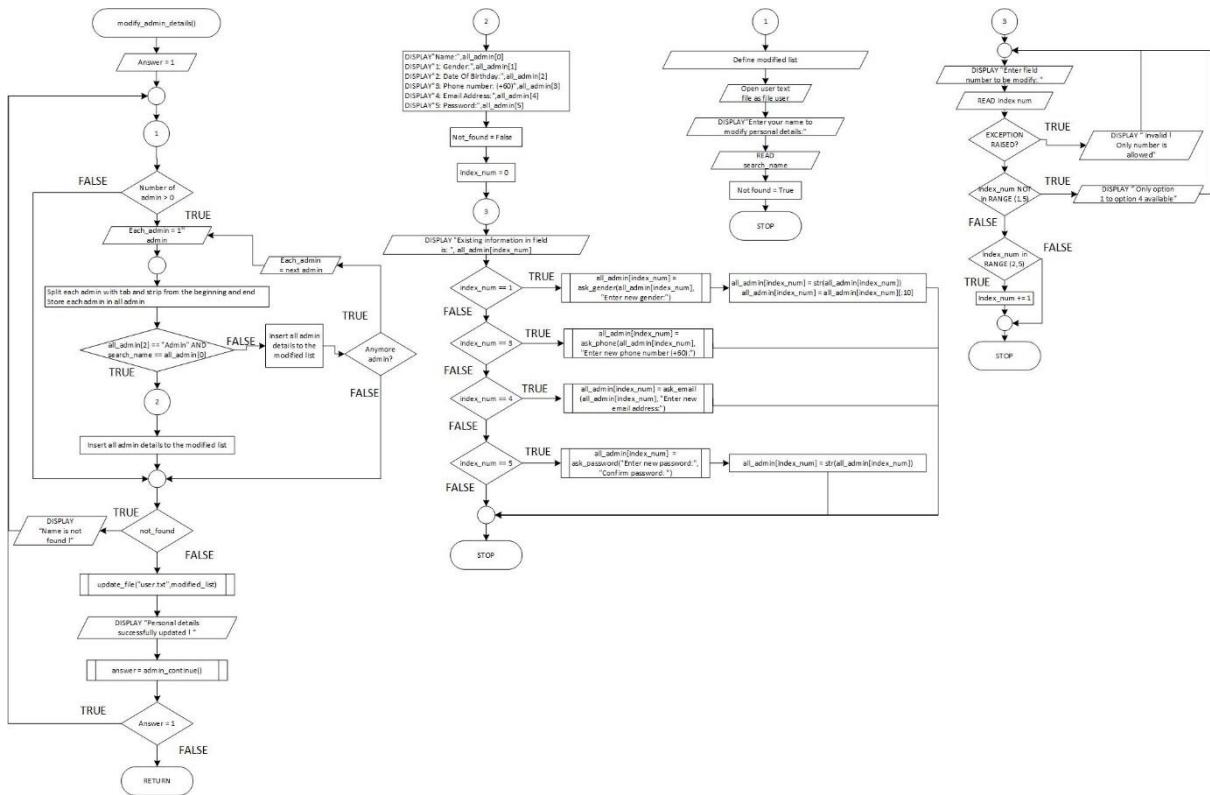


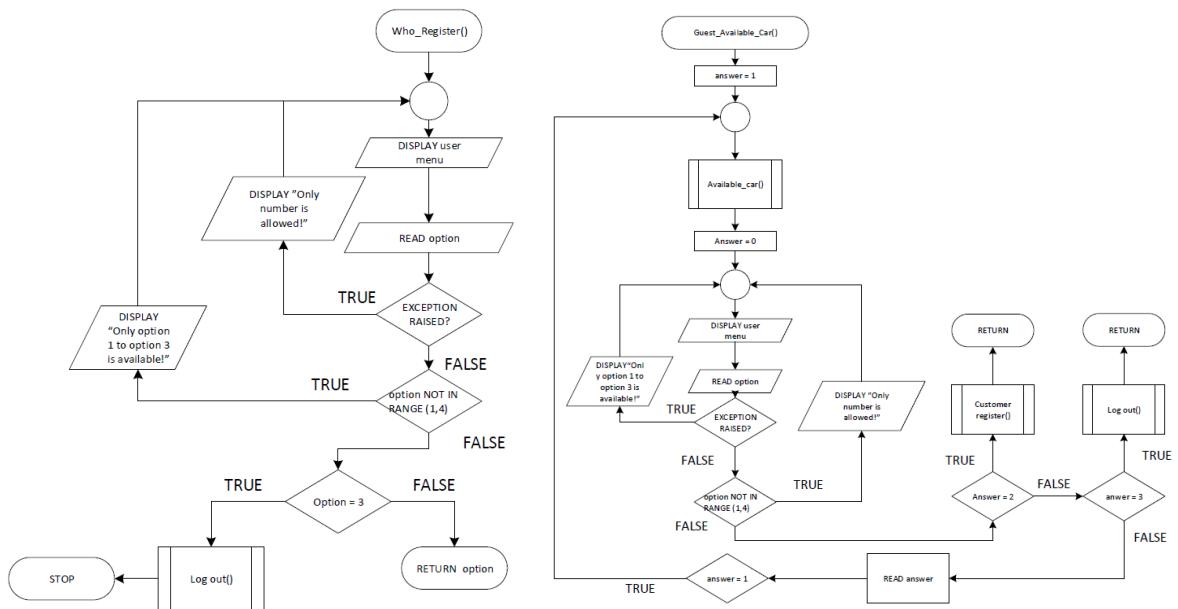
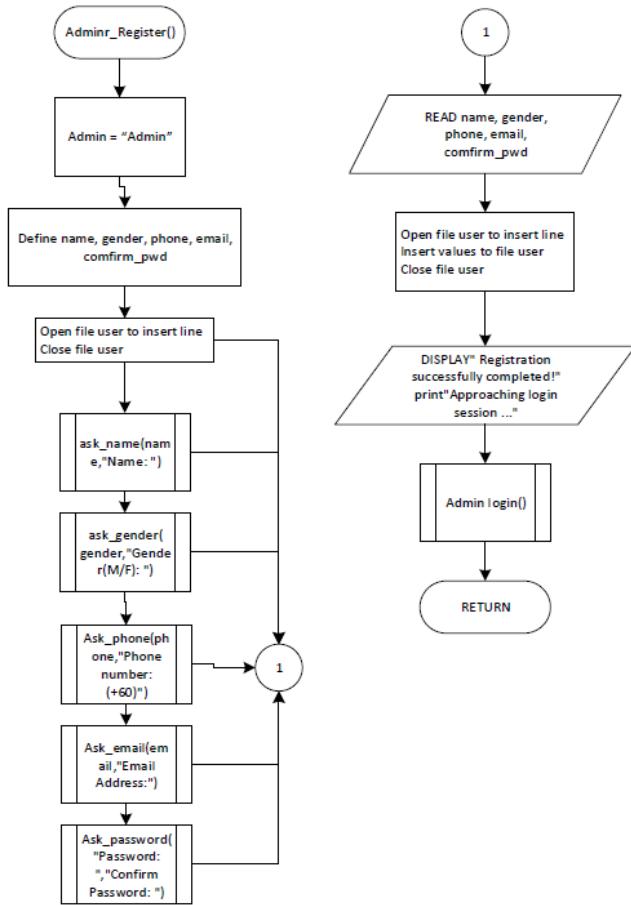


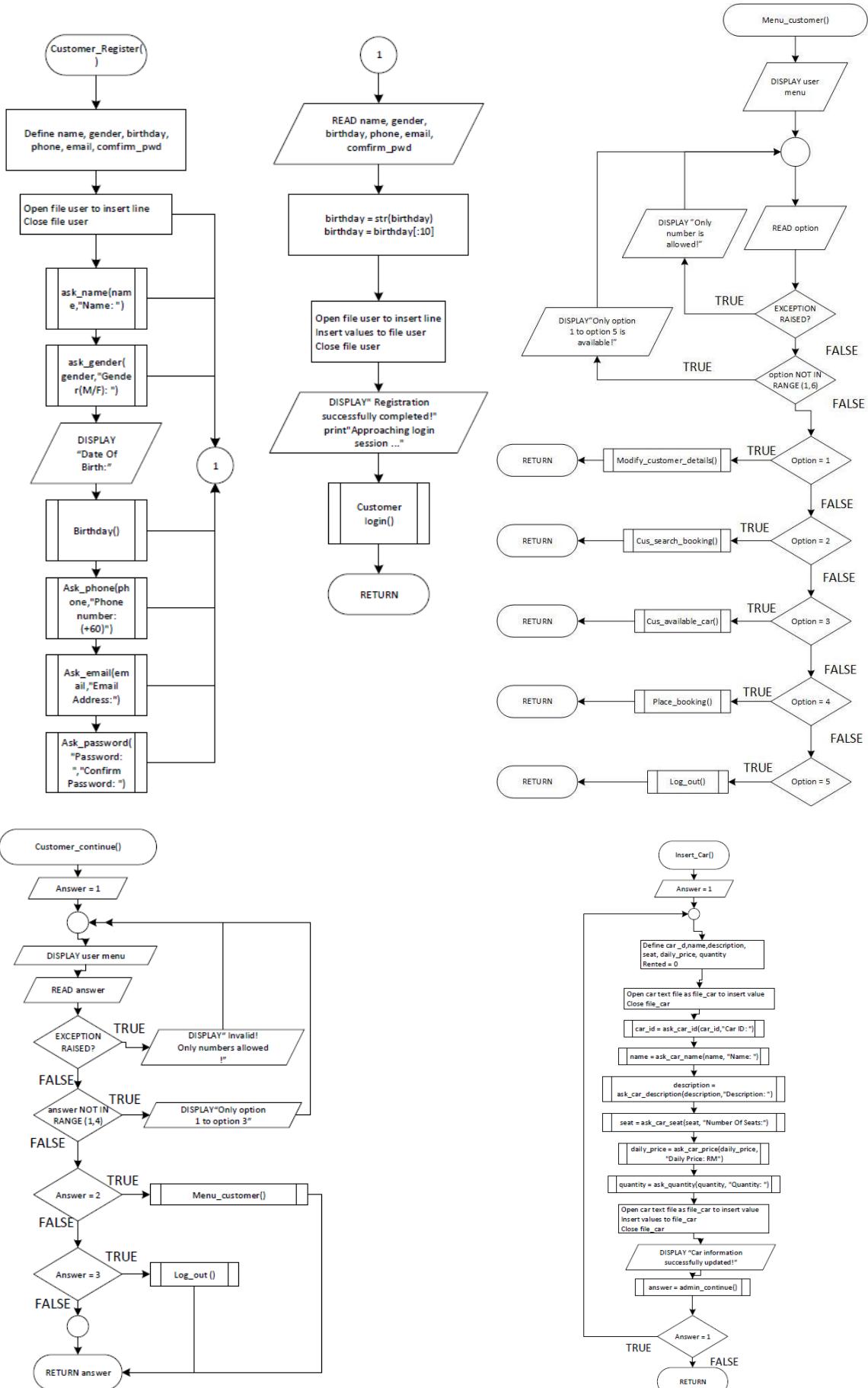


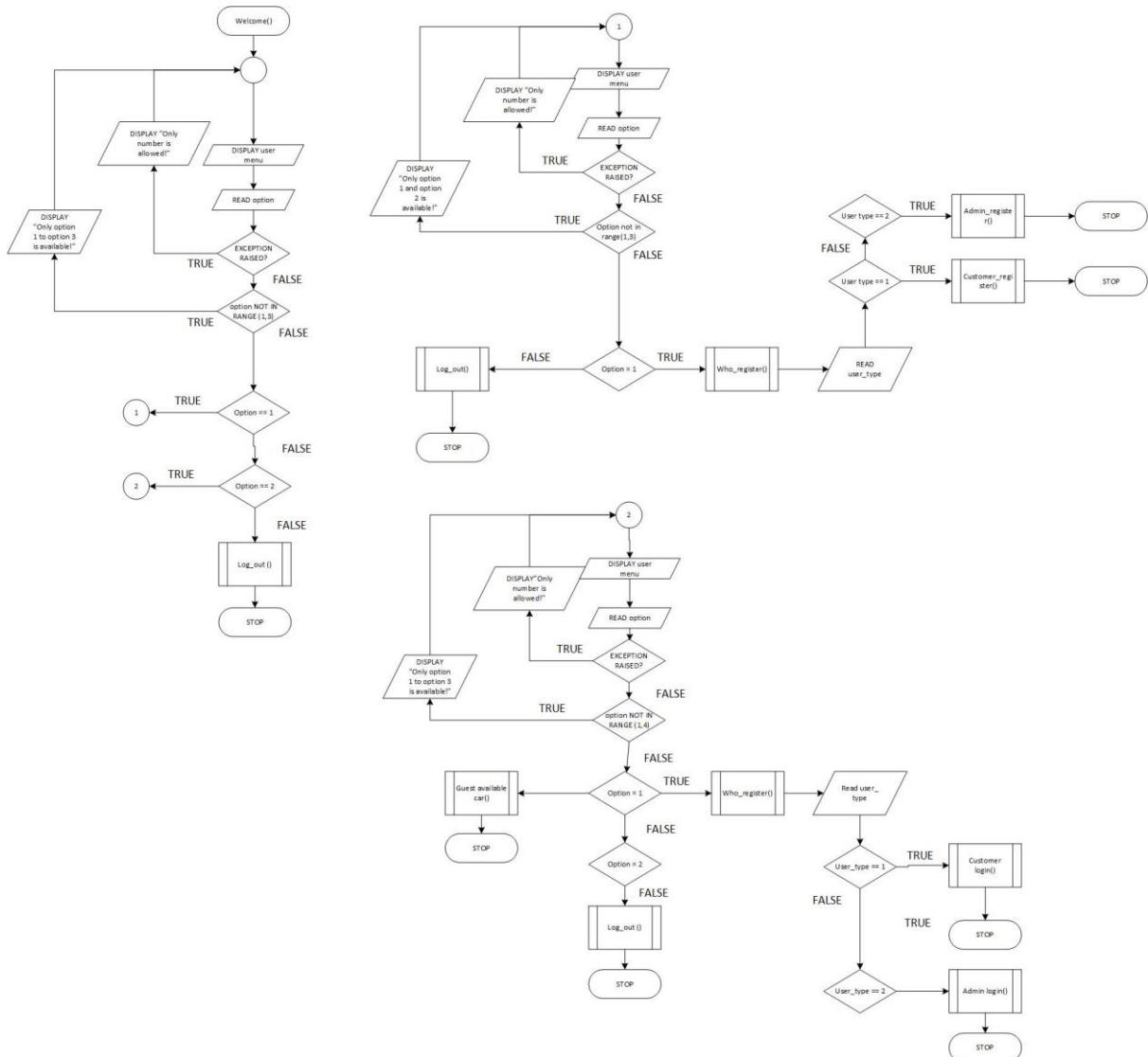


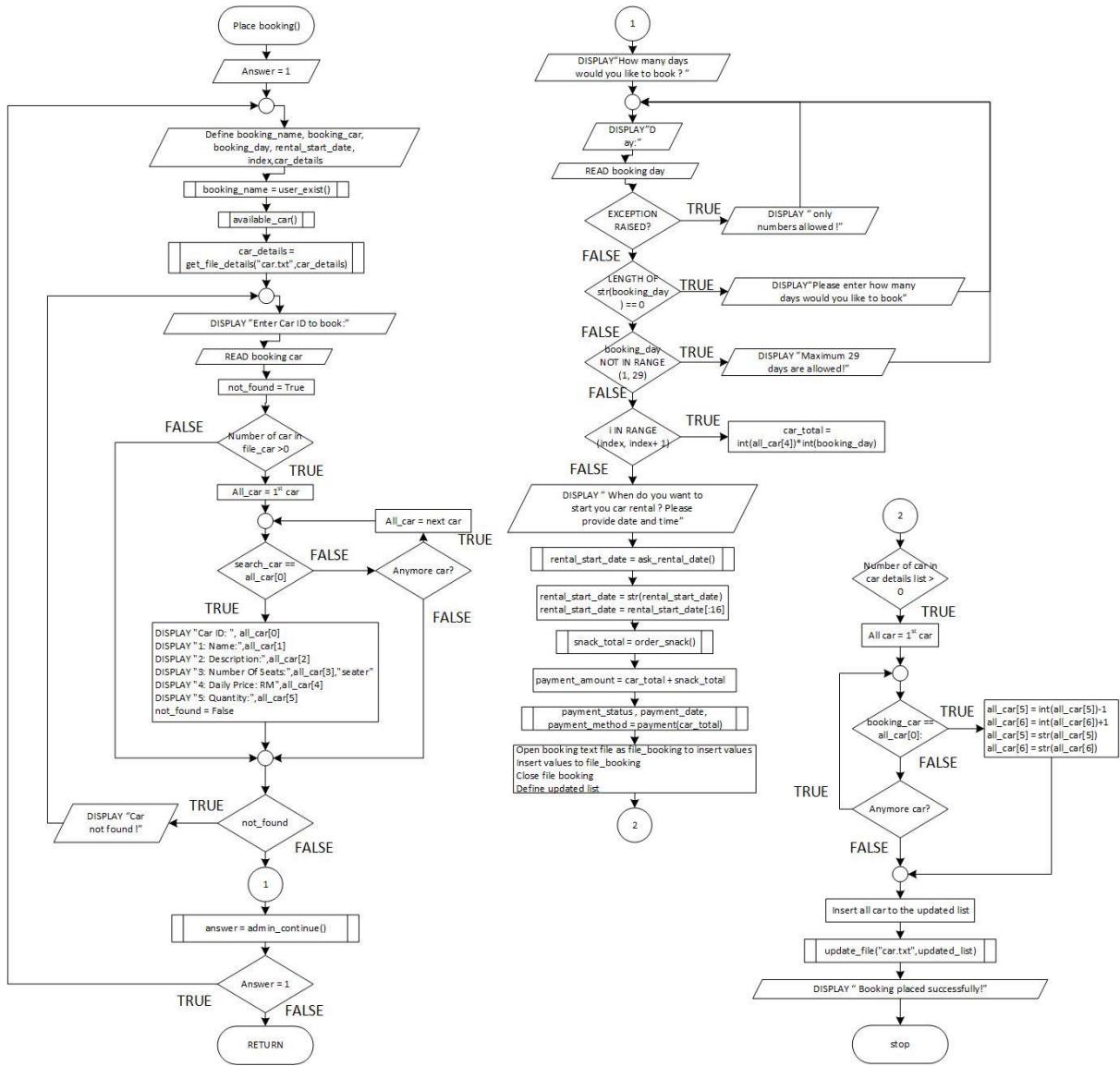


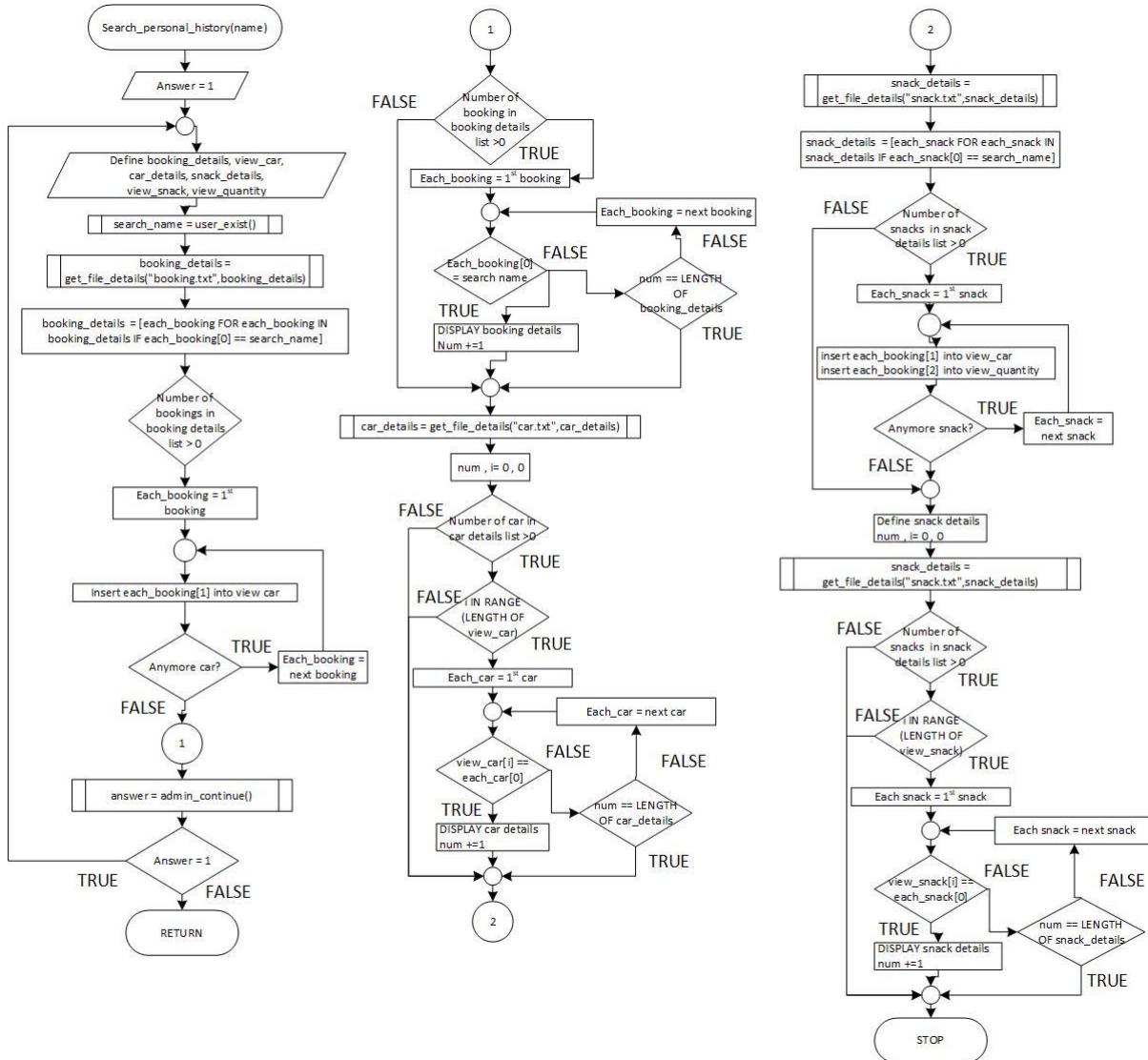


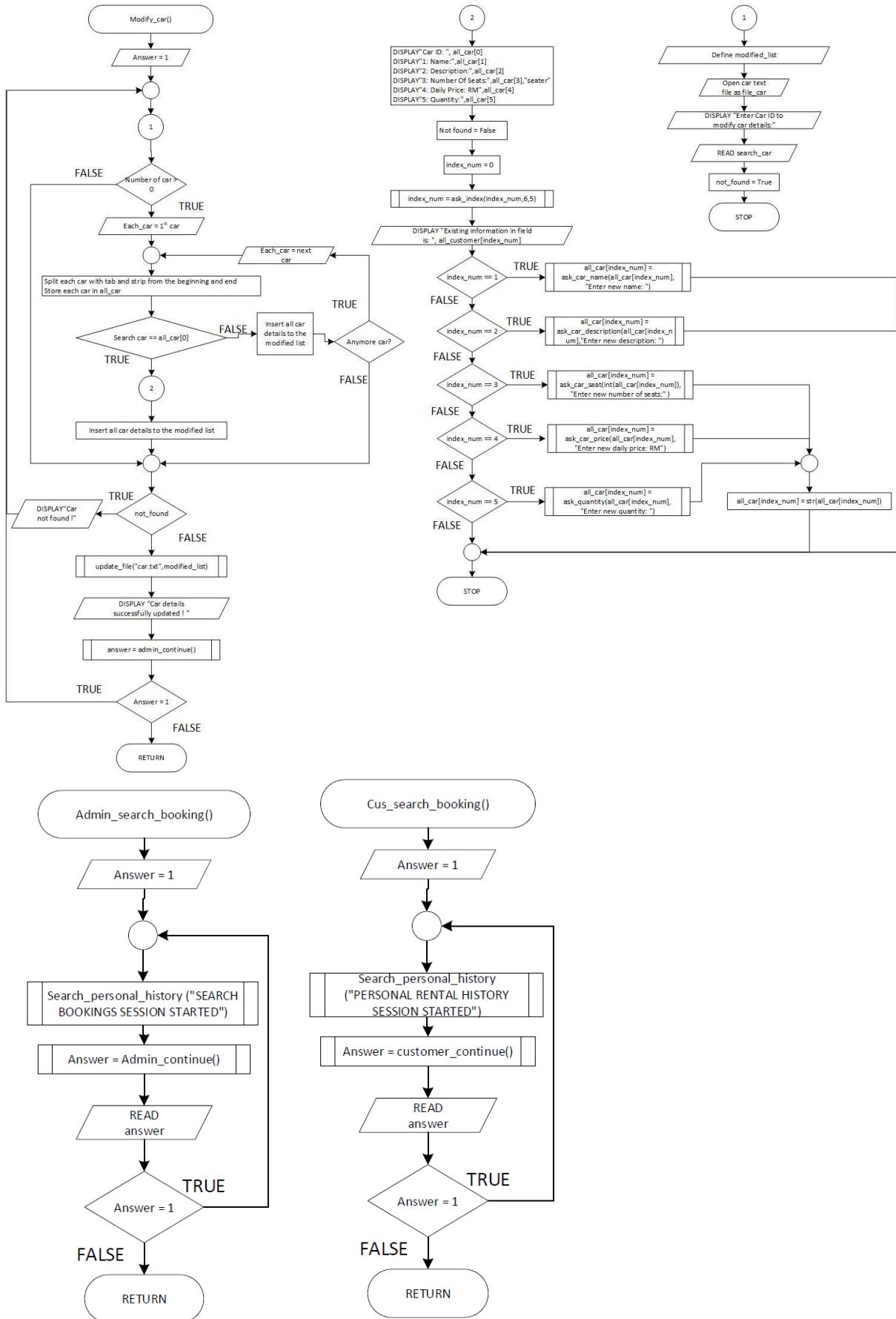


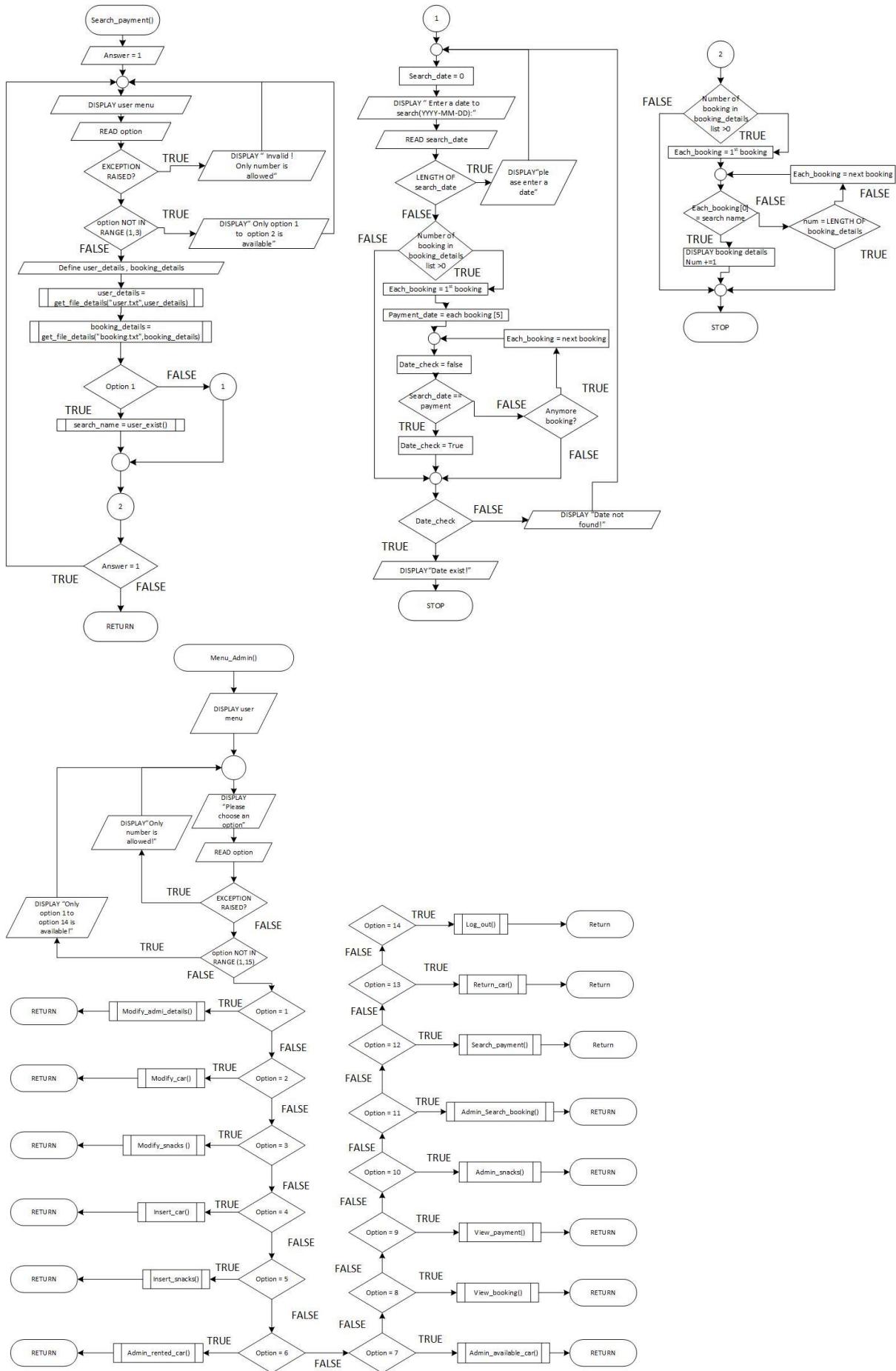


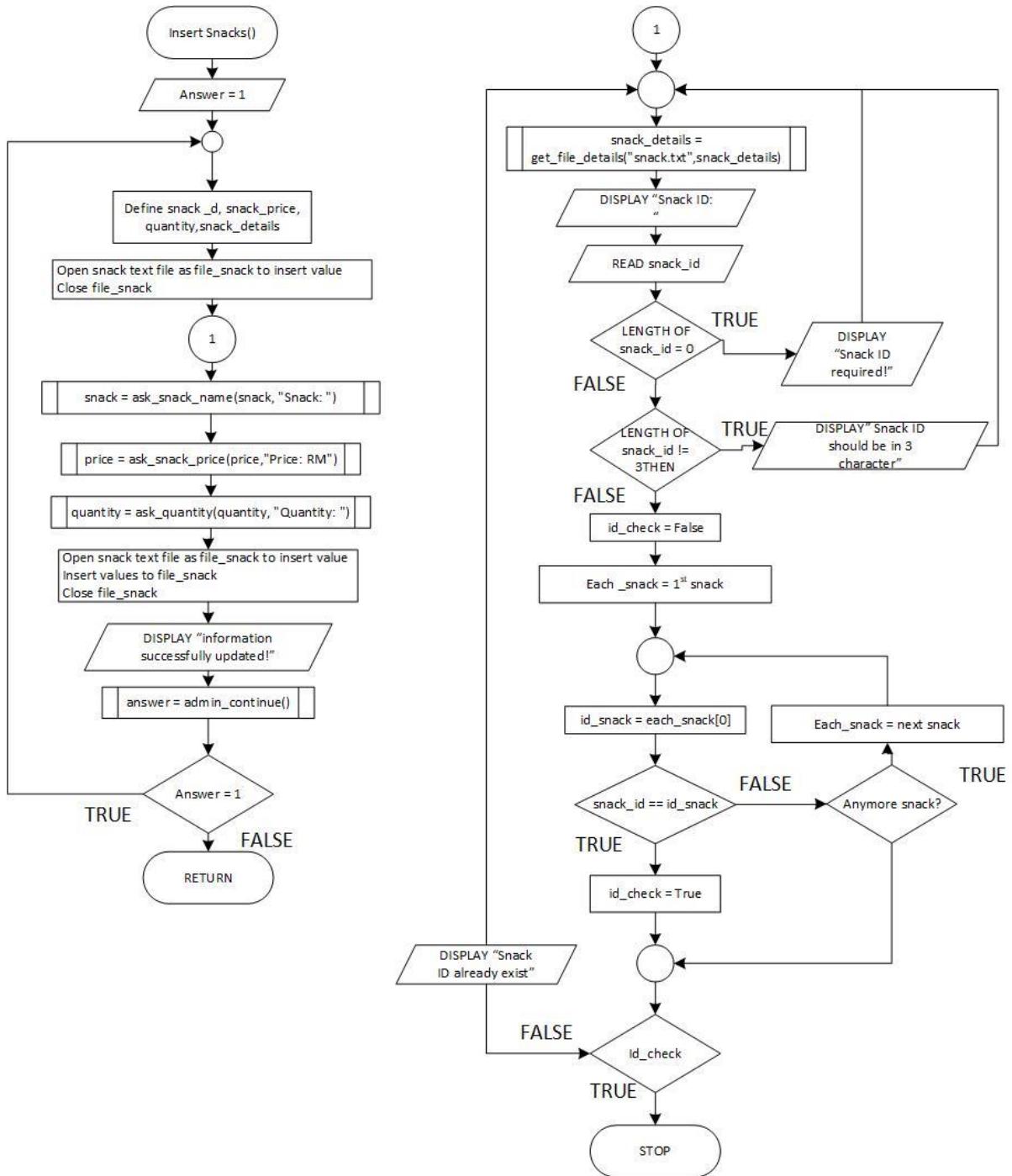


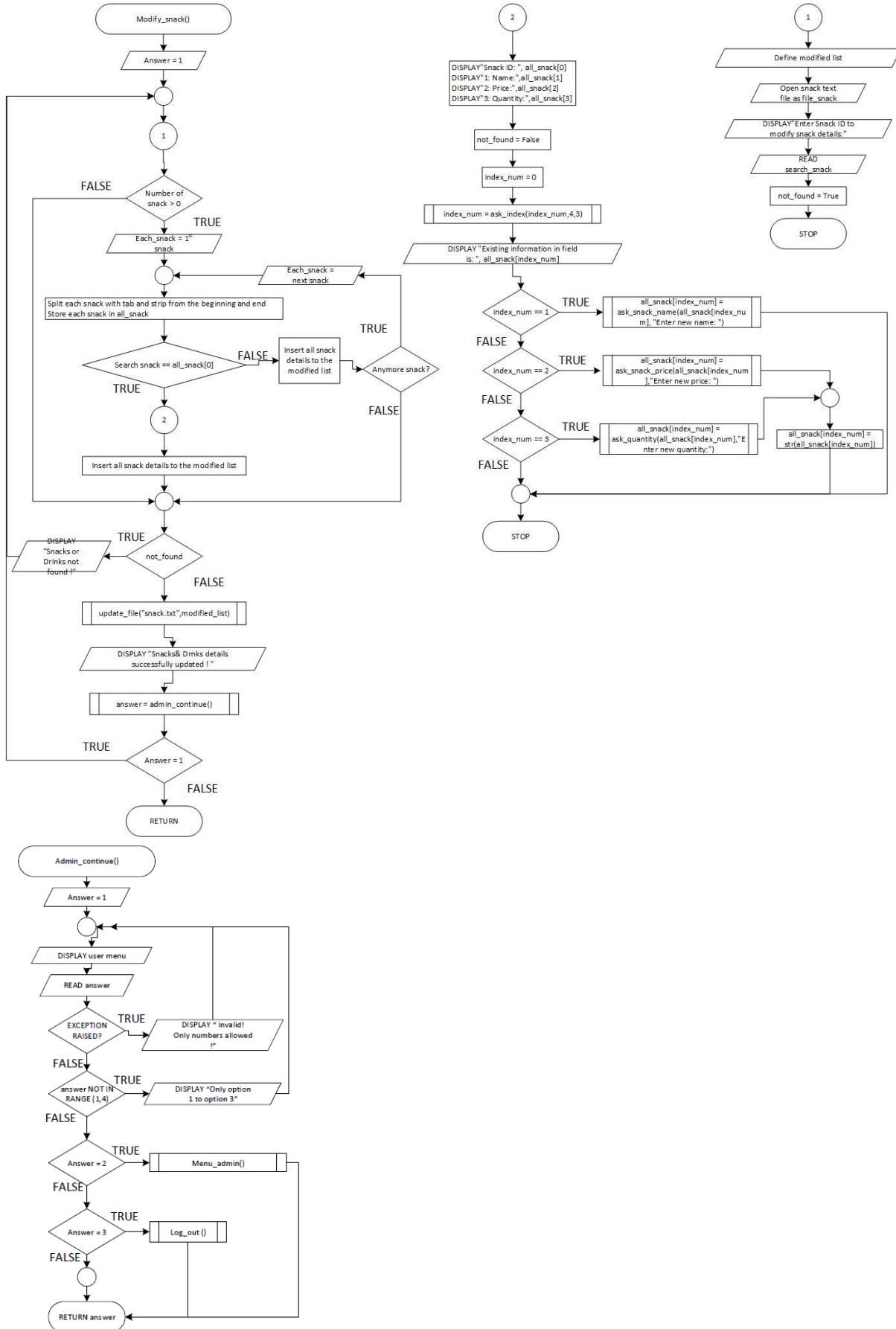












### 3.0 Program source code with explanation

#### Function of Welcome

```

def welcome():
    print("***** WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) ! *****")
    print("----- | WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) ! | -----")
    print("\nWe are glad that you are here !\n")

    while(True): #ask user login/register
        option = 0
        try:
            print("*****")
            option = int(input("Do you have an account? Please choose an option\n1: Yes\n2: No\n3: Exit\nOption: "))
            if option not in range(1,4):
                print("=====\nOnly option 1 to option 3 is available!\n=====")
            else:
                break
        except:
            print("=====\nInvalid! Only number is allowed!\n=====")
            continue

        if option == 1:
            while(True):
                option = 0
                try:
                    print("*****")
                    option = int(input("Do you want to login now? Please choose an option\n1: Login\n2: Exit\nOption: "))
                    if option not in range(1,3):
                        print("=====\nOnly option 1 to option 3 is available!\n=====")
                    else:
                        break
                except:
                    print("=====\nInvalid! Only number is allowed!\n=====")
                    continue

                if option == 1:
                    user_type = who_register()
                    if user_type == 1:
                        customer_login()
                    elif user_type == 2:
                        admin_login()
                else:
                    log_out()

        elif option == 2:
            while(True):
                option = 0
                try:
                    print("*****")
                    option = int(input("Do you want to register an account now?\n1: Register now\n2: View all cars available for rent\n3: Exit\nOption: "))
                    if option not in range(1,4):
                        print("=====\nOnly option 1 to option 3 is available!\n=====")
                    else:
                        break
                except:
                    print("=====\nInvalid! Only number is allowed!\n=====")
                    continue

                if option == 1:
                    user_type = who_register()
                    if user_type == 1:
                        customer_register()
                    elif user_type == 2:
                        admin_register()

                elif option == 2:
                    guest_available_car()

                elif option == 3:
                    log_out()
            else:
                log_out()
        return
    
```

As shown in the welcome function, option is define as an variable. The variable option initially stored “0” in order to refresh and receive input from the user. User will input the option of them by providing numbers. The try and except block will do the data validation. After that the option variable will store the number which is the choice of user.

Furthermore, control structure are used in the welcome function to indicate where the user should approach. The while loop used in this welcome function is to loop the message of asking user to input their option again until the user chooses an available option from the message given. While loop will only stop whenever the user gave an valid number in the welcome function.

Moreover, if statement is very useful in welcome function as the if statement created will lead the user to where the user should go. If statement will execute the code when the statement is true, else the if statement will go through the remaining if statement. If the remaining if statement does not have a true statement, the if statement will execute the else statement and end the if statement. If statement is used to test the user option in welcome function. For instance, when the user option is 3, if statement will go through the first two if statement and find the statement is false, when the third if statement is true, the function of log out will be executed.

Lastly, many functions are prepared to be called by user in this welcome function. Welcome function will lead the user to the features that users intend to do. The functions will be called when the user choose to access the features of the functions. For instance, log out function will be called when the option of the user is to exit the online car rental system.

### Function of Get File Details

```
def get_file_details(file,list):
    list = []
    with open(file,"r") as file_snack:
        for each in file_snack:
            each_details = each.strip().split("\t")
            list.append(each_details)
    return list
```

Get file details function convert the lines and element of the text file to a nested list. The file is open in read mode to read all lines in file. In this get file details function, for loop is used to get each line of the text file. Each line of the file will first split into a list by separating by a tab “\t”. After that, each index in the list will be stripped from the beginning and the end. With the string function of append, the list of each\_details will be appended to the list which is inserted at the end of the list. Later on, the for loop will loop again until all line are looped. Thus, the next line in the file will go through the same process as the previous line and append to the list. The get file details function will return the list with all file details after the for loop end. The list will then be able to used in other functions by giving a file name and list to get a file details.

## Function of Update File

```
def update_file(file_name,list):
    with open(file_name, "w") as file_name:
        i = 0
        while (i< len(list)):
            detail = "\t".join(list[i])
            file_name.write(detail+"\n")
            i += 1
```

Update file is used to update all the file details when modifying is needed. When the file name and list is given as arguments, update file function will open the file in write mode, which everything in the file will be erased. A variable I will be defined by assigning 0 to it. While loop will run when I is smaller than the length of the list given. The join string function will join elements in the list with a tab “\t” and store into the detail list. The detail will be write to the file name line by line with a “\n” new line at the end of each line. I will be increase by one for each iteration and stop when the I is smaller than the length of the list.

## **4.0 Additional Features Source Code with Explanation**

The additional features is to sell snack to the customer when customer do a booking. All snacks purchased will be provided in the car, when the customer comes to collect the rental car.

## **Function Of Order Snack**

```

#decrease quantity
updated_list = []
with open("snack.txt","r") as file_snack:
    for each_snack in file_snack:
        all_snack = each_snack.strip().split("\t") #split info into list
        if order_snack_id == all_snack[0]:
            while(True):
                all_snack[3] = int(all_snack[3])-order_quantity # minus quantity
                all_snack[3] = str(all_snack[3])
                break
            updated_list.append(all_snack) #append all info to list
#update the file
update_file("snack.txt",updated_list)
print("~~~ * * * * * Snacks ordered successfully! ~ ~ ~ ~ ~ ~ ~")

while(True):
    try:
        #user input
        print(" * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *")
        answer = int(input("Do you want to continue? Please choose an option\n1: Yes\n2: No\nOption: "))
    except:
        print("=====\\nInvalid! Only number is allowed!\\n=====")
    #check valid user input
    if answer not in range(1,3):
        print("=====\\nOnly option 1 and option 2 is available !\\n=====")
    else:
        break
    if answer == 2:
        break
return snack_total

```

Order snacks function allow user to order snack when user place a booking. User can choose to order and not order snacks and drinks. The payment date will be stored in the booking text file. While the other information of snacks and drinks order will be stored in the snack text file. Customers need to enter the registered name in order to order the snacks and drinks. Quantity of snacks will be updated to the snack file.

## Function of Ask Snack Name

```

def ask_snack_name(variable_name, instruction):
    while (True):
        variable_name = input(" >< >< >< >< >< >< >< >< \n"+instruction)
        if len(variable_name) == 0:
            print("=====\\nSnack Name required!\\n=====")
        else:
            break
    return variable_name

```

Ask snack name function is used to get the snack name from admin. Admin need to enter snack name to proceed.

## Function Of Ask Snack Price

```

def ask_snack_price(variable_name, instruction):
    while(True):
        try:
            variable_name = int(input(" >< >< >< >< >< >< >< >< \n"+instruction))
            if variable_name not in range(1, 10):
                print("=====\\nPrice in RM 1 to RM9 !\\n=====")
            else:
                break
        except:
            print("=====\\nOnly whole numbers allowed!\\n=====")
            continue
    return variable_name

```

Ask snack price function is used to get the snack price from admin. Admin needs to ensure that snack price is between RM1 to RM9.

## Function of Insert Snack

```

def insert_snacks():
    print("----- | INSERT SNACKS & DRINKS SESSION STARTED | ----- \nKindly fill up the following details:")

    answer = 1
    while answer == 1:

        #define variables
        snack_id = ""
        snack = ""
        price = ""
        quantity = ""

        file_snack = open("snack.txt","a") #open new file when no file
        file_snack.close()

        # snack id
        while (True):

            #get snack details
            snack_details = []
            snack_details = get_file_details("snack.txt",snack_details)

            #id input
            snack_id = str(input("Snack ID:"))

            id_check = False
            for each_snack in snack_details:
                id_snack = each_snack[0] #check id existence
                if snack_id == id_snack:
                    id_check = True
                    break
                else:
                    id_check = False
            if len(snack_id) == 0:
                print("===== \nSnack ID required !\n===== ")
            elif len(snack_id) != 3:
                print("===== \nSnack ID should be in 3 characters\n===== ")
            elif not id_check:
                break
            elif id_check:
                print("===== \nSnack ID already exist!\n===== ")

            #snack
            snack = ask_snack_name(snack, "Snack: ")

            #price
            price = ask_snack_price(price,"Price: RM")

            #quantity
            quantity = ask_quantity(quantity,"Quantity: " )

            file_snack = open("snack.txt", "a") #open snack file
            file_snack.write(str(snack_id)+"\t"+snack+"\t"+str(price)+"\t"+str(quantity)+"\n") #insert values
            file_snack.close() #close snack file
            print("~ ` ~ ` ~ ` ~ Snacks information successfully updated! ~ ` ~ ` ~ ` ~")
            answer = admin_continue()

    return

```

Insert snack function is used to get all snack and drinks details from the admin. Information of snack ID, snack name, snack price and snack quantity is got from the user and store into the snack text file. The snack id is checked in the snack file to ensure the snack ID did not repeat.

## Function Of Modify Snacks

```

def modify_snacks():
    print("----- | MODIFY SNACKS & DRINKS SESSION STARTED | ----- \n")

    answer = 1
    while answer == 1:
        modified_list = []
        with open("snack.txt", "r") as file_snack:
            show_snacks() #view snacks and drinks

        #user input
        search_snack = input(" <> <> <> <> <> <> <> <> <> <> <> \nEnter Snack ID to modify details:")

        #check and display car details
        not_found = True
        for each_snack in file_snack:
            #split information with a space into list
            all_snack = each_snack.strip().split("\t")
            if search_snack == all_snack[0]:
                print("===== Snacks Details =====")
                print("Snack ID: ", all_snack[0])
                print("1: Name:",all_snack[1])
                print("2: Price:",all_snack[2])
                print("3: Quantity:",all_snack[3])
                print("===== ===== =====")
                not_found = False

        #user input
        while(True):
            index_num = 0
            index_num = ask_index(index_num,4,3)
            break

        #get new value
        while(True):
            print("Existing information in field is:",all_snack[index_num])

            #name
            if index_num == 1:
                all_snack[index_num] = ask_snack_name(all_snack[index_num], "Enter new name: ")
                break

            #price
            elif index_num == 2:
                all_snack[index_num] = ask_snack_price(all_snack[index_num],"Enter new price: ")
                all_snack[index_num] = str(all_snack[index_num])
                break

            #quantity
            elif index_num == 3:
                all_snack[index_num] = ask_quantity(all_snack[index_num],"Enter new quantity:")
                all_snack[index_num] = str(all_snack[index_num])
                break

        modified_list.append(all_snack)

        #if car not found
        if not_found:
            print("===== \nSnack or Drink is not found ! \n=====")
            continue
        else:
            update_file("snack.txt",modified_list)

    print("~ ~ ~ ~ Snacks & Drinks details successfully updated ! ~ ~ ~ ~")
    answer = admin_continue()
return

```

Modify snacks function allows admin to modify all snack details except the snack ID. First and foremost, admin will need to enter a snack ID in order to search for the snack details and get a specific snack and drinks detail. Then, admin will need to choose which field admin would like to modify before entering the new value. All snack and drinks details including not modified details will be updated to the snack text file.

## Function Of Show Snack

```

def show_snacks():
    print(" ----- | VIEW SNACKS & DRINKS | ----- \n")
    option = 0
    option = view_mode(option,"snack and drinks") #call function

    snack_details = [] #define list
    snack_details = get_file_details("snack.txt", snack_details)

    snack_details = [each_snack for each_snack in snack_details if len(each_snack[1]) > 3] #take required snack from list without creating a new list

    if option == 1:#All
        print("=====" ` View mode: All ` =====")

        #Arrange ascending according to ID
        for j in range(0, len(snack_details)-1):
            swapped = False
            for i in range(0, len(snack_details)-1):
                if snack_details[i][1:3] > snack_details[i+1][1:3]:
                    swap = snack_details[i] #store i in swap first
                    snack_details[i] = snack_details[i+1] #store i+1 to i
                    snack_details[i+1]= swap #store swap(which is i previously) to i+1
                    swapped = True
                if not swapped:
                    break

    elif option == 2: #Low to high
        print("===== ` View mode: Sort by Price (Low to High) ` =====")

        #Arrange ascending according to price
        for j in range(0, len(snack_details)-1):
            swapped = False
            for i in range(0, len(snack_details)-1):
                if snack_details[i][2][0] > snack_details[i+1][2][0]:
                    swap = snack_details[i] #store i in swap first
                    snack_details[i] = snack_details[i+1] #store i+1 to i
                    snack_details[i+1]= swap #store swap(which is i previously) to i+1
                    swapped = True
                if not swapped:
                    break

    elif option == 3: #high to low
        print("===== ` View mode: Sort by Price (High to Low) ` =====")

        #Arrange descending according to price
        for j in range(0, len(snack_details)-1):
            swapped = False
            for i in range(0, len(snack_details)-1):
                if snack_details[i][2][0] < snack_details[i+1][2][0]:
                    swap = snack_details[i] #store i in swap first
                    snack_details[i] = snack_details[i+1] #store i+1 to i
                    snack_details[i+1]= swap #store swap(which is i previously) to i+1
                    swapped = True
                if not swapped:
                    break

    elif option == 4:#Latest
        print("===== ` View mode: Latest ` =====")
        #reverse
        snack_details = latest(snack_details)

        print("-")
        print("$ ` $ ` $ ` $ ` $ ` $ ` | DRINKS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` ")
        print("-")
        print("ID`\t`Price`\t`Quantity`\t`Drinks")
        print("-")

        num = 0
        for each_snack in snack_details:
            if each_snack[0][0] == "D":
                print(each_snack[0]+\t+\t+each_snack[2]+\t+\t+each_snack[3]+\t\t+each_snack[1])
                num += 1
            elif num == len(snack_details):
                break

        print("-")
        print("$ ` $ ` $ ` $ ` $ ` $ ` | SNACKS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` ")
        print("-")
        print("ID`\t`Price`\t`Quantity`\t`Snacks")
        print("-")

```

```

num = 0
for each_snack in snack_details:
    if each_snack[0][0] == "S":
        print(each_snack[0]+\t"+ "+each_snack[2]+\t"+ "+each_snack[3]+\t\t+each_snack[1])
    num += 1
elif num == len(snack_details):
    break

print("-----")
return

```

Show snack function show all the snacks and drinks available to be sold. All snack details is stored in a snack details list to show the result of snacks and drinks. User can choose the desire view mode to view all the snacks and drinks.

### Function Of Admin Snacks

```

def admin_snacks():
    answer = 1
    while answer == 1:
        show_snacks()
        answer = admin_continue()
    return

```

Admin snacks function shows all the snacks and drinks to admin and give admin option to continue view snacks, go to the admin features menu or log out.

### Function Of Customer Snacks

```

def customer_snacks():
    answer = 1
    while answer == 1:
        show_snacks()
        answer = customer_continue()
    return

```

Customer snacks function shows all the snacks and drinks to customer and give customer option to continue view snacks, go to the customer features menu or log out.

## 5.0 Screenshots of sample input and output with explanation

### Welcome session

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
----- | WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) ! | ----- 

We are glad that you are here ! 

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
Do you have an account? Please choose an option
1: Yes
2: No
3: Exit
Option: 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
Do you want to login now? Please choose an option
1: Login
2: Exit
Option:
```

This is the welcome page when any user enters the Online Super Car Rental Services (**OCRS**) to rent a car and make a booking online. At first, OCSC will show a welcome session to the user and ask user to choose an option. If user already have a registered account, user can choose to login with registered name and password or exit **OCRS** right away if user do not want to continue anymore.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
Do you want to login now? Please choose an option
1: Login
2: Exit
Option: 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
What is your role? Please choose an option
1: Customer
2: Admin
3: Exit
Option: |
```

If user decide to login, online system prompt a message and ask the role of the user. Unless user choose to exit, or else **OCRS** will lead the customer or admin to the specific login session.

If user choose to exit, a log out session will be shown.

## Log out Session.

```
----- | LOG OUT SESSION STARTED | -----
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you wish to log out? Please choose an option
1: Yes
2: No
Option: 1
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
>>> |
```

In the whole OCRS, a log out session will start whenever user choose the exit option. In this log out session, question as shown in the figure above will be asked to ensure the user wants to exit the OCRS. This will ensure user do not exit OCRS right away when the exit option is chosen mistakenly.

```
----- | LOG OUT SESSION STARTED | -----
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you wish to log out? Please choose an option
1: Yes
2: No
Option: 2
=====
Approaching to the welcome page ...
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
----- | WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) ! | -----
We are glad that you are here !

* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you have an account? Please choose an option
1: Yes
2: No
3: Exit
Option: |
```

At last, user will be led to the welcome page once again if user regret the option of logging out OCRS.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
What is your role? Please choose an option
1: Customer
2: Admin
3: Exit
Option: 1
----- | LOGIN SESSION STARTED ! | -----
Kindly fill up the following details:
=====
Please enter name
Name: |
```

A login session would start if user chose to login OCRS with a registered account. Both customer and admin need to login in order to access the features of OCRS.

## Guest session

```
***** | WELCOME TO SUPER CAR RENTAL SERVICES (SCRS) ! | *****

We are glad that you are here !

***** Do you have an account? Please choose an option
1: Yes
2: No
3: Exit
Option: 2
***** Do you want to register an account now?
1: Register now
2: View all cars available for rent
3: Exit
Option: |
```

If a new customer or admin enters OCRS, this unregistered user can choose to register a new account, view cars available for rent and exit OCRS as shown in the figure above.

```
***** Do you want to register an account now?
1: Register now
2: View all cars available for rent
3: Exit
Option: 2
----- | VIEW AVAILABLE CARS | -----

***** Which mode would you like to view available cars for rent?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
```

User without an account can still view the available cars for rent in OCRS. While viewing available cars, four different mode is provided to the guest as shown in figure above.

===== ` View mode: Sort by Price (High to Low) ` =====				
-----				
Car ID	Seats	Price	Name	Description
-----				
MNC	5	730	Mini Cooper	S 2.0 Auto
ALP	7	690	Toyota Alphard	2.4 Auto
CRY	5	370	Toyota Camry	2.0 Auto
ACD	5	370	Honda Accord	2.0 Auto
HRV	5	310	Honda HR-V	1.8 Auto
INV	8	300	Toyota Innova	2.0 Auto
ALT	5	280	Toyota Altis	1.8 Auto (2016)
SGA	8	270	Proton Saga	2.0 Auto
SRN	8	270	Nissan Serena	S-Hybrid
CVC	5	260	Honda Civic	1.8 Auto
TNA	5	249	Nissan Teanna	2.5 Auto
JAZ	5	235	Honda Jazz	1.5 Auto
TVO	5	230	Toyota Vios	1.5 Auto New Model
EXR	7	210	Proton Exora	1.6 Auto
X70	5	180	Proton X70	SUV Auto
PSN	5	176	Proton Persona	1.6 Auto
VIO	5	170	Toyota Vios	1.5 Auto Old Model
MYV	5	170	Perodua Myvi	1.3 Auto VVT
HCT	5	160	Honda City	1.5 Auto
VVA	5	160	Perodua Viva	1.0 Auto
AXA	5	147	Perodua Axia	1.0 Auto
PBZ	5	120	Perodua Bezza	1.3 Auto Advance
AMR	5	100	Nissan Almera	1.5 Auto
-----				
* * * * *				
Do you want to register an account to place booking ?				
1: Continue to view all cars available for rent				
2: Register now				
3: Exit				
Option:				

In the example shown above, guest choose to sort cars by price from high to low. After available cars shown, user must decide to continue view available cars in other mode, register an account to place booking or even exit the OCRS.

## Register session.

```

Do you want to register an account now?
1: Register now
2: View all cars available for rent
3: Exit
Option: 1
* * * * *
What is your role? Please choose an option
1: Customer
2: Admin
3: Exit
Option: 1
----- | REGISTER SESSION STARTED | -----
Kindly fill up the following details:
>< >< >< >< >< >< >< >< ><
Name:

```

Once new user chose to register an account, the register session will start immediately. Customer will be requested to provide information of name, gender, date of birth, phone number, email address and password. All information will be stored in a user text file.

```

----- | REGISTER SESSION STARTED | -----
Kindly fill up the following details:
>< >< >< >< >< >< >< ><
Name:
=====
Please enter your name!
Name is required!
=====
>< >< >< >< >< >< >< ><
Name: Sophie
=====
Name unavailable ! Please choose another name
=====
>< >< >< >< >< >< >< ><
Name: Douglas

```

As show in the figure above, user is unable to proceed when name is not entered, or the name already exist in the text file. This ensures the unique identity of all users in OCRS. Problems of two users with same name will not occur in the OCRS.

```
>< >< >< >< >< >< >< >< >< ><  
Gender (M/F) :  
=====
```

Please provide your gender!

```
=====
```

```
>< >< >< >< >< >< >< >< ><  
Gender (M/F) : s  
=====
```

Provide gender with  
"M" as Male  
"F" as Female.

```
=====
```

```
>< >< >< >< >< >< >< >< >< ><  
Gender (M/F) : M
```

d

Each user needs to provide gender properly with “M” or “F”, else error messages will be given.

```

Date Of Birth:
>< >< >< >< >< >< >< >< ><
Day(DD) :
=====
Please enter according to the format given
=====
>< >< >< >< >< >< >< ><
Day(DD) : 32
=====
Please enter an appropriate day
=====
>< >< >< >< >< >< >< ><
Day(DD) : 31
>< >< >< >< >< >< >< ><
Month(MM) : 0
=====
Please enter an appropriate month
=====
>< >< >< >< >< >< >< ><
Month(MM) : 2
>< >< >< >< >< >< >< ><
Year(YYYY) : 2029
=====
Only 23 to 65 yrs old is allowed!
=====
>< >< >< >< >< >< >< ><
Year(YYYY) : 1994
=====
Invalid Date ! Please re-enter
=====
>< >< >< >< >< >< >< ><
Day(DD) : 11
>< >< >< >< >< >< >< ><
Month(MM) : 11
>< >< >< >< >< >< >< ><
Year(YYYY) : 1981

```

An appropriate date of birth needs to be entered to ensure age of user is within the range of 23 to 65 years old. User below and above the age range will be filtered out. If user enters an invalid date of birth such as 31/2/2029, error message will be shown, and user needs to reenter an appropriate date.

```
>< >< >< >< >< >< >< >< ><
Phone number: (+60)
=====
Please enter your phone number
=====
>< >< >< >< >< >< >< >< ><
Phone number: (+60)12
=====
Please enter your phone number in 9 digit
=====
>< >< >< >< >< >< >< ><
Phone number: (+60)128993847
>< >< >< >< >< >< >< ><
Email Address:aaa
=====
Please enter an appropriate email address
=====
>< >< >< >< >< >< >< ><
Email Address:douglas@gmail
=====
Please enter an appropriate email address
=====
>< >< >< >< >< >< >< ><
Email Address:douglas.mail.com
=====
Please enter an appropriate email address
=====
>< >< >< >< >< >< >< ><
Email Address:douglas@gmail.com
```

Users needs to provide proper phone number and email address. Error messages will be shown when user did not enter proper information.

```
>< >< >< >< >< >< >< ><  
Password: 123  
=====  
Password should be at least 8 character  
=====  
>< >< >< >< >< >< >< ><  
Password: 12345678  
>< >< >< >< >< >< >< ><  
Confirm Password: plsstayathome  
=====  
Password does not match!  
Please re-enter your password:  
>< >< >< >< >< >< >< ><  
Password: stayathome  
>< >< >< >< >< >< >< ><  
Confirm Password: stayathome  
~ ` ~ ` ~ ` ~ Registration successfully completed! ~ ` ~ ` ~ ` ~
```

User needs to create a 8 character password and confirm it. OCRS will let user to re-enter if the “confirm password” do not match the password. After all information is entered, registration process successfully completed as shown above. The slight difference between customer registration and admin registration will be admin registration do not need to provide date of birth.

## Login Session

```
----- |LOGIN SESSION STARTED !|-----
Kindly fill up the following details:
=====
Please enter name
Name:Fara
=====
User not found ! Please re-enter again !
=====
=====
Please enter name
Name:
=====
Please enter your name
=====
Please enter name
Name:Sophie
=====
User exist! Please continue
=====
>< >< >< >< >< >< >< ><
Password:
=====
Please enter your password
=====
>< >< >< >< >< >< >< ><
Password:fara
=====
Password wrong!
Please re-enter your password again
=====
>< >< >< >< >< >< >< ><
Password:sophietky
~ ` ~ ` ~ ` ~ ` ~ Succesfully Logged in! Welcome back! ~ ` ~ ` ~ ` ~ ` ~
```

In the login session for customer, customer will be asked to enter registered name and correct password. Error messages will be shown when user did not enter their name. Definitely, the user will not be allowed to proceed when the entered name is not found in the database which is the user text file.

## Customer Features

```

~ ` ~ ` ~ ` ~ ` ~ Successfully Logged in! Welcome back! ~ ` ~ ` ~ ` ~ ` ~
=====
Approaching Features Menu
----- | FEATURES MENU | -----
Welcome to the Features Menu !

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
What would you like to do?
1: Modify Personal Details
2: View Personal Rental History
3: View Details Of Available Car
4: Place A Booking
5: Exit

=====
Please choose an option
Option:|

```

The features menu as shown above will be shown to the customer once the customer has successfully login. Customer will be able to access all these features including modify personal details, check rental history, view available cars, and place a booking online.

```

Please choose an option
Option:1
----- | MODIFY PERSONAL DETAILS SESSION STARTED | -----
Kindly fill up the following details:
>< >< >< >< >< >< >< ><
Enter your name to modify personal details:Douglas
===== ` Personal Details ` =====
Name: Douglas
1: Gender: M
2: Date Of Birthday: 1981-11-11
3: Phone number: (+60) 128993847
4: Email Address: douglas@gmail.com
5: Password: stayathome
=====
>< >< >< >< >< >< >< ><
Enter field number to be modify: 4
Existing information in field is: douglas@gmail.com
>< >< >< >< >< >< >< ><
Enter new email address:dg
=====
Please enter an appropriate email address
=====
>< >< >< >< >< >< >< ><
Enter new email address:dg.mail.apu.edu.my
=====
Please enter an appropriate email address
=====
>< >< >< >< >< >< >< ><
Enter new email address:dg@mail.apu.edu.my
~ ` ~ ` ~ ` ~ ` ~ Personal details successfully updated ! ~ ` ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Features Menu
3: Exit
Option: |

```

Figure above shows customer enters the modify personal details session by choosing option from the features menu. As shown in example above, the customer decides to modify email. After done modifying personal details, customer can modify once again if the customer is not satisfied. Beside that, customer can go back to the features menu or exit OCRS.

```
-----+
Option:3
```

```
----- | VIEW AVAILABLE CARS | -----
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Which mode would you like to view available cars for rent?

- 1: All
- 2: Price (Low-High)
- 3: Price (High-Low)
- 4: Latest

Option: 2

```
===== ` View mode: Sort by Price (Low to High) ` =====
```

```
$ ` $ ` $ ` $ ` $ ` $ ` | CARS | ` $ ` $ ` $ ` $ ` $ ` $
```

Car ID	Seats	Price	Name	Description
AMR	5	100	Nissan Almera	1.5 Auto
PBZ	5	120	Perodua Bezza	1.3 Auto Advance
AXA	5	147	Perodua Axia	1.0 Auto
HCT	5	160	Honda City	1.5 Auto
VVA	5	160	Perodua Viva	1.0 Auto
VIO	5	170	Toyota Vios	1.5 Auto Old Model
MYV	5	170	Perodua Myvi	1.3 Auto VVT
PSN	5	176	Proton Persona	1.6 Auto
X70	5	180	Proton X70	SUV Auto
EXR	7	210	Proton Exora	1.6 Auto
TVO	5	230	Toyota Vios	1.5 Auto New Model
JAZ	5	235	Honda Jazz	1.5 Auto
TNA	5	249	Nissan Teanna	2.5 Auto
CVC	5	260	Honda Civic	1.8 Auto
SGA	8	270	Proton Saga	2.0 Auto
SRN	8	270	Nissan Serena	S-Hybrid
ALT	5	280	Toyota Altis	1.8 Auto (2016)
INV	8	300	Toyota Innova	2.0 Auto
HRV	5	310	Honda HR-V	1.8 Auto
CRY	5	370	Toyota Camry	2.0 Auto
ACD	5	370	Honda Accord	2.0 Auto
ALP	7	690	Toyota Alphard	2.4 Auto
MNC	5	730	Mini Cooper	S 2.0 Auto

```
* * * * * * * * * * * * * * * * * * * * * * *
```

Do you want to continue? Please choose an option

- 1: Yes
- 2: View Features Menu
- 3: Exit

Option:

Customer can also view all the available cars for rent through the features menu. Available cars will be shown with the view mode of customers desired.

**Booking session**

```
Option:4
----- | BOOKING SESSION STARTED |
Kindly fill up the following details:
=====
Please enter name
Name:D
=====
User not found ! Please re-enter again !
=====
Please enter name
Name:
=====
Please enter your name
=====
Please enter name
Name:Douglas
=====
User exist! Please continue
=====
----- | VIEW AVAILABLE CARS | -----

***** * * * * *
Which mode would you like to view available cars for rent?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 5
=====
Only option 1 to option 4 is available!
=====
```

Customer needs to enter their registered name to place booking. After that customer will be able to view available cars and choose a car to rent.

` View Mode: Show Latest Cars `				
\$ ` \$ ` \$ ` \$ ` \$ ` \$ `   CARS   ` \$ ` \$ ` \$ ` \$ ` \$ ` \$				
Car ID	Seats	Price	Name	Description
X70	5	180	Proton X70	SUV Auto
AMR	5	100	Nissan Almera	1.5 Auto
SRN	8	270	Nissan Serena	S-Hybrid
HRV	5	310	Honda HR-V	1.8 Auto
VVA	5	160	Perodua Viva	1.0 Auto
EXR	7	210	Proton Exora	1.6 Auto
HCT	5	160	Honda City	1.5 Auto
SGA	8	270	Proton Saga	2.0 Auto
TNA	5	249	Nissan Teanna	2.5 Auto
JAZ	5	235	Honda Jazz	1.5 Auto
PSN	5	176	Proton Persona	1.6 Auto
MNC	5	730	Mini Cooper	S 2.0 Auto
ALP	7	690	Toyota Alphard	2.4 Auto
AXA	5	147	Perodua Axia	1.0 Auto
MYV	5	170	Perodua Myvi	1.3 Auto VVT
TVO	5	230	Toyota Vios	1.5 Auto New Model
VIO	5	170	Toyota Vios	1.5 Auto Old Model
PBZ	5	120	Perodua Bezza	1.3 Auto Advance
ACD	5	370	Honda Accord	2.0 Auto
CRY	5	370	Toyota Camry	2.0 Auto
INV	8	300	Toyota Innova	2.0 Auto
ALT	5	280	Toyota Altis	1.8 Auto (2016)
CVC	5	260	Honda Civic	1.8 Auto

Customer can choose to view available cars in the mode of “Latest” which is newly added by the admin.

```

~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
Which car would you like to book?
>< >< >< >< >< >< >< >< >
Enter Car ID to book:ABC
=====
Car is not found !
=====
>< >< >< >< >< >< >< >
Enter Car ID to book:
=====
Car is not found !
=====
>< >< >< >< >< >< >< >
Enter Car ID to book:X70
===== ' Car Details ' =====
Car ID: X70
1: Name: Proton X70
2: Description: SUV Auto
3: Number Of Seats: 5 seater
4: Daily Price: RM 180
5: Quantity: 78
=====
~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
How many days would you like to book ?
Day:
=====
Only numbers allowed!
=====
Day: 100
=====
Maximum 28 days are allowed!
=====
Day: 15

```

Customer will need to enter the correct Car ID of an available car and the car details will be shown. Furthermore, customers can only rent a car for maximum 28 days. If customer enters more than that, OCRS will show the error message to customer.

```

~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
When do you want to start your car rental ?
Please provide date and time
>< >< >< >< >< >< >< >< >< >
Day (DD) : 31
>< >< >< >< >< >< >< >< >
Month (MM) : 2
>< >< >< >< >< >< >< >< >
Year (YYYY) : 2050
=====
You can only book a car 3 years before
=====
>< >< >< >< >< >< >< >< >
Year (YYYY) : 2023
>< >< >< >< >< >< >< >
Hour (HH) : 24
=====
Please enter an appropriate time in 24-hr format
Only 8:00 to 22:59 is allowed
=====
>< >< >< >< >< >< >< >< >
Hour (HH) : 10
>< >< >< >< >< >< >< >< >
Minutes (MM) : 61
=====
Please enter an appropriate time
=====
>< >< >< >< >< >< >< >< >
Minutes (MM) : 10
=====
Invalid Date or Time ! Please re-enter
=====
>< >< >< >< >< >< >< >< >
Day (DD) : 10
>< >< >< >< >< >< >< >< >
Month (MM) : 1
>< >< >< >< >< >< >< >< >
Year (YYYY) : 2021
>< >< >< >< >< >< >< >< >
Hour (HH) : 11
>< >< >< >< >< >< >< >< >
Minutes (MM) : 11

```

Customer will need to provide appropriate car rental date and time. ORCS will show error message to customer when conditions do not met. For instance, customer can only book a car

3 years before and the car can only be collected within the time range. Undeniably, date and time will need to be re-enter when it is invalid.

-----   VIEW SNACKS & DRINKS   -----			
* *			
Which mode would you like to view snack and drinks?			
1: All 2: Price(Low-High) 3: Price(High-Low) 4: Latest			
Option: 1			
===== ` View mode: All ` =====			
\$ ` \$ ` \$ ` \$ ` \$ ` \$ `   DRINKS   ` \$ ` \$ ` \$ ` \$ ` \$			
ID	Price	Quantity	Drinks
D01	1	42	Coca Cola
D02	1	50	Sprite
D03	1	50	100 plus
D04	1	50	Sprite
D05	1	48	Kickapoo
D06	1	50	Pepsi
D07	2	43	Yeo's Chrysanthemum Tea
D08	2	50	Yeo's Grass Jelly Cincau
D09	3	49	Nescafe Coffee
D10	3	50	Dutch Lady Full Cream Milk
D11	3	10	Ribena Cheerpak
D12	4	83	Minute Maid Pulpy Orange Juice
\$ ` \$ ` \$ ` \$ ` \$ ` \$ `   SNACKS   ` \$ ` \$ ` \$ ` \$ ` \$			
ID	Price	Quantity	Snacks
S01	3	20	Super Ring
S02	5	50	Double Decker Prawn Chicken Cheese
S03	2	48	Twisties
S04	4	50	Mammee
S05	2	41	Mini Chips More
S06	8	44	Lays Black Pepper Potato Chips
S07	5	43	Choki Choki
S08	1	50	Tam Tam
S09	3	28	J&J Potato Chips
S10	2	30	Bika Chicken
S11	3	47	Wang Wang Biscuits
S12	4	20	Burberry

Customer can choose to order or not order snacks and drinks while placing booking.

```

~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
Which snack or drink would you like to order?
>< >< >< >< >< >< >< >< ><
Enter Snack ID to order:D12
===== ` Snacks Details ` =====
Snack ID: D12
1: Name: Minute Maid Pulpy Orange Juice
2: Price: 4
3: Quantity: 80
=====
~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
How many would you like to order ?
>< >< >< >< >< >< >< >< ><
Quantity: 0
=====
Minimum one quantity
=====
>< >< >< >< >< >< >< >< ><
Quantity: 2
~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ ` ~ 
When is your car rental start date and time ?
>< >< >< >< >< >< >< >< ><
Day(DD): 10
>< >< >< >< >< >< >< ><
Month(MM): 1
>< >< >< >< >< >< >< ><
Year(YYYY): 2021
>< >< >< >< >< >< >< ><
Hour(HH): 11
>< >< >< >< >< >< >< ><
Minutes(MM): 11
~ ` ~ ` ~ ` ~ ` ~ Snacks ordered successfully! ~ ` ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 
Do you want to continue? Please choose an option
1: Yes
2: No
Option:

```

Customer can choose snacks and drinks from the list and select the quantity of the snacks or drinks. Customer will provide the car rental start date and time and decide to order another snacks and or not.

## Payment session

```
----- | PAYMENT SESSION STARTED | -----  
{ The total amount to be paid is RM 2700 }  
Kindly fill up the following details:  
=====Please choose a payment method:  
1 -> Pay In Cash  
2 -> Credit Card  
3 -> Debit Card  
4 -> Online Banking  
>< >< >< >< >< >< >< ><  
Payment Method: 4  
~ ` ~ ` ~ ` ~ Booking placed successfully! ~ ` ~ ` ~ ` ~  
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  
Do you want to continue? Please choose an option  
1: Yes  
2: View Features Menu  
3: Exit  
Option: |
```

After customer done booking car and ordering snacks, payment is required to confirm the booking. There are several payment methods provided for the customer.

```

option:2
----- | PERSONAL RENTAL HISTORY SESSION STARTED | -----
Kindly fill up the following details:
=====
Please enter name
Name:Douglas
=====
User exist! Please continue
=====

$ ` $ ` $ ` $ ` $ ` $ ` $ ` | BOOKINGS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ 
-----
Car ID Days Rental Date Amount (RM) Payment Date
-----
X70 15 2021-01-10 11:11:00 2708 2021-06-06

$ ` $ ` $ ` $ ` $ ` $ ` | CARS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ 
-----
Car ID Seats Price Name Description
-----
X70 5 180 Proton X70 SUV Auto

$ ` $ ` $ ` $ ` $ ` | SNACKS & DRINKS | ` $ ` $ ` $ ` $ ` $ 
-----
ID Price Quantity Item
-----
D12 4 3 Minute Maid Pulpy Orange Juice
D12 4 2 Minute Maid Pulpy Orange Juice
-----

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Features Menu
3: Exit
Option: |

```

Customer can view the personal rental history after booking is placed. Admin can still choose to continue, return to the customer features menu or log out.

## Admin Features

```

~ ` ~ ` ~ ` ~ ` ~ Successfully Logged in! Welcome back! ~ ` ~ ` ~ ` ~ ` ~
=====
Approaching Admin Features Menu
----- | ADMIN FEATURES MENU | -----
Welcome to the Admin Features Menu !

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
What would you like to do?

1. Modify Personal Details
2. Modify Car Details
3. Modify Snacks & Drinks Details
4. Add Cars To Be Rented Out
5. Add Snacks & Drinks To Be Sold
6. View Cars Rented Out
7. View Cars Available For Rent
8. View Customer Bookings
9. View Customer Payment
10. View Available Snacks & Drinks
11. Search Customer Booking
12. Search Customer Payment
13. Return Rented Car
14. Exit
=====
Please choose an option
Option:|

```

This is list of features available for the admin when the admin has successfully logged in.  
 Admin have the option to access all these features

```
option:1
----- | MODIFY PERSONAL DETAILS SESSION STARTED | -----
Kindly fill up the following details:
>< >< >< >< >< >< >< >< >
Enter your name to modify personal details:kaiyii
===== ` Personal Details ` =====
Name: kaiyii
1: Gender: F
2: Phone number: (+60) 192039485
3: Email Address: kaiyii@gmail.com
4: Password: tkytkytky
=====
>< >< >< >< >< >< >< >< >
Enter field number to be modify: 2
Existing information in field is: 192039485
>< >< >< >< >< >< >< >
Enter new phone number (+60):192839485
~ ` ~ ` ~ ` ~ ` ~ Personal details successfully updated ! ~ ` ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option: |
```

Admin can modify personal details through the admin features menu. After modifying personal details, admin can choose to continue modifying personal details, go back to the admin features menu or exit OCRS.

```

option:2
----- | MODIFY CAR SESSION STARTED | -----
----- | VIEW AVAILABLE CARS | -----
* * * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view available cars for rent?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 4
===== ` View Mode: Show Latest Cars ` =====
-----
$ ` $ ` $ ` $ ` $ ` $ ` | CARS | ` $ ` $ ` $ ` $ ` $ ` $
-----
Car ID Seats Price Name Description
-----
X70      5     180   Proton X70   SUV Auto
AMR      5     100   Nissan Almera 1.5 Auto
SRN      8     270   Nissan Serena S-Hybrid
HRV      5     310   Honda HR-V   1.8 Auto
VVA      5     160   Perodua Viva  1.0 Auto
EXR      7     210   Proton Exora  1.6 Auto
HCT      5     160   Honda City   1.5 Auto
SGA      8     270   Proton Saga   2.0 Auto
TNA      5     249   Nissan Teanna 2.5 Auto
JAZ      5     235   Honda Jazz   1.5 Auto
PSN      5     176   Proton Persona 1.6 Auto
MNC      5     730   Mini Cooper  S 2.0 Auto
ALP      7     690   Toyota Alphard 2.4 Auto
AXA      5     147   Perodua Axia  1.0 Auto
MYV      5     170   Perodua Myvi  1.3 Auto VVT
TVO      5     230   Toyota Vios   1.5 Auto New Model
VIO      5     170   Toyota Vios   1.5 Auto Old Model
PBZ      5     120   Perodua Bezza 1.3 Auto Advance
ACD      5     370   Honda Accord 2.0 Auto
CRY      5     370   Toyota Camry  2.0 Auto
INV      8     300   Toyota Innova 2.0 Auto
ALT      5     280   Toyota Altis  1.8 Auto (2016)
CVC      5     260   Honda Civic  1.8 Auto
-----
```

When admin enters modify car details session, list of available car will be shown based on the view mode admin desire. Admin can still choose to continue, return to the admin features menu or log out.

```
>< >< >< >< >< >< >< ><
Enter Car ID to modify car details:MNC
===== ` Car Details ` =====
Car ID: MNC
1: Name: Mini Cooper
2: Description: S 2.0 Auto
3: Number Of Seats: 5 seater
4: Daily Price: RM 730
5: Quantity: 40
=====
>< >< >< >< >< >< >< ><
Enter field number to be modify: 4
Existing information in field is: 730
>< >< >< >< >< >< ><
Enter new daily price: RM0
=====
Minimum RM100
=====
>< >< >< >< >< >< >< ><
Enter new daily price: RM750
~ ` ~ ` ~ ` ~ ` ~ Car details successfully updated !!! ~ ` ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option: |
```

As shown in the figure, admin can modify all the car details except the Car ID. Admin can still choose to continue, return to the admin features menu or log out.

```

Please choose an option
Option:3
----- | MODIFY SNACKS & DRINKS SESSION STARTED | -----
----- | VIEW SNACKS & DRINKS | -----
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view snack and drinks?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 3
===== ` View mode: Sort by Price (High to Low) ` =====
-----
$ ` $ ` $ ` $ ` $ ` $ ` $ ` | DRINKS | ` $ ` $ ` $ ` $ ` $ ` $ 
-----
ID      Price     Quantity      Drinks
-----
D12      4          78          Minute Maid Pulpy Orange Juice
D09      3          49          Nescafe Coffee
D10      3          50          Dutch Lady Full Cream Milk
D11      3          10          Ribena Cheerpack
D07      2          43          Yeo's Chrysanthemum Tea
D08      2          50          Yeo's Grass Jelly Cincau
D01      1          42          Coca Cola
D02      1          50          Sprite
D03      1          50          100 plus
D04      1          50          Sprite
D05      1          48          Kickapoo
D06      1          50          Pepsi
-----
$ ` $ ` $ ` $ ` $ ` $ ` $ ` | SNACKS | ` $ ` $ ` $ ` $ ` $ ` $ 
-----
ID      Price     Quantity      Snacks
-----
S06      8          44          Lays Black Pepper Potato Chips
S14      7          86          Tiger Biscuits
S07      5          43          Choki Choki
S02      5          50          Double Decker Prawn Chicken Cheese
S04      4          50          Mammee
S12      4          20          Burberry
S01      3          20          Super Ring
S09      3          28          J&J Potato Chips
S11      3          47          Wang Wang Biscuits
S03      2          48          Twisties
S05      2          41          Mini Chips More
S10      2          30          Bika Chicken

```

```
>< >< >< >< >< >< >< >< >
Enter Snack ID to modify details:D07
===== ` Snacks Details ` =====
Snack ID: D07
1: Name: Yeo's Chrysanthemum Tea
2: Price: 2
3: Quantity: 43
=====
>< >< >< >< >< >< >< >< >
Enter field number to be modify: 3
Existing information in field is: 43
>< >< >< >< >< >< >< >
Enter new quantity:40
~ ` ~ ` ~ ` ~ Snacks & Drinks details successfully updated ! ~ ` ~ ` ~ ` ~
```

Admin can modify all the snacks and drinks except the Snack ID.

```
Option:4
----- | INSERT CAR SESSION STARTED | -----
Kindly fill up the following details:
>< >< >< >< >< >< >< >
Car ID: VLF
>< >< >< >< >< >< >< >
Name: Toyota Vellfire
>< >< >< >< >< >< >< >
Description: 2.5 Auto
>< >< >< >< >< >< >< >
Number Of Seats:19
=====
Please enter appropriate number of seats
=====
>< >< >< >< >< >< >< >< >
Number Of Seats:7
>< >< >< >< >< >< >< >
Daily Price: RM0
=====
Minimum RM100
=====
>< >< >< >< >< >< >< >< >
Daily Price: RM360
>< >< >< >< >< >< >< >
Quantity: 100
Car information successfully updated!
* * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:
```

Admin has the features to add cars to be rented out to the OCRS and allow customer to place booking. Admin can still choose to continue, return to the admin features menu or log out.

```
Please choose an option
Option:5
----- | INSERT SNACKS & DRINKS SESSION STARTED | -----
Kindly fill up the following details:
Snack ID:D11
=====
Snack ID already exist!
=====
Snack ID:D13
>< >< >< >< >< >< >< >< ><
Snack: Mountain Dew
>< >< >< >< >< >< >< ><
Price: RM3
>< >< >< >< >< >< >< ><
Quantity: 20
~ ` ~ ` ~ ` ~ Snacks information successfully updated! ~ ` ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:
```

Admin has the features to add snacks and drinks to the OCRS and allow customer to order them. Admin can still choose to continue, return to the admin features menu or log out.

```
Option:6
----- | VIEW RENTED CARS | -----
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view rented cars?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 4
===== ` View Mode: Show Latest Cars ` =====
$ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ 
-----
Car ID  Seats  Price  Name          Description  Quantity Available  Quantity Rented
-----
X70      5       180   Proton X70    SUV Auto        77                  3
AMR      5       100   Nissan Almera  1.5 Auto       39                  1
HRV      5       310   Honda HR-V     1.8 Auto       28                  2
ALP      7       690   Toyota Alphard  2.4 Auto       69                  1
MYV      5       170   Perodua Myvi    1.3 Auto VVT   99                  1
ACD      5       370   Honda Accord   2.0 Auto       74                  1
-----
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:
```

Admin is able to view all the rented cars by choosing the desired view mode. Details of quantity available and quantity rented is provided. Admin can still choose to continue, return to the admin features menu or log out.

```
option:7
----- | VIEW AVAILABLE CARS | -----
***** * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view available cars for rent?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 3
===== ` View mode: Sort by Price (High to Low) ` =====
$ ` $ ` $ ` $ ` $ ` $ ` | CARS | ` $ ` $ ` $ ` $ ` $ ` $
-----
Car ID   Seats   Price    Name           Description
-----
MNC      5       750     Mini Cooper   S 2.0 Auto
ALP      7       690     Toyota Alphard 2.4 Auto
CRY      5       370     Toyota Camry   2.0 Auto
ACD      5       370     Honda Accord   2.0 Auto
VLF      7       360     Toyota Vellfire 2.5 Auto
HRV      5       310     Honda HR-V    1.8 Auto
INV      8       300     Toyota Innova  2.0 Auto
ALT      5       280     Toyota Altis   1.8 Auto (2016)
SGA      8       270     Proton Saga   2.0 Auto
SRN      8       270     Nissan Serena  S-Hybrid
CVC      5       260     Honda Civic   1.8 Auto
TNA      5       249     Nissan Teanna  2.5 Auto
JAZ      5       235     Honda Jazz    1.5 Auto
TVO      5       230     Toyota Vios   1.5 Auto New Model
EXR      7       210     Proton Exora   1.6 Auto
X70      5       180     Proton X70    SUV Auto
PSN      5       176     Proton Persona 1.6 Auto
VIO      5       170     Toyota Vios   1.5 Auto Old Model
MYV      5       170     Perodua Myvi   1.3 Auto VVT
HCT      5       160     Honda City   1.5 Auto
VVA      5       160     Perodua Viva   1.0 Auto
AXA      5       147     Perodua Axia   1.0 Auto
PBZ      5       120     Perodua Bezza  1.3 Auto Advance
AMR      5       100     Nissan Almera  1.5 Auto
-----
***** * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:
```

Admin is able to view all the available cars for rent by choosing the desired view mode. All details of cars were shown. Admin can still choose to continue, return to the admin features menu or log out.

Option:8
-----   BOOKINGS   -----
* * * * *
Which mode would you like to view bookings?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 2
===== ` View mode: Sort by Total Amount (Low to High) ` =====
\$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$   BOOKINGS   ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$
Car ID Days Rental Start Date Payment Status Total Amount(RM) Name
HCT 2 2021-07-22 10:11:00 2021-06-04 330 Shaila
HRV 1 2021-01-31 11:11:00 2021-06-04 358 Sofia
AMR 4 2024-04-04 14:14:00 2021-06-05 410 Lan
AMR 5 2024-04-04 14:14:00 2021-06-04 505 Sofia
X70 3 2022-10-22 10:10:00 2021-06-04 540 Oufishal
X70 3 2021-11-11 11:11:00 2021-06-04 560 Sofia
AXA 5 2021-07-01 20:20:00 2021-06-04 737 Sophie
X70 6 2024-04-04 14:04:00 2021-06-04 1089 Jeremy
CRY 3 2021-05-31 22:00:00 2021-06-04 1113 Sophie
X70 9 2022-03-04 14:20:00 2021-06-04 1623 Casty
HRV 5 2023-10-22 22:22:00 2021-06-04 1696 Oufishal
X70 15 2021-01-10 11:11:00 2021-06-06 2708 Douglas
ALP 4 2022-11-16 20:02:00 2021-06-04 2809 Sorila
VVA 19 2021-06-04 10:10:00 2021-06-04 3044 Lincoln Teo
\$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$   SNACKS & DRINKS   ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$ ` \$
ID Quantity Rental Start Date Name
S05 1 2021-06-04 10:10:00 Lincoln Teo
D09 1 2022-03-04 14:20:00 Casty
D02 2 2021-07-01 20:20:00 Sophie
D01 2 2021-06-04 10:10:00 Lincoln Teo
D12 2 2021-01-10 11:11:00 Douglas
S13 3 2021-05-31 22:00:00 Sophie
S11 3 2024-04-04 14:04:00 Johny
D12 3 2021-01-10 11:11:00 Douglas
S03 5 2021-07-22 10:11:00 Shaila

Admin has the feature to view all the customer bookings by choosing the desired view mode which the figure above choose to sort result by the total amount paid of bookings. All details of bookings and snack and drinks orders were shown.

```

Option:9
----- | PAYMENT | -----

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view payments?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 4
===== ` View mode: Show Latest Payment ` =====
----- $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` | PAYMENT | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` -----


Payment Status  Payment Date      Payment Amount(RM)   Payment Method     Name
-----



Paid           2021-06-06        2708             Online Banking    Douglas
Paid           2021-06-05        410              Pay In Cash      Lan
Paid           2021-06-04        540              Debit Card       Oufishal
Paid           2021-06-04        1696             Debit Card       Oufishal
Paid           2021-06-04        505              Pay In Cash      Sofia
Paid           2021-06-04        560              Credit Card      Sofia
Paid           2021-06-04        358              Credit Card      Sofia
Paid           2021-06-04        1089             Online Banking   Jeremy
Paid           2021-06-04        2809             Credit Card      Sorila
Paid           2021-06-04        1623             Debit Card       Casty
Paid           2021-06-04        3044             Pay In Cash      Lincoln Teo
Paid           2021-06-04        330              Online Banking   Shaila
Paid           2021-06-04        737              Credit Card      Sophie
Paid           2021-06-04        1113             Debit Card       Sophie
-----



* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:

```

As shown in figure above, Admin is able to see the payments of customer based on the payment date. Details of payment is shown properly. Admin can still choose to continue, return to the admin features menu or log out.

```

option:10
----- | VIEW SNACKS & DRINKS | -----
* * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view snack and drinks?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 1
===== ` View mode: All ` =====
-----
$ ` $ ` $ ` $ ` $ ` $ ` $ ` | DRINKS | ` $ ` $ ` $ ` $ ` $ ` $ 
-----
ID      Price     Quantity      Drinks
-----
D01      1          42          Coca Cola
D02      1          50          Sprite
D03      1          50          100 plus
D04      1          50          Sprite
D05      1          48          Kickapoo
D06      1          50          Pepsi
D07      2          40          Yeo's Chrysanthemum Tea
D08      2          50          Yeo's Grass Jelly Cincau
D09      3          49          Nescafe Coffee
D10      3          50          Dutch Lady Full Cream Milk
D11      3          10          Ribena Cheerpak
D12      4          78          Minute Maid Pulpy Orange Juice
D13      3          20          Mountain Dew
-----
$ ` $ ` $ ` $ ` $ ` $ ` $ ` | SNACKS | ` $ ` $ ` $ ` $ ` $ ` $ 
-----
ID      Price     Quantity      Snacks
-----
S01      3          20          Super Ring
S02      5          50          Double Decker Prawn Chicken Cheese
S03      2          48          Twisties
S04      4          50          Mam mee
S05      2          41          Mini Chips More
S06      8          44          Lays Black Pepper Potato Chips
S07      5          43          Choki Choki
S08      1          50          Tam Tam
S09      3          28          J&J Potato Chips
S10      2          30          Bika Chicken
S11      3          47          Wang Wang Biscuits

```

Admin has the feature to view available snack and drinks. Details of price and quantity can be shown clearly. Admin can still choose to continue, return to the admin features menu or log out.

```
Option:11
----- | SEARCH BOOKINGS |
Kindly fill up the following details:
=====
Please enter name
Name:Douglas
=====
User exist! Please continue
=====

$ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` | BOOKINGS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ 
-----
Car ID Days Rental Start Date Payment Status Payment Date Payment Method Amount(RM)
-----
X70 15 2021-01-10 11:11:00 Paid 2021-06-06 Online Banking 2708
-----
* * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option: |
```

OCRS provide the search feature for admin to search bookings of customers. The important information of customer bookings is shown by person. Admin can use this feature to search the bookings of a specific customer.

```
Option:12
----- | SEARCH PAYMENT |
Kindly fill up the following details:
* * * * *
How would you like to search? Please choose an option
1: Search by Name
2: Search by Date
Option: 2
>< >< >< >< >< >< >< >< 
Enter a date to search booking(YYYY-MM-DD):2021-06-6
=====
Date not found ! Please follow the format
YYYY-MM-DD
=====
>< >< >< >< >< >< >< 
Enter a date to search booking(YYYY-MM-DD):2021-06-06
=====
Date exist!
=====

$ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` | PAYMENT | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` 
-----
Status Date Method Amount(RM) Name
-----
Paid 2021-06-06 Online Banking 2708 Douglas
-----
* * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option: |
```

In addition, admin can also search payment of customer by name and date. The figure above shows the sample of searching payment by date. Admin will need to follow the format given and provide a date exist in the booking text file.

```

Please choose an option
Option:13
----- | RETURN RENTED CAR SESSION STARTED | -----
Kindly fill up the following details:
----- | VIEW RENTED CARS | -----
* * * * * * * * * * * * * * * * * * * * *
Which mode would you like to view rented cars?
1: All
2: Price(Low-High)
3: Price(High-Low)
4: Latest
Option: 1
===== ` View mode: All ` =====
-----
$ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` | CARS | ` $ ` $ ` $ ` $ ` $ ` $ ` $ ` $ 
-----
Car ID Seats Price Name Description Quantity Available Quantity Rented
-----
ACD 5 370 Honda Accord 2.0 Auto 74 1
MYV 5 170 Perodua Myvi 1.3 Auto VVT 99 1
ALP 7 690 Toyota Alphard 2.4 Auto 69 1
HRV 5 310 Honda HR-V 1.8 Auto 28 2
AMR 5 100 Nissan Almera 1.5 Auto 39 1
X70 5 180 Proton X70 SUV Auto 77 3
-----
>< >< >< >< >< >< >< ><
Enter Car ID to return a rented car:X70
Car ID: X70
1: Name: Proton X70
2: Description: SUV Auto
3: Number Of Seats: 5 seater
4: Daily Price: RM 180
5: Quantity: 77
~ ` ~ ` ~ ` ~ ` ~ Rented Car Return successfully! ~ ` ~ ` ~ ` ~
* * * * * * * * * * * * * * * * * * * * *
Do you want to continue? Please choose an option
1: Yes
2: View Admin Features Menu
3: Exit
Option:

```

Admin can return rented car after the car is returned by the customer while car rental has end.

Admin can still choose to continue, return to the admin features menu or log out.

## 6.0 Conclusion

By and large, Online Car Rental Service is designed for the SUPER CAR RENTAL SERVICES (SCRS). It provides customer convenience by allowing booking rental cars online. This system aims to save time of the customer as customer no longer need to book rental cars physically. This system suits SUPER CAR RENTAL SERVICES (SCRS) well as this business is growing fast. The business needs this system to allow customer book rental cars easily without going physically.

Through this module, I have learnt the basic of python including functions, variable, list, try except block, while loop, for loop, manipulate txt file, if statement and many more. By doing this assignment, I have learnt how to use the basic of python to build an Online Car Rental System. While researching Online Car Rental System I have also gained some knowledge about online car rental which would be useful if I plan to rent a car in future.

## 7.0 Workload Matrix

<b>Teo Kai Yii (TP058618)</b>	<b>Tang Kar Lok (TP060733)</b>
Python coding Introduction and assumptions Pseudocode and Flowchart Program source code with explanation Additional features source code with explanation Screenshot of sample input/output with explanation Conclusion	Part of Pseudocode and Flowchart

## 8.0 References

- Auto Europe, n.d. *Driving Information Malaysia*. [Online]  
Available at: <https://www.autoeurope.co.uk/driving-information-malaysia/>  
[Accessed 6 June 2021].
- Partridge, P., 2018. *How long can I rent a car for?*. [Online]  
Available at: <https://www.rhinocarhire.com/Car-Hire-Blog/October-2018/How-long-can-I-rent-a-car-for.aspx>  
[Accessed 6 June 2021].