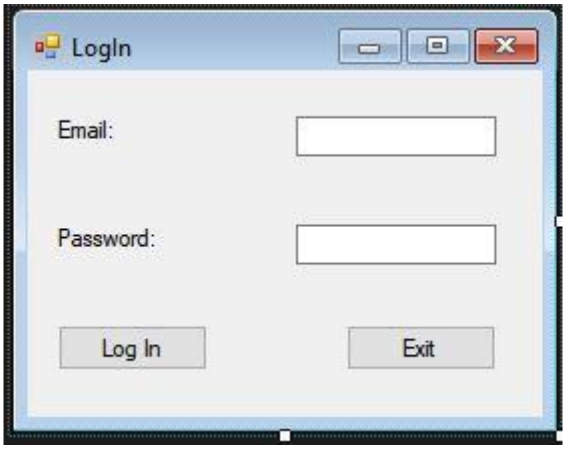
Car Management Codes and Forms

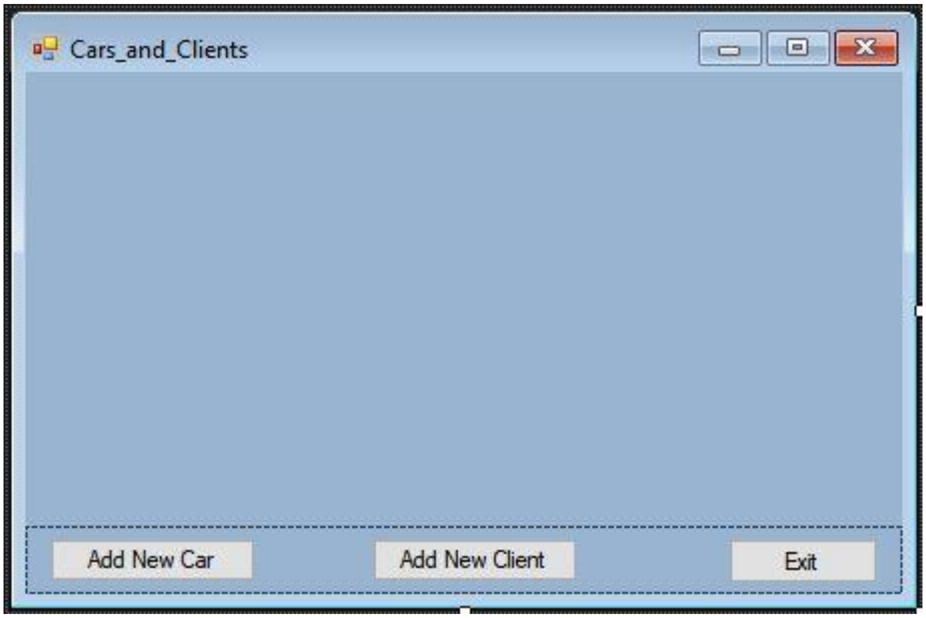
Login Form:



Login Codes:

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Car_Management
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnLogIn_Click(object sender, EventArgs e)
        {
            try
            {
                string connected;
                DatabaseConnection check = new DatabaseConnection();
                connected = check.checkDatabase();
                if (connected == "true")
                {
                    using (SqlConnection conn = new
SqlConnection(DatabaseConnection.connectionStr))
                    {
                        DataTable table = new DataTable();
                        SqlDataAdapter adapter = new SqlDataAdapter(@"select * from
Contacts", conn);

                        adapter.Fill(table);

                        if (table.Rows.Count > 0)
                        {
                            Global.Client = new Cars
                            {
                                serialNumber = table.Rows[0]["SerialNumber"].ToString(),
                                Names = table.Rows[0]["Names"].ToString(),
                                plateNumber = table.Rows[0]["NumberPlates"].ToString(),
                                phoneNumber = table.Rows[0]["PhoneNumber"].ToString(),
                                address = table.Rows[0]["Address"].ToString(),
                                clientID = table.Rows[0]["ID"].ToString(),
                                clientAccount = table.Rows[0]["Account"].ToString(),
                                Technician = table.Rows[0]["Technician"].ToString(),
                            };
                        }
                        DataTable table1 = new DataTable();
                        SqlDataAdapter adapter1 = new SqlDataAdapter(@"select * from 
Technicians where Email = '" + txtEmail.Text + "' and Password = '" + txtPassword.Text + 
"'", conn);
                        adapter1.Fill(table1);
                        if (table.Rows.Count > 0)
                        {
                            Global.Technician = new Technicians
                            {
                                ID = table1.Rows[0]["ID"].ToString(),
                                Email = table1.Rows[0]["Email"].ToString(),
                                Password = table1.Rows[0]["Password"].ToString(),
                                Role = table1.Rows[0]["Role"].ToString(),
                                Names = table1.Rows[0]["Name"].ToString(),
                                Surnames = table1.Rows[0]["Surname"].ToString(),
                                //Position = table.Rows[0]["Position"].ToString(),
                            };
                            if (Global.Technician.Role == "Admin")
                            {
                                //Cars_and_Clients frmCar = new Cars_and_Clients();
                                //frmCar.Show();
                                ListviewTest frmlst = new ListviewTest();
                                frmlst.Show();
                                this.Hide();
                            }
                            else
                            {
                                MessageBox.Show("Normal use connected!");
                            }
                        }
                        DataTable table2 = new DataTable();
                        SqlDataAdapter adapter2 = new SqlDataAdapter(@"select * from 
Users", conn);
                        adapter2.Fill(table2);

                        if (table2.Rows.Count > 0)
                        {
                            Global.User = new Users
                            {
                                Names = table2.Rows[0]["Name"].ToString(),
                                Surnames = table2.Rows[0]["Surname"].ToString(),
                                Email = table2.Rows[0]["Email"].ToString(),
                                ID = table2.Rows[0]["ID"].ToString(),
                                phoneNumber = table2.Rows[0]["PhoneNumber"].ToString(),
                                Technician = table.Rows[0]["Technician"].ToString(),
                            };
                        }
                    }
                }
                else
                {
                    MessageBox.Show("Connection to the database was not established.", 
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("No user exist with those credentials.Please try again!" 
+ ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            try
            {
                string connected;
                DatabaseConnection check = new DatabaseConnection();
                connected = check.checkDatabase();
                if (connected == "true")
                {
                    using (SqlConnection conn = new 
SqlConnection(DatabaseConnection.connectionStr))
                    {
                        SqlCommand cmd = new SqlCommand("Select Email FROM Technicians", 
conn);
                        conn.Open();
                        SqlDataReader reader = cmd.ExecuteReader();
                        AutoCompleteStringCollection MyCollection = new 
AutoCompleteStringCollection();
                        while (reader.Read())
                        {
                            MyCollection.Add(reader.GetString(0));
                        }
                        txtEmail.AutoCompleteCustomSource = MyCollection;
                        conn.Close();
                    }
                }
                else
                {
                    MessageBox.Show("Connection to the database was not established.", 
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, 
MessageBoxIcon.Error);
            }
        }
    }
}
```

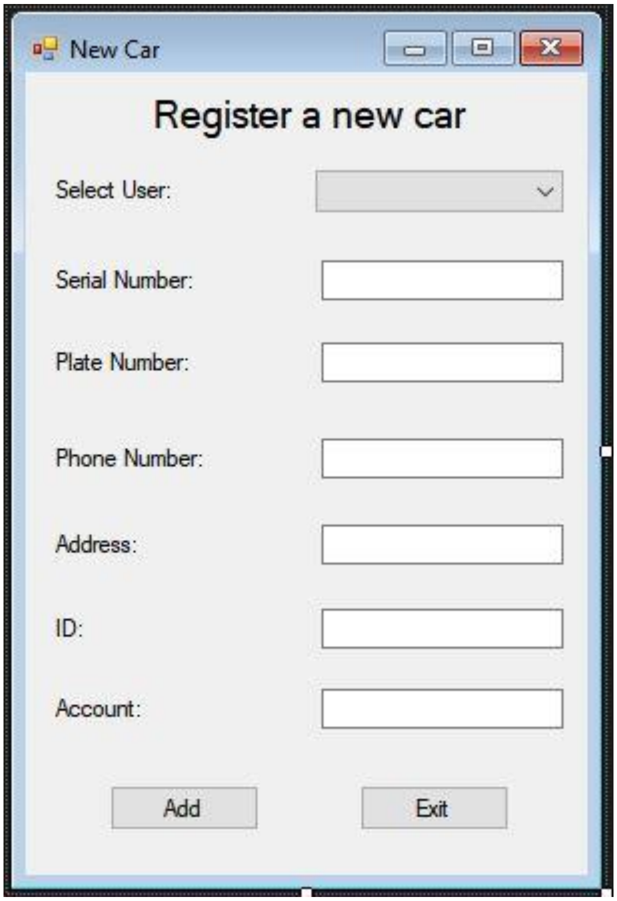
Existing client's car Form:



Codes:

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Car_Management
{
    public partial class Cars_and_Clients : Form
    {
        public Cars_and_Clients()
        {
            InitializeComponent();
        }

        private void Cars_and_Clients_Load(object sender, EventArgs e)
        {
            string connected;
            DatabaseConnection check = new DatabaseConnection();
            connected = check.checkDatabase();
            try
            {
                if (connected == "true")
                {
                    using (SqlConnection conn = new
SqlConnection(DatabaseConnection.connectionStr))
                    {
                        var select = "SELECT * FROM Contacts ";
                        var dataAdapter = new SqlDataAdapter(select, conn);
                        var commandBuilder = new SqlCommandBuilder(dataAdapter);
                        var ds = new DataSet();
                        dataAdapter.Fill(ds);
                        dataGridView1.ReadOnly = true;
                        dataGridView1.DataSource = ds.Tables[0];
                        dataGridView1.DefaultCellStyle.WrapMode =
DataGridViewTriState.True;
                    }
                }
                else
                {
                    MessageBox.Show("Connection to the database was not established.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            MessageBox.Show(Global.Technician.Names + " " + Global.Technician.Surnames);
            New_Car frmNewCar = new New_Car();
            frmNewCar.Show();
            this.Hide();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
            Form1 frm1 = new Form1();
            frm1.Show();
        }

        private void bntNewClient_Click(object sender, EventArgs e)
        {
            New_Client frmNewClient = new New_Client();
            frmNewClient.Show();
            this.Hide();
        }
    }
}
```

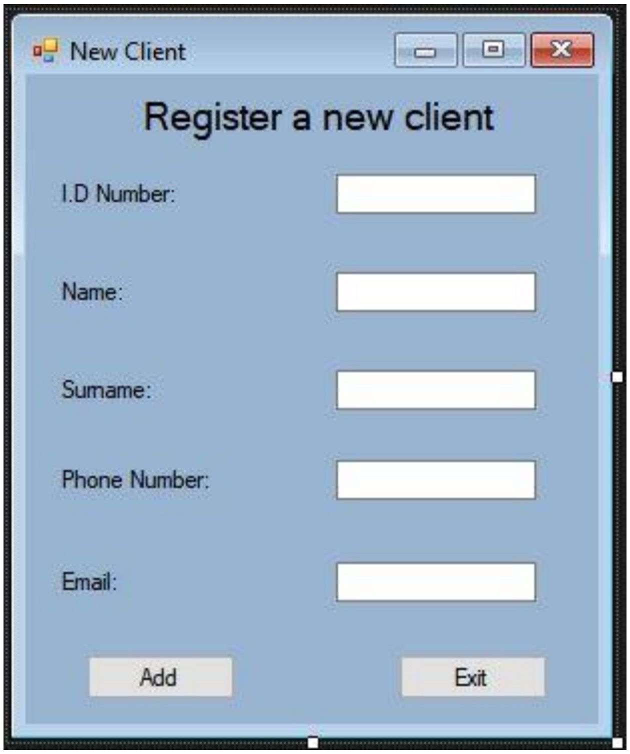
Registering New Car Form:



Codes:

```
using System;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Car_Management
{
    public partial class New_Car : Form
    {
        public New_Car()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                string connected;
                DatabaseConnection check = new DatabaseConnection();
                connected = check.checkDatabase();
                long i;
                if (connected == "true")
                {
                    if (txtSrlNum.Text != "" & txtPltNum.Text != "")
                    {
                        MessageBox.Show((long.TryParse(txtPltNum.Text, out
i)).ToString());
                        string names = comboUsers.Text;
                        long PhoneNum = Convert.ToInt64(txtPhnNum.Text);
                        long srlNum = Convert.ToInt64(txtSrlNum.Text);
                        string plateNum = txtPltNum.Text;
                        string address = txtAddress.Text;
                        long IDnum = Convert.ToInt64(txtID.Text);
                        long account = Convert.ToInt64(txtAcc.Text);
                        string tech = Global.Technician.Names + " " +
Global.Technician.Surnames;
                        using (SqlConnection conn = new
SqlConnection(DatabaseConnection.connectionStr))
                        {
                            conn.Open();
                            string querry = "INSERT INTO
Contacts(Names,PhoneNumber,SerialNumber,NumberPlates,Address,ID,Account,Technician) "
                                           + "Values('" + names + "','" + PhoneNum +
"','" + srlNum + "','" + plateNum + "','" + address + "','" + IDnum + "','" + account +
"','"+ tech+"')";
                            using (SqlCommand cmd = new SqlCommand(querry, conn))
                            {
                                cmd.ExecuteNonQuery();
                                MessageBox.Show("New Car added!");
                                txtSrlNum.Clear();
                                txtPltNum.Clear();
                            }
                            conn.Close();
                        }
                    }
                    else
                    {
                        MessageBox.Show("Please make sure you have entered all the
required information", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Not Connected");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            this.Close();
            Cars_and_Clients frmCars = new Cars_and_Clients();
            frmCars.Show();
        }

        private void New_Car_Load(object sender, EventArgs e)
        {
            try
            {
                string connected;
                DatabaseConnection check = new DatabaseConnection();
                connected = check.checkDatabase();
                if (connected == "true")
                {
                    using (SqlConnection conn = new 
SqlConnection(DatabaseConnection.connectionStr))
                    {
                        SqlCommand cmd = new SqlCommand("Select Name,Surname FROM Users", 
conn);
                        conn.Open();
                        SqlDataReader reader = cmd.ExecuteReader();
                        AutoCompleteStringCollection MyCollection = new 
AutoCompleteStringCollection();
                        while (reader.Read())
                        {
                            MyCollection.Add(reader.GetString(0) +" " + 
reader.GetString(1));
                        }
                        comboUsers.DataSource = MyCollection;
                        conn.Close();
                    }
                }
                else
                {
                    MessageBox.Show("Connection to the database was not established.", 
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, 
MessageBoxIcon.Error);
            }
        }
    }
}
```

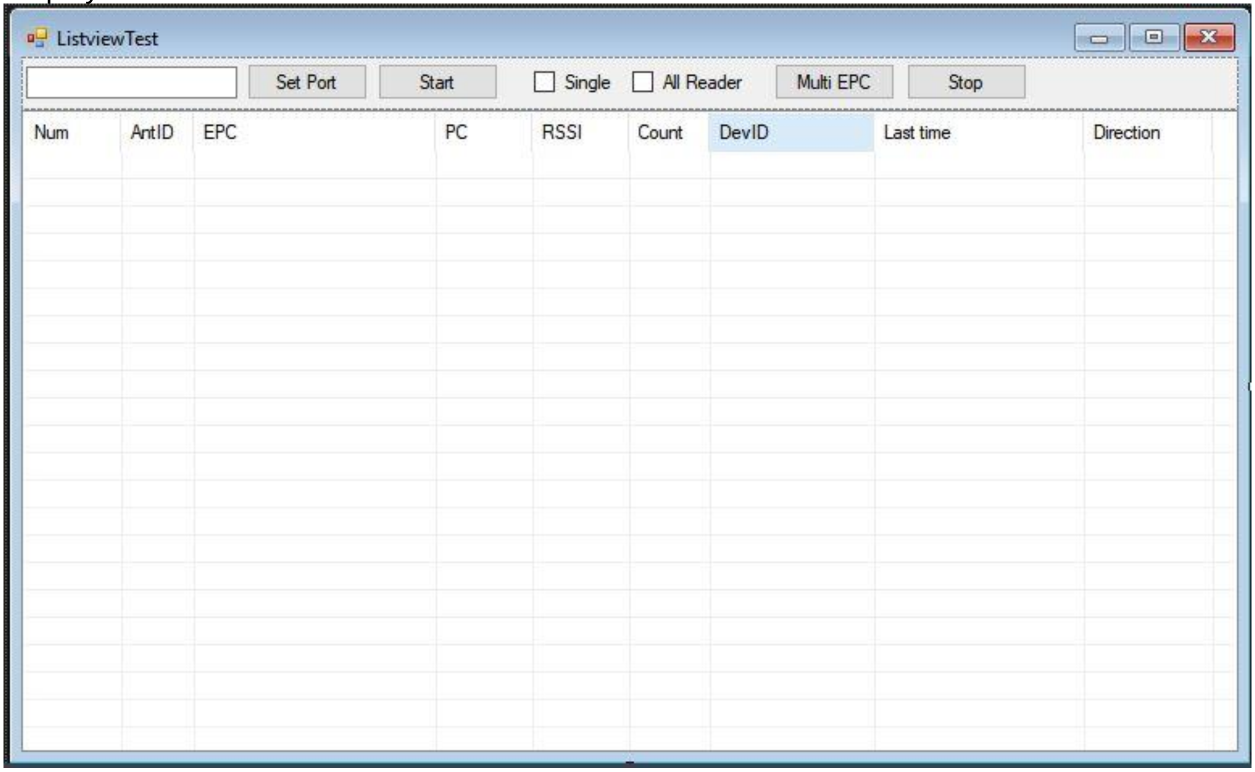
Registering New Client Form:



Codes:

```
using System;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Car_Management
{
    public partial class New_Client : Form
    {
        public New_Client()
        {
            InitializeComponent();
        }

        private void btnExit_Click(object sender, EventArgs e)
        {
            this.Close();
            Cars_and_Clients frmCars = new Cars_and_Clients();
            frmCars.Show();
        }

        private void btnAddClient_Click(object sender, EventArgs e)
        {
            try
            {
                string connected;
                DatabaseConnection check = new DatabaseConnection();
                connected = check.checkDatabase();
                if (connected == "true")
                {
                    if (txtIDNum.Text != "" & txtName.Text != "")
                    {
                        string name = txtName.Text;
                        string surname = txtSurname.Text;
                        long PhoneNum = Convert.ToInt64(txtPhnNum.Text);
                        long IDNum = Convert.ToInt64(txtIDNum.Text);
                        string email = txtEmail.Text;
                        string tech = Global.Technician.Names + " " +
Global.Technician.Surnames;
                        using (SqlConnection conn = new
SqlConnection(DatabaseConnection.connectionStr))
                        {
                            conn.Open();
                            string querry = "INSERT INTO
Users(ID,Name,Surname,Email,PhoneNumber,Technician) "
                                           + "Values('" + IDNum + "','" + name + "','" +
surname + "','" + email + "','" + PhoneNum + "','" + tech + "')";
                            using (SqlCommand cmd = new SqlCommand(querry, conn))
                            {
                                cmd.ExecuteNonQuery();
                                MessageBox.Show("New Client added!");
                            }
                            conn.Close();
                        }
                    }
                    else
                    {
                        MessageBox.Show("Please make sure you have entered all the
required information", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                    }
                }
                else
                {
                    MessageBox.Show("Not Connected");
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void New_Client_Load(object sender, EventArgs e)
        {

        }
    }
}
```

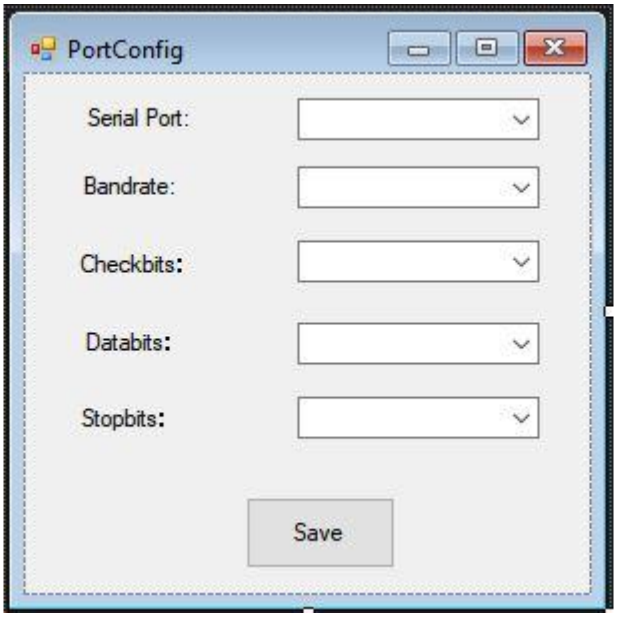
Display data from the device-Form:



Codes:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Runtime.InteropServices;
using System.Threading;             //thread
using System.IO.Ports;              //SerialPort
using System.Text.RegularExpressions;
using System.Net;
using System.Data.SqlClient;
using System.Net.Sockets;
using NetFrame.Net.TCP.Sock.Asynchronous;
using System.Reflection;
using System.Resources;
using System.Globalization;
//using SQL;

namespace Car_Management
{
    public partial class ListviewTest : Form
    {
        public ListviewTest()
        {
            InitializeComponent();
        }
        private long totalnum1 = 0x00;
        private long totalnum2 = 0x00;
        private long totaltime = 0x00;
        private const int listView_md_epc_Num = 0;
        private const int listView_md_epc_AntID = 1;
        private const int listView_md_epc_EPC = 2;
        private const int listView_md_epc_PC = 3;
        private const int listView_md_epc_Rssi = 4;
        private const int listView_md_epc_Count = 5;
        private const int listView_md_epc_IP = 6;
        private const int listView_md_epc_Last_Time = 7;
        private const int listView_md_epc_Direction = 8;
        private volatile List<_epc_t> epcs_list = new List<_epc_t>(1000);

        private string portname = "";
        private int baudRate = 230400;
        private int dataBits = 8;
        private Parity parity = Parity.None;
        private StopBits stopbits = StopBits.One;
        private void btnSet_Click(object sender, EventArgs e)
        {
            PortConfig SerialPortForm = new PortConfig();
            SerialPortForm.ShowDialog();
            if (SerialPortForm.result == true)
            {
                textBox1.Text = SerialPortForm.PortName;
                portname = textBox1.Text;
                baudRate = SerialPortForm.BuadRate;
                dataBits = SerialPortForm.dataBits;
                parity = SerialPortForm.parity;
                stopbits = SerialPortForm.stopbits;
            }
        }
        bool serialisstart = false;
        bool serverisstart = false;

        private Reader ReaderControllor = new Reader();
        private AsyncSocketState currentclient;

        private void btnStartPort_Click(object sender, EventArgs e)
        {
            if (btnStartPort.Text == "Start")
            {
                portname = textBox1.Text;
                try
                {
                    ReaderControllor.ComStart(portname, baudRate, dataBits, parity,
stopbits);
                    if (timer_md_query_Tick.Enabled == false)
                    {
                        timer_md_query_Tick.Enabled = true;
                    }
                }
                catch (Exception ex)
                {
                   // UpdateLog("Error:" + ex.ToString());
                }
                serialisstart = true;
                btnStartPort.Text = "Stop";
                //UpdateLog(openserial + success);
            }
            else
            {
                serialisstart = false;
                ReaderControllor.SerialPortClose();
                if (serverisstart == false && serialisstart == false &&
timer_md_query_Tick.Enabled == true)
                {
                    timer_md_query_Tick.Enabled = false;
                }
                btnStartPort.Text = "Start";
                //UpdateLog(closeserial + success);
            }
        }

        private void btnMultiEPC_Click(object sender, EventArgs e)
        {
            try
            {
                if (checkBoxMulti.Checked == true)
                {
                    if (checkBoxSingle.Checked == true)
                    {
                        ReaderControllor.SingleEPC();
                    }
                    else
                    {
                        ReaderControllor.SatrtMultiEPC();
                    }
                }
                else
                {
                    if (checkBoxSingle.Checked == true)
                    {
                        ReaderControllor.SingleEPC(currentclient);
                    }
                    else
                    {
                        ReaderControllor.SatrtMultiEPC(currentclient);
                    }
                }
                //UpdateLog(start + multiepc + success);
            }
            catch (Exception ex)
            {
                //UpdateLog(ex.ToString());
            }
        }
        private void timer_md_query_Tick_Tick(object sender, EventArgs e)
        {

            totalnum1 = 0;
            totaltime++;
            //label10.Text = totaltime.ToString();
            //epcs_list = ReaderControllor.GetMultiEPC();
            //label26_total.Text = epcs_list.Count.ToString();
            for (int index = 0; index < epcs_list.Count; index++)
            {
                string str_epc = epcs_list[index].epc;
                string str_pc = epcs_list[index].PC.ToString("X2");
                string str_read_cnt = epcs_list[index].count.ToString();
                string str_ant_id = epcs_list[index].antID.ToString();
                string str_dev = epcs_list[index].dev;
                string str_ip = epcs_list[index].ClientIP;
                string str_time = epcs_list[index].time;
                string str_rssi = epcs_list[index].RSSI.ToString("f1");
                string direction = epcs_list[index].direction.ToString();
                totalnum1 += epcs_list[index].count;
                bool Exist = false;
                int item_index = 0;
                foreach (ListViewItem viewitem in this.listView_md_epc.Items)
                {
                    if ((viewitem.SubItems[listView_md_epc_EPC].Text == str_epc) &&
(viewitem.SubItems[listView_md_epc_IP].Text == str_dev))
                    {
                        viewitem.SubItems[listView_md_epc_AntID].Text = str_ant_id;
                        viewitem.SubItems[listView_md_epc_Count].Text = str_read_cnt;
                        viewitem.SubItems[listView_md_epc_Last_Time].Text = str_time;
                        viewitem.SubItems[listView_md_epc_PC].Text = str_pc;
                        viewitem.SubItems[listView_md_epc_Rssi].Text = str_rssi;
                        viewitem.SubItems[listView_md_epc_Direction].Text = direction;
                        Exist = true;
                        break;
                    }
                    item_index++;
                }
                if (!Exist)
                {
                    ListViewItem item = new
ListViewItem((this.listView_md_epc.Items.Count + 1).ToString());
                    item.SubItems.Add(str_ant_id);
                    item.SubItems.Add(str_epc);
                    item.SubItems.Add(str_pc);
                    item.SubItems.Add(str_rssi);
                    item.SubItems.Add(str_read_cnt);
                    item.SubItems.Add(str_dev);
                    item.SubItems.Add(str_time);
                    item.SubItems.Add(direction);
                    this.listView_md_epc.Items.Add(item);
                    this.listView_md_epc.Items[this.listView_md_epc.Items.Count -
1].EnsureVisible();
                    this.listView_md_epc.Items[this.listView_md_epc.Items.Count -
1].Selected = true;
                    // this.listView_md_epc.Items[this.listView_md_epc.Items.Count -
1].BackColor = System.Drawing.Color.FromArgb(red, green, blue);
                }

            }
            //label8.Text = (totalnum1 - totalnum2).ToString();
            totalnum2 = totalnum1;
        }
        private void ListviewTest_Load(object sender, EventArgs e)
        {
        }

        private void btnStop_Click(object sender, EventArgs e)
        {
            try
            {
                if (checkBoxMulti.Checked == true)
                {
                    if (checkBoxSingle.Checked == true)
                    {
                        ;
                    }
                    else
                    {
                        ReaderControllor.StopMultiEPC();
                        //UpdateLog(stop + multiepc + success);
                    }
                }
                else
                {
                    if (checkBoxSingle.Checked == true)
                    {
                        ;
                    }
                    else
                    {
                        ReaderControllor.StopMultiEPC(currentclient);
                        //UpdateLog(stop + multiepc + success);
                    }
                }
            }
            catch (Exception ex)
            {
                //UpdateLog(ex.ToString());
            }
        }
    }
}
```

Port configuration Form:



Codes:

```
using System;
using System.Windows.Forms;
using System.IO.Ports;

namespace Car_Management
{
    public partial class PortConfig: Form
    {
        public string PortName = "";
        public int BuadRate = 0;
        public int dataBits = 0;
        public Parity parity = 0;
        public StopBits stopbits = 0;
        public bool result = false;
        public PortConfig()
        {
            InitializeComponent();
            string[] names = SerialPort.GetPortNames();
            foreach (string name in names)
            {
                comboBox1.Items.Add(name);
            }
            comboBox1.SelectedIndex = 0;
            comboBox2.SelectedIndex = 7;
            comboBox3.SelectedIndex = 0;
            comboBox4.SelectedIndex = 3;
            comboBox5.SelectedIndex = 1;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            PortName = comboBox1.Text;
            BuadRate = int.Parse(comboBox2.Text);
            dataBits = int.Parse(comboBox4.Text);
            parity = (Parity)comboBox3.SelectedIndex;
            stopbits = (StopBits)comboBox5.SelectedIndex;
            result = true;
            this.Close();
        }
    }
}
```