



UMS
UNIVERSITI MALAYSIA SABAH

KK14203

PENGATURCARAAN BERORIENTASIKAN OBJEK

SECTION 1

INDIVIDUAL TASK

PROJECT 2 (MY CAR RENTAL)

DR. SAMRY @ MOHD. SHAMRIE SAININ

STUDENT NAME

FRANCES LYNNE WILLIAM

MATRIC NUM.

BI19160328

1.0 JAVA CODE

rental.java

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Rental{

    private static final int WIDTH = 600;
    private static final int HEIGHT = 400;

    ////////////////////////////////////////MAIN
    ////////////////////////////////////////
    public static void main(String[] args) {

        JFrame frame = new RentalView();
        ImageIcon img = new ImageIcon("../images/logo.GIF");
        frame.setIconImage(img.getImage());
        frame.setTitle("Welcome to MyCar Rental");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(WIDTH, HEIGHT);
        frame.setVisible(true);
        frame.setResizable(false);
    }
}
```

Car.java

```
import java.util.*;
import java.io.*;

////////////////////////////////////Car
class////////////////////////////////////
public class Car {

    private String carType, carModel;
    private double carPrice;

    public Car() {

        carType = "";
        carModel = "";
        carPrice = 0;
    }

    public void setCar(String cT, String cM, double cP){

        carType = cT;
        carModel = cM;
        carPrice = cP;
    }

    public String getCarType() {
        return carType;
    }

    public String getCarModel() {
        return carModel;
    }

    public String getCar() {
        return getCarType() + "-" + getCarModel();
    }

    public double getCarPrice() {
        return carPrice;
    }

    // public void saveToFile() throws IOException {

    //     PrintWriter outFile = new PrintWriter(new
    FileOutputStream(new File("Cars_Inventory.txt"), true));
```

```
//      outFile.println(getCar() + " " + getCarPrice());
//      outFile.close();
//  }
}
```

Customer.java

```
import java.util.*;
import java.io.*;

////////////////////////////////////Customer
Class////////////////////////////////
public class Customer {

    private String firstName, lastName, gender, status,
location, contact;

    public Customer() {

        firstName = "";
        lastName = "";
        gender = "";
        status = "";
        location = "";
        contact = "";
    }

    public void setCustomer(String fN, String lN, String g,
String s, String l, String c){

        firstName = fN;
        lastName = lN;
        gender = g;
        status = s;
        location = l;
        contact = c;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
```

```
        return lastName;
    }

    public String getFullName() {
        return getFirstName() + " " + getLastName();
    }

    public String getGender() {
        return gender;
    }

    public String getStatus() {
        return status;
    }

    public String getLocation() {
        return location;
    }

    public String getContact() {
        return contact;
    }

    // public void saveToFile() throws IOException {

    //     PrintWriter outFile = new PrintWriter(new
    FileOutputStream(new File("Customers_Inventory.txt"), true));
    //     outFile.println(getFullName() + " " + getGender() + " "
    //     + getLocation() + " " + getContact() + " " +
    getStatus());
    //     outFile.close();
    // }
}
```

RentalView.java

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.*;
import java.util.*;

public class RentalView extends JFrame{

    private Transaction aTrans = new Transaction(); //access my
class Transaction as global declaration
    private Customer aCustomer = new Customer(); //access class
Customer as global declaration
    private Car aCar = new Car(); //access class car as global
declaration
    WarningReserve openWarningReserve; //another class to
access warning window
    WarningCustomer openWarningCustomer;
    WarningCar openWarningCar;

    Container pane = getContentPane();
    private JPanel mainP, reserveP, customerP, carP, finalP;
//panels for CardLayout

////////////////////////welcome////////////////////////////////////
////////////////////////
    private JLabel titleL, dateL; //inside the panel of mainP
    private JButton mainNextB;
    private JComboBox cMCB, cDCB, cYCB;

////////////////////////date
reservation////////////////////////////////////
    private JLabel dayL, pickupDateL, returnDateL, birthDateL;
//inside the panel of reserveP
    private JComboBox pUMCB, pUDCB, pUYCB, rMCB, rDCB, rYCB,
        bMCB, bDCB, bYCB;
    private JLabel dateChecker, ageChecker;
    private JButton reserveNextB, reserveBackB;

////////////////////////customer
info////////////////////////////////////
    private JLabel customerL, firstNameL, lastNameL, locationL,
contactL, genderL, ageL, statusL, showAgeL; //inside the panel
```

```

of customerP
    private JTextField firstNameTF, lastNameTF, locationTF,
contactTF, genderTF, statusTF;
    private JLabel firstNameChecker, lastNameChecker,
genderChecker, statusChecker, locationChecker, contactChecker;
    private JButton customerNextB, customerBackB;

//////////////////////////////////////////pick a
car//////////////////////////////////////////
    private JLabel carL; //inside the panel of carP
    private JButton firstCarB, secondCarB, thirdCarB,
forthCarB, carNextB, carBackB;
    private JTextArea carDetailsTA;

//////////////////////////////////////////receipt//////////
//////////////////////////////////////////

    private JTextArea receiptA;

    public RentalView() {

//////////////////////////////////////////First
Panel "Welcome" (mainP)//////////////////////////////////////////
    //-----setting
PANEL and LABELS for Welcome section

        pane.setLayout(new CardLayout());
        mainP = new JPanel(null);
        mainP.setBackground(Color.WHITE);

        ImageIcon titleImage = new
ImageIcon("../images/logo2.GIF");
        titleL = new JLabel(titleImage);
        titleL.setBounds(100, 40, 310, 300);
        mainP.add(titleL);

        dateL = new JLabel("DATE TODAY"); //date today label
        dateL.setFont(new Font("Serif", Font.BOLD, 23));
        dateL.setForeground(Color.RED);
        dateL.setBounds(360, 210, 150, 25);
        mainP.add(dateL);

    //-----COMBO BOX for
current date
        String[] currentMonthCB = {"Jan", "Feb", "Mar", "Apr",
"May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
        cMCB = new JComboBox(currentMonthCB);
        cMCB.setBounds(360, 240, 60, 25);

```

```

        mainP.add(cMCB);

        Integer[] currentDateCB = {1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24,
25, 26, 27, 28, 29, 30, 31};
        cDCB = new JComboBox(currentDateCB);
        cDCB.setBounds(425, 240, 60, 25);
        mainP.add(cDCB);

        Integer[] currentYearCB = {2013, 2014, 2015, 2016, 2017,
2018, 2019, 2020};
        cYCB = new JComboBox(currentYearCB);
        cYCB.setBounds(490, 240, 60, 25);
        mainP.add(cYCB);

        //-----BUTTONS to go reserve
section
        ImageIcon mainNextImage = new
ImageIcon("../images/next.GIF");
        mainNextB = new JButton(mainNextImage);
        mainNextB.setBackground(Color.WHITE);
        mainNextB.addActionListener(new mainNextButton());
        mainNextB.setBounds(460, 300, 100, 50);
        mainNextB.setRolloverEnabled(true);
        mainNextB.setRolloverIcon(mainNextImage);
        mainNextB.setPressedIcon(mainNextImage);
        mainNextB.setActionCommand("NextToReserve");
        mainP.add(mainNextB);
        pane.add(mainP, "welcomeForm");

        //////////////////////////////////////////Second
Panel "reserve"
(reserveP)////////////////////////////////////////
        //-----LABELS for
reservation date.
        reserveP = new JPanel(null);
        reserveP.setBackground(Color.WHITE);

        dayL = new JLabel("Reserve Rental Day(s):");
        dayL.setFont(new Font("Verdana", Font.PLAIN, 18));
        dayL.setForeground(Color.BLACK);
        dayL.setBounds(50, 1, 250, 100);
        reserveP.add(dayL);

```



```

pickupDateL = new JLabel("Pick-up Date:");
pickupDateL.setFont(new Font("Serif", Font.BOLD, 15));
pickupDateL.setForeground(Color.BLUE);
pickupDateL.setBounds(50, 40, 100, 100);
reserveP.add(pickupDateL);

returnDateL = new JLabel("Return Date:");
returnDateL.setFont(new Font("Serif", Font.BOLD, 15));
returnDateL.setForeground(Color.BLUE);
returnDateL.setBounds(50, 80, 100, 100);
reserveP.add(returnDateL);

birthDateL = new JLabel("Birth Date:");
birthDateL.setFont(new Font("Serif", Font.BOLD, 15));
birthDateL.setForeground(Color.BLUE);
birthDateL.setBounds(60, 150, 100, 100);
reserveP.add(birthDateL);

//-----CHECK
BOX for pickup date, return date, birth date
//-----Pick up date
section
    String[] pickUpMonthCB = {"Jan", "Feb", "Mar", "Apr",
"May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
    pUMCB = new JComboBox(pickUpMonthCB);
    pUMCB.setBounds(150, 80, 60, 25);
    reserveP.add(pUMCB);

    Integer[] pickUpDateCB = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31};
    pUDCB = new JComboBox(pickUpDateCB);
    pUDCB.setBounds(215, 80, 60, 25);
    reserveP.add(pUDCB);

    Integer[] pickUpYearCB = {2013, 2014, 2015, 2016, 2017,
2018, 2019, 2020};
    pUYCB = new JComboBox(pickUpYearCB);
    pUYCB.setBounds(280, 80, 60, 25);
    reserveP.add(pUYCB);

//-----return date
section
    String[] returnMonthCB = {"Jan", "Feb", "Mar", "Apr",
"May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
    rMCB = new JComboBox(returnMonthCB);

```

```

rMCB.setBounds(150, 120, 60, 25);
reserveP.add(rMCB);

Integer[] returnDateCB = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31};
rDCB = new JComboBox(returnDateCB);
rDCB.setBounds(215, 120, 60, 25);
reserveP.add(rDCB);

Integer[] returnYearCB = {2013, 2014, 2015, 2016, 2017,
2018, 2019, 2020};
rYCB = new JComboBox(returnYearCB);
rYCB.setBounds(280, 120, 60, 25);
reserveP.add(rYCB);

//-----birth date
section
String[] birthMonthCB = {"Jan", "Feb", "Mar", "Apr",
"May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
bMCB = new JComboBox(birthMonthCB);
bMCB.setBounds(150, 190, 60, 25);
reserveP.add(bMCB);

Integer[] birthDateCB = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25,
26, 27, 28, 29, 30, 31};
bDCB = new JComboBox(birthDateCB);
bDCB.setBounds(215, 190, 60, 25);
reserveP.add(bDCB);

Integer[] birthYearCB = {1970, 1971, 1972, 1973, 1974,
1975, 1976, 1977, 1978, 1979, 1980,
1981, 1982, 1983, 1984, 1985, 1986,
1987, 1988, 1989, 1990,
1991, 1992, 1993, 1994, 1995, 1996,
1997, 1998, 1999, 2000,
2001, 2002, 2003, 2004, 2005, 2006,
2007, 2008, 2009, 2010,
2011, 2012, 2013, 2014, 2015, 2016,
2017, 2018, 2019, 2020};
bYCB = new JComboBox(birthYearCB);
bYCB.setBounds(280, 190, 60, 25);
reserveP.add(bYCB);

//-----checker for
date

```

```

dateChecker = new JLabel("");
dateChecker.setFont(new Font("Serif", Font.BOLD, 15));
dateChecker.setBounds(350, 83, 100, 100);
reserveP.add(dateChecker);

ageChecker = new JLabel("");
ageChecker.setFont(new Font("Serif", Font.BOLD, 15));
ageChecker.setBounds(350, 153, 100, 100);
reserveP.add(ageChecker);

//-----BUTTONS
//-----to go
customer section
    ImageIcon dateNextImage = new
ImageIcon("../images/customer.GIF");
    reserveNextB = new JButton(dateNextImage);
    reserveNextB.setBackground(Color.WHITE);
    reserveNextB.addActionListener(new reserveNextButton());
    reserveNextB.setBounds(460, 300, 100, 50);
    reserveNextB.setActionCommand("NextToCustomer");
    reserveP.add(reserveNextB);

//-----to go
back welcome section
    //ImageIcon dateNextImage = new
ImageIcon("../images/Arrow-Right-icon.GIF");
    ImageIcon dateBackImage = new
ImageIcon("../images/back.GIF");
    reserveBackB = new JButton(dateBackImage);
    reserveBackB.setBackground(Color.WHITE);
    reserveBackB.setBounds(330, 300, 100, 50);
    reserveBackB.addActionListener(new reserveBackButton());
    reserveBackB.setActionCommand("BackToWelcome");
    reserveP.add(reserveBackB);
pane.add(reserveP, "reserveForm");

////////////////////////////////////thir
d Panel "customer"
(customerP)////////////////////////////////
//-----LABELS for
customer details
    customerP = new JPanel(null);
    customerP.setBackground(Color.WHITE);

    customerL = new JLabel("Customer's Data");

```

```
customerL.setFont(new Font("Verdana", Font.PLAIN, 18));
customerL.setForeground(Color.BLACK);
customerL.setBounds(50, 10, 250, 30);
customerP.add(customerL);

firstNameL = new JLabel("First Name:");
firstNameL.setFont(new Font("Serif", Font.BOLD, 15));
firstNameL.setForeground(Color.BLUE);
firstNameL.setBounds(50, 60, 250, 30);
customerP.add(firstNameL);

lastNameL = new JLabel("Last Name:");
lastNameL.setFont(new Font("Serif", Font.BOLD, 15));
lastNameL.setForeground(Color.BLUE);
lastNameL.setBounds(50, 95, 250, 30);
customerP.add(lastNameL);

genderL = new JLabel("Gender:");
genderL.setFont(new Font("Serif", Font.BOLD, 15));
genderL.setForeground(Color.BLUE);
genderL.setBounds(50, 130, 250, 30);
customerP.add(genderL);

statusL = new JLabel("Status:");
statusL.setFont(new Font("Serif", Font.BOLD, 15));
statusL.setForeground(Color.BLUE);
statusL.setBounds(50, 165, 250, 30);
customerP.add(statusL);

locationL = new JLabel("Location:");
locationL.setFont(new Font("Serif", Font.BOLD, 15));
locationL.setForeground(Color.BLUE);
locationL.setBounds(50, 200, 250, 30);
customerP.add(locationL);

contactL = new JLabel("Contact:");
contactL.setFont(new Font("Serif", Font.BOLD, 15));
contactL.setForeground(Color.BLUE);
contactL.setBounds(50, 235, 250, 30);
customerP.add(contactL);

ageL = new JLabel("Age:");
ageL.setFont(new Font("Serif", Font.BOLD, 15));
ageL.setForeground(Color.BLUE);
ageL.setBounds(50, 270, 250, 30);
customerP.add(ageL);
```

```

showAgeL = new JLabel("");
showAgeL.setFont(new Font("Serif", Font.PLAIN, 15));
showAgeL.setHorizontalAlignment(JTextField.CENTER);
showAgeL.setForeground(Color.BLUE);
showAgeL.setBounds(145, 270, 60, 30);
customerP.add(showAgeL);

//-----TEXT FIELD for
customer details and get data from keyboard section
    firstNameTF = new JTextField("Type First Name Here"); //
set as 10 ha
    firstNameTF.setHorizontalAlignment(JTextField.CENTER);
    firstNameTF.setForeground(Color.GRAY);
    firstNameTF.setBounds(150, 60, 250, 30);
    customerP.add(firstNameTF);

    lastNameTF = new JTextField("Type Last Name Here"); //
kani pud
    lastNameTF.setHorizontalAlignment(JTextField.CENTER);
    lastNameTF.setForeground(Color.GRAY);
    lastNameTF.setBounds(150, 95, 250, 30);
    customerP.add(lastNameTF);

    genderTF = new JTextField("M/F");
    genderTF.setHorizontalAlignment(JTextField.CENTER);
    genderTF.setForeground(Color.GRAY);
    genderTF.setBounds(150, 130, 50, 30);
    customerP.add(genderTF);

    statusTF = new JTextField("S/Ma");
    statusTF.setHorizontalAlignment(JTextField.CENTER);
    statusTF.setForeground(Color.GRAY);
    statusTF.setBounds(150, 165, 50, 30);
    customerP.add(statusTF);

    locationTF = new JTextField("Type Complete Address
Here");
    locationTF.setHorizontalAlignment(JTextField.CENTER);
    locationTF.setForeground(Color.GRAY);
    locationTF.setBounds(150, 200, 250, 30);
    customerP.add(locationTF);

    contactTF = new JTextField("Type Contact Number Here");
    contactTF.setHorizontalAlignment(JTextField.CENTER);
    contactTF.setForeground(Color.GRAY);
    contactTF.setBounds(150, 235, 250, 30);
    customerP.add(contactTF);

```

```

//-----checker customer

    firstNameChecker = new JLabel("");
    firstNameChecker.setFont(new Font("Serif", Font.BOLD,
15));
    firstNameChecker.setBounds(410, 25, 100, 100);
    customerP.add(firstNameChecker);

    lastNameChecker = new JLabel("");
    lastNameChecker.setFont(new Font("Serif", Font.BOLD,
15));
    lastNameChecker.setBounds(410, 60, 100, 100);
    customerP.add(lastNameChecker);

    genderChecker = new JLabel("");
    genderChecker.setFont(new Font("Serif", Font.BOLD, 15));
    genderChecker.setBounds(210, 95, 100, 100);
    customerP.add(genderChecker);

    statusChecker = new JLabel("");
    statusChecker.setFont(new Font("Serif", Font.BOLD, 15));
    statusChecker.setBounds(210, 130, 100, 100);
    customerP.add(statusChecker);

    locationChecker = new JLabel("");
    locationChecker.setFont(new Font("Serif", Font.BOLD,
15));
    locationChecker.setBounds(410, 165, 100, 100);
    customerP.add(locationChecker);

    contactChecker = new JLabel("");
    contactChecker.setFont(new Font("Serif", Font.BOLD,
15));
    contactChecker.setBounds(410, 200, 100, 100);
    customerP.add(contactChecker);

//-----BUTTONS for
customer
//-----to go pick
a car section
    ImageIcon customerNextImage = new
ImageIcon("../images/car.GIF");
    customerNextB = new JButton(customerNextImage);
    customerNextB.setBackground(Color.WHITE);
    customerNextB.setBounds(460, 300, 100, 50);
    customerNextB.setRolloverEnabled(true);

```

```

        customerNextB.setRolloverIcon(customerNextImage);
        customerNextB.setPressedIcon(customerNextImage);
        customerNextB.addActionListener(new
customerNextButton());
        customerNextB.setActionCommand("NextToPickCar");
        customerP.add(customerNextB);

//-----to go back
reserve section
        //ImageIcon datenext = new ImageIcon("../images/Arrow-
Right-icon.GIF");
        ImageIcon customerBackImage = new
ImageIcon("../images/customer_back.GIF");
        customerBackB = new JButton(customerBackImage);
        customerBackB.setBackground(Color.WHITE);
        customerBackB.setBounds(330, 300, 100, 50);
        customerBackB.setRolloverEnabled(true);
        customerBackB.setRolloverIcon(customerBackImage);
        customerBackB.setPressedIcon(customerBackImage);
        customerBackB.addActionListener(new
customerBackButton());
        customerBackB.setActionCommand("backToReserve");
        customerP.add(customerBackB);
        pane.add(customerP, "customerForm");

//ImageIcon image = new ImageIcon("../images/one.GIF");
//ow = new JLabel(image);
//ow.setBounds(50, 40, 100, 100);
//pane.add(ow);

/////////////////////////////////////////forth
Panel "pick a car"
(carP)/////////////////////////////////////////
//-----LABELS for pick a car
        carP = new JPanel(null);
        carP.setBackground(Color.WHITE);

        carL = new JLabel("Car Detail's");
        carL.setFont(new Font("Verdana", Font.PLAIN, 18));
        carL.setForeground(Color.BLACK);
        carL.setBounds(435, 4, 140, 30);
        carP.add(carL);

//-----BUTTONS for pick a
car
//-----
Pick Car Buttton]

```

```

//-----first car button
    ImageIcon firstCarImage = new
ImageIcon("../images/van.GIF");
    firstCarB = new JButton(firstCarImage);
    firstCarB.setBackground(Color.WHITE);
    firstCarB.addActionListener(new firstCarButton());
    firstCarB.setRolloverEnabled(true);
    firstCarB.setRolloverIcon(firstCarImage);
    firstCarB.setPressedIcon(firstCarImage);
    firstCarB.setBounds(10, 10, 190, 130);
    carP.add(firstCarB);

//-----second car button
    ImageIcon secondCarImage = new
ImageIcon("../images/pickup.GIF");
    secondCarB = new JButton(secondCarImage);
    secondCarB.setBackground(Color.WHITE);
    secondCarB.addActionListener(new secondCarButton());
    secondCarB.setRolloverEnabled(true);
    secondCarB.setRolloverIcon(secondCarImage);
    secondCarB.setPressedIcon(secondCarImage);
    secondCarB.setBounds(210, 10, 190, 130);
    carP.add(secondCarB);

//-----third car button
    ImageIcon thirdCarImage = new
ImageIcon("../images/bezza.GIF");
    thirdCarB = new JButton(thirdCarImage);
    thirdCarB.setBackground(Color.WHITE);
    thirdCarB.addActionListener(new thirdCarButton());
    thirdCarB.setRolloverEnabled(true);
    thirdCarB.setRolloverIcon(thirdCarImage);
    thirdCarB.setPressedIcon(thirdCarImage);
    thirdCarB.setBounds(10, 150, 190, 130);
    carP.add(thirdCarB);

//-----forth car
button
    ImageIcon forthCarImage = new
ImageIcon("../images/axia.GIF");
    forthCarB = new JButton(forthCarImage);
    forthCarB.setBackground(Color.WHITE);
    forthCarB.addActionListener(new forthCarButton());
    forthCarB.setRolloverEnabled(true);
    forthCarB.setRolloverIcon(forthCarImage);
    forthCarB.setPressedIcon(forthCarImage);
    forthCarB.setBounds(210, 150, 190, 130);

```



```

        carP.add(forthCarB);

//-----to go
receipt section (Final step)
    ImageIcon carNextImage = new
ImageIcon("../images/submit.GIF");
    carNextB = new JButton(carNextImage);
    carNextB.setBackground(Color.WHITE);
    carNextB.setBounds(460, 300, 100, 50);
    carNextB.addActionListener(new finalStepButton());
    carNextB.setRolloverEnabled(true);
    carNextB.setRolloverIcon(carNextImage);
    carNextB.setPressedIcon(carNextImage);
    carP.add(carNextB);

//-----
-----to go back customer section
    ImageIcon carBackImage = new
ImageIcon("../images/car_back.GIF");
    carBackB = new JButton(carBackImage);
    carBackB.setBackground(Color.WHITE);
    carBackB.addActionListener(new carBackButton());
    carBackB.setBounds(330, 300, 100, 50);
    carBackB.setRolloverEnabled(true);
    carBackB.setRolloverIcon(carBackImage);
    carBackB.setPressedIcon(carBackImage);
    carBackB.setActionCommand("BackToCustomer");
    carP.add(carBackB);

//-----TEXT AREA for pick a
car putting details for car when we click buttons over top
    carDetailsTA = new JTextArea(6, 25);
    carDetailsTA.setText("Select a car");
    carDetailsTA.setAutoscrolls(true);
    carDetailsTA.setEditable(false);
    carDetailsTA.setBounds(420, 40, 150, 240);
    carP.add(carDetailsTA);
    pane.add(carP, "pickCarForm");

////////////////////////////////////////last
panel "recept" (finalP)////////////////////////////////////////
    finalP = new JPanel(null);

    receiptA = new JTextArea(2000, 2000);
    receiptA.setFont(new Font("Serif", Font.PLAIN, 15));
    receiptA.setForeground(Color.BLUE);

```

```

        receiptA.setEditable(false);
        receiptA.setBounds(80, 30, 400, 800); //
        finalP.add(receiptA);
        pane.add(finalP, "receipt");

    }

    //////////////////////////////////////CT
    ROLLERS////////////////////////////////////
    //-----main (go to
reserveForm)
    private class mainNextButton implements ActionListener{
        public void actionPerformed(ActionEvent event) {
            CardLayout cL = (CardLayout)pane.getLayout();

            if(event.getActionCommand().equals("NextToReserve"))
{
                cL.show(pane, "reserveForm");
            }
        }
    }

    //-----reserve
    //-----go to
customerForm
    //-----including
the validation if requirements are meet for Next Panel
    private class reserveNextButton implements ActionListener{
        public void actionPerformed(ActionEvent event) {
            CardLayout cL = (CardLayout)pane.getLayout();

            if(event.getActionCommand().equals("NextToCustomer"))
{

                String currentMonth, pickUpMonth, returnMonth,
birthMonth;
                int    currentDate, currentYear, pickUpDate,
pickUpYear, returnDate,
                    returnYear, birthDate, birthYear;

                currentMonth = (String)cMCB.getSelectedItemAt();
                currentDate = (int)cDCB.getSelectedItemAt();
                currentYear = (int)cYCB.getSelectedItemAt();

                pickUpMonth = (String)pUMCB.getSelectedItemAt();
                pickUpDate = (int)pUDCB.getSelectedItemAt();

```

```

        pickUpYear = (int)pUYCB.getSelectedItem();

        returnMonth = (String)rMCB.getSelectedItem();
        returnDate = (int)rDCB.getSelectedItem();
        returnYear = (int)rYCB.getSelectedItem();

        birthMonth = (String)bMCB.getSelectedItem();
        birthDate = (int)bDCB.getSelectedItem();
        birthYear = (int)bYCB.getSelectedItem(); //saving
dates for transaction, validation for date and age, will
prompt to WarningReserve

        aTrans.setTransaction(aCustomer, aCar,
currentMonth, currentDate, currentYear, pickUpMonth,
        pickUpDate, pickUpYear, returnMonth, returnDate,
returnYear,
        birthMonth, birthDate, birthYear);
        showAgeL.setText(aTrans.toStringAge()); // show
the text of age from class

        if(!aTrans.toStringDateChecker().equals("TRUE")) {
//check if valid or not for pick date and return date
            dateChecker.setForeground(Color.RED);
            dateChecker.setText("Invalid!");
        }

        else {
            dateChecker.setForeground(Color.GREEN);
            dateChecker.setText("Valid");
        }

        if(!aTrans.toStringAgeChecker().equals("TRUE")) {
// for age validation
            ageChecker.setForeground(Color.RED);
            ageChecker.setText("Invalid!");
        }

        else {
            ageChecker.setForeground(Color.GREEN);
            ageChecker.setText("Valid");
        }

        if(aTrans.toStringDateChecker().equals("TRUE") &&
aTrans.toStringAgeChecker().equals("TRUE")) { //checker to
proceed next panel
            cL.show(pane, "customerForm");
        }

```

```

        else { //pop up window for warning to customer
            if(openWarningReserve == null){
                openWarningReserve = new WarningReserve();
                ImageIcon warningReserveImage = new
ImageIcon("../images/warning_top_icon.GIF");

openWarningReserve.setIconImage(warningReserveImage.getImage()
);

                openWarningReserve.setTitle("WARNING !!");

openWarningReserve.setDefaultCloseOperation(JFrame.DISPOSE_ON_
CLOSE);

                openWarningReserve.setSize(300, 150);
                openWarningReserve.setVisible(true);
                openWarningReserve.setResizable(false);
            }

            else{

openWarningReserve.setVisible(!openWarningReserve.isVisible())
;

            }
        }
    }
}

//-----go back
to welcomeForm
private class reserveBackButton implements ActionListener{
    public void actionPerformed(ActionEvent event) {
        CardLayout cL = (CardLayout)pane.getLayout();

        if(event.getActionCommand().equals("BackToWelcome"))
{
            cL.show(pane, "welcomeForm");
        }
    }
}

//-----customer
//-----go to
pickCarForm
//-----including
the validation if requirements are meet for Next Panel
private class customerNextButton implements ActionListener{

```

```

    public void actionPerformed(ActionEvent event) {
        CardLayout cL = (CardLayout)pane.getLayout();

        if(event.getActionCommand().equals("NextToPickCar"))
        {

            String firstName, lastName, gender, status,
location, contact;

            firstName = firstNameTF.getText();
            lastName = lastNameTF.getText();
            gender = genderTF.getText();
            status = statusTF.getText();
            location = locationTF.getText();
            contact = contactTF.getText();

            aCustomer.setCustomer(firstName, lastName, gender,
status, location, contact);

            if(aCustomer.getFirstName().equals("Type First
Name Here")) { // checker for firstname if empty , its invalid
                firstNameChecker.setForeground(Color.RED);
                firstNameChecker.setText("Invalid!");
            }

            else {
                firstNameChecker.setForeground(Color.GREEN);
                firstNameChecker.setText("Valid");
            }

            if(aCustomer.getLastName().equals("Type Last Name
Here")) { //for last name
                lastNameChecker.setForeground(Color.RED);
                lastNameChecker.setText("Invalid!");
            }

            else {
                lastNameChecker.setForeground(Color.GREEN);
                lastNameChecker.setText("Valid");
            }

            if(aCustomer.getGender().equals("M/F")) { //for
gender
                genderChecker.setForeground(Color.RED);
                genderChecker.setText("Invalid!");
            }

```

```

else {
    genderChecker.setForeground(Color.GREEN);
    genderChecker.setText("Valid");
}

status
if(aCustomer.getStatus().equals("S/Ma")) { //for

    statusChecker.setForeground(Color.RED);
    statusChecker.setText("Invalid!");
}

else {
    statusChecker.setForeground(Color.GREEN);
    statusChecker.setText("Valid");
}

if(aCustomer.getLocation().equals("Type Complete
Address Here")) { //for location
    locationChecker.setForeground(Color.RED);
    locationChecker.setText("Invalid!");
}

else {
    locationChecker.setForeground(Color.GREEN);
    locationChecker.setText("Valid");
}

if(aCustomer.getContact().equals("Type Contact
Number Here")) { //for contact
    contactChecker.setForeground(Color.RED);
    contactChecker.setText("Invalid!");
}

else {
    contactChecker.setForeground(Color.GREEN);
    contactChecker.setText("Valid");
}

if(!aCustomer.getFirstName().equals("Type First
Name Here") && !aCustomer.getLastName().equals("Type Last Name
Here") && !aCustomer.getGender().equals("M/F") &&
!aCustomer.getStatus().equals("S/Ma") &&
!aCustomer.getLocation().equals("Type Complete Address Here")
&& !aCustomer.getContact().equals("Type Contact Number
Here")) {

    firstNameTF.setText("Type First Name Here");

```

```

//if all of the above is valid , proceed to Car Form
    lastNameTF.setText("Type Last Name Here");
    genderTF.setText("M/F");
    statusTF.setText("S/Ma");
    locationTF.setText("Type Complete Address
Here");

    contactTF.setText("Type Contact Number Here");
    cL.show(pane, "pickCarForm");
}

else { //if invalid, pop out window for warning
    if(openWarningCustomer == null){
        openWarningCustomer = new WarningCustomer();
        ImageIcon warningCustomerImage = new
ImageIcon("../images/warning_top_icon.GIF");

openWarningCustomer.setIconImage(warningCustomerImage.getImage
());

        openWarningCustomer.setTitle("WARNING
INVALID!!");

openWarningCustomer.setDefaultCloseOperation(JFrame.DISPOSE_ON
_CLOSE);

        openWarningCustomer.setSize(300, 150);
        openWarningCustomer.setVisible(true);
        openWarningCustomer.setResizable(false);
    }

    else{

openWarningCustomer.setVisible(!openWarningCustomer.isVisible(
));

    }

}

}

}

}

//-----go back to
reserveForm
    private class customerBackButton implements ActionListener{
        public void actionPerformed(ActionEvent event) {
            CardLayout cL = (CardLayout)pane.getLayout();

            if(event.getActionCommand().equals("backToReserve"))
{
                cL.show(pane, "reserveForm");

```

```

    }
}

//-----car
//-----go
back to customerForm
    private class firstCarButton implements ActionListener{
//when its clicked show up the car detail of Van
        public void actionPerformed(ActionEvent event) {

            carDetailsTA.setText("");
            carDetailsTA.setText("Brand: Toyota Hiace\nPlate #: SAB
9511 C\nType: Van\nModel: 2014\nColor: White\nRental Price:
MYR550.00\n\n- Used for tourists, family \nuse, business
purpose. \nThe Vans capacity can \nhold onto 10 persons
\nexcluding the driver.");
            carNextB.setActionCommand("Car1");
        }
    }

    private class secondCarButton implements ActionListener{
//button to show pick-up details
        public void actionPerformed(ActionEvent event) {

            carDetailsTA.setText("");
            carDetailsTA.setText("Brand: Ford\nPlate #: SAB 828
R\nType: Pick-up\nModel: 2017\nColor: Dirty White\nRental
Price: MYR700.00\n\n- Oftenly used for travelers \nsuch as
adventureseekers. \nCan hold loads from its \nback open space.
It's tires \nare good for long travels \nand bumpy roads.");
            carNextB.setActionCommand("Car2");
        }
    }

    private class thirdCarButton implements ActionListener{
//button to show Taxi details
        public void actionPerformed(ActionEvent event) {

            carDetailsTA.setText("");
            carDetailsTA.setText("Brand: Perodua Bezza\nPlate #: SYC
698 Q\nType: Sedan\nModel: 2016\nColor: Sugar Brown\nRental
Price: MYR250.00\n\n- Malaysia's famous \ntransport nation
that can be used \nprivately.Tourists can \nexpernience riding
on \nthis type of car and \nits daily use of commuting \nto
places.");
            carNextB.setActionCommand("Car3");
        }
    }
}

```



```

    }
}
//
private class forthCarButton implements ActionListener{
//button to show Bus details
    public void actionPerformed(ActionEvent event) {

        carDetailsTA.setText("");
        carDetailsTA.setText("Brand: Perodua Axia\nPlate #: SYC
009 R\nType: Hatchback\nModel: 2015\nColor: White\nRental
Price:MYR150.00\n\n- Good for events, such as \nweddings and
proms. \nIts not only for events but \ncan be used for touring
too. \nThe luxurious lookgives off \na good feeling to the
\npassengers because of \nits comfortable outlook.");
        carNextB.setActionCommand("Car4");
    }
}

private class finalStepButton implements ActionListener{
    public void actionPerformed(ActionEvent event) {
        CardLayout cL = (CardLayout)pane.getLayout();

        if(event.getActionCommand().equals("Car1")) {
//passing the pick-up details to car details class and final
transactions.

            String carType = "Van", carModel = "2013";
            double carPrice = 550.00;

            aCar.setCar(carType, carModel, carPrice);
            try { //to save in file Transaction_Inventory
                aTrans.saveToFile();
            }catch (IOException e) {}
            receiptA.setText(aTrans.toStringReceipt());
            cL.show(pane, "rceipt");
        }

        else if(event.getActionCommand().equals("Car2")) {
//pass the details of Jepp to car details

            String carType = "Pick-up", carModel = "2017";
            double carPrice = 700.00;

            aCar.setCar(carType, carModel, carPrice);

            try { //to save in file Transaction_Inventory
                aTrans.saveToFile();
            }

```

```

        }catch (IOException e) {}

        receiptA.setText(aTrans.toStringReceipt());
        cL.show(pane, "receipt");
    }

    else if(event.getActionCommand().equals("Car3")) {
//pass Van details to car details

        String carType = "Sedan", carModel = "2016";
        double carPrice = 250.00;

        aCar.setCar(carType, carModel, carPrice);

        try { //to save in file Transaction_Inventory
            aTrans.saveToFile();
        }catch (IOException e) {}

        receiptA.setText(aTrans.toStringReceipt());
        cL.show(pane, "receipt");
    }

    else if(event.getActionCommand().equals("Car4")) {
//pass mr.bean car details to car details

        String carType = "Hacthback", carModel = "2015";
        double carPrice = 150.00;

        aCar.setCar(carType, carModel, carPrice);

        try { //to save in file Transaction_Inventory
            aTrans.saveToFile();
        }catch (IOException e) {}

        receiptA.setText(aTrans.toStringReceipt());
        cL.show(pane, "receipt");
    }

    else { //if customer havent yet pick a car, show's
the warning information
        if(openWarningCar == null){
            openWarningCar = new WarningCar();
            ImageIcon warningCarImage = new
ImageIcon("../images/warning_top_icon.GIF");

openWarningCar.setIconImage(warningCarImage.getImage());
            openWarningCar.setTitle("WARNING INVALID!!");

```

```

openWarningCar.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        openWarningCar.setSize(300, 150);
        openWarningCar.setVisible(true);
        openWarningCar.setResizable(false);
    }

    else{

openWarningCar.setVisible(!openWarningCar.isVisible());
    }
    }
    }

    private class carBackButton implements ActionListener{ //go
back to customer form when it is clicked and performed
        public void actionPerformed(ActionEvent event) {
            CardLayout cL = (CardLayout)pane.getLayout();

            if(event.getActionCommand().equals("BackToCustomer"))
{
                cL.show(pane, "customerForm");
            }
        }
    }
}

```

Transaction.java

```
import java.util.*;
import java.io.*;

public class Transaction {

    final static int MONTHS_DAYS = 30, MONTHS = 12; //lets
    assume that 30 is one month for every month in this program
    private Customer aCustomer; //from the Customer class in
    same directory
    private Car aCar; //frm the Customer class in same
    directory
    private int monthe = 0, days = 0;

    private String CurrentMonth;
    private int CurrentDate, Currentyear;

    private String PickMonth;
    private int PickDate, PickYear;

    private String returnMonth;
    private int returnDate, returnYear;

    private String birthMonth;
    private int birthDate, birthYear, age = 0;

    public Transaction() { //constructor
        aCustomer = new Customer();
        aCar = new Car();
        CurrentMonth = "";
        CurrentDate = 0;
        Currentyear = 0;
        PickMonth = "";
        PickDate = 0;
        PickYear = 0;
        returnMonth = "";
        returnDate = 0;
        returnYear = 0;
        birthMonth = "";
        birthDate = 0;
        birthYear = 0;
    }

    public void setTransaction(Customer cU, Car cA, String m,
    int d, int y, String pM, int pD, int pY, String rM, int rD,
```

```

int rY, String bM, int bD, int bY) { //constructor
    aCustomer = cU;
    aCar = cA;
    CurrentMonth = m;
    CurrentDate = d;
    Currentyear = y;
    PickMonth = pM;
    PickDate = pD;
    PickYear = pY;
    returnMonth = rM;
    returnDate = rD;
    returnYear = rY;
    birthMonth = bM;
    birthDate = bD;
    birthYear = bY;
}

public String getCurrentMonth() {
    return CurrentMonth;
}

public int getCurrentDate() {
    return CurrentDate;
}

public int getCurrentYear() {
    return Currentyear;
}

////////////////////////////////////getting the value
of current month////////////////////////////////
public int getCurrentMonthValue() {
    if(getCurrentMonth().equals("Jan"))
        monthe = 1;

    else if(getCurrentMonth().equals("Feb"))
        monthe = 2;

    else if(getCurrentMonth().equals("Mar"))
        monthe = 3;

    else if(getCurrentMonth().equals("Apr"))
        monthe = 4;

    else if(getCurrentMonth().equals("May"))
        monthe = 5;

    else if(getCurrentMonth().equals("Jun"))

```

```

        monthe = 6;

    else if(getCurrentMonth().equals("Jul"))
        monthe = 7;

    else if(getCurrentMonth().equals("Aug"))
        monthe = 8;

    else if(getCurrentMonth().equals("Sep"))
        monthe = 9;

    else if(getCurrentMonth().equals("Oct"))
        monthe = 10;

    else if(getCurrentMonth().equals("Nov"))
        monthe = 11;

    else if(getCurrentMonth().equals("Dec"))
        monthe = 12;

    return monthe;
}

public String getPickMonth() {
    return PickMonth;
}

public int getPickDate() {
    return PickDate;
}

public int getPickYear() {
    return PickYear;
}

////////////////////////////////////getting the value of
pick up month////////////////////////////////////
    public int getPickMonthValue() {
        if(getPickMonth().equals("Jan"))
            monthe = 1;

        else if(getPickMonth().equals("Feb"))
            monthe = 2;

        else if(getPickMonth().equals("Mar"))
            monthe = 3;
    }

```

```

        else if(getPickMonth().equals("Apr"))
            monthe = 4;

        else if(getPickMonth().equals("May"))
            monthe = 5;

        else if(getPickMonth().equals("Jun"))
            monthe = 6;

        else if(getPickMonth().equals("Jul"))
            monthe = 7;

        else if(getPickMonth().equals("Aug"))
            monthe = 8;

        else if(getPickMonth().equals("Sep"))
            monthe = 9;

        else if(getPickMonth().equals("Oct"))
            monthe = 10;

        else if(getPickMonth().equals("Nov"))
            monthe = 11;

        else if(getPickMonth().equals("Dec"))
            monthe = 12;

        return monthe;
    }

    public String getReturnMonth() {
        return returnMonth;
    }

    public int getReturnDate() {
        return returnDate;
    }

    public int getReturnYear() {
        return returnYear;
    }

    ///////////////////////////////////////////getting the value of return
    month////////////////////////////////////
    public int getReturnMonthValue() {
        if(getReturnMonth().equals("Jan"))
            monthe = 1;

```

```
        else if(getReturnMonth().equals("Feb"))
            monthe = 2;

        else if(getReturnMonth().equals("Mar"))
            monthe = 3;

        else if(getReturnMonth().equals("Apr"))
            monthe = 4;

        else if(getReturnMonth().equals("May"))
            monthe = 5;

        else if(getReturnMonth().equals("Jun"))
            monthe = 6;

        else if(getReturnMonth().equals("Jul"))
            monthe = 7;

        else if(getReturnMonth().equals("Aug"))
            monthe = 8;

        else if(getReturnMonth().equals("Sep"))
            monthe = 9;

        else if(getReturnMonth().equals("Oct"))
            monthe = 10;

        else if(getReturnMonth().equals("Nov"))
            monthe = 11;

        else if(getReturnMonth().equals("Dec"))
            monthe = 12;

        return monthe;
    }

    public String getBirthMonth() {
        return birthMonth;
    }

    public int getBirthDate() {
        return birthDate;
    }

    public int getBirthYear() {
        return birthYear;
    }
}
```



```

        int age = 0;
        if((getCurrentYear() >
getBirthYear())&&(getCurrentMonthValue() ==
getBirthMonthValue())) {
            if(getCurrentDate() == getBirthDate())
                age = getCurrentYear() - getBirthYear();

            else if(getCurrentDate() < getBirthDate())
                age = getCurrentYear() - getBirthYear();

            else if(getCurrentDate() > getBirthDate())
                age = (getCurrentYear() - 1) - getBirthYear();
        }

        else if((getCurrentYear() >
getBirthYear())&&(getCurrentMonthValue() >
getBirthMonthValue()))
            age = getCurrentYear() - getBirthYear();

        else if((getCurrentYear() >
getBirthYear())&&(getCurrentMonthValue() <
getBirthMonthValue()))
            age = (getCurrentYear() - 1) - getBirthYear();

        return age;
    }

    ////////////////for getting How many days Car has been
    rented////////////////////////////////
    public int getDays(){

        int totalMonth = 0, totalYear = 0;

        if((getPickYear() == getReturnYear()) &&
(getPickMonthValue() == getReturnMonthValue())) {
            if(getPickDate() == getReturnDate())
                days = 0;
            else if(getPickDate() < getReturnDate())
                days = getReturnDate() - getPickDate();
        }

        else if((getPickYear() == getReturnYear()) &&
(getPickMonthValue() < getReturnMonthValue())) {
            if(getPickDate() == getReturnDate()) {
                totalMonth = getReturnMonthValue() -
getPickMonthValue();

```

```

        days = totalMonth * MONTHS_DAYS;
    }
    else if(getPickDate() < getReturnDate()) {
        totalMonth = getReturnMonthValue() -
getPickMonthValue();
        days = (totalMonth * MONTHS_DAYS) +
(getReturnDate() - getPickDate());
    }
}

    else if(getPickYear() < getReturnYear()) {
        if(getPickMonthValue() == getReturnMonthValue()) {
            totalYear = getReturnYear() - getPickYear();
            days = ((totalYear * MONTHS) * MONTHS_DAYS) +
(getReturnDate() - getPickDate());
        }

        else if(getPickMonthValue() < getReturnMonthValue())
{
            totalMonth = getReturnMonthValue() -
getPickMonthValue();
            days = ((totalMonth + MONTHS) * MONTHS_DAYS) +
(getReturnDate() - getPickDate());
        }

        else if(getPickMonthValue() > getReturnMonthValue())
{
            totalMonth = MONTHS - (getPickMonthValue() -
getReturnMonthValue());
            days = (totalMonth * 30) + (getReturnDate() -
getPickDate());
        }
    }

    return days;
}

////////////////////////////////////////to get the total Rental
price////////////////////////////////////////
    public double getRentalPrice() {

        return getDays() * aCar.getCarPrice();
    }

////////////////////////////////////////change to String for
checking the date////////////////////////////////////////
    public String toStringDateChecker() {

```

```

        String output = "";

        if((getPickMonthValue() > getReturnMonthValue()) &&
(getPickYear() == getReturnYear()))
            output += String.format("FALSE");

        if((getPickMonthValue() == getReturnMonthValue()) &&
(getPickDate() > getReturnDate()) && (getPickYear() ==
getReturnYear()))
            output += String.format("FALSE");

        if(getPickYear() > getReturnYear())
            output += String.format("FALSE");

        else
            output += String.format("TRUE");

        return output;
    }

    ////////////////////////////////////////////change to String for
checking the age of customer////////////////////////////////////
    public String toStringAgeChecker() {
        String output = "";

        if(getAge() >= 20)
            output += String.format("TRUE");

        else if(getAge() < 20)
            output += String.format("FALSE");

        return output;
    }

    ////////////////////////////////////////////(int)age to (String)age to print
in GUI////////////////////////////////////
    public String toStringAge() {
        String output = "";
        output += String.format("%d", getAge());
        return output;
    }

    ////////////////////////////////////////////Status and Gender cheking ,
"mr/mrs"////////////////////////////////////
    public String toStringMrMrs() {
        String output = "";

```

```

        if(aCustomer.getGender().equals("M") ||
aCustomer.getGender().equals("m"))
            outfowt += String.format("Mr.");

        else if((aCustomer.getStatus().equals("S") ||
aCustomer.getStatus().equals("s")) &&
aCustomer.getGender().equals("F") ||
aCustomer.getGender().equals("f"))
            outfowt += String.format("Ms.");

        else if((aCustomer.getStatus().equals("Ma") ||
aCustomer.getStatus().equals("ma")) &&
aCustomer.getGender().equals("F") ||
aCustomer.getGender().equals("f"))
            outfowt += String.format("Mrs.");

        return outfowt;
    }

//////////////////////////////////////////show all for
customer//////////////////////////////////////////
    public String toStringReceipt() {
        String outMe = "";

        //outMe += String.format(toStringMrMrs() +
aCustomer.getFullName());
        outMe += String.format("
MyCar Rental\n");
        outMe += String.format("
Owned & Operated By:\n");
        outMe += String.format("
Ms.Frances\n");
        outMe += String.format("
RENTAL TRANSACTION\n");
        outMe += String.format("-----
-----\n");
        outMe += String.format("Car
%s\n", aCar.getCar());
        outMe += String.format("Rental Days
%d\n", getDays());
        outMe += String.format("Rental Price
%.2f\n", aCar.getCarPrice());
        outMe += String.format("
-----\n");
        outMe += String.format("Total Payment
%.2f\n", getRentalPrice());
    }

```

```

        outMe += String.format(toStringMrMrs() +
aCustomer.getFullName());
        return outMe;
    }

    ////////////////////////////////////////////save all data to
    .txt file for the Owner////////////////////////////////////
    public void saveToFile() throws IOException {

        PrintWriter outFile = new PrintWriter(new
FileOutputStream(new File("Transactions_Inventory.txt"),
true));

        outFile.println("Current Date: " + getCurrentMonth() +
" " + getCurrentDate() + ", " + getCurrentYear());
        outFile.println(" ");
        outFile.println("Pick up Date: " + getPickMonth() + " "
+ getPickDate() + ", " + getPickYear());
        outFile.println("Return Date: " + getReturnMonth() + " "
+ getReturnDate() + ", " + getReturnYear() + " = Days: " +
getDays());
        outFile.println("Full Name: " +
aCustomer.getFullName());
        outFile.println("Age: " + getAge());
        outFile.println("Location: " + aCustomer.getLocation());
        outFile.println("Contact: " + aCustomer.getContact());
        outFile.println("Car: " + aCar.getCar());
        outFile.printf("Total Rent Price: %.2f", +
getRentalPrice());
        outFile.println(" ");
        outFile.println("-----
-----");
        outFile.close();
    }
}

```

WarningCar.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class WarningCar extends JFrame{

    private JPanel warningP;
    private JLabel textL, iconL;

    private Transaction aTrans = new Transaction(); //access my
class Transaction as global declaration

    public WarningCar(){

        Container pane = getContentPane();
        warningP = new JPanel(null);
        warningP.setBackground(Color.WHITE);

        ImageIcon warningImage = new
ImageIcon("../images/warning_icon.GIF");
        iconL = new JLabel(warningImage);
        iconL.setBounds(20, 35, 50, 50);
        warningP.add(iconL);

        textL = new JLabel("");
        textL.setText("Click a Car");
        textL.setFont(new Font("Serif", Font.BOLD, 15));
        textL.setForeground(Color.RED);
        textL.setBounds(80, 48, 200, 20);
        warningP.add(textL);
        pane.add(warningP);
    }
}
```

WarningCustomer.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class WarningCustomer extends JFrame{

    private JPanel warningP;
    private JLabel textL, iconL;

    private Transaction aTrans = new Transaction(); //access my
class Transaction as global declaration

    public WarningCustomer(){

        Container pane = getContentPane();
        warningP = new JPanel(null);
        warningP.setBackground(Color.WHITE);

        ImageIcon warningImage = new
ImageIcon("../images/warning_icon.GIF");
        iconL = new JLabel(warningImage);
        iconL.setBounds(20, 35, 50, 50);
        warningP.add(iconL);

        textL = new JLabel("");
        textL.setText("Please complete all details");
        textL.setFont(new Font("Serif", Font.BOLD, 15));
        textL.setForeground(Color.RED);
        textL.setBounds(80, 48, 200, 20);
        warningP.add(textL);
        pane.add(warningP);
    }
}
```


WarningReserve.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class WarningReserve extends JFrame{

    private JPanel warningP;
    private JLabel textL, iconL;

    private Transaction aTrans = new Transaction(); //access my
class Transaction as global declaration

    public WarningReserve(){

        Container pane = getContentPane();
        warningP = new JPanel(null);
        warningP.setBackground(Color.WHITE);

        ImageIcon warningImage = new
ImageIcon("../images/warning_icon.GIF");
        iconL = new JLabel(warningImage);
        iconL.setBounds(20, 35, 50, 50);
        warningP.add(iconL);

        textL = new JLabel("");
        textL.setText("Oops invalid");
        textL.setFont(new Font("Serif", Font.BOLD, 15));
        textL.setForeground(Color.RED);
        textL.setBounds(80, 48, 100, 20);
        warningP.add(textL);
        pane.add(warningP);
    }
}
```

2.0 OBJECT-ORIENTED IMPLEMENTATION

The main features of the system is to set a dates which is user have to pick-up date and return date to rent a car. Birth date also implemented for Legal Transaction Process. Other than that, user have to fill all the Customer's Information before proceed to the next steps which require the system to store data such as name, contact. When the form is not fill correctly it will shows the warning message.

After all the required form is complete the user may proceed to the next steps that is to pick or choose a car. This system is provided the car details and it contain guide for car's usage and features. At the end, this system also provided the transaction information which is receipt. It contains car type rented.

3.0 READ AND WRITE IMPLEMENTATION

Read Implementation

```
public class Customer {

    private String firstName, lastName, gender, status,
location, contact;

    public Customer() {

        firstName = "";
        lastName = "";
        gender = "";
        status = "";
        location = "";
        contact = "";
    }

    public void setCustomer(String fN, String lN, String
g, String s, String l, String c){

        firstName = fN;
        lastName = lN;
        gender = g;
        status = s;
        location = l;
        contact = c;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getFullName() {
        return getFirstName() + " " + getLastName();
    }

    public String getGender() {
        return gender;
    }
}
```

```

    public String getStatus() {
        return status;
    }

    public String getLocation() {
        return location;
    }

    public String getContact() {
        return contact;
    }

    // public void saveToFile() throws IOException {

    //     PrintWriter outFile = new PrintWriter(new
    FileOutputStream(new File("Customers_Inventory.txt"),
    true));
    //     outFile.println(getFullName() + " " + getGender() +
    " "
    //     + getLocation() + " " + getContact() + " " +
    getStatus());
    //     outFile.close();
    // }
}

```

Write Implementation

```

    ////////////////////////////////////////////save
all data to .txt file for the
Owner////////////////////////////////////
    public void saveToFile() throws IOException {

        PrintWriter outFile = new PrintWriter(new
FileOutputStream(new File("Transactions_Inventory.txt"),
true));

        outFile.println("Current Date: " +
getCurrentMonth() + " " + getCurrentDate() + ", " +
getCurrentYear());
        outFile.println(" ");
        outFile.println("Pick up Date: " + getPickMonth() +
" " + getPickDate() + ", " + getPickYear());
        outFile.println("Return Date: " + getReturnMonth()
+ " " + getReturnDate() + ", " + getReturnYear() + " =

```

```

Days: " + getDays());
    outFile.println("Full Name: " +
aCustomer.getFullName());
    outFile.println("Age: " + getAge());
    outFile.println("Location: " +
aCustomer.getLocation());
    outFile.println("Contact: " +
aCustomer.getContact());
    outFile.println("Car: " + aCar.getCar());
    outFile.printf("Total Rent Price: %.2f", +
getRentalPrice());
    outFile.println(" ");
    outFile.println("-----");
    outFile.close();
}
}

//-----TEXT FIELD for
customer details and get data from keyboard section
    firstNameTF = new JTextField("Type First Name
Here"); // set as 10 ha

firstNameTF.setHorizontalAlignment(JTextField.CENTER);
    firstNameTF.setForeground(Color.GRAY);
    firstNameTF.setBounds(150, 60, 250, 30);
    customerP.add(firstNameTF);

    lastNameTF = new JTextField("Type Last Name Here");
// kani pud

lastNameTF.setHorizontalAlignment(JTextField.CENTER);
    lastNameTF.setForeground(Color.GRAY);
    lastNameTF.setBounds(150, 95, 250, 30);
    customerP.add(lastNameTF);

    genderTF = new JTextField("M/F");
    genderTF.setHorizontalAlignment(JTextField.CENTER);
    genderTF.setForeground(Color.GRAY);
    genderTF.setBounds(150, 130, 50, 30);
    customerP.add(genderTF);

    statusTF = new JTextField("S/Ma");
    statusTF.setHorizontalAlignment(JTextField.CENTER);
    statusTF.setForeground(Color.GRAY);
    statusTF.setBounds(150, 165, 50, 30);
    customerP.add(statusTF);

```

```

        locationTF = new JTextField("Type Complete Address
Here");

locationTF.setHorizontalAlignment(JTextField.CENTER);
locationTF.setForeground(Color.GRAY);
locationTF.setBounds(150, 200, 250, 30);
customerP.add(locationTF);

        contactTF = new JTextField("Type Contact Number
Here");

contactTF.setHorizontalAlignment(JTextField.CENTER);
contactTF.setForeground(Color.GRAY);
contactTF.setBounds(150, 235, 250, 30);
customerP.add(contactTF);

//-----checker customer

        firstNameChecker = new JLabel("");
        firstNameChecker.setFont(new Font("Serif",
Font.BOLD, 15));
        firstNameChecker.setBounds(410, 25, 100, 100);
        customerP.add(firstNameChecker);

        lastNameChecker = new JLabel("");
        lastNameChecker.setFont(new Font("Serif",
Font.BOLD, 15));
        lastNameChecker.setBounds(410, 60, 100, 100);
        customerP.add(lastNameChecker);

        genderChecker = new JLabel("");
        genderChecker.setFont(new Font("Serif", Font.BOLD,
15));
        genderChecker.setBounds(210, 95, 100, 100);
        customerP.add(genderChecker);

        statusChecker = new JLabel("");
        statusChecker.setFont(new Font("Serif", Font.BOLD,
15));
        statusChecker.setBounds(210, 130, 100, 100);
        customerP.add(statusChecker);

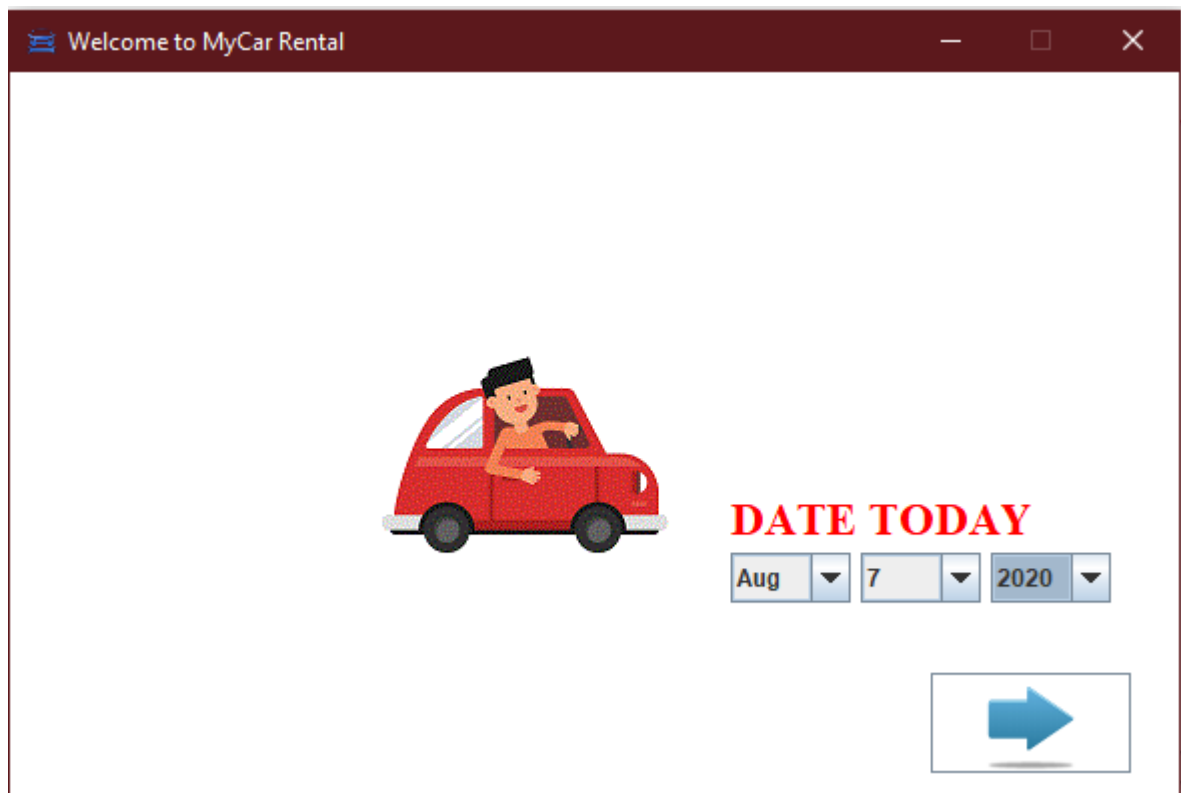
        locationChecker = new JLabel("");
        locationChecker.setFont(new Font("Serif",
Font.BOLD, 15));
        locationChecker.setBounds(410, 165, 100, 100);
        customerP.add(locationChecker);

```

```
contactChecker = new JLabel("");  
contactChecker.setFont(new Font("Serif", Font.BOLD,  
15));  
contactChecker.setBounds(410, 200, 100, 100);  
customerP.add(contactChecker);
```

4.0 USER MANUAL

i. Interface



ii. Select pick-up and return date



Welcome to MyCar Rental

Reserve Rental Day(s):

Pick-up Date: Aug ▼ 11 ▼ 2020 ▼

Return Date: Aug ▼ 13 ▼ 2020 ▼

Birth Date: Jul ▼ 24 ▼ 1998 ▼

iii. Complete the form

Welcome to MyCar Rental

Customer's Data

First Name: Angel

Last Name: Will



Gender: F

Status: S

Location: UMS





Contact: 0109499911

Age: 22

iv. Choose a car



Welcome to MyCar Rental



Car Detail's

Brand: Ford
Plate #: SAB 828 R
Type: Pick-up
Model: 2017
Color: Dirty White
Rental Price: MYR700.00

- Oftenly used for travelers such as adventureseekers. Can hold loads from its back open space. It's tires are good for long travels and bumpy roads.



v. The receipt will be displayed

Welcome to MyCar Rental

MyCar Rental
Owned & Operated By:
Ms.Frances
RENTAL TRANSACTION

Car	Pick-up-2017
Rental Days	2
Rental Price	700.00
Total Payment	1400.00
Ms.Angel Will	

