Group Assignment of Online Car Rental System

TECHNOLOGY PARK MALAYSIA

AAPP010-4-2-PWP

PROGRAMMING WITH PYTHON

UCDF2005-ICT(SE)/(1)ICT(SE)/BIT/(1)BIT/

ICT/(1)ICT/ICT(ITR)/(1)ICT(ITR)/ICT(DI)/ (1)ICT(DI)

HAND OUT DATE: **22TH APRIL 2021**

HAND IN DATE: **18TH JUNE 2021**

WEIGHTAGE: **100%**

Lecturer: Mr. Usman Hasim
Tutor: Mr. Liew Yee Jing

Group Members
YIP KAR FAI (TP060711) | NG LI SHENG (TP060612)

Table of Contents

1.0 Introduction and Assumption

With the development of science and technologies, super car rental services are one of the fast-growing digital services that have become a popular transportation option among people, especially in Malaysia. Therefore, the main objective of this assignment is to provide a fully developed software for SCRS to enhance their online car rental system in both client and administrator side. Objectives of the project included customer booking, transactions activities and administration. Meanwhile, a well-designed of software architecture and interface designed are required for client and administrator. These reports will be fully covering the system architecture, implementation, and user manual.

With the advancement of technology and innovation, fast expanding and in great demand digital businesses such as online vehicle rental has emerged. Meanwhile, major online car rental digital businesses such as SOCARS and Go Car have established themselves as well-known vehicle rental services in Malaysia. Computer-assisted services have been integrated into nearly every aspect of our everyday lives and the environment in which we live.

Based on the data stated above, we can declare that we believe that the use of online vehicle rental services will continue to rise and become one of the top commercial and digital services, as nearly everyone is reliant on them.

2.0 Pseudocode

1. Main Function

```
FUNCTION main ()
        defaultUsername = “admin”
        defaultPassword = “admin”
        checkLogin = True
        DOWHILE (checkLogin)
                TRY
                        Display “-” *106
                        Display “Welcome to Car Rental Management System\n”.center(100)”
                        Display “Please select an option:”
                        Display “1. Registered Customer”
                        Display “2. Unregistered Customer”
                        Display “3. Admin”
                        Display “4. Guest”
                        Display “5. Account Recovery”
                        Display “6. Support Us”
                        Display “7. Exit”
                        Display “Please enter your option:”
                        Read userDecision
                        IF (userDecision == 1) THEN
                                Display “-” *106
                                Display “Welcome to Car Rental Management System
                                (Registred Customer)\n”.center(100))
                                Access to userLogin function
                                checkLogin = False
                        ELIF (userDecision == 2) THEN
                                Display “-” *106
                                Display “Welcome to Car Rental Management System
                                (Unregistred Customer)\n”.center(100))
                                Display “Please fill the form below:”
                                Access to userRegister function
                        ELIF (userDecision == 3) THEN
                                Display “-” * 106
                                Display “Welcome to Car Rental Management System
                                (Administrator)\n”.center(100))
                                Display “Please enter Username:”
                                Read str(username)
                                Display “Please enter Password:”
                                Read password
```

```
                        IF (username == defaultUsername
                        and password == defaultPassword) THEN
                                Display “Loging into Admin .....”
                                Display “Login Successfully! You are now login as
                                Administrator Mode
                                Access to adminMenu function
                        ELSE
                                Display “Invalid Username and Password”
                                Continue access to main function
                        ENDIF
                ELIF (userDecision == 4) THEN
                        Display “Welcome to Car Rental Management System (Guest
                        Mode)\n”.center(100))
                        Display “1. View all cars available for rent.”
                        Display “2. New customer Register to Access other Details”
                        Display “3. Exit Guest Mode”
                        Display “Please enter your option:”
                        Read userDecision
                        IF (userDecision == 1) THEN
                                Access carNotRentOut function
                        ELIF (userDecision == 2) THEN
                                Access userRegister function
                        ELIF (userDecision == 3) THEN
                                Continue access to main function
                        ELSE
                                Display “Please select option again”
                        ENDIF
                ELIF (userDecision == 5) THEN
                        Access to account Recovery function
                        Continue access to main function
                ELIF (userDecision == 6) THEN
                        Access to license function
                ELIF (userDecision == 7) THEN
                        Display “-” *106
                        Display “Thank you for using our services!”
                        Display “Exiting...”
                        Display “-” *106
                        Exit main function
                ENDIF
        EXCEPT ValueError
                Display “Please only select the option above, retrying ...”
        ENDEXCEPT
    ENDDO
ENDFUNCTION
```

2. User Login Function (Registered Customer)

```
FUNCTION userLogin()
        authentication = True
        DOWHILE authentication
                Display “Login”
                Display “Please enter Username:”
                Read username
                Display “Please enter Password:”
                Read password
                Read lines in userDatabase.txt
                FOR EACH line IN lines
                        Strip space across line and Split space between line
                        IF (“Username:”+username == rec[0] and “Password:”+password
                        == rec[1]) THEN
                                authentication = False
                        ENDIF
                ENDFOR
                IF (authentication = False) THEN
                        Display f “Login successfully! Welcome {username}”
                        Return to userMenu function
                ELSE
                        Display “Invalid Username and Password”
                        Return to main function
                ENDIF
        ENDDO
ENDFUNCTION
```

3. User Menu Function (Registered Customer)

```
FUNCTION userMenu ()
        DOWHILE True:
                Display “Welcome to Car Rental Management System (Unregistred
                Customer)\n”.center(100)
                Display “Please select an option:”
                Display “1.Modify Personal Details.”
                Display “2.View Personal Rental History.”
                Display “3.View Detail of Cars to be Rented Out.”
                Display “4.Select and Book a car for a specific duration.”
                Display “5.Disable user account”
                Display “6.Log Out”
                Display “Please enter your option:” as integer
                Read userDecision
                IF (userDecision == 1) THEN
                        Access to modifyUser function
                        Continue access to userMenu function
                ELIF (userDecision == 2) THEN
                        Access to history function
                        Continue access to userMenu function
                ELIF (userDecision == 3) THEN
                        Access to carNotRentOut function
                        Continue access to userMenu function
                ELIF (userDecision == 4) THEN
                        Access to bookCar function
                        Continue access to userMenu function
                ELIF (userDecision == 5) THEN
                        Access to disableUserAccount function
                        Continue access to main function
                ELIF (userDecision == 6) THEN
                        Return to main function
                ELSE
                        Display “Please select option again”
                ENDIF
        ENDDO
ENDFUNCTION
```

4. Modify User Function (Registered Customer)

```
FUNCTION modifyUser()
        TRY
                DOWHILE True
                        userList = [ ]
                        Read userFile data in userDatabase.txt
                        Display “Please enter your password:”
                        Read password
                        modify = 0
                        FOR EACH line IN userFile
                                Strip space across line and split space between line
                                IF (“Password:”+password == record[1]) THEN
                                        Display “1.”, record[0]
                                        Display “2.”, record[1]
                                        Display “3.”, record[2]
                                        Display “4.”, record[3]
                                        Display “Please Enter A Record Number to Modify:”as
                                        integer
                                        Read inputNum
                                        Display “Dear User, Please Modify”
                                        Display “record[inputNum-1]
                                        IF (inputNum-1 == 0) THEN
                                                Display “Please Enter New Username:”
                                                Read new
                                                StringOne = “Username:”
                                                record[inputNum-1] = str(stringOne + new)
                                        IF (inputNum-1 == 1) THEN
                                                Display “Please Enter New Password:”
                                                Read new
                                                stringTwo = “Password:”
                                                record[inputNum-1] = str(stringTwo+ new)
                                        IF (inputNum-1 == 2) THEN
                                                Display “Please Enter New EmailAddress:”
                                                Read new
                                                stringThree = “EmailAddress:”
                                                record[inputNum-1] = str(stringThree + new)
                                        IF (inputNum-1 == 3) THEN
                                                Display “Please Enter New Contact:”
                                                Read new
                                                stringFour = “Contact:”
                                                record[inputNum-1] = str(stringFour + new)
                                        modify = 1
```

```
                                    Display record
                                    ENDIF
                            ENDIF
                            Append record into userList
                    ENDFOR
                    IF (modify == 0) THEN
                            Display “Dear User, You Have Enter Wrong Password”
                    ELSE
                            Read userList data in userDatabase.txt
                            i = 0
                            DOWHILE (I < len(userList))
                                    newRecord = join space between userList[i]
                                    Write newRecord+“\n”
                                    i = i + 1
                            ENDDO
                    ENDIF
                    Display “Dear User, do you wish to remodify? Yes ENTER (y),
                    NO ENTER (n):”
                    IF (remodify == “n” or remodofy == “N”)
                            Break
                    ENDIF
            ENDDO
        ENDTRY
        EXCEPT (IndexError)
                Display “Please only select the option”
                Return to userMenu function
        ENDEXCEPT
ENDFUNCTION
```

5. History Function (Registered Customer)

```
FUNCTION history()
        Display “-”*106
        Display “Search Rent History”.center(100)
        Display “-”*106)
        Display “Please enter IC: "
        Read search
        Display “-”*106)
        Display f “Your personal booking history:”.center(100)
        Display “-”*106)
        Read lines in transactionsDatabase.txt
        FOR EACH line IN lines
                Strip space across line and Split space between line
                IF (“ICNumber:”+search == record[1]) THEN
                        Display “i,.“\t”,record[3], “\t”,record[4], “\t”,record[9]”
                        Display ““\t”,record[5], “\t”,RentTime:”,record[6],
                        “\t”,record[7], “\t”, “ReturnTime:”,record[8], “\n””
                        i = i + 1
                ENDIF
        ENDFOR
ENDFUNCTION
```

6. Car Not Rent Out Function (Registered Customer / Guest)

```
FUNCTION carNotRentOut ()
        Display “-”*106
        Display “Available Cars For Rent".center(100)
        Display “-”*106)
        lines = [ ]
        Read lines in carsDatabase.txt
        i = 0
        FOR EACH line IN lines
                 IF (“Status:available” in line) THEN
                        i = i + 1
                        Display f “{i}.car = {line}”
                ENDIF
        ENDFOR
ENDFUNCTION
```

7. Book Car Function (Registered Customer)

```
FUNCTION bookCar ()
        Access carNotRentOut function
        Display “Please enter carID you wish to book”
        Read chooseCar
        Read lines in carsDatabase.txt
        Replace (“Price:”, “”) for each line
        FOR EACH line IN lines
                Strip space across line and Split between line
                IF (“CarID:”+chooseCar == record[0]) THEN
                        Display “This is the details of your selections:”
                        Display record[0]
                        Display record[1]
                        Display record[2]
                        Display “Price:”,record[3]
                        Display record[4]
                        Display “Please fill your information below”
                        Display “Please enter your name:”
                        Read bookName
                        Display “Please enter your ICNumber:”
                        Read idNumber
                        Display “Please enter a date for renting the car (in format YYYY-
                        MM-DD HH:MM): ”
                        Read inputDate
                        rentDate = datetime.strptime(inputDate, “Y-%m-%d %H:%M”)
                        Display int(“Please enter the total weeks you want to rent the car:”)
                        Read week
                        returnDate = rentDate + timedelta(weeks = week)
                        day = week * 7
                        Display “This is Your Payment Receipt”
                        Display record[0]
                        Display record[1]
                        Display record[2]
                        totalPrice = int(record[3]) * int(day)
                        Display “TotalPrice:”, totalPrice
                        Str(record[3])
                        Display “This is your total car rent duration:”
                        Display “Total Week Rent:”, week, “week”
                        Display “Total Day Rent:", day, “day”
                        Display “Your Rent Date and Time:”, rentDate
```

```
                    Display “Your Return Date and Time:”, returnDate
                    Display “Payment Has Been Made! Thank you!”
                    Append f “BookerName:{bookName} ICNumber:{idNumber}
                    {record[0]} {record[1]} {record[2]} RentDate:{str(rentDate)}
                    ReturnDate:{str(returnDate)} TotalPrice:{str(totalPrice)}\n” in
                    transactionsDatabase.txt
            ENDIF
        ENDFOR
        carList = []
        Read carFile data in carsDatabase.txt
        FOR EACH line IN carFile
            Strip space across line and Split space between line
            IF (“CarID:”+chooseCar == record[0]) THEN
                Record[4] = “Status:unavailable”
            ENDIF
            Append record to carList
        ENDFOR
        Write carFile data in carsDatabase.txt
        i = 0
        DOWHILE (i < len(carList))
            newRecord = join spaces between carList[i]
            Write newRecord+“\n”
            i = i + 1
        ENDDO
ENDFUNCTION
```

8. Disable User Account Function (Registered Customer)

```
FUNCTION disableUserAccount()
        Display “Please enter password:”
        Read password
        Read temp in userDatabase.txt
        Seek from index 0 to last index from userDatabase.txt
        FOR EACH line IN temp
                IF (not password in line) THEN
                        Write line
                ENDIF
        Truncate the data in userDatabase.txt
        ENDFOR
        Display “Account has been disable”
ENDFUNCTION
```

9. User Register Function (Unregistered Customer / Guest)

```
FUNCTION userRegister()
        Display “Sign up”
        Display “Please enter Username:”
        Read username
        Display “Please enter Password:”
        Read password
        Display “Please enter Email:”
        Read emailAddress
        Display “Please enter contact:”
        Read contactNumber
        Read lines in userDatabase.txt
        FOR EACH line IN lines
                Strip space across line and Split space between line
                IF (“Username:”+username == rec[0] or “Password:”+password == rec[1])
THEN
                        Display “account existed”
                        Return to main function
                ELIF (username == “” or password == “”) THEN
                        Display “username and password cannot be empty”
                        Return to main function
                ELIF (username == “ ” or password == “ ”) THEN
                        Display “username and password cannot be empty”
                        Return to main function
                ELSE:
                        Append “Username:”,username, “Password:”, password,
                        “EmailAddress:”, emailAddress and “Contact:”, contact into
                        userDatabase.txt
                        Display “Account successfully Registered”
                        Return to main function
                ENDIF
        ENDFOR
ENDFUNCTION
```

10. Admin Menu Function (Admin)

```
FUNCTION adminMenu()
        DOWHILE True:
                Display “-” *106
                Display “Welcome to Car Rental Management System (Administrator/
                Menu)\n”.center(100)”
                Display “Please select an option:”
                Display “1. Add Cars to be rented out”
                Display “2. Modify car details”
                Display “3. Display all records”
                Display “4. Search Specific record of”
                Display “5. Return a Rented Car.”
                Display “6. Exit Administration Mode”
                Display “-” *50
                Display “Please enter your option:” as integer
                Read userDecision
                IF (userDecision == 1) THEN
                        Display “-” *106)
                        Display “Welcome to Car Rental Management System
                        (Administrator/Add Cars) \n”.center(100))
                        Access to addCars function
                        Continue access to adminMenu function
                ELIF (userDecision == 2) THEN
                        Display “-”*106)
                        Display “Welcome to Car Rental Management System
                        (Administrator/Modify Cars)\n”.center(100)”
                        Access to modifyCarDetails function
                        Continue access to adminMenu function
                ELIF (userDecision == 3) THEN
                        Display “-”*106
                        Display “Welcome to Car Rental Management System
                        (Administrator/Display Cars Record)\n”.center(100)”
                        Display “1.Cars Rented Out”
                        Display “2.Cars available for Rent”
                        Display “3.Customer Bookings and Payments Details”
                        Display “4.Exit to Admin”
                        Display “Please enter your option:” as integer
                        Read userDecision
                        IF (userDecision == 1) THEN
                                Access to carRentOut function
                                Continue access to adminMenu function
```

```
                    ELIF (userDecision == 2) THEN
                        Access to carNotRentOut function
                        Continue access to adminMenu function
                    ELIF (userDecision == 3) THEN
                        Access to CusBookAndPay function
                        Continue access to adminMenu function
                    ELIF (userDecision == 4) THEN
                        Continue access to adminMenu function
                    ELSE
                        Display “Please select option again”
                    ENDIF
                ELIF (userDecision == 4) THEN
                    Display “-”*106
                    Display “Welcome to Car Rental Management System
                    (Administrator/Search Cars)\n".center(100)”
                    Display “1.Customer Booking”
                    Display “2.Customer Payment”
                    Display “3.Exit to Admin”
                    Display “-”*106
                    Display int(“Please enter your option:”)
                    Read userDecision
                    IF (userDecision == 1) THEN
                        Back to searchCusBook function
                        Continue access to adminMenu function
                    ELIF (userDecision == 2) THEN
                        Access searchCusPay function
                        Continue access to adminMenu function
                    ELIF (userDecision == 3) THEN
                        Continue access to adminMenu function
                    ELSE
                        Display “Please select option again”
                    ENDIF
                ELIF (userDecision == 5) THEN
                    Display “-”*106
                    Display “Welcome to Car Rental Management System
                    (Administrator/Return Rented Car)\n".center(100)”
                    Display “-”*106
                    Access to returnRentedCar function
                    Continue access to adminMenu function
                ELIF (userDecision == 6) THEN
                    Return to main function
                ELSE:
                    Display “Please select option again”
                ENDIF
        ENDDO
ENDFUNCTION
```

11. Add Cars Function (Admin)

```
FUNCTION addCars ()
        carStatus = “available”
        Display “Please enter number of cars you wish to add:”
        Read numberOfCars as integer
        FOR EACH count IN range of numberOfCars
                Display “Please enter car ID:”
                Read carID
                Display “Please enter car model name:”
                Read modelName
                Display “Please enter vehicle type:”
                Read vehicleType
                Display “Please enter price per day:”
                Read pricePerDay
                Display “carID:”, carID
                Display “ModelName:”, modelName
                Display “vehicleType:”, vehicleType
                Display “PricePerDay:”, pricePerDay
                Display “Status:”, carStatus
                Append f “CarID:{carID} ModelName:{modelName}
                VehicleType:{vehicleType}Price:{str(pricePerDay) Status:{carStatus}\n”
                in carsDatabase.txt file
        ENDFOR
ENDFUNCTION
```

12. Modify Car Detail (Admin)

```
FUNCTION modifyCarDetails ()
        TRY
                DOWHILE True
                        carList = [ ]
                        access carNotRentOut function to display available car for rent
                        Read carFile data in carsDatabase.txt
                        Display "Please enter car ID you wish to modify:"
                        Read carID
                        modify = 0
                        FOR EACH line IN carFile
                                Strip space across line and split space between line
                                IF ("CarID:"+carID == record[0]) THEN
                                        Display "1.", record[0]
                                        Display "2.", record[1]
                                        Display "3.", record[2]
                                        Display "4.", record[3]
                                        Display "Please Enter a Record Number to Modify:"as
                                        integer
                                        Read inputNum
                                        Display "Dear Admin, Please Modify"
                                        Display "record[inputNum-1]
                                        IF (inputNum-1 == 0) THEN
                                                Display "Please Enter New CarID:"
                                                Read new
                                                stringOne = "CarID:"
                                                record[inputNum-1] = str(stringOne + new)
                                        IF (inputNum-1 == 1) THEN
                                                Display "Please Enter New ModelName:"
                                                Read new
                                                stringTwo = "ModelName:"
                                                record[inputNum-1] = str(stringTwo+ new)
                                        IF (inputNum-1 == 2) THEN
                                                Display "Please Enter New VehicleType:"
                                                Read new
                                                stringThree = "VehicleType:"
                                                record[inputNum-1] = str(stringThree + new)
                                        IF (inputNum-1 == 3) THEN
                                                Display "Please Enter New Price:"
                                                Read new
                                                stringFour = "Price:"
                                                record[inputNum-1] = str(stringFour + new)
```

```
                                    Modify = 1
                                    Display record
                                    ENDIF
                            ENDIF
                            Append record into carList
                    ENDFOR
                    IF (modify == 0) THEN
                            Display “The value not exist”
                    ELSE
                            Write carFile data in carsDatabase.txt
                            i = 0
                            DOWHILE (i < len(carList))
                                    newRecord = join spaces between carList[i]
                                    Write newRecord + “\n”
                                    i = i + 1
                            ENDDO
                    ENDIF
                    Display “Dear Admin, do you wish to remodify? Yes ENTER (y),
                    NO ENTER (n):”
                    IF (remodify == “n”) or (remodify == “N”) THEN
                            Break
                    ENDIF
            ENDDO
        ENDTRY
        EXCEPT (IndexError)
                Display “Please only select above field number”
                Return to admin Menu Function
        ENDEXCEPT
ENDFUNCTION
```

13. Car Rented Out Function (Admin)

```
FUNCTION carRentOut ()
        Display "-"*106
        Display "Rented Out Cars".center(100)
        Display "-"*106)
        lines = [ ]
        Read lines in carsDatabase.txt
        i = 0
        FOR EACH line IN lines
                IF ("Status:unavailable" in line) THEN
                        i = i + 1
                        Display f "{i}.car = {line}"
                ENDIF
        ENDFOR
END FUNCTION
```

14. Customer Booking and Payment Function (Admin)

```
FUNCTION cusBookAndPay()
        Display "-"*106
        Display "Customer Booking and Payment".center(100)
        Display "-"*106)
        lines = [ ]
        Read lines in transactionsDatabase.txt
        i = 0
        FOR EACH line IN lines
                i = i + 1
                Display f "{i}.Booking = {line}"
        ENDFOR
ENDFUNCTION
```

15. Search Customer Booking Function (Admin)

```
FUNCTION searchCusBook()
        Display “-”*106
        Display “Search Customer Booking”.center(100)
        Display “-”*106)
        Display “Please enter customer name to check booking history: ".center(100)
        Read search
        Display “-”*106)
        Display f “Booking history of {search} customer:”
        Display “-”*106)
        Read lines in transactionsDatabase.txt
        FOR EACH line IN lines
                Strip space across line and Split space between line
                IF (“BookerName:”+search == record[0]) THEN
                        Display record[0],“\t”, record[1],“\t”,  record[2] ,“\t”,
                        record[3],“\t”,record[4]
                ENDIF
        ENDFOR
ENDFUNCTION
```

16. Search Customer Payment Function (Admin)

```
FUNCTION searchCusPay()
        Display “-”*106
        Display “Search Customer Payment”.center(100)
        Display “-”*106)
        Display “Please enter customer name to check payment history: "
        Read search
        Display “-”*106)
        Display f “Payment history of {search} customer:”.center(100)
        Display “-”*106)
        Read lines in transactionsDatabase.txt
        i = 1
        FOR EACH line IN lines
                Strip space across line and Split space between line
                IF (“BookerName:”+search == record[0]) THEN
                        Display “i,.“\t”,record[0], “\t”,record[1], “\t”,record[9]”
                        Display ““\t”,record[5], “\t”,RentTime:”,record[6],
                        “\t”,record[7], “\t”, “ReturnTime:”,record[8], “\n””
                        i = i + 1
                ENDIF
        ENDFOR
ENDFUNCTION
```

17. Return Rented Car Function (Admin)

```
FUNCTION returnRentedCar()
        Access carRentOut function
        Display “Please enter carID you wish to return:”
        Read chooseCar
        Read lines in carsDatabase.txt
        FOR EACH line IN lines
                Strip space across line and split space between line
                IF (“CarID:”+chooseCar == record[0]) THEN
                        Display “This is your returned carID:”
                        Display record[0]
                        Display record[1]
                        Display record[2]
                        Display record[3]
                        Display “You have successfully returned the car”
                ENDIF
        ENDFOR
        carList= [ ]
        Read carFile data in carsDatabase.txt
        FOR EACH line IN carFile
                Strip space across line and split space between line
                IF (“CarID:”+chooseCar == record[0]) THEN
                        record[4] = “Status:available”
                ENDIF
                Append record into carList
        ENDFOR
        Write carFile data in carsDatabase.txt
        i = 0
        DOWHILE (i < len(carList)):
                newRecord = join spaces between carList[i]
                Write newRecord+“\n”
                i = i + 1
        ENDDO
ENDFUNCTION
```

18. Account Recovery Function (Account Recovery for Registered Customer)

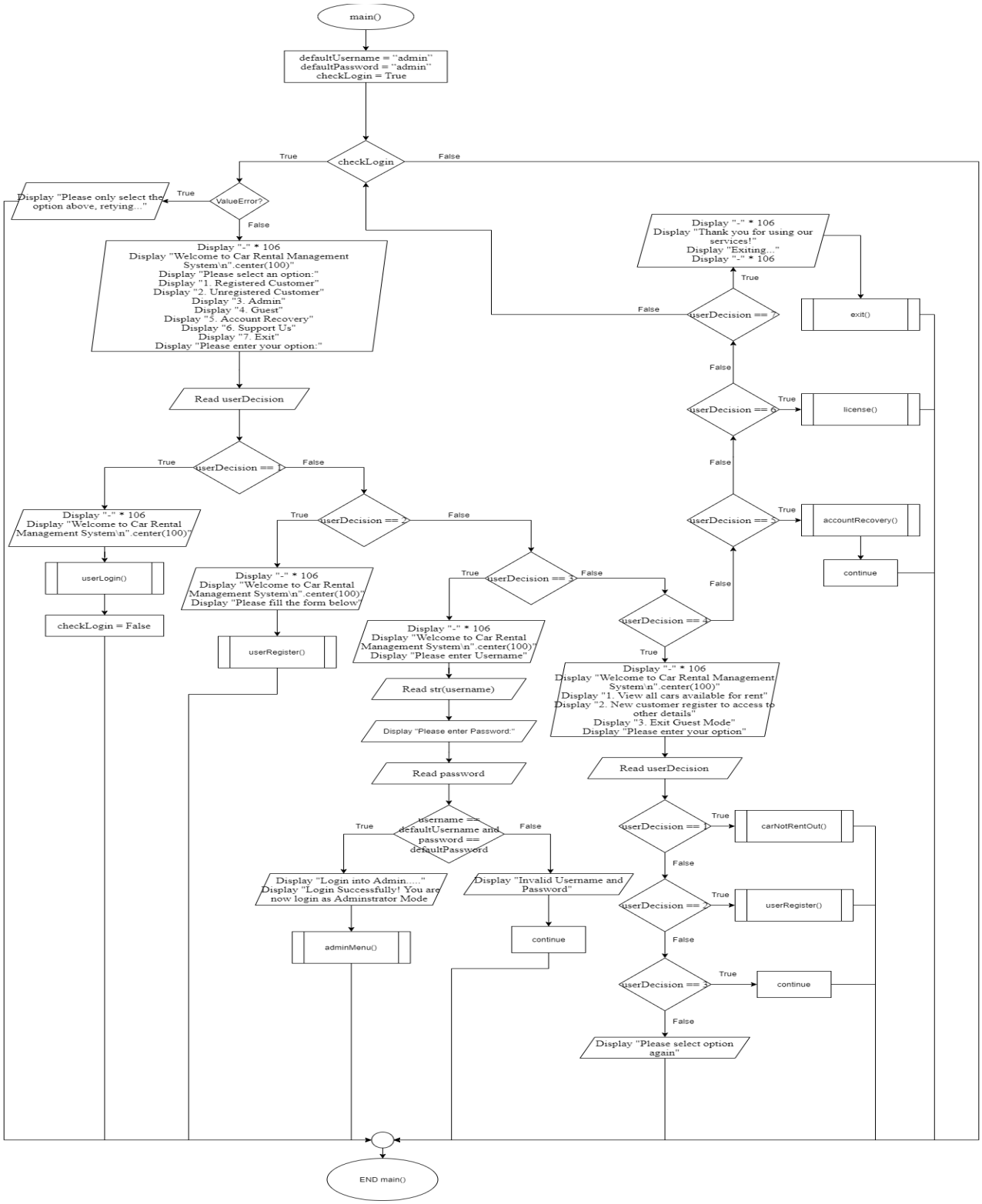
```
FUNCTION accountRecovery()
        Display “Welcome to Car Rental Management System (Account
        Recovery)\n”.center(100)”
        Display “Please enter phone number:”
        Read searchContact
        Read temp in userDatabase.txt file
        Seek from index 0 to last index from userDatabase.txt file
        FOR EACH line IN temp
                IF (searchContact == “”) THEN
                        Display “please enter something”
                ELIF (searchContact in line) THEN
                        Display “This is your account crendintials”
                        Display line
                ELSE:
                        Display “Please enter correct phone number”
                ENDIF
        ENDFOR
ENDFUNCTION
```

19. License Function (License)

```
FUNCTION license()
        Display “Thank you for Supporting Our System”
        Display “Donate Link”
        Display “License”
        Display “Github Repo”
        Display “Press 1 to exit:” as integer
        Read userDecision
        IF (userDecision == 1) THEN
                Return to main function
        ELSE
                Continue access license function
        ENDIF
ENDFUNCTION
```

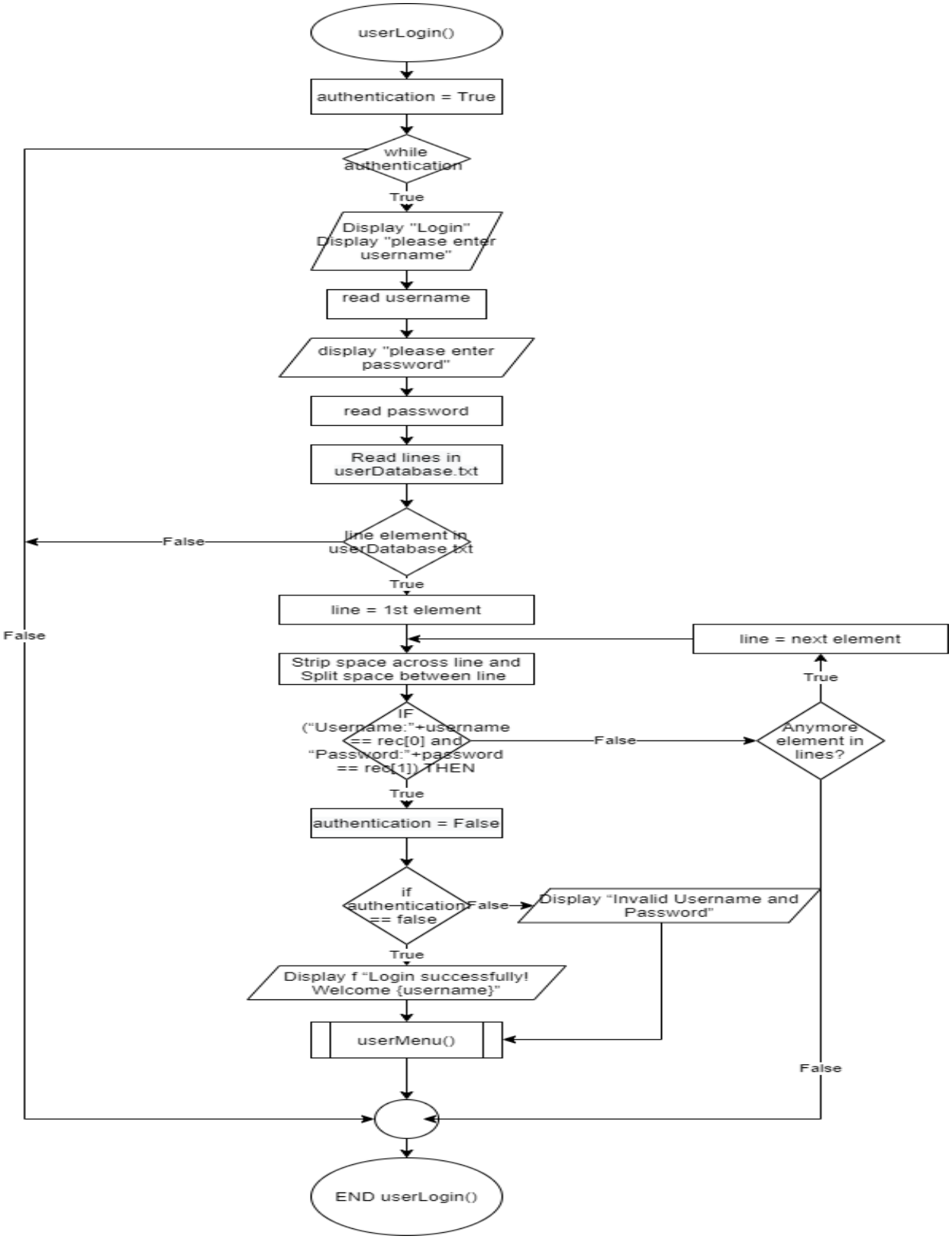
3.0 Flowchart

1. Main Function



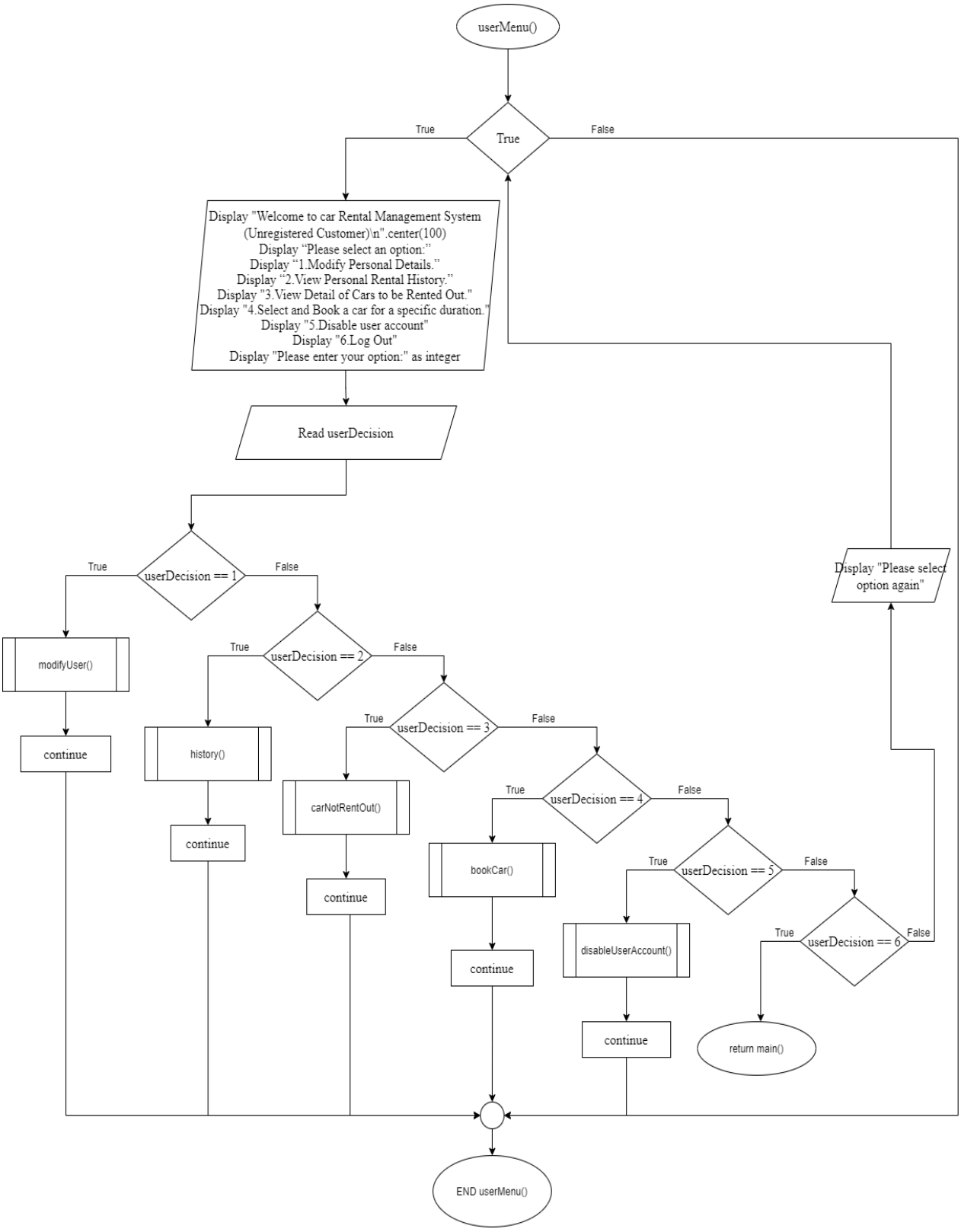
[Figure 3.1 Flowchart of Main Function]

2. User Login Function (Registered Customer)



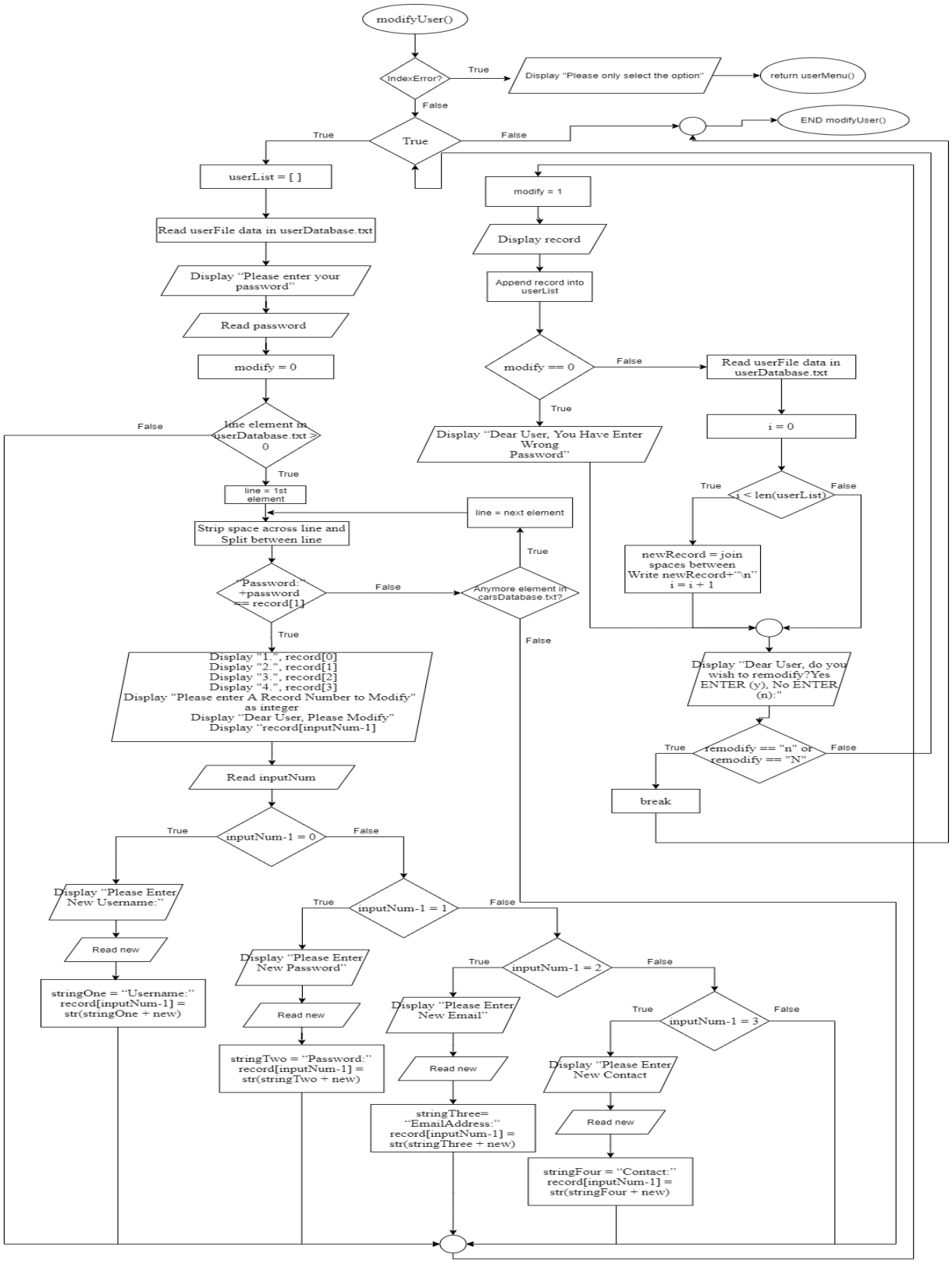
[Figure 3.2 Flowchart of User Login Function]

3. User Menu Function (Registered Customer)



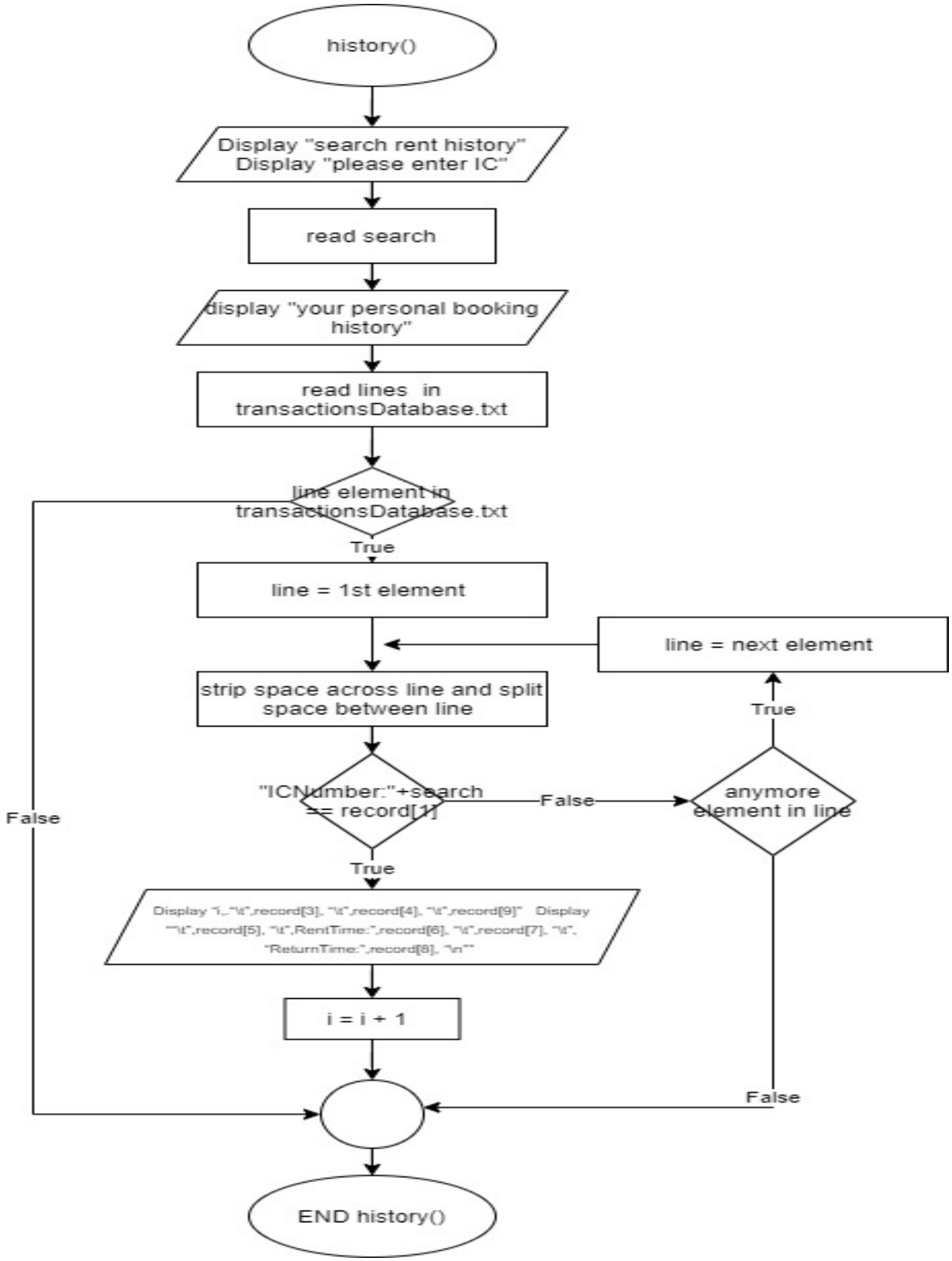
[Figure 3.3 Flowchart of User Menu Function]

4. Modify User Function (Registered Customer)



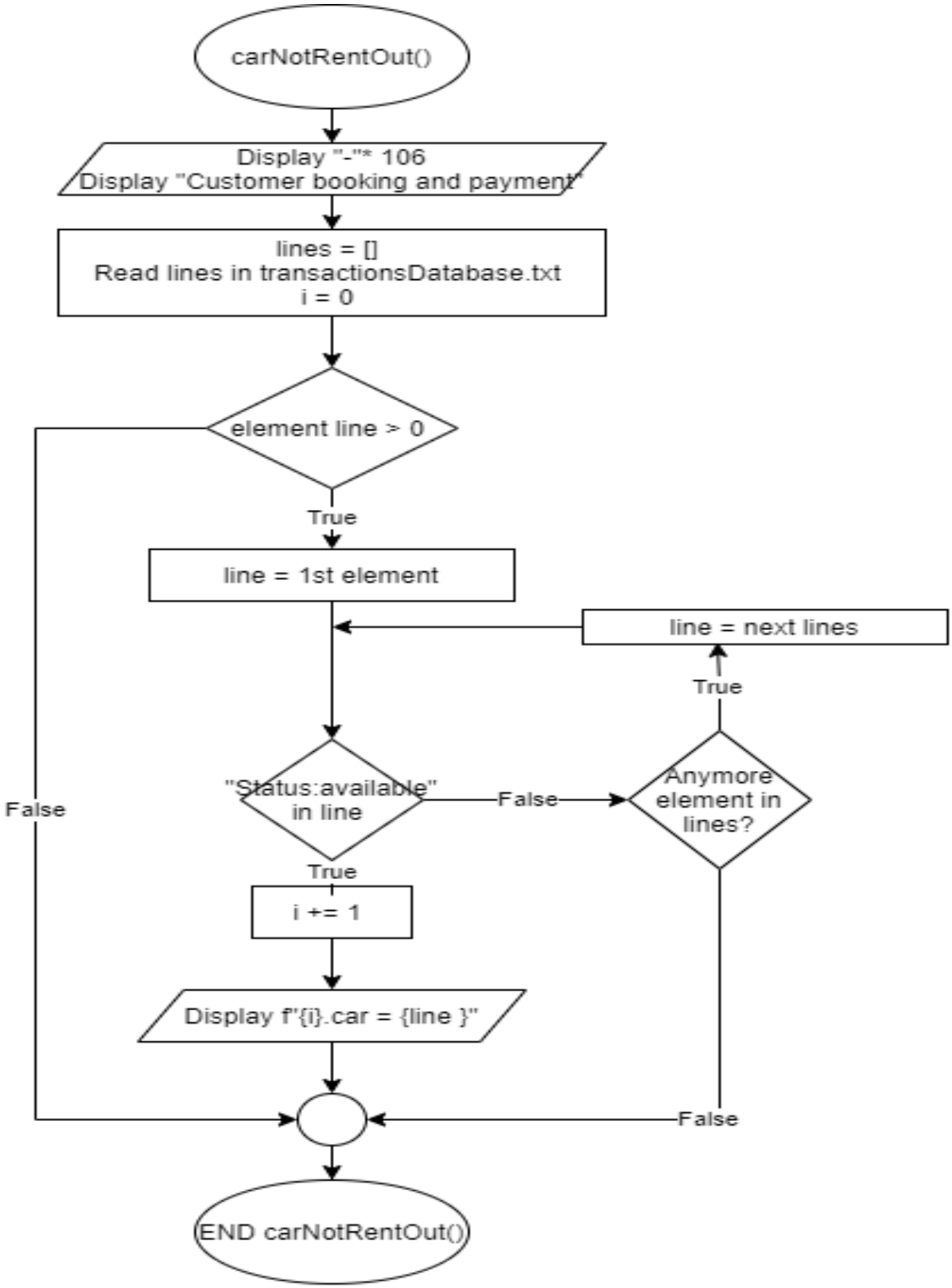
[Figure 3.4 Flowchart of Modify User Function]

5. History Function (Registered Customer)



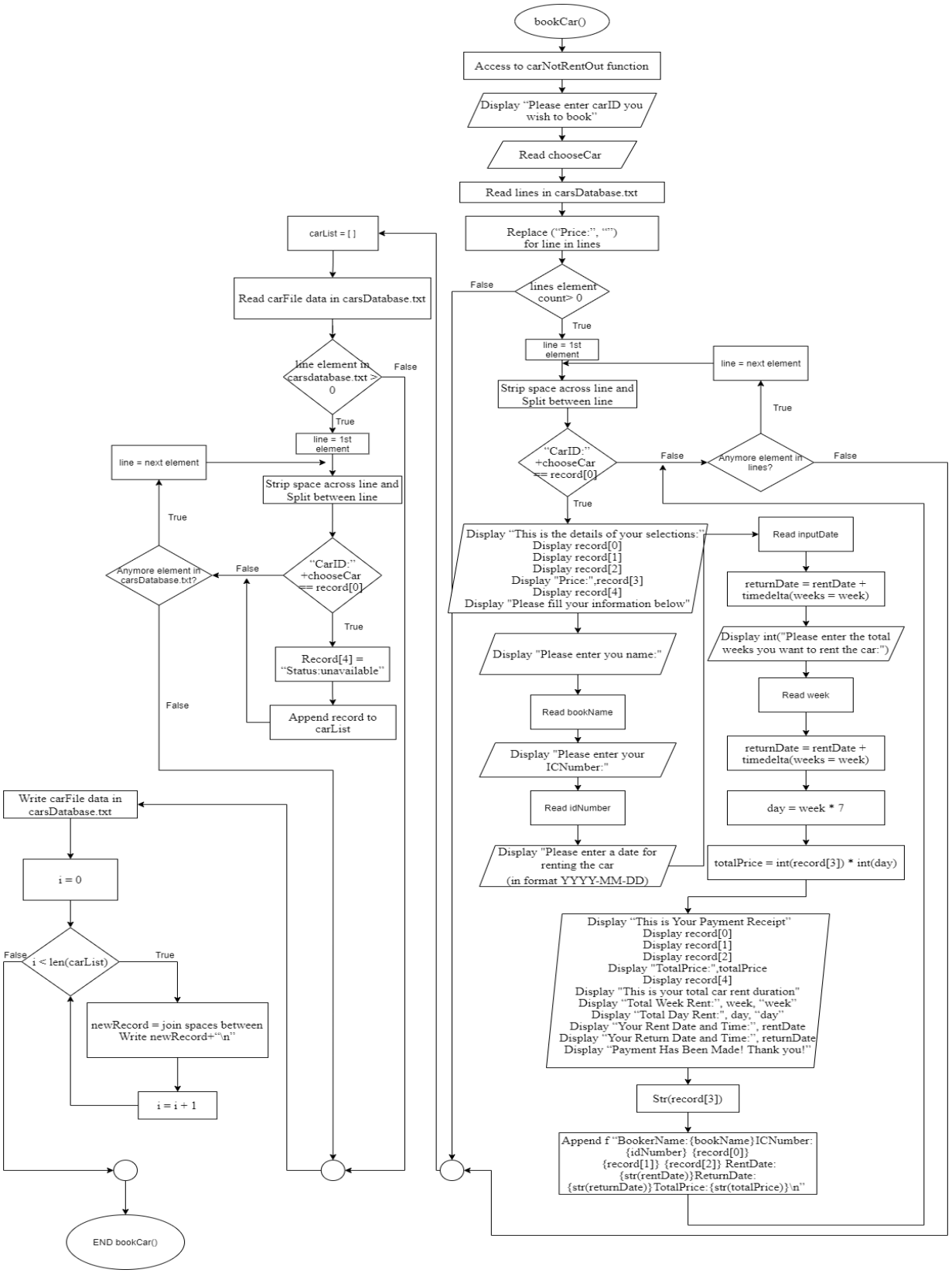
[Figure 3.5 Flowchart of History Function]

6. Car Not Rent Out Function (Registered Customer / Guest)



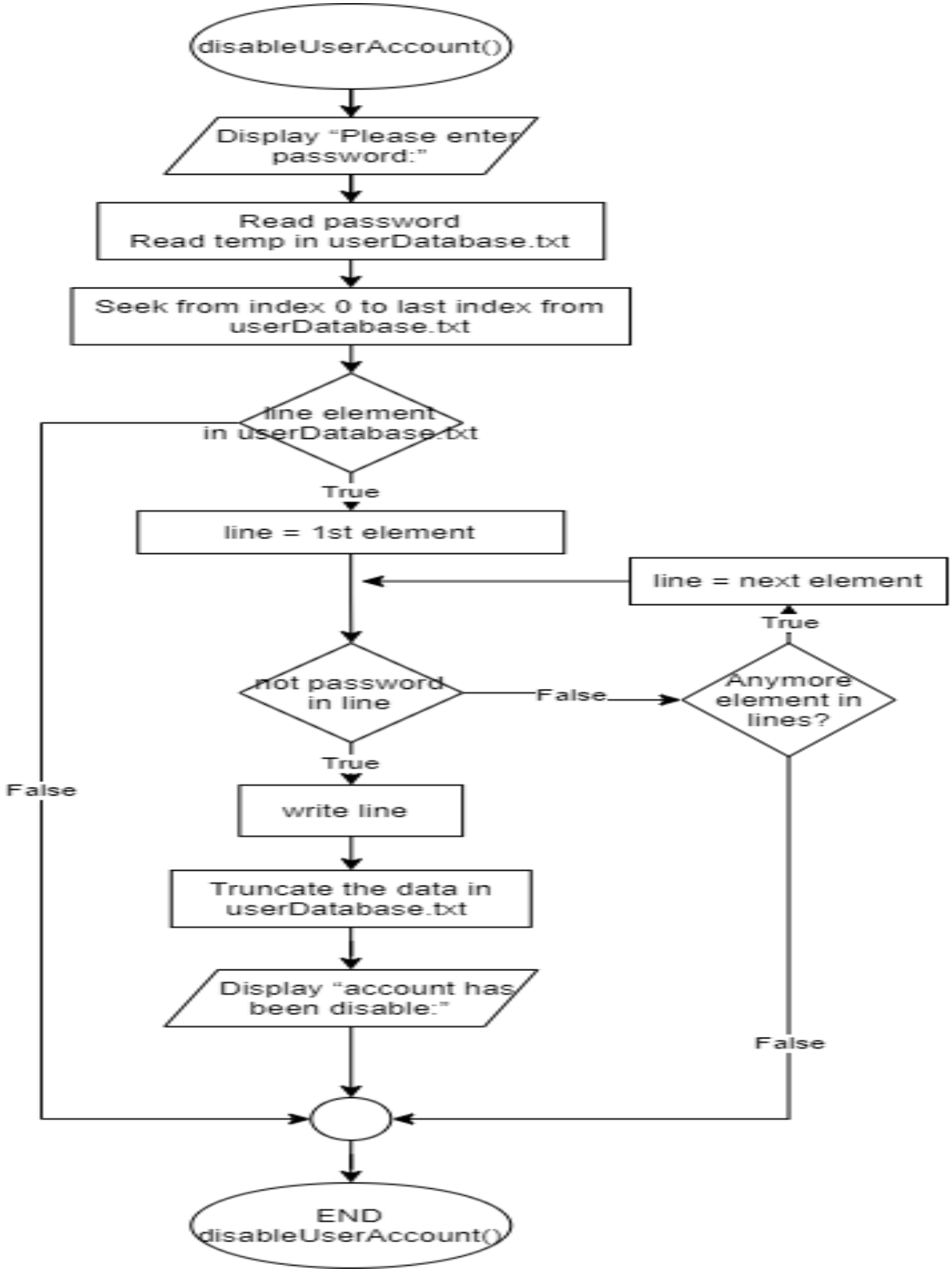
[Figure 3.6 Flowchart of Car Not Rent Out Function]

7. Book Car Function (Registered Customer)



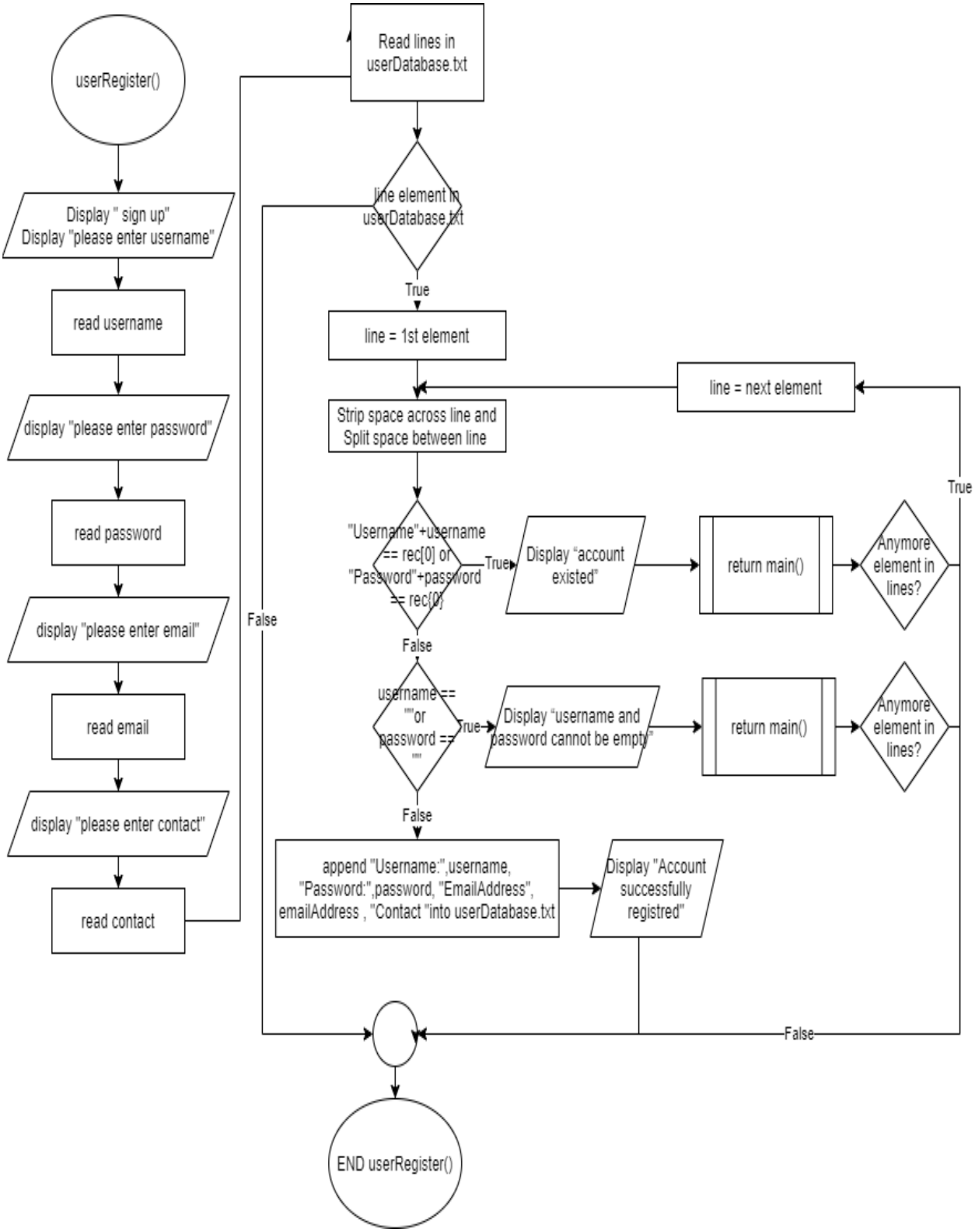
[Figure 3.7 Flowchart of Book Car Function]

8. Disable User Account Function (Registered Customer)



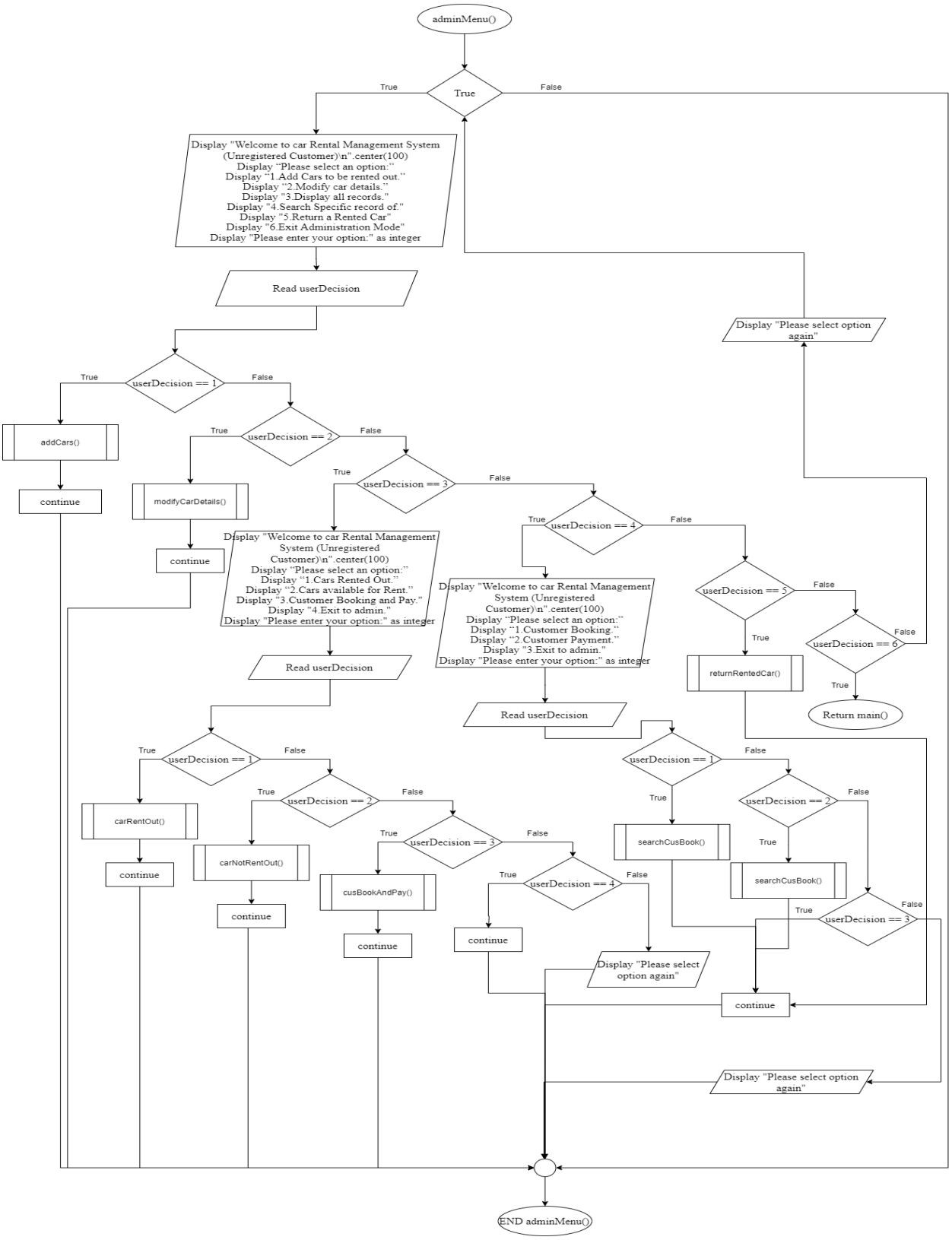
[Figure 3.8 Flowchart of Disable User Account Function]

9. User Register Function (Unregistered Customer / Guest)



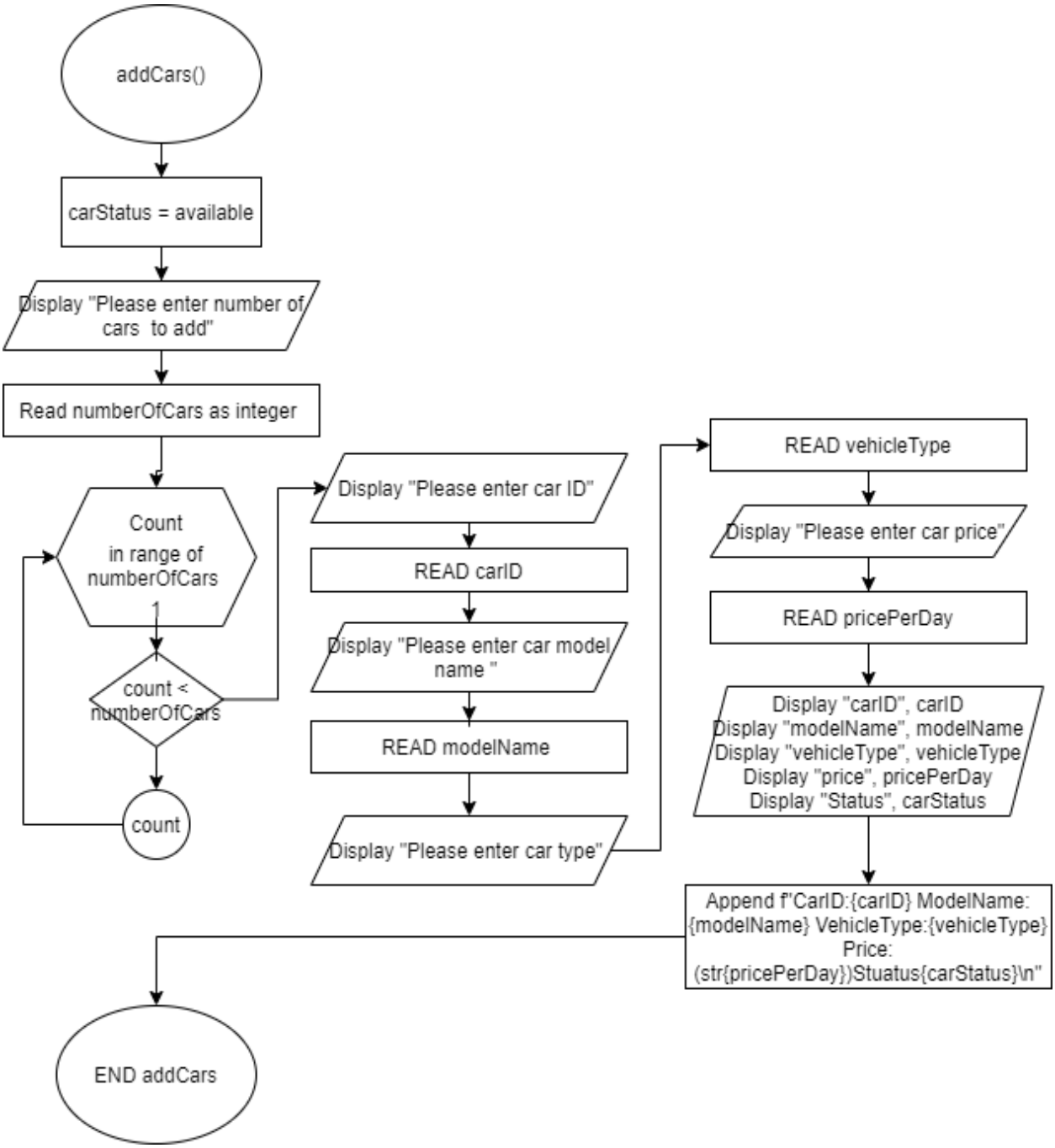
[Figure 3.9 Flowchart of User Register Function]

10. Admin Menu Function (Admin)



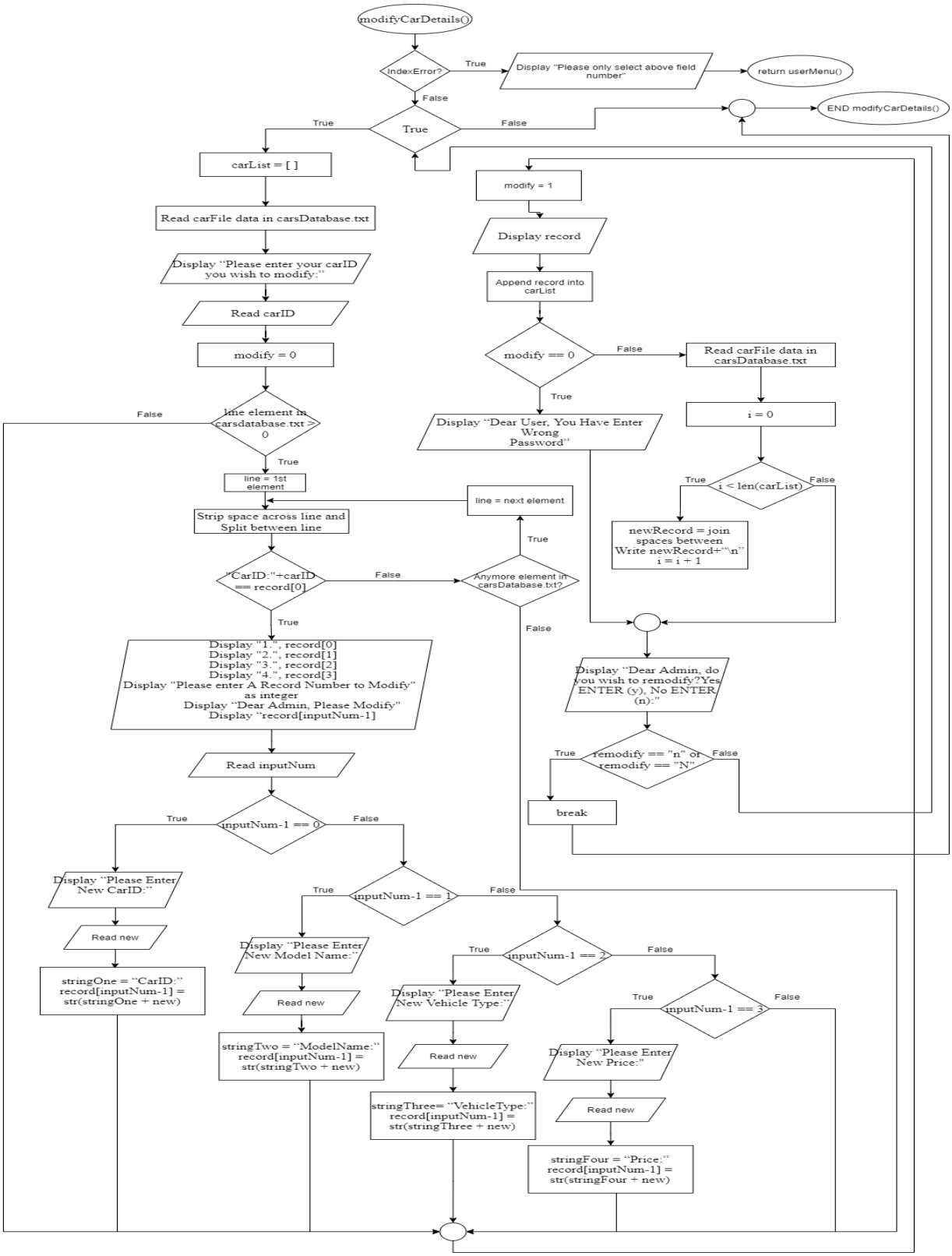
[Figure 3.10 Flowchart of Admin Menu Function]

11. Add Cars Function (Admin)



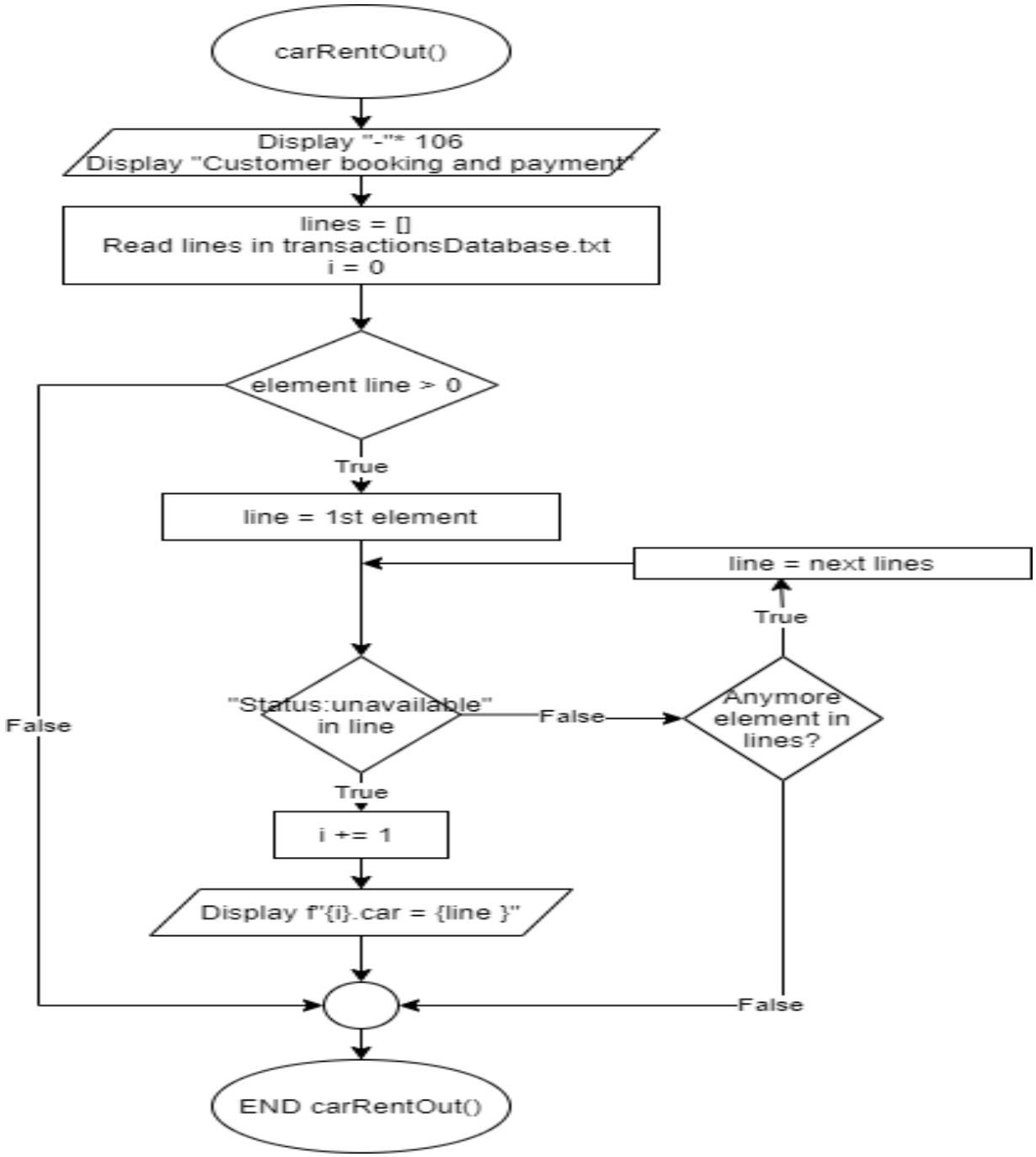
[Figure 3.11 Flowchart of Add Cars Function]

12. Modify Car Detail (Admin)



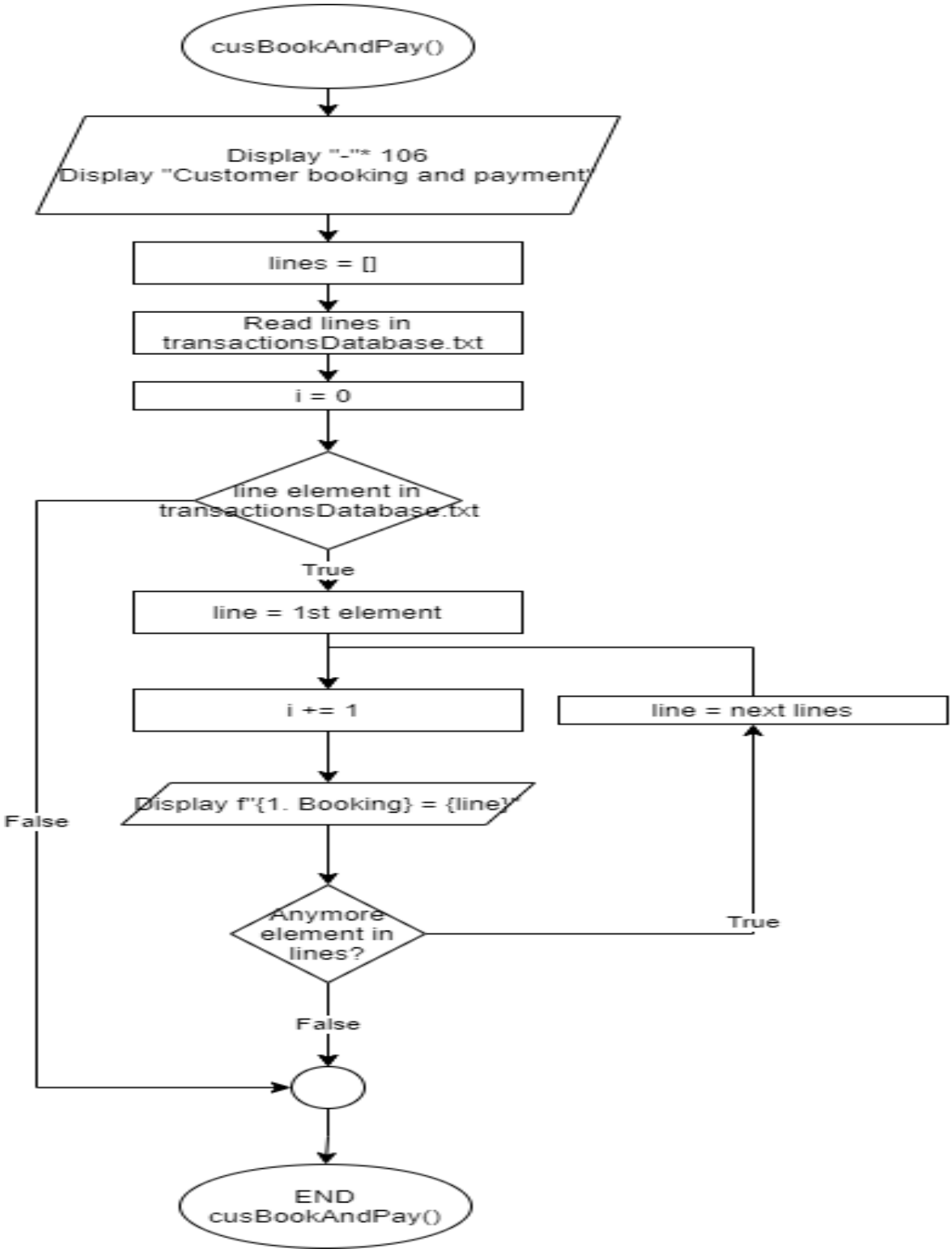
[Figure 3.12 Flowchart of Modify Car Detail Function]

13. Car Rented Out Function (Admin)



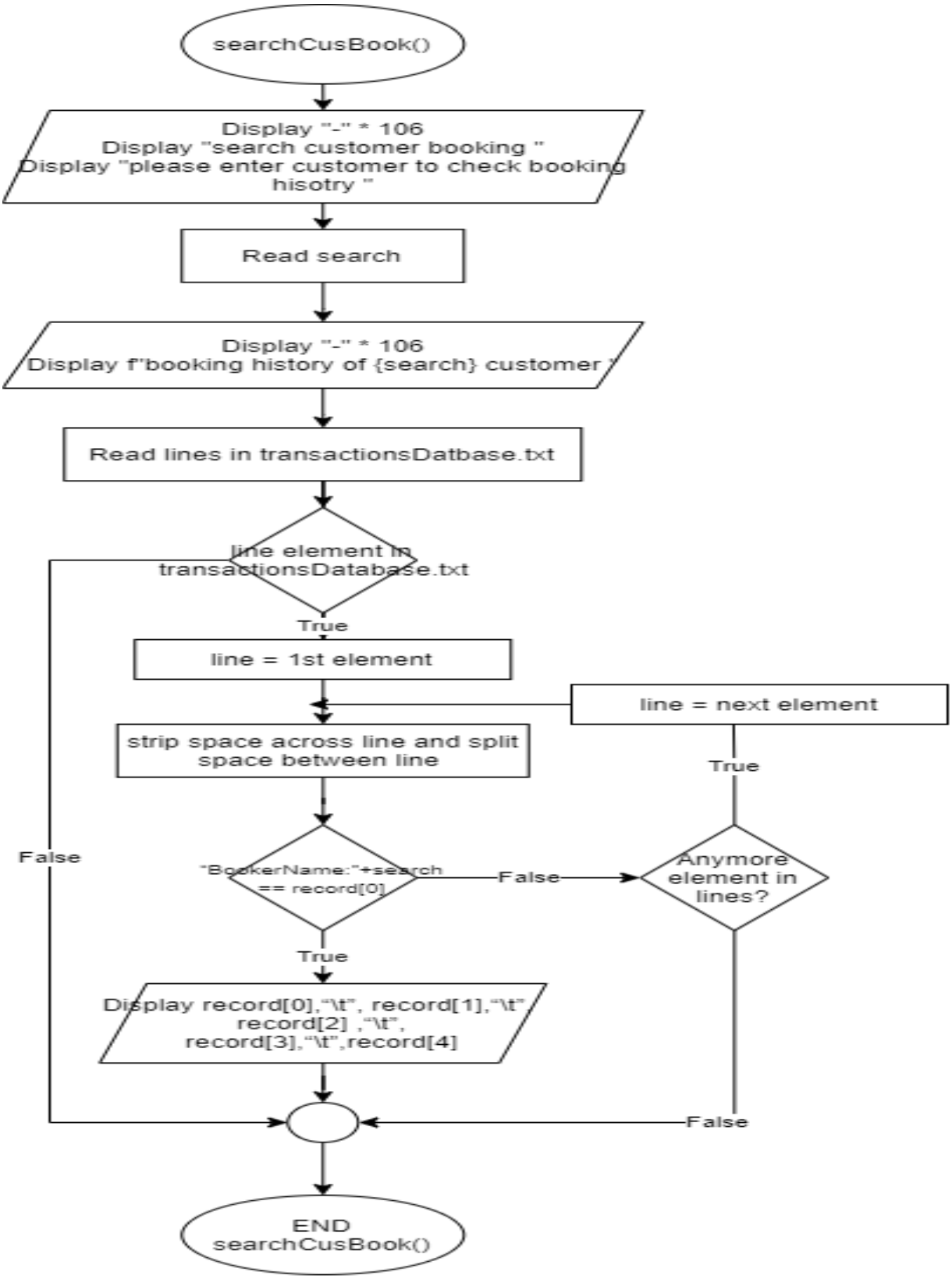
[Figure 3.13 Flowchart of Car Rented Out Function]

14. Customer Booking and Payment Function (Admin)



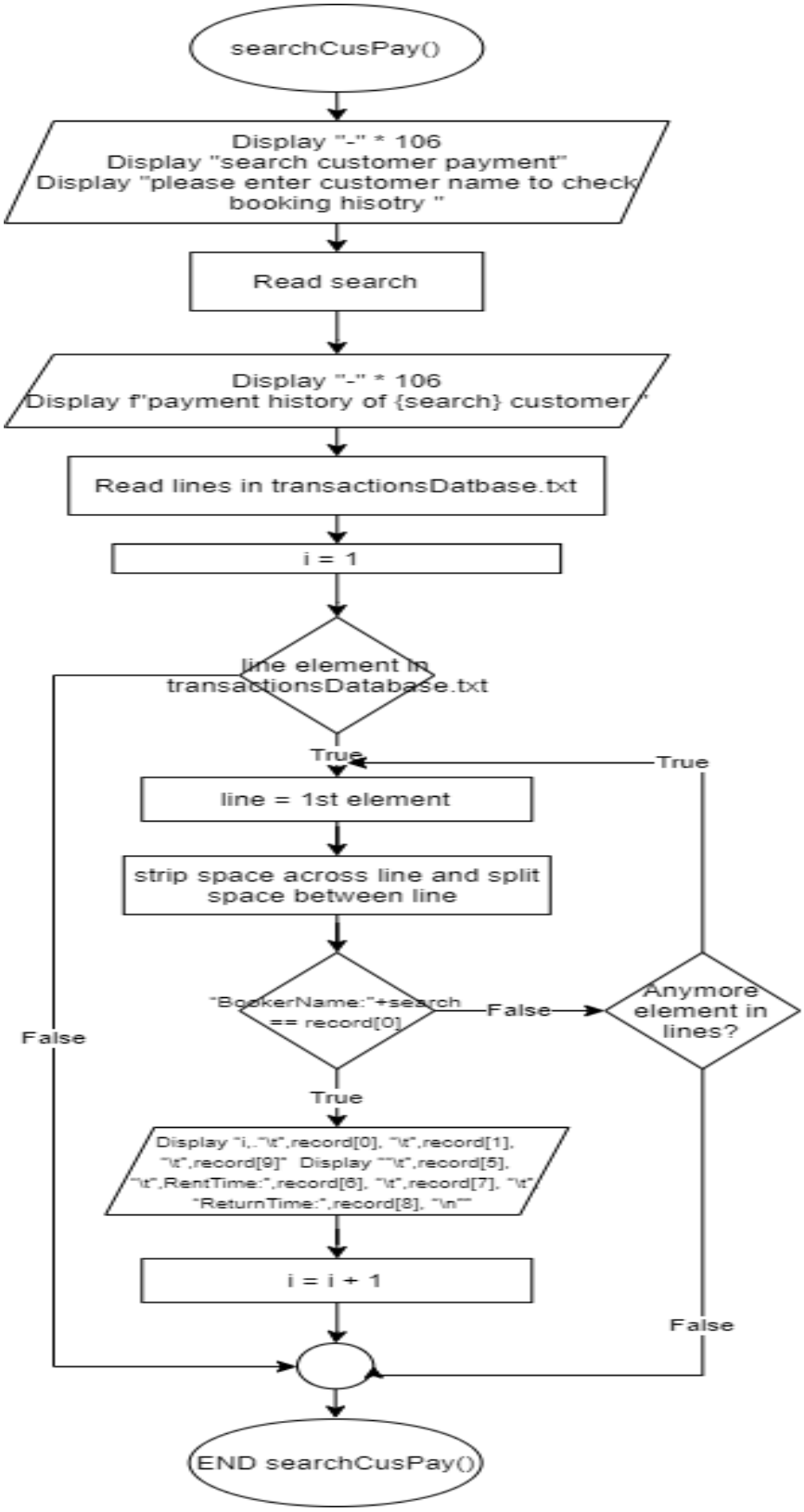
[Figure 3.14 Flowchart of Customer Booking and Payment Function]

15. Search Customer Booking Function (Admin)



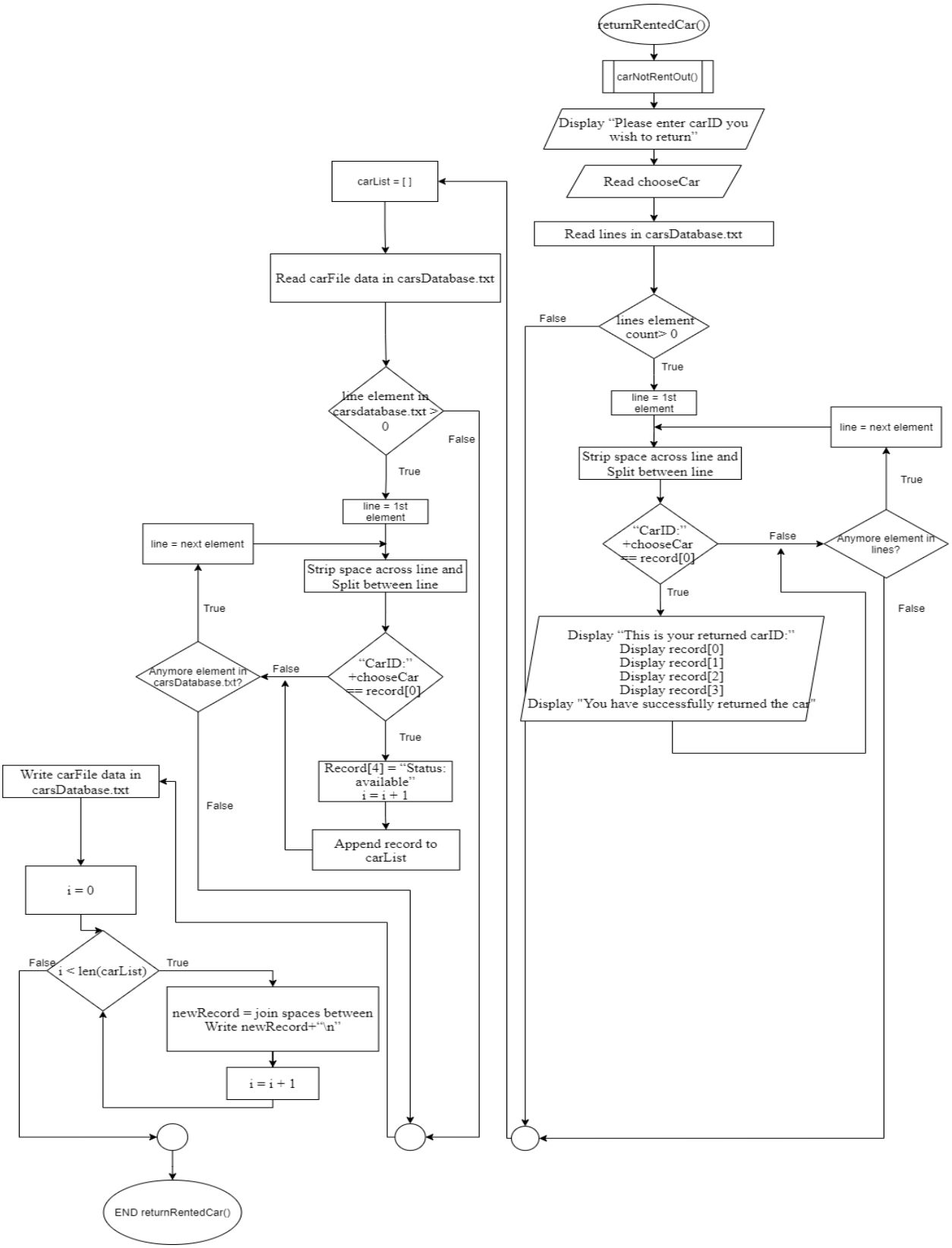
[Figure 3.15 Flowchart of Search Customer Booking Function]

16. Search Customer Payment Function (Admin)



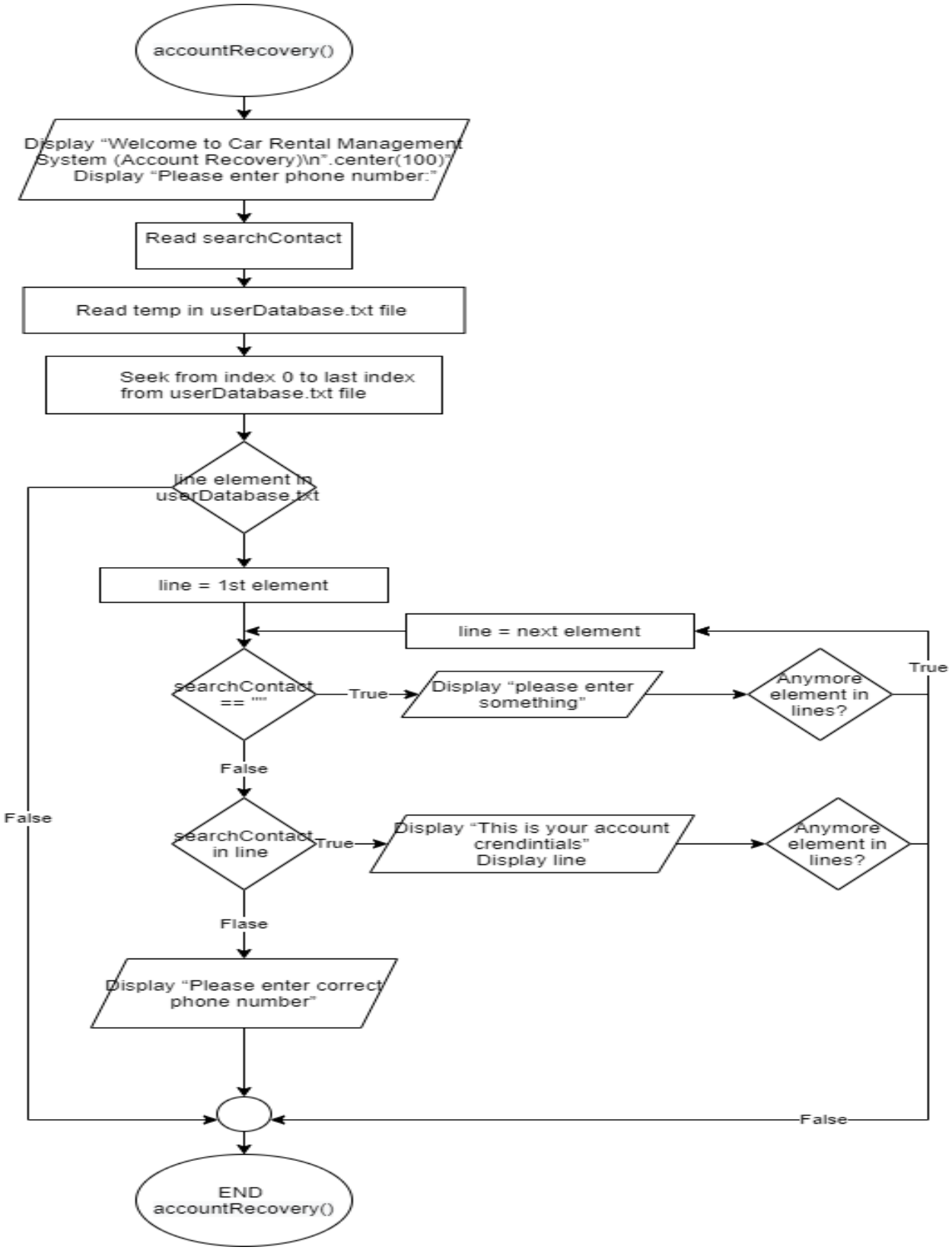
[Figure 3.16 Flowchart of Search Customer Payment Function]

17. Return Rented Car Function (Admin)



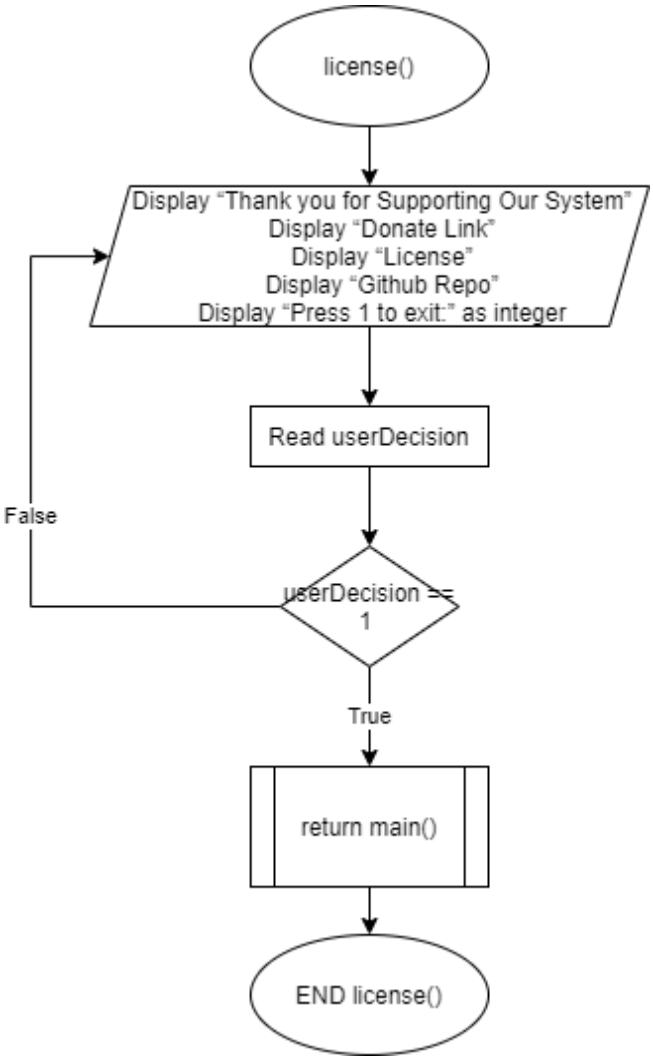
[Figure 3.17 Flowchart of Return Rented Car Function]

18. Account Recovery Function (Account Recovery for Registered Customer)



[Figure 3.18 Flowchart of Account Recovery Function]

19. License Function (License)



[Figure 3.19 Flowchart of license Function]

4.0 Source Code Explanation

1. Main Function

```
def main():
    #default username and password
    defaultUsername = "admin"
    defaultPassword = "admin"
    checkLogin = True
    # Loop through function
    while checkLogin :
        try:
            print("-"*106)
            print ("\033[1;34;40m Welcome to Car Rental Management System\n".center(100))
            print("    _____                 _____           _       _    _____           _                 ")
            print("   / ____|               |  __ \         | |     | |  / ____|         | |
")
            print(" | |      __ _ _ __   | |__) |___ _ __ | |_ __ _| | | (___   _   _ ___| |_ ___ _ __ ___
")
            print(" | |     / _` | '__|  |  _  // _ \ '_ \| __/ _` | |  \___ \| | | / __| __/ _ \ '_ ` _
\ ")
            print(" | |___| (_| | |     | | \ \  __/ | | | || (_| | |  ____) | |_| \__ \ ||  __/ | | | |
|")
            print("  \_____\__,_|_|     |_|  \_\___|_| |_|\__\__,_|_|  |_____/ \__, |___/\__\___|_| |_|
|_|")
            print("                                                              __/ |              ")
            print("                                                             |___/
  ")

            print("Please select an option: ")
            print( "1. Registered Customer ")
            print( "2. Unregistered Customer ")
            print( "3. Admin")
            print( "4. Guest")
            print( "5. Account Recovery")
            print( "6. Support Us "+"\u2764\ufe0f")
            print( "7. Exit")
            # userInput Option
```

[Figure 4.1 shows Part of Source Code from Main Function]

```
            # userInput Option
            userDecision = int(input("Please enter your option: "))
            if userDecision == 1:
                print("-"*106)
                print("Welcome to Car Rental Management System (Registered Customer/User
Login)\n".center(100))
                print("-"*106)
                userLogin()
                checkLogin = False

            elif userDecision == 2:
                print("-"*106)
                print("Welcome to Car Rental Management System (Unregistered Customer/User
Registration)\n".center(100))
                print("-"*106)
                print("Please fill the form below:")
                userRegister()


            elif userDecision == 3:
                print("-"*106)
                print("Welcome to Car Rental Management System (Administrator)\n".center(100))
                # userInput username and password
                username = str(input("Please enter Username: "))
                password = input("Please enter Password: ")
                # Check if username and password are same with defaultUsername and defaultPassword
                if username == defaultUsername and password == defaultPassword:
                    print("Loging into Admin ........")
                    print("Login Successfully! You are now login as Administrator Mode ! ")
                    adminMenu()
                else:
                    print("Invalid Username and Password")
                    continue
```

[Figure 4.1 shows Part of Source Code from Main Function]

Admin Username	Admin Password
admin	admin

[Table 4.1 shows Credentials of Administration Login]

```
            elif userDecision == 4:
                print("-"*106)
                print("Welcome to Car Rental Management System (Guest Mode)\n".center(100))
                print("-"*106)
                print("1. View all cars available for rent. ")
                print("2. New customer Register to Access other Details")
                print("3. Exit Guest Mode")
                print("-"*106)
                # userInput option
                userDecision = int(input("Please enter your option: "))
                if userDecision == 1:
                    carNotRentOut()
                elif userDecision == 2:
                    userRegister()
                elif userDecision == 3:
                    continue
            elif userDecision == 5:
                accountRecovery()
                continue
            elif userDecision == 6:
                license()
            elif userDecision == 7:
                print("-"*106)
                print("Thank you for using our services!")
                print("Exiting...")
                print("-"*106)
                exit()
        # Check if value error
        except ValueError:
            print("Please only select the option above, retrying ...")


main()
```

[Figure 4.1 shows Part of Source Code from Main Function]

The main() function will be the initial menu that the user sees when they run the application; it will display all of the choices, ascii art, and some nice welcome words. Meanwhile, it also allows users to select their own alternatives before moving on to the following tasks. If a user accidentally enters any characters or alphabet, it will be deemed a value error and will return the user to the main menu ()

2. User Login Function (Registered Customer)

```
def userLogin():
    authentication = True
    while authentication:
        print("Login")
        username = str(input("Please enter Username: "))
        password = input("Please enter Password: ")
        # Check Username and Password is Same As the List
        with open("userDatabase.txt","r") as userAccountData:
            lines = userAccountData.readlines()
            for line in lines:
                rec = line.strip("\n").split(" ")
                if "Username:"+username == rec[0] and "Password:"+password == rec[1]:
                    authentication = False
        if authentication == False:
            print(f"Login succesfully! Welcome {username}")
            userMenu()
            break
        else:
            print("Invalid Username and Password")
            main()
```

[Figure 4.2 shows Source Code of User Login]

Before a user may access any user features or activities, the userLogin() function will be used to authenticate them. The code above will prompt the user for their account credentials, which were previously entered into the userRegister ().

This code will also check the whether the input credentials is in userDatabase.txt, if not it will stop user for logging in the system.

3. User Menu Function (Registered Customer)

```
def userMenu():
    while True:
        print("-"*106)
        print("Welcome to Car Rental Management System (Registered Customer/ User Menu)\n".center(100))
        print("Please select an option:")
        print("1.Modify Personal Details.")
        print("2.View Personal Rental History.")
        print("3.View Detail of Cars to be Rented Out.")
        print("4.Select and Book a car for a specific duration.")
        print("5.Disable user account.")
        print("6.Log Out")
        # userInput Option
        userDecision = int(input("Please enter your option: "))
        if userDecision == 1:
            modifyUser()
            continue
        elif userDecision == 2:
            history()
            continue
        elif userDecision == 3:
            carNotRentOut()
            continue
        elif userDecision == 4:
            bookCar()
            continue
        elif userDecision == 5:
            disableUserAccount()
            return main()
        elif userDecision == 6:
            return main()
        else:
            print("Please select option again")
```

[Figure 4.3 shows Source Code of User Menu Function]

To prevent the user from returning to main() immediately after completing their activities, we built a userMenu() that displays all options for registered users while also preventing the application from returning the user to main().

4. Modify User Function (Registered Customer)

```
def modifyUser():
    while True:
        userList= []
        with open("userDatabase.txt","r") as userFile:
            password = input("Please enter your password: ")
            modify = 0
            for line in userFile:
                record = line.strip().split(" ")
                if "Password:"+password == record[1]:
                    print("1.",record[0])
                    print("2.",record[1])
                    print("3.",record[2])
                    print("4.",record[3])
                    inputNum = int(input("Please Enter A Record Number to Modify: "))
                    print(record[inputNum-1])
                    print("Dear Customer, Please Modify [EXACTLY BASED ON FORMAT BELOW]:)")
                    print("-"*100)
                    print("Username: (New Value)")
                    print("Password:(New Value)")
                    print("EmailAddress:(New Value)")
                    print("Contact:(New Value)")
                    print("-"*100)
```

[Figure 4.4 shows Part of Source Code from User Details Modification]

```
                    if inputNum-1 == 0:
                        new = input("Please Enter New Username:")
                        stringOne = "Username:"
                        record[inputNum-1] =  str(stringOne + new)
                    if inputNum-1 == 1:
                        new = input("Please Enter New Password:")
                        stringTwo = "Password:"
                        record[inputNum-1] =  str(stringTwo + new)
                    if inputNum-1 == 2:
                        new = input("Please Enter New EmailAddress:")
                        stringThree = "EmailAddress:"
                        record[inputNum-1] =  str(stringThree + new)
                    if inputNum-1 == 3:
                        new = input("Please Enter New Contact:")
                        stringFour = "Contact:"
                        record[inputNum-1] =  str(stringFour + new)
                        print("Please Enter A Valid Record Number to Remodify")
                        modifyUser()
                    modify = 1
                userList.append(record)
```

[Figure 4.4 shows Part of Source Code from User Details Modification]

```
if modify == 0:
    print("Dear User, You Have Enter Wrong Password")
else:
    with open("userDatabase.txt","w") as userFile:
        i = 0
        while (i < len(userList)):
            newRecord = " ".join(userList[i])
            userFile.write(newRecord+"\n")
            i += 1

remodify = input("To Leave ENTER N To Proceed ENTER ANY KEY:")
if remodify == "n" or remodify == "N":
    break
```

[Figure 4.4 shows Part of Source Code from User Details Modification]

The userModify() function is used to change a user's personal information and credentials. The user could change whatever information they like, such as their account username, password, email address, or phone number.

The code structure is straightforward and easy; it simply asks for the user's password, and once the account is verified, the program prints all password-related information. The user will be asked which field he or she wants to change. After the modification has been completed, the application will prompt the user to continue or exit the alteration process.

5. History Function (Registered Customer)

```
def history():
    print("-"*106)
    print("Rent History".center(100))
    print("-"*106)
    search= input("Please enter IC:")
    with open('transactionsDatabase.txt', 'r') as data:
        lines = data.readlines()
        for line in lines:
            record = line.strip("\n").split(" ")
            if "ICNumber:"+search == record[1]:
                print("This your personal booking history:")
                print(record[3])
                print(record[4])
                print(record[5])
                print("RentTime:",record[6])
                print(record[7])
                print("RentTime:",record[8])
                print(record[9])
                print("-"*100)
```

[Figure 4.5 shows Source Code of View Personal Rental History]

The history () function will be displaying all booking transaction in transactionDatabase.txt of the specific IC number. The registered customer could check their booking history by entering their IC Number which they have enter their IC Number when they are booking the car. After configuration, history () function could display booking history of the customer.

6. Car Not Rent Out Function (Registered Customer / Guest)

```
def carNotRentOut():
    print("-"*106)
    print("Available Cars For Rent".center(100))
    print("-"*106)
    lines = []
    with open('carsDatabase.txt', 'r') as carAccountData:
        lines = carAccountData.readlines()
    i = 0
    for line in lines:
        if "Status:available" in line:
            i += 1
            print(f"{i}.car = {line}")
    print("-"*106)
```

[Figure 4.6 shows Source Code of View Detail of Cars to be Rented Out.]

By reading all of the data from carsDatabase.txt, the carNotRentOut() function will display all available cars for the user. The reasoning behind the code is to display all lines if the keywords "Status:available" are detected. The carNotRentOut() functions are similar to the carRentOut() functions, with the exception that it displays all automobiles that are available for rent.

7. Book Car Function (Registered Customer)

```
def bookCar():
    carNotRentOut()
    chooseCar = input("Please enter carID you wish to book: ")
    with open("carsDatabase.txt","r") as carAccountData:
        lines = carAccountData.readlines()
        lines = [line.replace('Price:', '') for line in lines]
        for line in lines:
            record = line.strip("\n").split(" ")
            if "CarID:"+chooseCar == record[0]:
                print("This is the details of your selections:")
                print(record[0])
                print(record[1])
                print(record[2])
                print("Price:",record[3])
                print(record[4])
                print("-"*100)
                print("Welcome to Car Rental Management System (Registered Customer)\n".center(100))
                print("Please fill your information below")
                bookName = input("Please enter your name:")
                idNumber = input("Please enter your IC number:")
                inputDate = input("Please enter a date for renting the car (in format YYYY-MM-DD
HH:MM): ")
                rentDate = datetime.strptime(inputDate, "%Y-%m-%d %H:%M")
                week = int(input("Please enter the total weeks you want to rent the car: "))
                returnDate = rentDate + timedelta(weeks = week)
                day = week * 7

                print("-"*100)
                print("Welcome to Car Rental Management System (Registered Customer)\n".center(100))
                print("This is Your Payment Receipt")
                print(record[0])
                print(record[1])
                print(record[2])
                totalPrice = int(record[3]) * int(day)
                print("TotalPrice:",totalPrice)
                str(record[3])
                print("This is your total car rent duration:")
                print("Total Week Rent:", week, "week")
                print("Total Day Rent:", day, "day")
                print("Your Rent Date and Time:", rentDate)
                print("Your Return Date and Time:", returnDate)
                print("Payment Has Been Made! Thank you!")
                print("-"*106)
```

[Figure 4.7 shows Part of Source Code from Select and Book a car for a specific duration.]

```
            with open('transactionsDatabase.txt', 'a') as transaction:
                    transaction.write(f"BookerName:{bookName} ICNumber:{idNumber} {record[0]}
{record[1]} {record[2]} RentDate:{str(rentDate)} ReturnDate:{str(returnDate)} TotalPrice:
{str(totalPrice)}\n")

    carList= []
    with open("carsDatabase.txt","r") as carFile:
        for line in carFile:
            record = line.strip().split(" ")
            if "CarID:"+chooseCar == record[0]:
                record[4] = "Status:unavailable"
            carList.append(record)

    with open("carsDatabase.txt","w") as carFile:
        i = 0
        while (i < len(carList)):
            newRecord = " ".join(carList[i])
            carFile.write(newRecord+"\n")
            i += 1
```

[Figure 4.7 shows Part of Source Code from Select and Book a car for a specific duration.]

The bookCar() function will be used in the user booking session after the carNotRentOut () method, which prints out all available cars for hire from carsDatabase.txt. Following the selection of a specific desired carID, the program will prompt the user to enter details such as the booking name, ic number, pick-up date, time, and duration. In the interim, the system will determine their return date and time, as well as the overall amount of money they have to pay.

The status of the chosen car will be updated from available to unavailable in the carsDatabase after confirmation is received. This means the car can't be booked by the same or another user until the end of the booking term, unless the administrator returns the car, which we'll discuss in the next step.

8. User Register Function (Unregistered Customer / Guest)

```
def userRegister():
        print("Sign up")
        username = str(input("Please enter Username: "))
        password = input("Please enter Password: ")
        emailAddress = input("Please enter Email: ")
        contactNumber = input("Please enter contact: ")
        open('userDatabase.txt', 'a').close()
        with open("userDatabase.txt","r") as userAccountData:
            lines = userAccountData.readlines()
            for line in lines:
                rec = line.strip("\n").split(" ")
                if "Username:"+username == rec[0] or "Password:"+password == rec[1]:
                    print("account existed")
                    main()
                elif username  == "" or password == "":
                    print("username and password cannot be empty")
                    main()
            with open("userDatabase.txt","a+") as userAccountData:
                userAccountData.write(f"Username:{username} Password:{password} EmailAddress:
{emailAddress} Contact:{contactNumber}\n")
                print("Account succesfully Registered")
```

[Figure 4.8 shows Source Code of User Registration]

The userRegister () function is used to create a new user account before allowing them to login and perform additional tasks. The code for this section collects information such as username, password, email address, and phone number, and then writes those credentials and personal information into userDatabase.txt automatically.

```
        with open("userDatabase.txt","r") as userAccountData:
            lines = userAccountData.readlines()
            for line in lines:
                rec = line.strip("\n").split(" ")
                if "Username:"+username == rec[0] or "Password:"+password == rec[1]:
                    print("account existed")
                    main()
                elif username  == "" or password == "":
                    print("username and password cannot be empty")
                    main()
```

[Figure 4.8 shows Part of Source Code from User Registration]

This code will act as an authentication to prevent users from registering with the same user credentials or providing any empty input that could result in data being incorrect or duplicated.

9. Admin Menu Function (Admin)

```
def adminMenu():
    while True:
        print("-"*106)
        print("Welcome to Car Rental Management System (Administrator/ Menu)\n".center(100))
        print("Please select an option: ")
        print("1. Add Cars to be rented out ")
        print("2. Modify car details")
        print("3. Display all records")
        print("4. Search Specific record of")
        print("5. Return a Rented Car. ")
        print("6. Exit Administration Mode ")
        print("-"*50)
        # userInput Option
        userDecision = int(input("Please enter your option: "))
        if userDecision == 1:
            print("-"*106)
            print("Welcome to Car Rental Management System (Administrator/Add Cars)\n".center(100))
            addCars()
            continue
        elif userDecision == 2:
            print("-"*106)
            print("Welcome to Car Rental Management System (Administrator/Modify Cars)\n".center(100))
            modifyCarDetails()
            continue
        elif userDecision == 3:
            print("-"*106)
            print("Welcome to Car Rental Management System (Administrator/Display Cars
Record)\n".center(100))
            print("1.Cars Rented Out")
            print("2.Cars available for Rent")
            print("3.Customer Bookings and Payments Details")
            print("4.Exit to Admin")
            userDecision = int(input("Please enter your option: "))
```

[Figure 4.9 shows Part of Source Code from Admin Menu]

```
            if userDecision == 1:
                carRentOut()
                continue
            elif userDecision == 2:
                carNotRentOut()
                continue
            elif userDecision == 3:
                cusBookAndPay()
                continue
            elif userDecision == 4:
                continue
            else:
                print("Please select option again")
        elif userDecision == 4:
            print("-"*106)
            print("Welcome to Car Rental Management System (Administrator/Search Cars)\n".center(100)
            print("1.Customer Booking")
            print("2.Customer Payment")
            print("3.Exit to Admin")
            print("-"*106)
            userDecision = int(input("Please enter your option: "))
            if userDecision == 1:
                searchCusBook()
                continue
            elif userDecision == 2:
                searchCusPay()
                continue
            elif userDecision == 3:
                continue
            else:
                print("Please select option again")
        elif userDecision == 5:
            print("-"*106)
            print("Welcome to Car Rental Management System (Administrator/Return Rented
Car)\n".center(100))
            print("-"*106)
            returnRentedCar()
            continue

        elif userDecision == 6:
            return main()
```

[Figure 4.9 shows Part of Source Code from Admin Menu]

adminMenu() are similar to main() and userMenu(), the reason of having this function is to prevent the software system redirect to main after admin activities. Thus, the admin could keep access to admin staff to do admin activities. Once everything is done, admin could exit administration mode to return to main.

10. Add Cars Function (Admin)

```
def addCars():
    carStatus = "available"
    numberOfCars = int(input("Please enter number of cars you wish to add:"))
    # Looping through userInput for car numbers
    for count in range(numberOfCars):
        # userInput
        carID = input("Please enter car ID:")
        modelName = input("Please enter car model name:")
        vehicleType = input("Please enter vehicle type:")
        pricePerDay = input("Please enter price per day:")
        print("-"*106)
        print("carID:", carID)
        print("ModelName:", modelName)
        print("vehicleType:", vehicleType)
        print("PricePerDay:", pricePerDay)
        print("Status:", carStatus)
        print("-"*106)
        with open('carsDatabase.txt', 'a') as cars:
                # Write Data from List to Files
                cars.write(f"CarID:{carID} ModelName:{modelName} VehicleType:{vehicleType} Price:
{str(pricePerDay)} Status:{carStatus}\n")
```

[Figure 4.10 shows Source Code of Add Cars]

The function addCars() will be used to create a new available car record, and the status of all vehicles in this session will be changed to available immediately. General car information, such as carID, model name, vehicle type, and daily fee, will also be required by the software. It will save it to carsDatabase.txt once you've completed all of the fields and information.

11. Modify Car Detail (Admin)

```
def modifyCarDetails():
    while True:
        carList= []
        carNotRentOut()
        with open("carsDatabase.txt","r") as carFile:
            carID = input("Please enter car ID you wish to modify:")
            modify = 0
            for line in carFile:
                record = line.strip().split(" ")
                if "CarID:"+carID == record[0]:
                    print("1.",record[0])
                    print("2.",record[1])
                    print("3.",record[2])
                    print("4.",record[3])
                    inputNum = int(input("Please Enter A Record Number to Modify: "))
                    print("-"*100)
                    print("Dear Admin, Please Modify [EXACTLY BASED ON FORMAT BELOW]:)")
                    print(record[inputNum-1])
                    print("-"*100)
                    print("CarID: (New Values)")
                    print("ModelName:(New Values)")
                    print("VehicleType:(New Values)")
                    print("Price:(New Values)")
                    print("-"*100)
                    if inputNum-1 == 0:
                        new = input("Please Enter New CarID:")
                        stringOne = "CarID:"
                        record[inputNum-1] =  str(stringOne + new)
                    if inputNum-1 == 1:
                        new = input("Please Enter New ModelName:")
                        stringTwo = "ModelName:"
                        record[inputNum-1] =  str(stringTwo + new)
                    if inputNum-1 == 2:
                        new = input("Please Enter New VehicleType:")
                        stringThree = "VehicleType:"
                        record[inputNum-1] =  str(stringThree + new)
                    if inputNum-1 == 3:
                        new = input("Please Enter New Price:")
                        stringFour = "Price:"
                        record[inputNum-1] =  str(stringFour + new)
                    modify = 1
                carList.append(record)
```

[Figure 4.11 shows Part of Source Code from Modification Car Details.]

```
            print(record)
        if modify == 0:
            print("The value not exist")
        else:
            with open("carsDatabase.txt","w") as carFile:
                i = 0
                while (i < len(carList)):
                    newRecord = " ".join(carList[i])
                    carFile.write(newRecord+"\n")
                    i += 1

        remodify = input("Dear Admin, do you wish to remodify? YES ENTER (y), NO ENTER (n):")
        if remodify == "n":
            break
```

[Figure 4.11 shows Source Code of Modification Cars Details.]

In the same way that modifyUser() displays all cars in the carsDatabase.txt file for the user to select their ID, modifyCarDetails() does the same. The admin will first enter the car ID he wishes to modify the car details. The admin will then select the fields in the provided ID they want to change, and the change will be done. It will automatically update carsDatabase.txt and prompt the user to continue or stop the editing session.

12. Car Rented Out Function (Admin)

```
def carRentOut():
    print("-"*106)
    print("Rented Out Cars".center(100))
    print("-"*106)
    lines = []
    with open('carsDatabase.txt', 'r') as carAccountData:
        lines = carAccountData.readlines()
    i = 0
    for line in lines:
        if "Status:unavailable" in line:
            i += 1
            print(f"{i}.car = {line}")
    print("-"*106)
```

[Figure 4.12 shows Source Code of Cars Rent Out]

The carRentOut() function returns a list of all cars that are now unavailable for rental. All automobiles in carsDatabase.txt will be detected by the code, and all cars with an unavailable status will be printed out. Thus, the admin could simply search all the rented out car.

13. Customer Booking and Payment Function (Admin)

```
def cusBookAndPay():
    print("-"*106)
    print("Customer Booking and Payment".center(100))
    print("-"*106)
    lines = []
    with open('transactionsDatabase.txt', 'r') as carAccountData:
        lines = carAccountData.readlines()
    i = 0
    for line in lines:
        i += 1
        print(f"{i}.Booking = {line}")
    print("-"*106)
```

[Figure 4.13 shows Source Code of Customer Booking and Payment Details]

All customer booking and payment details are displayed using the cusBookAndPay() function. The code is straightforward and simply, displaying all of the data contained in transactionDatabase.txt. Thus, the admin could simply search all the customer booking and payment.

14. Search Customer Booking Function (Admin)

```
def searchCusBook():
    print("-"*106)
    print("Search Customer Booking".center(100))
    print("-"*106)
    search = input("Please enter customer name to check booking history: ")
    with open('transactionsDatabase.txt', 'r') as data:
        lines = data.readlines()
        for line in lines:
            record = line.strip("\n").split(" ")
            if "BookerName:"+search == record[0]:
                print("This is the booking history of customer:")
                print(record[0])
                print(record[1])
                print(record[2])
                print(record[3])
                print(record[4])
                print("-"*100)
```

[Figure 4.14 shows Source Code of Search Customer Booking Details]

The searchCusBook() function is used to look for all of the booking information by entering the customer's name. Thus, the admin could simply search the booking information of the customer.

15. Search Customer Payment Function (Admin)

```
def cusBookAndPay():
    print("-"*106)
    print("Customer Booking and Payment".center(100))
    print("-"*106)
    lines = []
    with open('transactionsDatabase.txt', 'r') as carAccountData:
        lines = carAccountData.readlines()
    i = 0
    for line in lines:
        i += 1
        print(f"{i}.Booking = {line}")
    print("-"*106)
```

[Figure 4.15 shows Source Code of Search Customer Payment Details]

The searchCusPay() function is used to look up all of the payment information for a certain client. Thus, the admin could simply search the payment information of specific customer.

16. Return Rented Car Function (Admin)

```
def returnRentedCar():
    carRentOut()
    chooseCar = input("Please enter carID you wish to return: ")
    with open("carsDatabase.txt","r") as carAccountData:
        lines = carAccountData.readlines()
        for line in lines:
            record = line.strip("\n").split(" ")
            if "CarID:"+chooseCar == record[0]:
                print("This is your returned carID:")
                print(record[0])
                print(record[1])
                print(record[2])
                print(record[3])
                print("You have successfully return the car")

    carList= []
    with open("carsDatabase.txt","r") as carFile:
        for line in carFile:
            record = line.strip().split(" ")
            if "CarID:"+chooseCar == record[0]:
                record[4] = "Status:available"
            carList.append(record)

    with open("carsDatabase.txt","w") as carFile:
        i = 0
        while (i < len(carList)):
            newRecord = " ".join(carList[i])
            carFile.write(newRecord+"\n")
            i += 1
```

[Figure 4.16 shows Source Code of Return Rented Car]

The returnRentedCar() function allows admins to manually return rented cars. This is typically done when a customer does not return on time.

The preceding code structure is quite straightforward; basically, the system will display all hired cars, and the administrator will select the precise car ID to return. The status of the chosen car will be changed to available after this process is completed, indicating that the automobile is ready for the next rental.

5.0 Additional Features

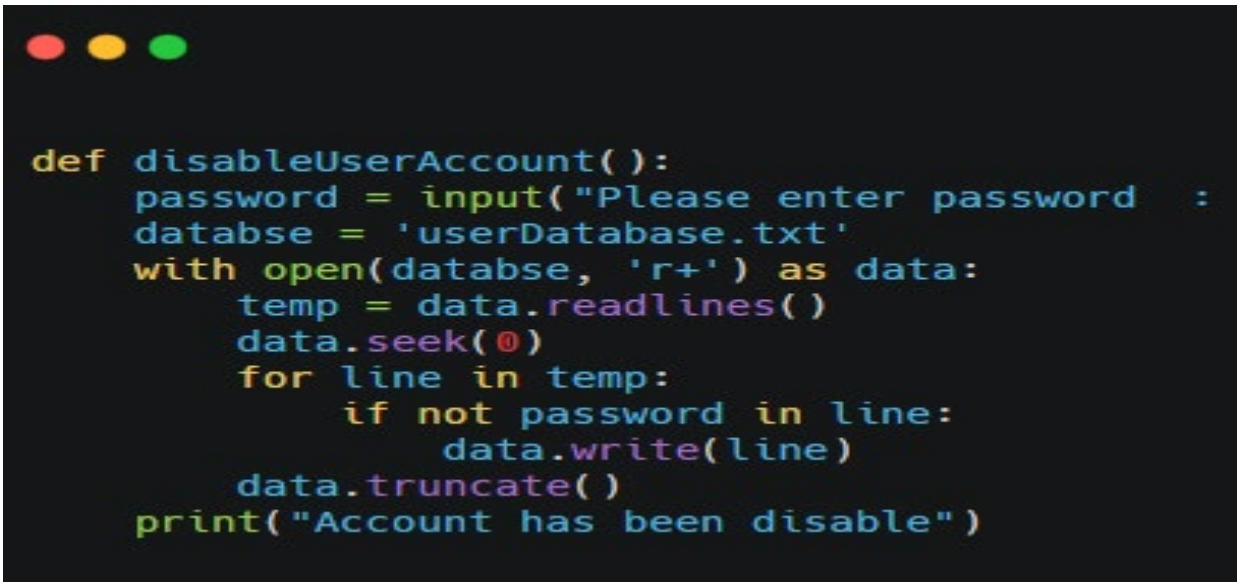
1. Account Recovery Function (Account Recovery for Registered Customer)

```python
def accountRecovery():
    print("Welcome to Car Rental Management System (Account Recovery)\n".center(100))
    # userInput phone number
    searchContact = input("Please enter phone number:")
    # Open file to read data
    with open('userDatabase.txt', 'r') as data:
        # Read every lines of the file into a list
        temp = data.readlines()
        # Check data in file from 0 index to last index
        data.seek(0)
        # Loop through line of list
        for line in temp:
            # Check if searchcontact leave empty
            if searchContact == "":
                print("please enter something")
            # Check if searchContact in list
            if searchContact in line:
                print("-"*106)
                print("This is your account credentials".center(100))
                print("-"*106)
                print(line)
            else:
                print("Please enter correct phone number")
```

[Figure 5.1 shows Source Code of disabled user account.]

accountRecovery() will be used when a user forgets their account credentials. This method will be used as a loop, and the user will be requested to input their phone number, and the software will check his phone number every line, and if it is discovered, it will print all credentials for the line.

2. Disable User Account Function (Registered Customer)



```
def disableUserAccount():
    password = input("Please enter password  :
    databse = 'userDatabase.txt'
    with open(databse, 'r+') as data:
        temp = data.readlines()
        data.seek(0)
        for line in temp:
            if not password in line:
                data.write(line)
        data.truncate()
    print("Account has been disable")
```

[Figure 5.2 shows Source Code of disabled user account.]

The deleteUserAccount() function will be used to delete a user account, including all personal account information and passwords; however, user activity and transactions will not be deleted owing to a data consistency issue.

The code above will prompt the user for their password; if the password matches the password in userDatabase.txt, the account and its data will be deleted using the truncate function.

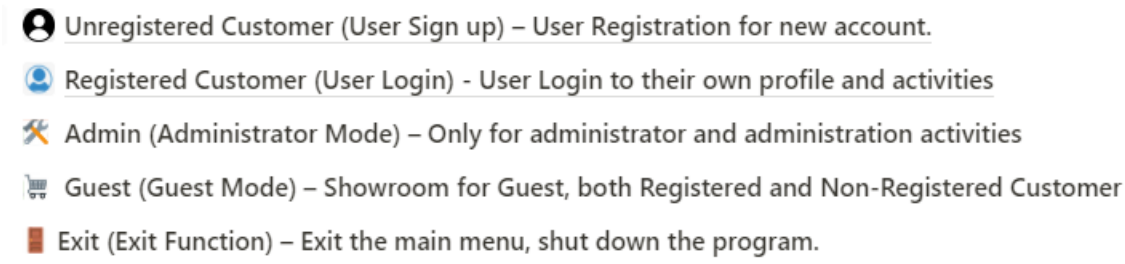
6.0 User Manual Guide

1. User Manual Guide

[Figure 6.1 shows User Menu Car Rental Management System]

1. Execute the python program file which called main.py

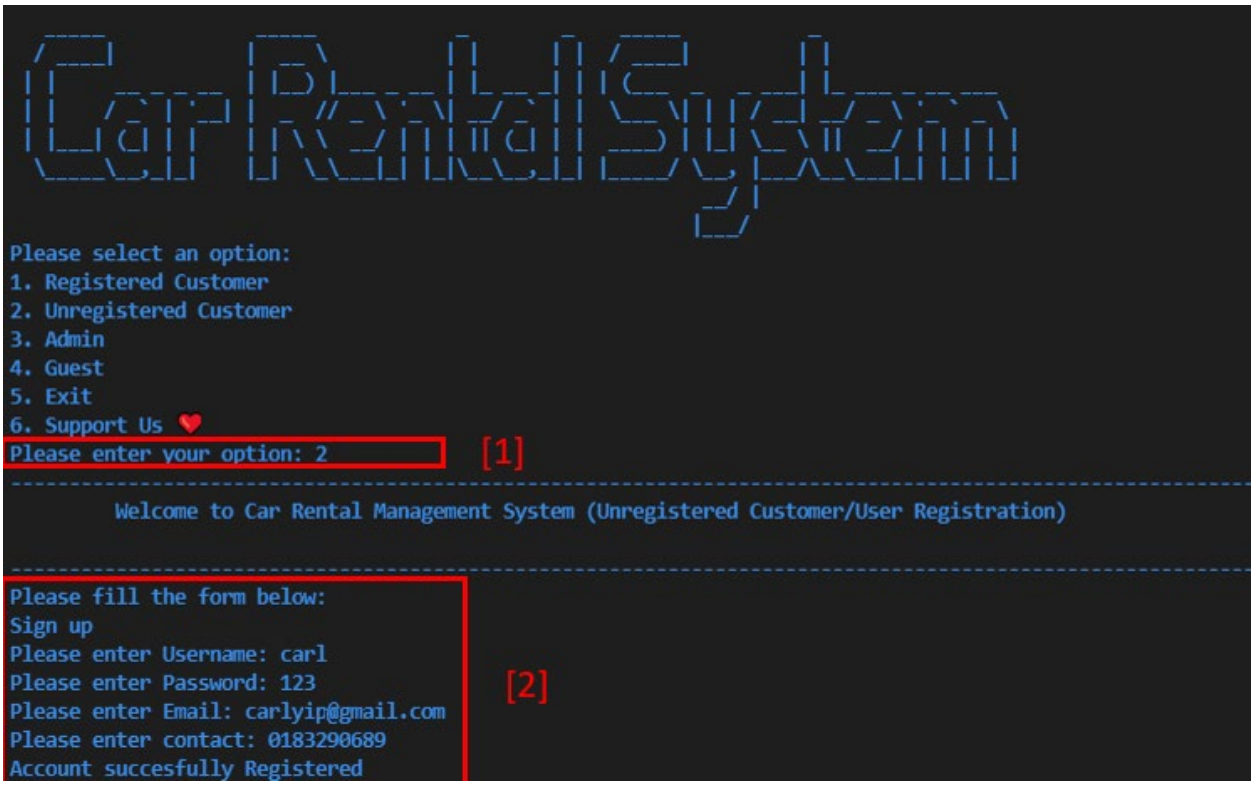
A stunning user menu is displayed after the software is executed. In order to go to the next step, the user will be required to select specified alternatives.



[Figure 6.2 shows Explanation of Menu's Functionalities]

All the user menu's features and selections are explained in detail in Figure 5.2. A new user can always refer to the diagram to familiarize themselves with the system.

2. User Manual Guide – Unregistered Customer

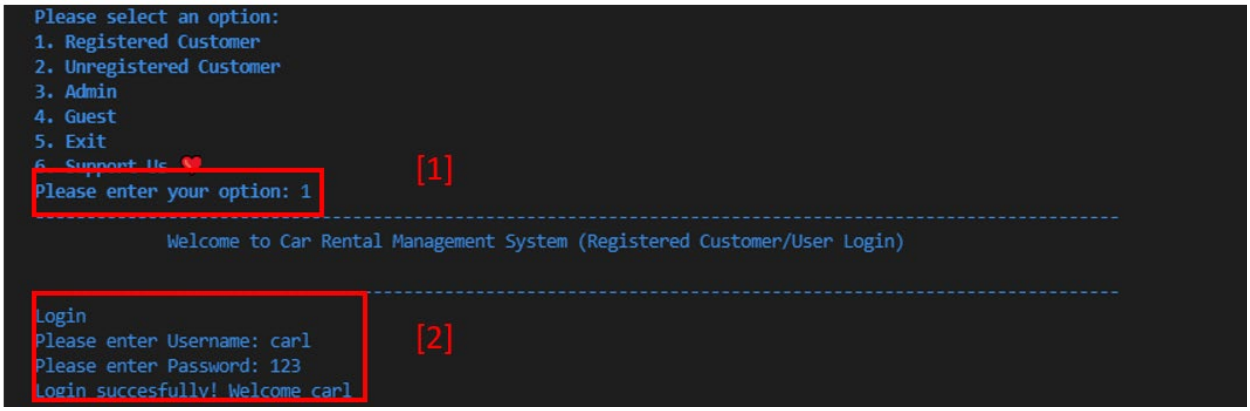


[Figure 6.3 shows Process of User Account Registration]

1. To create a new user account, press 2 to select Unregistered Customer.
2. Please complete all required fields and credentials.

Following the processes outlined above, the system will display Account Successfully Signed, indicating that the account has been created. When the user is ready, he or she can go straight to the login stage.

3. User Manual Guide – Registered Customer

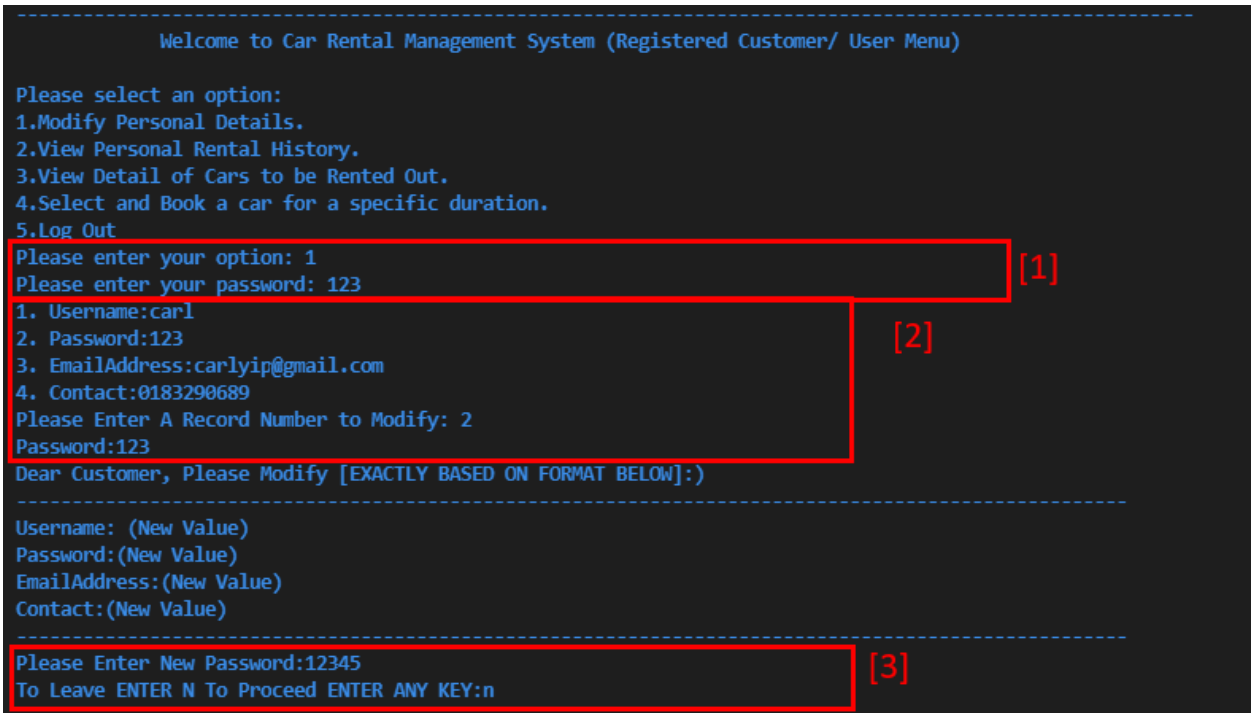


[Figure 6.4 shows Process of User Account Login]

1. To Login as user, press 1 to select Registered Customer.
2. Please complete all required fields and credentials.

Following the previous phases of registration. Our system databases will save user credentials and information. Unless the user disables their account, a registered user will be able to log in at any time.

4. User Manual Guide – Registered Customer (Modify User Details)

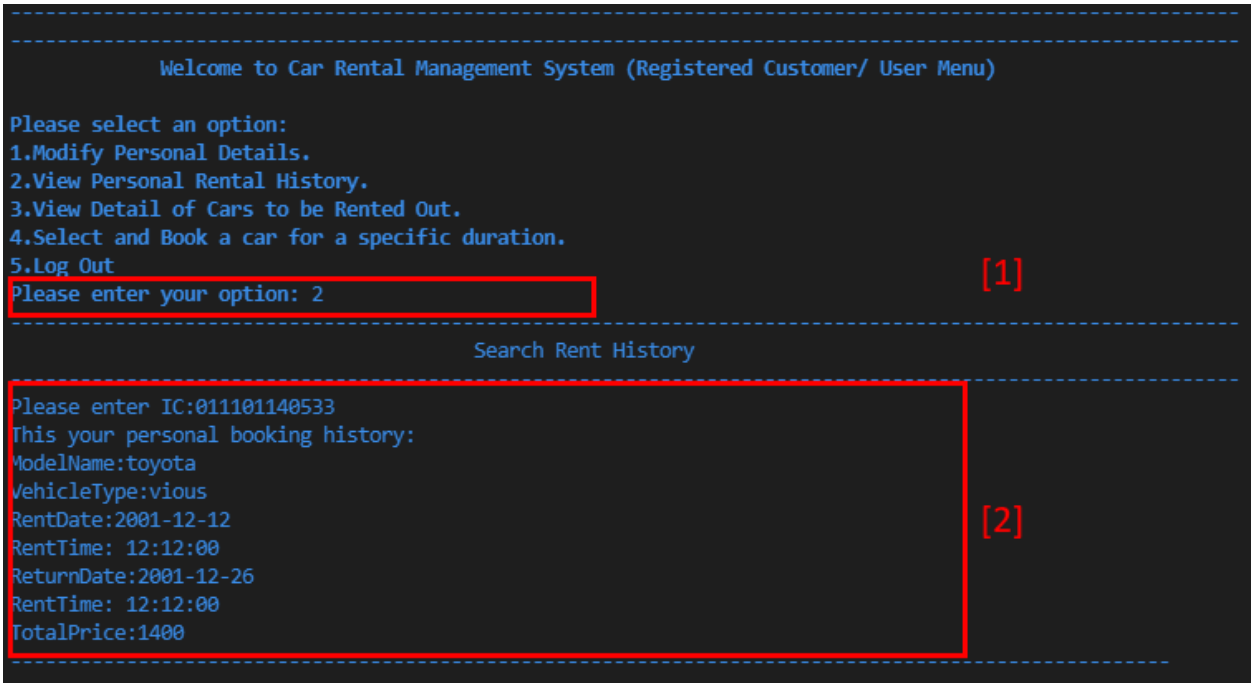


[Figure 6.5 shows Process of Modify User]

- To modify user details, press 1 to select Modify Personal Details.
- User account password is required.
- Please select specific field to modify.
- Require new value for that field.

When dealing with account activities and sensitive data, user account credentials such as username and password are frequently required; the goal is to avoid security issues while protecting user information and privacy.

5. User Manual Guide – Registered Customer (Personal Rental History)

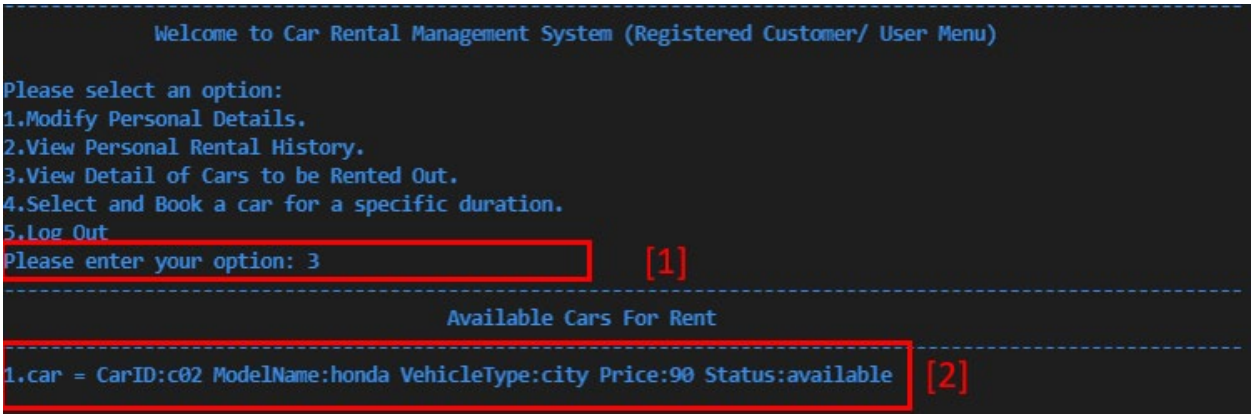


[Figure 6.6 shows Process of View Personal Rental History]

1. To view booking history, press 2 to select View Personal Rental History
2. Require user identity card number that use to book car previously.

The IC number is used for authentication purposes, and once it matches in the database, the user's whole rental history is presented.

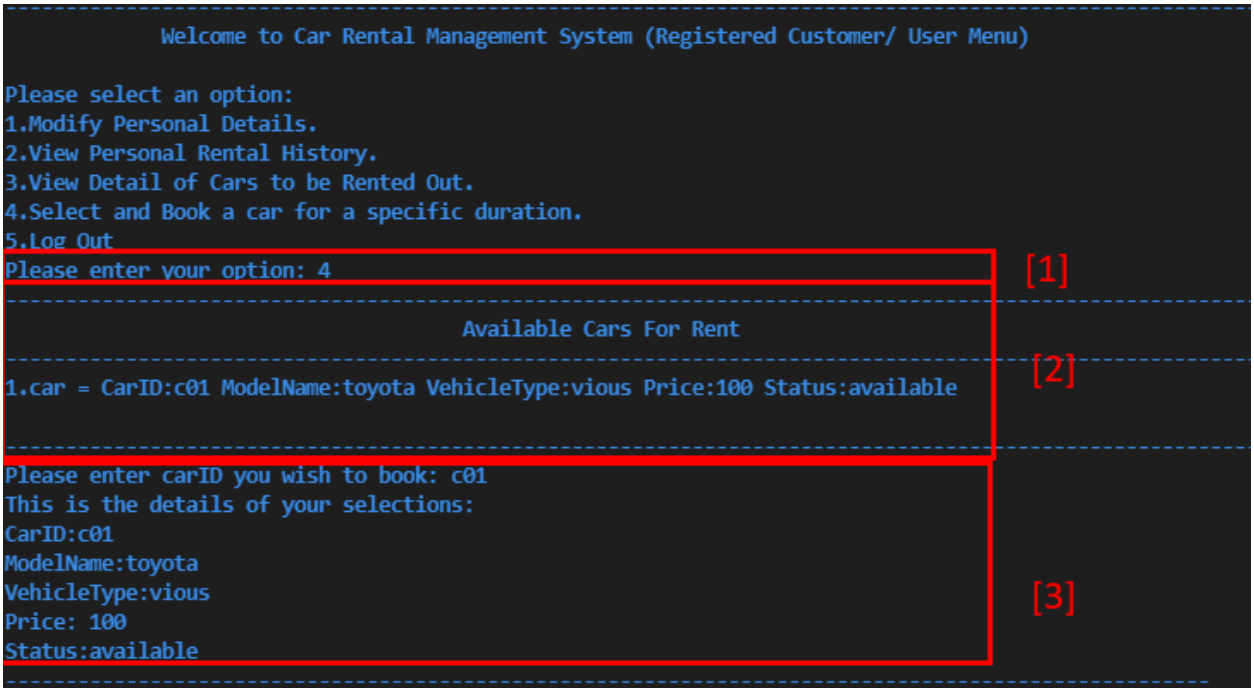
6. User Manual Guide – Registered Customer (Available Cars)

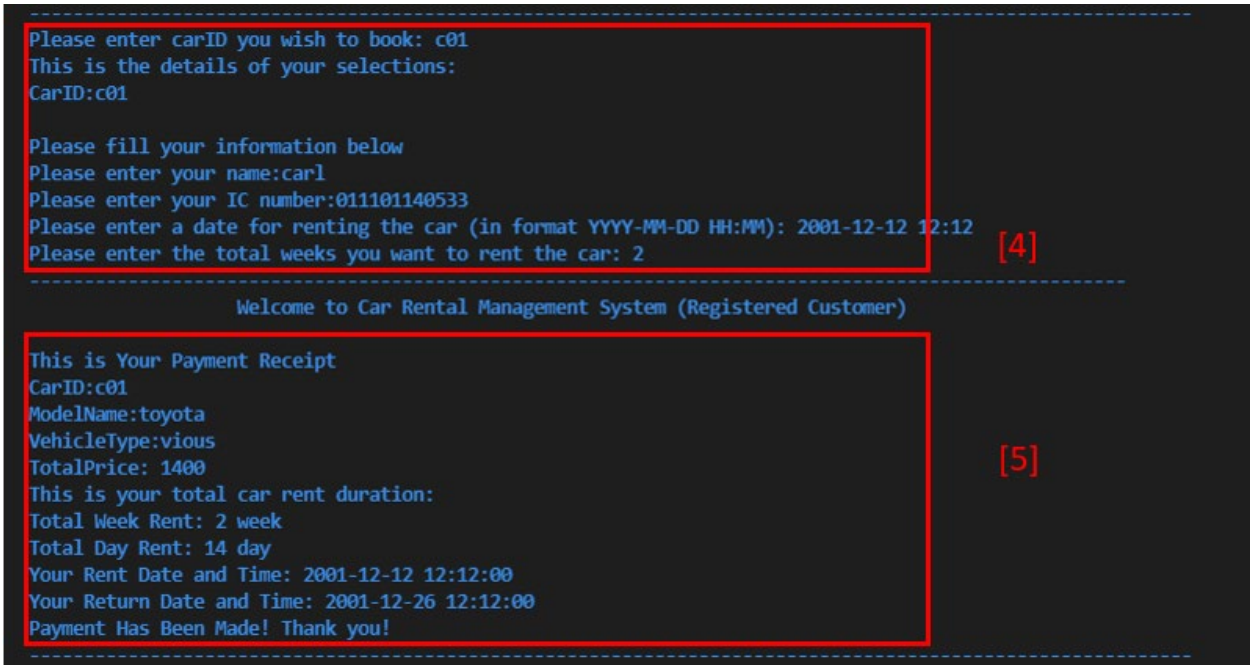


[Figure 6.7 shows Process of View Detail of Cars]

1. To view details of cars rented out, press 3 to select View Detail of Cars to be Rented Out.
2. Display all details of cars rented out.

7. User Manual Guide – Registered Customer (Book Cars)

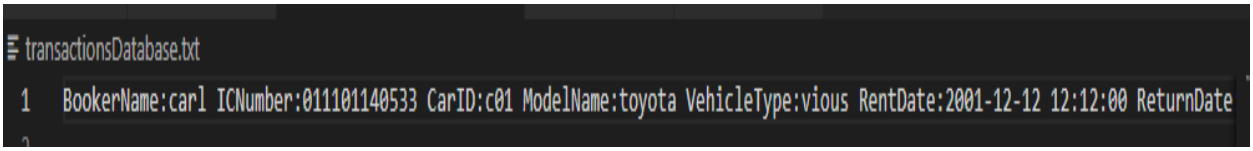




[Figure 6.8 shows Process of Booking]

- To book car, press 4 to select Select and Book a car for specific duration.
- All available car for rent will be display.
- Require user to select desired car ID.
- Require user to fill up information to rent.
- Confirm payment and generate receipt!

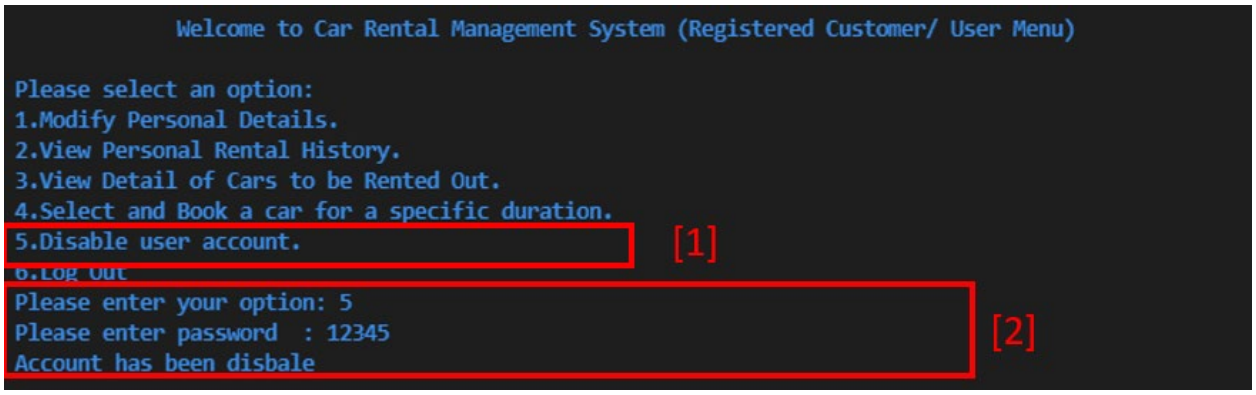
The system will calculate the entire fee based on the user's rental period, pick-up and return dates once the payment has been validated.



[Figure 6.9 shows transactionsDatabase after the process of Add Cars]

In the interim, all the information and transaction data will be saved in transactionsDatabase.txt, and this data will be used in subsequent activities for both the system's administrator and users.

8. User Manual Guide – Registered Customer Additional Features (Disable User Account)

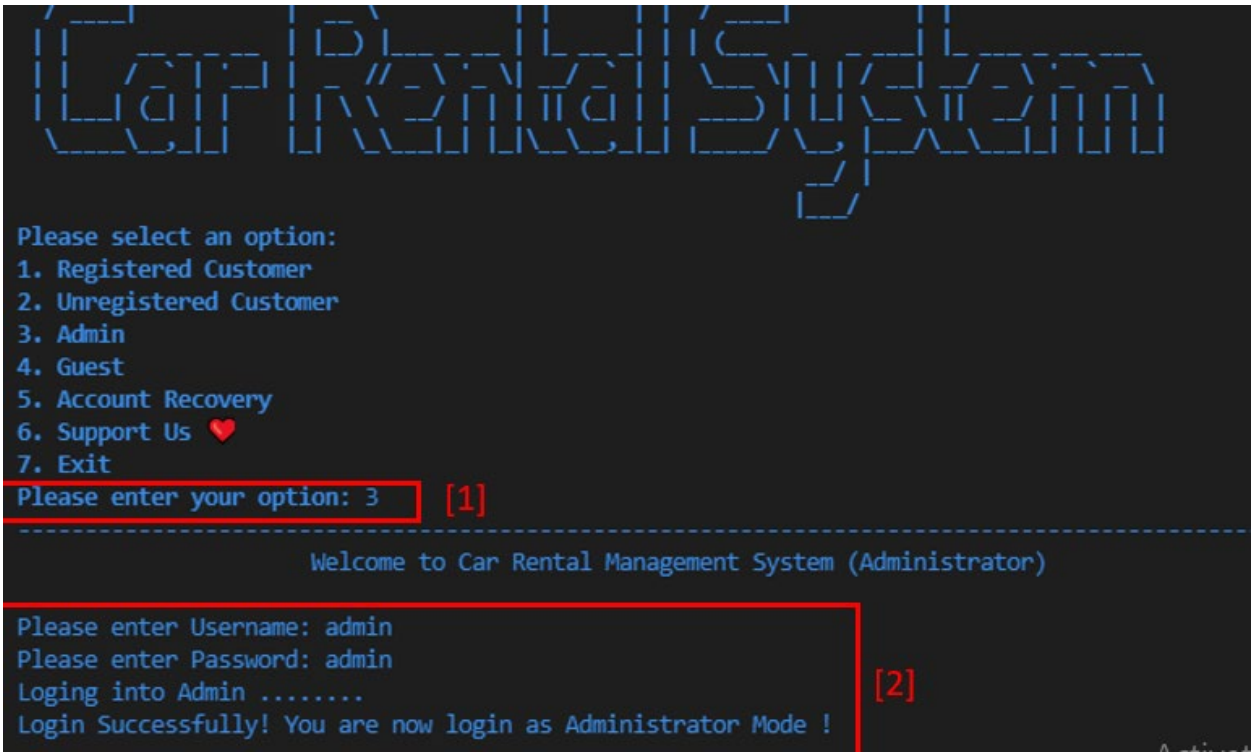


[Figure 6.10 shows Process of disabled user account]

1. To disable account, press 5 to select Disable User Account
2. Required user password for authentication.

When an account is disabled, it is no longer active; consequently, if the user changes their mind, they must create a new account on the sign-up page.

9. User Manual Guide – Administrator



[Figure 6.11 shows Process of Administrator Login]

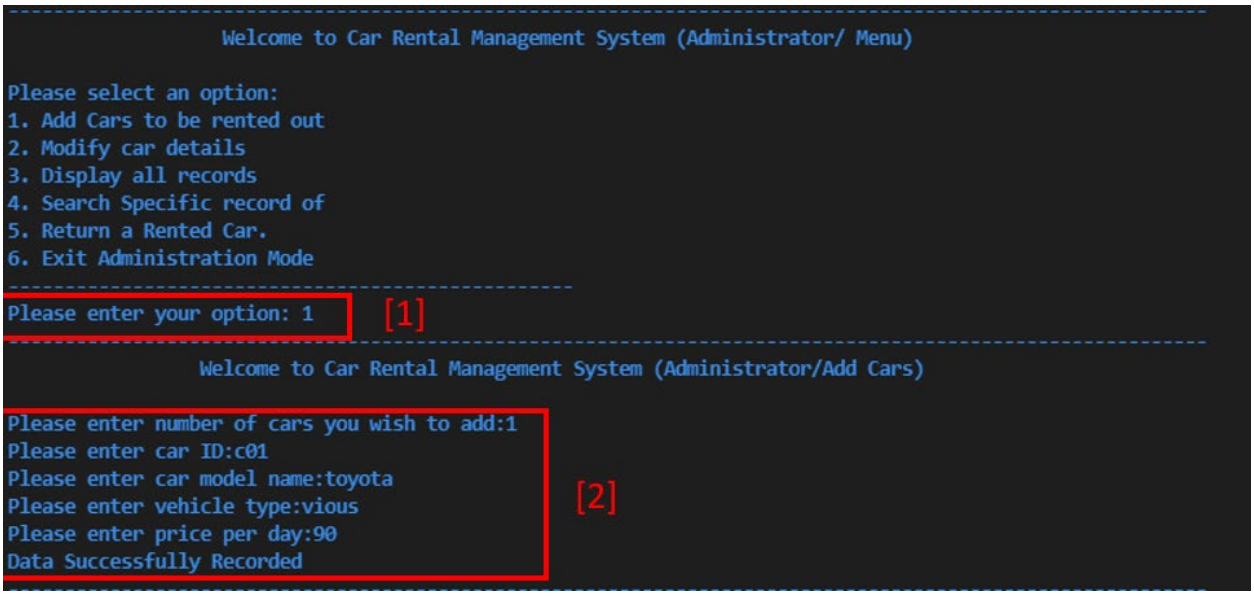
1. To Login as admin, press 3 to select Admin.
2. Please complete all required fields and credentials.

Admin Username	Admin Password
admin	admin

[Table 6.1 shows Credentials of Administration Login]

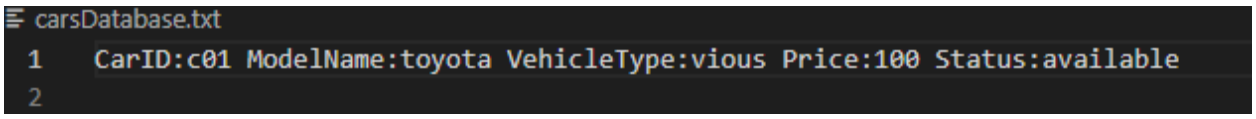
During the development phase, our system created default credentials for both the login and password for admin. As a result, neither the admin nor the user has the ability to modify the credentials to enter administrator mode unless they contact the developer.

10. User Manual Guide – Administrator (Add Cars)



[Figure 6.12 shows Process of Add Cars]

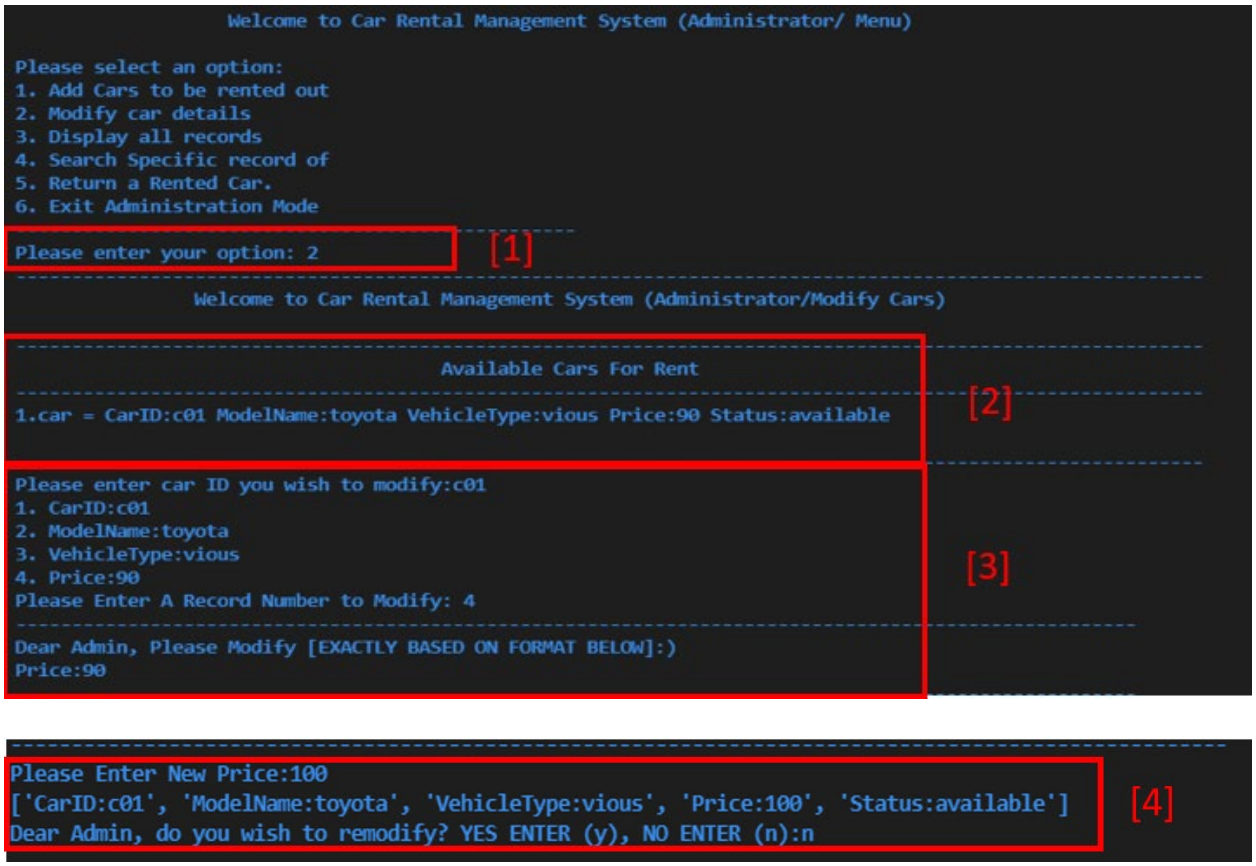
1. To add new cars, press 1 to select Add Cars to be rented out.
2. Please complete all required fields and information's.



[Figure 6.13 shows carsDatabase after the process of Add Cars]

Once the record has been entered to our system databases, positive notifications will be presented. The data and information about the car will be saved in our carsDatabase.txt file; these details are required and will be utilized for other purposes.

11. User Manual Guide – Administrator (Modify Cars)

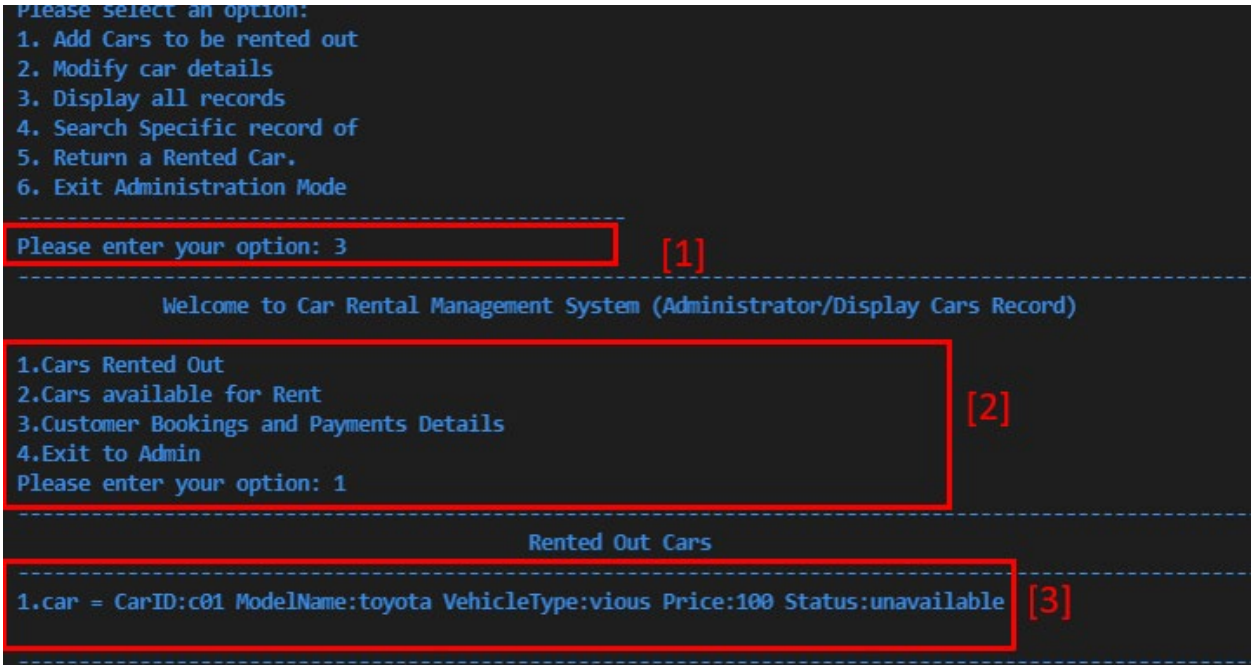


[Figure 6.14 shows Process of Modification Cars Details]

1. To modify cars details, press 2 to select Modify Cars Details.
2. Displaying all current cars details in the database
3. Displaying cars details of the specific car ID chosen by user
1. Required user to choose specific field to modify.
4. Required new value for specific field chosen by user.

Following the processes outlined above, the system will edit or update a specific field in a certain ID, and the updated information will be written out, indicating that the record has been updated.

12. User Manual Guide – Administrator (Display Records)



[Figure 6.15 shows Process of Display Records]

1. To display records, press 3 to select Display All Records.
2. Select display options.
3. Display records of the selections

❌ Car Rented Out - Display all unavailable or rented cars

🚘 Cars Available for Rent - Display all available cars

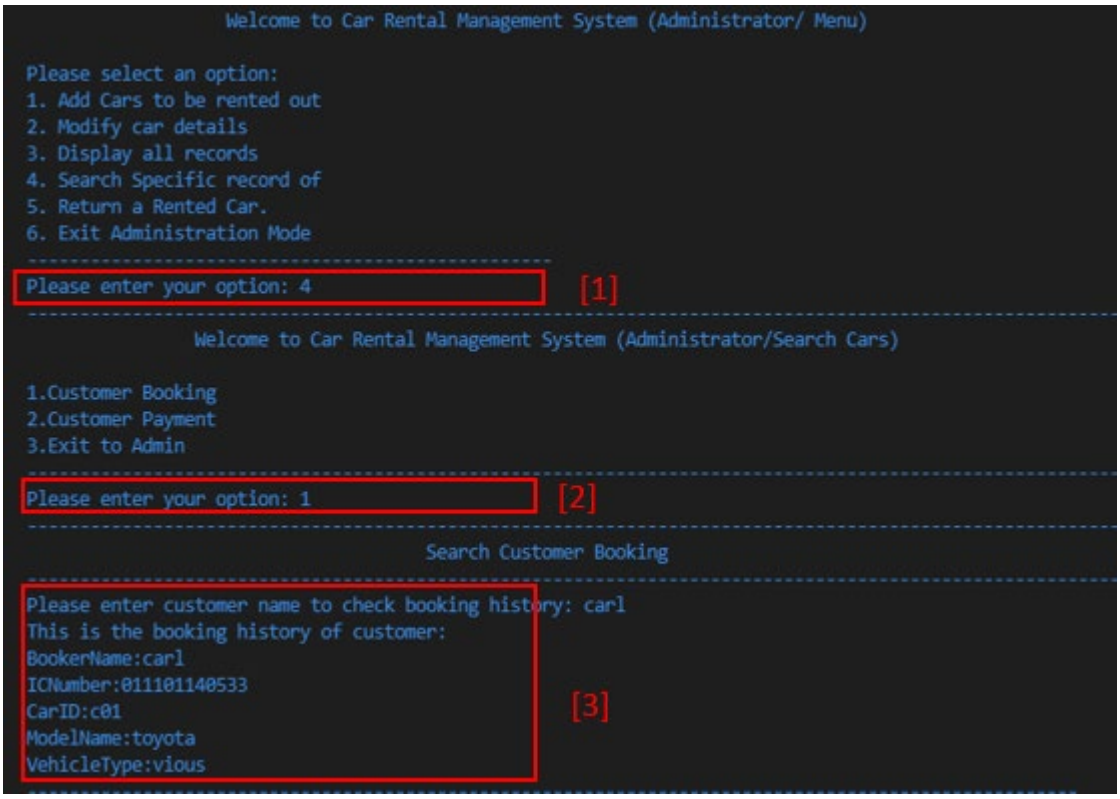
$ Customer Booking & Payment - display all transactions

🚪 Exit (Exit Function) – Exit the Display options, return to admin dashboard.

[Figure 6.16 shows Explanation of Admin Display Menu's Functionalities]

All the admin display menu's features and selections are explained in detail in Figure 5.16. A new user can always refer to the diagram to familiarize themselves with the system.

13. User Manual Guide – Administrator (Search Records)



[Figure 6.17 shows Process of Search Specific Record]

1. To search specific record, press 4 to select Search Specific Record.
2. Press 1 to search customer booking.
3. Display searching record by username

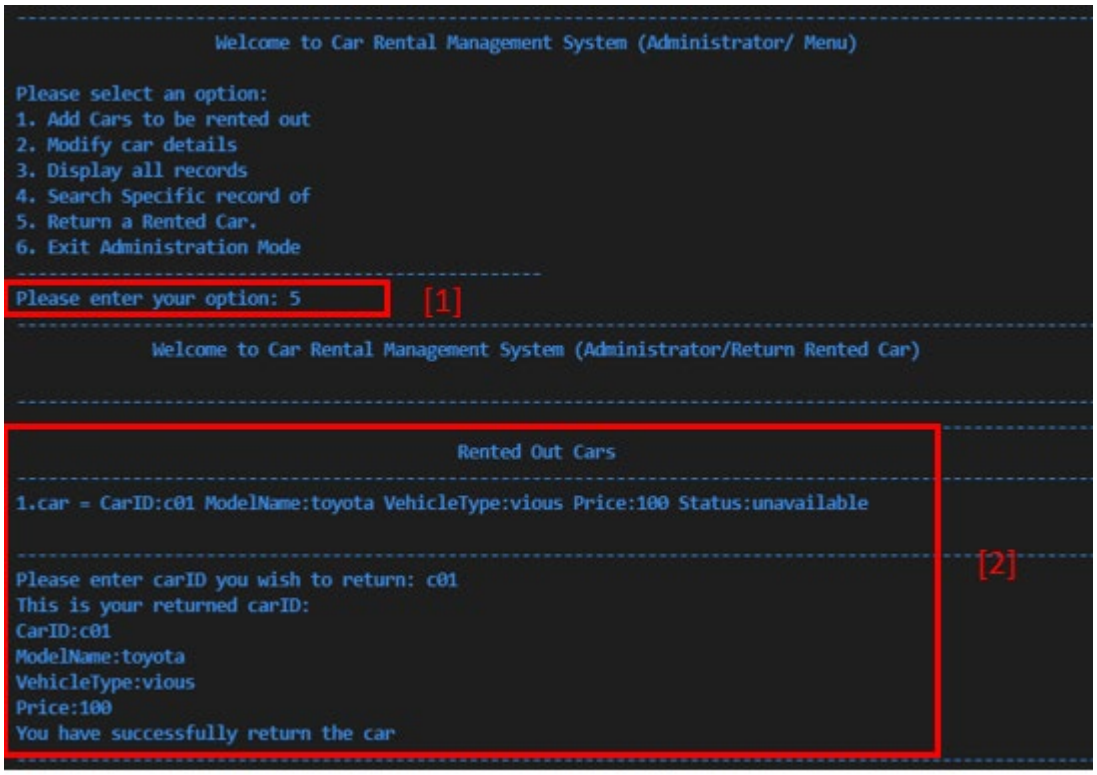
Customer Booking - Display IC number, Booker Name, Car ID, Model, Vehicle Type

$ Customer Payment - Display IC number, Booker Name, Pick & Return Date, Total Price

[Figure 6.18 shows Explanation of Admin Search Menu's Functionalities]

All the admin display menu's features and selections are explained in detail in Figure 5.18. A new user can always refer to the diagram to familiarize themselves with the system.

14. User Manual Guide – Administrator (Return Cars)



[Figure 6.19 shows Process of Return Cars]

1. To return rented cars, press 5 to select Return a rented Car.
2. Select CarID to return

```
CarID:c01 ModelName:toyota VehicleType:vious Price:100 Status:available
CarID:c02 ModelName:honda VehicleType:city Price:90 Status:unavailable
```

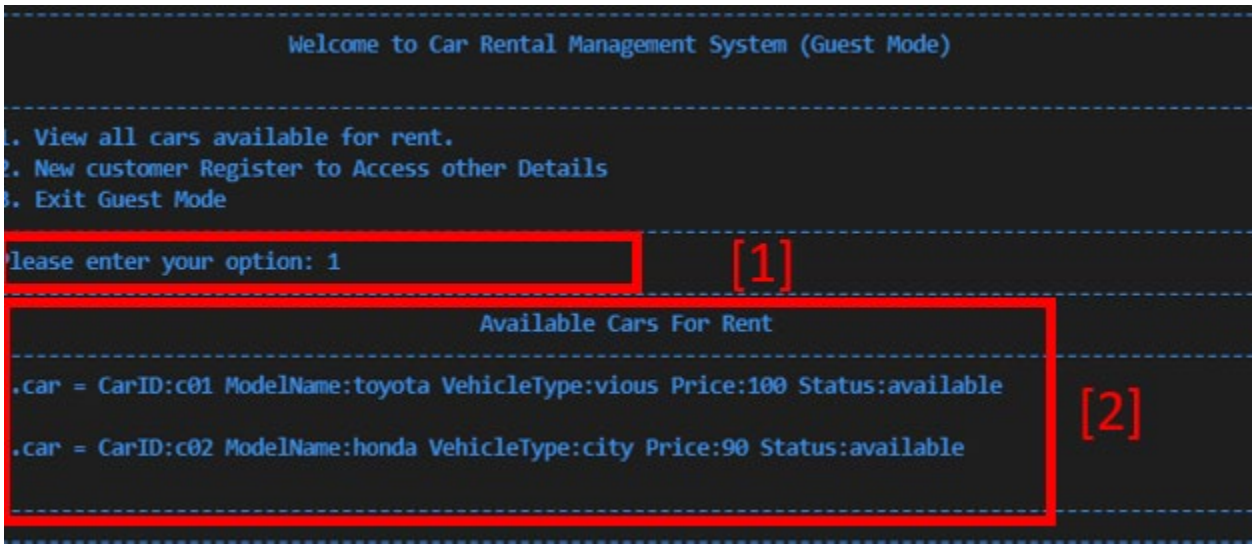
[Figure 6.20 shows carsDatabase.txt before Process Figure 5.19]

carsDatabase.txt

```
CarID:c01 ModelName:toyota VehicleType:vious Price:100 Status:available
CarID:c02 ModelName:honda VehicleType:city Price:90 Status:available
```

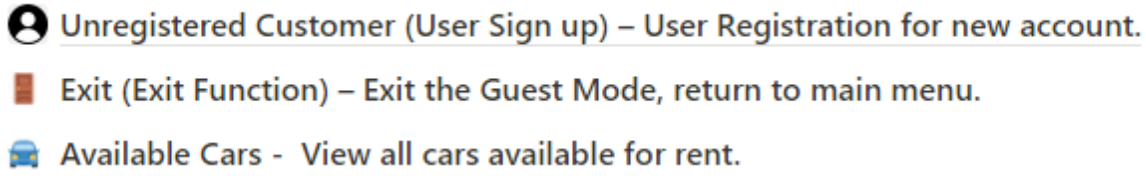
[Figure 6.21 shows carsDatabase.txt after Process in Figure 5.19]

15. User Manual Guide – Guest Mode (View All)



[Figure 6.22 shows Display all Available Car in Guest Mode]

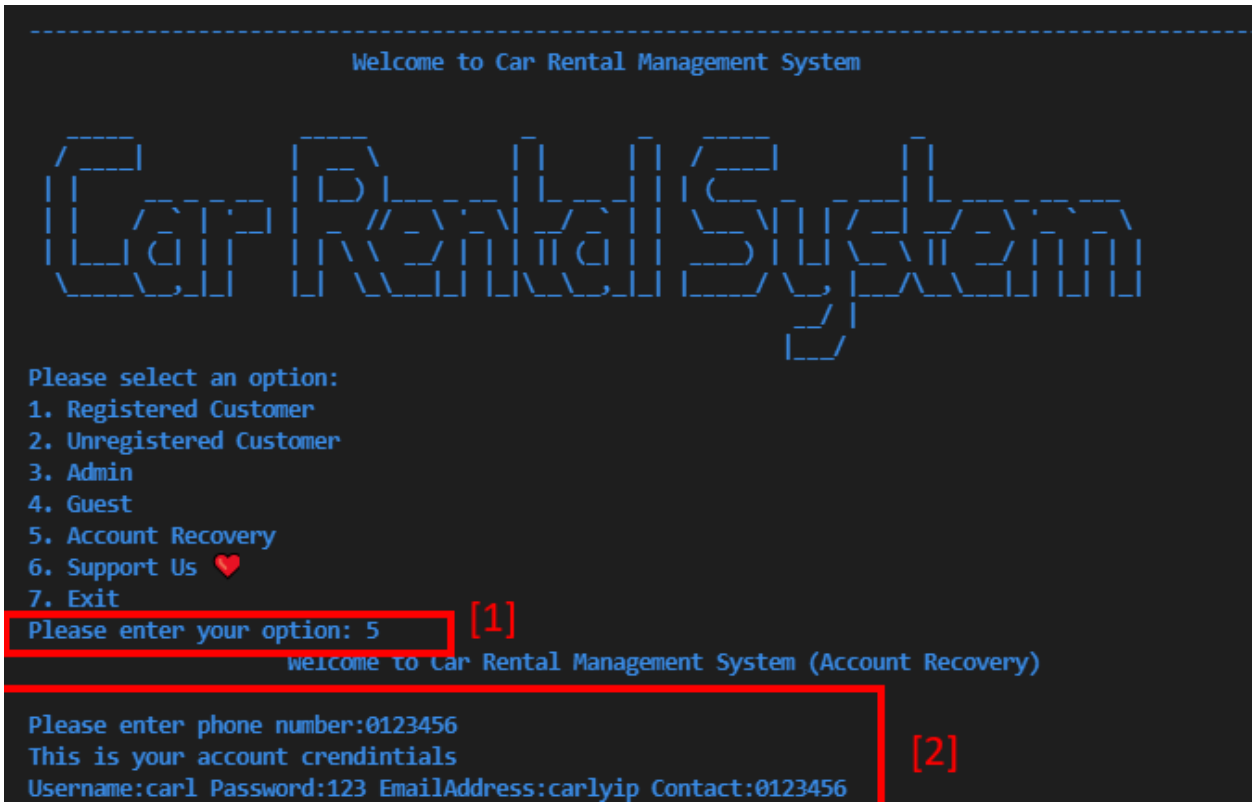
1. To view all available, press 1 to select View All Cars Available for Rent
2. Display all available cars.



[Figure 6.23 shows Explanation of Guest Mode Functionalities]

All the guest menu's features and selections are explained in detail in Figure 5.23. A new user can always refer to the diagram to familiarize themselves with the system.

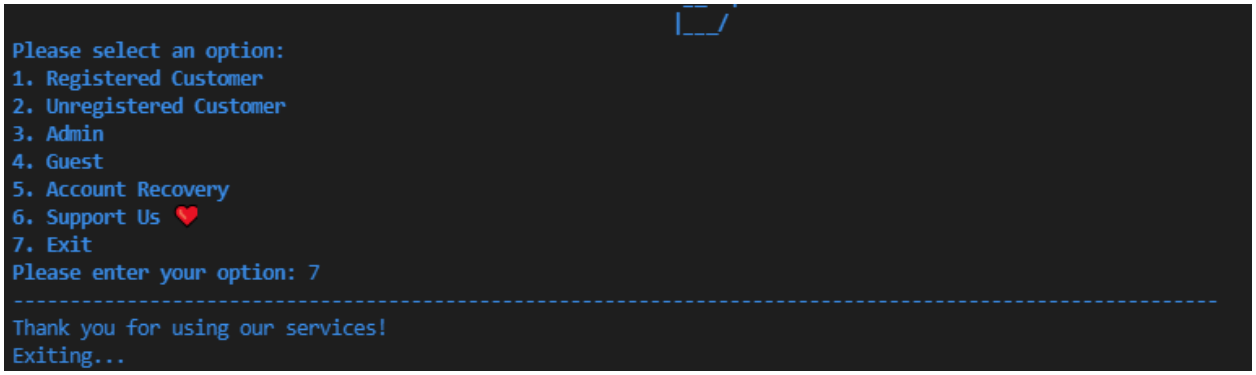
16. User Manual Guide – Account Recovery



[Figure 6.24 shows Process of Account Recovery]

1. To recover password credentials, please select 5
2. Require phone number that used in sign up

Because humans frequently forget account credentials and passwords, we developed an additional function called account recovery, which acts as a lookup tool for credentials using a phone number.



[Figure 6.25 shows Process of Account Recovery]

Once everything has done, to exit the program, kindly press 7. Thank you for using our services.

7.0 Conclusion

In conclusion, the car rental system that we have created in this documentation is highly functional and user friendly, and we believe that the well-developed end software products and documentation with explanation will be beneficial to both customers and administrators in any activities performed while using the system. We're not simply making software; we're making the future.

8.0 Workload Matrix

Name	TP NUMBER	Flowchart	Pseudocode	Python	Documentation
YIP KAR FAI	TP060711	50%	50%	50%	50%
NG LI SHENG	TP060612	50%	50%	50%	50%

9.0 References

Boris & Lother, 2019. *How to delete a specific line in a file?.* [Online]
Available at: https://stackoverflow.com/questions/4710067/how-to-delete-a-specific-line-in-a-file
[Accessed 20 5 2021].

GeeksforGeeks, 2018. *Python program to print Emojis.* [Online]
Available at: https://www.geeksforgeeks.org/python-program-to-print-emojis/
[Accessed 10 5 2021].