LONDON METROPOLITAN UNIVERSITY

ITAHARI INTERNATIONAL COLLEGE

Module Code & Module Title

Programming CS4001NT

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2022 Spring 2

Student Name: Dipesh Bharati

Group: L1C5

London Met ID: 22016049

College ID: NP05CP4S220022

Assignment Due Date: 5 Aug 2022

Assignment Submission Date: 5 Aug 2022

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

Contents

List of Figures

List of Tables

1. Introduction

Oracle supports Java, a popular programming language. It is a general-purpose programming language and platform. Java is widely used around the world. More than 3 billion devices use it. Java is a widely used programming language for the following reasons:

• It is free and open source.

• Java has a large worldwide-based community.

• Because java is closely related to C++ and C language, programmers can easily move between java and other programming languages.

Some of the important uses of java are:

- Java is used to develop web application.
- It is used to develop mobile applications especially android apps.
- It is used to develop games. (w3schools, 1999-2022)

Figure 1: Founder of Java (Deep, June-20,2012)

2. Class Diagram

TransportGUI
- homeFrame,frame,frameElectric= **JFrame** - fieldID ,fieldname ,fieldWeight ,fieldColor ,fieldSpeed ,fieldEngineDisplacement ,fieldTorque ,fieldTankCapacity ,fieldGroundClearance: **JTextField** - fieldiD ,fieldAmount ,fieldSeats: **JTextField** -fieldElectricId ,fieldElectricName ,fieldElectricWeight ,fieldElectricColor ,fieldElectricSpeed ,fieldElectricBattery: **JTextField** - fieldPurElectricId ,fieldBrand ,fieldPrice ,fieldChargingTime ,fieldMileage ,fieldRange: **JTextField** -fieldSellElectricID ,fieldSellPrice: **JTextField** -BTNBK,BTNDS,BTNCLR,BTN**ADD**,back: **JButton** -electricScooterBtn,autoRickshawBtn,BTNelectricDisplay: **JButton** -BTNADDScooter,BTNPurchase,BTNSell: **JButton** -d,m,y: **JComboBox<String>**
+ TransportGUI() +getVehicleName():String +getVehicleWeight():String +getVehicleId():int +getVehicleColor():String +getVehicleSpeed():String +getVehicleDisplacement():int +getFuelTankCapacity():int +getTorque():String +getClearance():String +getBookVehicleId():int +getBooked():String +getChargeAmount():int +getNumberofSeats():int +getaddElectricVehicleId():int +getElectricVehicleName():String +getElectricVehicleWeight():String +getElectricVehicleColor():String +getElectricVehicleSpeed():String +getElectricBatteryCapacity():int +getPurElectricVehicleId():int +getPrice():int +getChargingTime():String +getMileage():String +getRange():int +getSellElectricVehicleId():int +getSellPrice():int +checkIfUnique(int vehicleId):Boolean +addAutoRickshaw():void +addElectricScooter():void +bookAutoRickshaw():void

+purElectricScooter():void +sellElectricScooter():void +displayAutoRickshaw():void +displayElectricScooter():void +clearAutoRickshaw():void +clearElectricScooter:void

Table 1 : Class Diagram

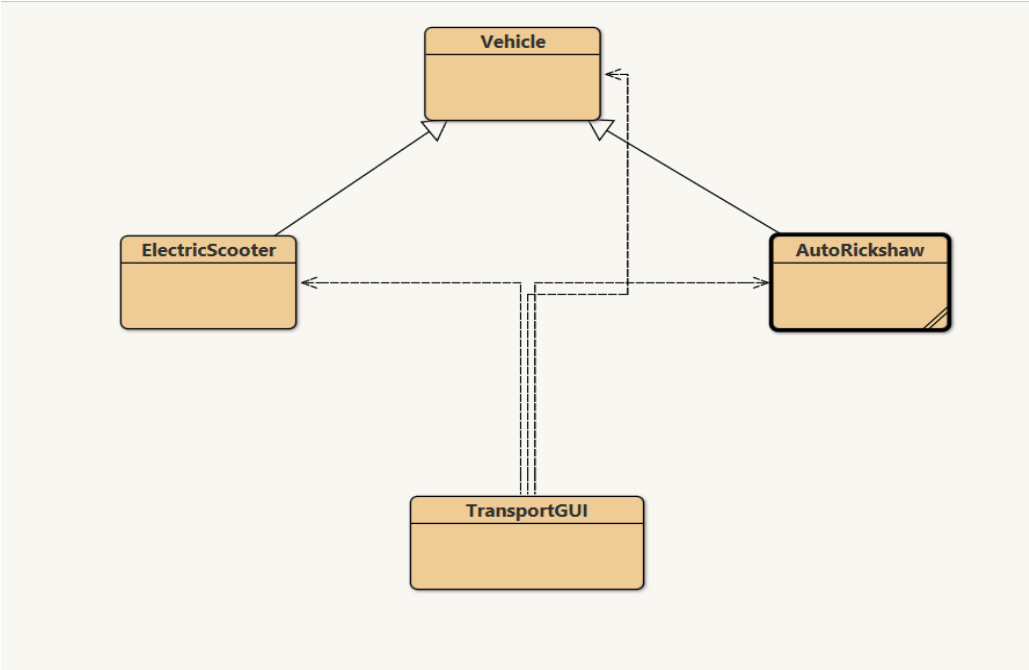

Figure 2: Class Diagram

3. Pseudocode

CREATE method TransportGUI():

CREATE JFrame "homeFrame"

TRY

SETContentPane of homeFrame with this image "darkwood.jpg"

CATCH

SHOW System.out.println("image doesn't exist");

ENDTRY

CREATE JFrame "frame"

TRY

SETContentPane of homeFrame with this image "light.jpg"

CATCH

SHOW System.out.println("image doesn't exist");

ENDTRY

CREATE JFrame "frameElectric"

TRY

SETContentPane of homeFrame with this image "dlight.jpg"

CATCH

SHOW System.out.println("image doesn't exist");

ENDTRY

CREATE JButton autoRickshawBtn

SET Bounds of autoRickshawBtn

ADD autoRickshawBtn to homeFrame

ADD actionListener to autoRickshawBtn

FUNCTION actionPerformed

SET frame to visible

SET homeFrame to visible(FALSE)

END

CREATE JButton electricScooterBtn

SET Bounds of electricScooterBtn

ADD electricScooterBtn to homeFrame

ADD actionListener to electricScooterBtn

FUNCTION actionPerformed

SET frameElectric to visible

SET homeFrame to visible(FALSE)

END

CREATE JLabel "title" text "Welcome”

SET Bounds of title

SET JLabel " title" font ("serif", Font.PLAIN, 50)

ADD JLabel " title" to homeFram

SET Foreground(Color.white)

CREATE JLabel "label" text "AutoRickshaw"

SET Bounds of label

SET JLabel " label" font ("serif", Font.PLAIN, 50)

ADD JLabel " label" to homeFrame

SET Foreground(Color.white)

CREATE JLabel "label2" text "Electric Scooter"

SET Bounds of label2

SET JLabel " label2" font ("serif", Font.PLAIN, 50)

ADD JLabel " label2" to homeFrame

SET Size of frame

SET Layout of frame

SET Visible of frame

SET DefaultCloseOperation of frame

SET Size of frameElectric

SET Layout of frameElectric

SET Visible of frameElectric

SET DefaultCloseOperation of frameElectric

SET Size of homeFrame

SET Layout of homeFrame

SET Visible of homeFrame

SET DefaultCloseOperation of homeframe

CREATE JLabel vehicleID text "Vehicle ID"

SET Bounds of vehicleID

ADD JLabel vehicleID to frame

CREATE JTextField fieldID

SET JTextField Bounds of fieldID

ADD JTextField fieldID to frame

CREATE JLabel vehicleName text "Vehicle Name"

SET Bounds of vehicleName

ADD JLabel vehicleName to frame

CREATE JTextField fieldName

SET JTextField Bounds of fieldName

ADD JTextField fieldName to frame

CREATE JLabel vehicleWeight text "Vehicle Weight"

SET Bounds of vehicleWeight

ADD JLabel vehicleWeight to frame

CREATE JTextField fieldWeight

SET JTextField Bounds of fieldWeight

ADD JTextField fieldWeight to frame

CREATE JLabel vehicleColor text "Vehicle Color"

SET Bounds of vehicleColor

ADD JLabel vehicleColor to frame

CREATE JTextField fieldColor

SET JTextField Bounds of fieldColor

ADD JTextField fieldColor to frame

CREATE JLabel vehicleSpeed text "Vehicle Speed"

SET Bounds of vehicleSpeed

ADD JLabel vehicleSpeed to frame

CREATE JTextField fieldSpeed

SET JTextField Bounds of fieldSpeed

ADD JTextField fieldSpeed to frame

CREATE JLabel engineDisplacement text "Engine Displacement"

SET Bounds of engineDisplacement

ADD JLabel engineDisplacement to frame

CREATE JTextField fieldEngineDisplacement

SET JTextField Bounds of fieldEngineDisplacement

ADD JTextField fieldEngineDisplacement to frame

CREATE JLabel torque text "Torque"

SET Bounds of torque

ADD JLabel torque to frame

CREATE JTextField fieldTorque

SET JTextField Bounds of fieldTorque

ADD JTextField fieldTorque to frame

CREATE JLabel fuelTankCapacity text "Fuel Tank Capacity"

SET Bounds of fuelTankCapacity

ADD JLabel fuelTankCapacity to frame

CREATE JTextField fieldTankCapacity

SET JTextField Bounds of fieldTankCapacity

ADD JTextField fieldTankCapacity to frame

CREATE JLabel groundClearance text "Ground Clearance"

SET Bounds of groundClearance

ADD JLabel groundClearance to frame

CREATE JTextField fieldGroundClearance

SET JTextField Bounds of fieldGroundClearance

ADD JTextField fieldGroundClearance to frame

CREATE JLabel iD text "Vehicle ID"

SET Bounds of iD

ADD JLabel iD to frame

CREATE JTextField fieldiD
SET JTextField Bounds of fieldiD
ADD JTextField fieldiD to frame

CREATE JLabel bookedDate text "Booked Date"
SET Bounds of bookedDate
ADD JLabel bookedDate to frame

CREATE JComboBox "y" (year)
SET JComboBox "y" bounds
ADD JComboBox "y" to frame

CREATE JComboBox "m" (month)
SET JComboBox "m" bounds
ADD JComboBox "m" to frame

CREATE JComboBox "d" (day)
SET JComboBox "d" bounds
ADD JComboBox "d" to frame

CREATE JLabel chargeAmount text "Charge Amount"
SET Bounds of chargeAmount
ADD JLabel chargeAmount to frame

CREATE JTextField fieldAmount
SET JTextField Bounds of fieldAmount
ADD JTextField fieldAmount to frame

CREATE JLabel numberOfSeats text "Number of Seats"

SET Bounds of numberOfSeats

ADD JLabel numberOfSeats to frame

CREATE JTextField fieldSeats

SET JTextField Bounds of fieldSeats

ADD JTextField fieldSeats to frame

CREATE JLabel electricVehicleID text "Vehicle ID"

SET Bounds of electricVehicleID

ADD JLabel electricVehicleID to frameElectric

CREATE JTextField fieldElectricId

SET JTextField Bounds of fieldElectricId

ADD JTextField fieldElectricId to frameElectric

CREATE JLabel electricVehicleName text "Vehicle Name"

SET Bounds of electricVehicleName

ADD JLabel electricVehicleName to frameElectric

CREATE JTextField fieldElectricName

SET JTextField Bounds of fieldElectricName

ADD JTextField fieldElectricName to frameElectric

CREATE JLabel electricVehicleWeight text "Vehicle Weight"

SET Bounds of electricVehicleWeight

ADD JLabel electricVehicleWeight to frameElectric

CREATE JTextField fieldElectricWeight

SET JTextField Bounds of fieldElectricWeight

ADD JTextField fieldElectricWeight to frameElectric

CREATE JLabel electricVehicleColor text "Vehicle Color"

SET Bounds of electricVehicleColor

ADD JLabel electricVehicleColor to frameElectric

CREATE JTextField fieldElectricColor

SET JTextField Bounds of fieldElectricColor

ADD JTextField fieldElectricColor to frameElectric

CREATE JLabel electricVehicleSpeed text "Vehicle Speed"

SET Bounds of electricVehicleSpeed

ADD JLabel electricVehicleSpeed to frameElectric

CREATE JTextField fieldElectricSpeed

SET JTextField Bounds of fieldElectricSpeed

ADD JTextField fieldElectricSpeed to frameElectric

CREATE JLabel electricBatteryCapacity text "Battery Capacity"

SET Bounds of electricBatteryCapacity

ADD JLabel electricBatteryCapacity to frameElectric

CREATE JTextField fieldElectricBattery

SET JTextField Bounds of fieldElectricBattery

ADD JTextField fieldElectricBattery to frameElectric

CREATE JLabel purElectricScooterID text "Vehicle ID"

SET Bounds of purElectricScooterID

ADD JLabel purElectricScooterID to frameElectric

CREATE JTextField fieldPurElectricId

SET JTextField Bounds of fieldPurElectricId

ADD JTextField fieldPurElectricId to frameElectric

CREATE JLabel brand text "Brand"

SET Bounds of brand

ADD JLabel brand to frameElectric

CREATE JTextField fieldBrand

SET JTextField Bounds of fieldBrand

ADD JTextField fieldBrand to frameElectric

CREATE JLabel price text "Price"

SET Bounds of price

ADD JLabel price to frameElectric

CREATE JTextField fieldPrice

SET JTextField Bounds of fieldPrice

ADD JTextField fieldPrice to frameElectric

CREATE JLabel chargingTime text "Charging Time"

SET Bounds of chargingTime

ADD JLabel chargingTime to frameElectric

CREATE JTextField fieldChargingTime

SET JTextField Bounds of fieldChargingTime

ADD JTextField fieldChargingTime to frameElectric

CREATE JLabel mileage text "Mileage"

SET Bounds of mileage

ADD JLabel mileage to frameElectric

CREATE JTextField fieldMileage

SET JTextField Bounds of fieldMileage

ADD JTextField fieldMileage to frameElectric

CREATE JLabel range text "Range"

SET Bounds of range

ADD JLabel range to frameElectric

CREATE JTextField fieldRange

SET JTextField Bounds of fieldRange

ADD JTextField fieldRange to frameElectric

CREATE JLabel sellElectricID text "Vehicle ID"

SET Bounds of sellElectricID

ADD JLabel sellElectricID to frameElectric

CREATE JTextField fieldSellElectricID

SET JTextField Bounds of fieldSellElectricID

ADD JTextField fieldSellElectricID to frameElectric

CREATE JLabel sellPrice text "Price"

SET Bounds of sellPrice

ADD JLabel sellPrice to frameElectric

CREATE JTextField fieldSellPrice

SET JTextField Bounds of fieldSellPrice

ADD JTextField fieldSellPrice to frameElectric

CREATE JButton back

SET Bounds of back

ADD back to frame

ADD actionListener to back

FUNCTION actionPerformed

SET frame to visible(FALSE)

SET homeFrame to visible

END

CREATE JButton BTNADD

SET Bounds of BTNADD

ADD BTNADD to frame

ADD actionListener to BTNADD

FUNCTION actionPerformed

addAutoRickshaw()

END

CREATE JButton BTNBK

SET Bounds of BTNBK

ADD BTNBK to frame

ADD actionListener to BTNBK

FUNCTION actionPerformed

bookAutoRickshaw()

END

CREATE JButton BTNCLR

SET Bounds of BTNCLR

ADD BTNCLR to frame

ADD actionListener to BTNCLR

FUNCTION actionPerformed

clearAutoRickshaw()

END

CREATE JButton BTNDS

SET Bounds of BTNDS

ADD BTNDS to frame

ADD actionListener to BTNDS

FUNCTION actionPerformed

displayAutoRickshaw()

END

CREATE JButton back

SET Bounds of back

ADD back to frameElectric

ADD actionListener to back

FUNCTION actionPerformed

SET frameElectric to visible(FALSE)

SET homeFrame to visible

END

CREATE JButton BTNADDScooter

SET Bounds of BTNADDScooter

ADD BTNADDScooter to frameElectric

ADD actionListener to BTNADDScooter

FUNCTION actionPerformed

addElectricScooter()

END

CREATE JButton BTNPurchase

SET Bounds of BTNPurchase

ADD BTNPurchase to frameElectric

ADD actionListener to BTNPurchase

FUNCTION actionPerformed

purElectricScooter()

END

CREATE JButton BTNSell

SET Bounds of BTNSell

ADD BTNSell to frameElectric

ADD actionListener to BTNSell

FUNCTION actionPerformed

sellElectricScooter()

END

CREATE JButton BTNelectricDisplay

SET Bounds of BTNelectricDisplay

ADD BTNelectricDisplay to frameElectric

ADD actionListener to BTNelectricDisplay

FUNCTION actionPerformed

displayElectricScooter()

END

CREATE JButton BTNelectricCLR

SET Bounds of BTNelectricCLR

ADD BTNelectricCLR to frameElectric

ADD actionListener to BTNelectricCLR

FUNCTION actionPerformed

clearElectricScooter()

END

FUNCTION getVehicleName

RETURN String as getText from fieldName

END

FUNCTION getVehicleWeight

RETURN String as getText from fieldWeight

END

FUNCTION getVehicleId

```
        INITIALIZE String vehicleIdText as getText from fieldID
        INITIALIZE int vehicleID to INVALID
        TRY
            UPDATE vehicleID with vehicleIdText
            IF vehicleID <=0
                INITIALIZE vehicleID to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN vehicleID
END


FUNCTION getVehicleColor
        RETURN String as getText from fieldColor
END

FUNCTION getVehicleSpeed
        RETURN String as getText from fieldSpeed
END

FUNCTION getVehicleDisplacement
        INITIALIZE String vehicleDisplacementText as getText from
        EngineDisplacement
```

```
        INITIALIZE int vehicleDisplacement to INVALID
        TRY
            UPDATE vehicleDisplacement with vehicleDisplacementText
            IF vehicleDisplacement <=0
                INITIALIZE vehicleDisplacement to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN vehicleDisplacement
END

FUNCTION getFuelTankCapacity
        INITIALIZE String fuelTankText as getText from fieldTankCapacity
        INITIALIZE int fuelTank to INVALID
        TRY
            UPDATE fuelTank with fuelTankText
            IF fuelTank <=0
                INITIALIZE fuelTank to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN fuelTank
END

FUNCTION getTorque
```

```
        RETURN String as getText from fieldTorque
END


FUNCTION getClearance
        RETURN String as getText from fieldGroundClearance
END


FUNCTION getBookVehicleId
        INITIALIZE String vehicleIdText as getText from fieldiD
        INITIALIZE int vehicleID to INVALID
        TRY
                UPDATE vehicleID with vehicleIdText
                IF vehicleID <=0
                        INITIALIZE vehicleID to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN vehicleID
END
FUNCTION getBooked
        Get String day from selected index
        Get String month from selected index
        Get String year from selected index
        RETURN day+month+year
```

```
END


FUNCTION getChargeAmount
        INITIALIZE String chargeAmountText as getText from fieldAmount
        INITIALIZE int chargeAmount to INVALID
        TRY
                UPDATE chargeAmount with chargeAmountText
                IF chargeAmount <=0
                        INITIALIZE chargeAmount to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN chargeAmount
END


FUNCTION getNumberOfSeats
        INITIALIZE String numberOfSeatsText as getText from fieldSeats
        INITIALIZE int numberOfSeats to INVALID
        TRY
                UPDATE numberOfSeats with numberOfSeatsText
                IF numberOfSeats <=0
                        INITIALIZE numberOfSeats to INVALID
         CATCH NumberFormatException e
         END TRY
```

```
        RETURN numberOfSeats
END


FUNCTION getAddElectricVehicleId
        INITIALIZE String addElectricVehicleIdText as getText from fieldElectricId
        INITIALIZE int addElectricVehicleID to INVALID
        TRY
                UPDATE addElectricVehicleID with addElectricVehicleIdText
                IF addElectricVehicleID <=0
                        INITIALIZE addElectricVehicleID to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN addElectricVehicleID
END


FUNCTION getElectricVehicleName
        RETURN String as getText from fieldElectricName
END


FUNCTION getElectricVehicleWeight
        RETURN String as getText from fieldElectricWeight
END
```

```
FUNCTION getElectricVehicleColor
    RETURN String as getText from fieldElectricColor
END

FUNCTION getElectricVehicleSpeed
    RETURN String as getText from fieldElectricSpeed
END

FUNCTION getElectricBatteryCapacity
    INITIALIZE String addBatteryCapacityText as getText from
               fieldElectricBattery
    INITIALIZE int addBatteryCapacity to INVALID
    TRY
        UPDATE addBatteryCapacity with addBatteryCapacityText
        IF addBatteryCapacity <=0
            INITIALIZE addBatteryCapacity to INVALID
    CATCH NumberFormatException e
    END TRY
    RETURN addBatteryCapacity
END

FUNCTION getBrand
    RETURN String as getText from fieldBrand
```

```
END

FUNCTION getPurElectricVehicleId

    INITIALIZE String purElectricVehicleIdText as getText from
            fieldPurElectricId

    INITIALIZE int purElectricVehicleID to INVALID

    TRY

        UPDATE purElectricVehicleID with purElectricVehicleIdText

        IF purElectricVehicleID <=0

            INITIALIZE purElectricVehicleID to INVALID

    CATCH NumberFormatException e

    END TRY

    RETURN purElectricVehicleID

END


FUNCTION getPrice

    INITIALIZE String priceText as getText from fieldPrice

    INITIALIZE int price to INVALID

    TRY

        UPDATE price with priceText

        IF price <=0

            INITIALIZE price to INVALID

    CATCH NumberFormatException e

    END TRY
```

```
        RETURN price
END

FUNCTION getChargingTime
        RETURN String as getText from fieldChargingTime
END

FUNCTION getMileage
        RETURN String as getText from fieldMileage
END

FUNCTION getRange
        INITIALIZE String rangeText as getText from fieldRange
        INITIALIZE int range to INVALID
        TRY
                UPDATE range with rangeText
                IF range <=0
                        INITIALIZE range to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN range
END
```

```
FUNCTION getSellElectricVehicleId
        INITIALIZE String sellElectricVehicleIdText as getText from
                fieldSellElectricID
    INITIALIZE int sellElectricVehicleID to INVALID
    TRY
        UPDATE sellElectricVehicleID with sellElectricVehicleIdText
        IF sellElectricVehicleID <=0
            INITIALIZE sellElectricVehicleID to INVALID
    CATCH NumberFormatException e
    END TRY
    RETURN sellElectricVehicleID
END

FUNCTION getSellPrice
    INITIALIZE String priceText as getText from fieldSellPrice
    INITIALIZE int price to INVALID
    TRY
        UPDATE price with priceText
        IF price <=0
            INITIALIZE price to INVALID
    CATCH NumberFormatException e
    END TRY
    RETURN price
```

```
END


FUNCTION checkifUnique Boolean"int vehicleId"
        INITIALIZE boolean isUnique to true
        INITIALIZE int range to INVALID
        TRY
                UPDATE range with rangeText
                IF range <=0
                        INITIALIZE range to INVALID
        CATCH NumberFormatException e
        END TRY
        RETURN range
END
FUNCTION addAutoRickshaw
        IF getVehicleName or getVehicleWeight or getVehicleColor or
          getVehicleSpeed  or getTorque or getClearance is empty
                DISPLAY JOptionPane message "Please fill all the fields"
                RETURN
        END IF
        IF getVehicleId or getVehicleDisplacement or getFuelTankCapacity is
                INVALID
                DISPLAY JOptionPane message "Message"
                RETURN
```

```
        END IF

        IF getVehicleId is unique

            ADD new AutoRickShaw to the vehicleList

            DISPLAY JOptionPane message "added"

        ENDIF

END


FUNCTION addElectricScooter

        IF getElectricVehicleName or getElectricVehicleSpeed or
           getElectricVehicleColor or getElectricVehicleWeight is empty

            DISPLAY JOptionPane message "Please fill all the fields"

            RETURN

        END IF

        IF getAddElectricVehicleId or getElectricBatteryCapacity is INVALID

            DISPLAY JOptionPane message "Message"

            RETURN

        END IF

        IF getAddElectricVehicleId is unique

            ADD new ElectricScooter to the vehicleList

            DISPLAY JOptionPane message "added"

        ENDIF

END
```

FUNCTION bookAutoRickshaw

IF getBookVehicleId or getChargeAmount or getNumberOfSeats is INVALID

DISPLAY JOptionPane message invalid VehicleID or ChargeAmount or Number of seats

RETURN

END IF

FOR loop through the vehicleList as autoVehicle

IF getBookVehicleId is equals to getVehicleID

IF autoVehicle instanceof AutoRickshaw

DOWNCAST Vehicle to AutoRickshaw(obj)

IF obj is booked

DISPLAY JOptionPane message AutoRickShaw is already booked

RETURN

ELSE

CALL book autoRickshaw

DISPLAY JOptionPane message AutoRickShaw is booked

RETURN

END IF

END IF

END IF

END FOR

IF vehicle doesn't contains book id

DISPLAY JOptionPane message message Vehicle is not in list

END IF

END

FUNCTION purElectricScooter

IF getPurElectricVehicleId or getPrice or getRange is INVALID

DISPLAY JOptionPane message invalid VehicleID or price or range

RETURN

END IF

FOR loop through the vehicleList as E_Vehicle

IF getPurElectricVehicleId is equals to getVehicleID

IF E_Vehicle instanceof ElectricScooter

DOWNCAST Vehicle to ElectricScooter(obj)

IF obj is Purchased

DISPLAY JOptionPane message ElectricScooter is already booked

RETURN

ELSE

CALL purchase ElectricScooter

DISPLAY JOptionPane message ElectricScooter is booked

RETURN

END IF

END IF

END IF

END FOR

IF vehicle doesn't contains purchase ID

DISPLAY JOptionPane message message Vehicle is not in list

END IF

END

FUNCTION sellElectricScooter

IF getSellElectricVehicleId or getSellPrice is INVALID

DISPLAY JOptionPane message invalid VehicleID or price

RETURN

END IF

FOR loop through the vehicleList as E_Vehicle

IF getSellElectricVehicleId is equals to getVehicleId

IF E_Vehicle instanceof ElectricScooter

DOWNCAST Vehicle to ElectricScooter(obj)

IF obj is Sold

DISPLAY JOptionPane message ElectricScooter is already sold

RETURN

ELSE

CALL sell ElectricScooter

DISPLAY JOptionPane message ElectricScooter is sold

RETURN

END IF

END IF

END IF

END FOR

IF vehicle doesn't contains sellElectricID

DISPLAY JOptionPane message message Vehicle is not in list

END IF

END

FUNCTION DISPLAY AutoRickshaw

IF vehicleList is empty

DISPLAY JOptionPane message There is no vehicle available

END IF

FOR loop through the vehicleList as autoVehicle

IF autoVehicle instanceof AutoRickshaw

DOWNCAST Vehicle to AutoRickshaw(obj)

CALL DISPLAY

DISPLAY JOptionPane message It is printed in Terminal

```
                    RETURN
                END IF
            END FOR
        END

        FUNCTION DISPLAYElectricScooter
            IF vehicleList is empty
                DISPLAY JOptionPane message There is no scooter available
            END IF
            FOR loop through the vehicleList as E_Vehicle
                IF E_Vehicle instanceof ElectricScooter
                    DOWNCAST Vehicle to ElectricScooter(obj)
                    CALL DISPLAY
                    DISPLAY JOptionPane message It is printed in Terminal
                    RETURN
                END IF
            END FOR
        END
        FUNCTION clearAutoRickshaw
             SET fieldID Text " "
            SET fieldName Text " "
            SET fieldWeight Text " "
            SET fieldColor Text " "
```

```
        SET fieldSpeed Text " "
        SET fieldEngineDisplacement Text " "
        SET fieldTorque Text " "
        SET fieldTankCapacity Text " "
        SET fieldGroundClearance Text " "
        SET fieldiD Text " "
        SET y SelectedIndex 0
        SET m SelectedIndex 0
        SET d SelectedIndex 0
        SET fieldSeats Text " "
        SET fieldAmount Text " "
END

FUNCTION clearElectricScooter
        SET fieldElectricName Text " "
        SET fieldElectricId Text " "
        SET fieldElectricWeight Text " "
        SET fieldElectricColor Text " "
        SET fieldElectricSpeed Text " "
        SET fieldElectricBattery Text " "
        SET fieldPurElectricId Text " "
        SET fieldBrand Text " "
        SET fieldPrice Text " "
```

```
            SET fieldChargingTime Text " "
            SET fieldMileage Text " "
            SET fieldRange Text " "
            SET fieldSellElectricID Text " "
            SET fieldSellPrice Text " "
        END
        FUNCTION main
            CALL TransportGUI()
        END
END
```

4. Method Description

1. TransportGUI()

This Method contains GUI aspect of the code such as JLabels, JTextfields, JFrames and JButtons of AutoRickshaw and ElectricScooter.

2. addAutoRickshaw()

This method is used for adding AutoRickshaw. It checks whether the conditions are fulfilled or not to add AutoRickshaw and displays the dialog box accordingly.

3. addElectricScooter()

This method is used for adding ElectricScooter. It checks whether the conditions are fulfilled or not to add ElectricScooter and displays the dialog box accordingly.

4. bookAutoRickshaw()

This method is used for booking AutoRickshaw. It checks whether the conditions are fulfilled or not to book AutoRickshaw and displays the dialog box accordingly.

5. purElectricScooter()

This method is used for purchasing ElectricScooter. It checks whether the conditions are fulfilled or not to purchase Electric Scooter and displays the dialog box accordingly.

6. sellElectricScooter()

This method is used for selling ElectricScooter. It checks whether the conditions are fulfilled or not to sell Electric Scooter and displays the dialog box accordingly.

7. displayAutoRickshaw()

This method is for the Display Button, on clicking the button it displays all the details of AutoRickshaw text fields in the terminal.

8. displayElectricScooter()

This method is for the Display Button, on clicking the button it displays all the details of ElectricScooter text fields in the terminal.

9. clearAutoRickshaw()

This method is used to clear all the JTextfield of AutoRickshaw when clear button is clicked.

10. clearElectricScooter()

This method is used to clear all the JTextfield of Electric Scooter when clear button is clicked.

11. actionPerformed(ActionEvent e)

This method is an abstract method in ActionListener Interface which call on an action occurs.

12. Getter Methods

- getVehicleId(): It is a method used to return the value of vehicleID from textfield.
- getVehicleWeight(): It is a method used to return the value of vehicleWeight from textfield.
- getVehicleColor(): It is a method used to return the value of vehicleColor from textfield.
- getVehicleSpeed():It is a method used to return the value of vehicleSpeed from textfield.
- getVehicleDisplacement():It is a method used to return the value of vehicleDisplacement from textfield.
- getTorque():It is a method used to return the value of torque from textfield.
- getNumberOfSeats():It is a method used to return the value of numberOfSeats from textfield.
- getFuelTankCapacity():It is a method used to return the value of fuelTankCapacity from textfield.
- getClearance():It is a method used to return the value of groundClearance from textfield.
- getChargeAmount():It is a method used to return the value of chargeAmount from textfield.
- getBooked():It is a method used to return the value of bookedDate from textfield.

- getBookVehicleId():It is a method used to return the value of vehicleId from textfield.
- getAddElectricVehicleId(): It is a method used to return the value of electricVehicleID from textfield.
- getElectricVehicleName():It is a method used to return the value of electricVehicleName from textfield.
- getElectricVehicleWeight():It is a method used to return the value of electricVehicleWeight from textfield.
- getElectricVehicleColor():It is a method used to return the value of electricVehicleColor from textfield.
- getElectricVehicleSpeed():It is a method used to return the value of electricVehicleSpeed from textfield.
- getElectricBatteryCapacity():It is a method used to return the value of electricBatteryCapacity from textfield.
- getBrand ():It is a method used to return the value of brand from textfield.
- getPurElectricVehicleId():It is a method used to return the value of purElectricScooterID from textfield.
- getPrice ():It is a method used to return the value of price from textfield.
- getChargingTime():It is a method used to return the value of chargingTime from textfield.
- getMileage ():It is a method used to return the value of mileage from textfield.
- getRange ():It is a method used to return the value of range from textfield.
- getSellElectricVehicleId ():It is a method used to return the value of sellElectricID from textfield.
- getSellPrice ():It is a method used to return the value of sellPrice from textfield.
- checkIfUnique():It is a method which returns true value when the vehicleId is unique and return's false when vehicleId is not Unique.

5. Testing

5.1 Test 1 Run program in Command prompt.

Objectives	To test the program can be compiled and run the program using command prompt.
Actions	Program was compiled and runed in command prompt using following syntax: javac TransportGUI.java java TransportGUI
Expectations	The program should be complied and display the GUI.
Actual Output	GUI of TransportationGUI was displayed.
Conclusion	Test was Successful.

Table 2 : Test 1

Figure 3 : Command Code to Compile and run

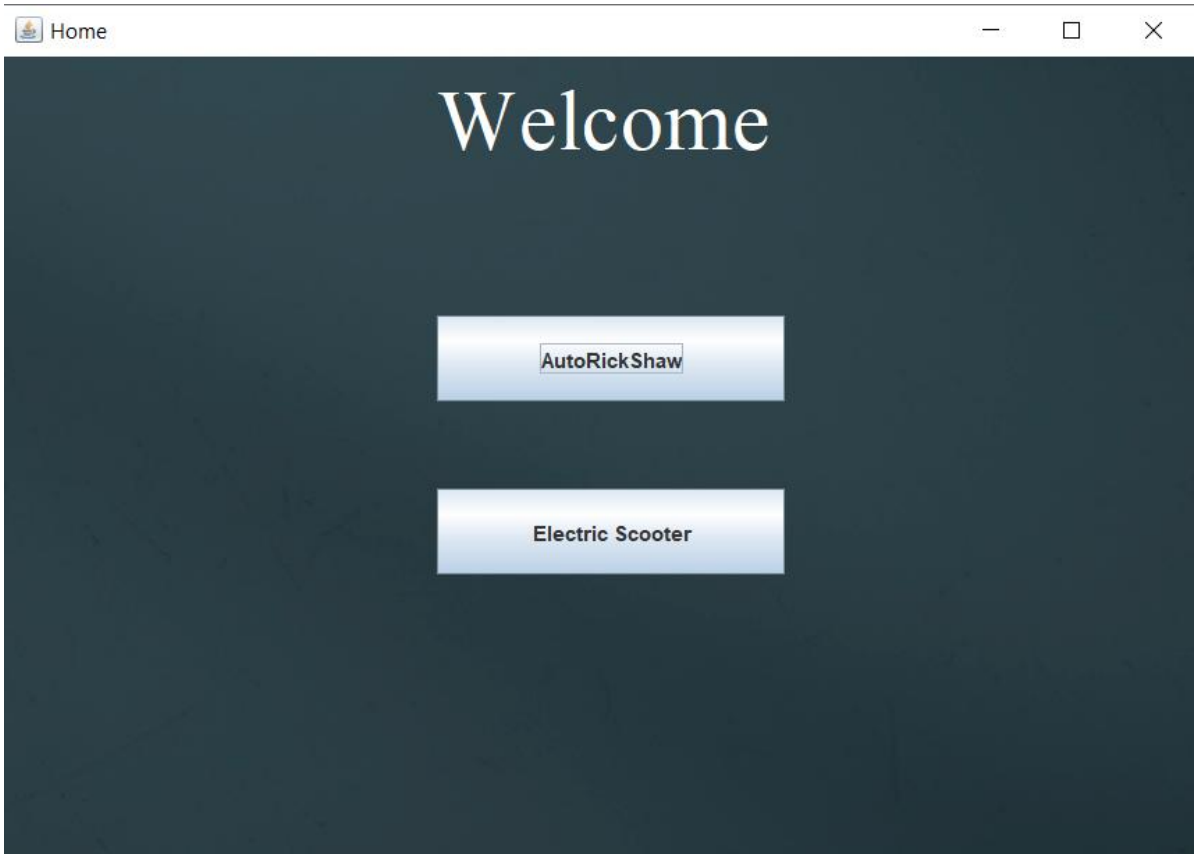

Figure 4 : GUI

5.2 Test 2

a.) To ADD AutoRickshaw

Objectives	To **ADD** AutoRickshaw
Actions	The textfield to **ADD** AutoRickshaw was filled according to the labels.
Expectations	The AutoRickshaw should be **ADD**ed displaying appropriate dialog box.
Actual Output	The AutoRickshaw was **ADD**ed displaying appropriate dialog box.
Conclusion	Test was Successful.

Table 3 : Test 2 a: Add AutoRickshaw

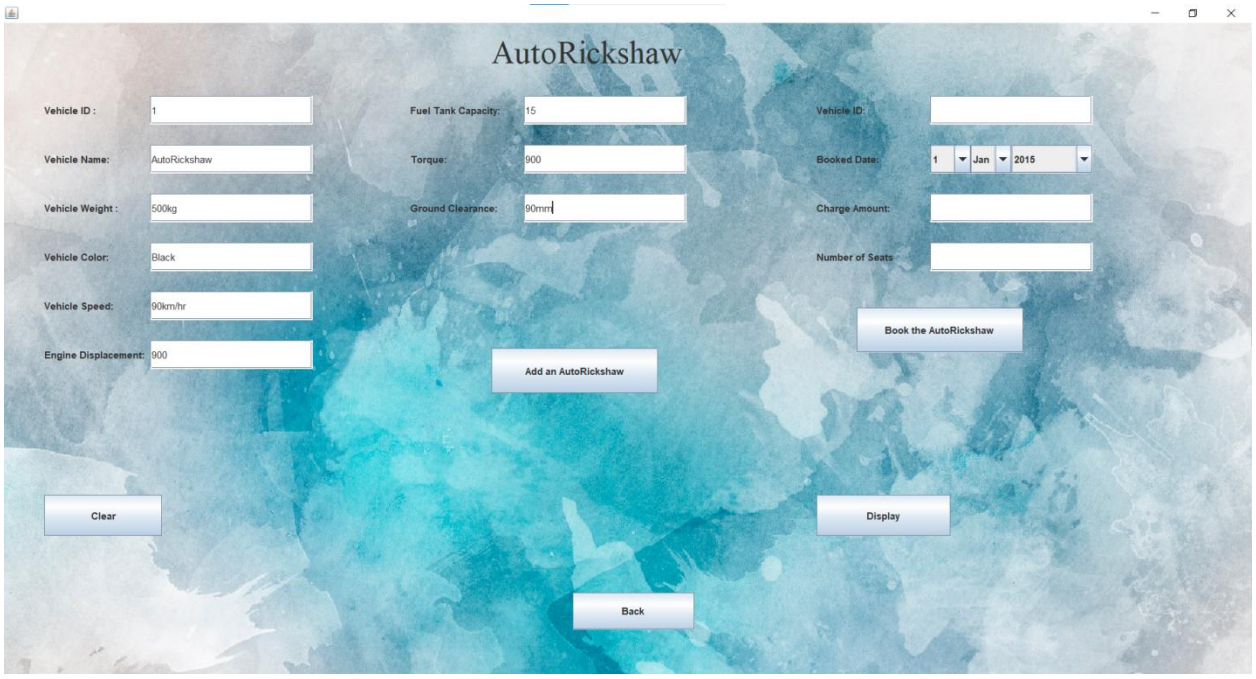

Figure 5 : ADDing Data of AutoRickshaw

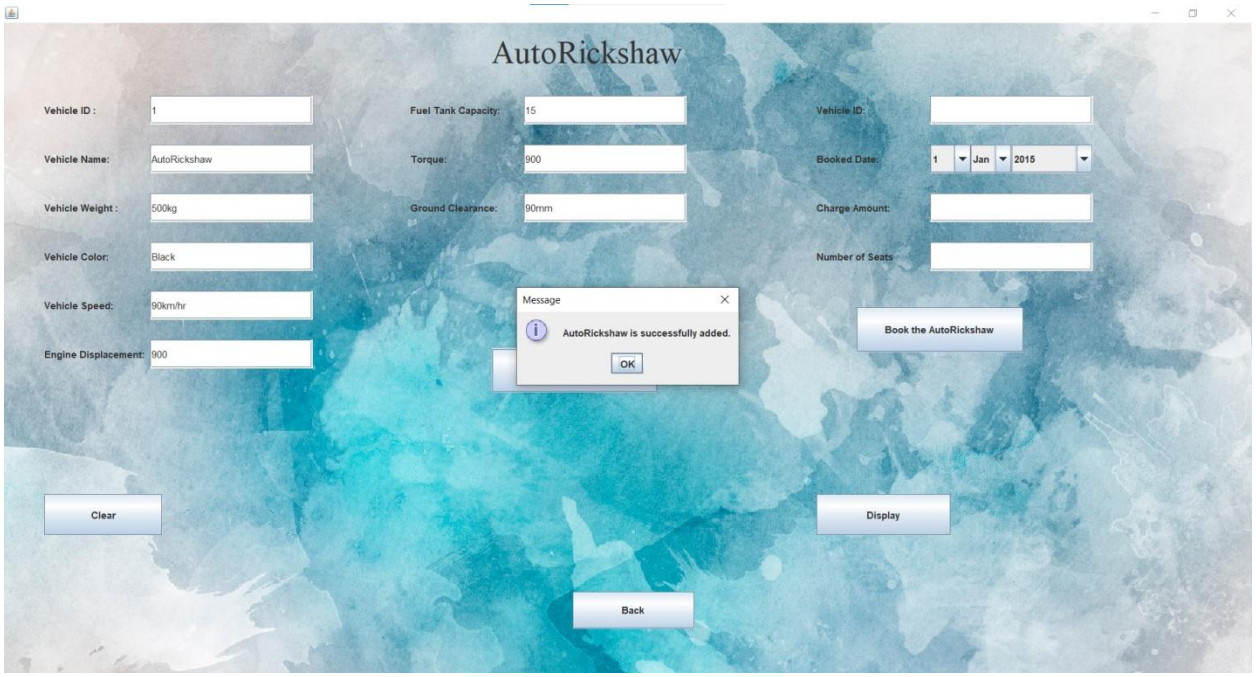

Figure 6 : Dialog box After Clicking ADD an AutoRickshaw

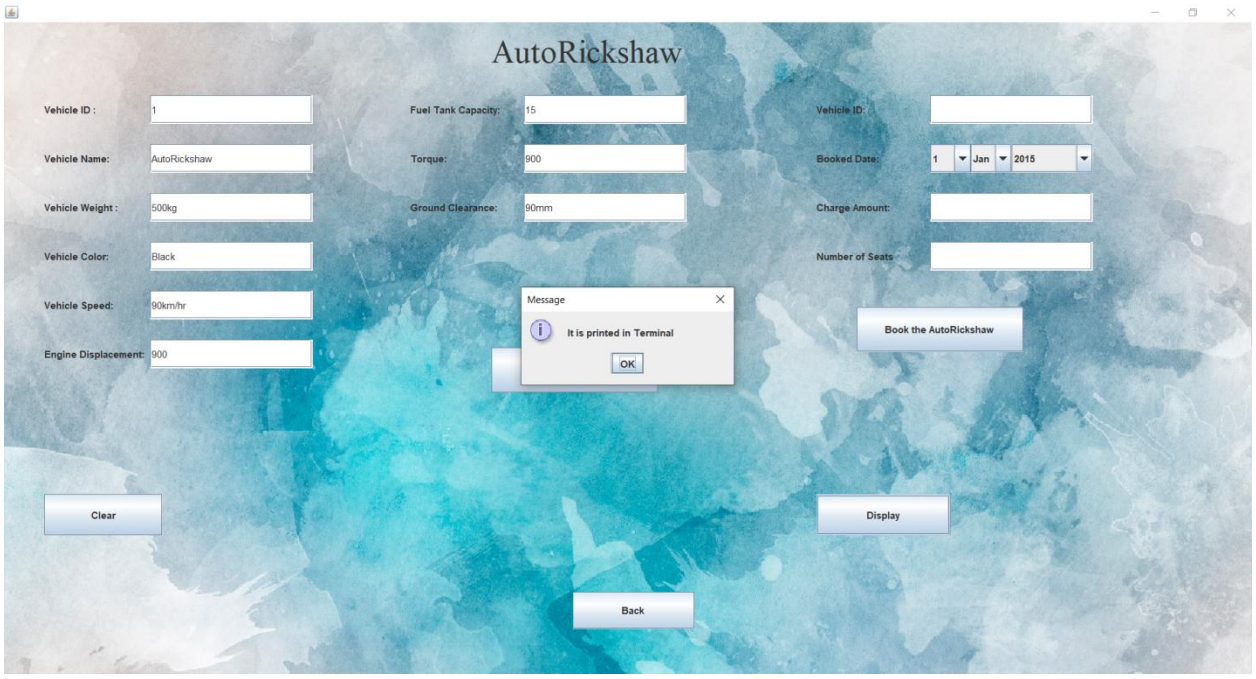

Figure 7 : Dialog box after clicking Display

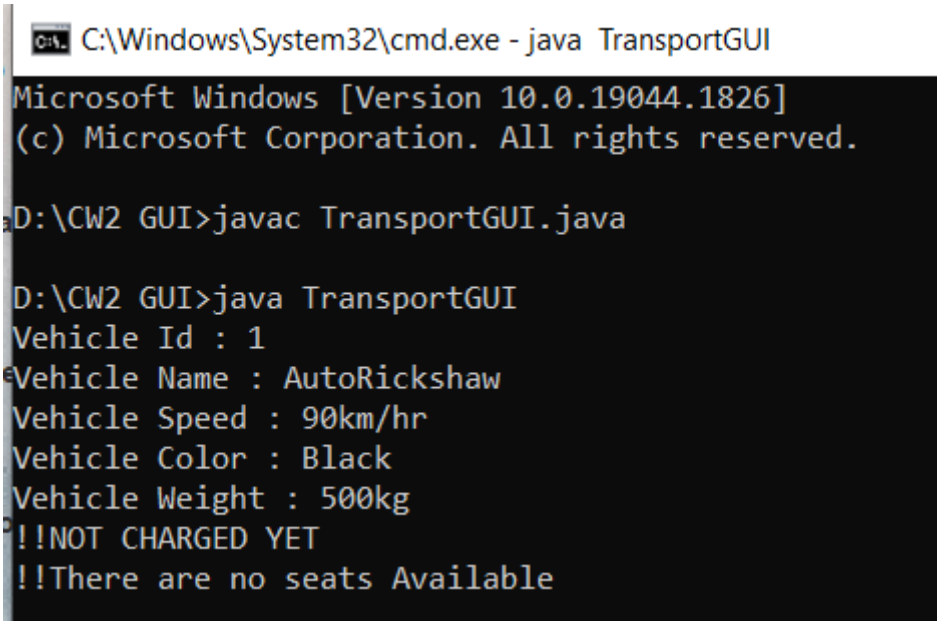

Figure 8 : Display

b.) To book AutoRickshaw

Objectives	To book AutoRickshaw
Actions	The textfield to book AutoRickshaw was filled according to the labels.

Expectations	The AutoRickshaw should be booked displaying appropriate dialog box.
Actual Output	The AutoRickshaw was booked displaying appropriate dialog box.
Conclusion	Test was Successful.

Table 4 : Test 2 b: Book AutoRickshaw

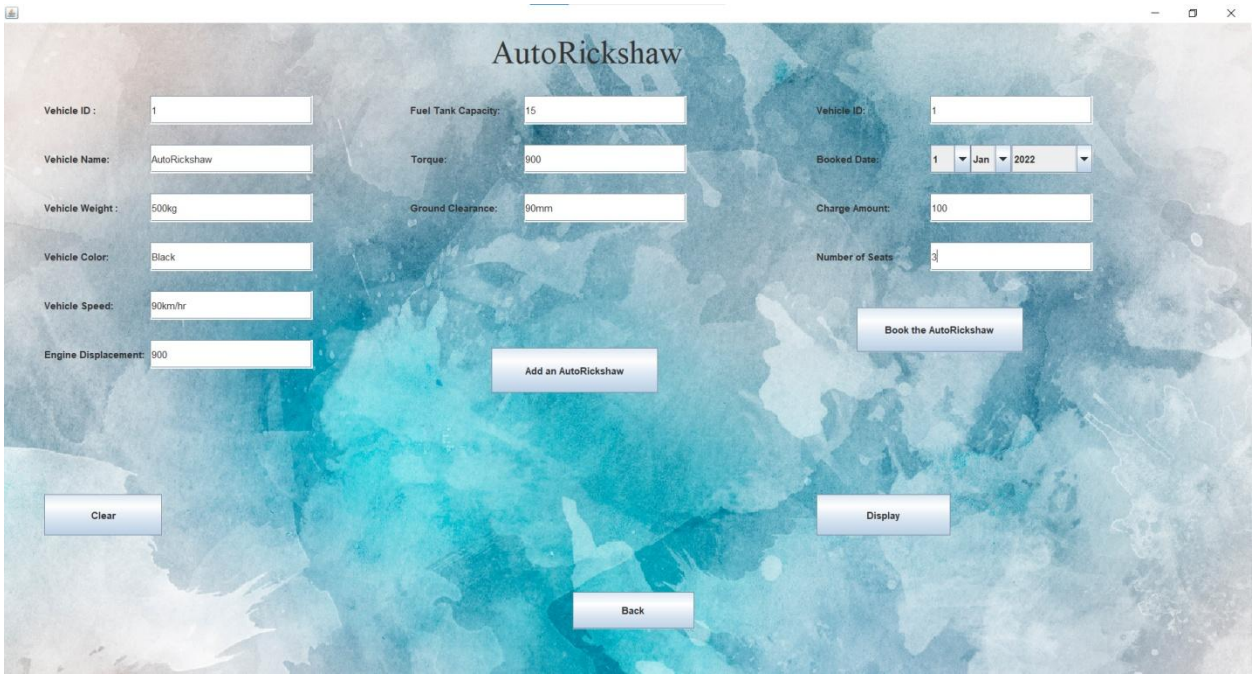

Figure 9 : ADDing data to book AutoRickshaw

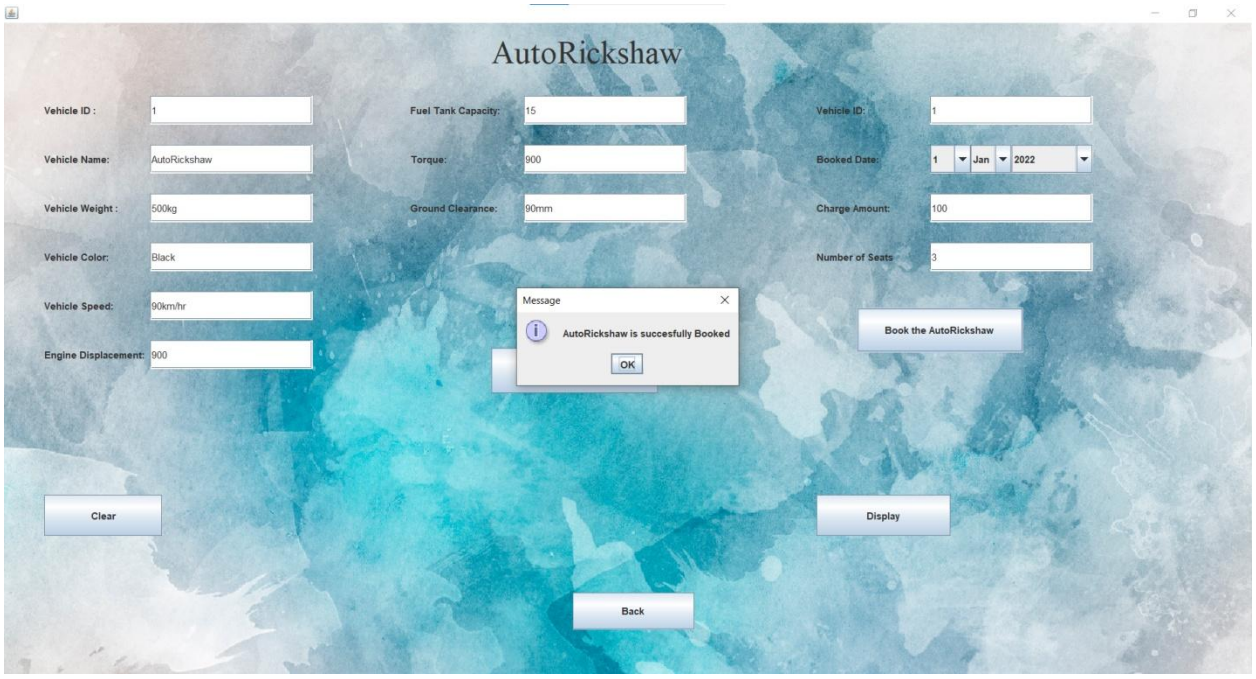

Figure 10 : Dialog box After Clicking Book the AutoRickshaw

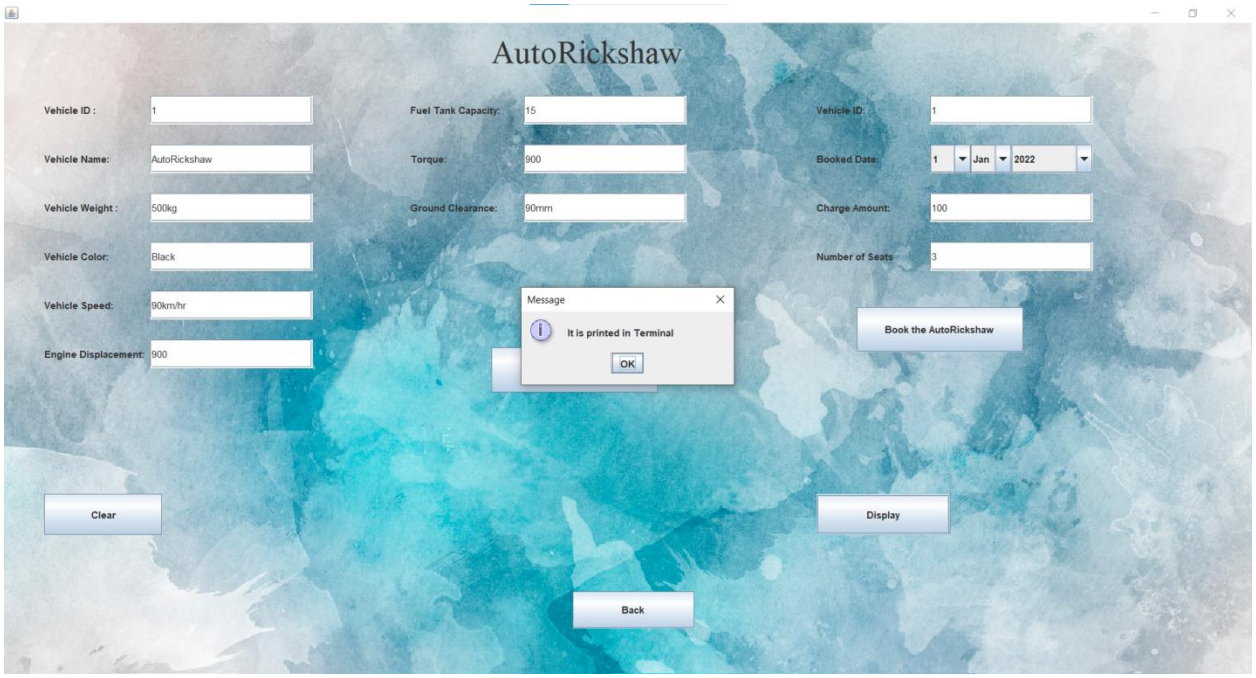

Figure 11 : Dialog box after clicking Display

Figure 12 : Display

c.) To ADD ElectricScooter

Objectives	To **ADD** Electric Scooter
Actions	The textfield to **ADD** ElectricScooter was filled according to the labels.
Expectations	The ElectricScooter should be **ADD**ed displaying appropriate dialog box.
Actual Output	The ElectricScooter was **ADD**ed displaying appropriate dialog box.
Conclusion	Test was Successful.

Table 5 : Test 2 c: Add ElectricScooter

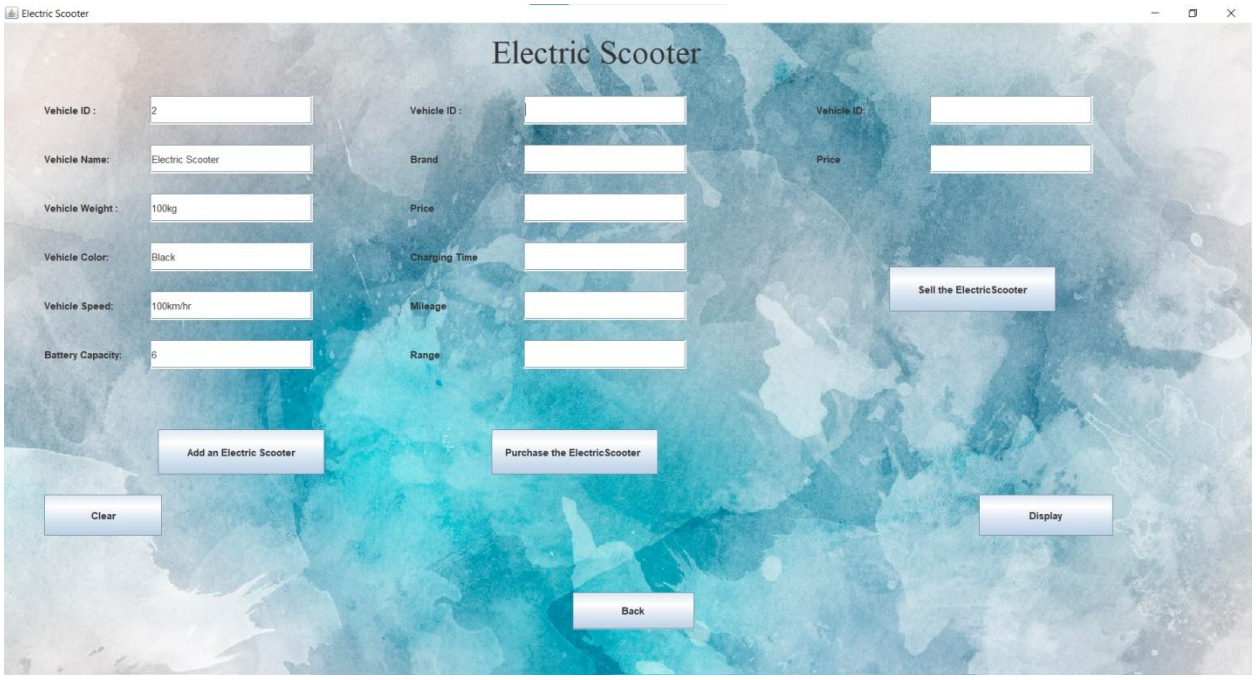

Figure 13 : Entering data to ADD Electric Scooter

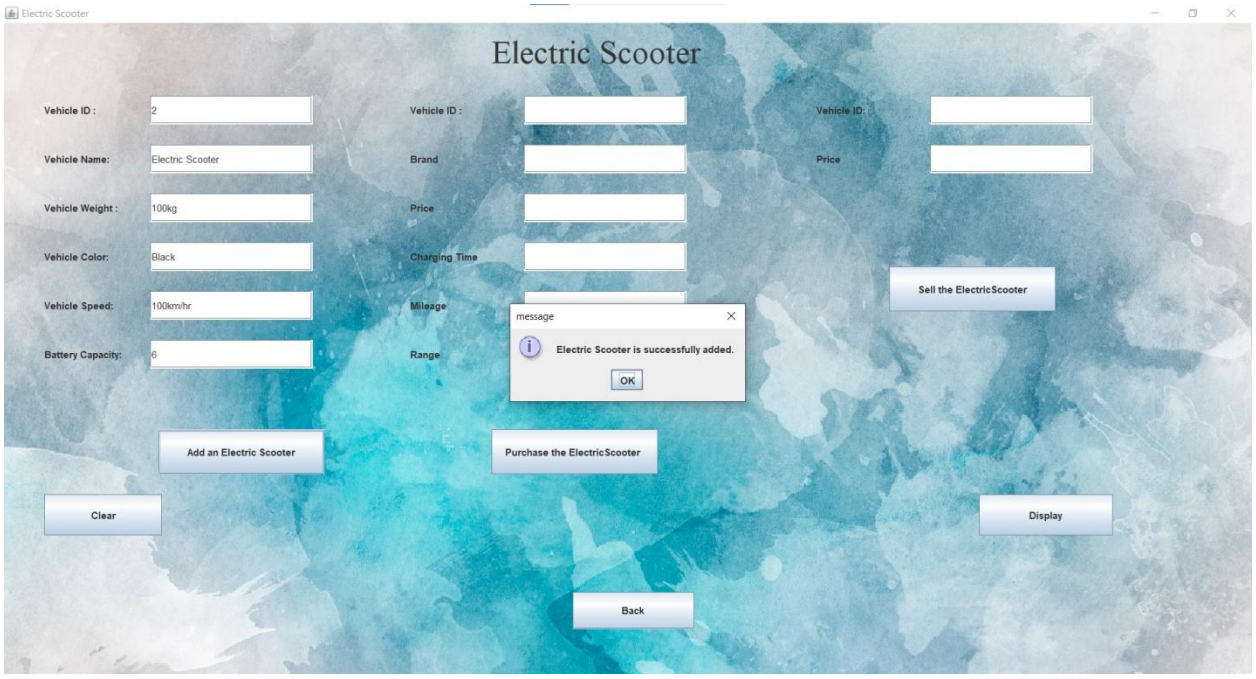

Figure 14 : Dialog box After Clicking ADD an Electric Scooter

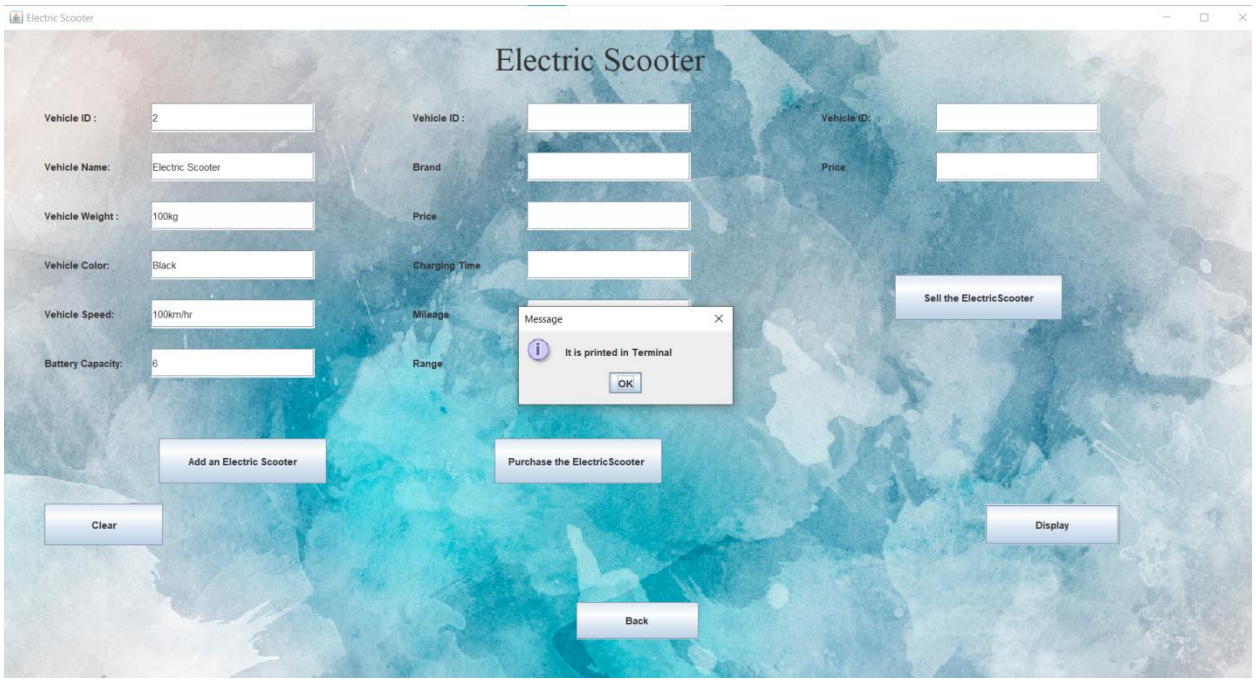

Figure 15 : Dialog box after clicking Display

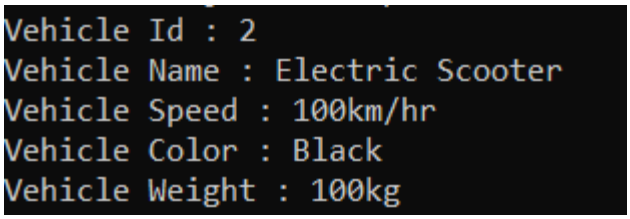

Figure 16 : Display

d.) To purchase Electric Scooter

Objectives	To purchase Electric Scooter
Actions	The textfield to purchase ElectricScooter was filled according to the labels.
Expectations	The ElectricScooter should be purchased displaying appropriate dialog box.
Actual Output	The ElectricScooter was purchased displaying appropriate dialog box.

Conclusion	Test was Successful.

Table 6 : Test 2 d: Purchase ElectricScooter

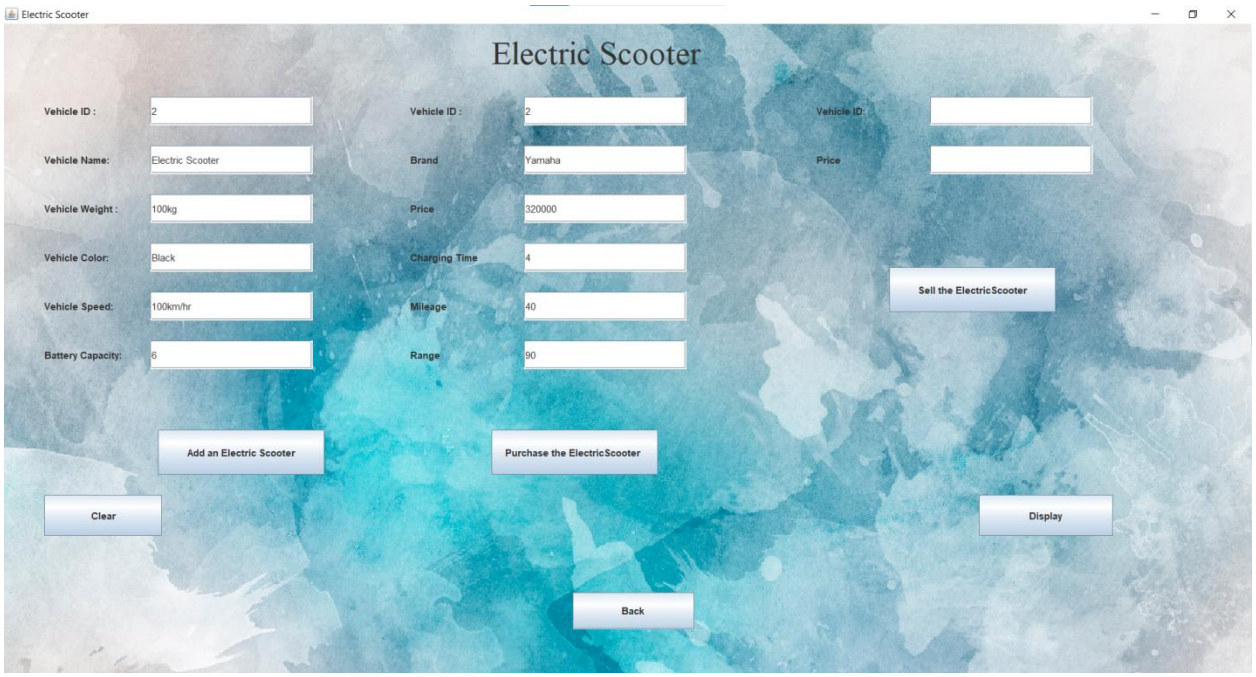

Figure 17 : Entering data to Purchase Electric Scooter

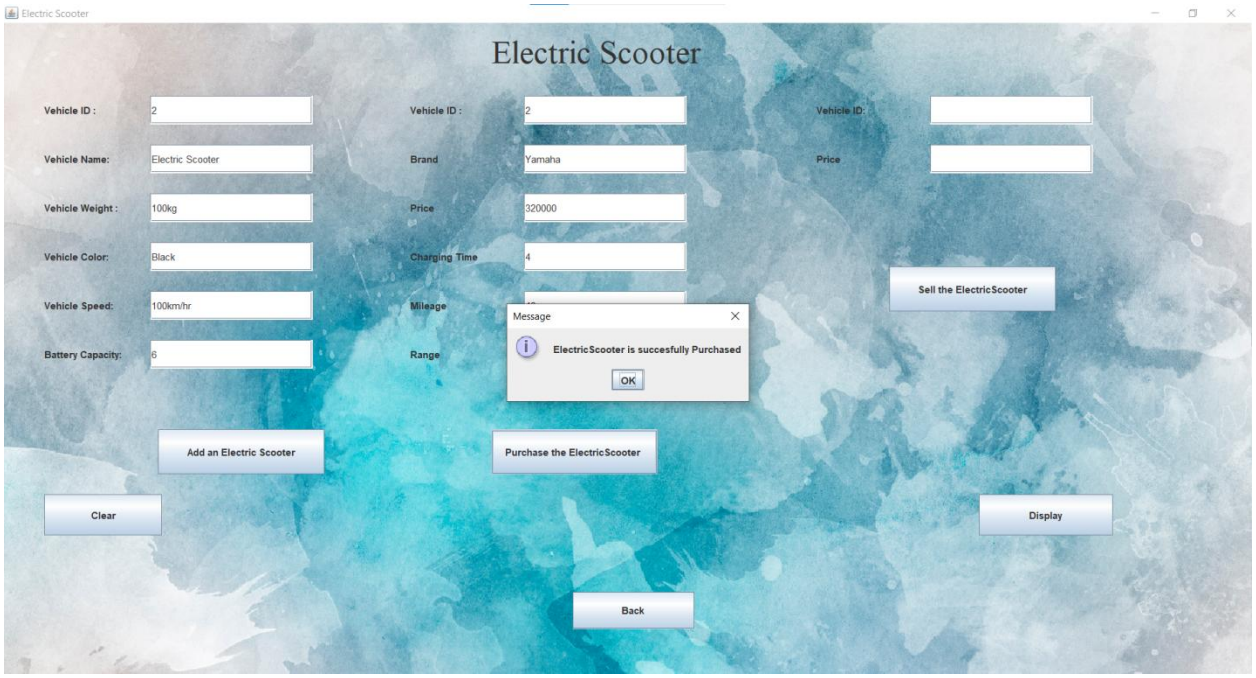

Figure 18 : Dialog box After Clicking Purchase the Electric Scooter

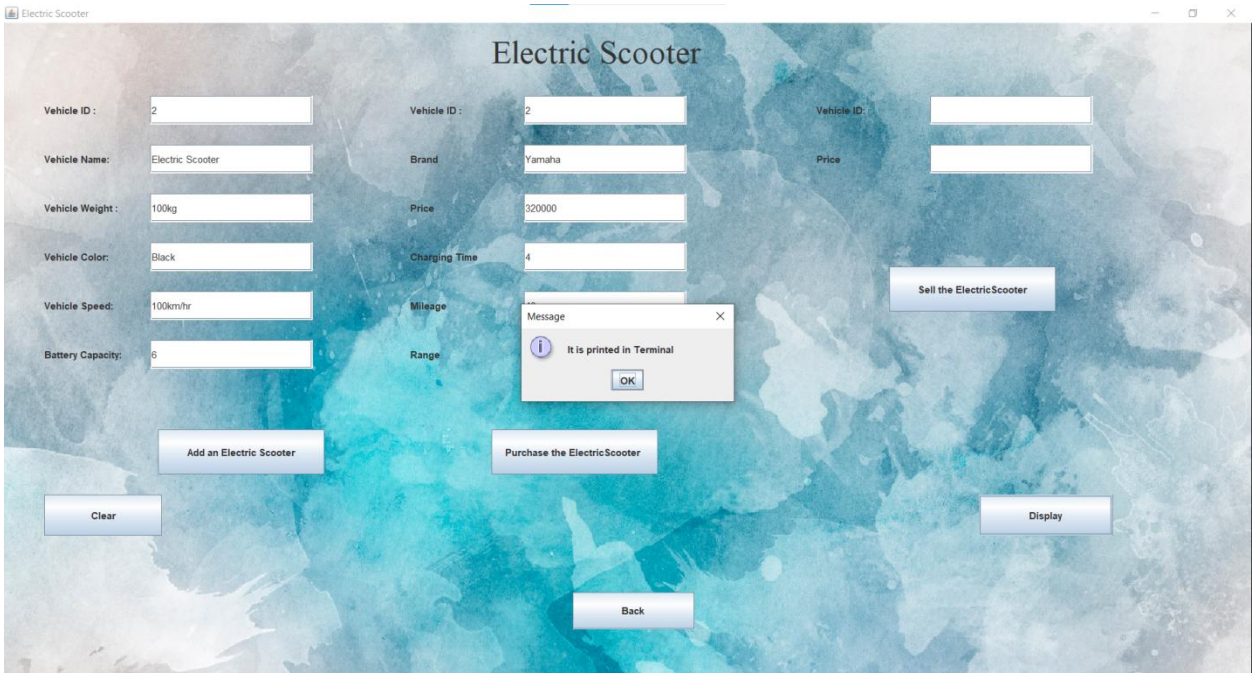

Figure 19 : Dialog box after clicking Display

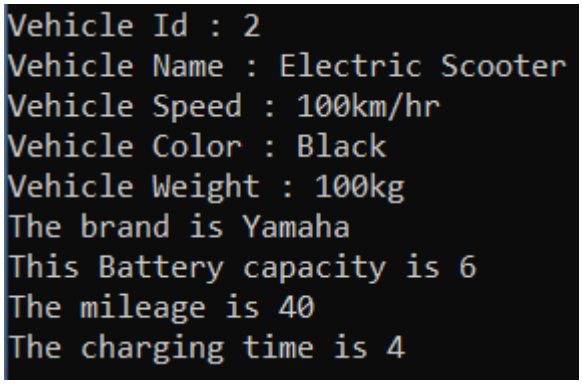

Figure 20 : Display

e.) To sell ElectricScooter

Objectives	To sell ElectricScooter
Actions	The textfield to sell ElectricScooter was filled according to the labels.
Expectations	The ElectricScooter should sold displaying appropriate dialog box.
Actual Output	The ElectricScooter was sold displaying appropriate dialog box.
Conclusion	Test was Successful.

Table 7 : Test 2 e: Sell ElectricScooter

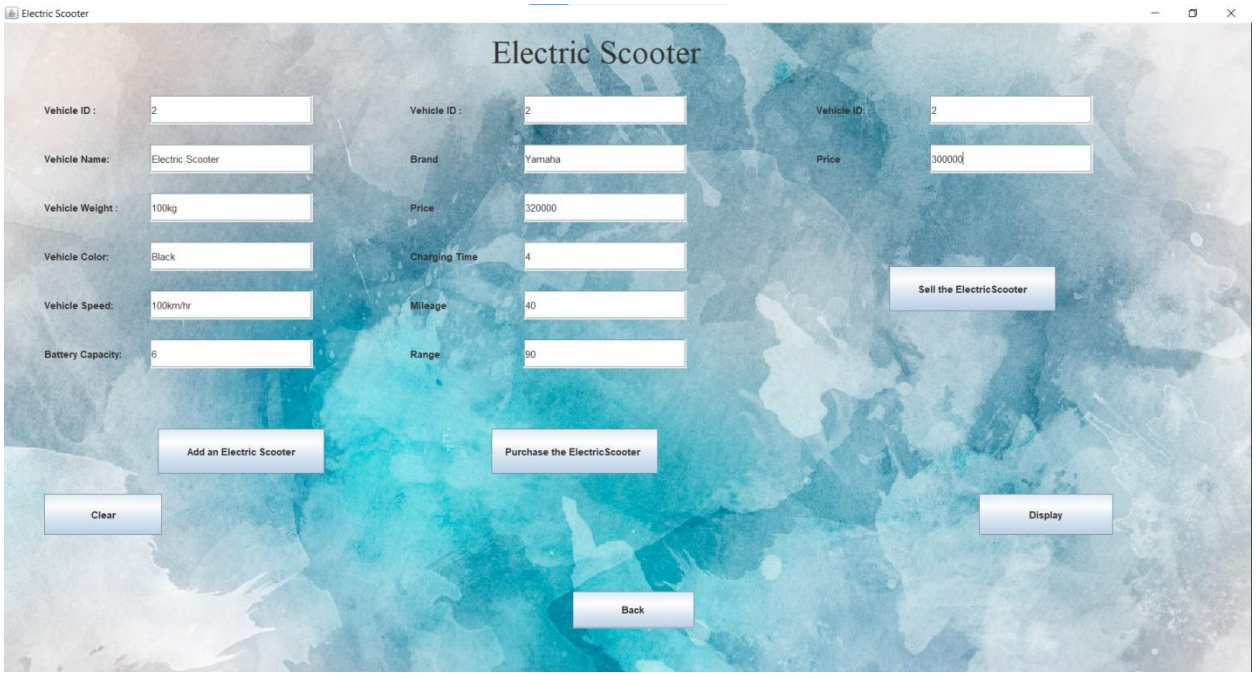

Figure 21 : Entering data to Sell Electric Scooter

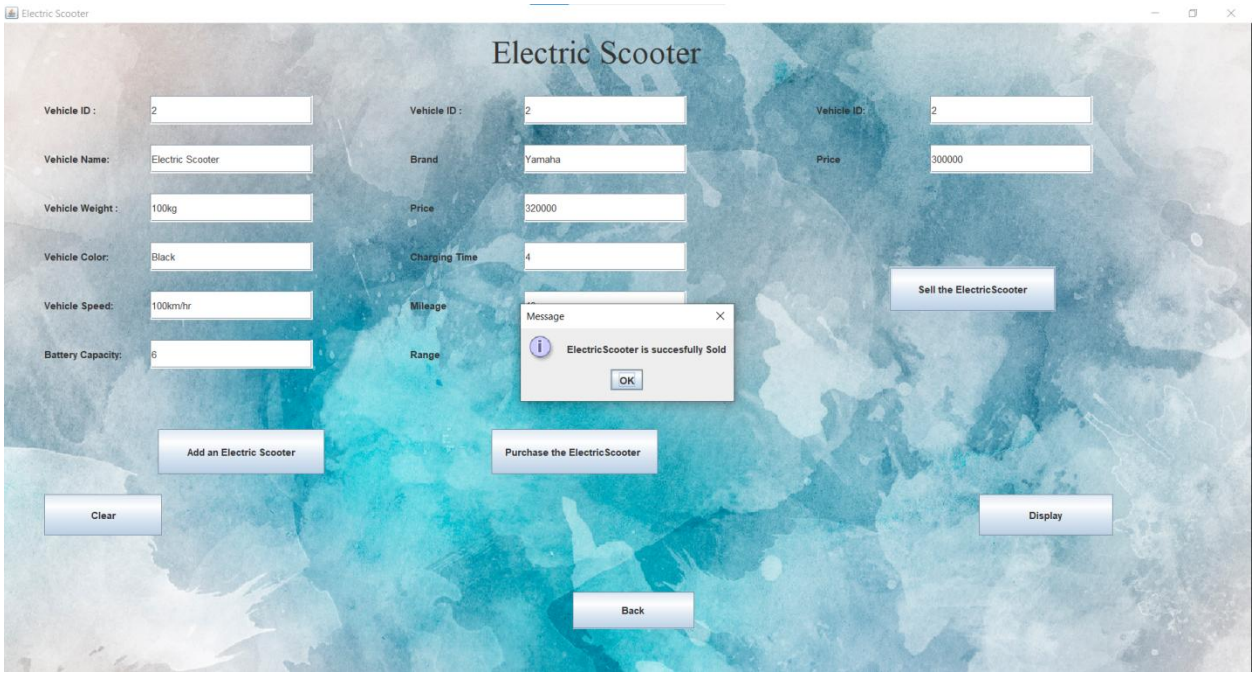

Figure 22 : Dialog box After Clicking Sell the Electric Scooter

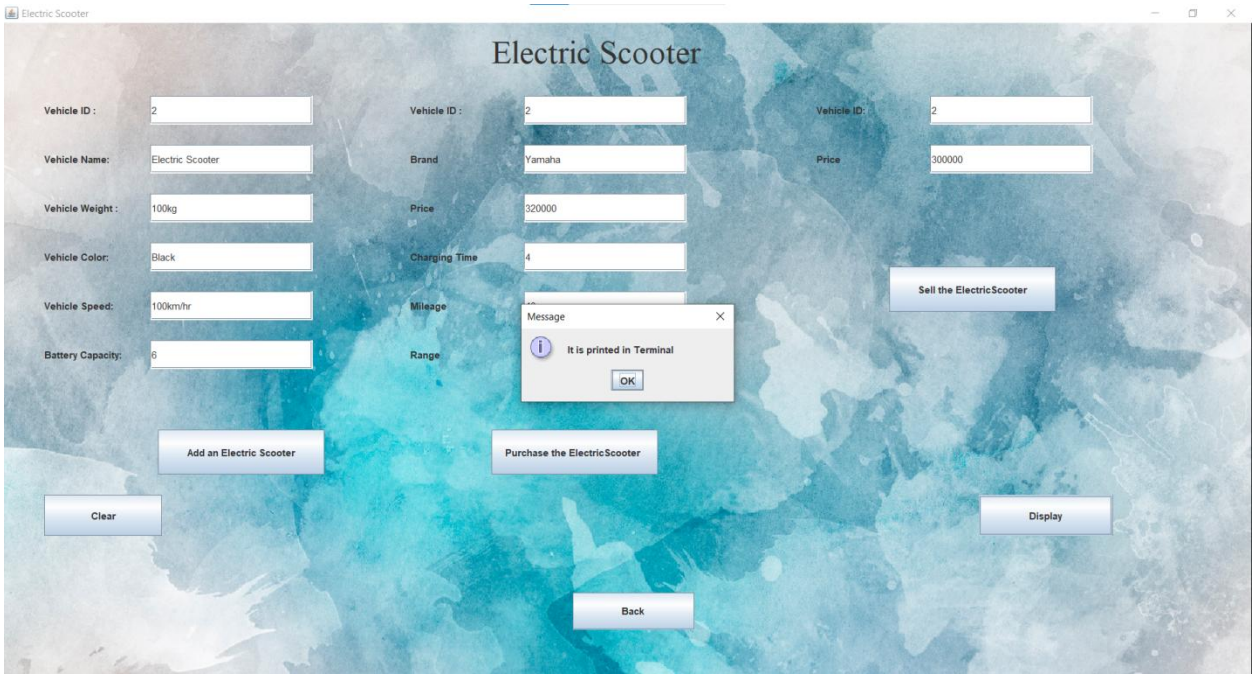

Figure 23 : Dialog box after clicking Display

```
Vehicle Id : 2
Vehicle Name : Electric Scooter
Vehicle Speed : 100km/hr
Vehicle Color : Black
Vehicle Weight : 100kg
The brand is Yamaha
This Battery capacity is 6
The mileage is 40
The charging time is 4
Vehicle Id : 2
Vehicle Name : Electric Scooter
Vehicle Speed : 100km/hr
Vehicle Color : Black
Vehicle Weight : 100kg
```

Figure 24 : Display

5.3 Test 3

a.) Trying to purchase an Electric Scooter which is not added.

Objectives	Try to purchase an Electric Scooter which is not added.
Actions	New VehicleId 4 was entered which was not added.
Expectations	Dialog box with" Vehicle is not on list" should appear.
Actual Output	Dialog box with" Vehicle is not on list" was appeared.
Conclusion	Test was Successful.

Table 8 : Test 3) a

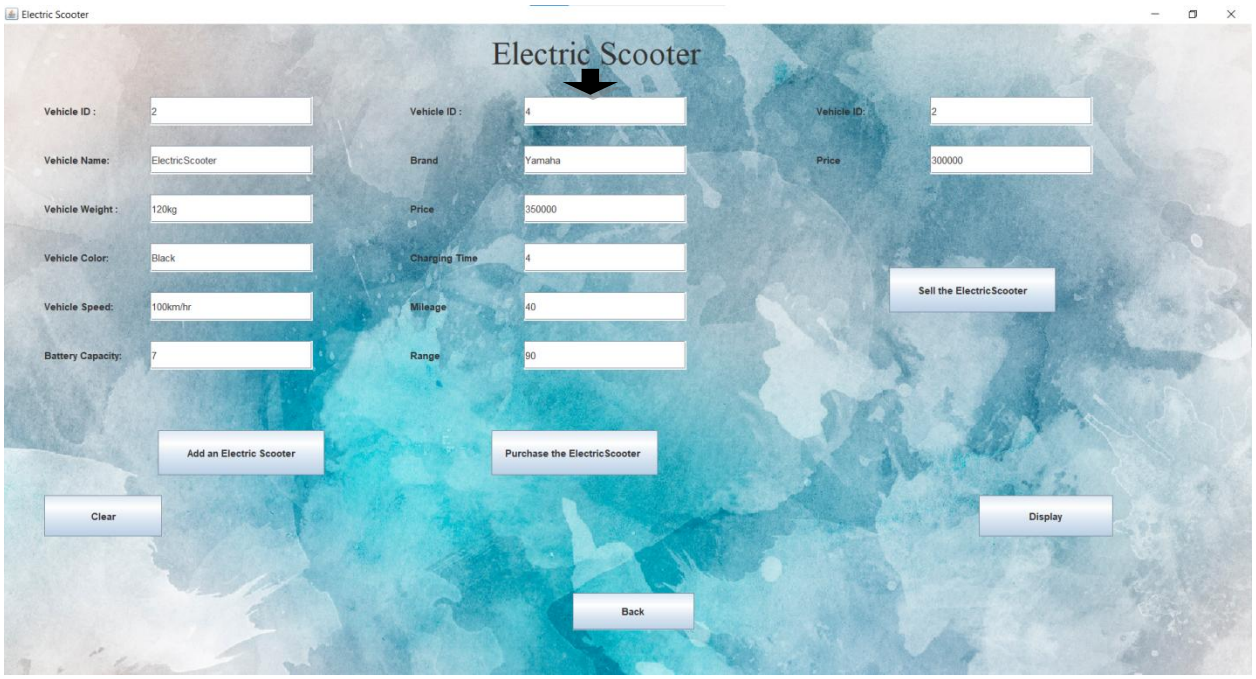

Figure 25 : Entering VehicleID

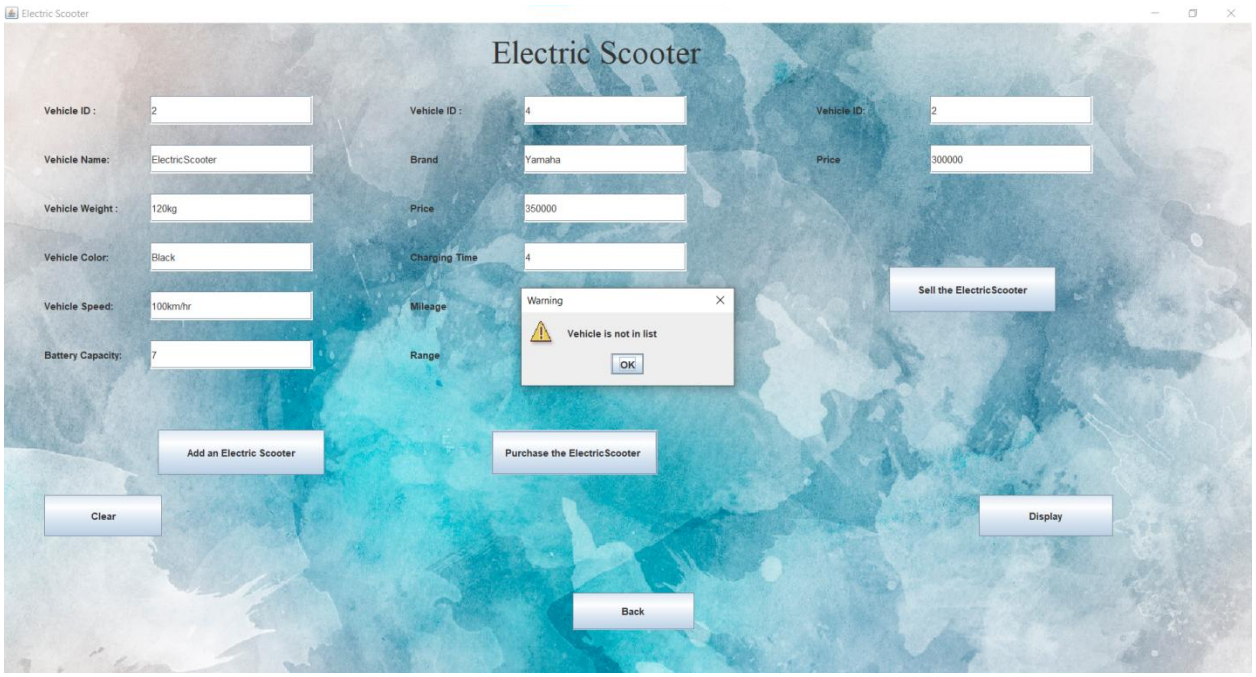

Figure 26 : Dialog Box after entering VehicleId

b.) Trying to ADD again the AutoRickshaw which is already added

Objectives	Trying to **ADD** again the AutoRickshaw which is already added.
Actions	**ADD** Autorickshaw with the same vehicleId twice.
Expectations	Dialog box with" Vehicle Id must be unique" Should appear.
Actual Output	Dialog box with" Vehicle Id must be unique" was appeared.
Conclusion	Test was Successful.

Table 9 : Test 3) b

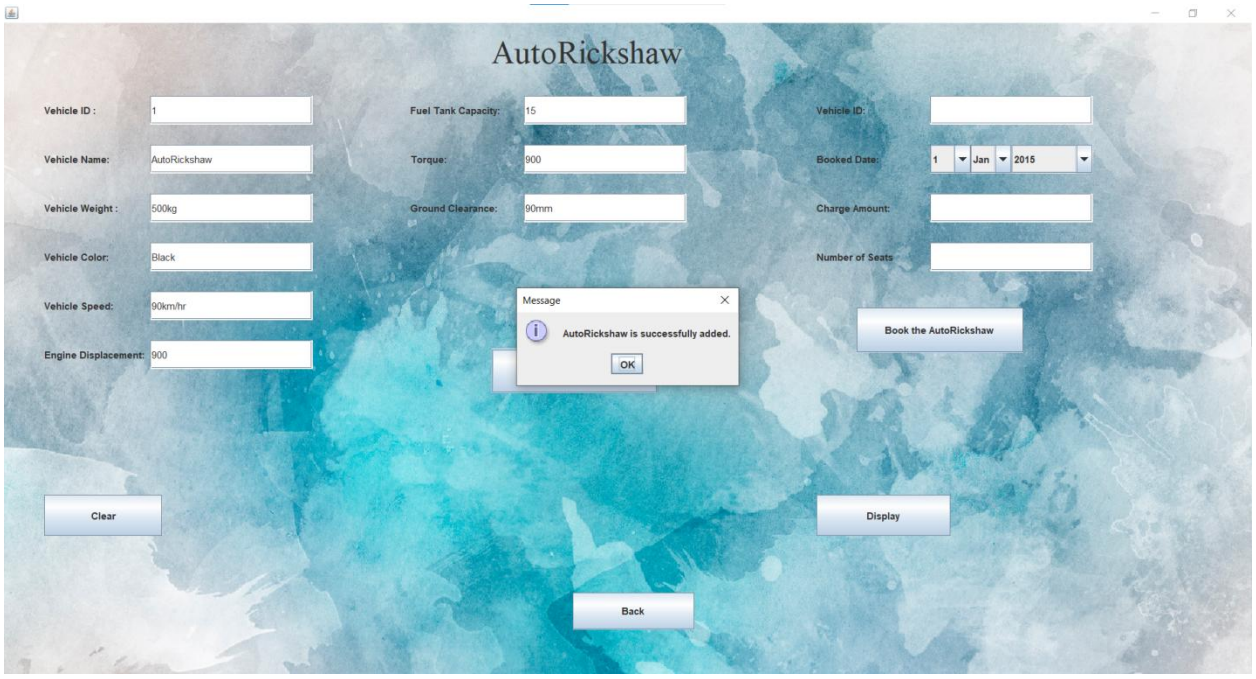

Figure 27 : ADDing AutoRickshaw before ADDing twice

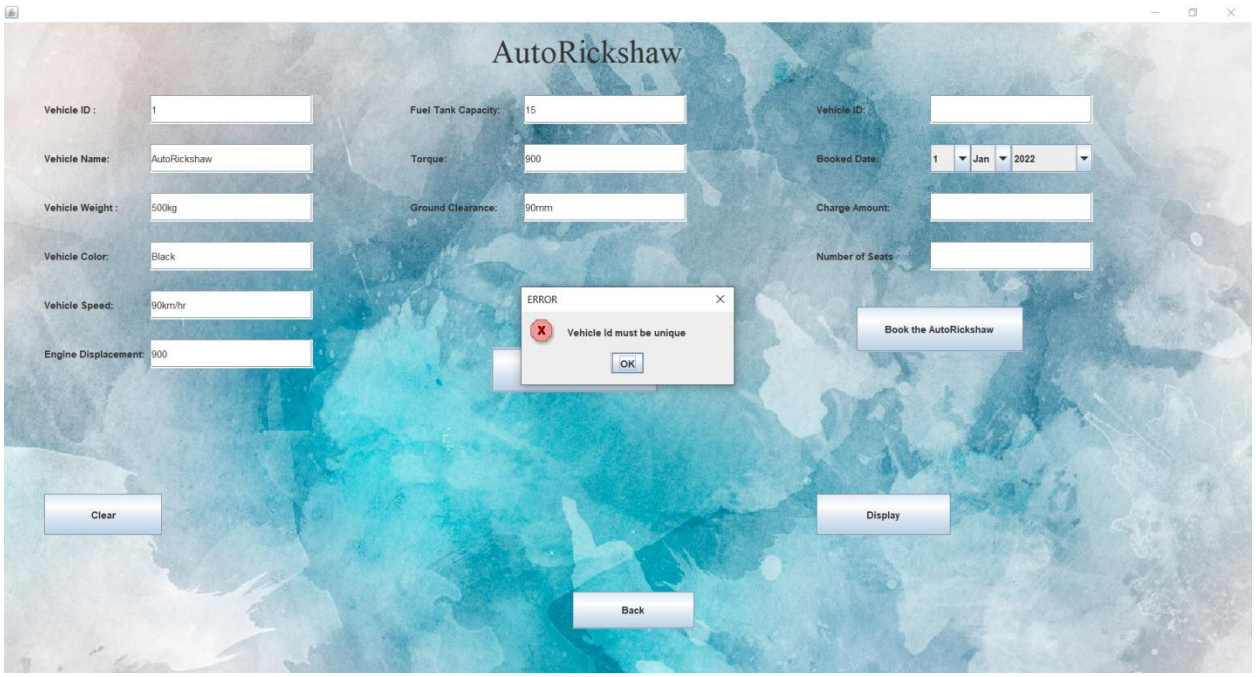

Figure 28 : ADDing Autorickshaw Again

c.) To check the validation of data type

Objectives	To check the validation of data type
Actions	String value was entered in Engine Displacement.
Expectations	Dialog box with" Please enter integer value in VehicleId or EngineDisplacement or Fuel Tank Capacity" Should appear.
Actual Output	Dialog box with" Please enter integer value in VehicleId or EngineDisplacement or Fuel Tank Capacity" was appeared.
Conclusion	Test was Successful.

Table 10 : Test 3) c

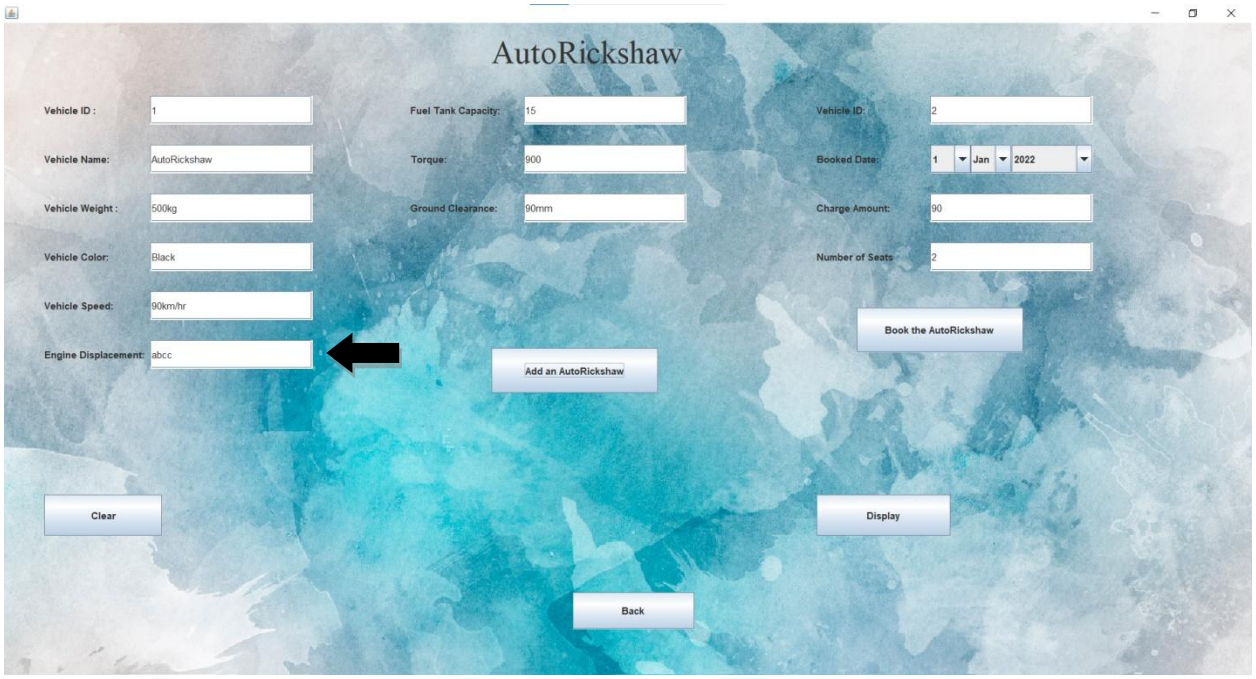

Figure 29 : Entering wrong data type

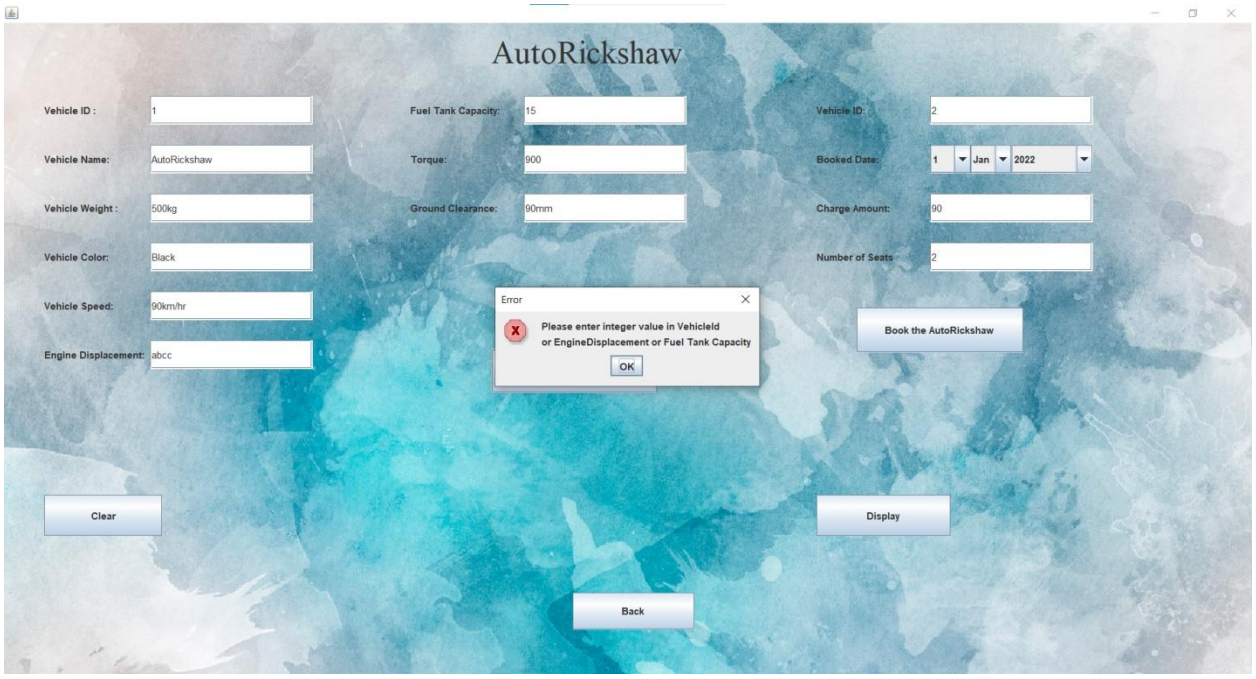

Figure 30 : Dialog Box after entering data

d.) Trying to sell the Electric Scooter which is already sold

Objectives	Trying to sell the Electric Scooter which is already sold
Actions	Sell Electric Scooter with the same VehicleId twice.
Expectations	Dialog box with ”Electric Scooter is already sold” Should appear
Actual Output	Dialog box with ”Electric Scooter is already sold” was appeared
Conclusion	Test was Successful.

Table 11 : Test 3) d

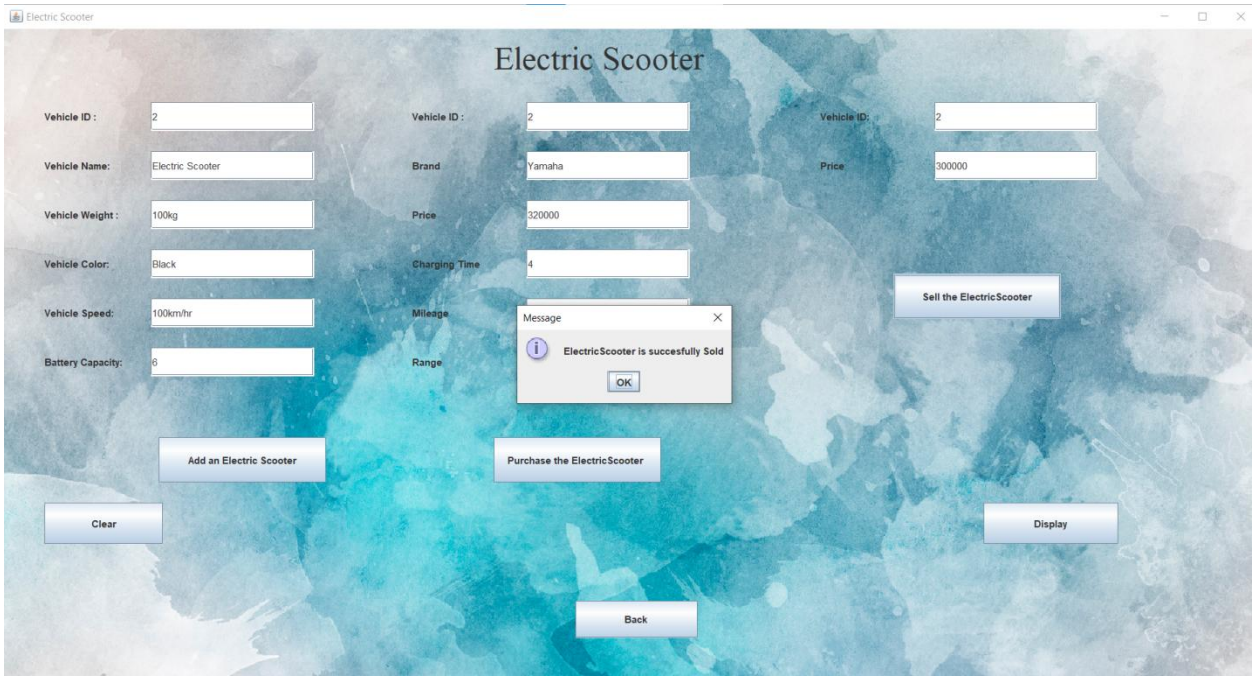

Figure 31 : Selling Electric Scooter before selling twice

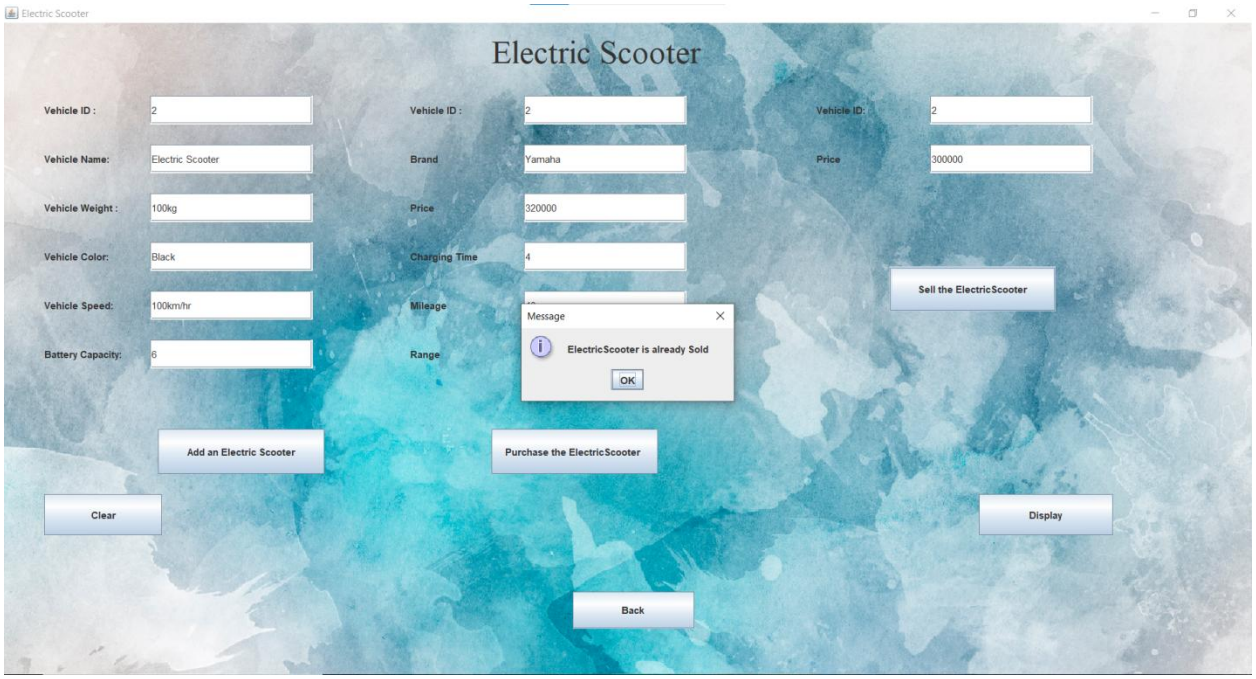

Figure 32 : Selling Electric Scooter again

e.) To Check wheather all the fields are filled or not

Objectives	Check wheather all the fields are filled or not
Actions	Only some of the fields were filled
Expectations	Dialog box with ”Please fill all the fields” Should appear
Actual Output	Dialog box with ”Please fill all the fields” was appeared
Conclusion	Test was Successful.

Table 12 : Test 3) e

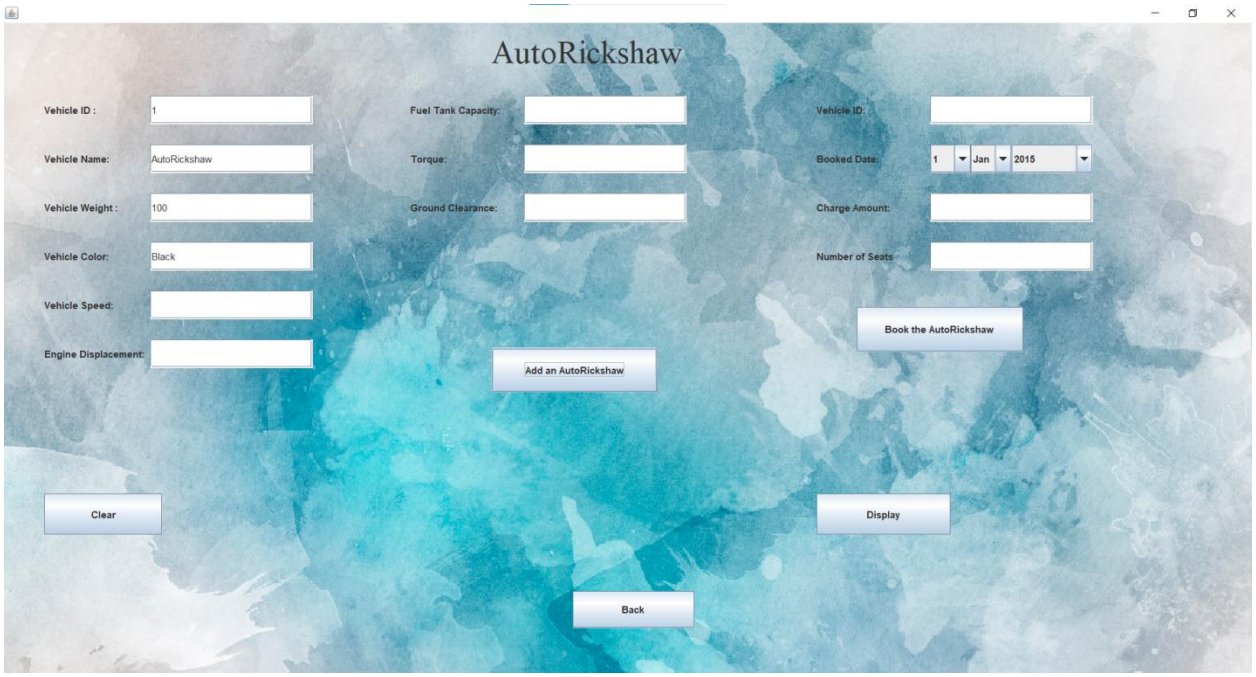

Figure 33 : Filling Some of the fields

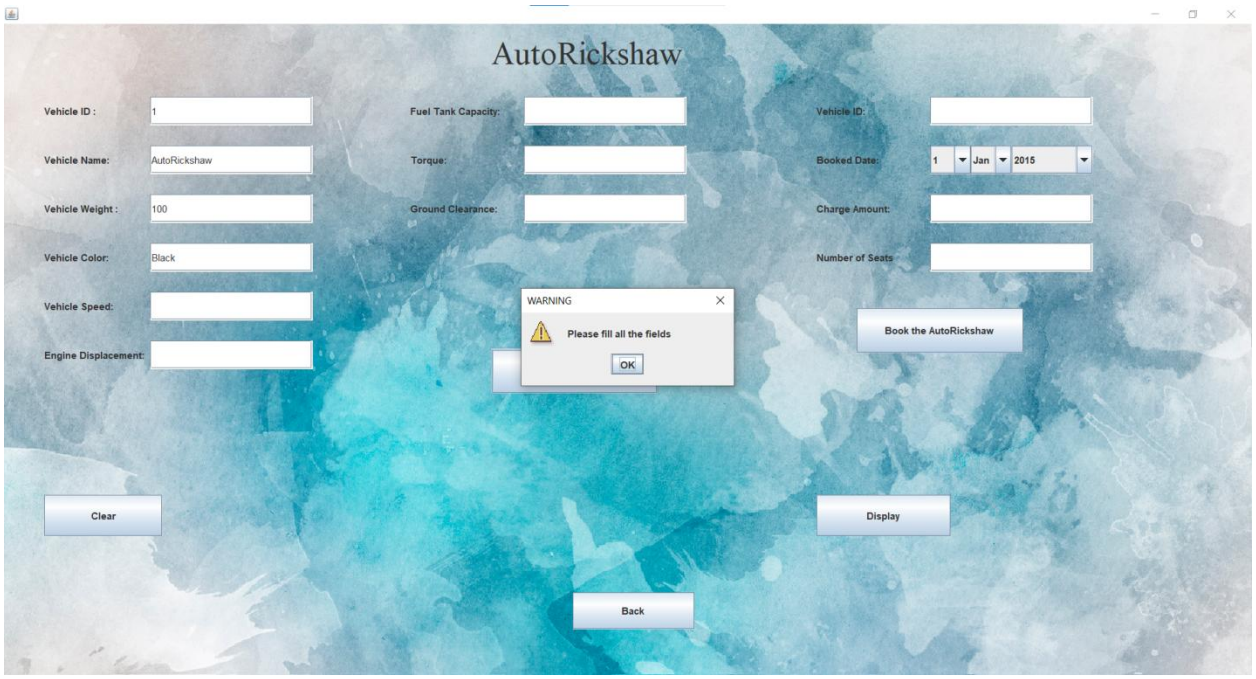

Figure 34 : Dialog box after clicking ADD to Autorickshaw

6. Error Detection and Correction

6.1 Syntax Error

Syntax errors are those errors, which occur when the programmer uses the wrong syntax. A compiler can detect most of these errors. Some of the common errors are enlisted below:

- Incorrect Capitalization of codes

- Splitting a line of code

- Absence of Parenthesis, braces or brackets

- Forget to import classes

- Mistyping a block of code

- Missing return statement

While compiling my code I found this error where one of the paranthesis was missing.

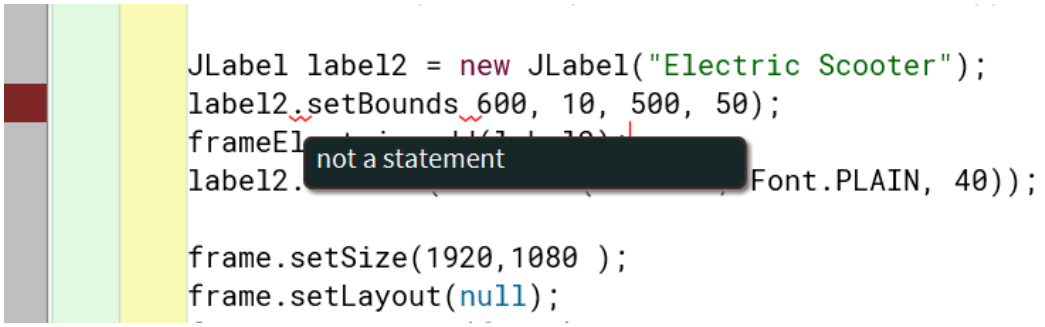

Figure 35 : Syntax Error

Error was corrected by Adding (after setBounds.

```
JLabel label2 = new JLabel("Electric Scooter");
label2.setBounds (600, 10, 500, 50);
frameElectric.add(label2);
label2.setFont(new Font("serif", Font.PLAIN, 40));

frame.setSize(1920,1080 );
frame.setLayout(null);
frame.setVisible(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Figure 36 : Correction of Syntax Error

6.2 Logical Error

The error, which changes the output of the program by messing up the logical equation, are called logical errors. These logical errors could simply change the output of the program just simply by using > instead of < symbol. So, While doing logical operation, negligence should be prohibited. This error won't be shown in the code but will hamper the output.

```
public int getVehicleId(){
    String vehicleIdText = fieldID.getText().trim();
    int vehicleID = INVALID;
    try{
        vehicleID = Integer.parseInt(vehicleIdText);
        if(vehicleID >=0){
            vehicleID = INVALID;
        }
    }
    catch(NumberFormatException e){
    }
    return vehicleID;
}
```

Figure 37 : Logical Error

```
public int getVehicleId(){
    String vehicleIdText = fieldID.getText().trim();
    int vehicleID = INVALID;
    try{
        vehicleID = Integer.parseInt(vehicleIdText);
        if(vehicleID <=0){
            vehicleID = INVALID;
        }
    }
    catch(NumberFormatException e){
    }
    return vehicleID;
}
```

Figure 38 : Correction of Logical Error

6.3 Semantic Error

Due to incompatible data types, a semantic error occurs. This problem is caused by the fact that I typed String instead of Int while checking the uniqueId of the vehicleId.

```
public boolean checkIfUnique(String vehicleId)
{
    boolean isUnique = true;
    for(Vehicle vl: vehicleList)
    {
        if(vl.getVehicleId() == vehicleId)
        {
            JOptionPane.showMessageDial                    ique","ERROR",
                JOptionPane.ERROR_MESSA
            isUnique = false;
            break;
        }
    }
    return isUnique;
}
```

bad operand types for binary operator '=='
first type: int
second type: java.lang.String

Figure 39 : Semantic Error

```
public boolean checkIfUnique(int vehicleId)
{
    boolean isUnique = true;
    for(Vehicle vl: vehicleList)
    {
        if(vl.getVehicleId() == vehicleId)
        {
            JOptionPane.showMessageDialog(frame,"Vehicle Id must be unique","ERROR",
                JOptionPane.ERROR_MESSAGE);
            isUnique = false;
            break;
        }
    }
    return isUnique;
}
```

Figure 40 : Correction of Semantic Error

7. Conclusion

This Coursework is all about developing a GUI for Adding, booking, purchasing, selling system of a Vehicle in a single class named TransportationGUI. This Coursework represents a GUI which will allow the users to Add both AutoRickshaw and ElectricScooter. The simple text field were made for the user to add details according to their labels.

This coursework helped me to create buttons, text areas and helped in increasing my understanding of Java's graphical user interface (GUI) as well as its use in software development. The whole program of this project was done under java programming language and a key console to code the program was Bluej. Bluej is an integrated development environment for java basically designed for beginners.

I initially found it tough to complete this assignment while coding, many problems occurred that took a long time to overcome, something I had never experienced before, but after asking our module teachers questions about it, I got a better understanding of the coursework module. They helped me in many ways, such as explaining the question and helping me through coding. The curriculum was not difficult, but it was lengthy.Despite the length of the coursework, I was able to complete it with the help of slides and lectures.

8. References

Deep, June-20,2012. *Father of java.* [Online] Available at: https://freefeast.info/personality-motivation/famous_it_personalities/history-of-james-gosling-father-of-java-java-creator/ [Accessed 15 Jul 2022].

w3schools, 1999-2022. *Java Introduction.* [Online] Available at: https://www.w3schools.com/java/java_intro.asp [Accessed 15 Jul 2022].

9. Appendix

```
/**
 *TransportGUI is the main class which holds three GUI JFrames
 * @author (Dipesh Bharati)
 * @ID (np05cp4s220022)
 */
//Creation of GUI for AutoRickshaw
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.io.IOException;
import javax.swing.JFrame;
import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JTextField;
import javax.swing.JOptionPane;
import java.awt.Font;
import javax.swing.JComboBox;
import java.util.ArrayList;
```

```
import java.awt.Color;


public class TransportGUI {
    ArrayList<Vehicle> vehicleList = new ArrayList<>();
    private JFrame homeFrame,frame,frameElectric;
    private JTextField fieldID,fieldName,fieldWeight,fieldColor,fieldSpeed,fieldEngineDisplacement,fieldTorque,fieldTankCapacity,fieldGroundClearance;
    private JTextField fieldiD,fieldAmount,fieldSeats;
    private JTextField fieldElectricId,fieldElectricName,fieldElectricWeight,fieldElectricColor,fieldElectricSpeed,fieldElectricBattery;
    private JTextField fieldPurElectricId,fieldBrand,fieldPrice,fieldChargingTime,fieldMileage,fieldRange;
    private JTextField fieldSellElectricID,fieldSellPrice;
    private JButton BTNBK;
    private JButton BTNDS;
    private JButton BTNCLR;
    private JButton BTNADD;
    private JButton back;
    private JButton electricScooterBtn,autoRickshawBtn,BTNelectricDisplay;
    private JButton BTNADDScooter,BTNPurchase,BTNSell,BTNelectricCLR;
    private JComboBox<String> d,m,y;
    final static int INVALID = -2;
```

```
    //Creating GUI
    public TransportGUI(){
        homeFrame = new JFrame("Home");
        try {
            homeFrame.setContentPane(new JLabel(new ImageIcon(ImageIO.read(new File("darkwood.jpg")))));

        } catch (IOException e) {
            System.out.println("image doesn't exist");
        }

        frame = new JFrame();
        try {
            frame.setContentPane(new JLabel(new ImageIcon(ImageIO.read(new File("light.jpg")))));

        } catch (IOException e) {
            System.out.println("image doesn't exist");
        }

        frameElectric = new JFrame("Electric Scooter");
        try {
```

```
        frameElectric.setContentPane(new JLabel(new ImageIcon(ImageIO.read(new File("light.jpg"))))));


    } catch (IOException e) {
        System.out.println("image doesn't exist");
    }


    autoRickshawBtn = new JButton("AutoRickShaw");
    autoRickshawBtn.setBounds(250,150,200,50);
    homeFrame.add(autoRickshawBtn);
    autoRickshawBtn.addActionListener(new ActionListener()
      {
        public void actionPerformed(ActionEvent e)
        {

            frame.setVisible(true);
            homeFrame.setVisible(false);
        }
      });


    electricScooterBtn = new JButton("Electric Scooter");
    electricScooterBtn.setBounds(250,250,200,50);
    homeFrame.add(electricScooterBtn);
```

```
electricScooterBtn.addActionListener(new ActionListener(){
        public void actionPerformed(ActionEvent e)
        {
            frameElectric.setVisible(true);
            homeFrame.setVisible(false);
        }

    });

JLabel title = new JLabel("Welcome");
title.setBounds(250, 10, 300, 50);
homeFrame.add(title);
title.setFont(new Font("serif", Font.PLAIN, 50));
title.setForeground(Color.white);

JLabel label = new JLabel("AutoRickshaw");
label.setBounds(600, 10, 500, 50);
frame.add(label);
label.setFont(new Font("serif", Font.PLAIN, 40));

JLabel label2 = new JLabel("Electric Scooter");
label2.setBounds (600, 10, 500, 50);
frameElectric.add(label2);
```

```
label2.setFont(new Font("serif", Font.PLAIN, 40));


frame.setSize(1920,1080 );
frame.setLayout(null);
frame.setVisible(false);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


frameElectric.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frameElectric.setLayout(null);
frameElectric.setSize(1920,1080 );


homeFrame.setLayout(null);
homeFrame.setSize(700,500);
homeFrame.setVisible(true);
homeFrame.setLocation(400, 150);


//GUI for AddAutoRickshaw


JLabel vehicleID = new JLabel("Vehicle ID : ");
vehicleID.setBounds(50, 90, 95, 35);
frame.add(vehicleID);


fieldID = new JTextField();
```

```
fieldID.setBounds(180, 90, 200, 35);
frame.add(fieldID);

JLabel vehicleName = new JLabel("Vehicle Name:");
vehicleName.setBounds(50, 150, 105, 35);
frame.add(vehicleName);

fieldName = new JTextField();
fieldName.setBounds(180, 150, 200, 35);
frame.add(fieldName);

JLabel vehicleWeight = new JLabel("Vehicle Weight : ");
vehicleWeight.setBounds(50, 210, 105, 35);
frame.add(vehicleWeight);

fieldWeight = new JTextField();
fieldWeight.setBounds(180, 210, 200, 35);
frame.add(fieldWeight);

JLabel vehicleColor = new JLabel("Vehicle Color: ");
vehicleColor.setBounds(50, 270, 105, 35);
frame.add(vehicleColor);
```

```
fieldColor = new JTextField();
fieldColor.setBounds(180, 270, 200, 35);
frame.add(fieldColor);

JLabel vehicleSpeed = new JLabel("Vehicle Speed:");
vehicleSpeed.setBounds(50, 330, 105, 35);
frame.add(vehicleSpeed);

fieldSpeed = new JTextField();
fieldSpeed.setBounds(180, 330, 200, 35);
frame.add(fieldSpeed);

JLabel engineDisplacement = new JLabel("Engine Displacement:");
engineDisplacement.setBounds(50, 390, 128, 35);
frame.add(engineDisplacement);

fieldEngineDisplacement = new JTextField();
fieldEngineDisplacement.setBounds(180, 390, 200, 35);
frame.add(fieldEngineDisplacement);

JLabel torque = new JLabel("Torque:");
torque.setBounds(500, 150, 105, 35);
frame.add(torque);
```

```
fieldTorque = new JTextField();
fieldTorque.setBounds(640, 150, 200, 35);
frame.add(fieldTorque);

JLabel fuelTankCapacity = new JLabel("Fuel Tank Capacity:");
fuelTankCapacity.setBounds(500, 90, 135, 35);
frame.add(fuelTankCapacity);

fieldTankCapacity = new JTextField();
fieldTankCapacity.setBounds(640, 90, 200, 35);
frame.add(fieldTankCapacity);

JLabel groundClearance = new JLabel("Ground Clearance:");
groundClearance.setBounds(500, 210, 135, 35);
frame.add(groundClearance);

fieldGroundClearance = new JTextField();
fieldGroundClearance.setBounds(640,210, 200, 35);
frame.add(fieldGroundClearance);

//GUI for bookAutoRickshaw
JLabel iD = new JLabel("Vehicle ID:");
```

```
        iD.setBounds(1000, 90, 135, 35);
        frame.add(iD);

        fieldiD = new JTextField();
        fieldiD.setBounds(1140, 90, 200, 35);
        frame.add(fieldiD);

        JLabel bookedDate = new JLabel("Booked Date:");
        String
day[]={"1","2","3","4","5","6","7","8","9","10","11","12","13","14","15","16","17","18","19","2
0","21","22","23","24","25","26","27","28","29","30","31"};
        String
month[]={"Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sept","Oct","Nov","Dec"};
        String year[]={"2015","2016","2017","2018","2019","2020","2021","2022"};
        d = new JComboBox<>(day);
        m = new JComboBox<>(month);
        y = new JComboBox<>(year);
        bookedDate.setBounds(1000, 150, 135, 35);
        d.setBounds(1140,150,50,35);
        m.setBounds(1190,150,50,35);
        y.setBounds(1240,150,100,35);
        frame.add(bookedDate);
        frame.add(d);
```

```
frame.add(m);

frame.add(y);

JLabel chargeAmount = new JLabel("Charge Amount:");

chargeAmount.setBounds(1000, 210, 135, 35);

frame.add(chargeAmount);


fieldAmount = new JTextField();

fieldAmount.setBounds(1140, 210, 200, 35);

frame.add(fieldAmount);


JLabel numberOfSeats = new JLabel("Number of Seats");

numberOfSeats.setBounds(1000, 270, 135, 35);

frame.add(numberOfSeats);


fieldSeats = new JTextField();

fieldSeats.setBounds(1140, 270, 200, 35);

frame.add(fieldSeats);


//GUI for Electric Scooter

JLabel electricVehicleID = new JLabel("Vehicle ID : ");

electricVehicleID.setBounds(50, 90, 105, 35);

frameElectric.add(electricVehicleID);
```

```
fieldElectricId = new JTextField();
fieldElectricId.setBounds(180, 90, 200, 35);
frameElectric.add(fieldElectricId);

JLabel electricVehicleName = new JLabel("Vehicle Name:");
electricVehicleName.setBounds(50, 150, 95, 35);
frameElectric.add(electricVehicleName);

fieldElectricName = new JTextField();
fieldElectricName.setBounds(180, 150, 200, 35);
frameElectric.add(fieldElectricName);

JLabel electricVehicleWeight = new JLabel("Vehicle Weight : ");
electricVehicleWeight.setBounds(50, 210, 105, 35);
frameElectric.add(electricVehicleWeight);

fieldElectricWeight = new JTextField();
fieldElectricWeight.setBounds(180, 210, 200, 35);
frameElectric.add(fieldElectricWeight);

JLabel electricVehicleColor = new JLabel("Vehicle Color: ");
electricVehicleColor.setBounds(50, 270, 105, 35);
frameElectric.add(electricVehicleColor);
```

```
fieldElectricColor = new JTextField();
fieldElectricColor.setBounds(180, 270, 200, 35);
frameElectric.add(fieldElectricColor);

JLabel electricVehicleSpeed = new JLabel("Vehicle Speed:");
electricVehicleSpeed.setBounds(50, 330, 105, 35);
frameElectric.add(electricVehicleSpeed);

fieldElectricSpeed = new JTextField();
fieldElectricSpeed.setBounds(180, 330, 200, 35);
frameElectric.add(fieldElectricSpeed);

JLabel electricBatteryCapacity = new JLabel("Battery Capacity:");
electricBatteryCapacity.setBounds(50, 390, 128, 35);
frameElectric.add(electricBatteryCapacity);

fieldElectricBattery = new JTextField();
fieldElectricBattery.setBounds(180, 390, 200, 35);
frameElectric.add(fieldElectricBattery);

///GUI for Purchasing Scooter
```

```
JLabel purElectricScooterID = new JLabel("Vehicle ID : ");
purElectricScooterID.setBounds(500, 90, 135, 35);
frameElectric.add(purElectricScooterID);

fieldPurElectricId = new JTextField();
fieldPurElectricId.setBounds(640, 90, 200, 35);
frameElectric.add(fieldPurElectricId);

JLabel brand = new JLabel("Brand");
brand.setBounds(500, 150, 105, 35);
frameElectric.add(brand);

fieldBrand = new JTextField();
fieldBrand.setBounds(640, 150, 200, 35);
frameElectric.add(fieldBrand);

JLabel price = new JLabel("Price");
price.setBounds(500, 210, 135, 35);
frameElectric.add(price);

fieldPrice = new JTextField();
fieldPrice.setBounds(640,210, 200, 35);
frameElectric.add(fieldPrice);
```

```
JLabel chargingTime = new JLabel("Charging Time");
chargingTime.setBounds(500, 270, 105, 35);
frameElectric.add(chargingTime);

fieldChargingTime = new JTextField();
fieldChargingTime.setBounds(640, 270, 200, 35);
frameElectric.add(fieldChargingTime);

JLabel mileage = new JLabel("Mileage");
mileage.setBounds(500, 330, 105, 35);
frameElectric.add(mileage);

fieldMileage = new JTextField();
fieldMileage.setBounds(640, 330, 200, 35);
frameElectric.add(fieldMileage);

JLabel range = new JLabel("Range");
range.setBounds(500, 390, 128, 35);
frameElectric.add(range);

fieldRange = new JTextField();
fieldRange.setBounds(640, 390, 200, 35);
```

```
frameElectric.add(fieldRange);

///GUI for Selling Scooter

JLabel sellElectricID = new JLabel("Vehicle ID:");

sellElectricID.setBounds(1000, 90, 135, 35);

frameElectric.add(sellElectricID);


fieldSellElectricID = new JTextField();

fieldSellElectricID.setBounds(1140, 90, 200, 35);

frameElectric.add(fieldSellElectricID);


JLabel sellPrice = new JLabel("Price");

sellPrice.setBounds(1000, 150, 135, 35);

frameElectric.add(sellPrice);


fieldSellPrice = new JTextField();

fieldSellPrice.setBounds(1140, 150, 200, 35);

frameElectric.add(fieldSellPrice);


//GUI for AutoRickshaw button


back = new JButton("Back ");

back.setBounds(700,700,150,45);

frame.add(back);
```

```
back.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         homeFrame.setVisible(true);
         frame.setVisible(false);
      }
   });

BTNADD = new JButton("Add an AutoRickshaw");
BTNADD.setBounds(600, 400, 205, 55);
BTNADD.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         addAutoRickshaw();

      }
   });

frame.add(BTNADD);

BTNBK = new JButton("Book the AutoRickshaw ");
BTNBK.setBounds(1050, 350, 205, 55);
BTNBK.addActionListener(new ActionListener()
```

```
    {
        public void actionPerformed(ActionEvent e){
            bookAutoRickshaw();

        }
    });

frame.add(BTNBK);

BTNCLR = new JButton("Clear");
BTNCLR.setBounds(50, 580, 145, 50);

frame.add(BTNCLR);
BTNCLR.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e){
            clearAutoRickshaw();

        }
    });

BTNDS = new JButton("Display");
BTNDS.setBounds(1000, 580, 165, 50);
```

```
BTNDS.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         displayAutoRickshaw();


      }
   });
frame.add(BTNDS);
///Buttons for Electric Scooter


back = new JButton("Back ");
back.setBounds(700,700,150,45);
frameElectric.add(back);
back.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         homeFrame.setVisible(true);
         frameElectric.setVisible(false);
      }
   });


BTNADDScooter = new JButton("Add an Electric Scooter");
BTNADDScooter.setBounds(190, 500, 205, 55);
```

```
BTNADDScooter.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         addElectricScooter();
      }
   });

frameElectric.add(BTNADDScooter);

BTNPurchase = new JButton("Purchase the ElectricScooter ");
BTNPurchase.setBounds(600, 500, 205, 55);
BTNPurchase.addActionListener(new ActionListener()
   {
      public void actionPerformed(ActionEvent e){
         purElectricScooter();
      }
   });

frameElectric.add(BTNPurchase);

BTNSell = new JButton("Sell the ElectricScooter");
BTNSell.setBounds(1090, 300, 205, 55);
BTNSell.addActionListener(new ActionListener()
```

```
    {
        public void actionPerformed(ActionEvent e){
            sellElectricScooter();
        }
    });

frameElectric.add(BTNSell);

BTNelectricDisplay = new JButton("Display");
BTNelectricDisplay.setBounds(1200, 580, 165, 50);
BTNelectricDisplay.addActionListener(new ActionListener()
    {
        public void actionPerformed(ActionEvent e){
            displayElectricScooter();

        }
    });
frameElectric.add(BTNelectricDisplay);

BTNelectricCLR = new JButton("Clear");
BTNelectricCLR.setBounds(50, 580, 145, 50);

frameElectric.add(BTNelectricCLR);
```

```
    BTNelectricCLR.addActionListener(new ActionListener()
        {
            public void actionPerformed(ActionEvent e){
                clearElectricScooter();


            }
        });
}

///Get for adding AutoRickshaw

public String getVehicleName(){
    return fieldName.getText().trim();
}

public String getVehicleWeight()
{
    return fieldWeight.getText().trim();
}

public int getVehicleId(){
    String vehicleIdText = fieldID.getText().trim();
    int vehicleID = INVALID;
```

```
        try{
            vehicleID = Integer.parseInt(vehicleIdText);
            if(vehicleID <=0){
                vehicleID = INVALID;
            }
        }
        catch(NumberFormatException e){
        }
        return vehicleID;
    }

    public String getVehicleColor(){
        return fieldColor.getText().trim();
    }

    public String getVehicleSpeed(){
        return fieldSpeed.getText().trim();
    }

    public int getVehicleDisplacement(){
        String vehicleDisplacementText = fieldEngineDisplacement.getText().trim();
        int vehicleDisplacement = INVALID;
        try{
```

```
            vehicleDisplacement = Integer.parseInt(vehicleDisplacementText);
            if(vehicleDisplacement <=0){
                vehicleDisplacement = INVALID;
            }
        }
        catch(NumberFormatException e){

        }
        return vehicleDisplacement;
    }

    public int getFuelTankCapacity()
    {
        String fuelTankText = fieldTankCapacity.getText().trim();
        int fuelTank = INVALID;
        try {
            fuelTank = Integer.parseInt(fuelTankText);
            if(fuelTank <=0) {
                fuelTank = INVALID;
            }
        }
        catch(NumberFormatException e) {
        }
```

```
    return fuelTank;
}

public String getTorque()
{
    return fieldTorque.getText().trim();
}

public String getClearance()
{
    return fieldGroundClearance.getText().trim();
}

///Get for Booking AutoRickshaw
public int getBookVehicleId(){
    String vehicleIdText = fieldiD.getText().trim();
    int vehicleID = INVALID;
    try{
        vehicleID = Integer.parseInt(vehicleIdText);
        if(vehicleID <=0){
            vehicleID = INVALID;
        }
    }
```

```
        catch(NumberFormatException e){


        }
        return vehicleID;
    }

    public String getBooked(){
        String day = d.getSelectedItem().toString();
        String month = m.getSelectedItem().toString();
        String year = y.getSelectedItem().toString();
        return day+month+year;
    }

    public int getChargeAmount(){
        String chargeAmountText = fieldAmount.getText().trim();
        int chargeAmount = INVALID;
        try{
            chargeAmount = Integer.parseInt(chargeAmountText);
            if(chargeAmount <= 0) {
                chargeAmount = INVALID;
            }
        }
        catch(NumberFormatException e){
```

```
        }
        return chargeAmount;
    }

    public int getNumberOfSeats(){
        String numberOfSeatsText = fieldSeats.getText().trim();
        int numberOfSeats = INVALID;
        try{
            numberOfSeats = Integer.parseInt(numberOfSeatsText);
            if(numberOfSeats <=0){
                numberOfSeats = INVALID;
            }
        }
        catch(NumberFormatException e){
        }
        return numberOfSeats;
    }

    ///Get for adding Electric Scooter
    public int getAddElectricVehicleId(){
        String addElectricVehicleIdText = fieldElectricId.getText().trim();
        int addElectricVehicleID = INVALID;
```

```
        try{
            addElectricVehicleID = Integer.parseInt(addElectricVehicleIdText);
            if(addElectricVehicleID <=0){
                addElectricVehicleID = INVALID;
            }
        }
        catch(NumberFormatException e){

        }
        return addElectricVehicleID;
    }

    public String getElectricVehicleName(){
        return fieldElectricName.getText().trim();
    }

    public String getElectricVehicleWeight()
    {
        return fieldElectricWeight.getText().trim();
    }

    public String getElectricVehicleColor(){
        return fieldElectricColor.getText().trim();
```

```
    }

    public String getElectricVehicleSpeed(){
        return fieldElectricSpeed.getText().trim();
    }

    public int getElectricBatteryCapacity(){
        String addBatteryCapacityText = fieldElectricBattery.getText().trim();
        int addBatteryCapacity = INVALID;
        try{
            addBatteryCapacity = Integer.parseInt(addBatteryCapacityText);
            if(addBatteryCapacity <=0){
                addBatteryCapacity = INVALID;
            }
        }
        catch(NumberFormatException e){

        }
        return addBatteryCapacity;
    }

    public String getBrand(){
        return fieldBrand.getText().trim();
```

```
    }

    ///Get for Purchase
    public int getPurElectricVehicleId(){
        String purElectricVehicleIdText = fieldPurElectricId.getText().trim();
        int purElectricVehicleID = INVALID;
        try{
            purElectricVehicleID = Integer.parseInt(purElectricVehicleIdText);
            if(purElectricVehicleID <=0){
                purElectricVehicleID = INVALID;
            }
        }
        catch(NumberFormatException e){

        }
        return purElectricVehicleID;
    }

    public int getPrice(){
        String priceText = fieldPrice.getText().trim();
        int price = INVALID;
        try{
            price = Integer.parseInt(priceText);
```

```
            if(price <=0){
                price = INVALID;
            }
        }
        catch(NumberFormatException e){
        }
        return price;
    }

    public String getChargingTime(){
        return fieldChargingTime.getText().trim();
    }

    public String getMileage(){
        return fieldMileage.getText().trim();
    }

    public int getRange(){
        String rangeText = fieldRange.getText().trim();
        int range = INVALID;
        try{
            range = Integer.parseInt(rangeText);
            if(range <=0){
```

```
            range = INVALID;
        }
    }
    catch(NumberFormatException e){


    }
    return range;
}
///Get for Selling
public int getSellElectricVehicleId(){
    String sellElectricVehicleIdText = fieldSellElectricID.getText().trim();
    int sellElectricVehicleID = INVALID;
    try{
        sellElectricVehicleID = Integer.parseInt(sellElectricVehicleIdText);
        if(sellElectricVehicleID <=0){
            sellElectricVehicleID = INVALID;
        }
    }
    catch(NumberFormatException e){


    }
    return sellElectricVehicleID;
}
```

```
public int getSellPrice(){
    String priceText = fieldSellPrice.getText().trim();
    int price = INVALID;
    try{
        price = Integer.parseInt(priceText);
        if(price <=0){
            price = INVALID;
        }
    }
    catch(NumberFormatException e){


    }
    return price;
}

public boolean checkIfUnique(int vehicleId)
{
    boolean isUnique = true;
    for(Vehicle vl: vehicleList)
    {
        if(vl.getVehicleId() == vehicleId)
        {
```

```
            JOptionPane.showMessageDialog(frame,"Vehicle Id must be unique","ERROR",
                JOptionPane.ERROR_MESSAGE);
            isUnique = false;
            break;
        }
    }
    return isUnique;
}
//Method of Add AutoRickshaw
public void addAutoRickshaw()
{
    if (getVehicleName().isEmpty()||getVehicleWeight().isEmpty()||getVehicleColor().isEmpty()||getVehicleSpeed().isEmpty()||getTorque().isEmpty()||getClearance().isEmpty()){
        JOptionPane.showMessageDialog(frame,"Please fill all the fields","WARNING",
            JOptionPane.WARNING_MESSAGE);
        return;
    }


if(getVehicleId()==INVALID||getVehicleDisplacement()==INVALID||getFuelTankCapacity()==INVALID){
```

```
            JOptionPane.showMessageDialog(frame, "Please enter integer value in VehicleId\nor EngineDisplacement or Fuel Tank Capacity","Error",
                JOptionPane.ERROR_MESSAGE);
            return;
        }

        if (checkIfUnique(getVehicleId()))
        {
            vehicleList.add(new AutoRickshaw(getVehicleId(), getVehicleName(), getVehicleWeight(), getVehicleColor(), getVehicleSpeed(), getVehicleDisplacement(), getTorque(), getFuelTankCapacity(), getClearance()));
            JOptionPane.showMessageDialog(frame,"AutoRickshaw is successfully added.","Message",
                JOptionPane.INFORMATION_MESSAGE);
        }
    }

    //method for add Electric scooter
    public void addElectricScooter()
    {
        if (getElectricVehicleName().isEmpty()||getElectricVehicleSpeed().isEmpty()||getElectricVehicleColor().isEmpty()||getElectricVehicleWeight().isEmpty()){
            JOptionPane.showMessageDialog(frame,"Please fill all the fields","WARNING",JOptionPane.WARNING_MESSAGE);
```

```
            return;
        }

        if(getAddElectricVehicleId()==INVALID||getElectricBatteryCapacity()==INVALID){
            JOptionPane.showMessageDialog(frame, "Please enter integer value in VehicleId\nor Electric battery Capacity","Error",
                JOptionPane.ERROR_MESSAGE);
            return;
        }

        if (checkIfUnique(getAddElectricVehicleId()))
        {
            vehicleList.add(new ElectricScooter(getAddElectricVehicleId(), getElectricVehicleName(), getElectricVehicleWeight(), getElectricVehicleColor(), getElectricVehicleSpeed(), getElectricBatteryCapacity()));
            JOptionPane.showMessageDialog(frame,"Electric Scooter is successfully added. ","message",JOptionPane.INFORMATION_MESSAGE);
        }
    }
    ///method for booking autoRickshaw

    public void bookAutoRickshaw()
    {
```

```
if (getBookVehicleId()==INVALID || getChargeAmount()==INVALID || getNumberOfSeats()==INVALID){
    JOptionPane.showMessageDialog(frame, "Please enter integer value in VehicleID or ChargeAmount or Number of seats","Error",
        JOptionPane.ERROR_MESSAGE);
    return;
}

for(Vehicle autoVehicle:vehicleList){
    if(getBookVehicleId()==autoVehicle.getVehicleId()){
        if(autoVehicle instanceof AutoRickshaw){
            AutoRickshaw obj = (AutoRickshaw)autoVehicle;
            if (obj.getIsBooked()) {
                JOptionPane.showMessageDialog(frame, "AutoRickshaw is already booked","Warning",
                    JOptionPane.WARNING_MESSAGE);
                return;
            }
            obj.book(getBooked(),getChargeAmount(), getNumberOfSeats());
            JOptionPane.showMessageDialog(frame, "AutoRickshaw is succesfully Booked","Message",
                JOptionPane.INFORMATION_MESSAGE);
            return;
        }
```

```
        }


    }


    if (!vehicleList.contains(getBookVehicleId())){
        JOptionPane.showMessageDialog(frame, "Vehicle is not in list","Warning",
            JOptionPane.WARNING_MESSAGE);
    }


}
//method for purchasing Electric Scooter
public void purElectricScooter()
{
    if (getPurElectricVehicleId()==INVALID || getPrice()==INVALID ||
getRange()==INVALID){
        JOptionPane.showMessageDialog(frame, "Please enter integer value in
VehicleID or Price or Range","Error",
            JOptionPane.ERROR_MESSAGE);
        return;
    }


    for(Vehicle E_Vehicle:vehicleList){
        if(getPurElectricVehicleId()==E_Vehicle.getVehicleId()){
```

```
            if(E_Vehicle instanceof ElectricScooter){
                ElectricScooter obj = (ElectricScooter)E_Vehicle;
                if (obj.getHasPurchased()) {
                    JOptionPane.showMessageDialog(frame, "ElectricScooter is already Purchased","Message",
                        JOptionPane.WARNING_MESSAGE);
                    return;
                }
                obj.purchase(getBrand(), getPrice(), getChargingTime(), getMileage(), getRange());
                JOptionPane.showMessageDialog(frame, "ElectricScooter is succesfully Purchased","Message",
                    JOptionPane.INFORMATION_MESSAGE);
                return;
            }
        }

    }

    if (!vehicleList.contains(getPurElectricVehicleId())) {
        JOptionPane.showMessageDialog(frameElectric, "Vehicle is not in list","Warning",
            JOptionPane.WARNING_MESSAGE);
```

```
        }
    }
    ///Method for selling Electric Scooter


    public void sellElectricScooter()
    {
        if (getSellElectricVehicleId()==INVALID || getSellPrice()==INVALID){
            JOptionPane.showMessageDialog(frame, "Please enter integer value in VehicleID or Price","Error",
                JOptionPane.ERROR_MESSAGE);
            return;
        }


        for(Vehicle E_Vehicle:vehicleList){
            if(getSellElectricVehicleId()==E_Vehicle.getVehicleId()){
                if(E_Vehicle instanceof ElectricScooter){
                    ElectricScooter obj = (ElectricScooter)E_Vehicle;
                    if (obj.getHasSold()) {
                        JOptionPane.showMessageDialog(frame, "ElectricScooter is already Sold","Message",
                            JOptionPane.INFORMATION_MESSAGE);
                        return;
                    }
```

```
                obj.sell(getSellPrice());
                JOptionPane.showMessageDialog(frame, "ElectricScooter is succesfully Sold","Message",
                    JOptionPane.INFORMATION_MESSAGE);
                return;
            }
        }

    }

    if (!vehicleList.contains(getSellElectricVehicleId())) {
        JOptionPane.showMessageDialog(frameElectric, "Vehicle is not in list","Warning",
            JOptionPane.WARNING_MESSAGE);

    }
  }
  ///method for displaying autorickshaw
  public void displayAutoRickshaw(){
    if(vehicleList.isEmpty()){
        JOptionPane.showMessageDialog(frame, "There is no vehicle available Right now","Message",
            JOptionPane.INFORMATION_MESSAGE);
```

```
        }


        for(Vehicle autoVehicle:vehicleList){
            if(autoVehicle instanceof AutoRickshaw){
                AutoRickshaw obj = (AutoRickshaw)autoVehicle;
                obj.display();
                JOptionPane.showMessageDialog(frame, "It is printed in Terminal","Message",
                    JOptionPane.INFORMATION_MESSAGE);
                    return;
            }
        }


    }
    ///method for displaying electric scooter
    public void displayElectricScooter(){
        if(vehicleList.isEmpty()){
            JOptionPane.showMessageDialog(frame, "There is no Scooter available Right now","Message",
                JOptionPane.INFORMATION_MESSAGE);
        }


        for(Vehicle E_Vehicle:vehicleList){
```

```
            if(E_Vehicle instanceof ElectricScooter){
                ElectricScooter obj = (ElectricScooter)E_Vehicle;
                obj.display();
                JOptionPane.showMessageDialog(frame, "It is printed in Terminal","Message",
                    JOptionPane.INFORMATION_MESSAGE);
                    return;
            }
        }

    }

    //Function for clear AutoRickshaw
    public void clearAutoRickshaw(){
        fieldID.setText(null);
        fieldName.setText(null);
        fieldWeight.setText(null);
        fieldColor.setText(null);
        fieldSpeed.setText(null);
        fieldEngineDisplacement.setText(null);
        fieldTorque.setText(null);
        fieldTankCapacity.setText(null);
        fieldGroundClearance.setText(null);
```

```
        fieldiD.setText("");
        y.setSelectedIndex(0);
        m.setSelectedIndex(0);
        d.setSelectedIndex(0);
        fieldSeats.setText("");
        fieldAmount.setText("");
    }
    ///Function for Clear Electric Scooter
    public void clearElectricScooter()
    {
        fieldElectricName.setText("");
        fieldElectricId.setText("");
        fieldElectricWeight.setText("");
        fieldElectricColor.setText("");
        fieldElectricSpeed.setText("");
        fieldElectricBattery.setText("");
        fieldPurElectricId.setText("");
        fieldBrand.setText("");
        fieldPrice.setText("");
        fieldChargingTime.setText("");
        fieldMileage.setText("");
        fieldRange.setText("");
        fieldSellElectricID.setText("");
```

```
        fieldSellPrice.setText("");
    }

    // Main method
    public static void main (String[] args){
        new TransportGUI();
    }
}
```