ASSIGNMENT
TECHNICAL REPORT

COURSE CODE	:	**WIX1002**
COURSE NAME	:	**FUNDAMENTALS OF PROGRAMMING**
LECTURER	:	**ASSOC. PROF. DR. MOHAMAD NIZAM BIN AYUB**
SEMESTER & SESSION	:	**SEMESTER 1, 2023/2024**
OCC	:	**OCC 1**
GROUP NAME	:	**CODE CRAFTERS (GROUP 8)**
TOPIC	:	**TOPIC 7: LECARS SALES MANAGEMENT SYSTEM**

NO.	NAME	MATRIC NO.
1	NG SEAN SEAN	23004914/1
2	YONG XIN TEEN	23004942/1
3	LAVINE SEE YU TIAN	23005231/1
4	TAN CHIEN LING	23004992/1
5	ELYSIA YUNG SHI XI	23004935/1

TABLE OF CONTENTS

1. INTRODUCTION TO LECARS SALES MANAGEMENT SYSTEM

Like many others, we've encountered the frustration of dealing with a buggy system, and the MAYA system is no exception. Since the establishment of our LeCars Company in June 2023, we took the initiative to address the challenges by assembling a team of skilled programmers to develop a management system. While the initial design showcased an appealing aesthetic, it became evident that crucial features were lacking, and frequent bugs plagued its functionality. The codebase was found to be disorganized, impeding new development efforts. Recognizing the need for a robust solution, the management has decided to scrap the current system and embark on a comprehensive revamp. Fortunately, we plan to preserve existing sales logs for seamless integration into the new system.

2. REQUIREMENTS OF LECARS SALES MANAGEMENT SYSTEM

There are several fundamental features that should be presented in LeCars Sales Management System.

a. Login and Registration

- Allow new user to register as sales-level employee by providing user ID and password.
- Allow existing user to log in with existing credentials.
- The program still be able to run after logging out and user can log in with the same/another user without terminating the program.
- The new user is required to input a secret key while registering, which if fails to do so, ends the registration process and terminates the program.
- Store users' credentials offline.

b. Import Data

- Import customer, sales, vehicle and employee data.
- Allow management level employee to add more Management level employee.

c. Entering New Data

Both sales and management level employees can enter:

i. Customer data:

- Customer ID is automatically incrementally generated.
- Customer Name
- Phone Number
- Postcode

ii. Sales data:

- Sales ID is automatically incrementally generated.
- Date/Time is taken using a method in Java library.
- Car Plate
- Customer ID
- Employee ID is automatically filled in as current logged in user.

iii. Vehicle data:

- Car number plate
- Car model
- Acquired price
- Car status (Boolean: 0 is sold, 1 is in stock)
- Sold price (Can be null)

d. View Info

i. Sales employee can view:

- Own customer data
- Own sales records
- All vehicle data

ii. Management employee can view:

- All customer data
- All sales records
- All vehicle data
- All employee data

And retrieve current logs and display them in a tabular form.

e. Additional Features

i. Sales Insight

- Management employees can:
  - calculate and view the total of annual sales.
  - calculate and view the total and average of monthly sales.
  - view the number of cases in each month in line chart.

ii. *Salary Calculation*

- Management level employees are able to calculate the salary of employees.
- Sales Employee:
  - Basic Salary: RM1200
  - Allowance: Up to RM250
  - Base Commission: 1% of car sales price
  - Total Salary Formula: Basic + Allowance + Commission
- Management Employee:
  - Basic Salary: RM2200
  - Allowance: Up to RM350
  - Base Commission: 1% of car sales price
  - Total Salary Formula: Basic + Allowance + Commission

iii. Employee Bonus

- Management level employees are able to calculate the bonus of employees.
- For sales employees:
    - RM500 bonus will be given to employees who manage to have more than 15 car sales records, or more than RM1 million sales amount within a calendar month.
- For Management employees,

Sales Amount (RM)	Comm %
800000.00 or less	1.00
800000.01 – 1600000.00	1.15
1600000.01 – 2500000.00	1.25
2500000.01 or more	1.35

iv. Search and Filter

- Both sales and management level employees can search / apply filters for:
    - Customer record
    - Employees record
    - Sales record
    - Vehicle record

- Sales employees can:
    - search for their own customers and sales records.
    - management employees can:
    - search for every record and also car sales with prices of certain range.

3. APPROACH TAKEN TO SOLVE THE TASK

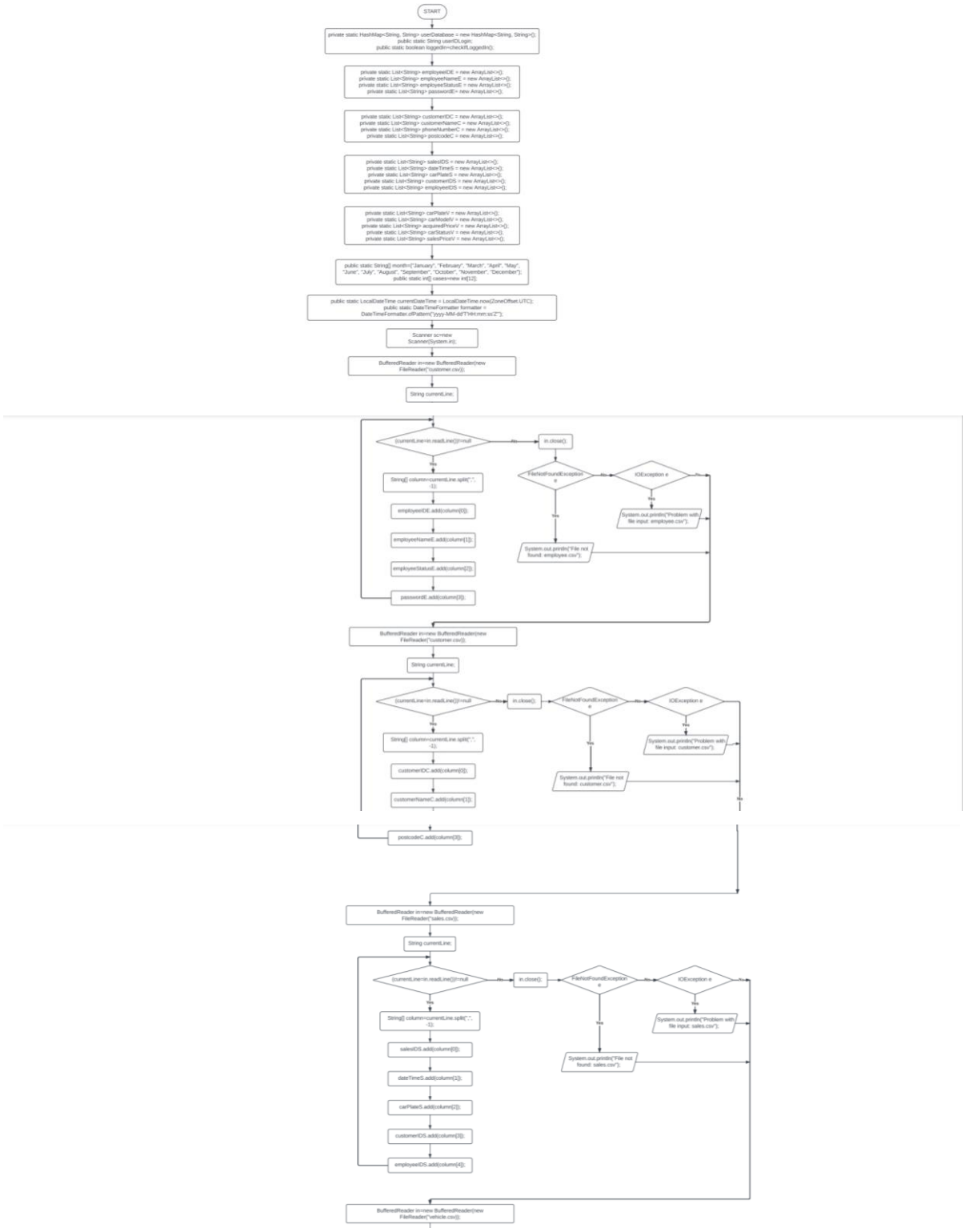
Our team developed a strategic plan, featuring a structured schedule and weekly goals, to ensure the project's success. Regular online meetings and discussions facilitated clear communication, fostering a shared understanding of roles and tasks. Online resources, including YouTube tutorials and forums, were employed to enhance our team's skills collaboratively.

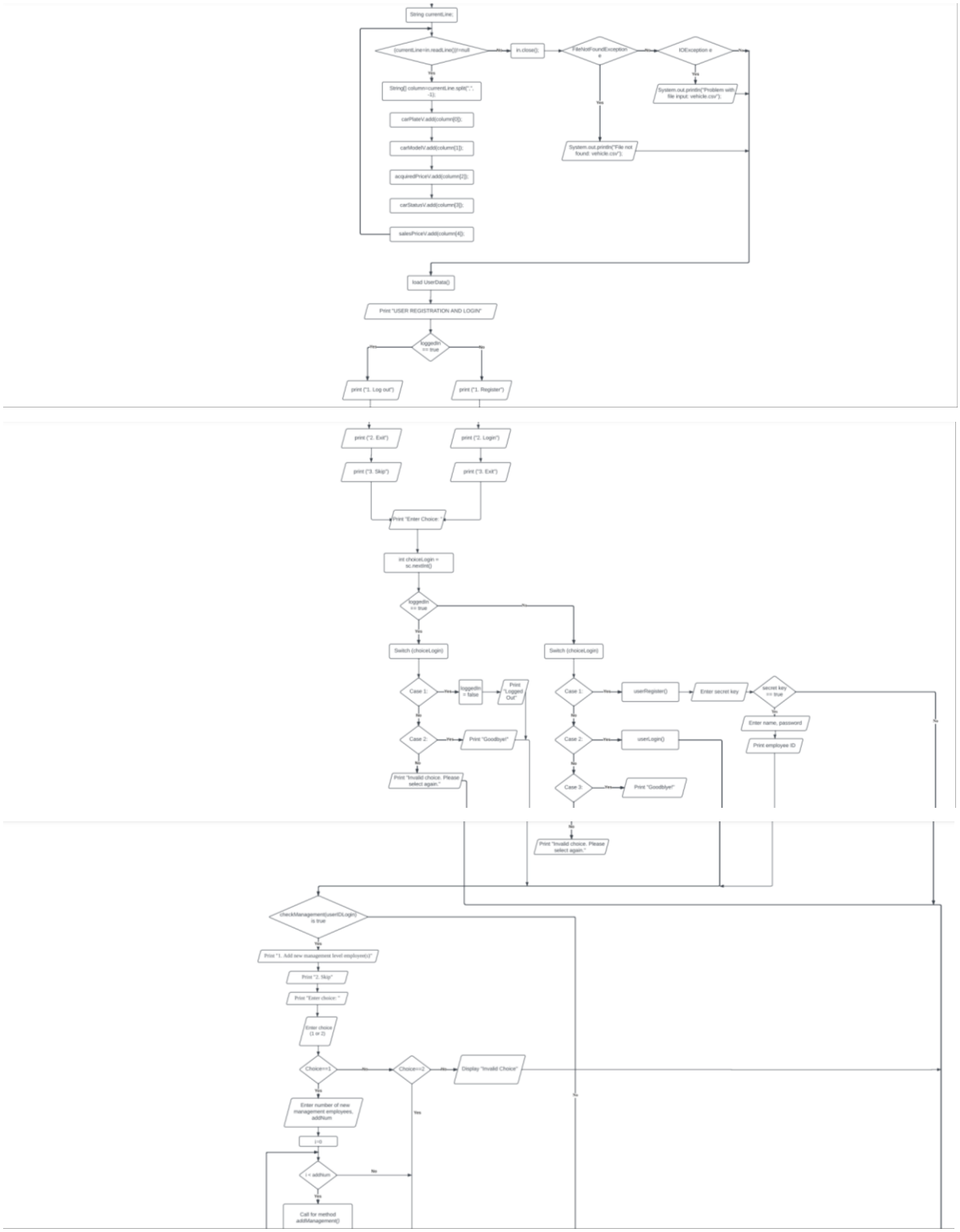
The implementation phase commenced with meticulous algorithm planning and structured coding. Adhering to the restriction on traditional databases, we opted for CSV files for data storage. Embracing File Input/Output and Object-Oriented Programming (OOP) principles, our goal was to create a robust, modular solution.

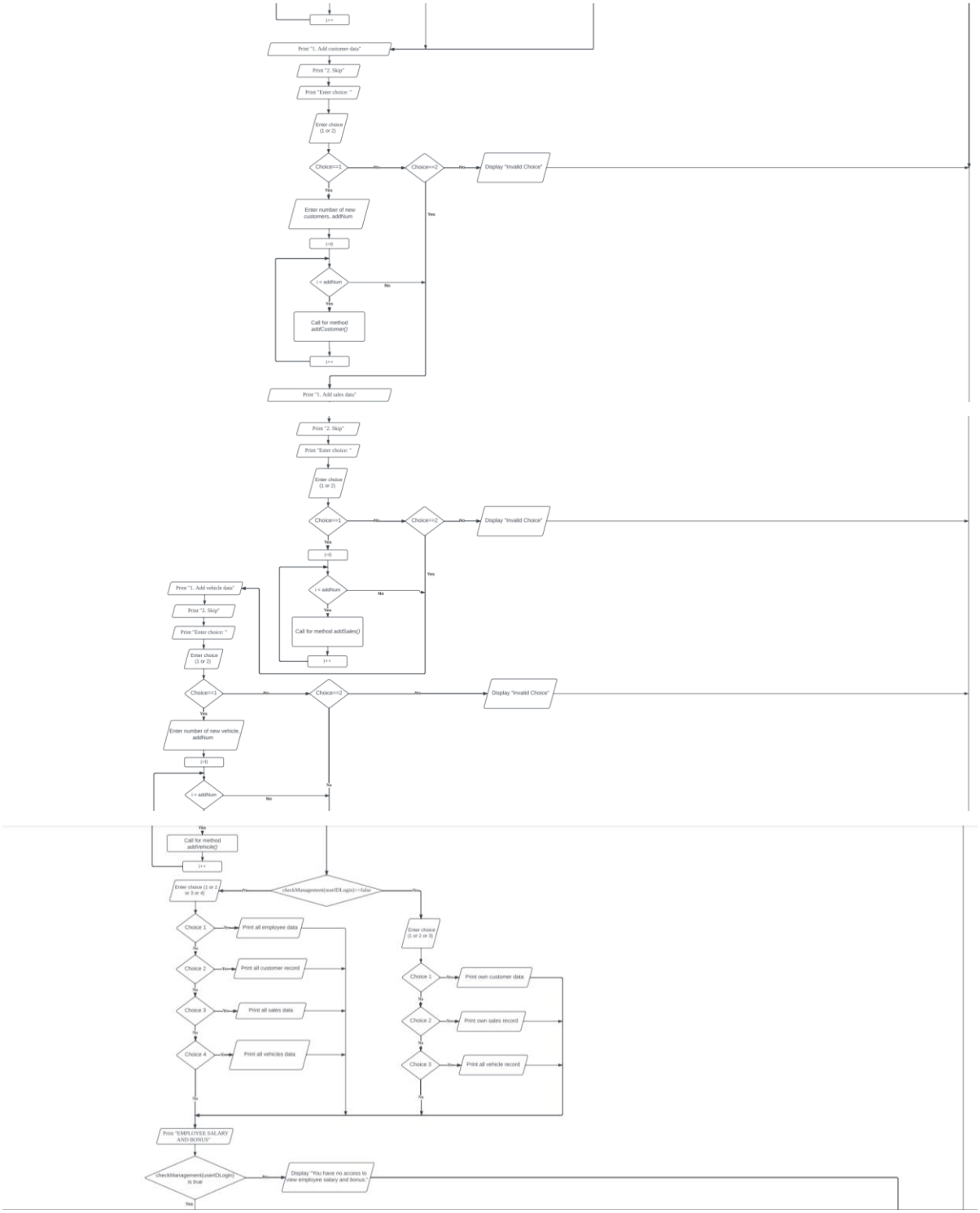
Rigorous testing throughout the development lifecycle identified and rectified potential errors. This iterative process ensured program reliability, allowing us to fine-tune the code for desired outcomes. Adopting a disciplined and systematic approach, our team successfully overcame project challenges, delivering a solution that met specified criteria and included additional features.

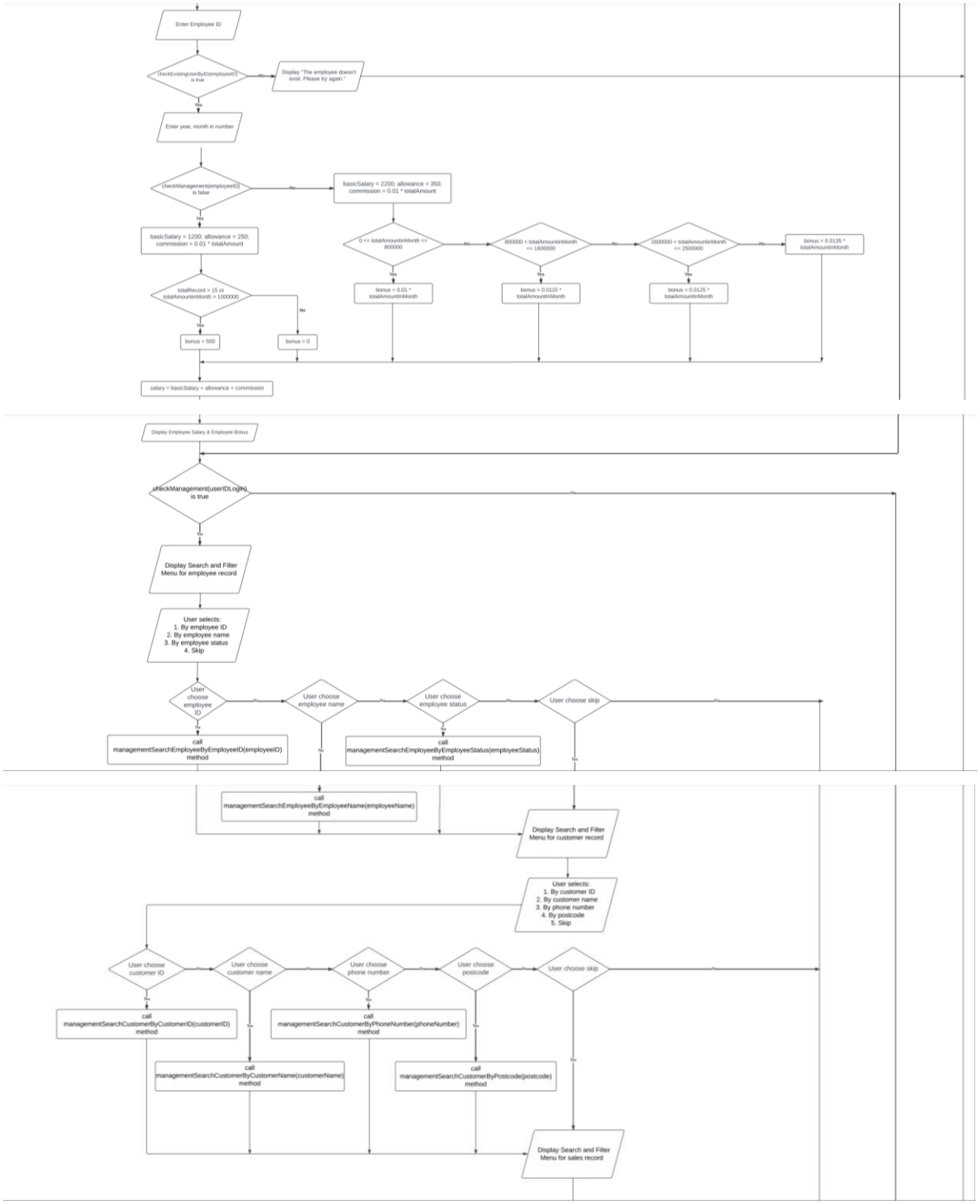
4. DETAILED DESCRIPTION OF SOLUTION

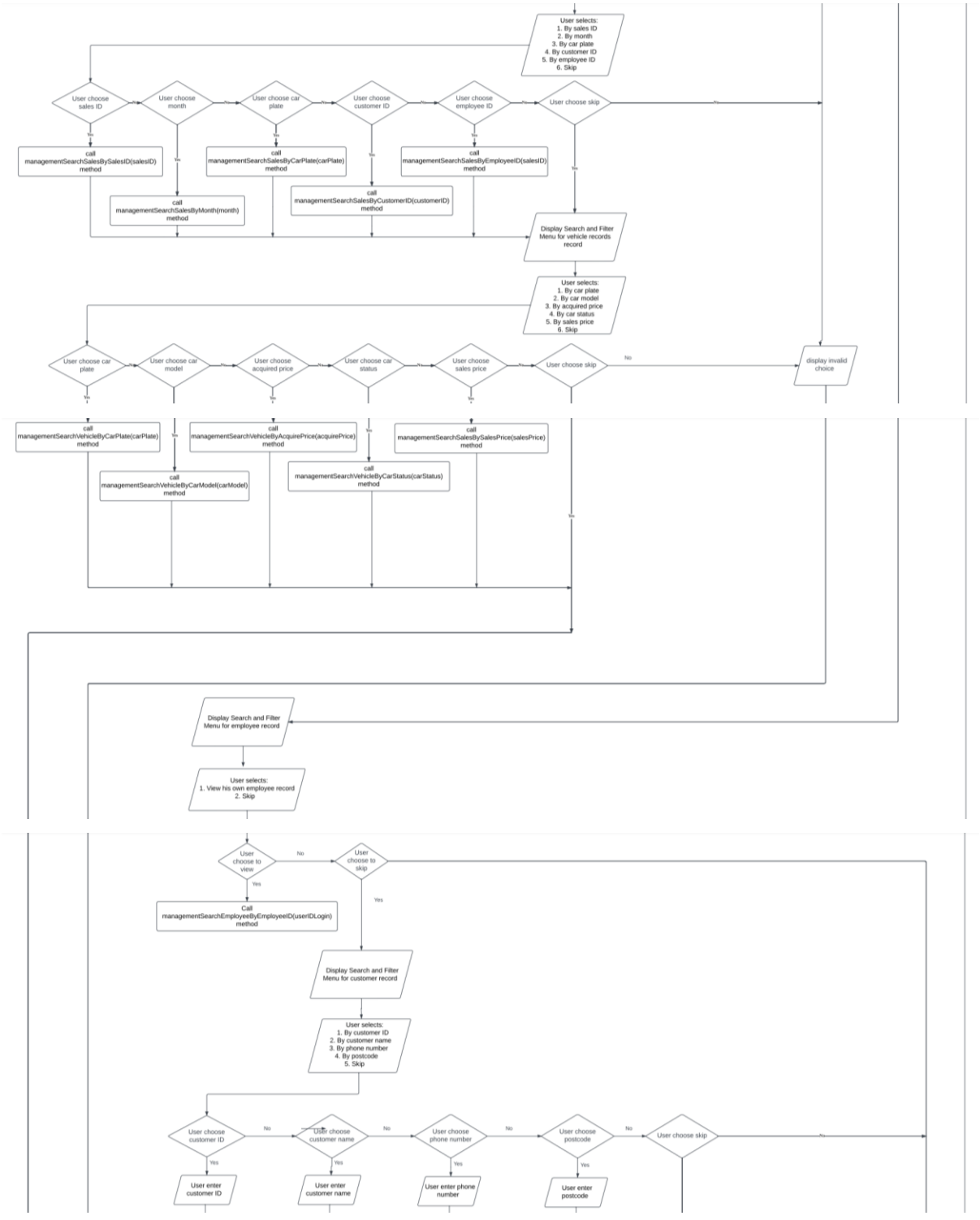
FLOWCHART:

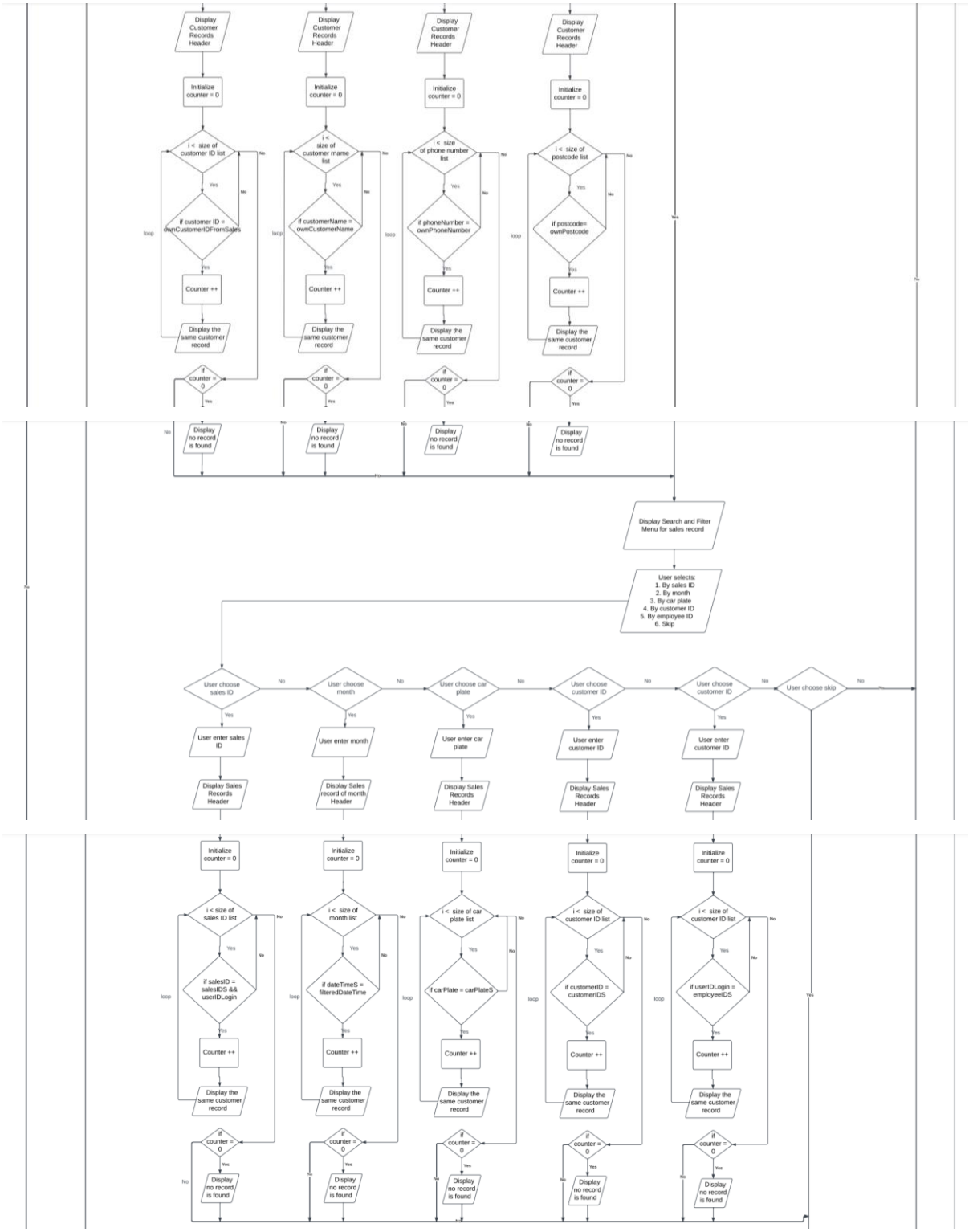


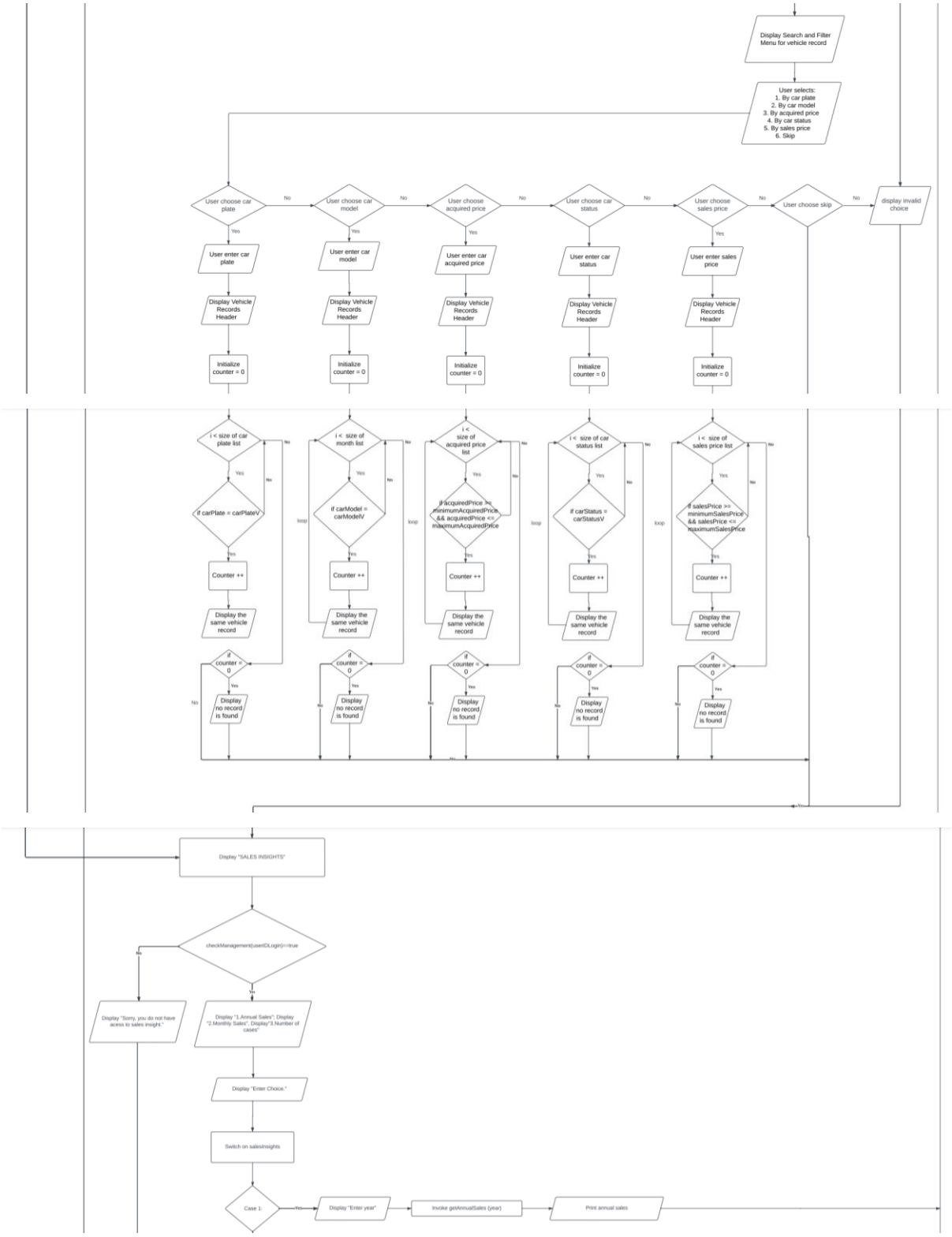

String currentLine;
(currentLine=in.readLine())!=null
in.close();
FileNotFoundException e
IOException e
String[] column=currentLine.split(",", -1);
carPlateV.add(column[0]);
carModelV.add(column[1]);
acquiredPriceV.add(column[2]);
carStatusV.add(column[3]);
salesPriceV.add(column[4]);
System.out.println("Problem with file input: vehicle.csv");
System.out.println("File not found: vehicle.csv");
load UserData()
Print "USER REGISTRATION AND LOGIN"
loggedIn == true
print ("1. Log out")
print ("1. Register")
print ("2. Exit")
print ("2. Login")
print ("3. Skip")
print ("3. Exit")
Print "Enter Choice: "
int choiceLogin = sc.nextInt()
loggedIn == true
Switch (choiceLogin)
Case 1:
loggedIn = false
Print "Logged Out"
Case 2:
Print "Goodbye!"
Print "Invalid choice. Please select again."
userRegister()
Enter secret key
secret key == true
Enter name, password
Print employee ID
userLogin()
Case 3:
Print "Goodbye!"
checkManagement(userIDLogin) is true
Print "1. Add new management level employee(s)"
Print "2. Skip"
Print "Enter choice: "
Enter choice (1 or 2)
Choice==1
Choice==2
Display "Invalid Choice"
Enter number of new management employees, addNum
i=0
i < addNum
Call for method addManagement()

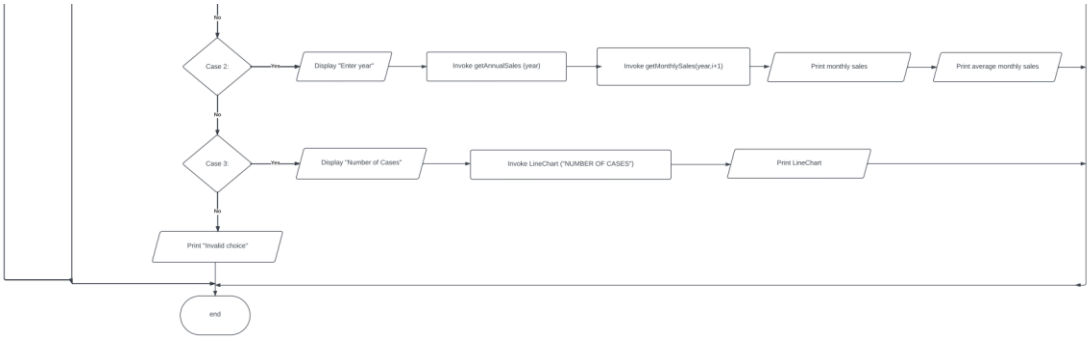

i++
Print "1. Add customer data"
Print "2. Skip"
Print "Enter choice: "
Enter choice (1 or 2)
Choice==1
No
Choice==2
No
Display "Invalid Choice"
Yes
Enter number of new customers, addNum
i=0
i < addNum
No
Yes
Call for method addCustomer()
i++
Yes
Print "1. Add sales data"
Print "2. Skip"
Print "Enter choice: "
Enter choice (1 or 2)
Choice==1
No
Choice==2
No
Display "Invalid Choice"
Yes
i=0
i < addNum
No
Yes
Call for method addSales()
i++
Yes
Print "1. Add vehicle data"
Print "2. Skip"
Print "Enter choice: "
Enter choice (1 or 2)
Choice==1
No
Choice==2
No
Display "Invalid Choice"
Yes
Enter number of new vehicle, addNum
i=0
i < addNum
No
Yes
Call for method addVehicle()
i++
Enter choice (1 or 2 or 3 or 4)
No
checkManagement(userIDLogin)==false
Yes
Choice 1
Yes
Print all employee data
No
Choice 2
Yes
Print all customer record
No
Choice 3
Yes
Print all sales data
No
Choice 4
Yes
Print all vehicles data
No
Enter choice (1 or 2 or 3)
Choice 1
Yes
Print own customer data
No
Choice 2
Yes
Print own sales record
No
Choice 3
Yes
Print all vehicle record
No
Print "EMPLOYEE SALARY AND BONUS"
checkManagement(userIDLogin) is true
No
Display "You have no access to view employee salary and bonus."
Yes


Enter Employee ID
checkExistingUserByEmployeeID() is true
No
Display "The employee doesn't exist. Please try again."
Yes
Enter year, month in number
checkManagement(employeeID) is false
No
basicSalary = 2200; allowance = 350; commission = 0.01 * totalAmount
Yes
basicSalary = 1200; allowance = 250; commission = 0.01 * totalAmount
0 <= totalAmountInMonth <= 800000
No
800000 < totalAmountInMonth <= 1600000
No
1600000 < totalAmountInMonth <= 2500000
No
bonus = 0.0135 * totalAmountInMonth
Yes
bonus = 0.01 * totalAmountInMonth
Yes
bonus = 0.0115 * totalAmountInMonth
Yes
bonus = 0.0125 * totalAmountInMonth
totalRecord > 15 or totalAmountInMonth > 1000000
No
Yes
bonus = 500
bonus = 0
salary = basicSalary + allowance + commission
Display Employee Salary & Employee Bonus
checkManagement(userIDLogin) is true
No
Yes
Display Search and Filter Menu for employee record
User selects:
1. By employee ID
2. By employee name
3. By employee status
4. Skip
User choose employee ID
No
User choose employee name
No
User choose employee status
No
User choose skip
No
Yes
call managementSearchEmployeeByEmployeeID(employeeID) method
No
Yes
call managementSearchEmployeeByEmployeeStatus(employeeStatus) method
Yes
call managementSearchEmployeeByEmployeeName(employeeName) method
Display Search and Filter Menu for customer record
User selects:
1. By customer ID
2. By customer name
3. By phone number
4. By postcode
5. Skip
User choose customer ID
No
User choose customer name
No
User choose phone number
No
User choose postcode
No
User choose skip
No
Yes
call managementSearchCustomerByCustomerID(customerID) method
Yes
call managementSearchCustomerByPhoneNumber(phoneNumber) method
Yes
call managementSearchCustomerByCustomerName(customerName) method
Yes
call managementSearchCustomerByPostcode(postcode) method
Yes
Display Search and Filter Menu for sales record


User selects:
1. By sales ID
2. By month
3. By car plate
4. By customer ID
5. By employee ID
6. Skip
User choose sales ID
User choose month
User choose car plate
User choose customer ID
User choose employee ID
User choose skip
call managementSearchSalesBySalesID(salesID) method
call managementSearchSalesByMonth(month) method
call managementSearchSalesByCarPlate(carPlate) method
call managementSearchSalesByCustomerID(customerID) method
call managementSearchSalesByEmployeeID(salesID) method
Display Search and Filter Menu for vehicle records record
User selects:
1. By car plate
2. By car model
3. By acquired price
4. By car status
5. By sales price
6. Skip
User choose car plate
User choose car model
User choose acquired price
User choose car status
User choose sales price
User choose skip
display invalid choice
call managementSearchVehicleByCarPlate(carPlate) method
call managementSearchVehicleByCarModel(carModel) method
call managementSearchVehicleByAcquirePrice(acquirePrice) method
call managementSearchVehicleByCarStatus(carStatus) method
call managementSearchSalesBySalesPrice(salesPrice) method
Display Search and Filter Menu for employee record
User selects:
1. View his own employee record
2. Skip
User choose to view
User choose to skip
Call managementSearchEmployeeByEmployeeID(userIDLogin) method
Display Search and Filter Menu for customer record
User selects:
1. By customer ID
2. By customer name
3. By phone number
4. By postcode
5. Skip
User choose customer ID
User choose customer name
User choose phone number
User choose postcode
User choose skip
User enter customer ID
User enter customer name
User enter phone number
User enter postcode

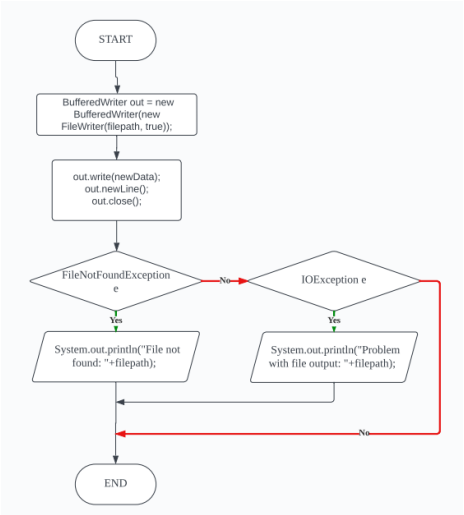

Display Customer Records Header
Initialize counter = 0
i < size of customer ID list
if customer ID = ownCustomerIDFromSales
Counter ++
Display the same customer record
if counter = 0
Display no record is found
i < size of customer mame list
if customerName = ownCustomerName
i < size of phone number list
if phoneNumber = ownPhoneNumber
i < size of postcode list
if postcode= ownPostcode
Display Search and Filter Menu for sales record
User selects:
1. By sales ID
2. By month
3. By car plate
4. By customer ID
5. By employee ID
6. Skip
User choose sales ID
User choose month
User choose car plate
User choose customer ID
User choose skip
User enter sales ID
User enter month
User enter car plate
User enter customer ID
Display Sales Records Header
Display Sales record of month Header
i < size of sales ID list
if salesID = salesIDS && userIDLogin
i < size of month list
if dateTimeS = filteredDateTime
i < size of car plate list
if carPlate = carPlateS
i < size of customer ID list
if customerID = customerIDS
i < size of customer ID list
if userIDLogin = employeeIDS
loop
No
Yes


Display Search and Filter Menu for vehicle record
User selects:
1. By car plate
2. By car model
3. By acquired price
4. By car status
5. By sales price
6. Skip
User choose car plate
User choose car model
User choose acquired price
User choose car status
User choose sales price
User choose skip
display invalid choice
User enter car plate
User enter car model
User enter car acquired price
User enter car status
User enter sales price
Display Vehicle Records Header
Initialize counter = 0
i < size of car plate list
i < size of month list
i < size of acquired price list
i < size of car status list
i < size of sales price list
if carPlate = carPlateV
if carModel = carModelV
if acquiredPrice >= minimumAcquiredPrice && acquiredPrice <= maximumAcquiredPrice
if carStatus = carStatusV
if salesPrice >= minimumSalesPrice && salesPrice <= maximumSalesPrice
Counter ++
Display the same vehicle record
if counter = 0
Display no record is found
loop
Display "SALES INSIGHTS"
checkManagement(userIDLogin)==true
Display "Sorry, you do not have acess to sales insight."
Display "1.Annual Sales"; Display "2.Monthly Sales"; Display"3.Number of cases"
Display "Enter Choice."
Switch on salesInsights
Case 1:
Display "Enter year"
Invoke getAnnualSales (year)
Print annual sales

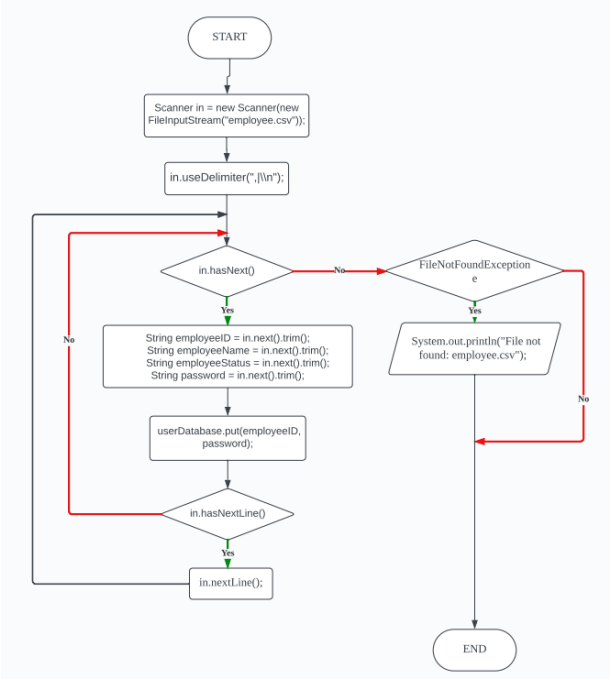


Methods

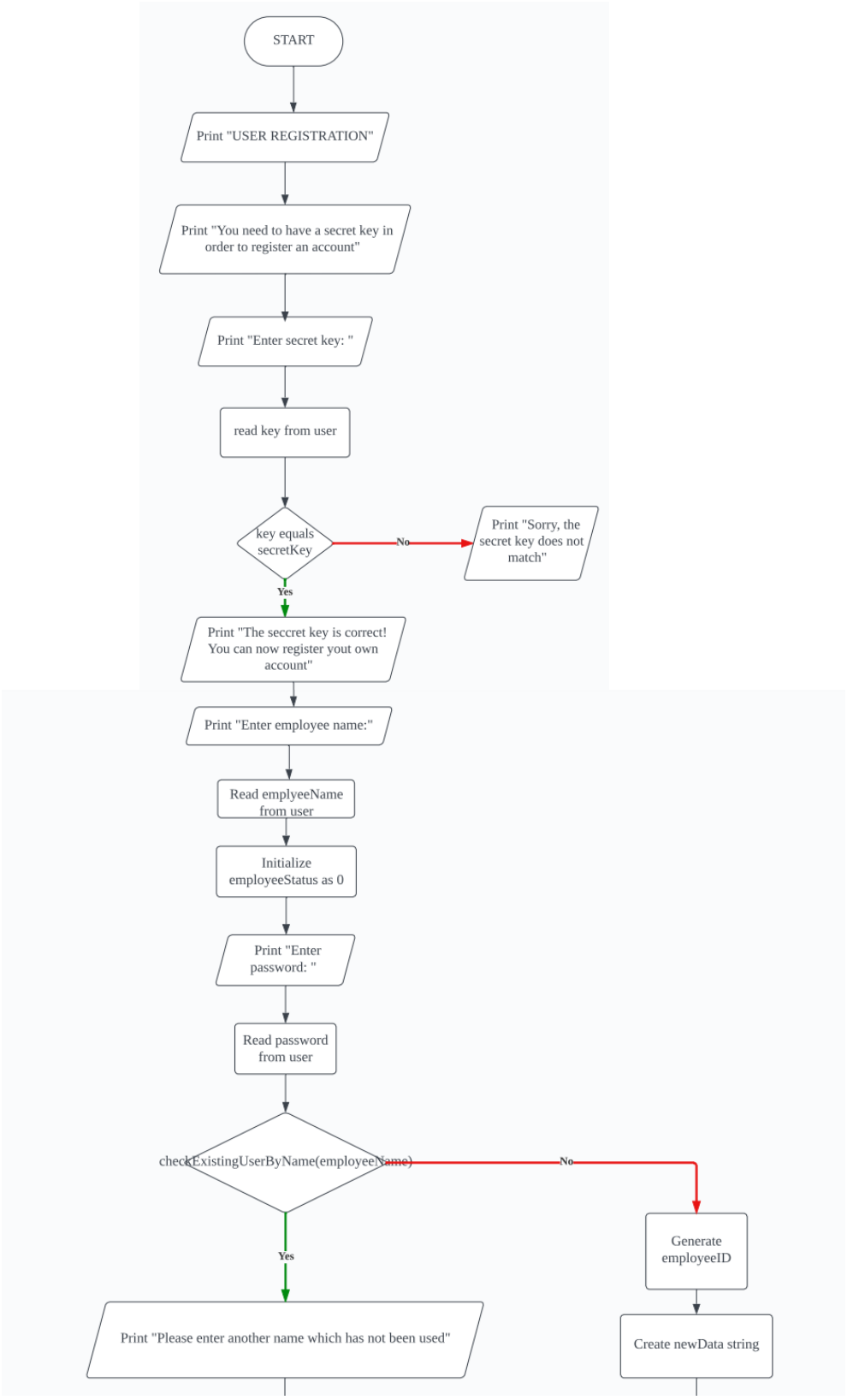
1. *writeToFile(String filepath, String newData)*

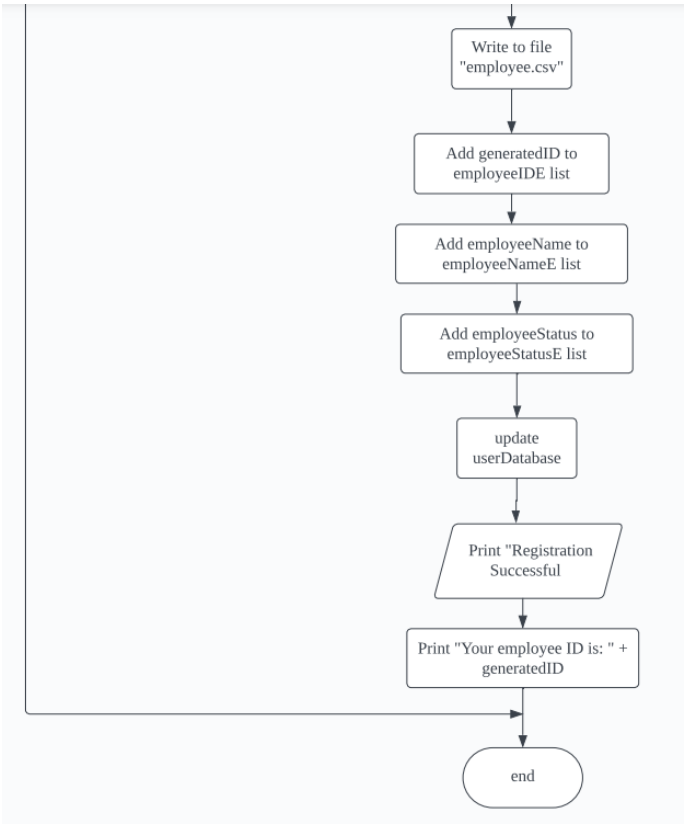


2. *loadUserData()*

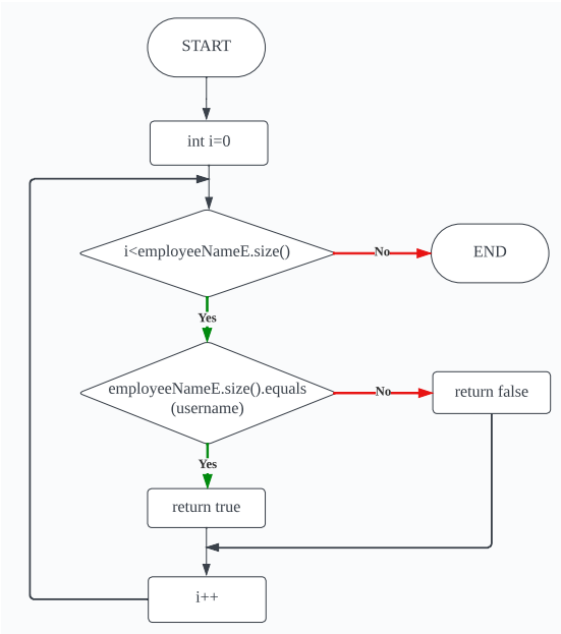


3. *userRegister()*

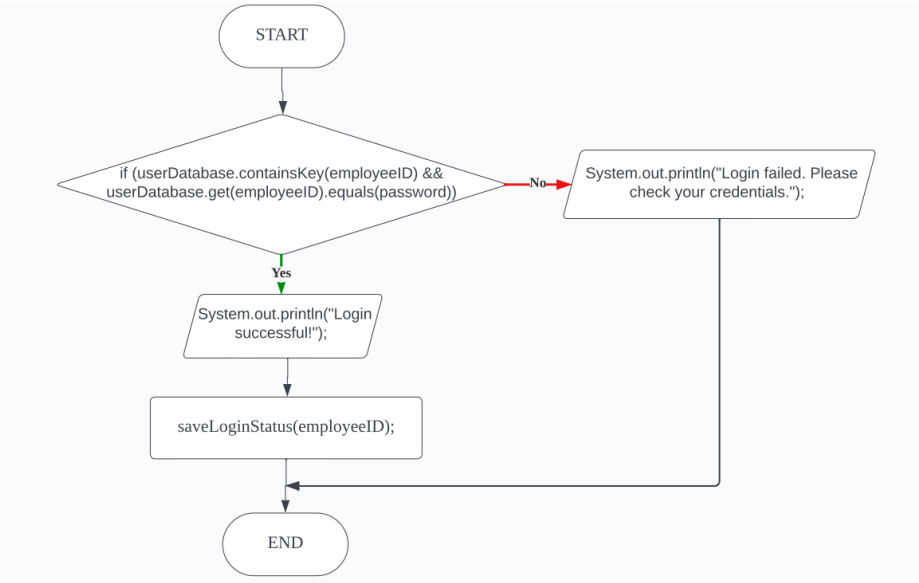




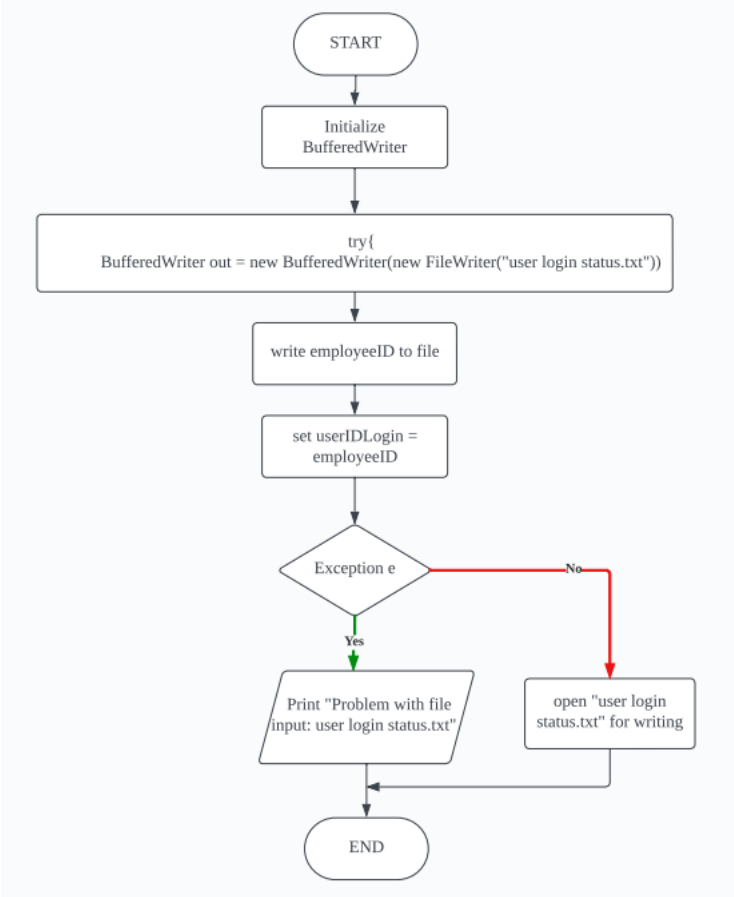
4. *checkExistingUserByName(String username)*



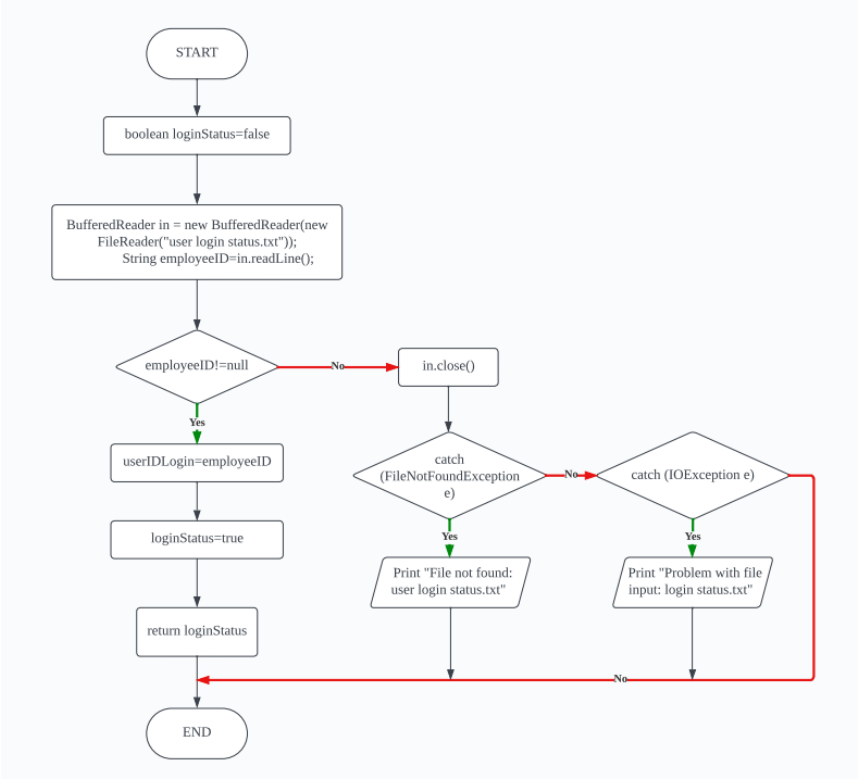
5. *userLogin(String employeeID, String password)*



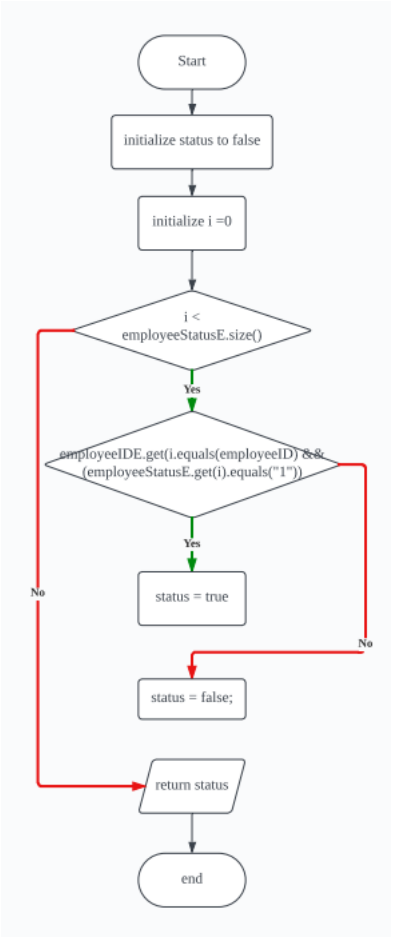
6. *saveLoginStatus(String employeeID)*



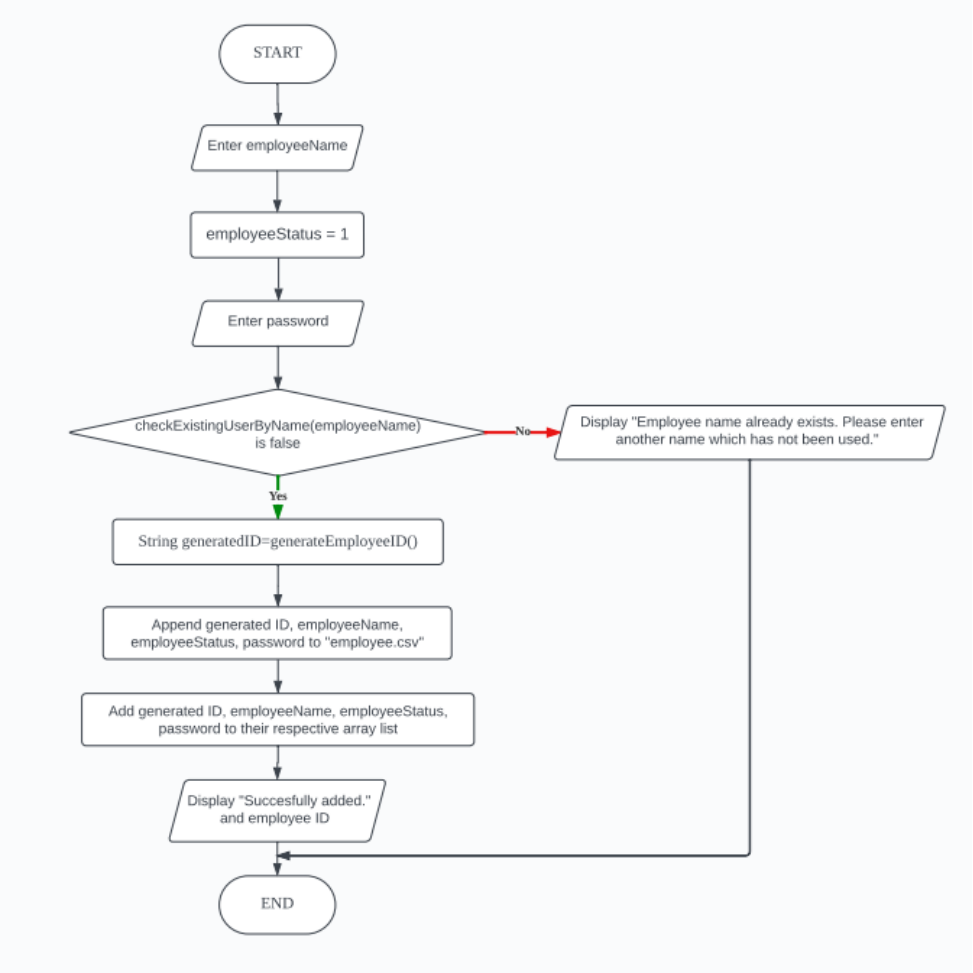
7. *checkIfLoggedIn()*



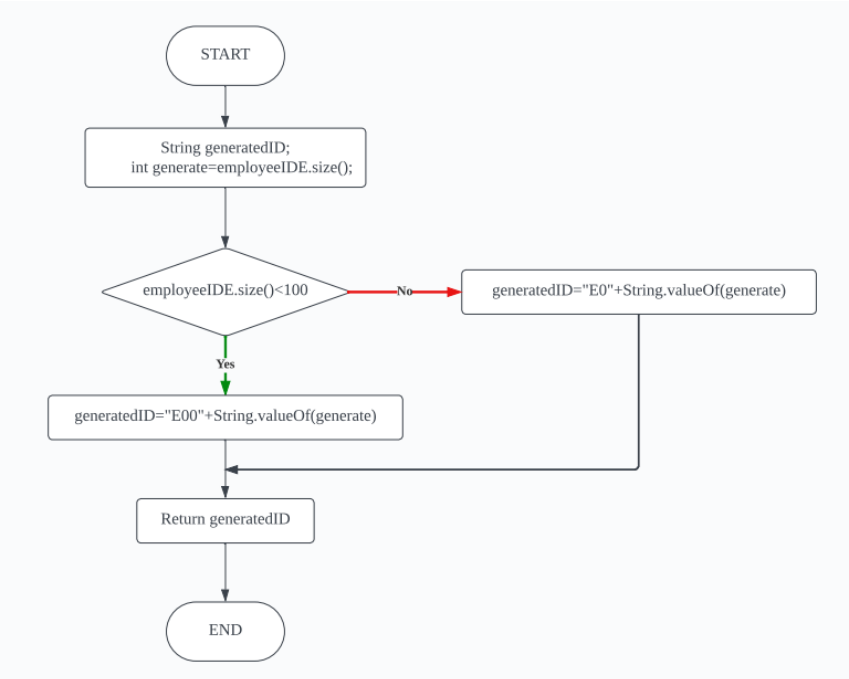
8. *checkManagement(String employeeID)*



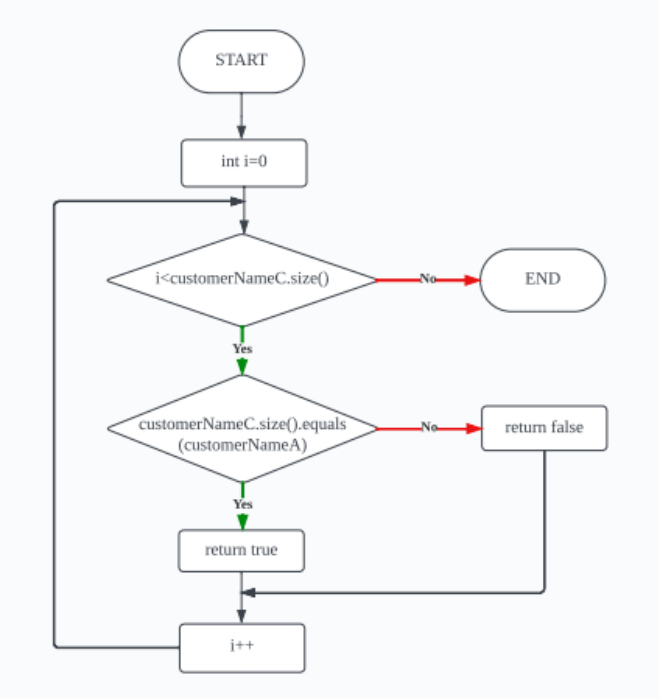
9. addManagement()



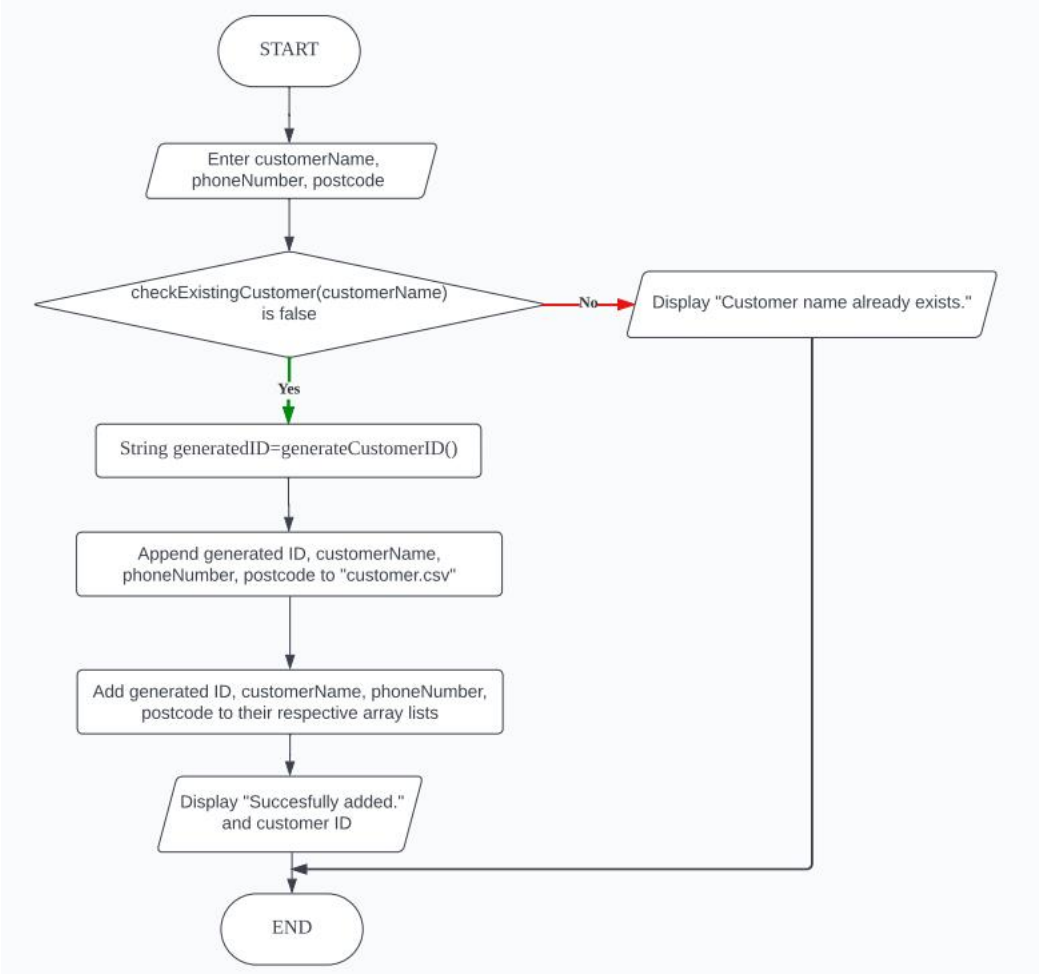
10. generateEmployeeID()



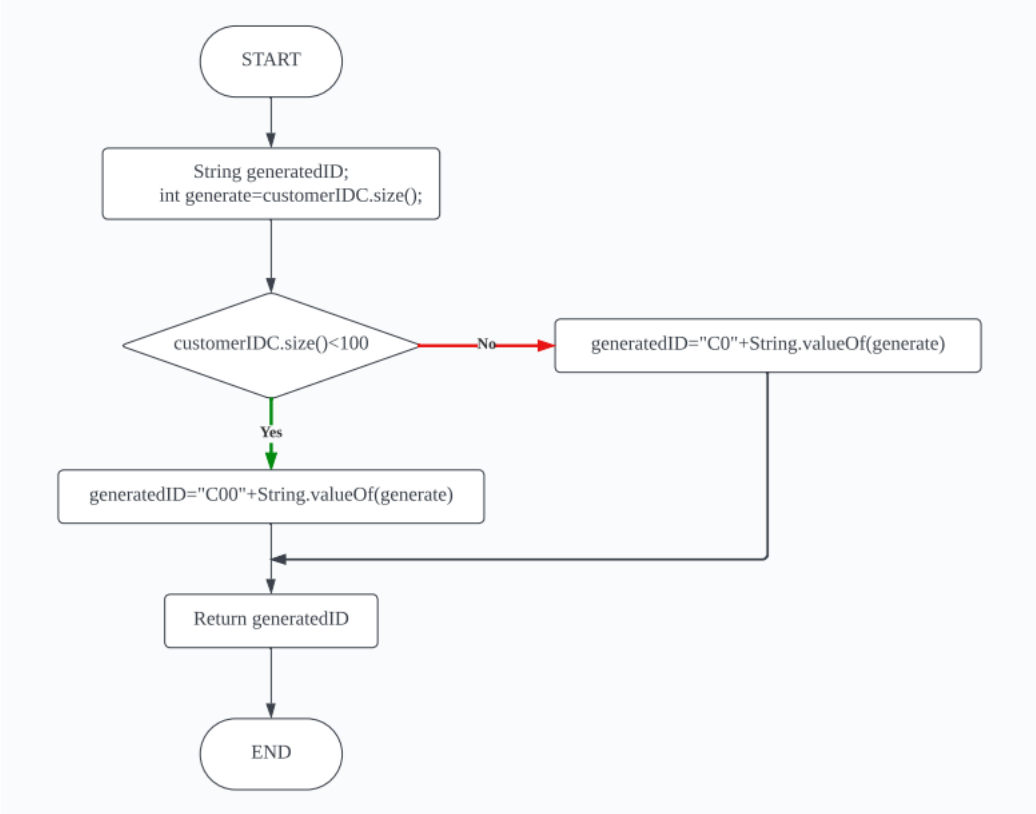
11. checkExistingCustomer(String customerNameA)



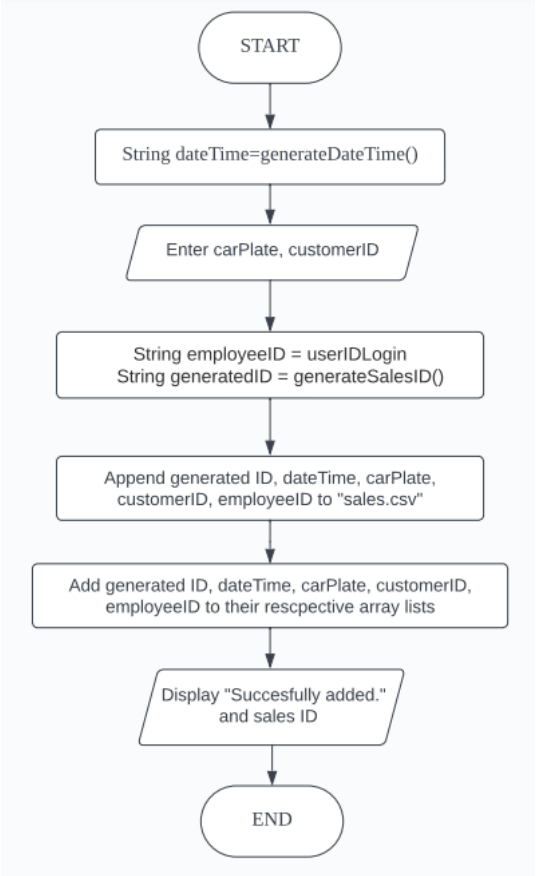
12. addCustomer()



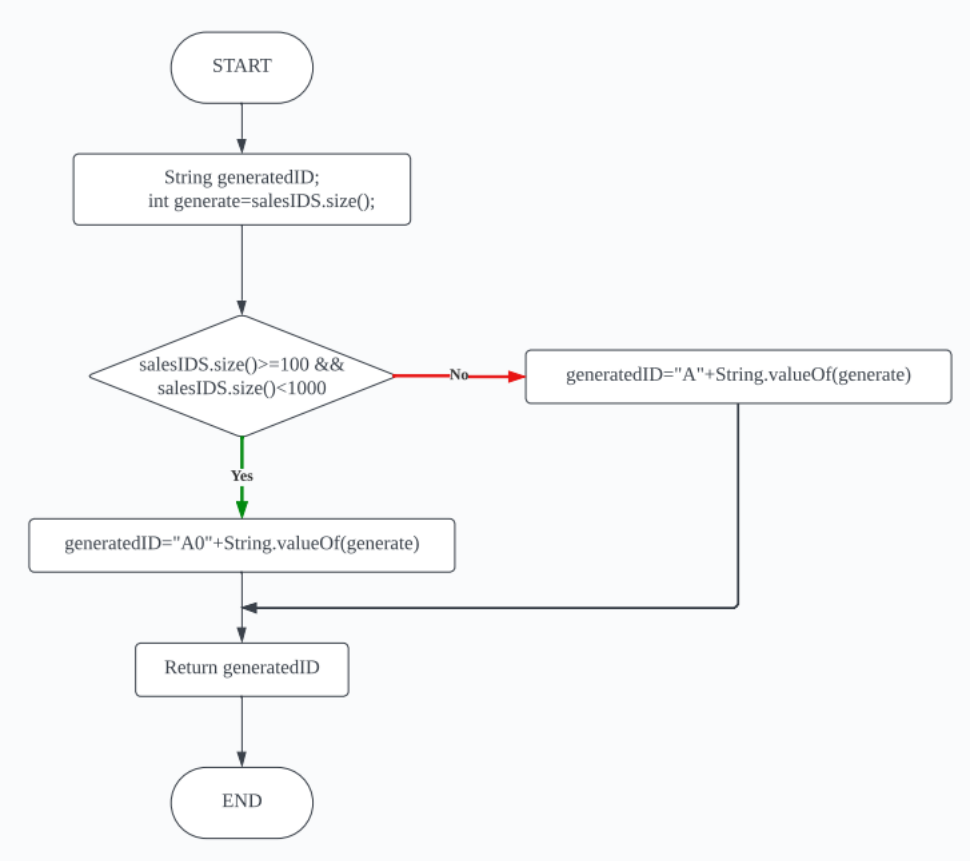
13. generateCustomerID()



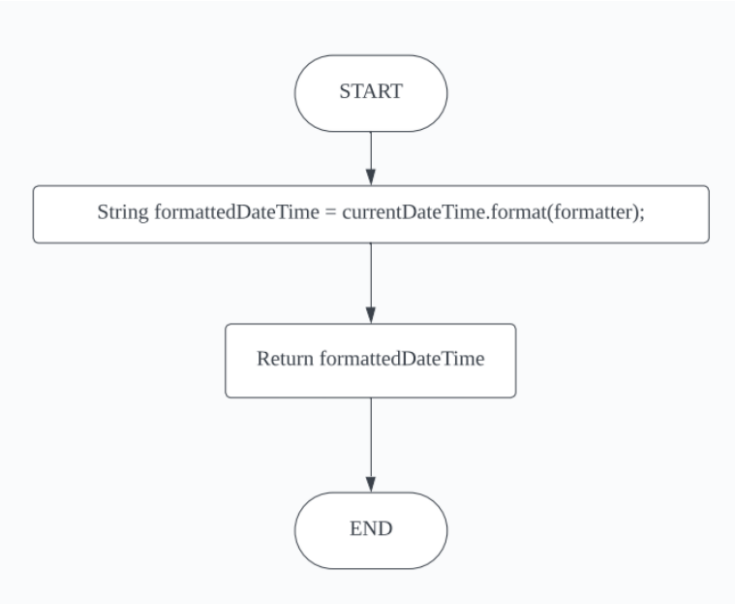
14. addSales()



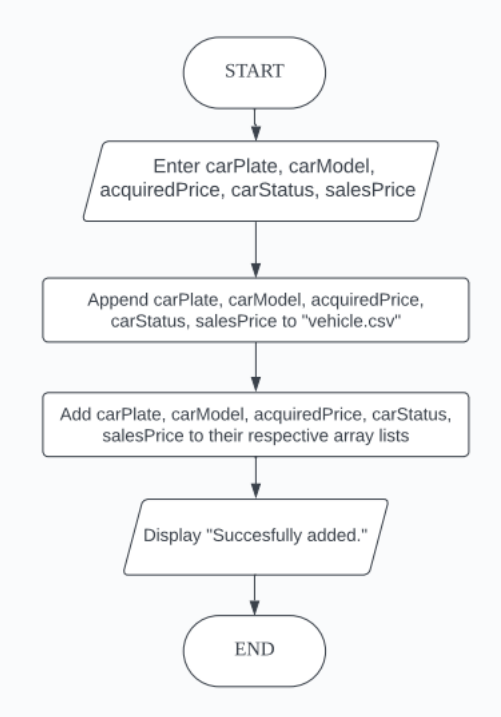
15. generateSalesID()



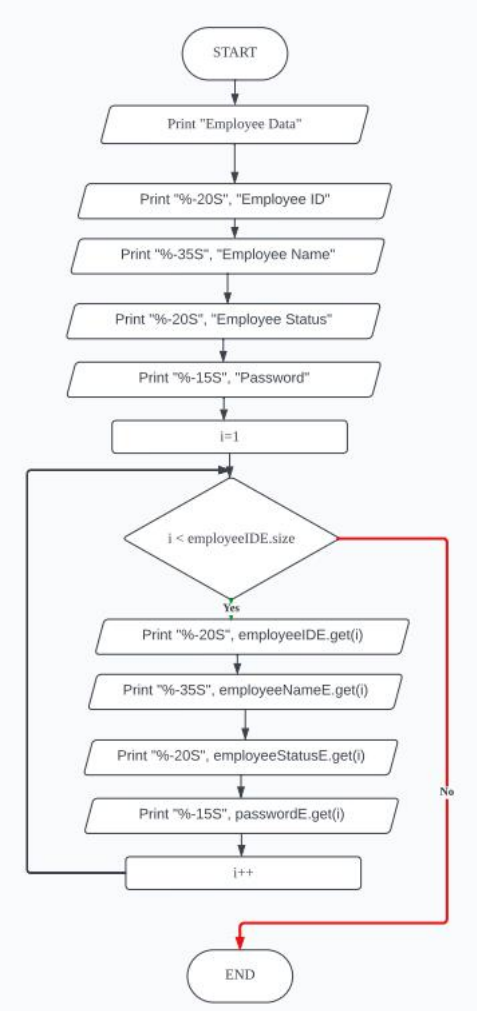
16. generateDateTime()



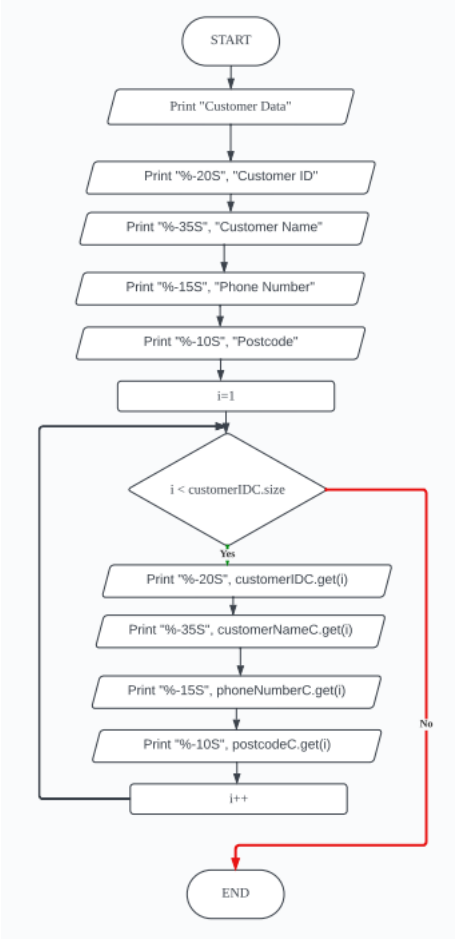
17. addVehicle()



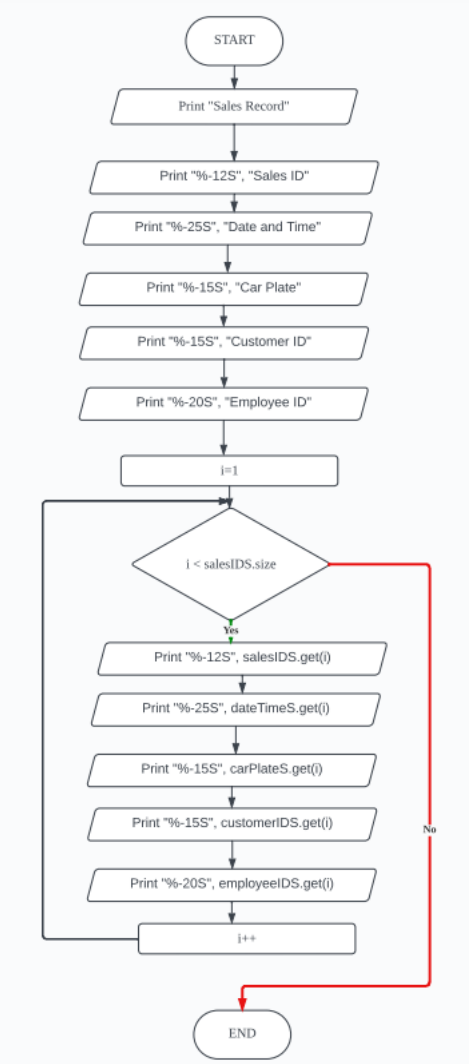
18. viewAllEmployee()



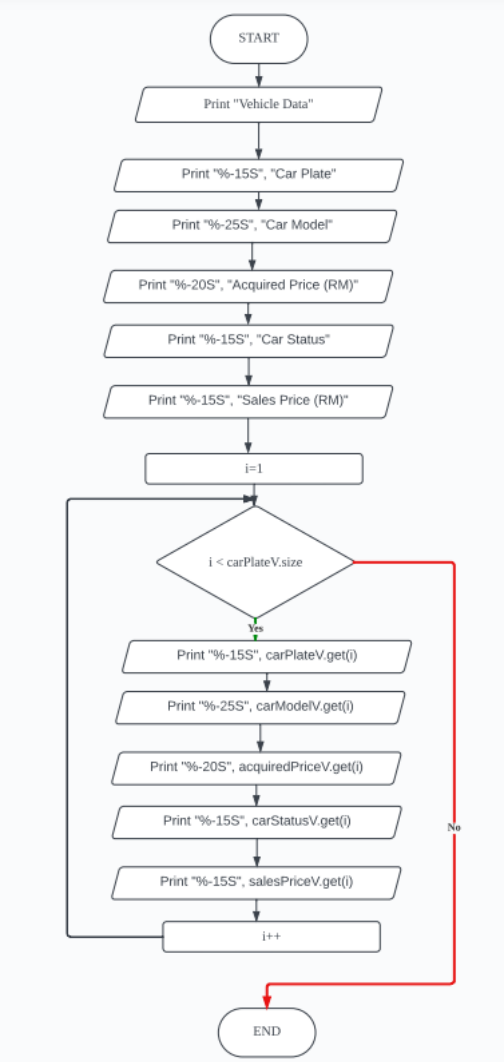
19. viewAllCustomer()



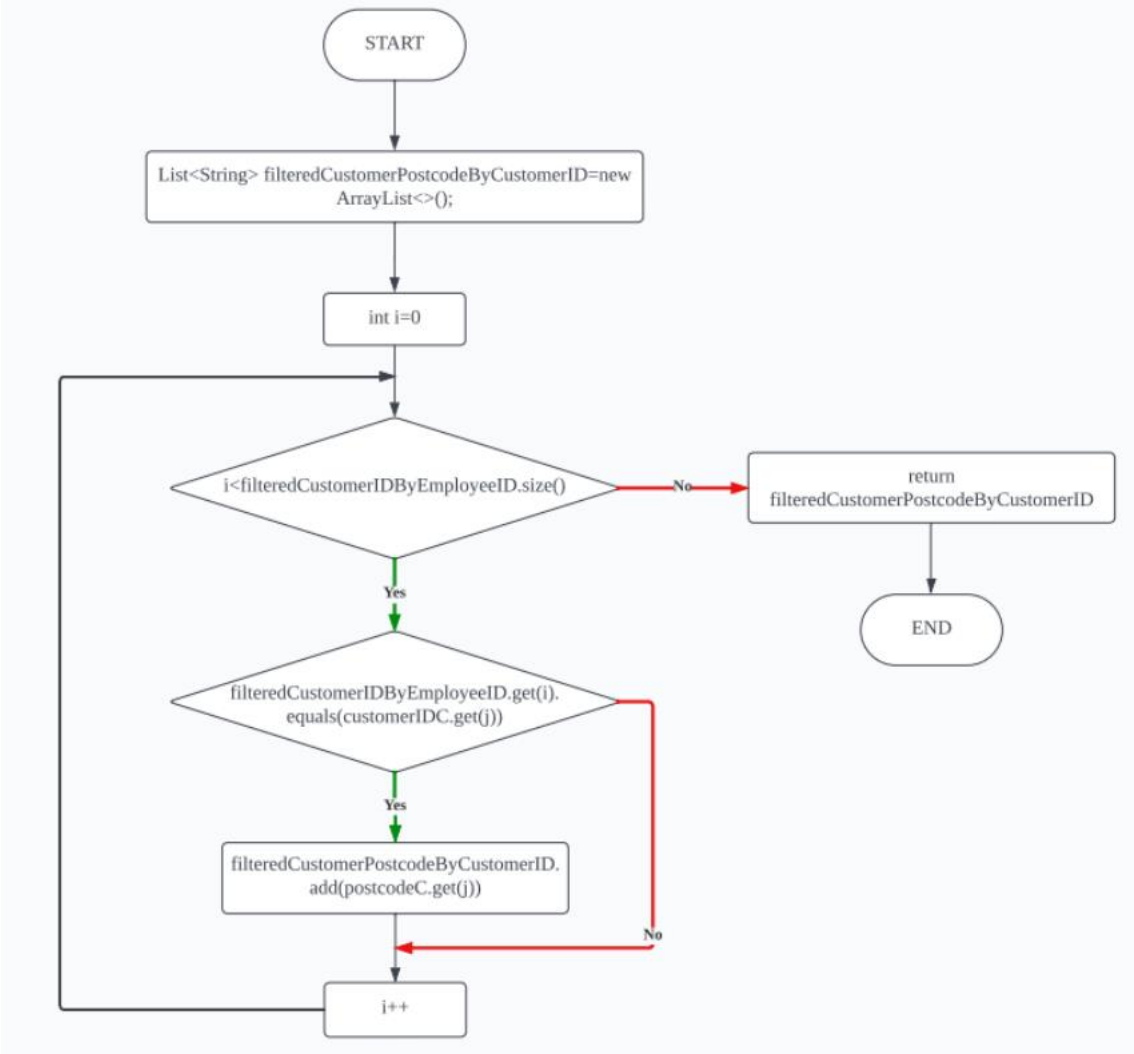
20. viewAllSales()



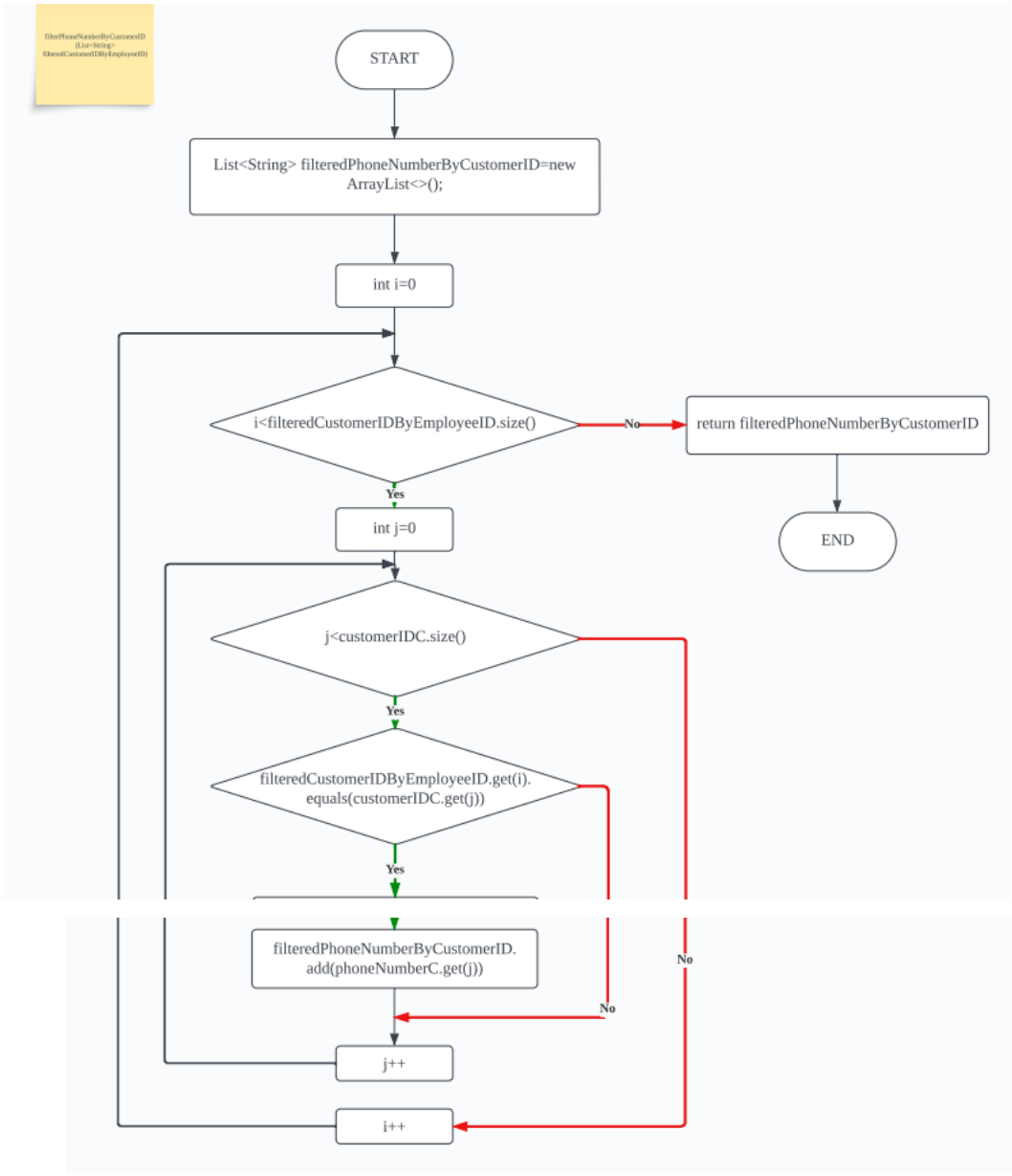
21. viewAllVehicle()



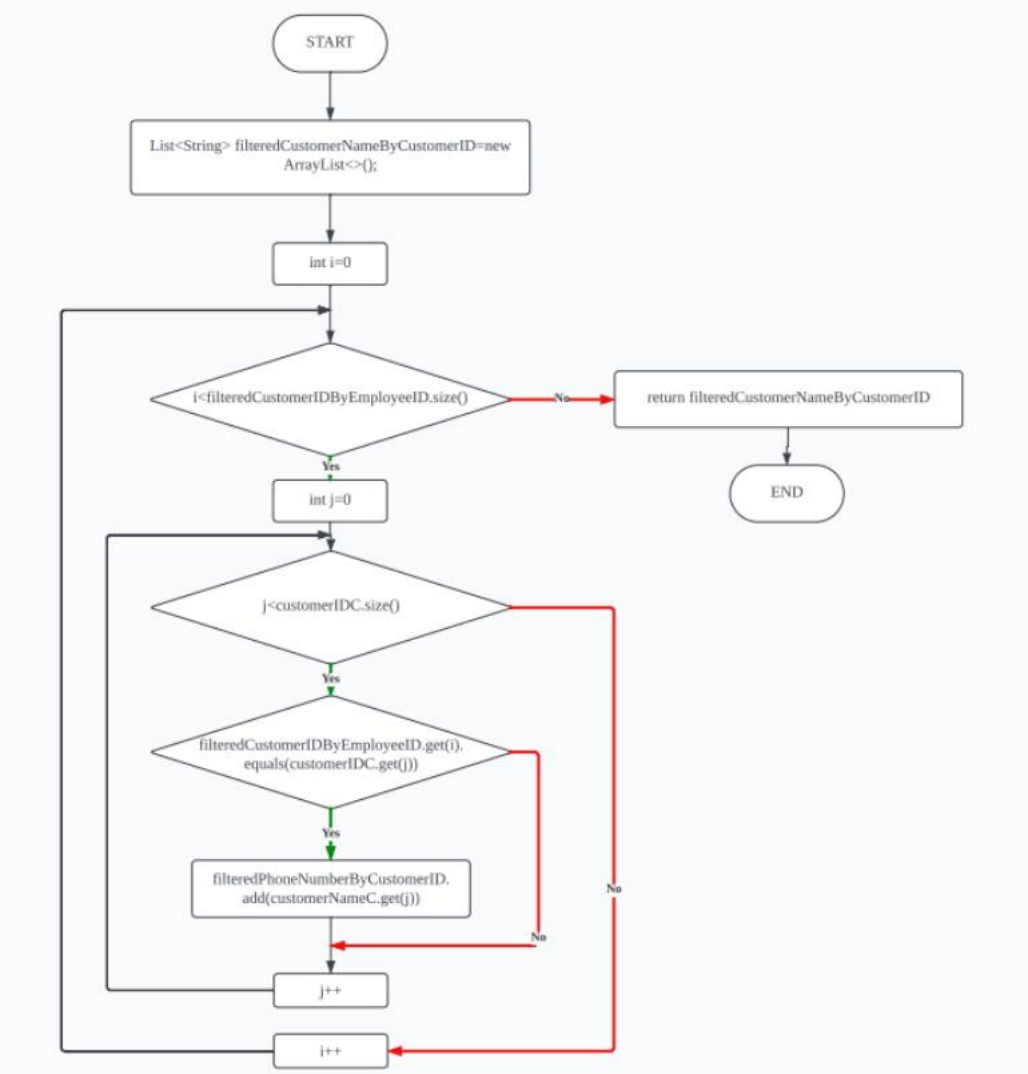
22. *filterCustomerPostcodeByCustomerID (List<String> filteredCustomerIDByEmployeeID)*



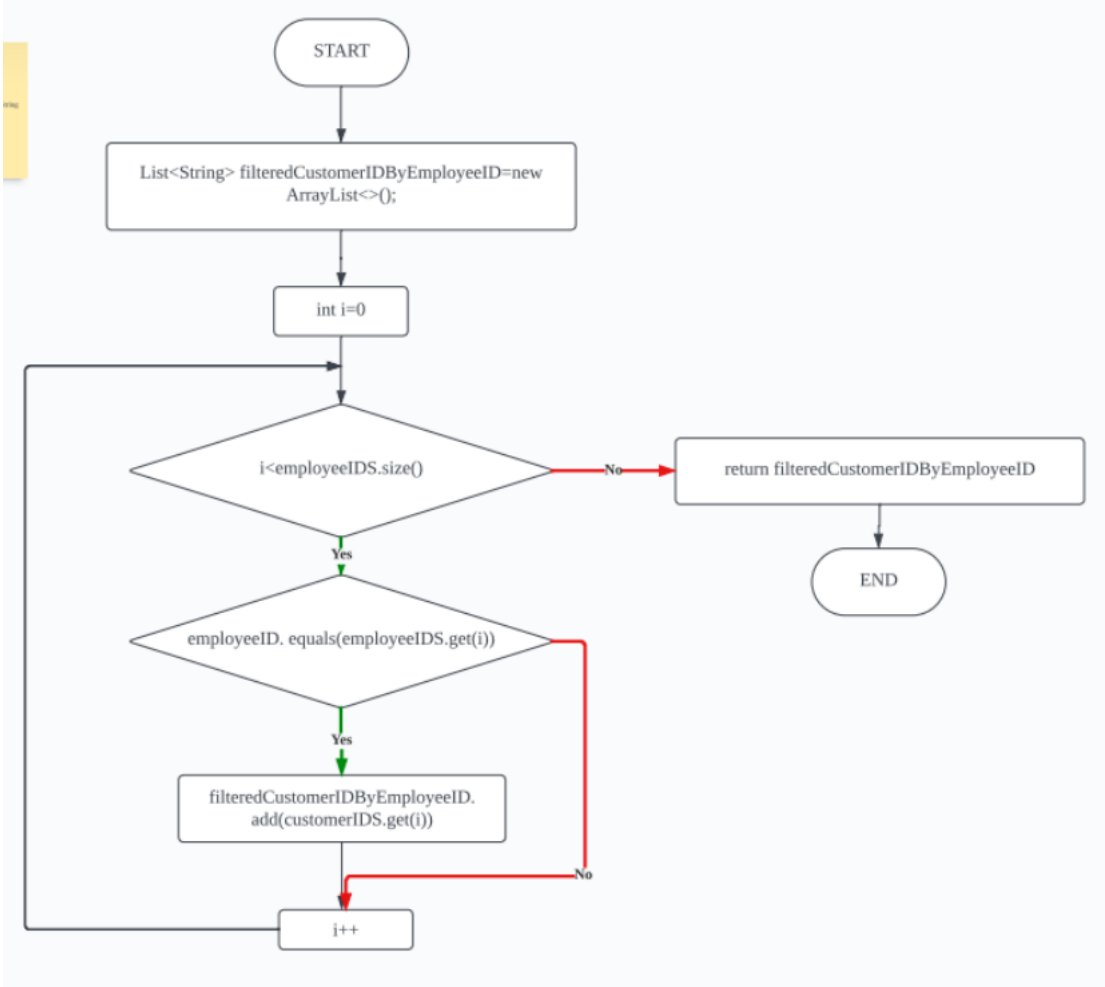
23. filterPhoneNumberByCustomerID (List<String> filteredCustomerIDByEmployeeID)



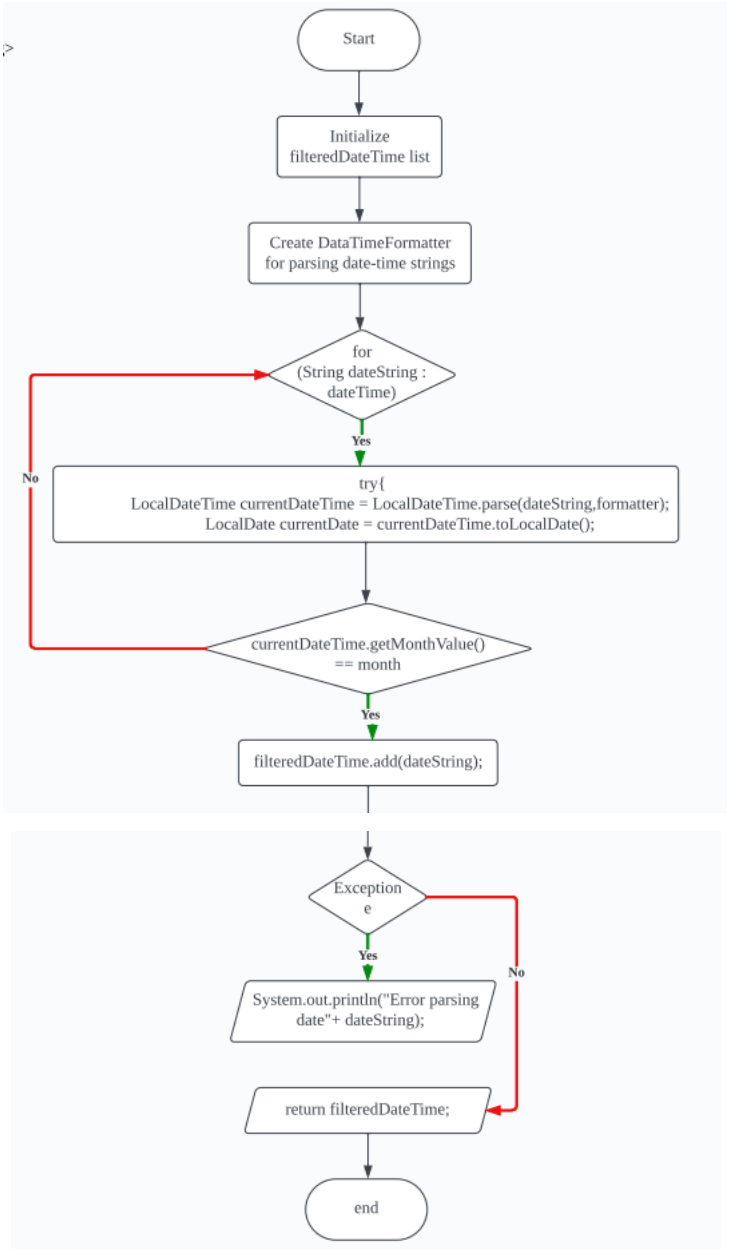
24. filterCustomerNameByCustomerID (List<String> filteredCustomerIDByEmployeeID)



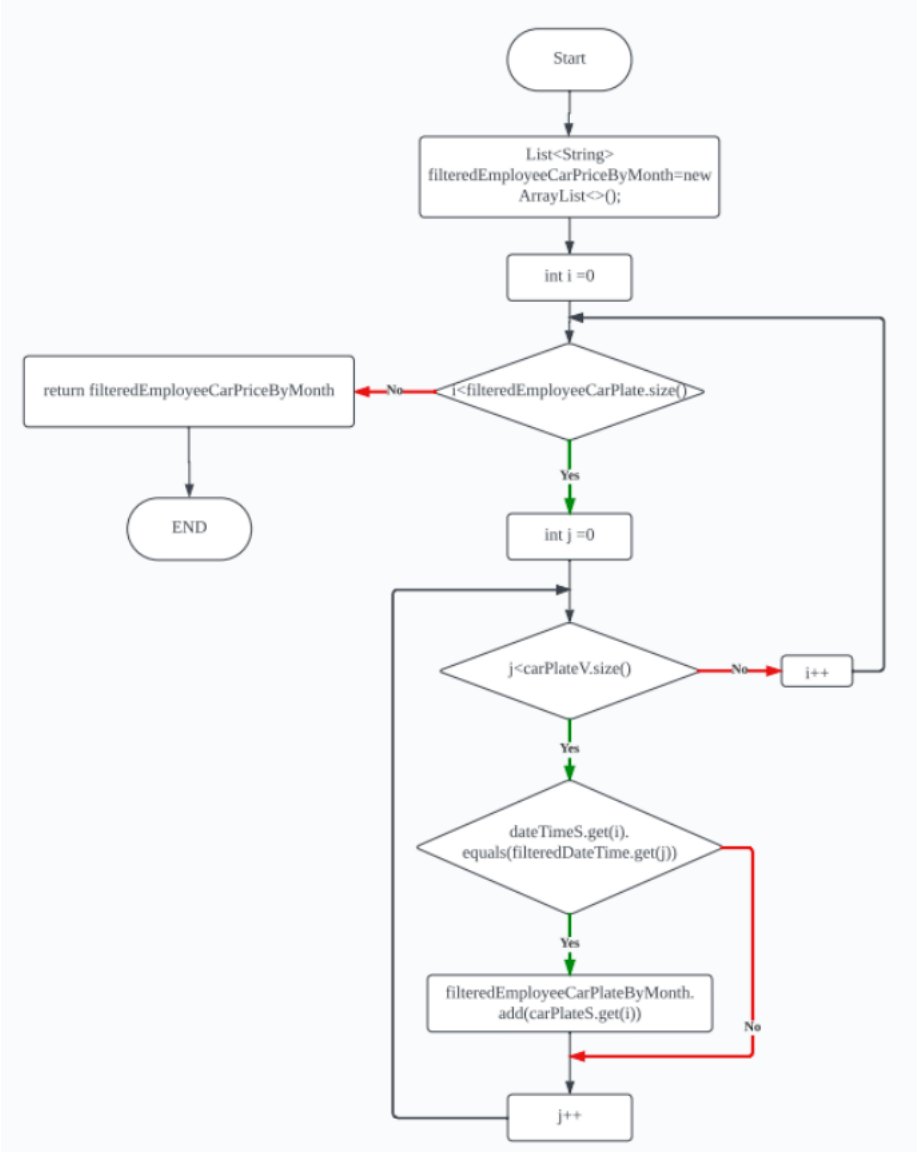
25. filterCustomerIDByEmployeeID(String employeeID)



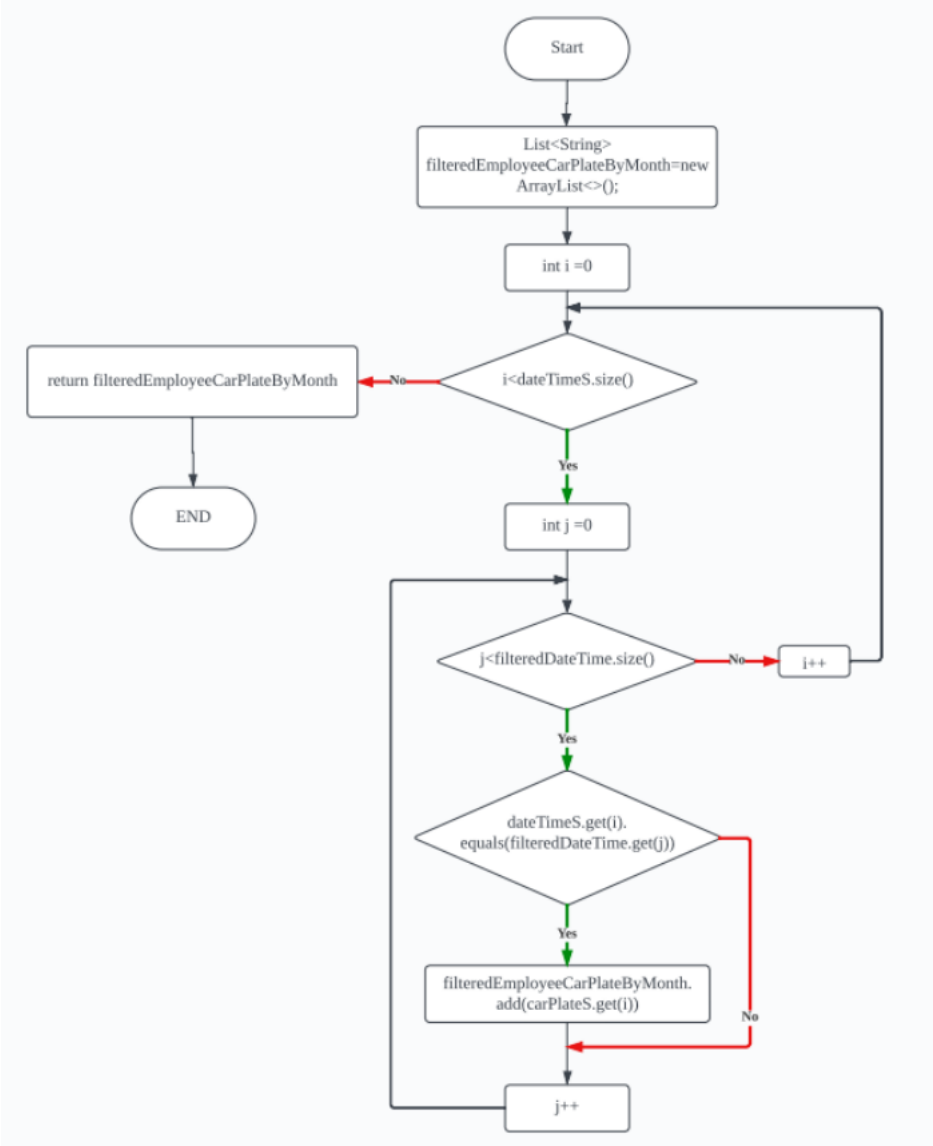
26. filterDateTimeByMonth(List<String> dateTime, int month)



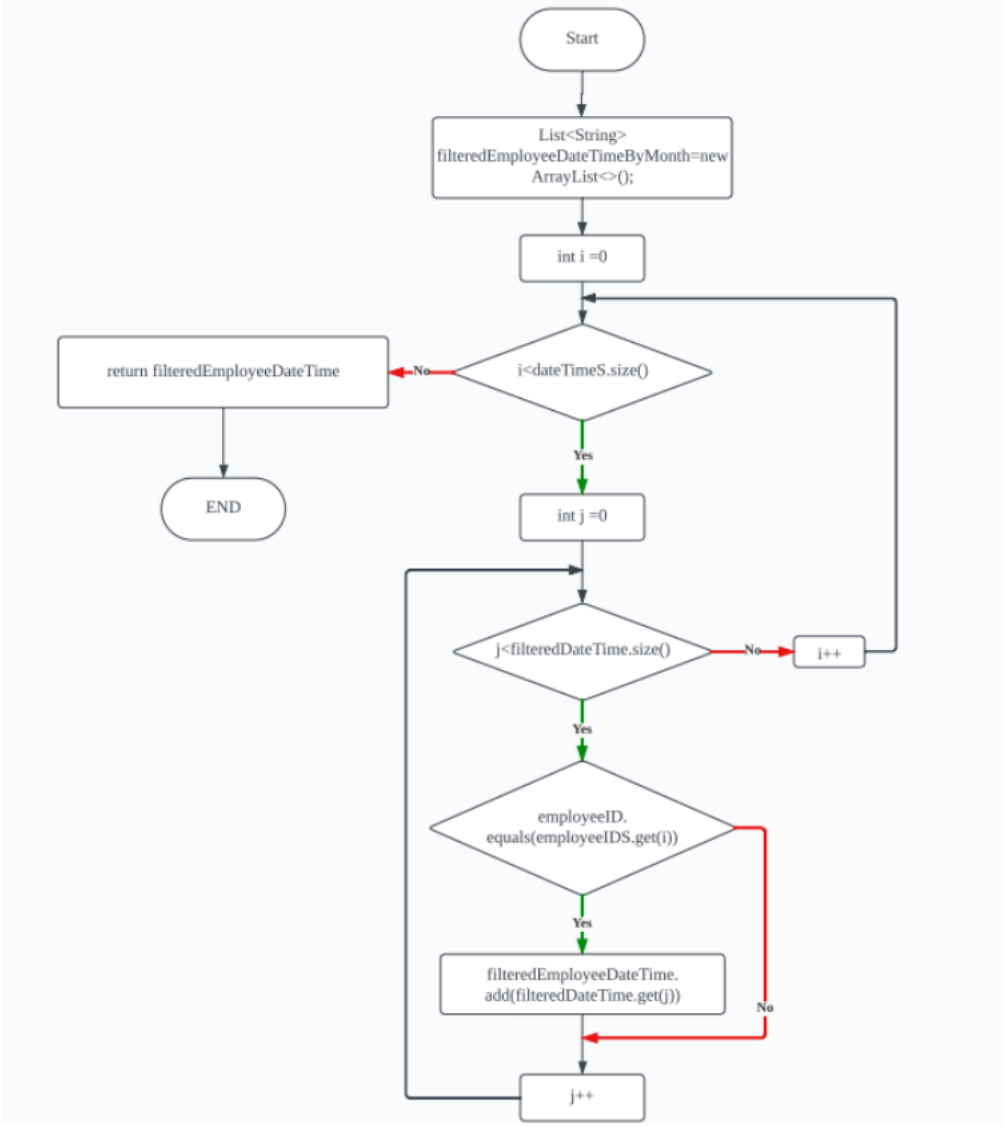
27. filterEmployeeCarPriceByMonth(List<String> filteredEmployeeCarPlate)



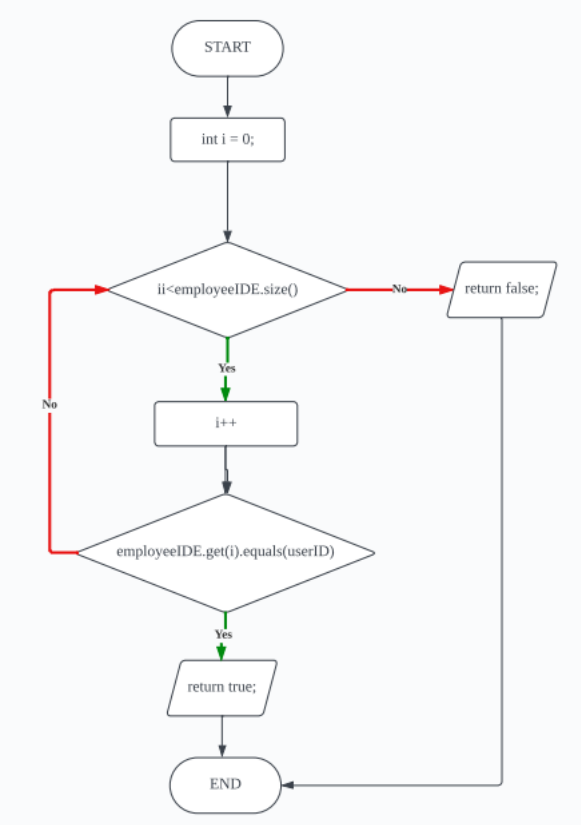
28. filterEmployeeCarPlateByMonth(List<String> filteredDateTime)



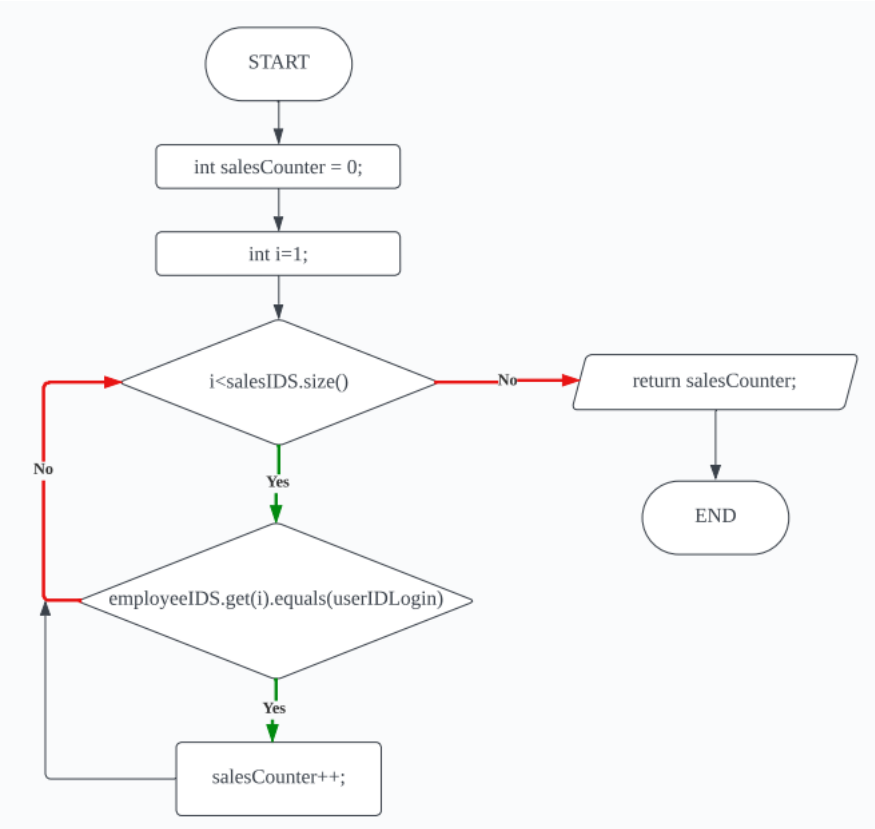
29. filterEmployeeDateTimeByMonth(List<String> filteredDateTime, String employeeID)



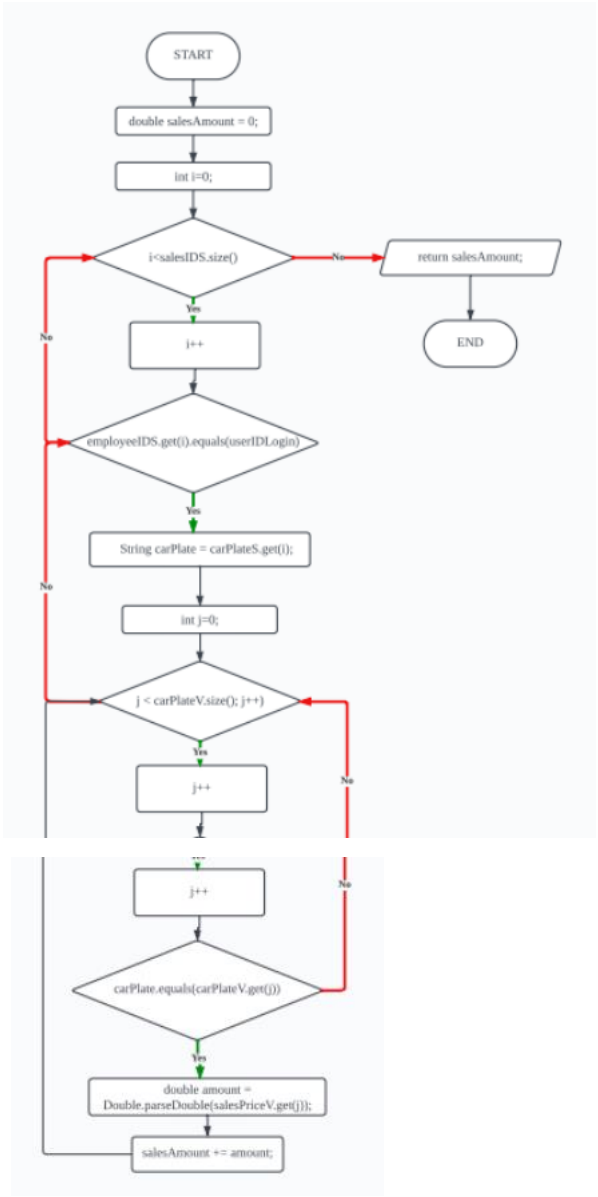
30. checkExistingUserByID(String userID)



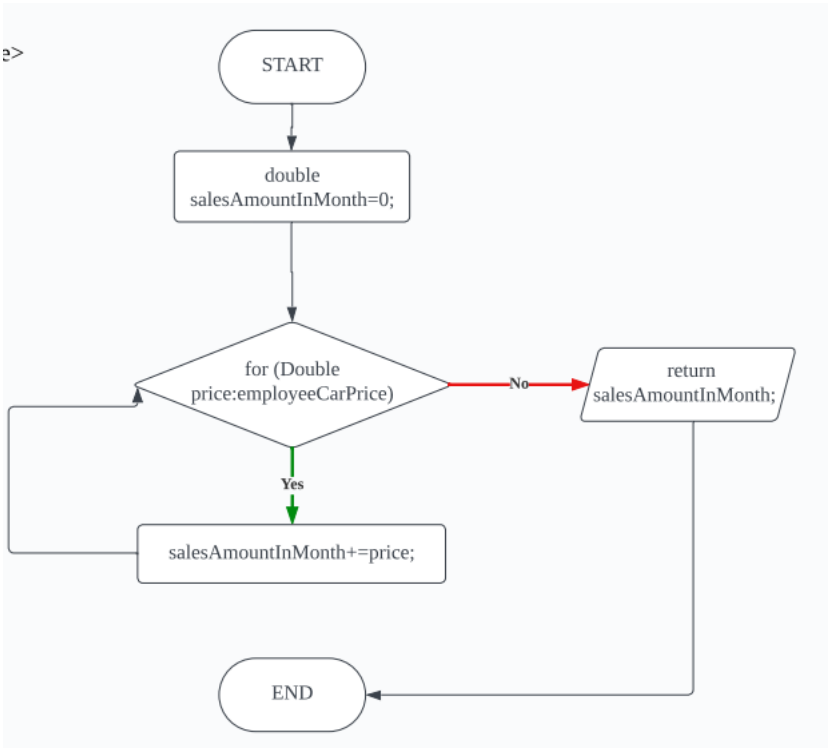
31. calSalesRecord(String employeeID)



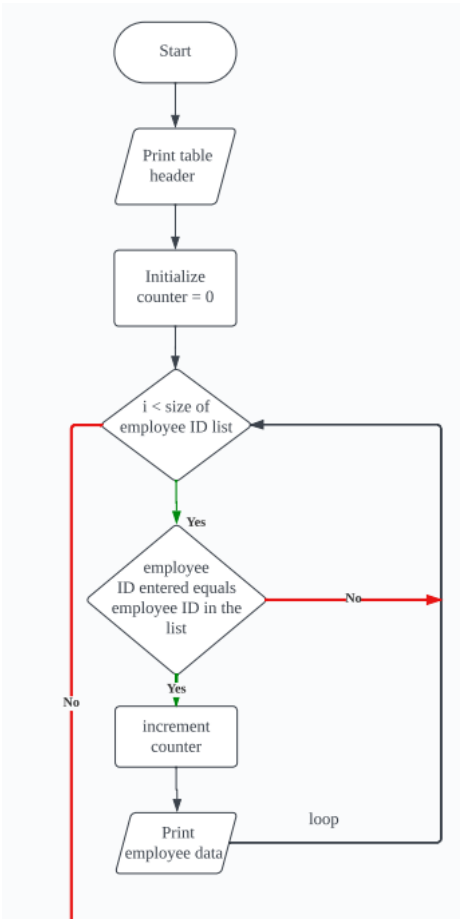
32. calSalesAmount(String employeeID)

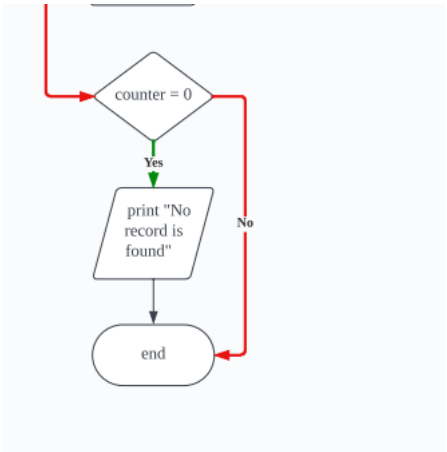


33. *calSalesAmountInMonth(List<Double> employeeCarPrice*

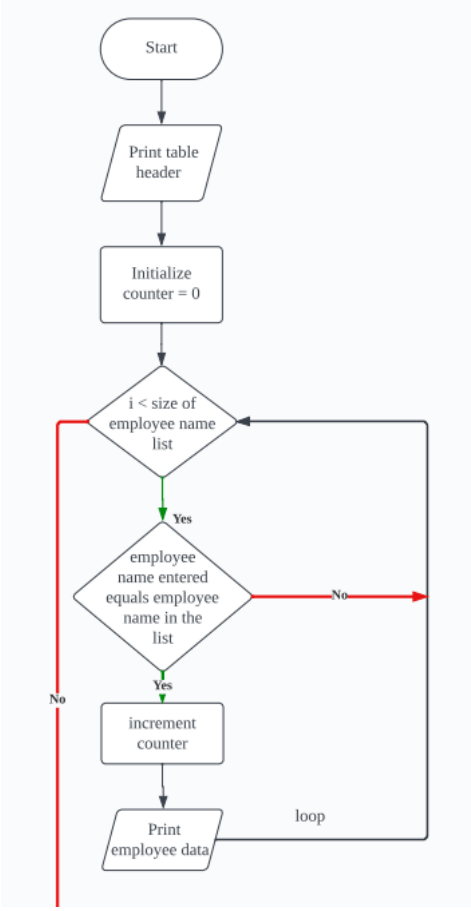


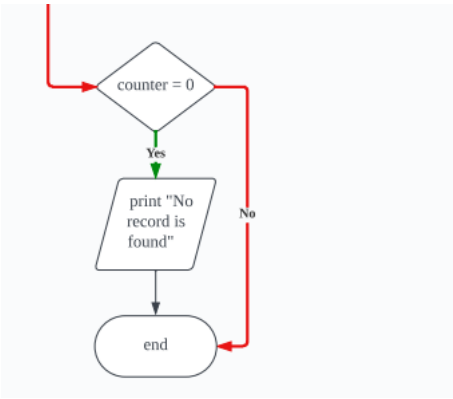
34. *managementSearchEmployeeByEmployeeID(String employeeID)*



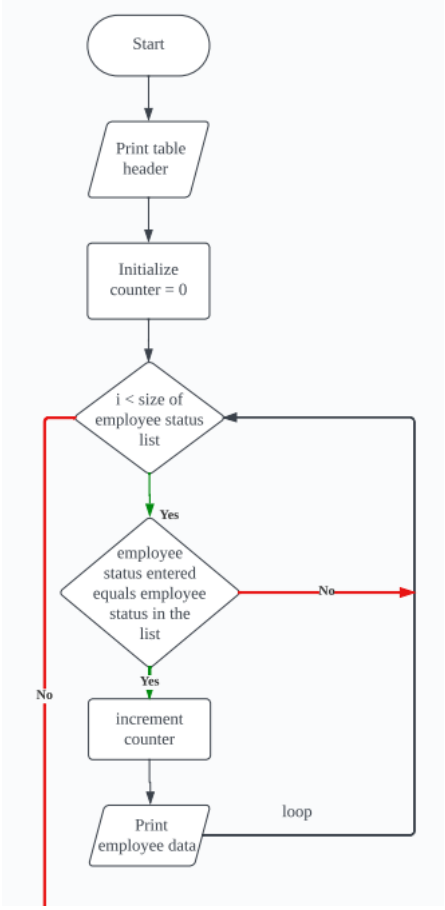


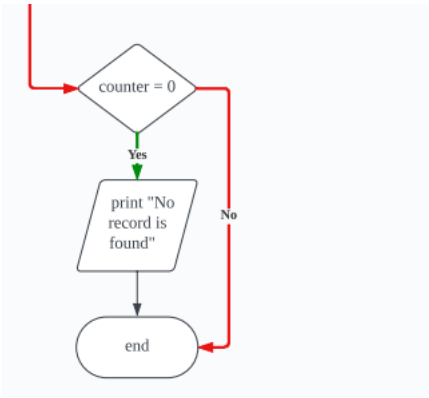
35. *managementSearchEmployeeByEmployeeName(String employeeName)*



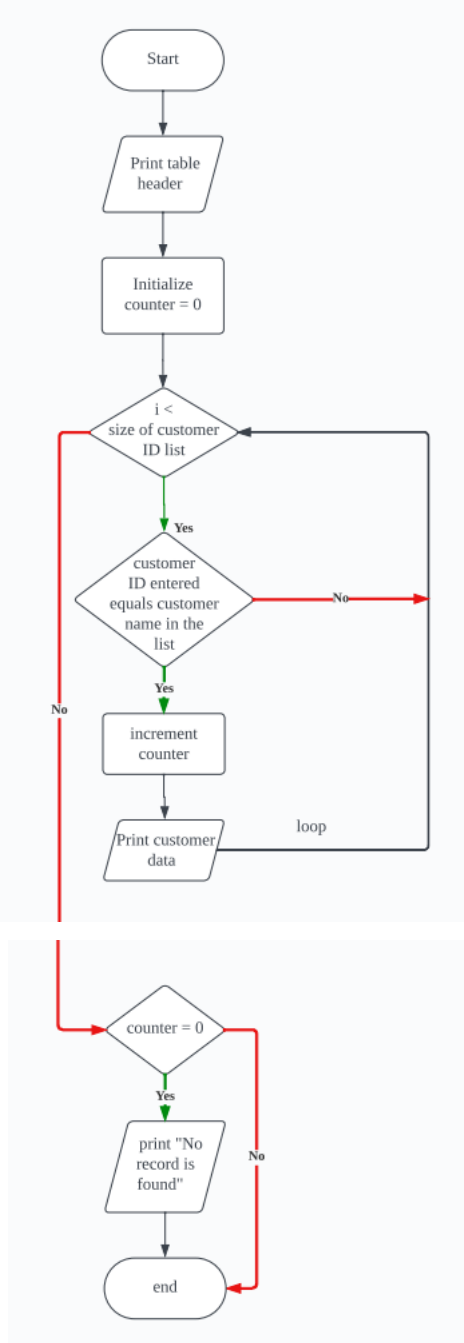


36. *managementSearchEmployeeByEmployeeStatus(String employeeStatus)*

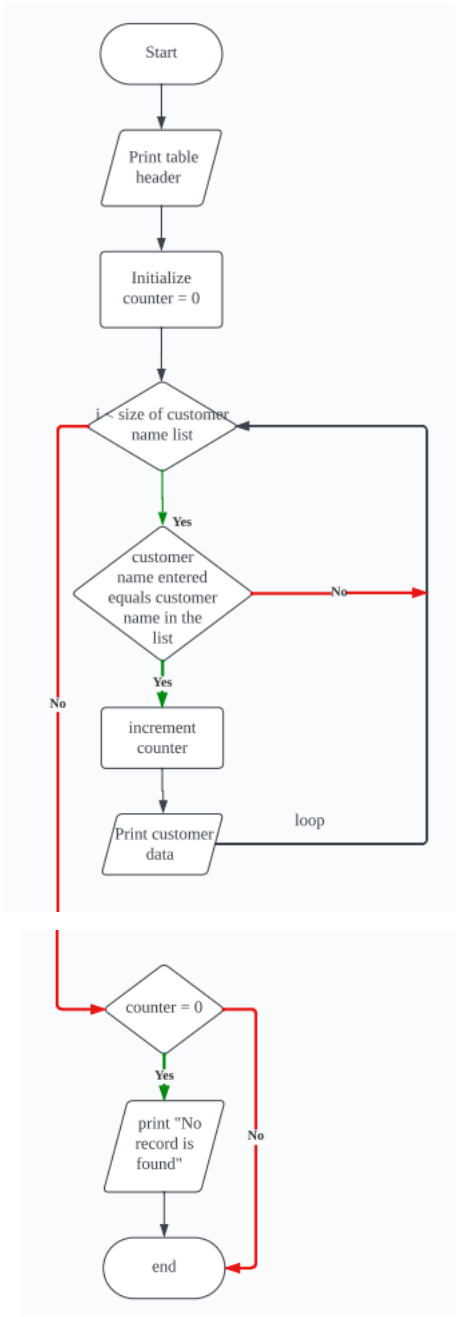




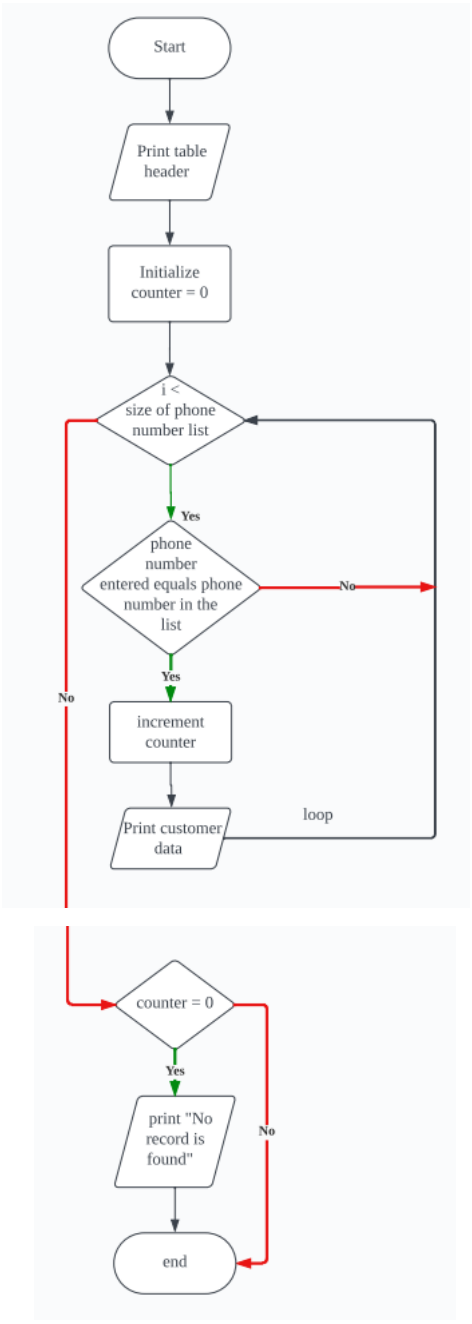
37. managementSearchCustomerByCustomerID(String customerID)



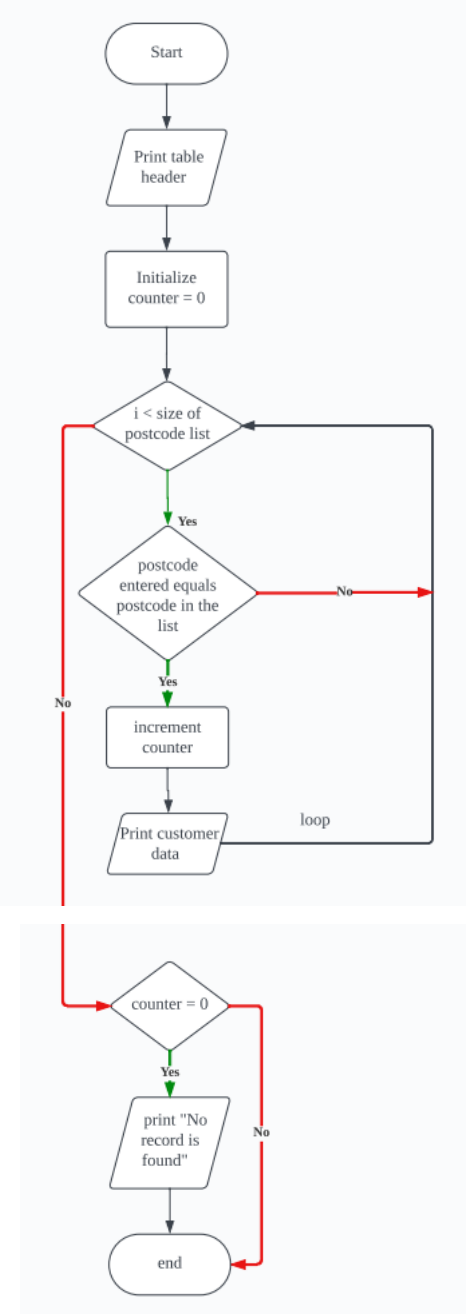
38. *managementSearchCustomerByCustomerName(String customerName)*



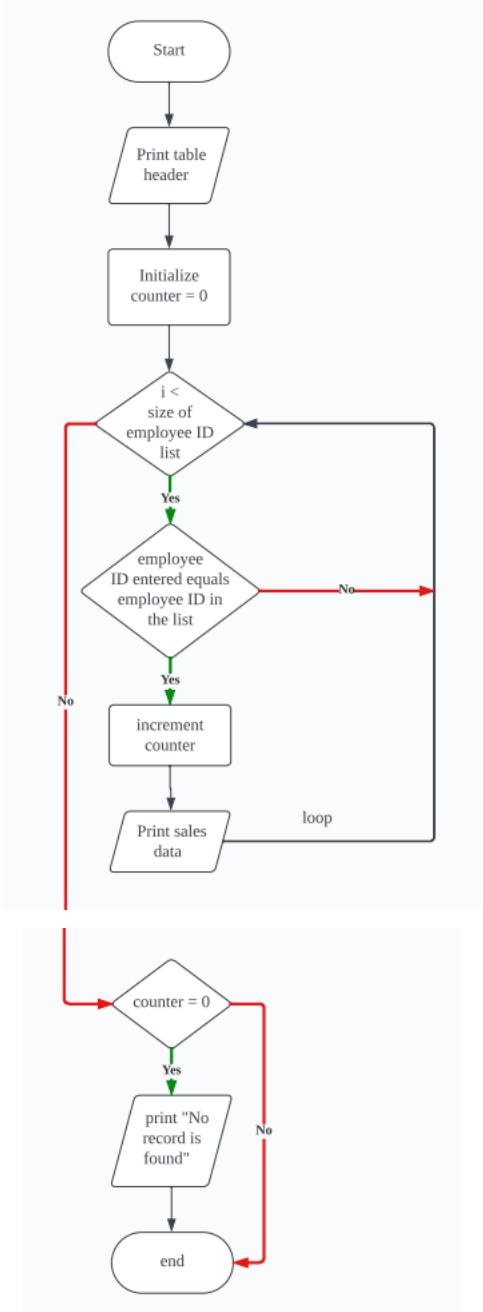
39. *managementSearchCustomerByPhoneNumber(String phoneNumber)*



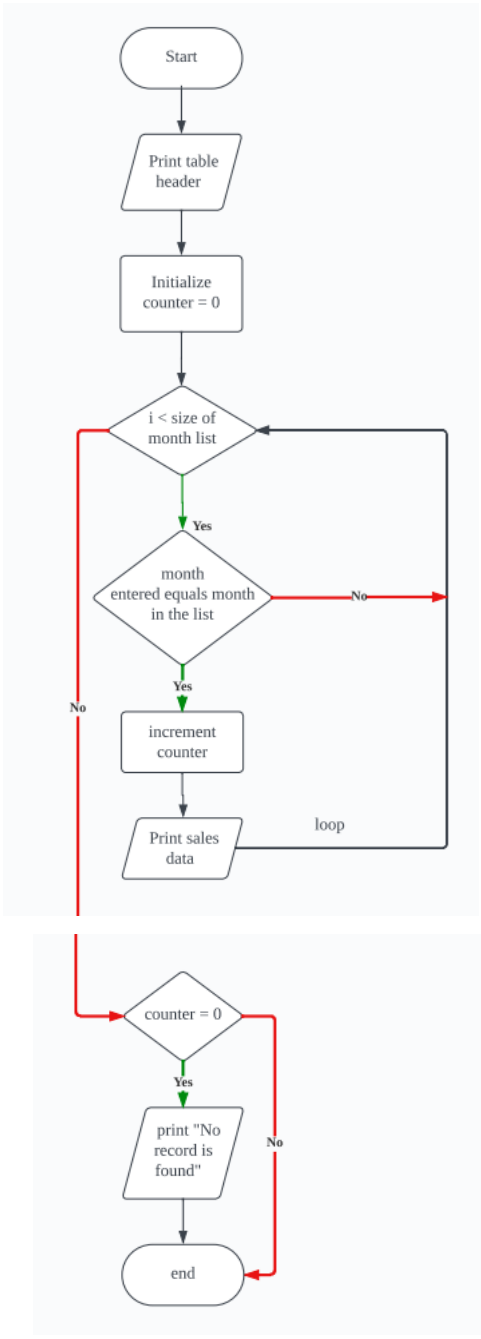
40. managementSearchCustomerByPostcode(String postcode)



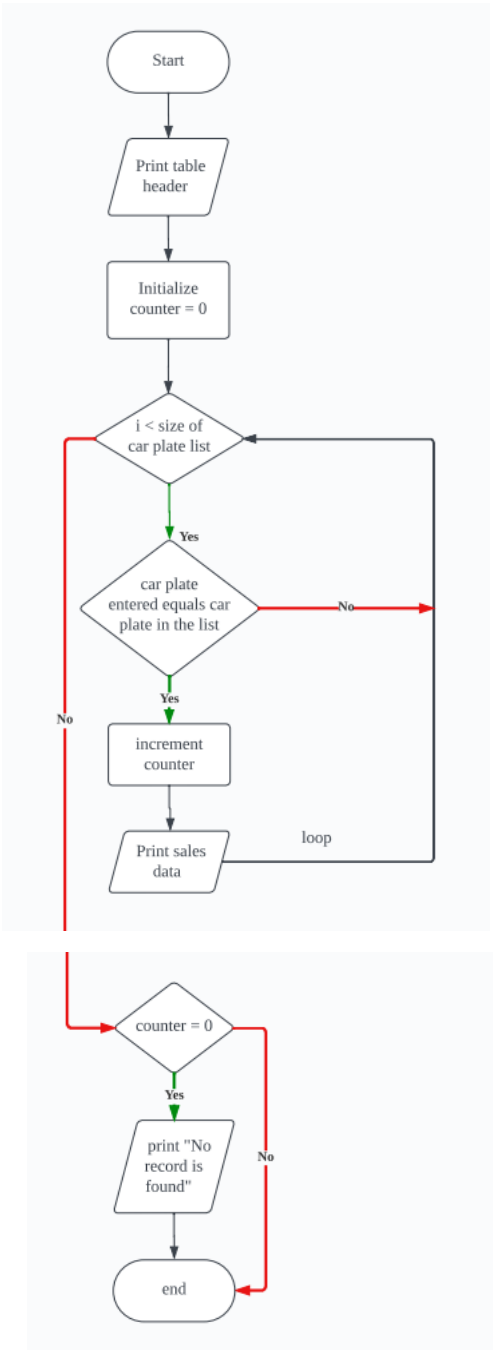
41. managementSearchSalesBySalesID(String salesID)



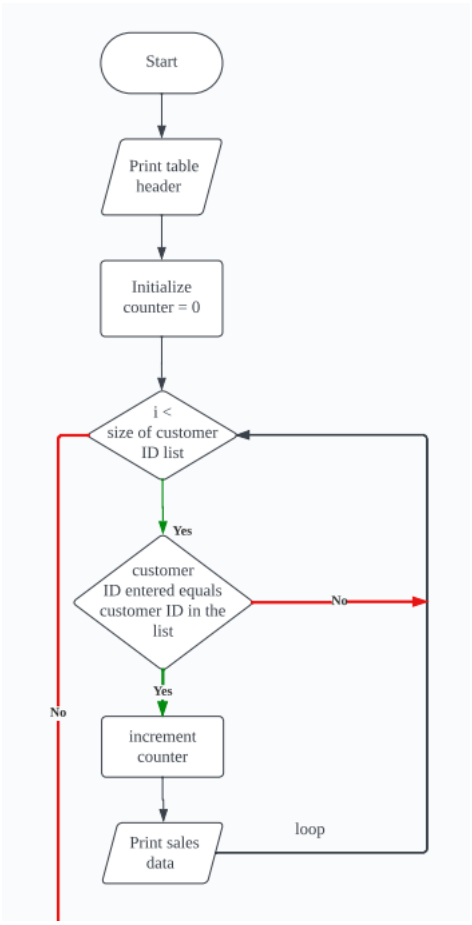
42. managementSearchSalesByMonth(int salesMonth)

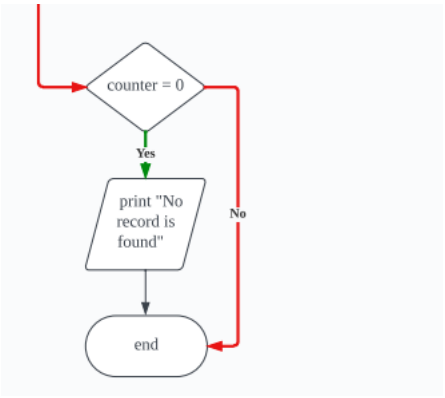


43. managementSearchSalesByCarPlate(String carPlate)

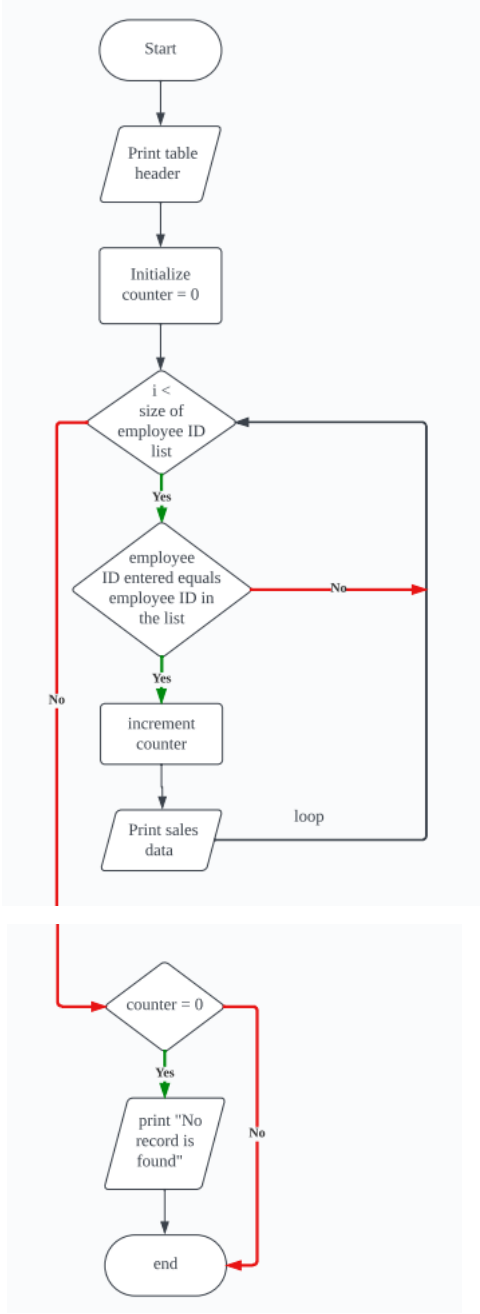


44. managementSearchSalesByCustomerID(String customerID)

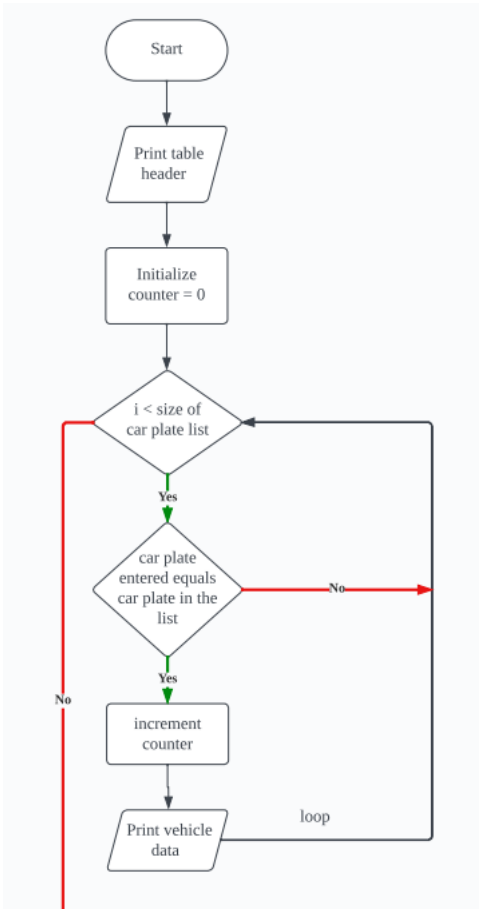


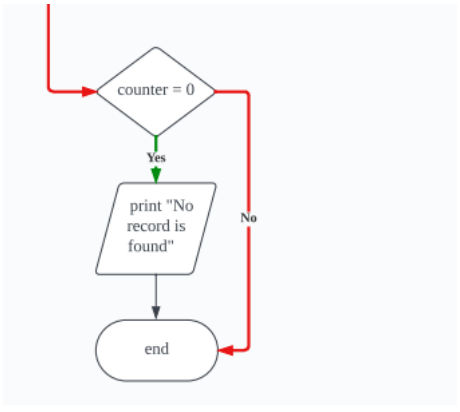


45. managementSearchSalesByEmployeeID(String employeeID)

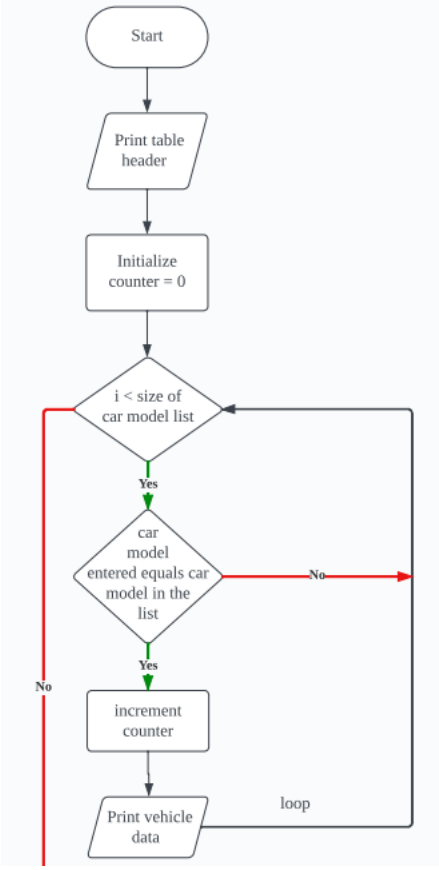


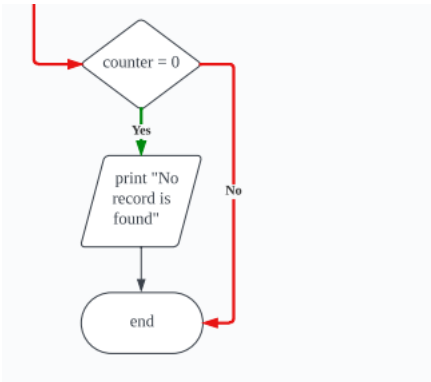
46. managementSearchVehicleByCarPlate(String carPlate)



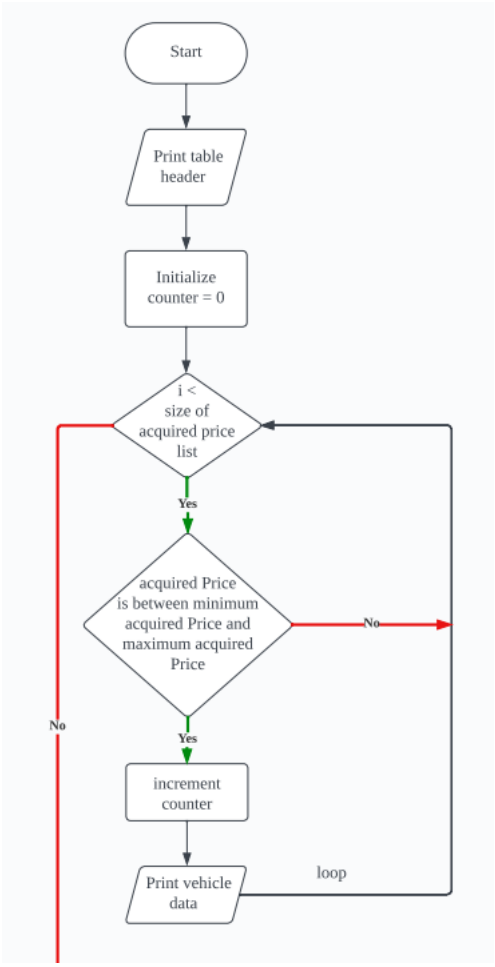


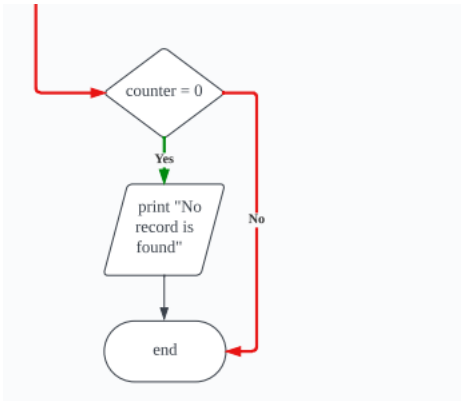
47. managementSearchVehicleByCarModel(String carModel)



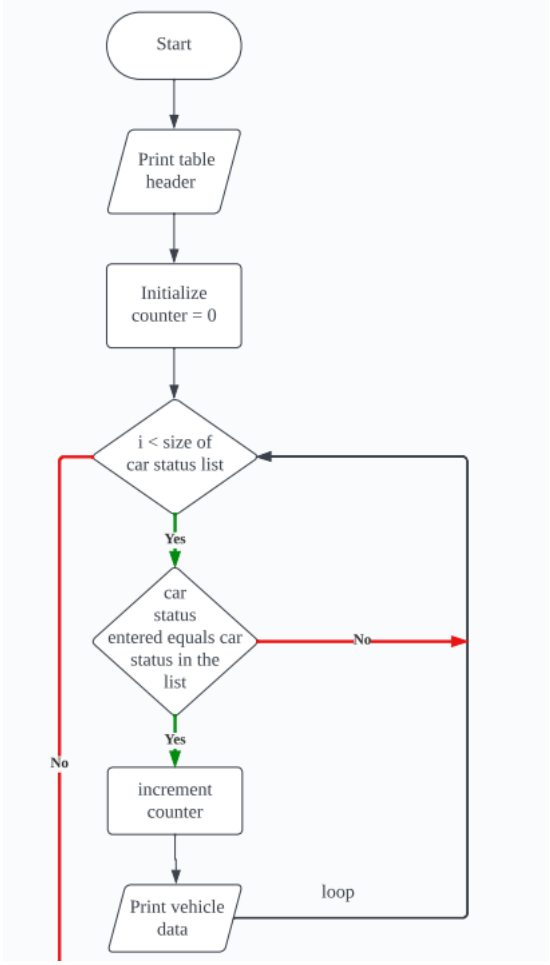


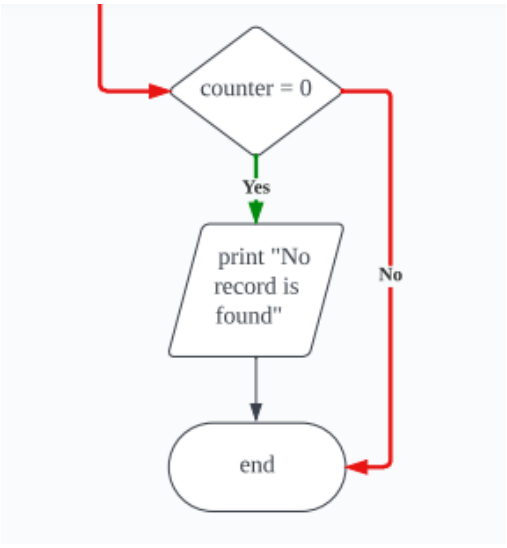
48. managementSearchVehicleByAcquiredPrice(double minimumAcquiredPrice, double maximumAcquiredPrice)



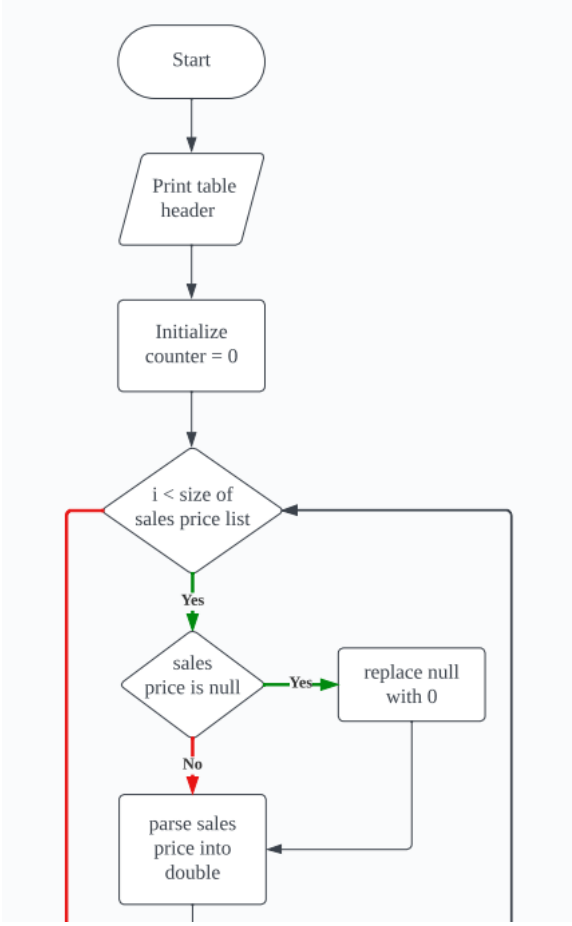


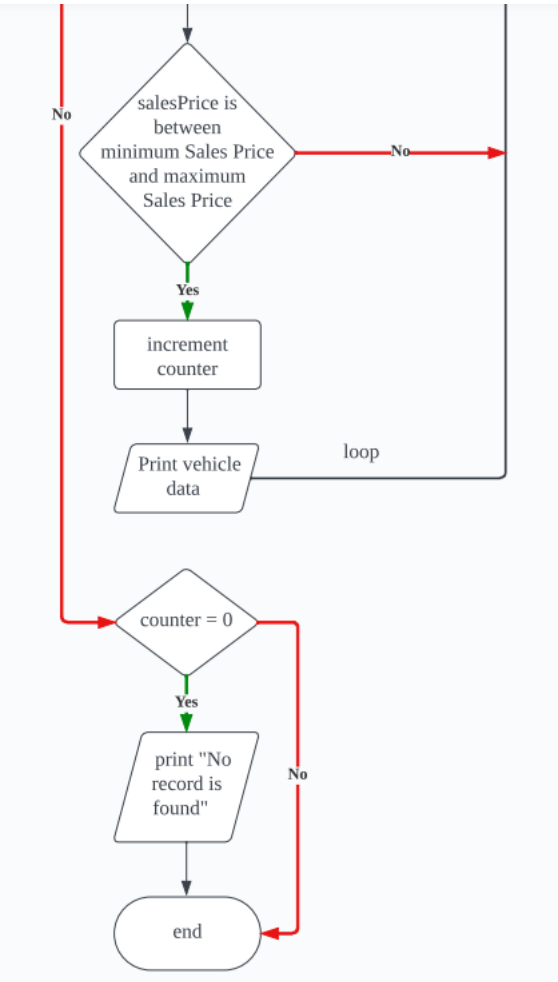
49. managementSearchVehicleByCarStatus(String carStatus)



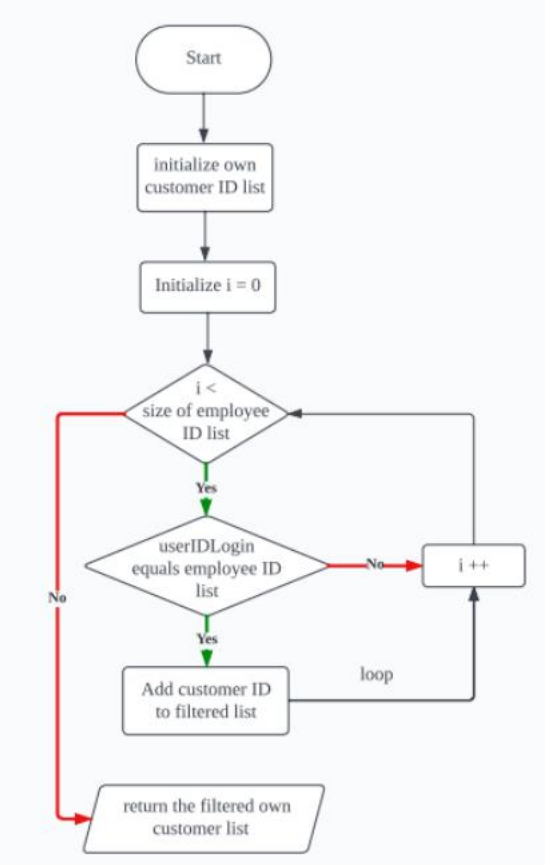


50. managementSearchVehicleBySalesPrice(double minimumSalesPrice, double maximumSalesPrice)

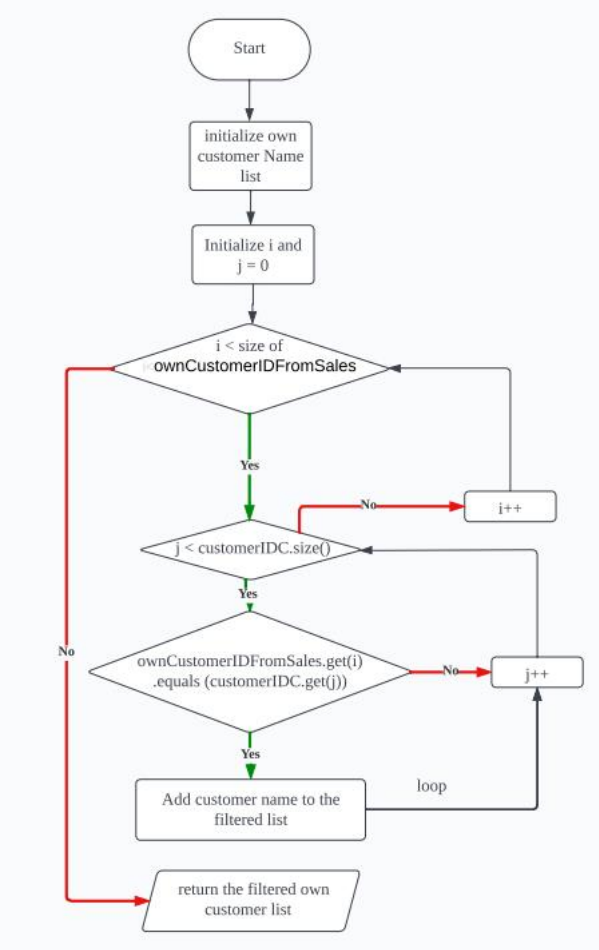



salesPrice is between minimum Sales Price and maximum Sales Price
No
No
Yes
increment counter
Print vehicle data
loop
counter = 0
Yes
print "No record is found"
No
end

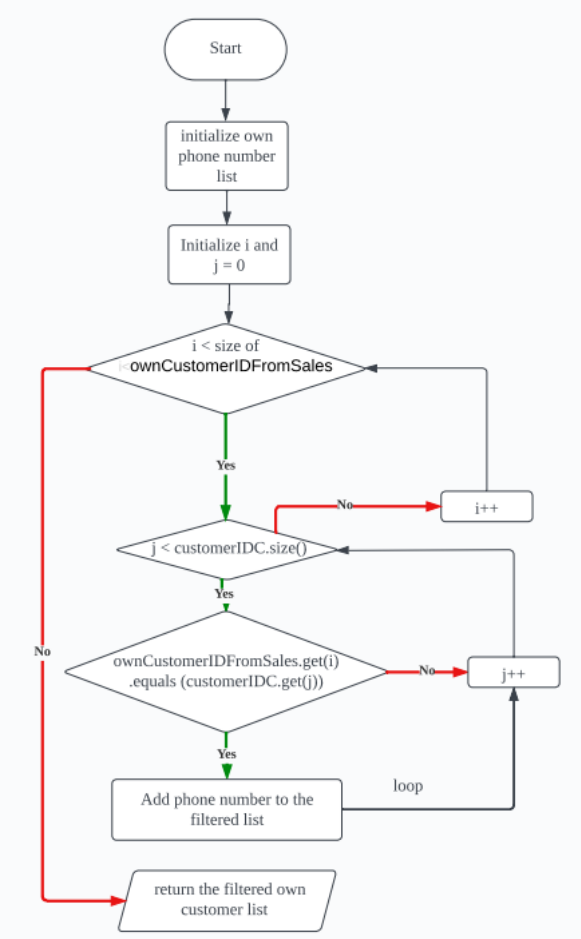
51. filterOwnCustomerIDFromSales()



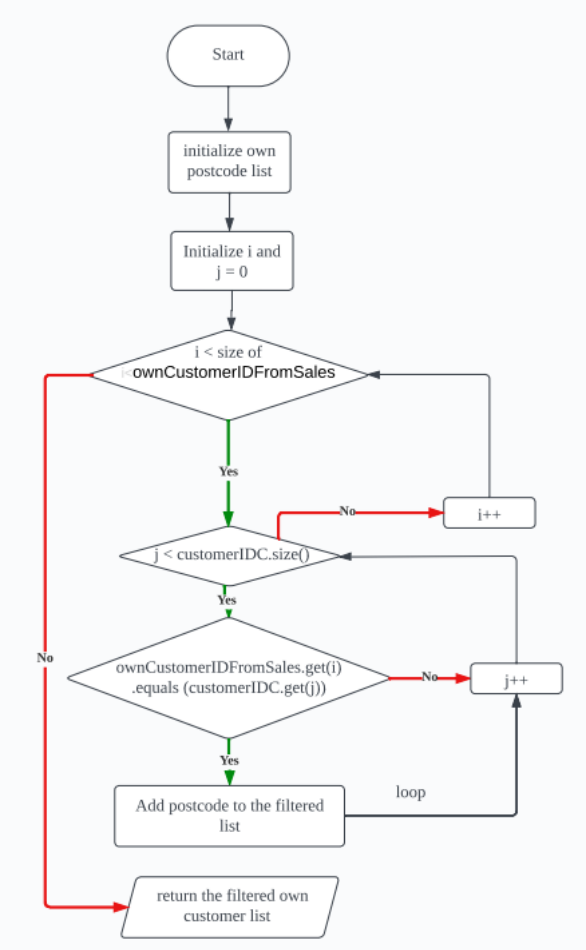
52. filterOwnCustomerName (List<String> ownCustomerIDFromSales)



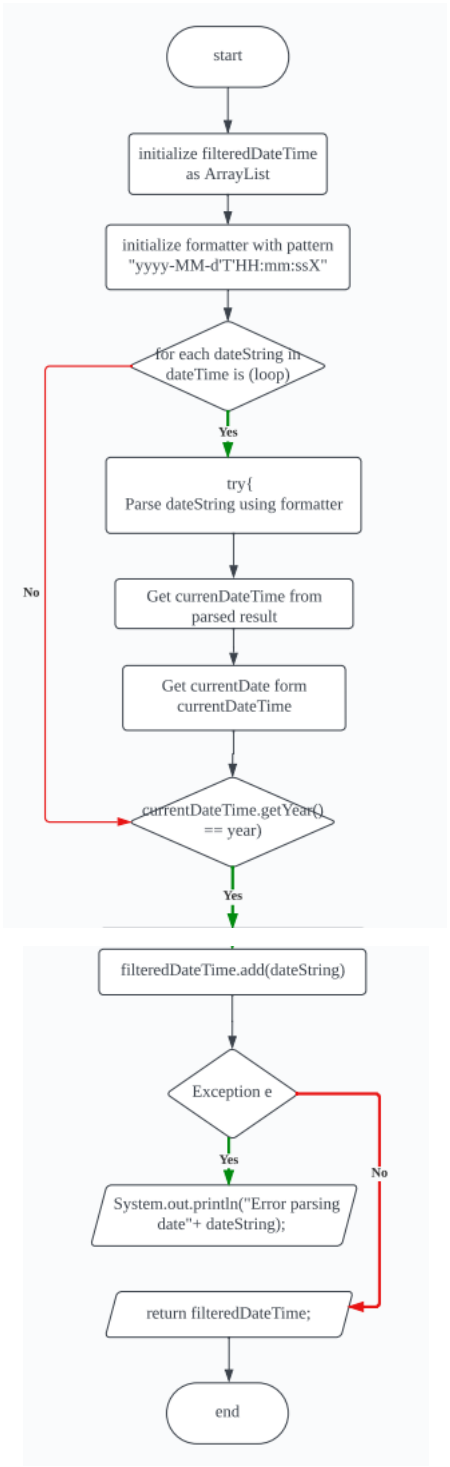
53. filterOwnPhoneNumber (List<String> ownCustomerIDFromSales)



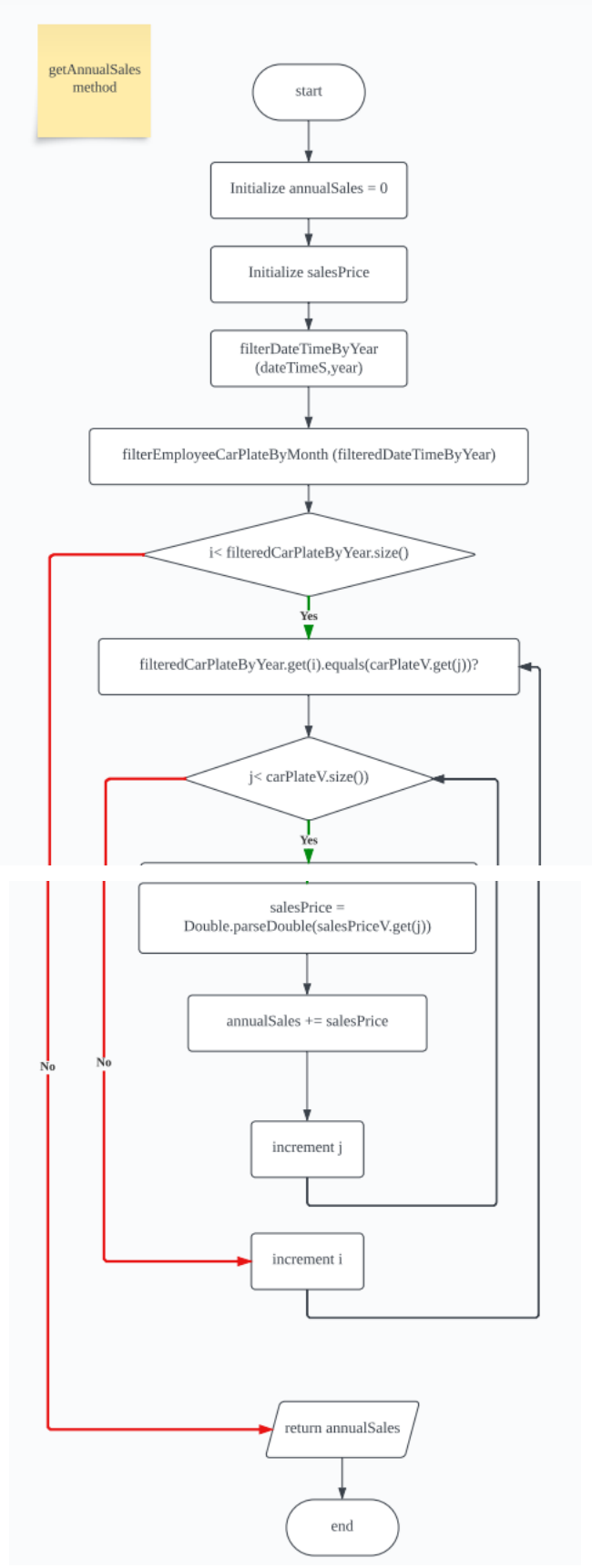
54. filterOwnPostcode (List<String> ownCustomerIDFromSales)



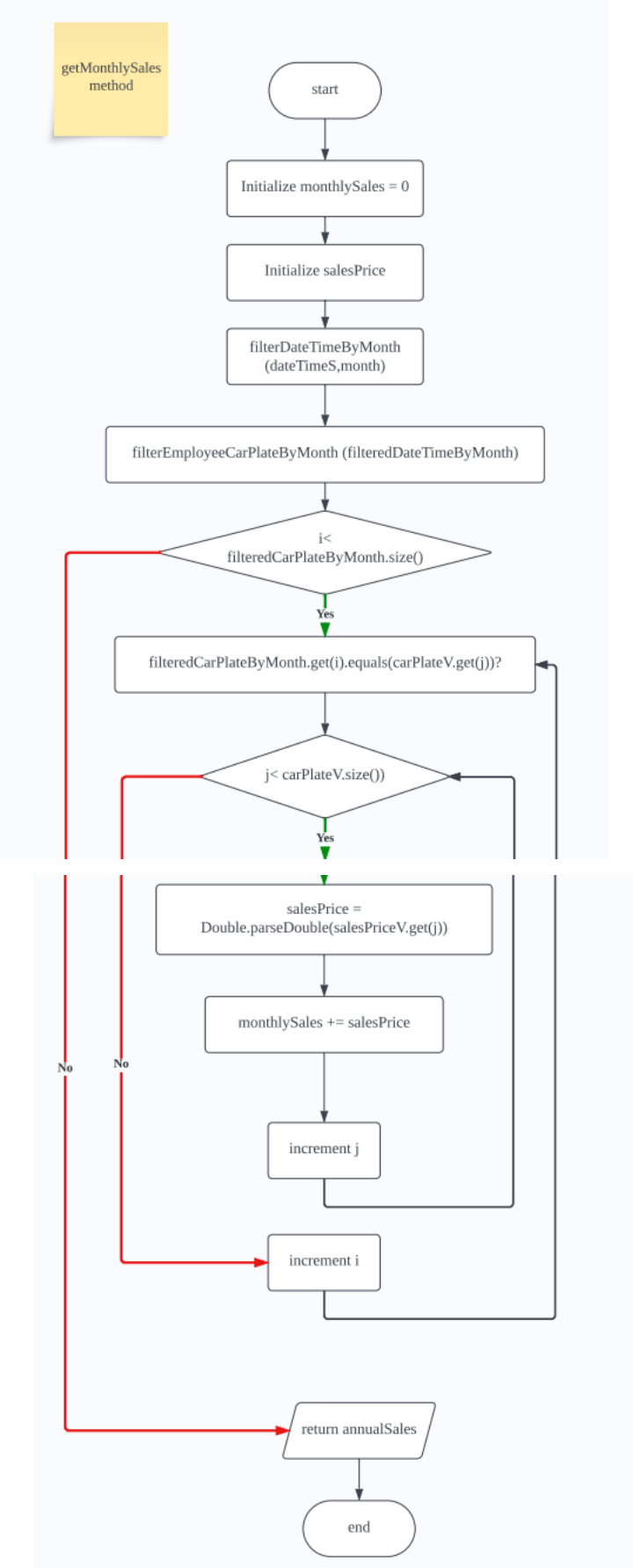
55. filterDateTimeByYear(List<String> dateTime, int year)



56. getAnnualSales(int year)



57. getMonthlySales(int year, int month)



MODULES:

1. **User Authentication Module:**
    - *userLogin*: Validates the employee ID and password for user login.
    - *generateEmployeeID*: Generates a unique employee ID during registration.
    - *checkExistingUserByName*: Checks if an employee name already exists.

2. **User Registration Module:**
    - *userRegister*: Allows users to register with a secret key, creating a new employee account.

3. **Data Loading and Writing Module:**
    - *loadUserData*: Loads employee data from a file during system startup.
    - *writeToFile*: Writes new data to various CSV files (employee, customer, sales, vehicle).

4. **Employee Management Module:**
    - *addManagement*: Adds a new employee to the system with a generated employee ID.
    - *checkManagement*: Checks if an employee has management privileges.

5. **Customer Management Module:**
    - *generateCustomerID*: Generates a unique customer ID during registration.
    - *checkExistingCustomer*: Checks if a customer name already exists.
    - *addCustomer*: Adds a new customer to the system with a generated customer ID.

6. **Sales Management Module:**
    - *generateSalesID*: Generates a unique sales ID for each transaction.
    - *generateDateTime*: Generates the current date and time.
    - *addSales*: Records a new sales transaction with details such as car plate, customer ID, and employee ID.
    - *calSalesAmount*: Calculates the total sales amount for a given employee.

7. **Vehicle Management Module:**
    - *addVehicle*: Adds a new vehicle to the system with details such as car plate, model, and prices.

8. **View Data Module:**
    - *viewAllEmployee*, *viewAllCustomer*, *viewAllSales*, *viewAllVehicle*: Displays all employee, customer, sales, and vehicle data respectively.

9. **Filtering and Analysis Module:**
    - Methods like *filterEmployeeDateTimeByMonth*, *filterEmployeeCarPlateByMonth*, etc., perform data filtering based on specific criteria.

10. **User Interface Module:**
    - The main method contains a console-based user interface for interaction, providing options for login, registration, and various management tasks.

EXPLANATION BY PART OF EACH FEATURE:

a. User Login & Registration:

i. Load User Data:

- *loadUserData()* is called, presumably to load any existing user data.

ii. Display Options:

- The program then prints a menu for user registration and login based on whether a user is already logged in or not.

iii. User Login Status Check:

- Check if the user is already logged in (*loggedIn* variable).

iv. User Logged In:

- If the user is logged in:
  - ✓ Display a welcome message and options to log out, exit, or skip.

- If the user chooses to log out (case 1):
  - Set *loggedIn* to false.
  - Display a logged-out message.
  - Asks if the user wants to log in again or exit.
    - If yes, prompts for employee ID and password and calls *userLogin.*
    - If no, exits the program.

- If the user chooses to exit (case 2), the program exits.
- If the user chooses to skip (case 3), nothing happens.

v. User Not Logged In:

- If the user is not logged in:
  - Display options to register, log in, or exit.

- If the user chooses to register (case 1), calls *userRegister().*
  - After registration, prompts for employee ID and password and calls *userLogin.*

- If the user chooses to log in (case 2):
  - Prompts for employee ID and password and calls *userLogin.*

- If the user chooses to exit (case 3), the program exits.

vi. Saving Login Status:

- After successful login or logout, it calls *saveLoginStatus* to save the login status.

Methods Used

i. ***loadUserData()*** **Method:**

- It reads each line of the file, extracts employee details (ID, name, status, password), and adds them to the *userDatabase.*
- The method uses a Scanner to read data from the "employee.csv" file.

ii. ***writeToFile(String filepath, String newData)*** **Method:**

- Appends new data (*newData*) to the specified file (*filepath*).
- It creates a BufferedWriter to write data to the file and appends a new line with the provided data.

iii. ***userLogin(String employeeID, String password)*** **Method:**

- Validates user login credentials and saves login status.
- Checks if the provided employee ID and password match an entry in *userDatabase*. If yes, it prints a success message and saves the login status. Otherwise, it prints a failure message and exits the program.

iv. ***userRegister()*** **Method:**

- Registers a new user by collecting information, generating an ID, and storing the data in "employee.csv".
- Asks for a secret key, verifies it, takes user input for name and password, generates an ID, writes the data to the file, and updates relevant data structures.

v. ***saveLoginStatus(String employeeID)*** **Method:**

- Saves the current user's login status to "user login status.txt".
- Writes the provided employee ID to the file, indicating a successful login.

b. <u>Import Data:</u>

i. Create Array Lists to Store the Records from All CSV Files:

CSV Files	Array Lists
employee.csv	*a. employeeIDE* *b. employeeNameE* *c. employeeStatusE* *d. passwordE*
customer.csv	*a. customerIDC* *b. customerNameC* *c. phoneNumberC* *d. postcodeC*
sales.csv	*a. salesIDS* *b. dateTimeS* *c. carPlateS* *d. customerIDS* *e. employeeIDS*
vehicle.csv	*a. carPlateV* *b. carModelV* *c. acquiredPriceV* *d. carStatusV* *e. salesPriceV*

ii. Read the CSV Files:

- The CSV files are by using *BufferedReader* and *FileReader* classes which are in the java.io package.
- The records are read line by line until the end of the CSV Files
- The line will be split into different columns based on the number of commas since the files are in CSV format.

iii. Add the records:

- This function is for management-level employee to add more data of management-level employee into CSV file by using *writeToFile* method.

- The columns are added to their respective array lists for further usage in the source code.

iv. Error Handling

- *FileNotFoundException e* – To ensure the specified file path which will be read is correct and can be accessible.
- *IOException e* – A message will be displayed if there is any input or outpur error.

Method Used

i. *writeToFile* method

- This method is used to write any new data into a specific CSV file.
- The method receives a file path and a new record which are both string types.
- The new data is appended to that particular file path which the method receives.

c. Entering New Data:

i. Management Level Employee Addition:

- Checks if the logged-in user has management-level privileges (*checkManagement(userIDLogin) == true*).
- If true, presents options to the user:
  - ✓ Option 1: Add new management level employee(s).
  - ✓ Option 2: Skip.
- If the user chooses to add new management level employees (Option 1), the program prompts for the number of new management employees (*addNum*). It then iteratively calls the *addManagement()* method to input the details for each new employee.

ii. Customer Data Addition:

- Presents options to the user:
  - ✓ Option 1: Add new customer data.
  - ✓ Option 2: Skip.
- If the user chooses to add new customer data (Option 1), the program prompts for the number of new customers (*addNum*). It then iteratively calls the *addCustomer()* method to input the details for each new customer.

iii. Sales Data Addition:

- Presents options to the user:
  - ✓ Option 1: Add new sales data.
  - ✓ Option 2: Skip.
- If the user chooses to add new sales data (Option 1), the program prompts for the number of new sales (*addNum*). It then iteratively calls the *addSales()* method to input the details for each new sale.

iv. Vehicle Data Addition:

- Presents options to the user:
    - ✓ Option 1: Add new vehicle data.
    - ✓ Option 2: Skip.
- If the user chooses to add new vehicle data (Option 1), the program prompts for the number of new vehicles (*addNum*). It then iteratively calls the *addVehicle()* method to input the details for each new vehicle.

Methods Used

i. *addManagement()* Method:

- Collects information for adding a new management-level employee.
- Prompts the user to input the employee name and password.
- Checks if the employee name already exists in the system. If so, it displays an error message and terminates the program.
- If the employee name is unique, generates a new employee ID and constructs a CSV-formatted string with the employee details.
- Appends the data to the "employee.csv" file and updates corresponding data structures and hash maps.
- Outputs a success message with the generated employee ID.

ii. *addCustomer()* Method:

- Gathers information for adding a new customer.
- Prompts the user to input the customer name, phone number, and postcode.
- Checks if the customer name already exists. If so, it displays an error message and terminates the program.
- If the customer name is unique, generates a new customer ID and constructs a CSV-formatted string with the customer details.
- Appends the data to the "customer.csv" file and updates corresponding data structures.
- Outputs a success message with the generated customer ID.

iii. *addSales()* Method:

- Captures information for adding a new sales record.
- Automatically generates the date and time.
- Prompts the user to input the car plate, customer ID, and employee ID (taken from the logged-in user).
- Generates a new sales ID and constructs a CSV-formatted string with the sales details.
- Appends the data to the "sales.csv" file and updates corresponding data structures.
- Outputs a success message with the generated sales ID.

iv. *addVehicle()* Method:

- Collects information for adding a new vehicle.
- Prompts the user to input the car plate, car model, acquired price, car status, and sales price.
- Constructs a CSV-formatted string with the vehicle details.
- Appends the data to the "vehicle.csv" file and updates corresponding data structures.
- Outputs a success message.

d. Information Viewing & Access Control:

i. Customer Data

- Usage of multiple methods are used to retrieve specific information about customers based on the employee's ID.
- Details obtained will then be stored in different array lists (*filteredCustomerIDByEmployeeID*, *filteredCustomerNameByCustomerID*, *filteredPhoneNumberByCustomerID*, *filteredCustomerPostcodeByCustomerID*)

ii. Checking User Permissions

- The code checks whether the logged-in user has management permissions or not. Based on this check, the subsequent actions will vary.

Displaying Menu Options and Handling User Choice

iii. Sales Employee Access

- The program displays options for viewing specific data
    - Option 1: View own customer data
    - Option 2: View own sales record
    - Option 3: View all vehicle data
    - Option 4: Skip

It then captures the user's choice and proceeds accordingly to display the chosen data or execute the related action.

iv. Management Employee Access

- The program displays different options for viewing specific data
    - Option 1: View all customer records
    - Option 2: View all sales data
    - Option 3: View all vehicle data
    - Option 4: Skip

It then captures the user's choice and proceeds accordingly to display the chosen data or execute the related action.

Methods Used:

Filter Methods

i. ***filterCustomerIDByEmployeeID(String employeeID)***

- **Input** **:** Takes an *employeeID* as a parameter.
- **Functionality :** Filters customer IDs based on the provided employeeID.
- **Process** **:** Iterates through *employeeIDS* list, compares *employeeID*, and adds corresponding *customerIDS* to the resulting list.

ii. ***filterCustomerNameByCustomerID(List<String> filteredCustomerIDByEmployeeID)***

- **Input** **:** Takes a list of filtered customer IDs as input.
- **Functionality :** Filters customer names based on the provided list of *filteredCustomerIDByEmployeeID.*
- **Process** **:** Matches *filteredCustomerIDByEmployeeID* with *customerIDC* list and adds corresponding *customerNameC* to the resulting list.

iii. ***filterPhoneNumberByCustomerID(List<String> filteredCustomerIDByEmployeeID)***

- **Input** **:** Takes a list of filtered customer IDs as input.
- **Functionality :** Filters phone numbers based on the provided list of *filteredCustomerIDByEmployeeID.*
- **Process** **:** Matches *filteredCustomerIDByEmployeeID* with *customerIDC* list and adds corresponding *phoneNumberC* to the resulting list.

iv. filterCustomerPostcodeByCustomerID(List<String> filteredCustomerIDByEmployeeID)

- **Input** **:** Takes a list of filtered customer IDs as input.
- **Functionality :** Filters customer postcodes based on the provided list of *filteredCustomerIDByEmployeeID*.
- **Process** **:** Matches *filteredCustomerIDByEmployeeID* with *customerIDC* list and adds corresponding *postcodeC* to the resulting list.

View Methods

v. viewAllVehicle()

- **Functionality :** Displays all vehicle data.
- **Process** **:** Prints vehicle details including car plate, model, acquired price, status, and sales price from respective lists.

vi. viewAllCustomer()

- **Functionality :** Displays all customer data.
- **Process** **:** Prints customer details including ID, name, phone number, and postcode from respective lists.

vii. viewAllSales()

- **Functionality :** Displays all sales records.
- **Process** **:** Prints sales record details like ID, date & time, car plate, customer ID, and employee ID from respective lists.

viii. viewAllEmployee()

- **Functionality :** Displays all employee data.
- **Process** **:** Prints employee details including ID, name, status, and password from respective lists.

Overall Functionality:

This code consists of functions that filter and display data associated with employees, customers, sales records, and vehicles. It filters customer data based on employee ID, retrieving customer IDs and subsequently filtering customer names, phone numbers, and postcodes. Depending on user access level, it presents different menu options for users to view specific data sets. For non-management users, it enables access to their own customer data, sales records, all vehicle data, or skipping. Management users have broader access, allowing them to view all employee, customer, sales, and vehicle data, or skip as needed. This code facilitates selective data access and display based on user roles and preferences within a system.

e. <u>Additional Feature 1: Sales Insights</u>

✓ **Access Control:**

- Purpose: Determine whether the user has management-level access.
- Components:
  - *checkManagement* checks if the user, identified by *userIDLogin*, has management-level access.
  - If the user lacks management access, a message is displayed indicating the user's lack of access to sales insights.

✓ **Sales Insights Menu for Management:**

- Purpose: Provide management-level employees with options to gain insights into sales data.
- Components:
  - ✓ Menu Options:
    - **Annual Sales:**
      - Prompts the user to input a year.
      - Calls *getAnnualSales(year)* to calculate and display the total annual sales.
    - **Monthly Sales:**
      - Prompts the user to input a month.
      - Displays monthly sales, including average monthly sales, using *getMonthlySales(i+1)* for each month.
    - **Number of Cases Growth:**
      - Calculates the number of sales cases for each month using *filterDateTimeByMonth* and *LineChart* classes.

- Uses *LineChart* to visualize the growth in the number of cases.

✓ **Error Handling:**

- Purpose: Handle invalid user choices.
- Components:
  - If the user makes an invalid choice, an error message is displayed, and the program exits.

✓ **Additional Information:**

- Data Sources:
  - The code appears to rely on various lists such as *dateTimeS*, *carPlateV*, *salesPriceV*, etc., presumably containing sales-related data.
- Charting:
  - The use of *SwingUtilities.invokeLater(() -> new LineChart("NUMBER OF CASES"))* suggests the creation of a line chart for visualizing the number of cases growth.
- Data Manipulation:
  - Methods like *getAnnualSales*, *getMonthlySales*, and filtering methods are used to process and retrieve relevant sales data.

Overall Functionality:

The code is part of a larger system where management-level employees can access insights into sales data, including total annual sales, monthly sales, and the growth in the number of cases over different months. The use of charts indicates a visual representation of data for better analysis and understanding. Error handling ensures a smooth user experience even in the case of invalid choices.

Methods Used:

1. ***checkManagement* Method**
    - Determine if the user has management-level access.
    - This method checks the user access level against predefined criteria. (e.g., role or permissions)

2. ***getAnnualSales* Method**
    - Filter date-time strings for the specified year using *filterDateTimeByYear*.
    - Filters car plates based on the data-time strings corresponding to the year using *filterEmployeeCarPlateBymonth*.
    - Iterates through the filtered car plates and the entire dataset to match and accumulate sales prices.

3. ***getMonthlySales* Method**
    - Filters date-time strings for the specified month using *filterDateTimeByMonth*.
    - Filters car plates based on the date-time strings corresponding to the month using *filterEmployeeCarPlateByMonth*.
    - Iterates through the filtered car plates and the entire dataset to match and accumulate sales prices.

4. ***filterDateTimeByYear* Method**
    - Takes a list of date-time strings *dateTime* and an integer *year* as input.
    - Uses *DateTimeFormatter* to parse date-time strings
    - Iterates through date-time strings, extracts the year, and adds matching strings to the filtered list.
    - Output: returns a list of date-time strings filtered by the specified year.

5. ***filterOwnPostcode*** **Method**
    - Retrieves postcodes associated with the customer IDs.
    - *ownCustomerIDFromSales*: Lists of customer IDs.
    - Matches customer IDs from the sales data *ownCustomerIDFromSales* with the corresponding postcodes.
    - Returns a list of postcodes associated with the provided customer IDs.

6. ***filterOwnPhoneNumber*** **Method**
    - Retrieve phone numbers associated with customer IDs.
    - Matches customer IDs from the sales data *ownCustomerIDFromSales* with the corresponding phone numbers (*phoneNumberC*).
    - Returns a list of phone numbers associated with the provided customer IDs.

7. ***filterDateTimeByMonth*** **Method**
        - Takes a list of date-time strings and filters them based on the specified month.
        - Uses the *DateTimeFormatter* and *LocalDateTime* to parse and compare month values.
        - Returns a list of date-time strings filtered by the specified month.

8. ***filterEmployeeCarPlateByMonth*** **Method**
    - Takes a list of date-time strings (*filteredDateTime*) and retrieves the corresponding car plates.
    - Compares the filtered date-time strings with the original date-time strings (*dateTimeS*).
    - Adds the car plates to a new list (*filteredEmployeeCarPlateByMonth*).
    - Returns the filtered list of car plates.

f. <u>Additional Feature 2: Salary Calculation</u>

i. Check Management Privileges:

- Verifies if the logged-in user has management-level privileges using *checkManagement(userIDLogin)*.
- If true, the user is allowed to proceed; otherwise, a message is displayed, and the program exits.

ii. Input Employee ID and Month:

- Prompts the user to input the employee ID for whom the salary and bonus are to be calculated.
- Checks if the provided employee ID exists in the system using *checkExistingUserByID(employeeID)*.

iii. Calculate Sales Metrics:

- Asks for the month in numeric form and filters sales records for the specified month.
- Calculates various metrics such as the total number of records, total sales amount in the month, and total sales amount overall.

iv. Determine Salary and Bonus:

- Based on whether the employee is a management-level employee, calculates the basic salary, allowance, commission, and bonus.
- For non-management employees, bonus is awarded for exceeding a certain number of sales records or achieving a high total sales amount in a month.
- For management employees, bonus is calculated based on specific ranges of total sales amount in a month.

v. Output Results:

- Prints the calculated employee salary and bonus in a formatted manner.

vi. Error Handling:

- If the employee ID doesn't exist, displays an error message and exits the program.
- If the logged-in user does not have management privileges, inform the user that they don't have access to view employee salary and bonus.

Methods Used

i. *filterDateTimeByMonth* Method:

- Takes a list of date-time strings and filters them based on the specified month.
- Uses the *DateTimeFormatter* and *LocalDateTime* to parse and compare month values.
- Returns a list of date-time strings filtered by the specified month.

ii. *filterEmployeeDateTimeByMonth* Method:

- Takes a list of date-time strings (*filteredDateTime*) and an employee ID.
- Compares the filtered date-time strings with the original date-time strings (*dateTimeS*) and employee IDs (*employeeIDS*).
- Adds the matching date-time strings to a new list (*filteredEmployeeDateTime*) for the specified employee.
- Returns the filtered list.

iii. *filterEmployeeCarPlateByMonth* Method:

- Takes a list of date-time strings (*filteredDateTime*) and retrieves the corresponding car plates.
- Compares the filtered date-time strings with the original date-time strings (*dateTimeS*).
- Adds the car plates to a new list (*filteredEmployeeCarPlateByMonth*).

- Returns the filtered list of car plates.

iv. ***filterEmployeeCarPriceByMonth*** **Method:**

- Takes a list of car plates (*filteredEmployeeCarPlate*) and retrieves the corresponding sales prices.
- Compares the filtered car plates with the original car plates (*carPlateV*).
- Adds the sales prices to a new list (*filteredEmployeeCarPriceByMonth*).
- Returns the filtered list of sales prices.

v. ***calSalesRecord*** **Method:**

- Calculates the total number of sales records for a given employee ID.
- Iterates through the sales records and increments a counter for each record associated with the specified employee ID.
- Returns the total sales record count.

vi. ***calSalesAmountInMonth*** **Method:**

- Takes a list of sales prices (*employeeCarPrice*) and calculates the total sales amount in a month.
- Iterates through the list of sales prices and accumulates the amounts.
- Returns the total sales amount in a month.

vii. ***calSalesAmount*** **Method:**

- Takes an employee ID and calculates the total sales amount for that employee.
- Iterates through the sales records, compares employee IDs, retrieves the corresponding sales prices, and accumulates the amounts.
- Returns the total sales amount for the specified employee ID.

g. <u>Additional Feature 3: Employee Bonus</u>

The program checks if the employee ID entered belongs to a sales or management level employee using the *checkManagement* method which has been used previously. If the employee is a sales level employee, his/her bonus will be based on the total number of sales records or the total sales amount in a month.

- If the total number of sales records of the sales level employee is more than 15 or the total sales amount is more than is RM1 000 000 within a month, the employee will get a bonus of RM 500.
- The bonus of management level employee depends on the total sales amount in a month based on the table below.

Sales Amount in a Month (RM)	Commission Rate (%)
800000.00 or less	1.00
800000.01 – 1600000.00	1.15
1600000.01 – 2500000.00	1.25
2500000.01 or more	1.35

h. Additional Feature 4: Search & Filter

i. Search and Filter Section:

- The code starts with a conditional statement checking if the logged-in user has management privileges (***checkManagement(userIDLogin)***).
- If the user is a manager, they are given the ability to search and filter various records.

ii. Employee Records Search:

- Managers can search for employee records based on ID, name, or status.
- The user is prompted to choose a search criterion and then input the relevant information.

iii. Customer Records Search:

- Managers can search for customer records based on ID, name, phone number, or postcode.
- Similar to employee records, the user selects a search criterion and provides input accordingly.

iv. Sales Records Search:

- Managers can search for sales records based on sales ID, month, car plate, customer ID, or employee ID.
- The user selects a search criterion and provides input accordingly.

v. Vehicle Records Search:

- Managers can search for vehicle records based on car plate, car model, acquired price, car status, or sales price.
- Similar to previous sections, the user selects a search criterion and provides input accordingly.

vi. Employee-Specific Search (Sales Role):

- If the logged-in user is not a manager but a salesperson, they have limited access and can only search for their own records.
- Salespeople can view their own employee records and search for customer and sales records related to their sales.

vii. Filtering Own Customer and Sales Records:

- Salespeople are provided with options to filter their own customer and sales records based on customer ID, name, phone number, postcode, sales ID, month, car plate, customer ID, or employee ID.

viii. Output of Search Results:

- The code includes output formatting for displaying the search results in a tabular format.

ix. Exiting on Invalid Choice:

- If the user enters an invalid choice at any point, the program prints an error message and exits.

Methods Used:

i. ***checkManagement* Method**

- Determine if the user has management-level access
- This method checks the user access level against some predefined criteria. (e.g., role or permissions)

ii. **Employee Search Methods:**

- ***managementSearchEmployeeByEmployeeID(String employeeID)***
  - Searches for and displays employee records based on the provided employee ID.
  - Prints the Employee ID, Employee Name, Employee Status, and Password.

- ***managementSearchEmployeeByEmployeeName(String employeeName)***
  - Searches for and displays employee records based on the provided employee name.
  - Prints the Employee ID, Employee Name, Employee Status, and Password.

- ***managementSearchEmployeeByEmployeeStatus(String employeeStatus)***
  - Searches for and displays employee records based on the provided employee status.
  - Prints the Employee ID, Employee Name, Employee Status, and Password.

iii. Vehicle Search Methods:

- ***managementSearchVehicleBySalesPrice(double minimumSalesPrice, double maximumSalesPrice)***
    - Searches for and displays vehicle records based on the sales price range.
    - Prints Car Plate, Car Model, Acquired Price, Car Status, and Sales Price.

- ***managementSearchVehicleByCarStatus(String carStatus)***
    - Searches for and displays vehicle records based on the provided car status.
    - Prints Car Plate, Car Model, Acquired Price, Car Status, and Sales Price.

- ***managementSearchVehicleByAcquiredPrice(double minimumAcquiredPrice, double maximumAcquiredPrice)***
    - Searches for and displays vehicle records based on the acquired price range.
    - Prints Car Plate, Car Model, Acquired Price, Car Status, and Sales Price.

- ***managementSearchVehicleByCarModel(String carModel)***
    - Searches for and displays vehicle records based on the provided car model.
    - Prints Car Plate, Car Model, Acquired Price, Car Status, and Sales Price.

- ***managementSearchVehicleByCarPlate(String carPlate)***
    - Searches for and displays vehicle records based on the provided car plate.
    - Prints Car Plate, Car Model, Acquired Price, Car Status, and Sales Price.

iv. Sales Search Methods:

- ***managementSearchSalesByEmployeeID(String employeeID)***
    - Searches for and displays sales records based on the provided employee ID.
    - Prints Sales ID, Date and Time, Car Plate, Customer ID, and Employee ID.

- ***managementSearchSalesByCustomerID(String customerID)***
    - Searches for and displays sales records based on the provided customer ID.
    - Prints Sales ID, Date and Time, Car Plate, Customer ID, and Employee ID.

- ***managementSearchSalesByCarPlate(String carPlate)***
    - Searches for and displays sales records based on the provided car plate.
    - Prints Sales ID, Date and Time, Car Plate, Customer ID, and Employee ID.

- ***managementSearchSalesByMonth(int salesMonth)***
    - Searches for and displays sales records based on the provided sales month.
    - Prints Sales ID, Date and Time, Car Plate, Customer ID, and Employee ID.

- ***managementSearchSalesBySalesID(String salesID)***
    - Searches for and displays sales records based on the provided sales ID.
    - Prints Sales ID, Date and Time, Car Plate, Customer ID, and Employee ID.

v. Customer Search Methods:

- ***managementSearchCustomerByPostcode(String postcode)***
    - Searches for and displays customer records based on the provided postcode.
    - Prints Customer ID, Customer Name, Phone Number, and Postcode.

- ***managementSearchCustomerByPhoneNumber(String phoneNumber)***
    - Searches for and displays customer records based on the provided phone number.
    - Prints Customer ID, Customer Name, Phone Number, and Postcode.

- ***managementSearchCustomerByCustomerName(String customerName)***
    - Searches for and displays customer records based on the provided customer name.
    - Prints Customer ID, Customer Name, Phone Number, and Postcode.

- ***managementSearchCustomerByCustomerID(String customerID)***
    - Searches for and displays customer records based on the provided customer ID.
    - Prints Customer ID, Customer Name, Phone Number, and Postcode.

vi. **Filter methods**

- ***filterOwnPostcode(List<String> ownCustomerIDFromSales)*** **Method:**
  - This method filters and returns a list of postcodes corresponding to customer IDs found in the ***ownCustomerIDFromSales*** list.
  - It iterates over the elements in ***ownCustomerIDFromSales*** and, for each customer ID, finds the matching index in the global ***customerIDC*** list to retrieve the corresponding postcode, adding it to the ***ownPostcode*** list.

- ***filterOwnPhoneNumber(List<String> ownCustomerIDFromSales)*** **Method:**
  - Similar to the first method, this one filters and returns a list of phone numbers corresponding to customer IDs found in the ***ownCustomerIDFromSales*** list.
  - It iterates over the elements in ***ownCustomerIDFromSales***, finds the matching index in the global ***customerIDC*** list, and retrieves the corresponding phone number, adding it to the ***ownPhoneNumber*** list.

- ***filterOwnCustomerName(List<String> ownCustomerIDFromSales)*** **Method:**
  - This method filters and returns a list of customer names corresponding to customer IDs found in the ***ownCustomerIDFromSales*** list.
  - Implementation: It iterates over the elements in ***ownCustomerIDFromSales***, finds the matching index in the global ***customerIDC*** list, and retrieves the corresponding customer name, adding it to the ***ownCustomerName*** list.

- ***filterOwnCustomerIDFromSales()*** **Method:**
    - his method filters and returns a list of customer IDs associated with sales records for the currently logged-in user (***userIDLogin***).Compares the filtered car plates with the original car plates (*carPlateV*).
    - It iterates over the global ***employeeIDS*** list and, for each index where the employee ID matches ***userIDLogin***, it retrieves the corresponding customer ID from the global ***customerIDS*** list, adding it to the ***filteredOwnCustomerIDFromSales*** list.Returns the filtered list of sales prices.

- ***filterDateTimeByMonth*** **Method**
    - Takes a list of date-time strings and filters them based on the specified month.
    - Uses the *DateTimeFormatter* and *LocalDateTime* to parse and compare month values.
    - Returns a list of date-time strings filtered by the specified month.

5. SAMPLE OUTPUT OF LECARS SALES MANAGEMENT SYSTEM

Access Level	Sample Output
New Sales Employee	run: USER REGISTRATION AND LOGIN 1. Register 2. Login 3. Exit Enter choice: 1 USER REGISTRATION You need to have a secret key in order to register an account. Enter secret key: ABCDEFG The secret key is correct! You can now register your own account! Enter employee name: Ali bin Mohammad Enter password: ali23456 Registration successful! Your employee ID is : E0016 USER LOGIN Enter employee ID: E0016 Enter password: ali23456 Login successful! 1. Add customer data 2. Skip Enter choice: 1 Number of new customers: 1 Enter customer name: Mary Lee Xin Mei Enter phone number(without "-"):0112314785 Enter postcode: 98000 Successfully added. The Customer ID is C0099 1. Add sales data 2. Skip Enter choice: 1 Number of new sales: 1 Enter car plate: MNP789 Enter customer ID: C0099 Successfully added. Sales ID is A0101 1. Add vehicle data 2. Skip Enter choice: 2 1. View own customer data 2. View own sales record 3. View all vehicle data 4. Skip Enter choice: 2 Sales Record of E0016 SALES ID DATE AND TIME CAR PLATE CUSTOMER ID EMPLOYEE ID A0101 2024-01-01T13:02:37Z MNP789 C0099 E0016 EMPLOYEE SALARY AND BONUS You have no access to view employee salary and bonus. SEARCH AND FILTER Search for employee records 1. View your own employee record 2. Skip Enter choice: 1 Employee records of E0016 EMPLOYEE ID EMPLOYEE NAME EMPLOYEE STATUS PASSWORD E0016 Ali bin Mohammad 0 ali23456

```
Search for own customer records
1. By customer ID
2. By customer name
3. By phone number
4. By postcode
5. Skip
Enter choice: 4
Enter postcode: 98000

Customer records of 98000
CUSTOMER ID         CUSTOMER NAME                    PHONE NUMBER    POSTCODE
C0099               Mary Lee Xin Mei                 0112314785      98000


 Search for own sales records
 1. By sales ID
 2. By month
 3. By car plate
 4. By customer ID
 5. By employee ID
 6. Skip
 Enter choice: 2
 Enter month: 1

 Sales record of month 1
 SALES ID    DATE AND TIME            CAR PLATE      CUSTOMER ID    EMPLOYEE ID
 A0101       2024-01-01T13:02:37Z     MNP789         C0099          E0016

 Search for vehicle records
 1. By car plate
 2. By car model
 3. By acquired price
 4. By car status
 5. By sales price
 6. Skip
 Enter choice: 1
 Enter car plate: mnp789

 Vehicle record of mnp789
 CAR PLATE      CAR MODEL               ACQUIRED PRICE(RM)  CAR STATUS     SALES PRICE(RM)
 MNP789         Ford Explorer           170500              1


 SALES INSIGHTS
 Sorry, you do not have the access to sales insights.
 BUILD SUCCESSFUL (total time: 2 minutes 38 seconds)
```

```
run:
USER REGISTRATION AND LOGIN
E0016, welcome back
1. Log out
2. Exit
3. Skip
Enter choice: 1
Logged out.
Want to login again?
1. Yes
2. No
Enter choice: 2
BUILD SUCCESSFUL (total time: 12 seconds)
```

Sales Employee

```
run:
USER REGISTRATION AND LOGIN
1. Register
2. Login
3. Exit
Enter choice: 2

USER LOGIN
Enter employee ID: E0002
Enter password: j4k5l6m7
Login successful!

1. Add customer data
2. Skip
Enter choice: 2

1. Add sales data
2. Skip
Enter choice: 2

1. Add vehicle data
2. Skip
Enter choice: 1
Number of new vehicle: 1
Enter car plate: QMX8065
Enter car model: Toyota Avanza
Enter acquired price: 60000
Enter car status: 1
Enter sales price: null
Successfully added.
```

```
1. View own customer data
2. View own sales record
3. View all vehicle data
4. Skip
Enter choice: 1

Customer Data of E0002
CUSTOMER ID         CUSTOMER NAME                    PHONE NUMBER    POSTCODE
C0027               LIM WEI HONG                     0167890245      80000
C0025               RAJINDER SINGH                   0145678023      78000
C0031               ALI BIN HASSAN                   0101234689      84000
C0034               TAN KOK WAH                      0134568023      87050
C0033               LEE KOK SENG                     0123456912      86000
C0020               WONG MEI YEE                     0190123467      73050
C0003               LEE WEI MING                     0123456789      56000

EMPLOYEE SALARY AND BONUS
You have no access to view employee salary and bonus.

SEARCH AND FILTER
Search for employee records
1. View your own employee record
2. Skip
Enter choice: 2

Search for own customer records
1. By customer ID
2. By customer name
3. By phone number
4. By postcode
5. Skip
Enter choice: 2
Enter customer name: lee wei ming

Customer records of lee wei ming
CUSTOMER ID         CUSTOMER NAME                    PHONE NUMBER    POSTCODE
C0003               Lee Wei Ming                     0123456789      56000
```

```
Search for own sales records
1. By sales ID
2. By month
3. By car plate
4. By customer ID
5. By employee ID
6. Skip
Enter choice: 4
Enter customer ID: C0003

Sales record of C0003
SALES ID    DATE AND TIME           CAR PLATE      CUSTOMER ID    EMPLOYEE ID
A0098       2023-10-16T12:00:21Z    YUI607         C0003          E0002

Search for vehicle records
1. By car plate
2. By car model
3. By acquired price
4. By car status
5. By sales price
6. Skip
Enter choice: 4
Enter car status: 0

Vehicle record of status0
CAR PLATE      CAR MODEL               ACQUIRED PRICE(RM)  CAR STATUS     SALES PRICE(RM)
ABC123         Honda Civic             90000               0              95000
XYZ789         Toyota Camry            120000              0              126000
MNO456         Nissan Almera           80000               0              85600
PQR234         Mazda 3                 110000              0              115600
GHI567         Ford Focus              100000              0              107620
JKL678         Chevrolet Cruze         95000               0              99750
STU901         Kia Forte               85000               0              89500
VWX234         Honda Accord            130000              0              147000
CDE345         Mitsubishi Lancer       90000               0              95550
FGH456         Nissan Sylphy           100000              0              105550
IJK567         Mazda 6                 140000              0              148800
LMO678         Ford Fiesta             80000               0              85500
NOP789         Chevrolet Malibu        120000              0              127000
QRS901         Kia Optima              110000              0              113500
TUV012         Honda City              85000               0              89150
WXY123         Mitsubishi Attrage      75000               0              78050
ZAB234         Nissan Teana            130000              0              135400
CDF345         Mazda CX-5              150000              0              159900
GHI456         Ford Escape             140000              0              145670
JLM567         Chevrolet Captiva       130000              0              137000
MNO678         Ford Fiesta             80100               0              85000
RST901         Mitsubishi ASX 130000 0 136700              0              153000
ZBC234         Nissan X-Trail          150300              0              157000
FEG456         Mazda CX-9              180400              0              198800
QRT901         Chevrolet Traverse      160600              0              168000
UVW012         Kia Sorento             150700              0              154300
XYZ123         Honda Odyssey           160800              0              169000
DEF345         Honda Civic             90100               0              96500
HJK456         Toyota Camry            121100              0              126300
LNO567         Nissan Almera           80100               0              83500
PQR678         Mazda 3                 111200              0              115500
STU789         Ford Focus              100300              0              105000
WXY901         Chevrolet Cruze         95400               0              99750
ZDE012         Kia Forte               85500               0              89000
FGL123         Honda Accord            130600              0              140000
JMO234         Mitsubishi Lancer       90700               0              94500
TWX456         Mazda 6                 140900              0              147000
ZGH567         Kia Optima              111000              0              115500
BKL678         Honda City              85100               0              89250
BFG678         Kia Optima              110200              0              115500
DHI789         Honda City              85200               0              89250
HLM012         Mazda CX-5              150600              0              157500
LPR234         Chevrolet Captiva       130100              0              136500
NST345         Kia Sportage            120300              0              126000
PVW456         Nissan X-trail          150500              0              157500
RCD123         Mazda CX-9              180700              0              189000
VGH345         Chevrolet Traverse      160100              0              168000
ZLM567         Honda Odyssey           160500              0              169500
BNO678         Mitsubishi Grandis      140700              0              151800
FST901         Mitsubishi ASX          130200              0              136500
HUV012         Nissan X-Trail          150400              0              160000
```

```
JWX123          Mazda CX-9              180600          0               189000
LZY234          Ford Explorer           170800          0               182100
NAB345          Chevrolet Traverse      160000          0               167700
RGH567          Honda Odyssey           160000          0               168800
VLM789          Honda CR-V              140000          0               158800
XNO901          Mitsubishi ASX          130000          0               136500
BST123          Mazda CX-9              180000          0               189000
DUV234          Ford Explorer           170000          0               178000
FXY345          Chevrolet Traverse      160000          0               168000
JEF567          Honda Odyssey           160000          0               168000
NIJ789          Honda Civic             85000           0               89250
PKL901          Toyota Camry            90000           0               94500
RNO012          Nissan Almera           80000           0               84000
TQR123          Mazda 3                 110000          0               115500
VST234          Ford Focus              100000          0               105000
WUV345          Chevrolet Cruze         95000           0               99750
YBC456          Kia Forte               85000           0               89250
JKT567          Mitsubishi Lancer       90000           0               94500
BNG789          Mazda 6                 140000          0               147000
XYP4567         Toyota Yaris            70940           0               75000
MVC901          Perodua Bezza           42550           0               47000
LGH2345         Toyota Avanza           73800           0               76700
WFD678          Perodua Myvi            41200           0               44000
ZTR3456         Honda Civic             109000          0               115900
NML5678         Nissan Almera           80000           0               85300
CDE304          Mazda CX5               132000          0               140000
VBN6789         Hyundai Kona            115000          0               132000
QWE405          Kia Picanto             50000           0               56250
YUI607          Honda Jazz              72800           0               73900
OPR7890         Toyota Vios             74200           0               76500
GFD908          Perodua Kancil          5000            0               5200
PQR7890         Perodua Ativa           66100           0               66500
MNP7892         Honda City              92900           0               104000
BAC2343         Toyota Vios             93600           0               107000
NRS3454         Proton X50              94800           0               105500
FJK9015         Nissan Almera           87900           0               91800
JNO1236         Mazda CX-5              138000          0               147850
TEF2347         Mitsubishi Xpander      91400           0               100000
XIJ4568         Kia Seltos              115900          0               121600
DQR7899         Hyundai Kona            123900          0               128000
PDE4560         Mercedes-Benz C200      259900          0               275000
TIJ6781         BMW 3 Series            248800          0               265000
ZQR0122         Audi A4                 206500          0               217000
HBC4563         Volvo XC40              241500          0               270000
LGH6784         Peugeot 3008            150900          0               160000
ZDE5675         Subaru XV               117800          0               123050
RQZ1230         Renault Captur          105300          0               111000
HJK1026         Ford Ranger             90900           0               98500
ASD1237         Isuzu D-Max             88600           0               92000


SALES INSIGHTS
Sorry, you do not have the access to sales insights.
BUILD SUCCESSFUL (total time: 3 minutes 50 seconds)
run:
USER REGISTRATION AND LOGIN
E0002, welcome back
1. Log out
2. Exit
3. Skip
Enter choice: 2
Goodbye!
BUILD SUCCESSFUL (total time: 5 seconds)
```

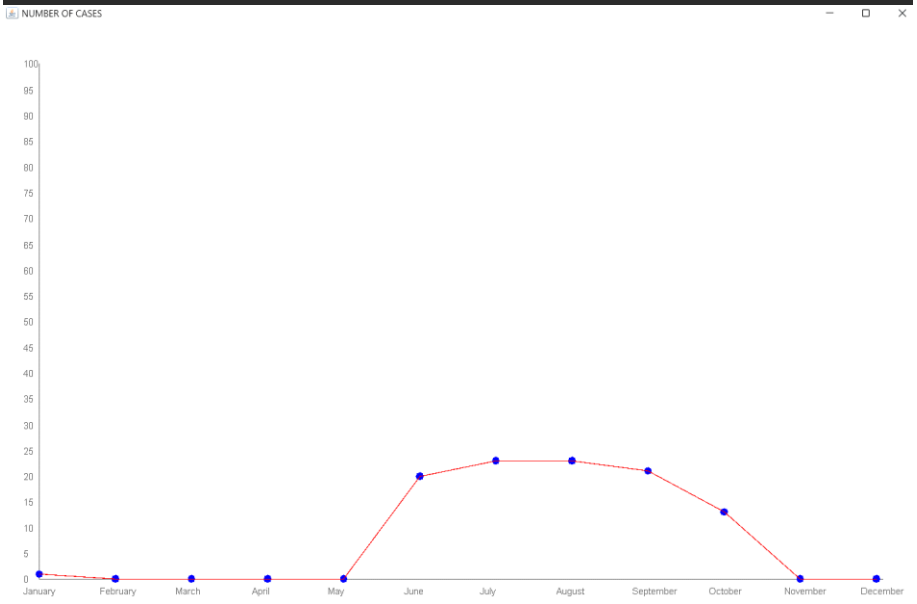
Management Employee

```
run:
USER REGISTRATION AND LOGIN
E0002, welcome back
1. Log out
2. Exit
3. Skip
Enter choice: 1
Logged out.
Want to login again?
1. Yes
2. No
Enter choice: 1

USER LOGIN
Enter employee ID: E0001
Enter password: 3f7g9h2k
Login successful!

1. Add new management level employee(s)
2. Skip
Enter choice: 1
Number of new management employees: 1
Enter employee name: Siti Amelia binti Mohammad
Enter password: sitiame3412
Successfully added.
The employee ID is E0017


1. Add customer data
2. Skip
Enter choice: 2

1. Add sales data
2. Skip
Enter choice: 2

1. Add vehicle data
2. Skip
Enter choice: 2

1. View all employee data
2. View all customer record
3. View all sales data
4. View all vehicle data
5. Skip
Enter choice: 3

Sales Record
SALES ID    DATE AND TIME            CAR PLATE      CUSTOMER ID    EMPLOYEE ID
A0001       2023-06-01T02:38:22Z     ABC123         C0005          E0007
A0002       2023-06-02T10:40:15Z     XYZ789         C0012          E0010
A0003       2023-06-05T18:47:22Z     MNO456         C0034          E0004
A0004       2023-06-06T20:49:35Z     PQR234         C0076          E0009
A0005       2023-06-07T22:51:48Z     GHI567         C0045          E0012
A0006       2023-06-09T00:54:01Z     JKL678         C0089          E0006
A0007       2023-06-10T02:56:14Z     STU901         C0067          E0014
A0008       2023-06-13T09:02:53Z     VWX234         C0023          E0008
A0009       2023-06-14T11:05:06Z     CDE345         C0056          E0011
A0010       2023-06-15T13:07:19Z     FGH456         C0098          E0005
A0011       2023-06-16T15:09:32Z     IJK567         C0078          E0015
A0012       2023-06-17T17:11:45Z     LMO678         C0032          E0001
A0013       2023-06-18T19:13:58Z     NOP789         C0006          E0013
A0014       2023-06-22T01:20:37Z     QRS901         C0027          E0002
A0015       2023-06-23T03:22:50Z     TUV012         C0065          E0003
A0016       2023-06-24T05:25:03Z     WXY123         C0084          E0010
A0017       2023-06-26T09:29:29Z     ZAB234         C0091          E0004
A0018       2023-06-27T11:31:42Z     CDF345         C0059          E0009
A0019       2023-06-29T15:36:08Z     GHI456         C0028          E0012
A0020       2023-06-30T20:49:35Z     JLM567         C0036          E0006
A0021       2023-07-01T22:51:48Z     MNO678         C0044          E0014
A0022       2023-07-03T00:54:01Z     PQR7890        C0021          E0008
A0023       2023-07-04T02:56:14Z     RST901         C0017          E0011
A0024       2023-07-05T04:58:27Z     ZBC234         C0090          E0005
A0025       2023-07-07T09:02:53Z     FEG456         C0074          E0015
A0026       2023-07-08T11:05:06Z     MNP7892        C0030          E0001
A0027       2023-07-09T13:07:19Z     QRT901         C0051          E0013
A0028       2023-07-10T15:09:32Z     UVW012         C0025          E0002
A0029       2023-07-11T17:11:45Z     XYZ123         C0047          E0003
A0030       2023-07-13T21:16:11Z     BAC2343        C0063          E0010
A0031       2023-07-14T23:18:24Z     DEF345         C0082          E0004
```

```
A0032       2023-07-17T03:22:50Z    HJK456      C0094       E0009
A0033       2023-07-18T05:25:03Z    LNO567      C0072       E0012
A0034       2023-07-19T07:27:16Z    PQR678      C0038       E0006
A0035       2023-07-20T09:29:29Z    STU789      C0053       E0014
A0036       2023-07-21T09:31:24Z    WXY901      C0029       E0008
A0037       2023-07-22T11:33:37Z    ZDE012      C0061       E0011
A0038       2023-07-23T13:35:50Z    FGL123      C0086       E0005
A0039       2023-07-24T15:38:03Z    JMO234      C0041       E0015
A0040       2023-07-26T19:42:29Z    NRS3454     C0057       E0001
A0041       2023-07-27T21:44:42Z    TWX456      C0075       E0013
A0042       2023-07-30T01:49:08Z    ZGH567      C0031       E0002
A0043       2023-07-31T03:51:21Z    BKL678      C0055       E0003
A0044       2023-08-01T05:53:34Z    BFG678      C0077       E0010
A0045       2023-08-02T07:55:47Z    DHI789      C0035       E0004
A0046       2023-08-03T09:58:00Z    FJK9015     C0067       E0009
A0047       2023-08-04T12:00:13Z    HLM012      C0092       E0012
A0048       2023-08-06T16:04:39Z    JNO1236     C0051       E0006
A0049       2023-08-07T18:06:52Z    LPR234      C0095       E0011
A0050       2023-08-09T22:11:18Z    NST345      C0043       E0005
A0051       2023-08-10T12:13:31Z    PVW456      C0040       E0015
A0052       2023-08-11T14:15:44Z    RCD123      C0007       E0001
A0053       2023-08-12T16:17:57Z    TEF2347     C0071       E0013
A0054       2023-08-13T18:20:10Z    VGH345      C0034       E0002
A0055       2023-08-14T20:22:23Z    XIJ4568     C0058       E0003
A0056       2023-08-15T22:24:36Z    ZLM567      C0020       E0010
A0057       2023-08-17T00:26:49Z    BNO678      C0018       E0004
A0058       2023-08-18T02:29:02Z    DQR7899     C0001       E0009
A0059       2023-08-19T04:31:15Z    FST901      C0073       E0012
A0060       2023-08-21T08:35:41Z    HUV012      C0039       E0006
A0061       2023-08-22T10:37:54Z    JWX123      C0062       E0014
A0062       2023-08-26T18:46:46Z    LZY234      C0085       E0008
A0063       2023-08-27T20:48:59Z    NAB345      C0048       E0011
A0064       2023-08-28T22:51:12Z    PDE4560     C0024       E0005
A0065       2023-08-30T00:53:25Z    RGH567      C0014       E0015
A0066       2023-08-31T02:55:38Z    TIJ6781     C0097       E0001
A0067       2023-09-01T04:57:51Z    VLM789      C0042       E0013
A0068       2023-09-03T09:02:17Z    XNO901      C0024       E0005
A0069       2023-09-04T11:04:30Z    ZQR0122     C0066       E0015
A0070       2023-09-06T15:08:56Z    BST123      C0097       E0001
A0071       2023-09-08T19:13:22Z    DUV234      C0042       E0013
A0072       2023-09-10T23:17:48Z    FXY345      C0033       E0002
A0073       2023-09-12T01:20:01Z    HBC4563     C0052       E0003
A0074       2023-09-13T03:22:14Z    JEF567      C0064       E0010
A0075       2023-09-14T05:24:27Z    LGH6784     C0087       E0004
A0076       2023-09-15T12:34:56Z    NIJ789      C0045       E0009
A0077       2023-09-16T23:45:12Z    PKL901      C0022       E0012
A0078       2023-09-18T10:11:22Z    RNO012      C0015       E0006
A0079       2023-09-19T14:15:16Z    TQR123      C0037       E0014
A0080       2023-09-20T18:19:10Z    VST234      C0046       E0008
A0081       2023-09-22T07:23:04Z    WUV345      C0026       E0011
A0082       2023-09-23T21:26:58Z    YBC456      C0008       E0005
A0083       2023-09-25T08:30:52Z    ZDE5675     C0070       E0015
A0084       2023-09-26T13:34:46Z    JKT567      C0054       E0001
A0085       2023-09-27T19:38:40Z    RQZ1230     C0060       E0013
A0086       2023-09-29T06:42:34Z    BNG789      C0020       E0002
A0087       2023-09-30T20:46:28Z    XYP4567     C0013       E0004
A0088       2023-10-02T07:50:22Z    MVC901      C0002       E0009
A0089       2023-10-03T12:54:16Z    LGH2345     C0057       E0012
A0090       2023-10-04T18:58:10Z    WFD678      C0044       E0006
A0091       2023-10-06T05:02:04Z    ZTR3456     C0022       E0014
A0092       2023-10-07T19:05:58Z    HJK1026     C0016       E0008
A0093       2023-10-09T06:09:52Z    NML5678     C0054       E0011
A0094       2023-10-10T11:13:46Z    CDE304      C0050       E0005
A0095       2023-10-11T17:17:40Z    VBN6789     C0026       E0015
A0096       2023-10-13T04:21:34Z    QWE405      C0009       E0001
A0097       2023-10-14T18:25:28Z    ASD1237     C0079       E0013
A0098       2023-10-16T12:00:21Z    YUI607      C0003       E0002
A0099       2023-10-20T13:14:15Z    OPR7890     C0058       E0003
A0100       2023-10-21T08:50:55Z    GFD908      C0020       E0010
A0101       2024-01-01T13:02:37Z    MNP789      C0099       E0016

EMPLOYEE SALARY AND BONUS
Enter Employee ID: E0008
Enter year in number: 2023
Enter month in number: 10
Employee Salary: RM 2435.00
Employee Bonus: RM 0.00
```

```
SEARCH AND FILTER
Search for employee records
1. By employee ID
2. By employee name
3. By employee status
4. Skip
Enter choice: 3
Enter employee status: 0

Employee records of status 0
EMPLOYEE ID         EMPLOYEE NAME                   EMPLOYEE STATUS     PASSWORD
E0002               Siti binti Ali                  0                   j4k5l6m7
E0003               Tan Wei Ling                    0                   n8o9p0q1
E0004               Lee Chee Seng                   0                   r2s3t4u5
E0005               Rajesh Kumar                    0                   v6w7x8y9
E0006               Ng Mei Yen                      0                   z0a1b2c3
E0007               Mohd Zaki bin Omar              0                   d4e5f6g7
E0008               Lim Hui Min                     0                   h8i9j0k1
E0009               Wong Kai Sheng                  0                   l2m3n4o5
E0011               Goh Wei Xian                    0                   t0u1v2w3
E0013               Ramesh Singh                    0                   b8c9d0e1
E0014               Tan Siew Hoon                   0                   f2g3h4i5
E0016               Ali bin Mohammad                0                   ali23456
Search for customer records
1. By customer ID
2. By customer name
3. By phone number
4. By postcode
5. Skip
Enter choice: 4
Enter postcode: 70000

Customer records of postcode 70000
CUSTOMER ID         CUSTOMER NAME                   PHONE NUMBER    POSTCODE
C0017               Lim Chee Seng                   0167890134      70000


Search for sales records
1. By sales ID
2. By month
3. By car plate
4. By customer ID
5. By employee ID
6. Skip
Enter choice: 2
Enter month: 8
Sales record of month 8
SALES ID    DATE AND TIME           CAR PLATE       CUSTOMER ID     EMPLOYEE ID
A0044       2023-08-01T05:53:34Z    BFG678          C0077           E0010
A0045       2023-08-02T07:55:47Z    DHI789          C0035           E0004
A0046       2023-08-03T09:58:00Z    FJK9015         C0067           E0009
A0047       2023-08-04T12:00:13Z    HLM012          C0092           E0012
A0048       2023-08-06T16:04:39Z    JNO1236         C0051           E0006
A0049       2023-08-07T18:06:52Z    LPR234          C0095           E0011
A0050       2023-08-09T22:11:18Z    NST345          C0043           E0005
A0051       2023-08-10T12:13:31Z    PVW456          C0040           E0015
A0052       2023-08-11T14:15:44Z    RCD123          C0007           E0001
A0053       2023-08-12T16:17:57Z    TEF2347         C0071           E0013
A0054       2023-08-13T18:20:10Z    VGH345          C0034           E0002
A0055       2023-08-14T20:22:23Z    XIJ4568         C0058           E0003
A0056       2023-08-15T22:24:36Z    ZLM567          C0020           E0010
A0057       2023-08-17T00:26:49Z    BNO678          C0018           E0004
A0058       2023-08-18T02:29:02Z    DQR7899         C0001           E0009
A0059       2023-08-19T04:31:15Z    FST901          C0073           E0012
A0060       2023-08-21T08:35:41Z    HUV012          C0039           E0006
A0061       2023-08-22T10:37:54Z    JWX123          C0062           E0014
A0062       2023-08-26T18:46:46Z    LZY234          C0085           E0008
A0063       2023-08-27T20:48:59Z    NAB345          C0048           E0011
A0064       2023-08-28T22:51:12Z    PDE4560         C0024           E0005
A0065       2023-08-30T00:53:25Z    RGH567          C0014           E0015
A0066       2023-08-31T02:55:38Z    TIJ6781         C0097           E0001
```

```
Search for vehicle records
1. By car plate
2. By car model
3. By acquired price
4. By car status
5. By sales price
6. Skip
Enter choice: 6

SALES INSIGHTS
1. Annual sales
2. Monthly sales
3. Number of cases
Enter choice: 3
Number of Cases
```




```
BUILD SUCCESSFUL (total time: 7 minutes 41 seconds)
run:
USER REGISTRATION AND LOGIN
E0001, welcome back
1. Log out
2. Exit
3. Skip
Enter choice: 1
Logged out.
Want to login again?
1. Yes
2. No
Enter choice: 2
BUILD SUCCESSFUL (total time: 4 seconds)
```

6. LIMITATIONS OF LECARS SALES MANAGEMENT SYSTEM

Limitation	Explanation
Login Page Limitation	Whether the user is a management-level employee or sales-level employee, he cannot reset his password once he forgets it.
Difficulty in Viewing Menu	User is unable to go back to previous menu for choosing other options.
Termination of Program if Error is Present	User has to restart the entire process from the beginning, even if there is a single error in input.
Restriction on Only Performing a Specific Activity	Mandatory to go through the entire process even if user only wants to perform a specific activity from the entire code.
Data Unsynchronisation Across Devices	Inconsistencies in data alterations occur due to data is stored offline and doesn't update uniformly across different platforms.