CPSC3131 – Database Design
Mr. Husainy – Group Project Phase 3

PART 1: Provide the DDL script to create the database. (Darius, Glenn)
A-B:

```
create table Location
(
Loc_id number(3) not null,
Loc_Phon varchar2(15) not null,
Loc_Addr varchar2(50) not null,
Loc_Emal varchar2(50) not null,
constraint pk_Location primary key(Loc_id)
);

create table Vehicle
(
Veh_Did number(6) not null,
Veh_Colr varchar2(15) not null,
Veh_Modl varchar2(15) not null,
Veh_Pric decimal(20,2) not null,
Veh_Vin  varchar2(20)not null,
Veh_Make varchar2(20)not null,
Loc_Id number(8)not null,
constraint Veh_pk_Vehicle primary key(Veh_Vin)
);

create table Vendor
(
Ven_Id number(4)not null,
Ven_Phon varchar2(15)not null,
Ven_Ophr varchar2(10)not null,
Ven_addr varchar2(50)not null,
Ven_Emal varchar2(50)not null,
Ven_Cont varchar2(15) not null,
Ven_Date date not null
);

alter table Vendor
add constraint ven_pk_Location primary key(Ven_Id);
```

Part 1 - A-B: Continued

```
create table Employee
(
Emp_Id number(8)not null,
Emp_Lnam varchar2(20)not null,
Emp_Fnam varchar2(10)not null,
Emp_Dob date not null,
Emp_Dept varchar2(20)not null,
Emp_Dl varchar2(3) default 'Yes',
Emp_Gend varchar2(12) default 'Male',
Emp_Ssn number(9) check(Emp_Ssn between 000000000 and 999999999) not null,
Emp_Phon varchar2(15)not null,
Emp_Addr varchar2(50)not null,
Emp_Mana varchar2(3) default 'Yes',
Emp_Stat varchar2(10)default 'Single',
Emp_Emal varchar2(50) not null,
Emp_DEGR varchar(3) default 'Yes',
constraint Emp_pk_Employee primary key(Emp_Id)
);
alter table Vendor modify emp_id number(8);

alter table Vendor
add constraint fk_ven_vendor_id
foreign key(Emp_Id) references Employee(Emp_Id);

create table Customer
(
Cus_Id number(9)not null,
Cus_Lnam varchar2(20)not null,
Cus_Fnam varchar2(10)not null,
Cus_Phon varchar2(15)not null,
Cus_Addr varchar2(50)not null,
Cus_Emal varchar2(50)not null,
Cus_Dob date not null,
Cus_Stat varchar2(10) not null
);
alter table Customer
add Veh_Did number(6)not null;

alter table Customer
add Emp_Id number(8)not null;
```

Part 1 - A-B: Continued

```
alter table Customer
add Veh_vin varchar2(20)not null;

alter table Customer
add Fin_Id number(8) check(Fin_Id between 00000000 and 99999999);

alter table Customer
add constraint fk_Customer_id foreign key(Veh_Id) references Vehicle(Veh_Id);

constraint fk_Employee_id foreign key(Emp_Id) references Employee(Emp_Id),
constraint fk_Vehicle_vin foreign key(Veh_Vin) references Vehicle(Veh_Vin)


create table ServiceCenter
(
Svc_Id number(3)not null,
Svc_Date date not null ,
Svc_Type number(1) not null,
Svc_Phon varchar2(15)not null,
Svc_Prov number(1)
);
alter table ServiceCenter
add constraint  fk_Svc_Location_id foreign key(Loc_Id) references Location(Loc_Id);

create table FinancialServices
(
Fin_Id number(8)not null,
Fin_Date date not null,
Fin_Amnt number(10)check(Fin_Amnt between 0000000000 and 9999999999),
Fin_Plan varchar2(15)not null
);
alter table FinancialServices
add constraint  fk_Fin_Service_Did foreign key(Veh_Did)references Vehicle(Veh_Did);

alter table FinancialServices
add constraint  fk_Fin_Service_id foreign key(Emp_Id)references Employee(Emp_Id);

alter table FinancialServices
add constraint  fk_Cus_id foreign key(Cus_Id)references Customer(Cus_Id);
```

Part 1 - A-B: Continued

```
alter table FinancialServices
add constraint  fk_Veh_Vin foreign key(Veh_Vin)references Vehicle(Veh_Vin);

alter table Vehicle
add constraint Veh_pk_Vin primary key(Veh_Vin);



alter table FinancialServices
add constraint  fk_Veh_Did foreign key(Veh_did)references Vehicle(Veh_Did);

alter table Customer
add constraint pk_Cus_id primary key(Cus_Id);

alter table Customer
Customer modify Cus_id not null;


alter table FinancialServices
add constraint fin_pk_services primary key(Fin_Id);

alter table FinancialServices
add Veh_Did number(6)not null;

alter table FinancialServices
add Emp_Id number(8)not null;

alter table FinancialServices
add Cus_Id number(9)not null;

alter table FinancialServices
add Veh_Vin varchar2(20)not null;


alter table Customer
add constraint fk_Customer_id foreign key(Veh_Id) references Vehicle(Veh_Id);

alter table Customer
constraint fk_Employee_id foreign key(Emp_Id) references Employee(Emp_Id);
constraint fk_Vehicle_vin foreign key(Veh_Vin) references Vehicle(Veh_Vin)
```

Part 1 - A-B: Continued

```
alter table Location
add Svc_Id number(3);
alter table Location
add constraint fk_Svc_id foreign key(Svc_Id) references ServiceCenter(Svc_Id);



select* from Location;
select* from vehicle;

alter table vehicle
add Svc_Id number(3);


alter table vehicle
add Cus_Id number(9)not null;

alter table vehicle
add Ven_Id number(4)not null;
alter table vehicle
add Svc_Type number(1)not null;
alter table vehicle
add Fin_Id number(8);
alter table vehicle
add Emp_Id number(8)not null;

alter table vehicle
add constraint fk_Svc_id foreign key(Svc_Id) references ServiceCenter(Svc_Id);

alter table vehicle
add constraint fk_Cust_id foreign key(Cus_Id) references Customer(Cus_Id);

alter table vehicle
add constraint fk_Ven_id foreign key(Ven_Id) references Vendor(Ven_Id);

SELECT * FROM vehicle;

alter table vehicle
add constraint fk_Fin_id foreign key(Fin_Id) references FinancialServices(Fin_Id);
alter table vehicle
add constraint fk_Emplo_id foreign key(Emp_Id) references Employee(Emp_Id);
select* from Vehicle;
```

Part 1 - A-B: Continued

```
desc from employee;
select* from vendor;

update vendor
set emp_id='12225648'
where ven_id=2500;

select* from vehicle ;
insert into
Vehicle(veh_did,veh_colr,veh_modl,veh_pric,veh_vin,veh_make,loc_id,svc_id,cus_id,ve
n_id,svc_type,fin_id,emp_id)
values (900900,'Gold','HighLander
LE',25999,'11111000002222245','Toyota',111,100,101,2500,1,11112222,'12225648');


select* from financialservices ;
select* from customer ;
desc from financialservices;

insert into
customer(cus_id,cus_lnam,cus_fnam,cus_phon,cus_addr,cus_emal,cus_dob,cus_stat,
Veh_did,emp_id,veh_vin,fin_id)
values (101,'Lewis','Joseph','678-123-2398','2300 boardwallk Ave
Columbus,GA','Lew@whatever.com','10-apr-
2010','M','125643','15325648','22340878512385630','00000000');

insert into
FinancialServices(Fin_Id,Fin_Date,Fin_Amnt,Fin_Plan,Veh_Did,Emp_Id,Cus_Id,veh_vi
n)
values (11112222,'18-mar-16',30000,'10 Year
Loan',900900,'15325648',101,'11111000002222245');


 alter table financialservices
 modify veh_vin varchar2(20) null ;
```

Part 1 - A-B: Continued

```
insert into
customer(cus_id,cus_lnam,cus_fnam,cus_phon,cus_addr,cus_emal,cus_dob,cus_stat,
Veh_did,emp_id,veh_vin,fin_id)
values (101,'Lewis','Joseph','678-123-2398','2300 boardwallk Ave
Columbus,GA','Lew@whatever.com','10-apr-
1980','M','125643','15325648','22340878512385630','00000000');

insert into
customer(cus_id,cus_lnam,cus_fnam,cus_phon,cus_addr,cus_emal,cus_dob,cus_stat,
Veh_did,emp_id,veh_vin,fin_id)
values (201,'Ariza','Alex','678-555-6666','1000 Atlantic Ave Columbus
,Ga','Ariza@whatever.com','5-May-
1968','S','235643','13325648','21340878512385630','11111111');

insert into
employee(emp_id,emp_lnam,emp_fnam,emp_dob,emp_dept,emp_dl,emp_gend,emp_s
sn,emp_phon,emp_addr,emp_mana,emp_stat,emp_emal,emp_degr)
values ('15325648','James','Travis','5-Mar-1970','Financial
Services','yes','Male',123456789,'340-905-1769','1200 Baltic Ave
Columbus,Ga','Yes','M','James@CarDealership.com','Y');

insert into
employee(emp_id,emp_lnam,emp_fnam,emp_dob,emp_dept,emp_dl,emp_gend,emp_s
sn,emp_phon,emp_addr,emp_mana,emp_stat,emp_emal,emp_degr)
values ('13325648','Williams','Jesse','10-June-1972','Service
Center','yes','Female',212345678,'340-111-2222','4520 125th St
Columbus,Ga','Yes','S','Will@CarDealership.com','Y');

insert into
employee(emp_id,emp_lnam,emp_fnam,emp_dob,emp_dept,emp_dl,emp_gend,emp_s
sn,emp_phon,emp_addr,emp_mana,emp_stat,emp_emal,emp_degr)
values ('12225648','Rashad','Adam','1-Jan-1980','Car
Dealer','yes','Male',312345687,'404-222-3333','1120 Wall Dr
Columbus,Ga','No','M','Rashad@CarDealership.com','N');

insert into
FinancialServices(Fin_Id,Fin_Date,Fin_Amnt,Fin_Plan,Veh_Did,Emp_Id,Cus_Id,Veh_Vi
n)
```

Part 1 - A-B: Continued

```
values (11112222,'18-mar-2016',30000,'10 Year
Loan',900900,'15325648',101,'11111000002222245');

insert into location(loc_id,loc_phon,loc_addr,loc_emal,emp_id,svc_id)
values (111,'1800-647-0095','4224 University Ave
Columbus,Ga','Questions@CarDealership.com','12225648',100);

insert into serviceCenter(svc_id,svc_Date,svc_type,svc_phon,svc_prov,emp_id,loc_id)
values (100,'01-jan-2016',1,'1800-111-9999',1,'13325648',111);

insert into
Vehicle(veh_did,veh_colr,veh_modl,veh_pric,veh_vin,veh_make,loc_id,svc_id,cus_id,ve
n_id,svc_type,emp_id)
values (900900,'Gold','HighLander
LE',25999,'11111000002222245','Toyota',111,100,201,2500,1,12225648);

insert into
Vehicle(veh_did,veh_colr,veh_modl,veh_pric,veh_vin,veh_make,loc_id,svc_id,cus_id,ve
n_id,svc_type,emp_id,veh_year)
values
(800800,'White','S550',72999,'22222000002222245','Mercedes',111,100,201,2500,1,12
225648,'2014');
```

PART 2: Program creation (front-end). (Morrison)
A-B:

- Used Visual Studio to create a program that accesses the “CUSTOMER” database and manipulates if allowing for insert, update, delete, and read functions. Provides clients with current customer database and allows client’s to add new customers, update preexisting ones, and delete customers as needed.

PART 3: Provide all the DML and SQL data statements in program. (Morrison)
A-B:

----------ConnectionManager----------

```
using System;
using System.Data.OleDb;

namespace DAL
{
    public class ConnectionManager
    {
        protected OleDbConnection conn;

        public ConnectionManager()
        {
            try
            {
                conn = new OleDbConnection("Data Source= (DESCRIPTION = (ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = lora07-scan.columbusstate.edu)(PORT =
1521)))(CONNECT_DATA =(SERVICE_NAME = acad_taf.columbusstate.edu)));Persist Security
Info=True; User ID=GROUP5_SPRING2016;Password=bgs87hr0;Provider=OraOLEDB.Oracle;");
            }

            catch (Exception e)
            {
                Console.WriteLine(e);
            }
        }

    }
}
```

----------CustomerController----------

```
using System;
using System.Collections.Generic;
using System.Data.OleDb;
using System.Data;

namespace DAL
{
    public class CustomerController : ConnectionManager
    {
        public List<Customer> ReadFromDB()
        {
            List<Customer> customers = new List<Customer>();

            conn.Open();
            OleDbCommand cmd = new OleDbCommand();
            OleDbDataReader reader;

            try
            {
                cmd.CommandType = CommandType.Text;
                cmd.Connection = conn;
                cmd.CommandText = "select * from customer";
```

Part 3 - A-B: Continued (CustomerController)

```
                reader = cmd.ExecuteReader(CommandBehavior.SingleResult);

                while (reader.Read())
                {
                    Customer customer = new Customer();

                    customer.ID = Convert.ToInt64(reader["CUS_ID"]);
                    customer.lastName = Convert.ToString(reader["CUS_LNAM"]);
                    customer.firstName = Convert.ToString(reader["CUS_FNAM"]);
                    customer.phone = Convert.ToString(reader["CUS_PHON"]);
                    customer.address = Convert.ToString(reader["CUS_ADDR"]);
                    customer.email = Convert.ToString(reader["CUS_EMAL"]);
                    customer.dob = Convert.ToDateTime(reader["CUS_DOB"]);
                    customer.status = Convert.ToString(reader["CUS_STAT"]);
                    customer.dealerID = Convert.ToInt64(reader["VEH_DID"]);
                    customer.empID = Convert.ToInt64(reader["EMP_ID"]);
                    customer.vehicleVIN = Convert.ToString(reader["VEH_VIN"]);
                    customer.financeID = Convert.ToInt64(reader["FIN_ID"]);

                    customers.Add(customer);
                }

            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
            finally
            {
                conn.Close();
            }

            return customers;

        }

        public void WriteIntoDB(Customer customer)
        {
            conn.Open();
            OleDbCommand cmd = new OleDbCommand();

            try
            {
                cmd.CommandType = CommandType.Text;
                cmd.Connection = conn;
                cmd.CommandText = "insert into customer (cus_id, cus_lnam, cus_fnam,
cus_phon, cus_addr, cus_emal, cus_dob, cus_stat, veh_did, emp_id, veh_vin, fin_id) values
(:cus_id, :cus_lnam, :cus_fnam, :cus_phon, :cus_addr, :cus_emal, :cus_dob, :cus_stat,
:veh_did, :emp_id, :veh_vin, :fin_id)";
                cmd.Parameters.AddWithValue("301", customer.ID);
                cmd.Parameters.AddWithValue("Gateson", customer.lastName);
                cmd.Parameters.AddWithValue("Bill", customer.firstName);
                cmd.Parameters.AddWithValue("404-111-0000", customer.phone);
                cmd.Parameters.AddWithValue("5500 Baltic Ave Columbus,GA",
customer.address);
```

Part 3 - A-B: Continued (CustomerController)

```
            cmd.Parameters.AddWithValue("BillyG@whatever.com", customer.email);
            cmd.Parameters.AddWithValue("20-Dec-1955", customer.dob);
            cmd.Parameters.AddWithValue("M", customer.status);
            cmd.Parameters.AddWithValue("125643", customer.dealerID);
            cmd.Parameters.AddWithValue("12225648", customer.empID);
            cmd.Parameters.AddWithValue("22340878512385630", customer.vehicleVIN);
            cmd.Parameters.AddWithValue("12-DEC-16", customer.financeID);

            cmd.ExecuteNonQuery();


        }
        catch (Exception e)
        {
            Console.WriteLine(e);
        }
        finally
        {
            conn.Close();
        }
    }

    public void DeleteFromDB(Customer customer)
    {
        conn.Open();
        OleDbCommand cmd = new OleDbCommand();

        try
        {
            cmd.CommandType = CommandType.Text;
            cmd.Connection = conn;
            cmd.CommandText = "delete from customer where cus_id = :cus_id";
            cmd.Parameters.AddWithValue("301", customer.ID);

            cmd.ExecuteNonQuery();


        }
        catch (Exception e)
        {
            Console.WriteLine(e);
        }
        finally
        {
            conn.Close();
        }
    }

    public void UpdateAddressDB(Customer customer)
    {
        conn.Open();
        OleDbCommand cmd = new OleDbCommand();

        try
        {
```

Part 3 - A-B: Continued (CustomerController)

```
                cmd.CommandType = CommandType.Text;
                cmd.Connection = conn;
                cmd.CommandText = "update customer set cus_addr = :cus_address where
cus_id = :cus_id ";
                cmd.Parameters.AddWithValue("4225 University Ave.", customer.address);
                cmd.Parameters.AddWithValue("301", customer.ID);

                cmd.ExecuteNonQuery();


            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
            finally
            {
                conn.Close();
            }
        }

        public void UpdatePhoneDB(Customer customer)
        {
            conn.Open();
            OleDbCommand cmd = new OleDbCommand();

            try
            {
                cmd.CommandType = CommandType.Text;
                cmd.Connection = conn;
                cmd.CommandText = "update customer set cus_phon = :cus_phone where cus_id
= :cus_id ";
                cmd.Parameters.AddWithValue("333-555-7777", customer.phone);
                cmd.Parameters.AddWithValue("301", customer.ID);

                cmd.ExecuteNonQuery();


            }
            catch (Exception e)
            {
                Console.WriteLine(e);
            }
            finally
            {
                conn.Close();
            }
        }
    }
}
```

Part 3 - A-B: Continued

----------Customer----------

```
using System;
namespace DAL
{
    public class Customer
    {
        public Int64 ID { get; set; }
        public String lastName { get; set; }
        public String firstName { get; set; }
        public String phone { get; set; }
        public String address { get; set; }
        public String email { get; set; }
        public DateTime dob { get; set; }
        public String status { get; set; }
        public Int64 dealerID { get; set; }
        public Int64 empID { get; set; }
        public String vehicleVIN { get; set; }
        public Int64 financeID { get; set; }
    }
}
```

----------Form----------

```
using System;
using System.Windows.Forms;
using DAL;
namespace TesterApp
{
    public partial class customerServices : Form
    {
        public customerServices()
        {
            InitializeComponent();
        }

        private void btnSubmit_Click(object sender, EventArgs e)
        {
            Customer _customer = new Customer();
            _customer.ID = Convert.ToInt64(customerID.Text);
            _customer.lastName = Convert.ToString(lastName.Text);
            _customer.firstName = Convert.ToString(firstName.Text);
            _customer.phone = Convert.ToString(phone.Text);
            _customer.address = Convert.ToString(address.Text);
            _customer.email = Convert.ToString(email.Text);
            _customer.dob = Convert.ToDateTime(dob.Text);
            _customer.status = Convert.ToString(status.Text);
            _customer.dealerID = Convert.ToInt64(vehicleDID.Text);
            _customer.empID = Convert.ToInt64(empID.Text);
            _customer.vehicleVIN = Convert.ToString(VIN.Text);
            _customer.financeID = Convert.ToInt64(financeID.Text);

            CustomerController _customerController = new CustomerController();
            _customerController.WriteIntoDB(_customer);

            customerID.Text = "";
            lastName.Text = "";
```

Part 3 - A-B: Continued (Form)

```
            firstName.Text = "";
            phone.Text = "";
            address.Text = "";
            email.Text = "";
            dob.Text = "";
            status.Text = "";
            vehicleDID.Text = "";
            empID.Text = "";
            VIN.Text = "";
            financeID.Text = "";

            LoadData();

        }

        private void btnLoad_Click(object sender, EventArgs e)
        {
            LoadData();
        }

        private void LoadData()
        {
            CustomerController _customerController = new CustomerController();
            grdEmployee.DataSource = _customerController.ReadFromDB();
        }

        private void deleteCustomer_Click(object sender, EventArgs e)
        {
            Customer _customer = new Customer();
            _customer.ID = Convert.ToInt64(deleteCusID.Text);

            CustomerController _customerController = new CustomerController();
            _customerController.DeleteFromDB(_customer);

            deleteCusID.Text = "";

            LoadData();
        }

        private void btnUpdateAddress_Click(object sender, EventArgs e)
        {
            Customer _customer = new Customer();
            _customer.ID = Convert.ToInt64(cusIDupdateAddress.Text);
            _customer.address = Convert.ToString(updateAddress.Text);

            CustomerController _customerController = new CustomerController();
            _customerController.UpdateAddressDB(_customer);

            cusIDupdateAddress.Text = "";
            updateAddress.Text = "";

            LoadData();
        }

        private void btnUpdatePhone_Click(object sender, EventArgs e)
```

Part 3 - A-B: Continued (Form)

```
        {
            Customer _customer = new Customer();
            _customer.ID = Convert.ToInt64(cusIDupdatePhone.Text);
            _customer.phone = Convert.ToString(updatePhone.Text);

            CustomerController _customerController = new CustomerController();
            _customerController.UpdatePhoneDB(_customer);

            cusIDupdatePhone.Text = "";
            updatePhone.Text = "";

            LoadData();
        }
    }
}
```

----------End of Code----------

PART 4: Presentation. (Carr, Glenn, Little, Morrison, Talley, Upchurch)

A: Introduction of Members (6)

- Justin Carr, Talmage Glenn, Darius Little, Nate Morrison, Miles Talley, Michael Upchurch

B: Concept / Business Logic of Program

- Employed by car dealerships
- Offers data management of key dealership aspects
  - +Customers, Employees, Vehicles, Vendors, Locations, Service Centers & Finance
- Final concept opens with login page, opens to 1 of 4 user interfaces (Finance, Sales, Service, Management)
- Each interface allows department-specific employees to access, edit and input important information
- We split up the work amongst group members
  - +Justin – ER Diagrams
  - +Talmage – Tables, SQL, Crow's Foot
  - +Darius – Tables, Most SQL, Crow's Foot
  - +Nate – Project management, Visual Studio, ERD
  - +Miles – Tables
  - +Michael – Tables
- -DEMO
  - +Viewing the information
  - +Update information
  - + Create a new customer
  - +Delete customer

PART 5: How to use Pro-Dealer Software. (Morrison)

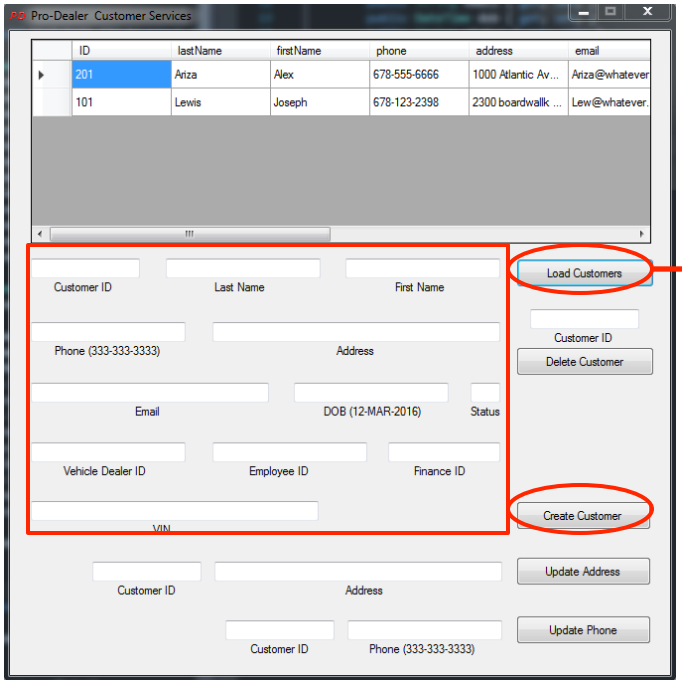
-Run program in Visual Studio (with connectivity to database network)

+User and password saved in code.

-To add a customer:

+Fill out all the fields in this highlighted area:

+Once information fields are populated click the “Create Customer” button.

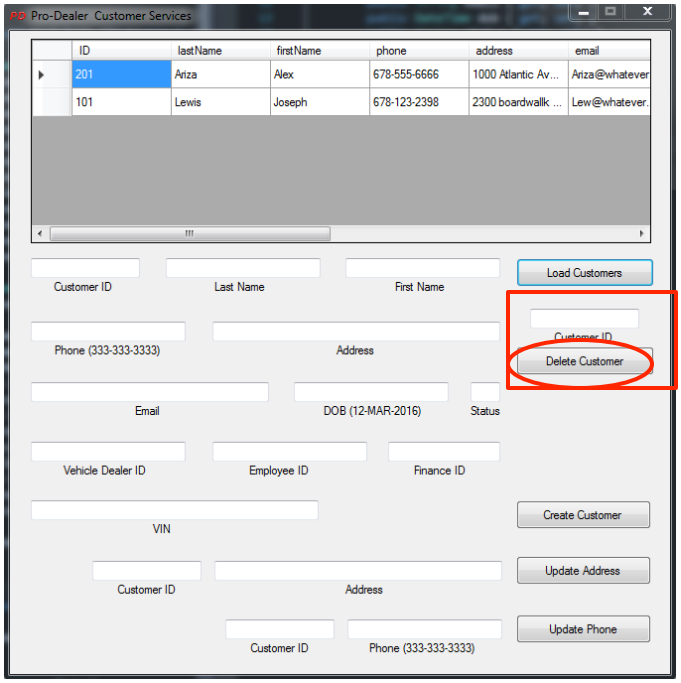


To view data simply click the “Load Customers” button.

-To Delete a Customer:

+Enter the Customer ID in the highlighted section below:

+Once Id has been entered, click “Delete Customer”



-To update the address or phone number:

+ Enter the customer ID into the appropriate field next the address or phone space, respectively, in the highlighted area:

+ Then click either “Update Address” or “Update Phone.” Only one can be updated at a time. If you wish to update both values you must do so in separate commands.