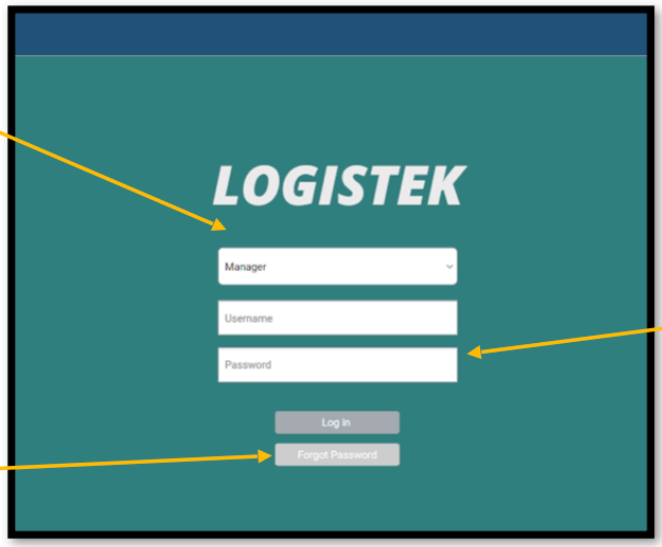
CRITERION B: Design Documentation
B1.2: TEST PLAN

Test Type	Nature of Test	Example
The program will be connected to a database with log in credentials. The user's input will be validated against the stored values to allow the login.	Inputting the correct login credentials and usertype and attempting to login. Repeating the same with incorrect login credentials and usertype to check if program will provide an inccorect log-in message prompt.	Correct Login User type: Manager User name: Manager01 Password:123456 Incorrect Login User type: Manager User name: Driver01 Password:432913
The program will send the user to different panels depending on user type selected	Logging into the Manager user type and checking if goes to the correct user panel. Repeating the same with the Driver user type.	Displays Manager panel for Manager user type and Driver panel for Driver user type.
The program has a clear form function to clear input boxes	Click clear form button and it should remove all the data in the input boxes allowing new inputs.	If the clear form button is clicked, all input boxes are cleared
The program has a sign out function and a back page function	Clicking signout button to see if it redircts user back to log in. Clicking back button to see if it redirctes user to previous page.	If the signout button is clicked, user is redirected to log in page If the back button is clicked, user is directed to previous page.
The program should allow only integer inputs for the truck and trailer ID inputs for the driver user.	Inputting integer values for the trailer and truck id inputs to see if it is accepted. Repeating the same with string values and checking it's rejected.	Input: 101 Output: Information saved to database Input: One hundred and one Output: Incorrect data type

The program should allow only string inputs for the driver name input for the driver user.	Inputting string values for the driver name to see if it is accepted. Repeating the same with integer values and checking it's rejected.	Input: John Doe Output: Information saved to database Input: J0HN DO3 Output: Incorrect data type
The program should store the drivers input in the MySQL database	Click save button and be prompted with a saved to database message.	Saving the driver's input and crosschecking with the MySql database to see if it's been saved
The program allows the manager user to view diiferent tables with different set criterias from the menu.	User selects the criteria set in the menu and a table should display with the fetched data from the MySQL database.	Table displayed when set criteria button clicked in Manager panel menu.
The program has a refesh function for the table.	User clicks refresh button and the table is refreshed and populated with new inputs	Table refresh button is clicked and table is populated with new inputs
The program has a Pie Chart function which displays the total avaliable trucks	User clicks Pie Chart button which displays a pie chart with the percentages and number of truck available and unavailable.	Pie chart button clicked. Pie chart displayed to user. Displays statstics when hovered over For example : 50%, 12 truck available.

B1.3: Graphical User Interface

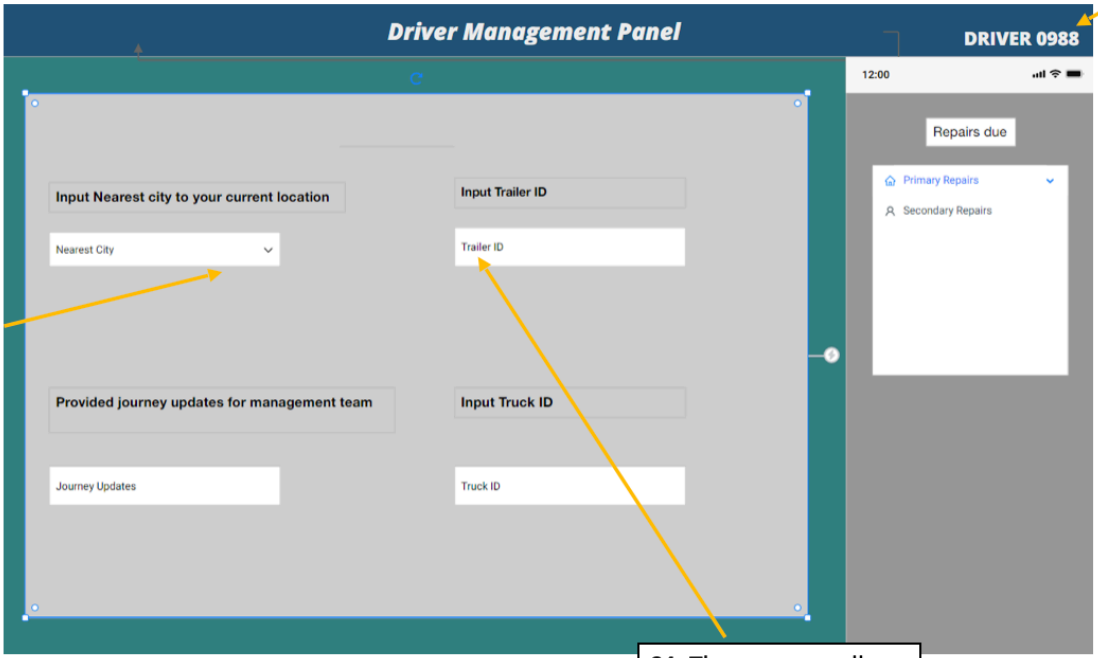
S1. The program provides a hierarchy of login credentials: Manager, Driver and Mechanic.



Dropdown menu to select login level and input box for user name and password

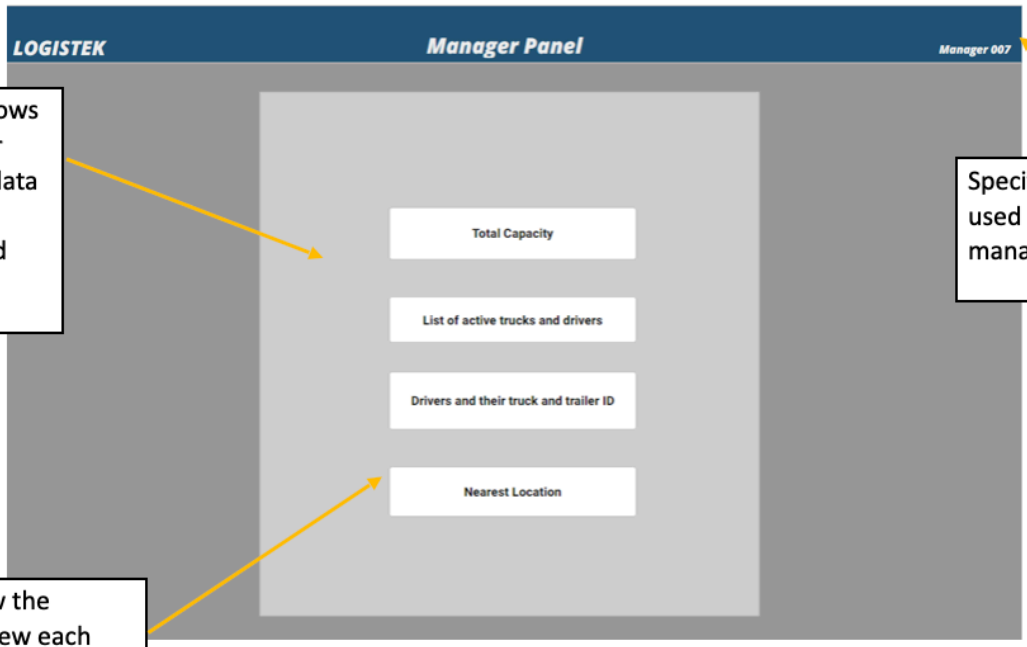
Forgot my password option that will provide a message to contact the system administrator

Specific log in credentials used to identify the driver



S3. The program allows the driver to input the location of the truck

S4. The program allows the driver to input the truck and trailer ID.

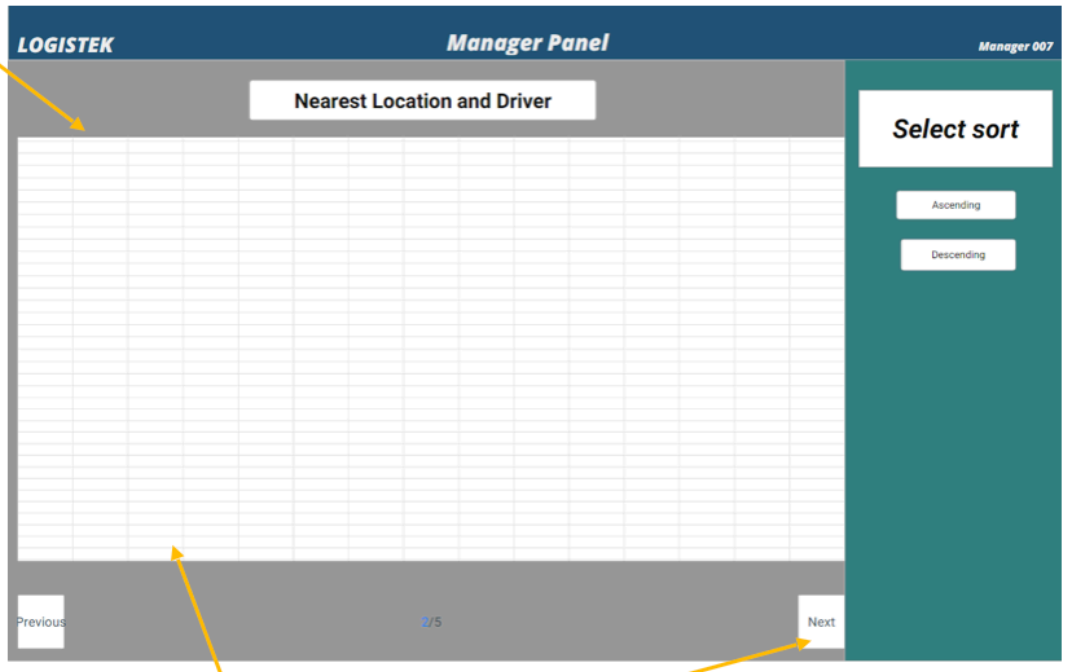


S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics

Specific log in credentials used to identify the manager

Menu to allow the manager to view each data set independently to improve usability

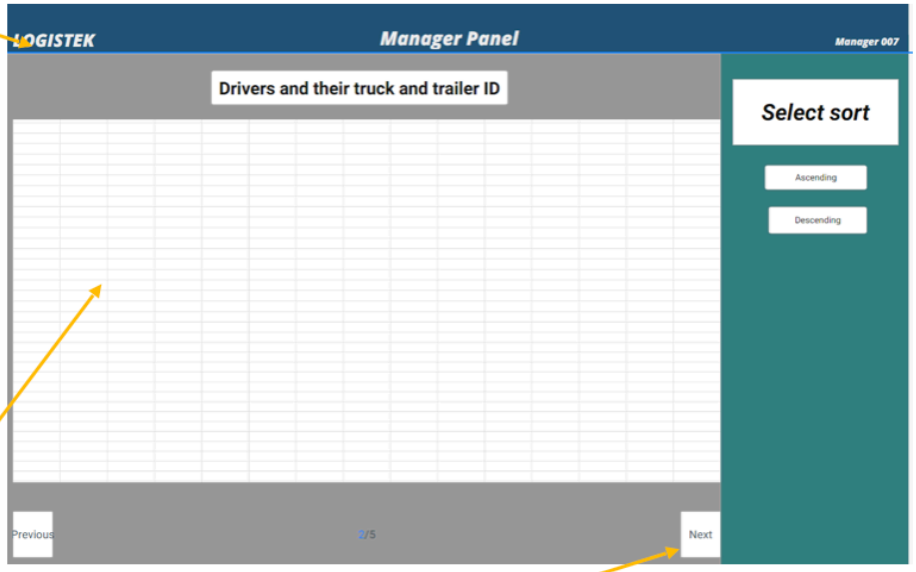
S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics



S8. The program should interpret the data and display the nearest location and drivers

Previous and Next feature to view multiple pages of data.

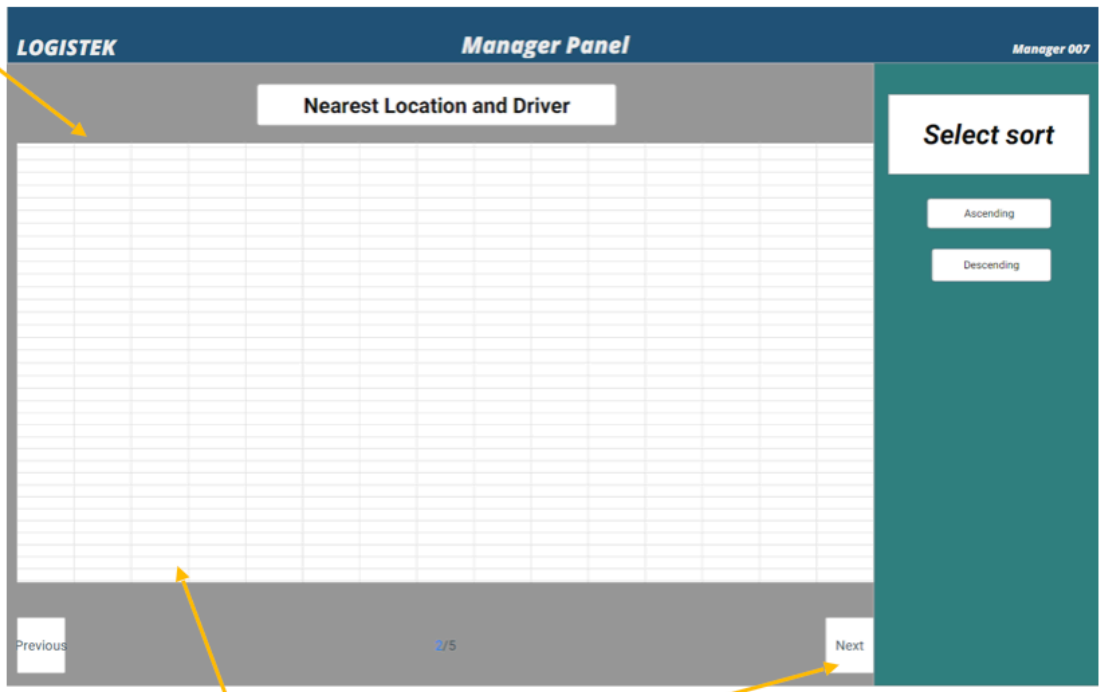
S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics



S8. The program should interpret the data and display the driver and trailer ID

Previous and Next feature to view multiple pages of data.

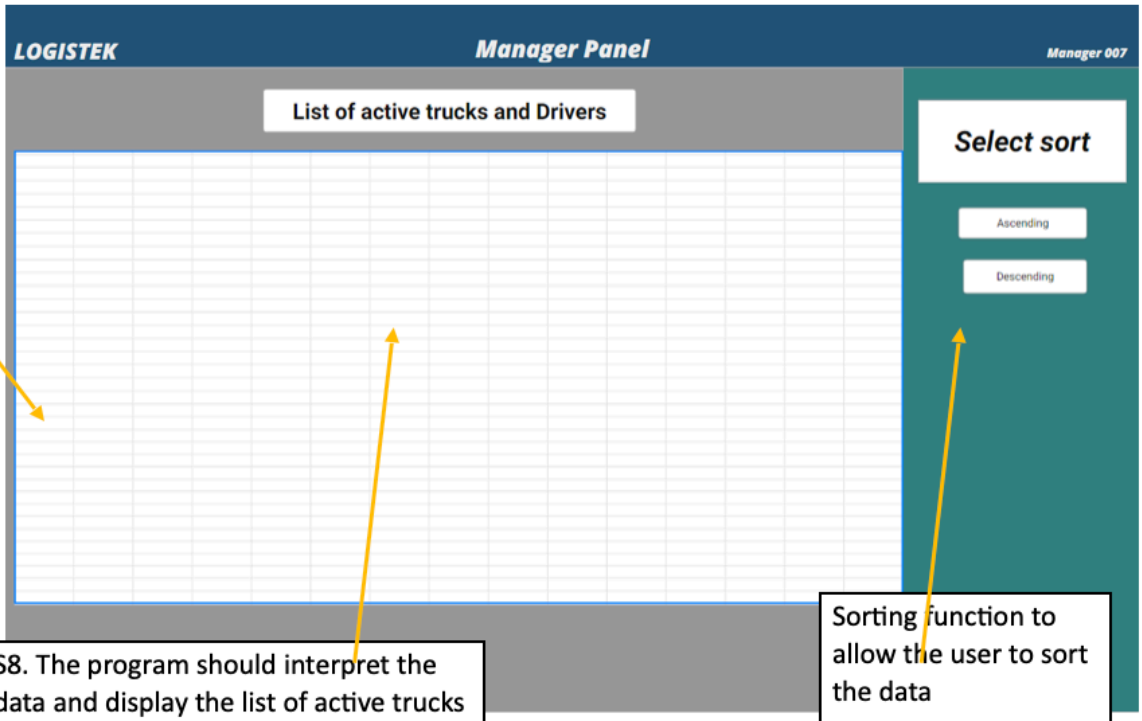
S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics



S8. The program should interpret the data and display the nearest location and drivers

Previous and Next feature to view multiple pages of data.

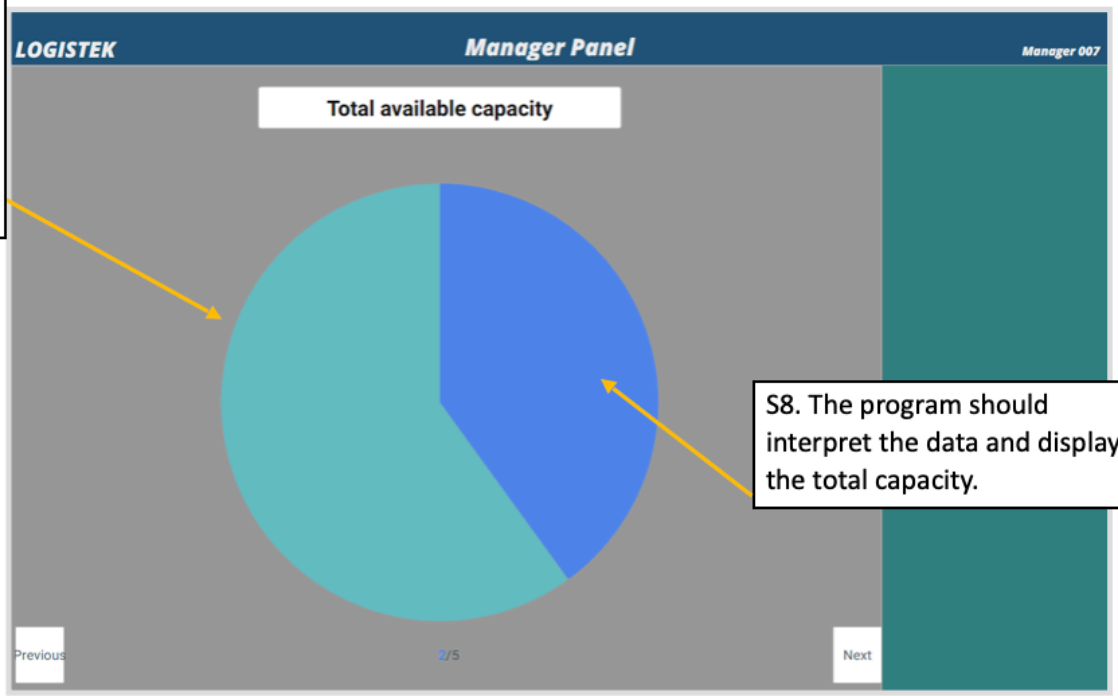
S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics



S8. The program should interpret the data and display the list of active trucks and drivers

Sorting function to allow the user to sort the data

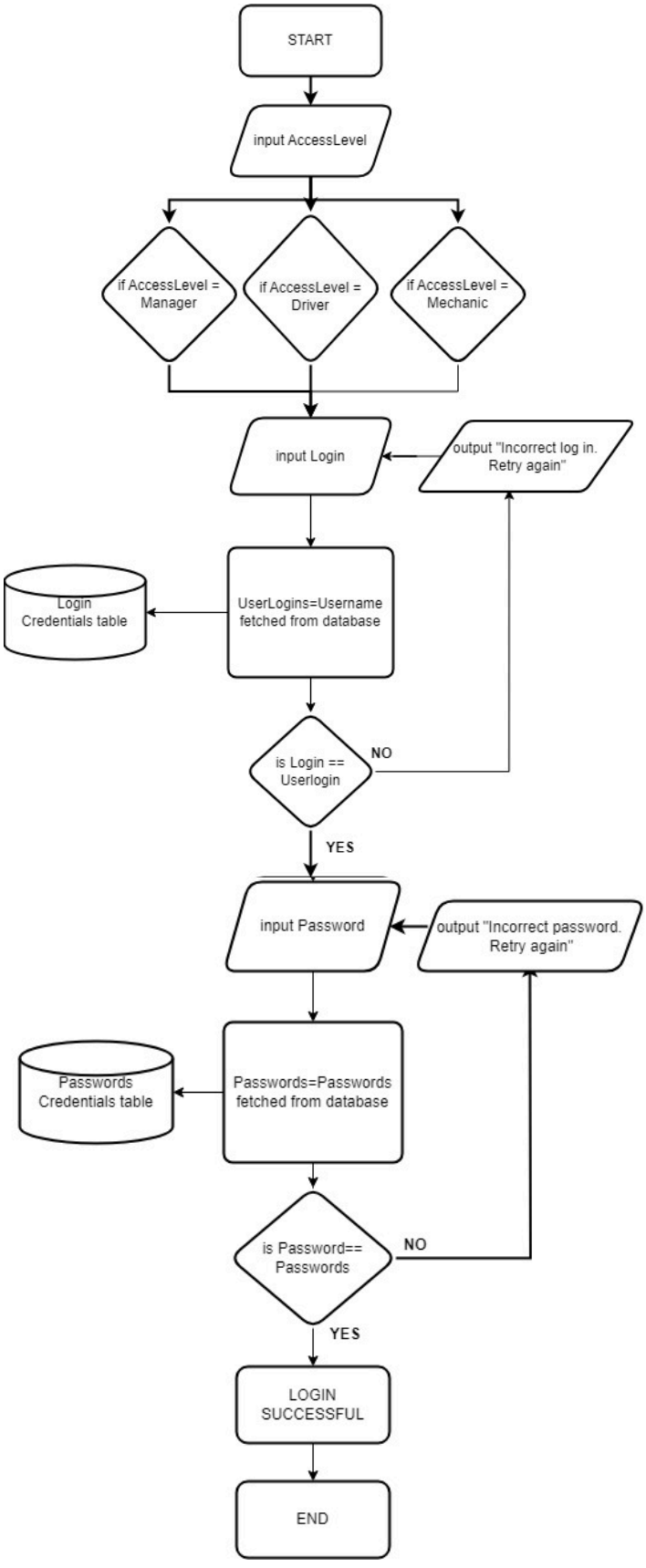
S7. The program allows those with manager access to view the data based on the inputs from the drivers and mechanics



S8. The program should interpret the data and display the total capacity.

B1.4 Flowcharts

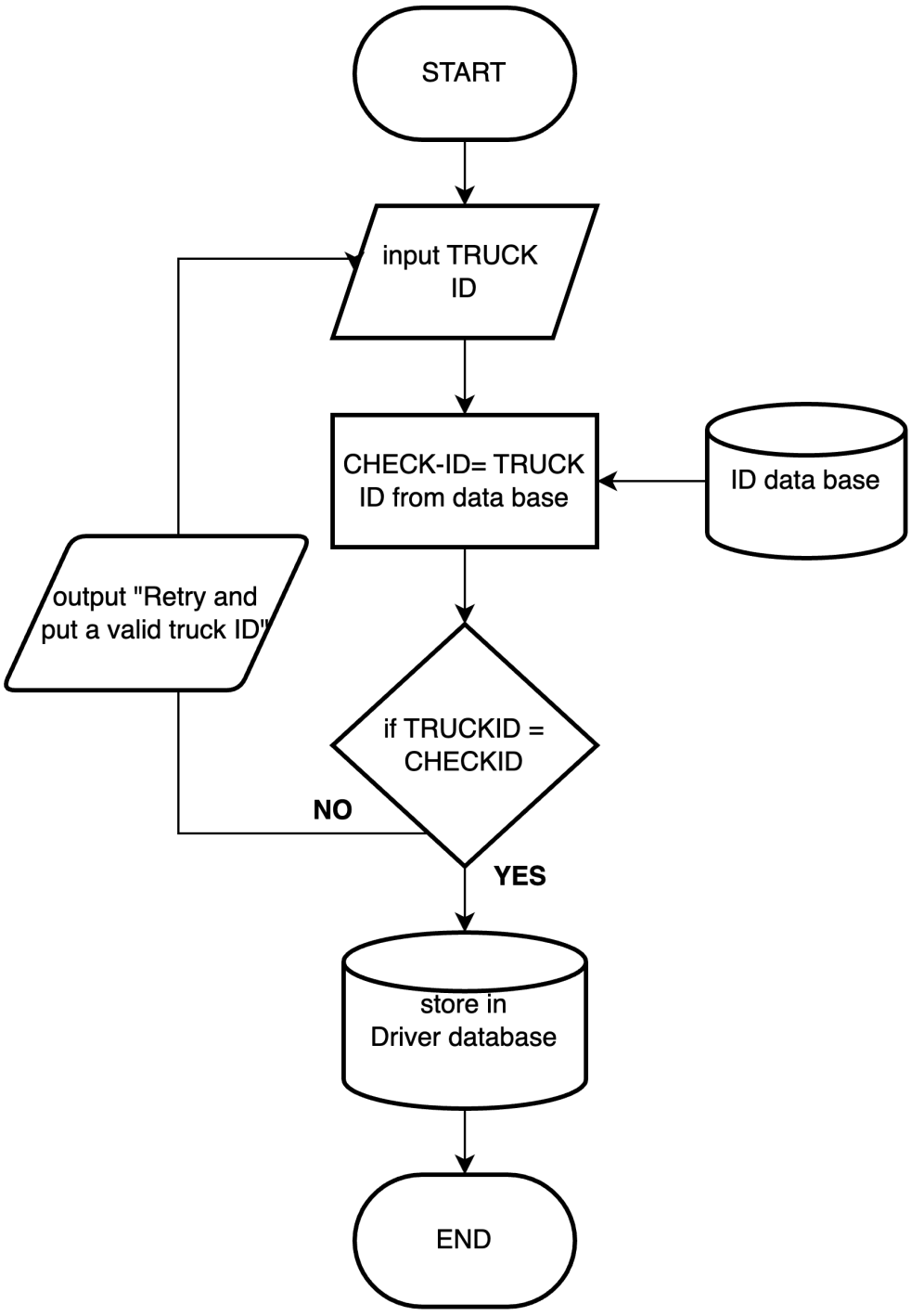
Log-In flowchart



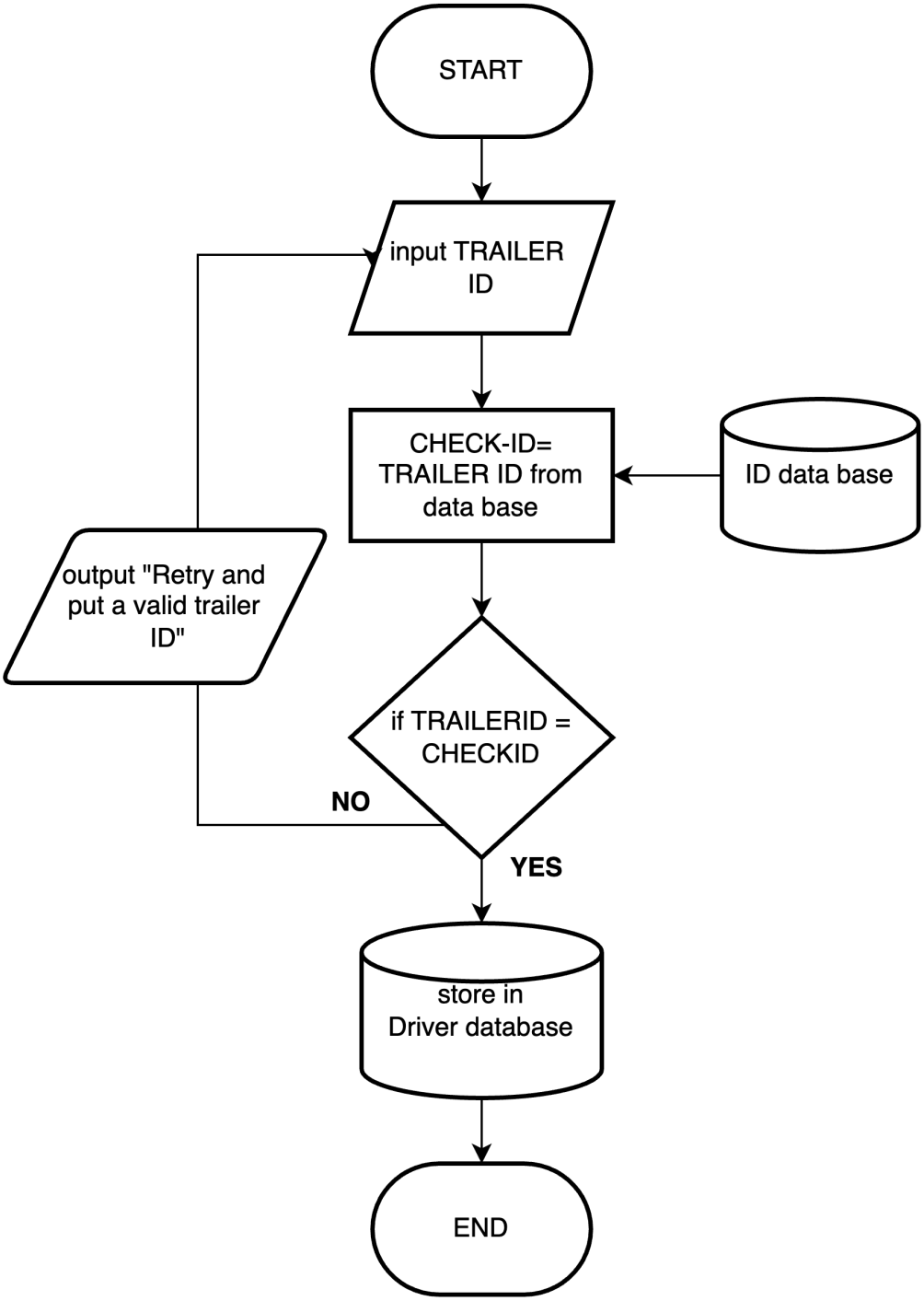
Log-In Pseudocode

```
  OUTPUT "Select user type (1 for Driver, 2 for Manager): "
  INPUT userType

  OUTPUT "Enter username: "
  INPUT username
  OUTPUT "Enter password: "
  INPUT password

 IF userType == 1 THEN
    IF username == storedDriverUsername AND password == storedDriverPassword THEN
      OUTPUT "Login successful as Driver."
    ELSE
      OUTPUT "Incorrect username or password. Please try again.”
       INPUT username
       INPUT password
    ENDIF
  ELSE IF userType == 2 THEN
    IF username == storedManagerUsername AND password == storedManagerPassword THEN
      OUTPUT "Login successful as Manager."
    ELSE
      OUTPUT "Incorrect username or password. Please try again.”
       INPUT username
       INPUT password
    ENDIF
ENDIF
```

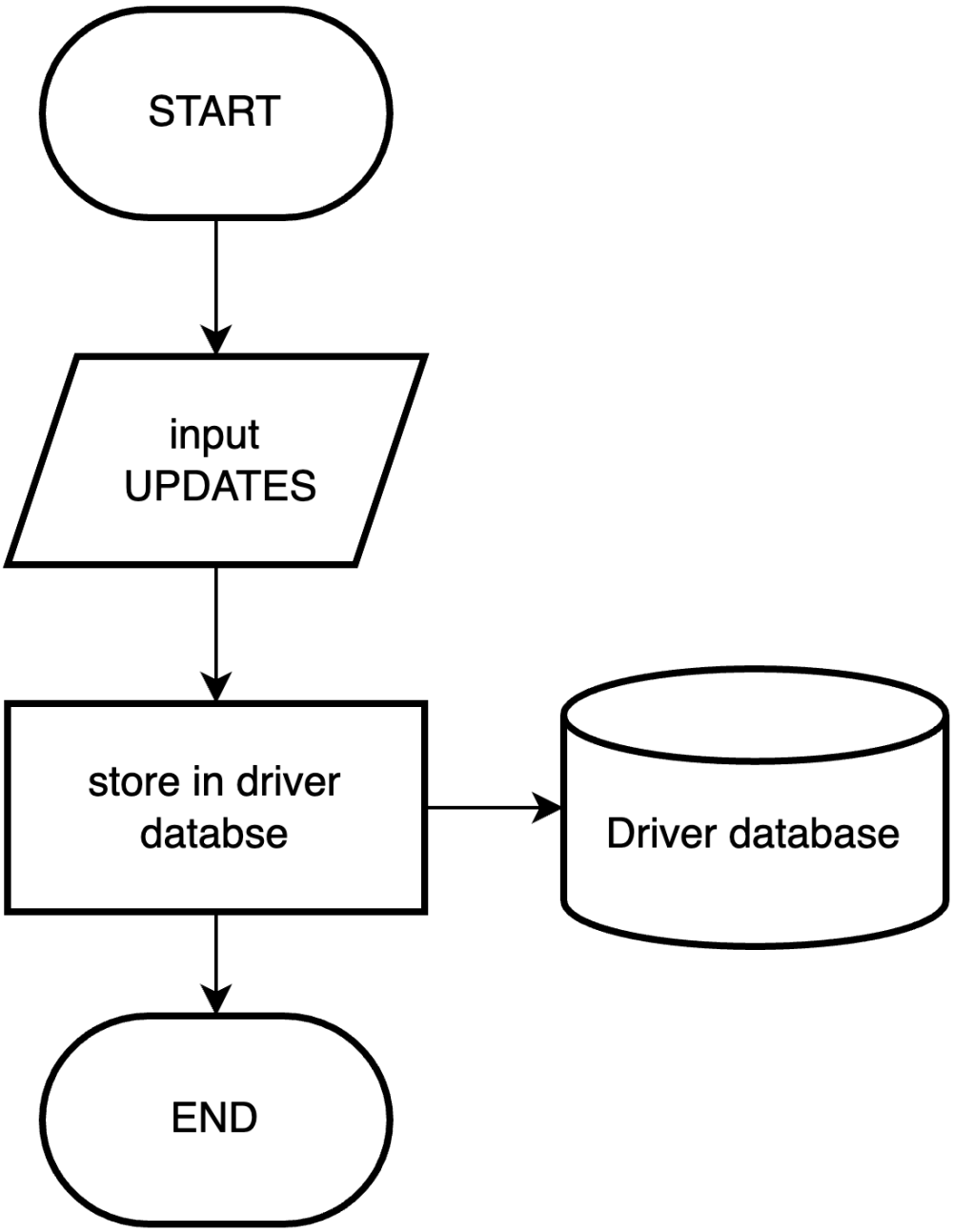
Driver Truck ID input flowchart



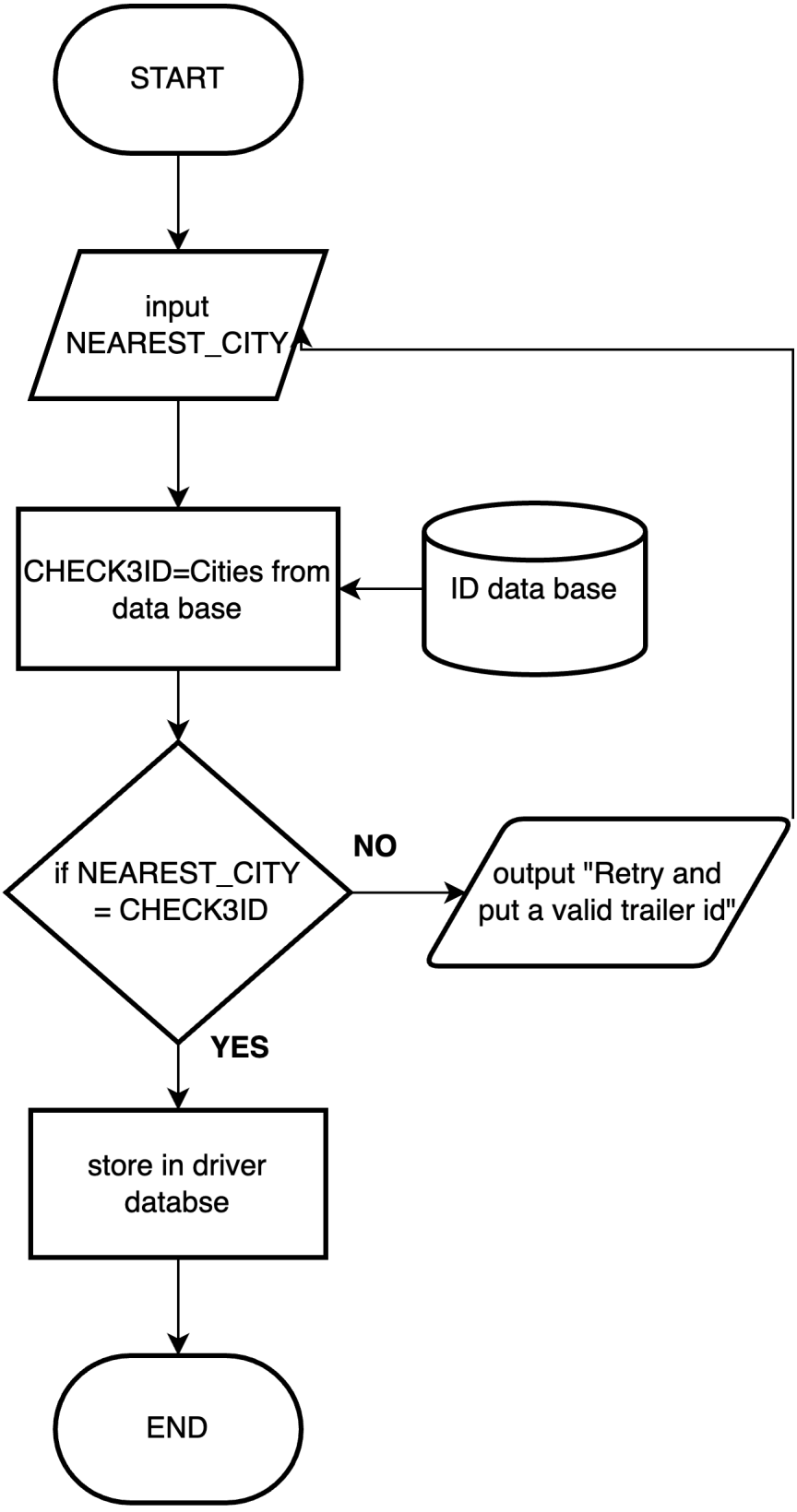
Driver Trailer ID input flowchart



Driver Updates input flowchart



Driver Nearest city input flowchart

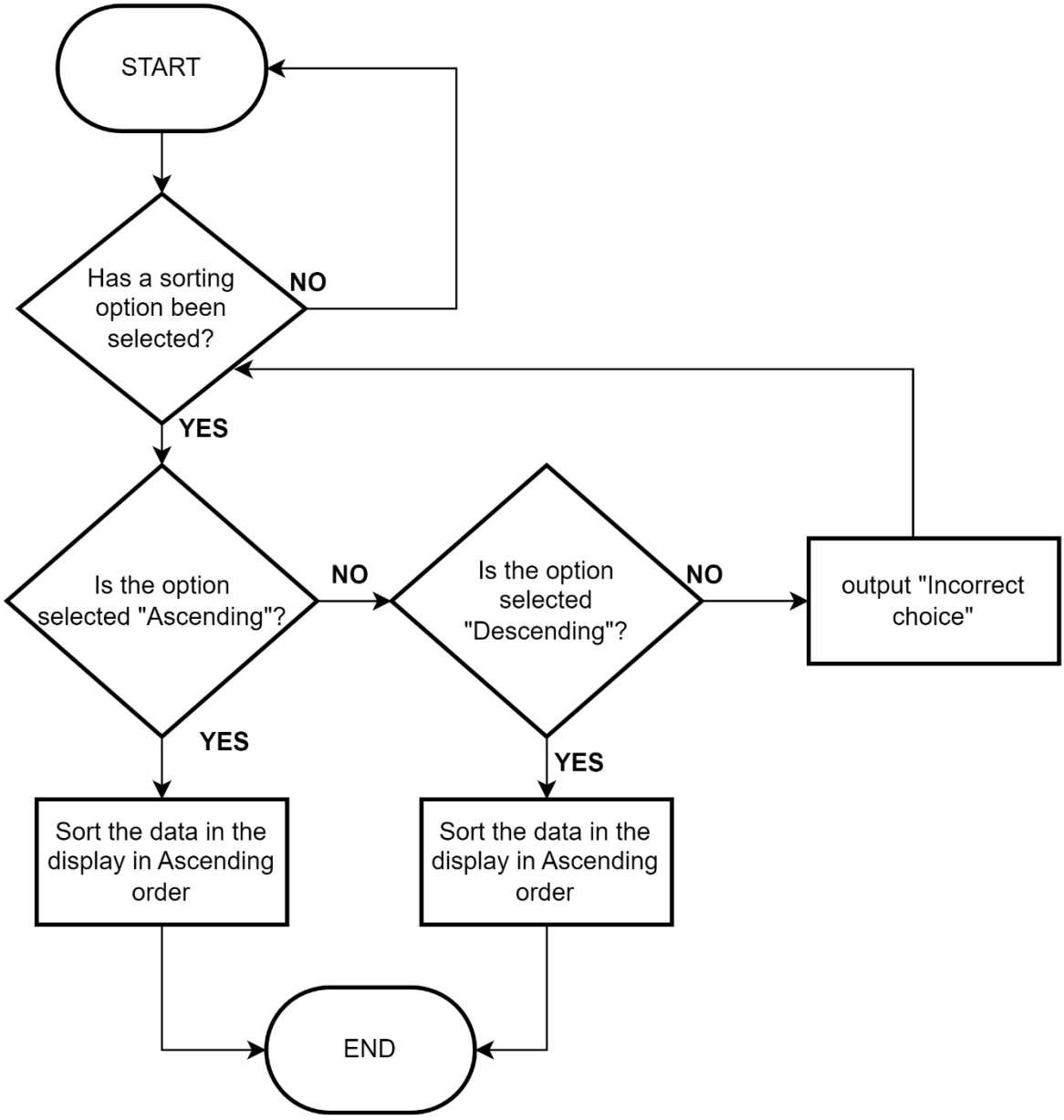


Driver input pseudocode

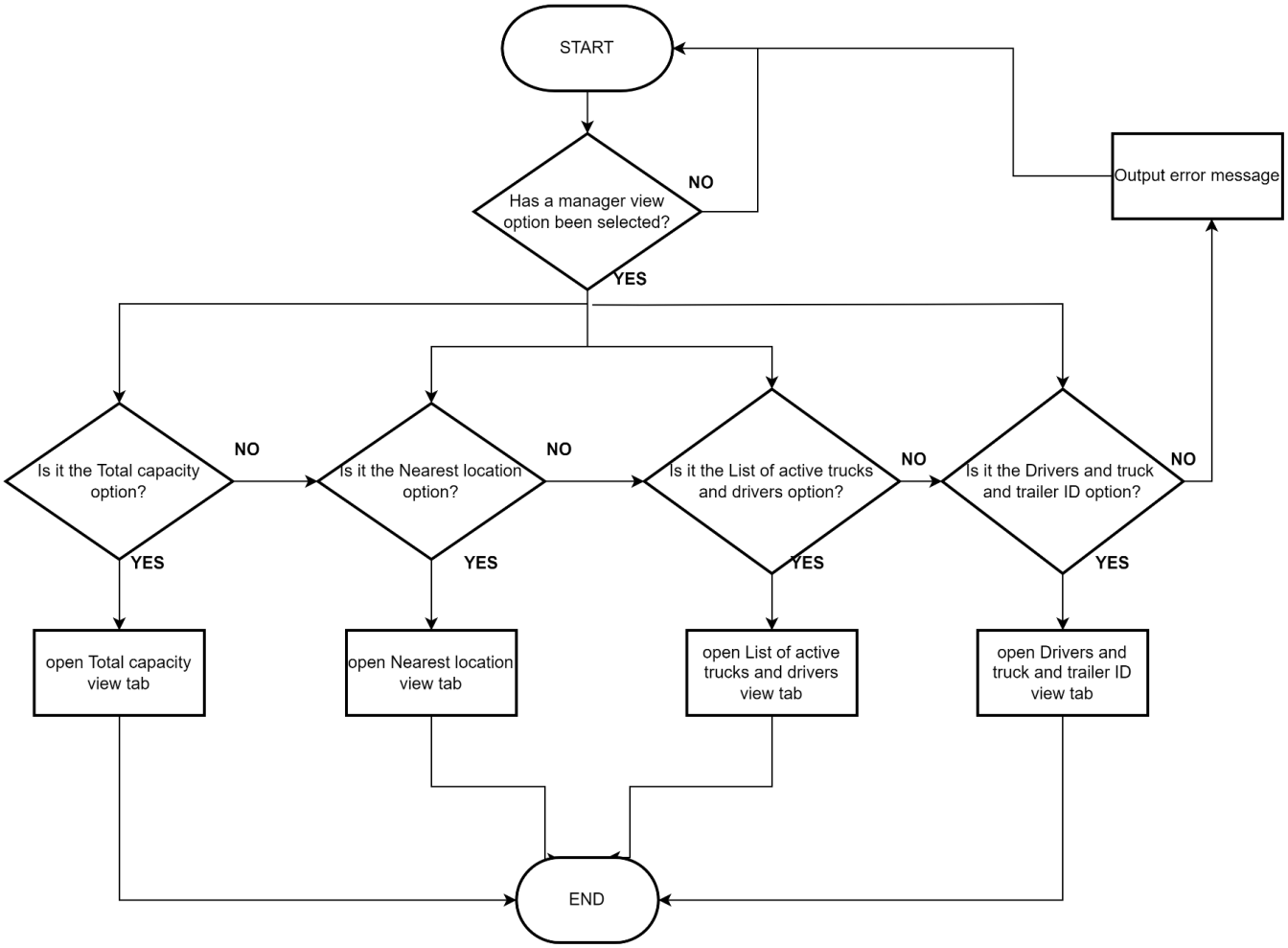
```
Print "Enter Driver Name: "
driverName = input()
IF driverName is empty
    Print "Driver Name cannot be empty. Please enter a valid name."
    INPUT driverName
END IF

Print "Enter Nearest City: "
nearestCity = input()
IF nearestCity is empty
    Print "Nearest City cannot be empty. Please enter a valid city."
    INPUT nearestCity
END IF

Print "Enter Journey Updates: "
journeyUpdates = input()
IF journeyUpdates is empty
    Print "Journey Updates cannot be empty. Please enter valid updates."
    INPUT journeyUpdates
END IF

Print "Enter Activity Status: "
activityStatus = input()
IF activityStatus is empty
    Print "Activity Status cannot be empty. Please enter a valid status."
    INPUT activityStatus
END IF

Print "Enter Truck ID: "
truckIdInput = input()
IF truckIdInput is empty
    Print "Truck ID cannot be empty. Please enter a valid ID."
    INPUT truckId
```

```
END IF

Print "Enter Trailer ID: "
trailerIdInput = input()
IF trailerIdInput is empty
    Print "Trailer ID cannot be empty. Please enter a valid ID."
    INPUT trailerId
END IF
```

Sorting algorithm



Manager view selection panel flowchart



Manager view selection panel Pseudocode

Display “1. Nearest Location and Truck ID”

Display “2. Total Capacity”

Display “3. Truck ID and Driver Name”

Display “4. Truck ID and Activity Status”

Display “Click one of the four options:”

If choice == “Nearest Location and Truck ID” Then

    Redirect("NearestLocationAndTruckIDPage")

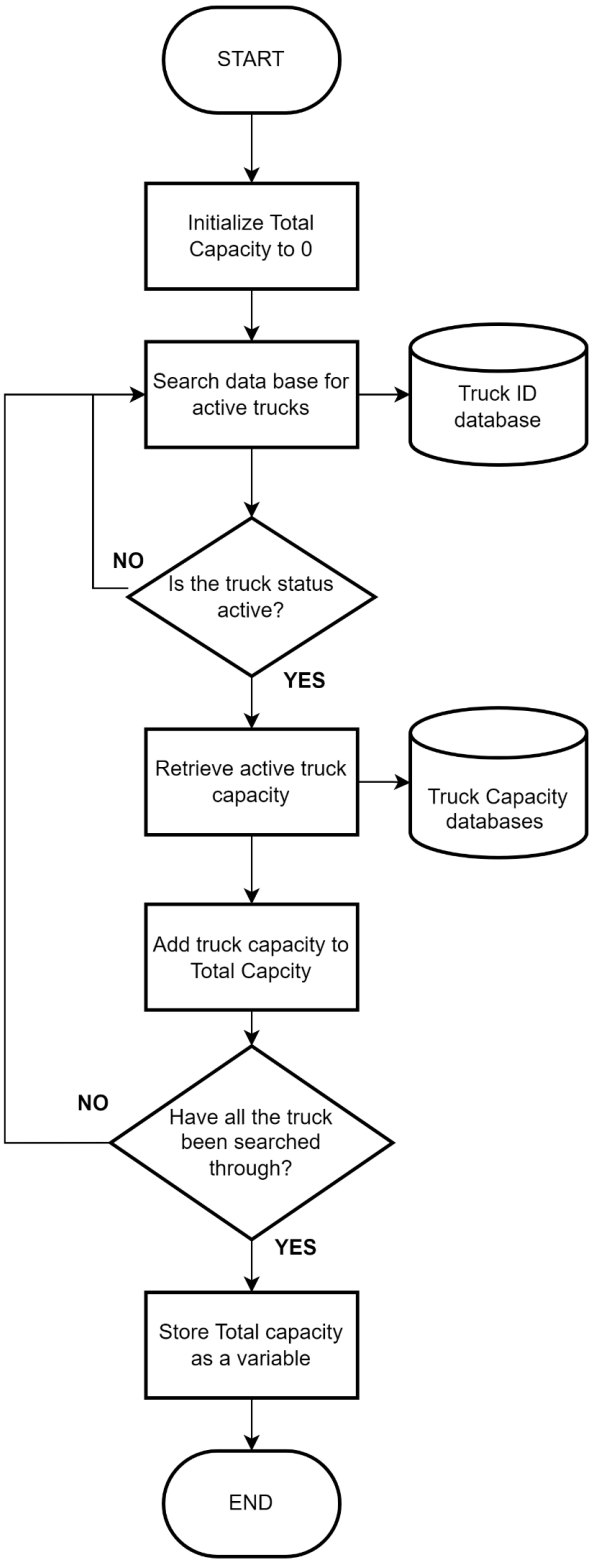
ElseIf choice == “Total Capacity" Then

    Redirect("TotalCapacityPage")

ElseIf choice == "Truck ID and Driver Name" Then

```
    Redirect("TruckIDAndDriverNamePage")
ElseIf choice == ‘Truck ID and Activity Status" Then
    Redirect("TruckIDAndActivityStatusPage")
Endif
```

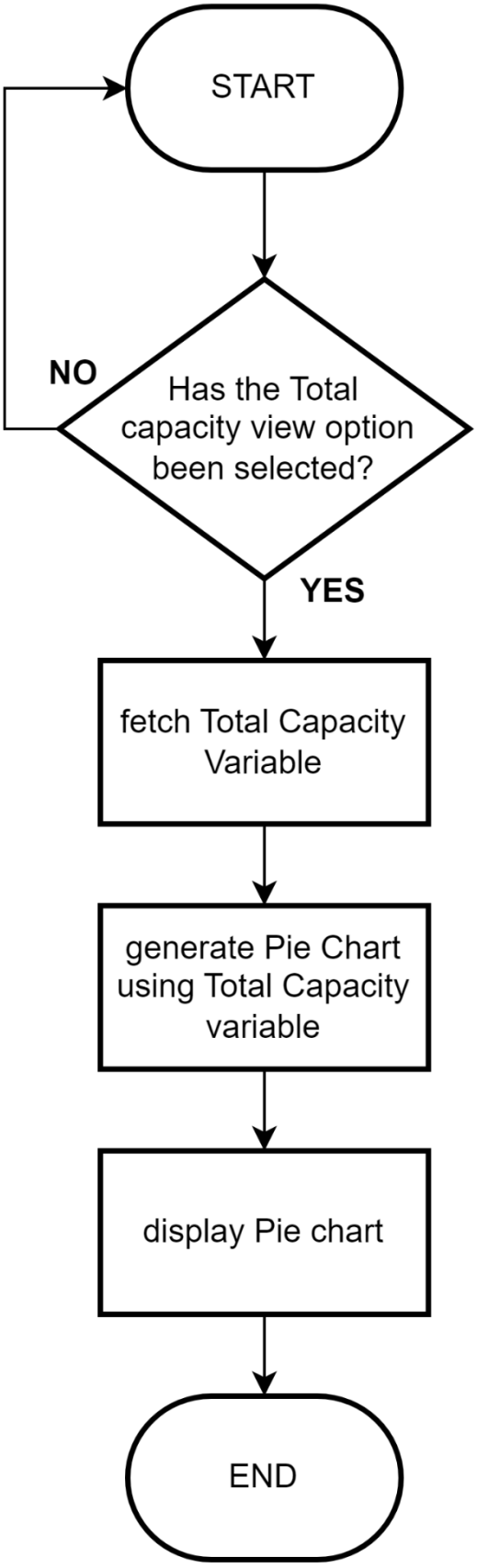
Calculating total capacity



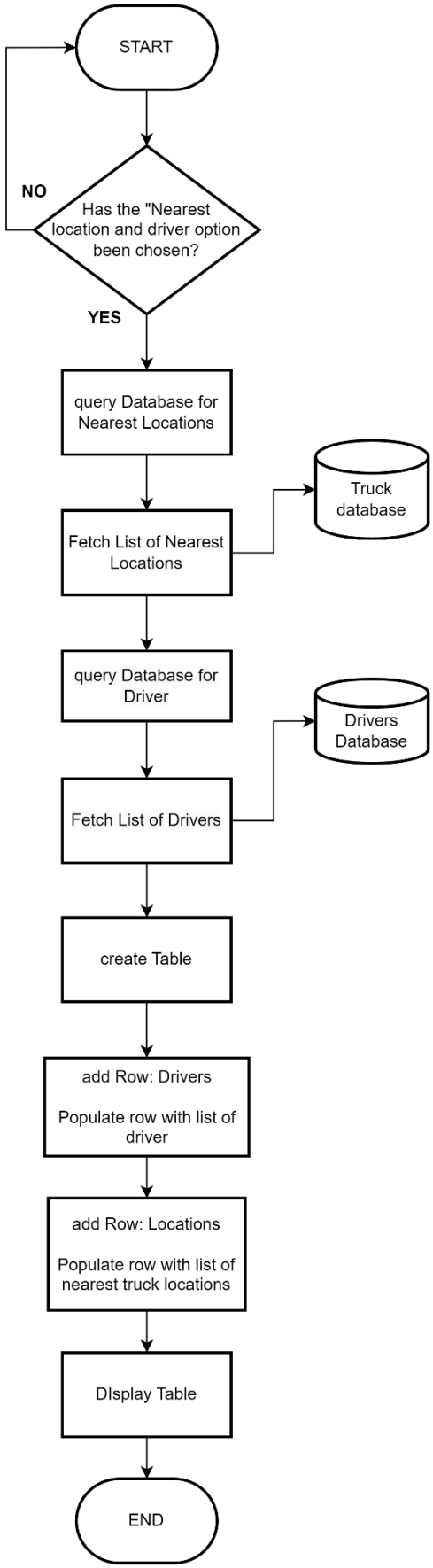
Calculating Total capacity Pseudocode

```
Totaltrucks = 0
Activetrucks = 0
Inactivetrucks = 0
CapacityPerTruck = 6500

 for 1 to length(Activitystatus) loop
        currentstatus = Activitystatus[index]
         if currentTruck.ActivityStatus = "Active" then
                 Activetrucks = Activetrucks + 1
        else
                Inactivetrucks = Inactivetrucks + 1
        end if
                Totaltrucks = Totaltrucks + 1
end loop

   Percentageactive= (Activetrucks / Totaltrucks) * 100
   Percentagenotactive = (Inactivetrucks / Totaltrucks) * 100
   totalCapacity = Totaltrucks * CapacityPerTruck
```

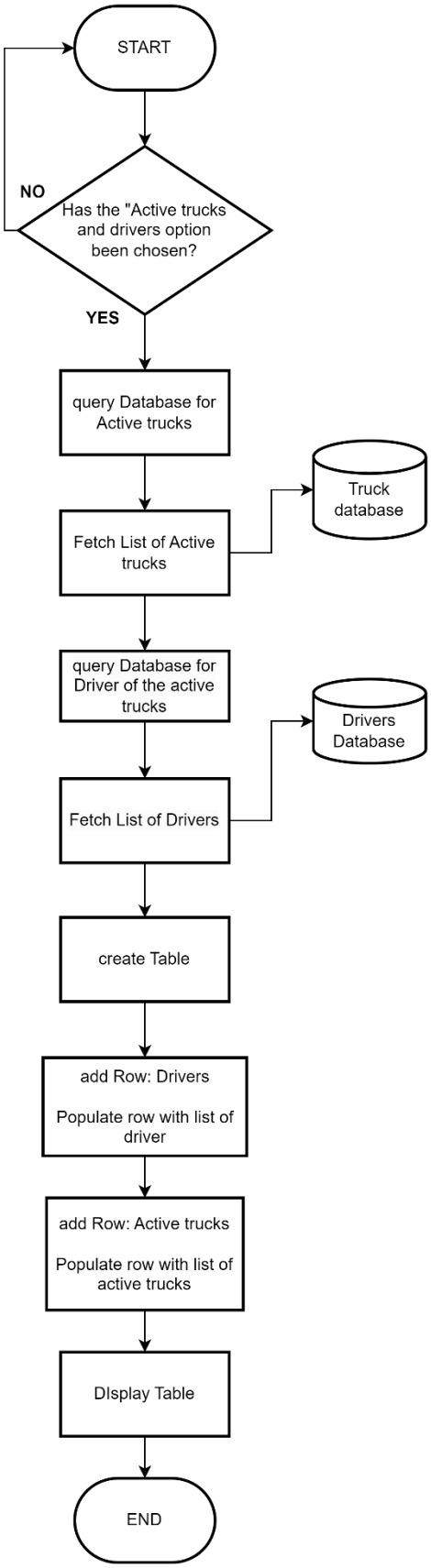
Displaying Total capacity



Displaying Nearest Location Flowchart



Displaying List of active drivers and trucks flowchart



Drivers with truck and trailer ID flowchart