1. INTRODUCTION

The project RTO MANAGEMENT SYSTEM is proposed to install a system that shall enable the proposed RTO SYSTEM interface with the existing system with maximum accuracy. This project is very useful for those who are concern with the different processes on RTO. During study phase, we have done a preliminary analysis sufficient depth to permit a technical and economic evaluation of proposed system.

As a total manpower-based system is currently running for the whole procedures, designing a new system which makes the process tranquil, demands a deep knowledge about the existing system. Throughout the project we focus on presenting information and commands in an easy and intelligible manner.

The purpose of our RTO management system is to provide a leading technological tool for the ease of RTO functions such as Registration, Driver's License and Violations etc... It will reduce considerably the difficulties faced on existing system, with minimum error and difficulties.

1.2 PROBLEM STATEMENT

To design and develop a **RTO Management System**, where the RTO officer can login by providing essential details require for verification. The system analyses the data entered and based on that data, it provides the RTO officer with a options to view. The officer can search or insert new record.

The various relations or tables are depicted in relational mapping. The ER diagram is object centred whereas relational mapping sheds light on tables or the relations. To bring out the mapping, the strong entities of the ER diagram are converted into individual tables in the mapping.

1.1.1 EXISTING SYSTEM:

The effectiveness of the system depends on the way in which the data is organized. In the existing system, RTO officer cannot search a record easily and it can be very time consuming. When records are accessed frequently, managing such records becomes difficult. Therefore, organizing data becomes difficult.

The major limitations are:

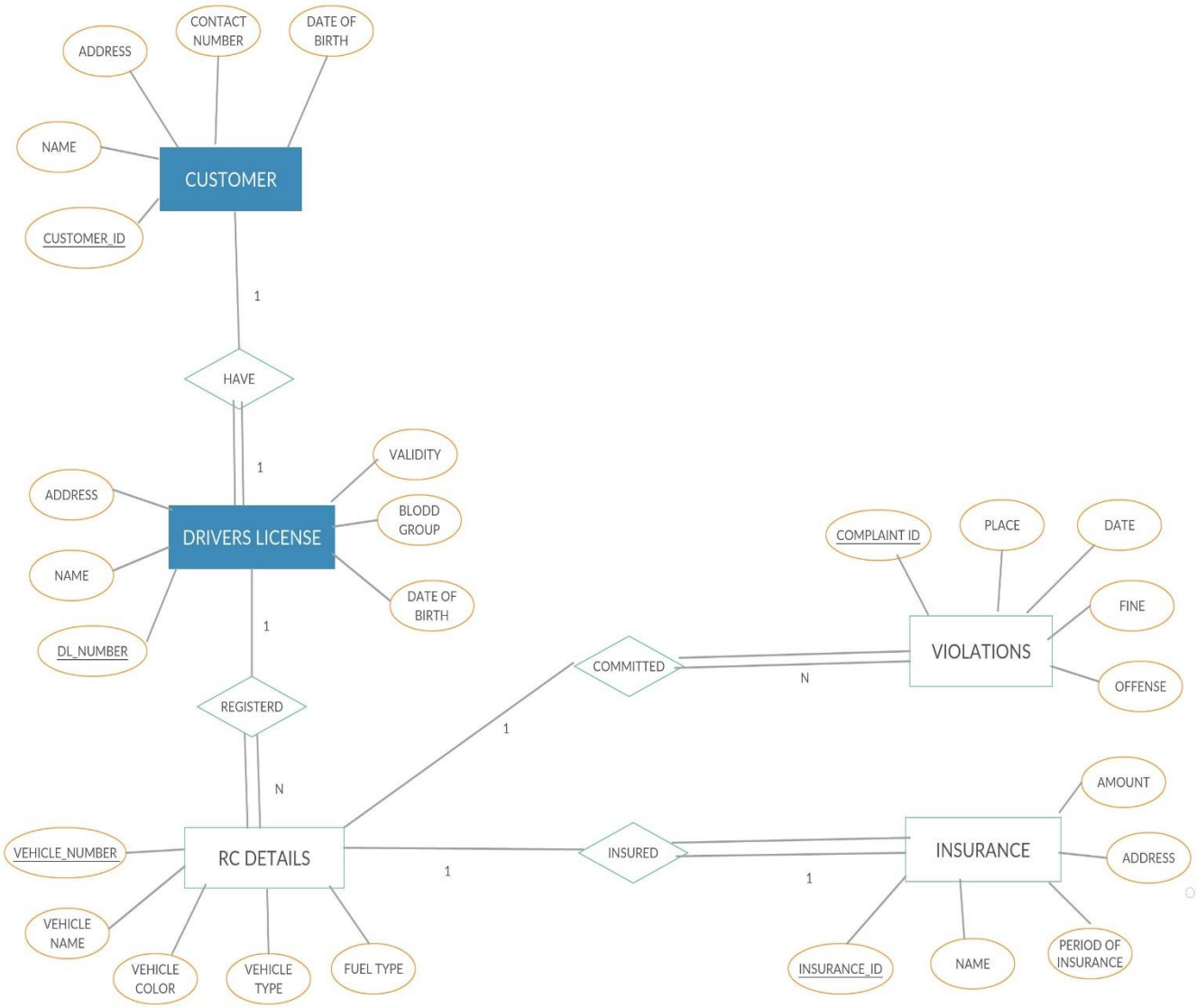
- Modifications to Applications are complicated.
- Much time consuming.
- Error prone.

1.1.2 PROPOSED SYSTEM:

The proposed system is better and more efficient than existing system by keeping in mind all the drawbacks of the present system to provide a permanent to them. The primary aim of the new system is to speed up the transactions. User friendliness is another peculiarity of the proposed system. Messages are displayed in message boxes to make the system user friendly. The main advantage of the proposed system is the reduction in the labour as it will be possible to access from anywhere. Every record is checked for completeness and accuracy and then it is entered into the database. Another important feature is the data security provided by the system. The main objectives of the proposed system are:

- Complex functions are done automatically, such as searching facilities
- Processing time can be minimized
- Simple and easy to manage
- Chances of errors reduced
- Faster and more accurate than the existing system
- Easy for handling reports

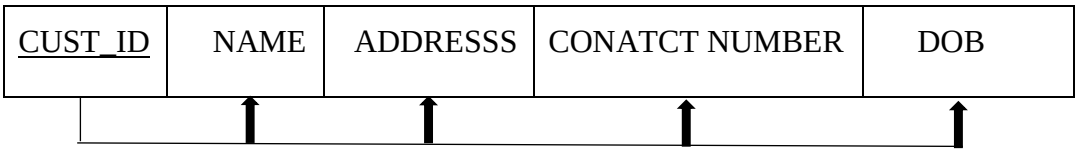
2.1 CONCEPTUAL DATABASE DESIGN:



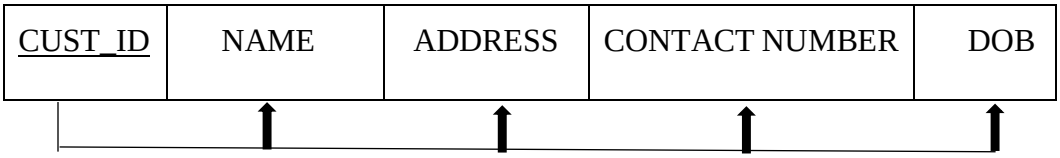
2.2 LOGICAL DATABASE DESIGN:

2.3 NORMALIZATION:

CUSTOMER

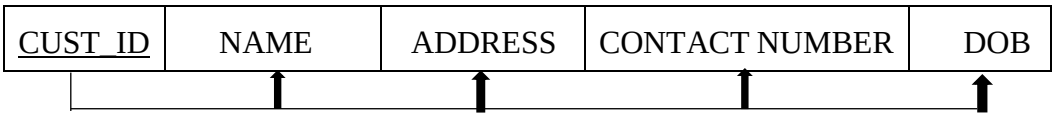


1st -NORMALIZATION_FORM



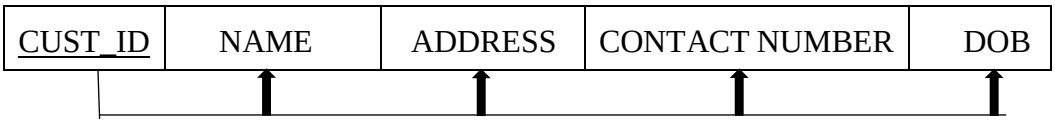
Relation is already in 1NF because it has no multivalued attributes or nested relations.

2nd NORMALIZATION FORM



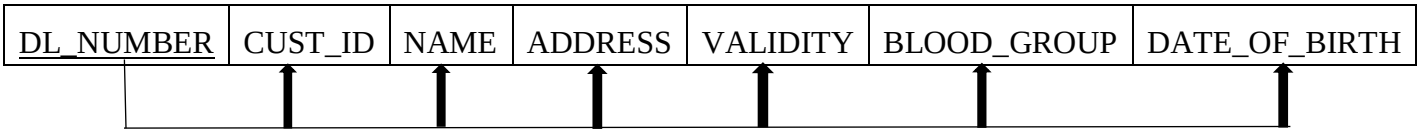
Relation is already in 2NF since the primary keys does not contain multiple attributes.

3rd NORMALIZATION FORM

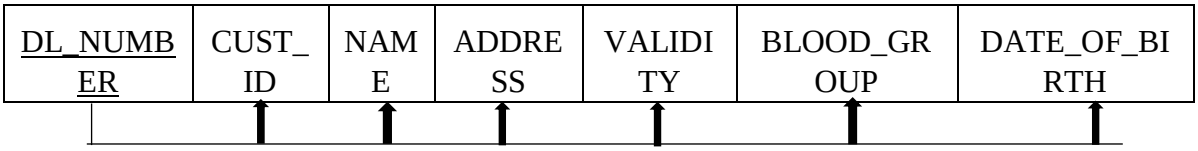


Relation is already is already in 3NF no non-key attribute is dependent on the key attribute via transitive dependency.

DRIVER'S LICENSE

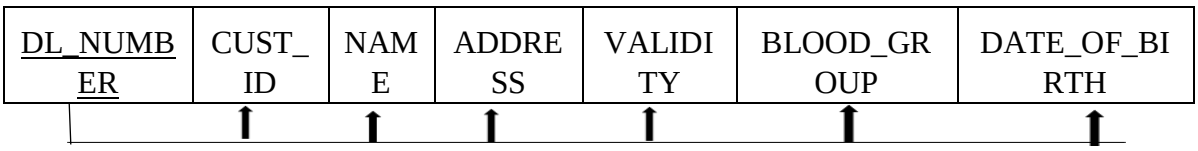


1st NORMALIZATION FORM



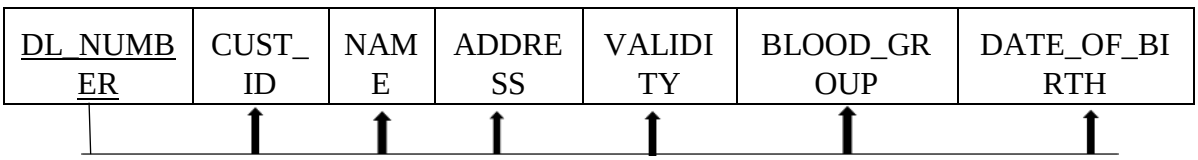
Relation is already in 1NF because it has no multivalued attributes or nested relations.

2nd NORMALIZATION FORM



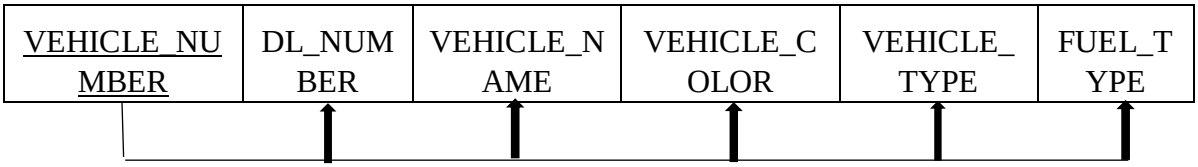
Relation is already in 2NF since the primary keys does not contain multiple attributes.

3rd NORMALIZATION FORM

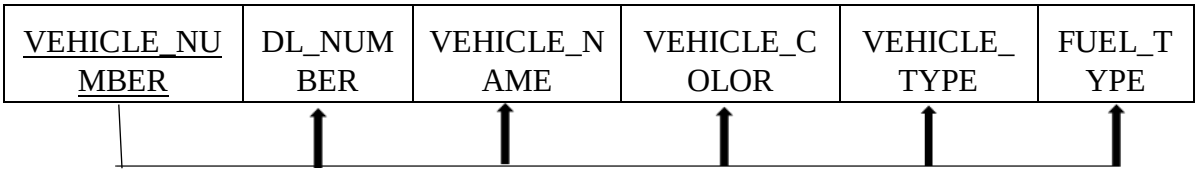


Relation is already is already in 3NF no non-key attribute is dependent on the key attribute via transitive dependency.

RC DETAILS

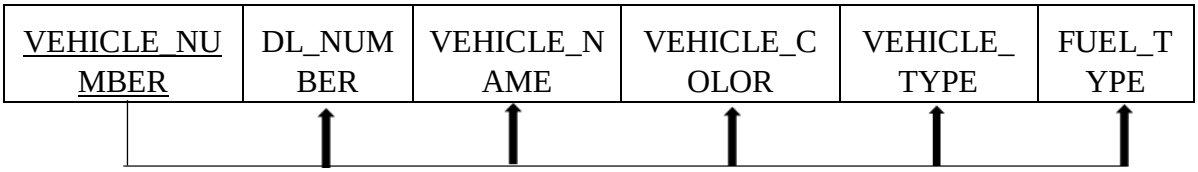


1st NORMALIZATION FORM



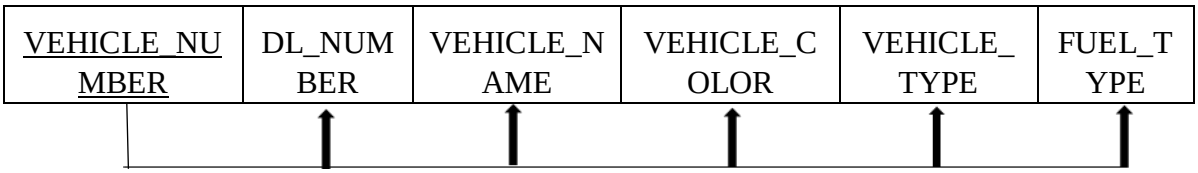
Relation is already in 1NF because it has no multivalued attributes or nested relations.

2nd NORMALIZATION FORM



Relation is already in 2NF since the primary keys does not contain multiple attributes.

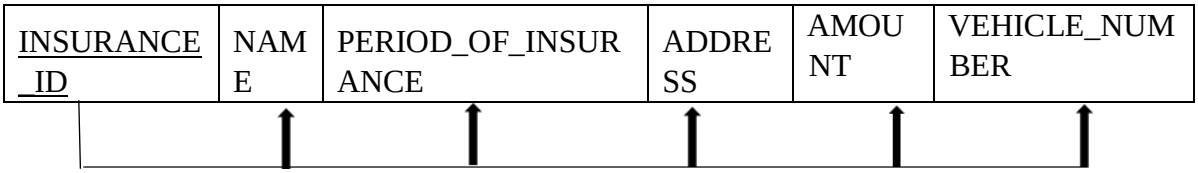
3rd NORMALIZATION FORM



Relation is already is already in 3NF no non-key attribute is dependent on the key attribute via transitive dependency.

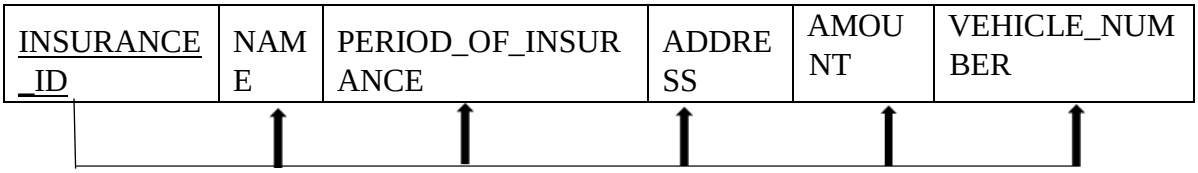
INSURANCE

1st NORMALIZATION FORM



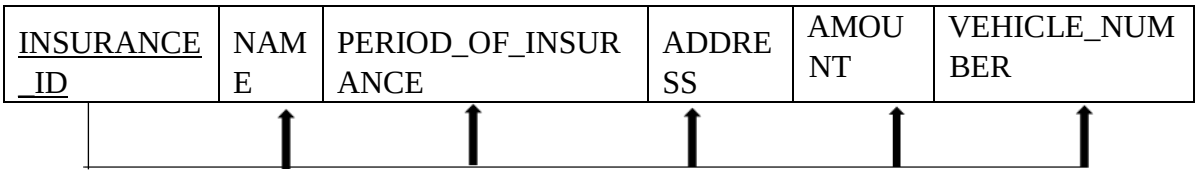
Relation is already in 1NF because it has no multivalued attributes or nested relations.

2nd NORMALIZATION FORM



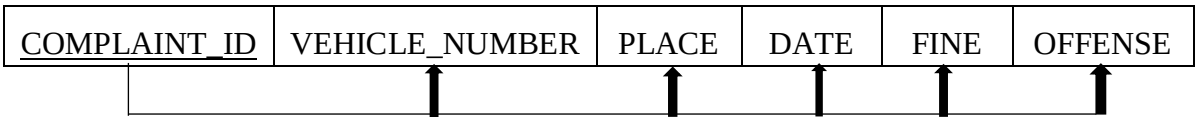
Relation is already in 2NF since the primary keys does not contain multiple attributes.

3rd NORMALIZATION FORM

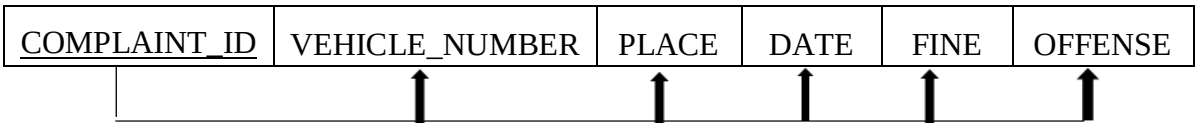


Relation is already is already in 3NF no non-key attribute is dependent on the key attribute via transitive dependency.

VIOLATIONS

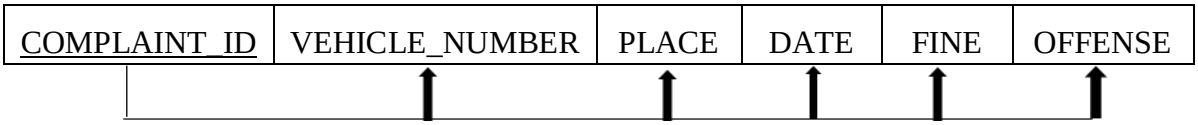


1st NORMALIZATION FORM



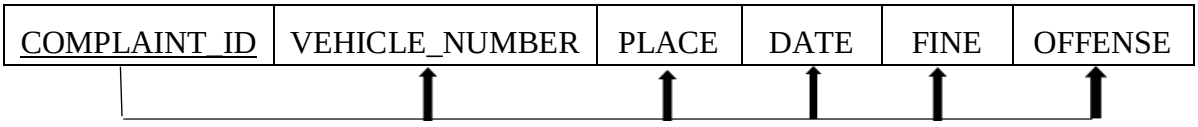
Relation is already in 1NF because it has no multivalued attributes or nested relations.

2nd NORMALIZATION FORM



Relation is already in 2NF since the primary keys does not contain multiple attributes.

3rd NORMALIZATION FORM



Relation is already is already in 3NF no non-key attribute is dependent on the key attribute via transitive dependency.

3.1 SCREEN LAYOUT DESIGN FOR FORMS

LOGIN FORM: The login form consists of **two text fields** and one **login button**. The text fields consist of User name where user enters the **username** with which he has registered and **password** where the user enters the password given when he had registered. The **login button** posts the data the connector, and displays the list of services page. If the user entered wrong USERNAME or Password, it displays the message, "Login failed. Wrong credentials!".

DISPLAY FORM: Once the user logins, the user will options to choose which details he want to update or view etc. The user selects any one of the display options. There are other two buttons 'EXIT' and 'LOGOUT'. Exit button is used to close the application and logout button is used to return back to the previous frame.

CUSTOMER DETAILSFORM: Here the user can view all the details of the CUSTOMER by the help of show button and he can insert or update or delete and any record with the help of the buttons. The user can also search any particular record with respect the options provided. There are other two buttons 'EXIT' and 'BACK'. Exit button is used to close the application and back button is used to return back to the previous frame.

DRIVER'S LICENSE FORM: Here the user can view all the details of the DRIVER'S LICENSE by the help of show button and he can insert or update or delete and any record with the help of the buttons. The user can also search any particular record with respect the options provided. There are other two buttons 'EXIT' and 'BACK'. Exit button is used to close the application and back button is used to return back to the previous frame.

RC DETAILS FORM: Here the user can view all the details of the VEHICLE by the help of show button and he can insert or update or delete and any record with the help of the buttons. The user can also search any particular record with respect the options provided. There are other two buttons 'EXIT' and 'BACK'. Exit button is used to close the application and back button is used to return back to the previous frame.

INSURANCE FORM: Here the user can view all the details of the INSURANCE by the help of show button and he can insert or update or delete and any record with the help of the buttons. The user can also search any particular record with respect the options provided. There are other two buttons 'EXIT' and 'BACK'. Exit button is used to close the application and back button is used to return back to the previous frame.

VIOLATIONS FORM: Here the user can view all the details of the VIOLATIONS committed by the driver with the help of search button after entering the vehicle number and he can insert or update or delete and any record with the help of the buttons. The user can also search any particular record with respect the options provided. There are other two buttons 'EXIT' and 'BACK'. Exit button is used to close the application and back button is used to return back to the previous frame.

3.2 Connection between Front End and Back End

3.2.1 Connecting to a SQL database

You need your SQL connection. The connection string is obtained from the database properties. Import the package java.sql.* into your java programme. Create a new class for **DriverManager**. Create new connection by entering the database name, user name and password for MySQL and create a statement for to access the connection

```
Class.forName("java.sql.DriverManager");
   Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","root","server");
   Statement s=(Statement)c.createStatement();
```

Where rto is database name and root is user_name and server is the password.

The major modules of RTO MANAGEMENT System are:

1. CUSTOMER module
2. DRIVER'S LICENSE module
3. RC DETAILS module
4. INSURANCE MODULE
5. VIOLATIONS MODULE

1. CUSTOMER module

This gives details of all CUSTOMERS. We can also insert new customer records and update the existing records and also delete the existing records. We can also search a particular record by either customer_id or name or contact number and displayed it in the table.

2. **DRIVER'S LICENSE module**

This gives details of all DRIVER'S LICENSE. We can also insert new dl records and update the existing records and also delete the existing records. We can also search a particular record by either dl_number or name or contact number and displayed it in the table.

3. RC DETAILS module

This gives details of all VEHICLE registered to dl_number. We can also insert new rc records and update the existing records and also delete the existing records. We can also search a particular record by either vehicle_number or dl_number or contact number and displayed it in the table.

4. INSURANCE module

This gives details of all INSURANCE insured to the vehicles. We can also insert new insurance records and update the existing records and also delete the existing records. We can also search a particular record by either insurance_id or vehicle_number or contact number and displayed it in the table.

5. VIOLATIONS module

This gives details of all the VIOLATIONS committed by the vehicle. We can also insert new customer records and update the existing records and also delete the existing records. We can search the violations of all vehicles by entering the vehicle_number and all the violations are displayed in the table

DATABASE CODE:

TABLES CREATION

1.CUSTOMER TABLE

```
CREATE TABLE CUSTOMER(CUST_ID VARCHAR(10),NAME
VARCHAR(25),ADDRESS VARCHAR(20),DOB
VARCHAR(10),CONTACT VARCHAR(10),
 CONSTRAINT PK_CUS PRIMARY
KEY(CUST_ID);
```

2. DL TABLE

```
CREATE TABLE DL(CUST_ID VARCHAR(10),DL_NUMBER
VARCHAR(20),NAME VARCHAR(25),ADDRESS
VARCHAR(20),VALIDITY VARCHAR(10),BLD_GROUP
VARCHAR(5),DOB VARCHAR(10),
   -> CONSTRAINT PK_DL PRIMARY KEY(DL_NUMBER),
   -> CONSTRAINT FK_DL FOREIGN
KEY(CUST_ID) REFERENCES
CUSTOMER(CUST_ID) ON DELETE
CASCADE);
```

3. RC TABLE

```
CREATE TABLE RC(DL_NUMBER
VARCHAR(20),VEHICLE_NUMBER
VARCHAR(10),VEHICLE_NAME
VARCHAR(15),VEHICLE_COLOR
VARCHAR(10),VEHICLE_TYPE
VARCHAR(10),FUEL_TYPE
VARCHAR(10),CONSTRAINT PK_RC
PRIMARY
KEY(VEHICLE_NUMBER),CONSTRAI
NT FK_RC FOREIGN
KEY(DL_NUMBER) REFERENCES
```

DL(DL_NUMBER) ON DELETE
CASCADE);

4. INSURANCE TABLE

CREATE TABLE INSURANCE(INSURANCE_ID VARCHAR(20),VEHICLE_NUMBER VARCHAR(10),NAME VARCHAR(15),PERIOD_OF_INSURANCE VARCHAR(10),AMOUNT VARCHAR(10),ADDRESS VARCHAR(10),CONSTRAINT PK_IS PRIMARY KEY(INSURANCE_ID),CONSTRAINT FK_RC FOREIGN KEY(VEHICLE_NUMBER) REFERENCES RC(VEHICLE_NUMBER) ON DELETE CASCADE);

5. VIOLATIONS TABLE

CREATE TABLE VIOLATIONS (COMPLAINT_ID VARCHAR (20), VEHICLE_NUMBER VARCHAR (10),PLACE VARCHAR(15),DATE VARCHAR(10),FINE VARCHAR(10),OFFENSE VARCHAR(10),CONSTRAINT PK_IS PRIMARY KEY(COMPLAINT_ID),CONSTRAINT FK_RC FOREIGN KEY(VEHICLE_NUMBER) REFERENCES RC(VEHICLE_NUMBER) ON DELETE CASCADE);

INSERTION OF VALUES

INSERT INTO CUSTOMER VALUES('1000','PRASID S','#28 RAJAJINAGAR','1998-09-22','9916421622');

INSERT INTO CUSTOMER VALUES('1001','PRAKASH KUMAR','#141 JAYANAGAR','1998-09-20','9591549402');

INSERT INTO CUSTOMER VALUES('1002','RAME GOWDA','#8 RT NAGAR','1968-10-12','7996471022');

INSERT INTO CUSTOMER VALUES('1003','AKSHAY K','#11 THYAGRAJNAGAR','1988-02-02','8989577722');

INSERT INTO CUSTOMER VALUES('1004','DAVID LOBO','#48 RICHMOND TOWN','1978-03-21','8884592514');

INSERT INTO CUSTOMER VALUES('1005','MAX DAVIDSON','#41 JP NAGAR','1998-04-01','9884992518');

INSERT INTO CUSTOMER VALUES('1006','AZAM KHAN','#89 SADASHIVNAGAR','1958-12-11','9696633678','KA4120011225596');

INSERT INTO DL VALUES('1000','KA0412','PRASID S','#28 RAJAJINAGAR','22-08-2030','A+VE','1998-09-22');

INSERT INTO RC VALUES('KA0412','KA1807','Q7','BLK','SUV','DIESEL');

INSERT INTO INSURANCE VALUES('2016128','KA1807','PRASID S','1 YEAR','80000','RAJAJINAGAR');

INSERT INTO VIOLATIONS VALUES('1014','KA1807','CKM','19-07-2018','SPEEDING','500');

FRONT END IMPLEMENTATION IN NETBEANS(JAVA)

CODE FOR LOGIN PAGE

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String user=t1.getText();
char[] pwd=t2.getPassword();
if(user.equals("Pavan")||user.equals("Pradyumna")||user.equals("USERS")&&p
wd.equals("BIT"))
{
   JOptionPane.showMessageDialog(this,"login
Sucecessful\n\tWELCOME\t\t\t"+user);
       display frame3=new display();
login.this.setVisible(false);
frame3.setVisible(true);
}
else
{
   JOptionPane.showMessageDialog(this,"Invalid USERNAME OR
PASSWORD");
}
   }

   private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
Welcome frame1=new Welcome();
       login.this.setVisible(false);
frame1.setVisible(true);
   }
```

CODE FOR CUSTOMER DETAILS FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
   Statement s=(Statement)c.createStatement();
   ResultSet r=s.executeQuery("select * from customer");
   while(r.next())
   {
      int c_id=r.getInt(1);
      String name=r.getString(2);
      String add=r.getString(3);
      String date=r.getString(4);
      long phone=r.getLong(5);
      t.addRow(new Object[]{c_id,name,add,date,phone});
            }
}
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,"NO RECORD FOUND");
}

   }

   private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
      cupdate frame6=new cupdate();
  customer.this.setVisible(false);
  frame6.setVisible(true);
   }

   private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
csearch frame2=new csearch();
  customer.this.setVisible(false);
  frame2.setVisible(true);
   }
```

```
 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    display frame5=new display();
customer.this.setVisible(false);
frame5.setVisible(true);
 }
```

CODE FOR CUSTOMER UPDATE FRAME

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        try
        {
           Class.forName("java.sql.Driver");
           Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
           Statement s=(Statement)c.createStatement();
           PreparedStatement pf=null;
           String s1=t1.getText();
           String s2=t2.getText();
           String s3=t3.getText();
           String s4=t4.getText();
           String s5=t5.getText();

        pf=c.prepareStatement("Update customer set name='"+s2+"',
address='"+s3+"', dob='"+s4+"',contact='"+s5+"' where cust_id=? ");
            pf.setString(1,s1);
            pf.executeUpdate();
        JOptionPane.showMessageDialog(null,"Record sucessfully updated");
        }
        catch(Exception e)
        {
           JOptionPane.showMessageDialog(null,"Error in table updation!");
        }

   }
```

CODE FOR CUSTOMER SEARCH FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
      try
   {
         Class.forName("java.sql.Driver");
         Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
         Statement s=(Statement)c.createStatement();
         PreparedStatement pf=null;
         if(r1.isSelected())
         {
            ResultSet r=s.executeQuery("SELECT * FROM CUSTOMER
WHERE cust_id="+t11.getText()+";");

             while(r.next())
   {
      int c_id=r.getInt(1);
      String name=r.getString(2);
      String add=r.getString(3);
      String date=r.getString(4);
      long phone=r.getLong(5);
      t.addRow(new Object[]{c_id,name,add,date,phone,});
               }
         }

         else if(r2.isSelected())
         {
            String cq=t11.getText();
            pf=c.prepareStatement("SELECT * FROM CUSTOMER WHERE
name=?");
            pf.setString(1,cq);
            ResultSet r=pf.executeQuery();

             while(r.next())
   {
      int c_id=r.getInt(1);
      String name=r.getString(2);
```

```
        String add=r.getString(3);
        String date=r.getString(4);
        long phone=r.getLong(5);
        t.addRow(new Object[]{c_id,name,add,date,phone});
                }
            }
            else if(r3.isSelected())
            {
                String cq=t11.getText();
                pf=c.prepareStatement("SELECT * FROM CUSTOMER WHERE
address=?");
                pf.setString(1,cq);
                ResultSet r=pf.executeQuery();


                 while(r.next())
    {
        int c_id=r.getInt(1);
        String name=r.getString(2);
        String add=r.getString(3);
        String date=r.getString(4);
        long phone=r.getLong(5);
        t.addRow(new Object[]{c_id,name,add,date,phone});
                }
            }

            else if(r4.isSelected())
            {
                ResultSet r=s.executeQuery("SELECT * FROM CUSTOMER
WHERE contact="+t11.getText()+";");

                 while(r.next())
    {
        int c_id=r.getInt(1);
        String name=r.getString(2);
        String add=r.getString(3);
        String date=r.getString(4);
        long phone=r.getLong(5);
```

```
        t.addRow(new Object[]{c_id,name,add,date,phone});
                }
            }


            }
            catch(Exception e)
                {
                JOptionPane.showMessageDialog(null,"No such record found");
                }

    }
```

CODE FOR DL DETAILS FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
   Statement s=(Statement)c.createStatement();
   ResultSet r=s.executeQuery("select * from dl");
   while(r.next())
   {
      int cust_id=r.getInt(1);
      String dl_number=r.getString(2);
      String name=r.getString(3);
      String add=r.getString(4);
      String validity=r.getString(5);
      String blood=r.getString(6);
      String date=r.getString(7);
      t.addRow(new
Object[]{cust_id,dl_number,name,add,validity,blood,date});
                }
}
catch(Exception e)
```

```
{
   JOptionPane.showMessageDialog(this,"NO RECORD FOUND");
}
   }
```

CODE FOR DL UPDATE FRAME

```
try
        {
           Class.forName("java.sql.Driver");
           Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
           Statement s=(Statement)c.createStatement();
           PreparedStatement pf=null;
           String s1=t11.getText();
           String s2=t2.getText();
           String s3=t3.getText();
           String s4=t4.getText();
           String s5=t5.getText();
           String s6=t6.getText();
           String s7=t7.getText();



        pf=c.prepareStatement("Update dl set  name='"+s2+"', address='"+s3+"',
validity='"+s4+"',bld_group='"+s5+"', dob='"+s6+"' where dl_number=? ");
            pf.setString(1,s1);
            pf.executeUpdate();
        JOptionPane.showMessageDialog(null,"Record sucessfully updated");
        }
        catch(Exception e)
        {
           JOptionPane.showMessageDialog(null,"Error in table updation!");
        }

   }
```

CODE FOR DL SEARCH FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
      try
      {
         Class.forName("java.sql.Driver");
         Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
         Statement s=(Statement)c.createStatement();
         PreparedStatement pf=null;
         if(r1.isSelected())
         {
            ResultSet r=s.executeQuery("SELECT * FROM dl WHERE
customer_id="+t11.getText()+";");

           while(r.next())
   {
      int cust_id=r.getInt(1);
      int dl_number=r.getInt(2);
      String name=r.getString(3);
      String add=r.getString(4);
      String validity=r.getString(5);
      String blood=r.getString(6);
      String date=r.getString(7);
      t.addRow(new
Object[]{cust_id,dl_number,name,add,validity,blood,date});
               }
         }
         if(r2.isSelected())
         {
            String cq=t11.getText();
   pf=c.prepareStatement("SELECT * FROM dl WHERE  dl_number=?");
    pf.setString(1,cq);
    ResultSet r=pf.executeQuery();


           while(r.next())
   {
      int cust_id=r.getInt(1);
```

```
        int dl_number=r.getInt(2);
        String name=r.getString(3);
        String add=r.getString(4);
        String validity=r.getString(5);
        String blood=r.getString(6);
        String date=r.getString(7);
        t.addRow(new
Object[]{cust_id,dl_number,name,add,validity,blood,date});
                }
           }

          else if(r3.isSelected())
           {
              String cq=t11.getText();
       pf=c.prepareStatement("SELECT * FROM CUSTOMER WHERE
name=?");
               pf.setString(1,cq);
               ResultSet r=pf.executeQuery();

               while(r.next())
    {
        int cust_id=r.getInt(1);
        int dl_number=r.getInt(2);
        String name=r.getString(3);
        String add=r.getString(4);
        String validity=r.getString(5);
        String blood=r.getString(6);
        String date=r.getString(7);
        t.addRow(new
Object[]{cust_id,dl_number,name,add,validity,blood,date});
                }
           }

        }
        catch(Exception e)
        {
           JOptionPane.showMessageDialog(null,"No such record found");
        }
```

CODE FOR RC DETAILS FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
   Statement s=(Statement)c.createStatement();
   ResultSet r=s.executeQuery("select * from rc");
   while(r.next())
   {
      String vehicle_number=r.getString(1);
      String dl_name=r.getString(2);
      String vehicle_name=r.getString(3);
      String vehicle_color=r.getString(4);
      String vehicle_type=r.getString(5);
      String fuel_type=r.getString(6);
      t.addRow(new
Object[]{vehicle_number,dl_name,vehicle_name,vehicle_color,vehicle_type,f
uel_type});
               }
}
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,"NO RECORD FOUND");
}
```

CODE FOR RC UPDATE FRAME

```
 try
        {
           Class.forName("java.sql.Driver");
           Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
           Statement s=(Statement)c.createStatement();
```

```
            String s1=t1.getText();  //vehicle number
            String s3=t3.getText();  // vehicle name
            String s4=t4.getText();  //vehicle color
            String s5=t5.getText(); //vehicle type
            String s6=t6.getText(); //fuel type
            String query="Update rc set  vehicle_name='"+s3+"',
vehicle_color='"+s4+"',vehicle_type='"+s5+"', fuel_type='"+s6+"' where
vehicle_number='"+s1+"' ";
        s.executeUpdate(query);
        JOptionPane.showMessageDialog(null,"Record sucessfully updated");
        }
        catch(Exception e)
        {
           JOptionPane.showMessageDialog(null,"Error in table updation!");
        }
   }

   private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
rc frame6=new rc();
rcupdate.this.setVisible(false);
frame6.setVisible(true);
   }

   private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
   }

   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
String i1=t1.getText();
            String i2=t2.getText();
            String i3=t3.getText();
            String i4=t4.getText();
            String i5=t5.getText();
            String i6=t6.getText();
      try
   {
```

```
        Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
        Statement s=(Statement)c.createStatement();
     s.executeUpdate("insert into rc
(vehicle_number,dl_number,vehicle_name,vehicle_color,vehicle_type,fuel_typ
e) values('"+i1+"','"+i2+"','"+i3+"','"+i4+"','"+i5+"','"+i6+"')");
     JOptionPane.showMessageDialog(null,"Inserted Successfully!");
     }
    catch(Exception e)
    {
    JOptionPane.showMessageDialog(null,"Error in insertion");
          }

    }
```

CODE FOR RC SEARCH FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
       try
    {
          Class.forName("java.sql.Driver");
          Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
          Statement s=(Statement)c.createStatement();
          PreparedStatement pf=null;
          if(r1.isSelected())
          {
             String cq=t11.getText();
             pf=c.prepareStatement("SELECT * FROM rc WHERE
vehicle_number=?");
             pf.setString(1,cq);
             ResultSet r=pf.executeQuery();


             while(r.next())
    {
```

```
        String dl_number=r.getString(1);
        String vehicle_number=r.getString(2);
        String vehicle_name=r.getString(3);
        String vehicle_color=r.getString(4);
        String vehicle_type=r.getString(5);
        String fuel_type=r.getString(6);
        t.addRow(new
Object[]{dl_number,vehicle_number,vehicle_name,vehicle_color,vehicle_type
,fuel_type});
                }
            }

            else if(r2.isSelected())
            {
                String cq=t11.getText();
                pf=c.prepareStatement("SELECT * FROM rc WHERE
dl_number=?");
                pf.setString(1,cq);
                ResultSet r=pf.executeQuery();

                while(r.next())
    {
        String dl_number=r.getString(1);
        String vehicle_number=r.getString(2);
        String vehicle_name=r.getString(3);
        String vehicle_color=r.getString(4);
        String vehicle_type=r.getString(5);
        String fuel_type=r.getString(6);
        t.addRow(new
Object[]{dl_number,vehicle_number,vehicle_name,vehicle_color,vehicle_type
,fuel_type});
                }
            }

    }
            catch(Exception e)
                {
                JOptionPane.showMessageDialog(null,"No such record found");
```

```
            }
    }
```

CODE FOR INSURANCE DETAILS FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
try
{
   Class.forName("java.sql.DriverManager");
   Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
   Statement s=(Statement)c.createStatement();
   ResultSet r=s.executeQuery("select * from insurance");
   while(r.next())
   {
      String insurance_id=r.getString(1);
      String vehicle_number=r.getString(2);
      String name=r.getString(3);
      String insurance_period=r.getString(4);
      String address=r.getString(6);
      int amount=r.getInt(5);
      t.addRow(new
Object[]{insurance_id,vehicle_number,name,insurance_period,amount,address
});
            }
}
catch(Exception e)
{
   JOptionPane.showMessageDialog(this,"NO RECORD FOUND");
}
   }
```

CODE FOR INSURANCE UPDATE FRAME

```
try
        {
           Class.forName("java.sql.Driver");
```

```
            Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
            Statement s=(Statement)c.createStatement();
            PreparedStatement pf=null;
            String s1=t1.getText();
            String s3=t3.getText();
            String s4=t4.getText();
            String s5=t5.getText();
            String s6=t6.getText();
          pf=c.prepareStatement("Update insurance set   name='"+s3+"',
period='"+s4+"',amount='"+s5+"', address='"+s6+"' where insurance_id=? ");
             pf.setString(1,s1);
             pf.executeUpdate();
          JOptionPane.showMessageDialog(null,"Record sucessfully updated");
          }
          catch(Exception e)
          {
             JOptionPane.showMessageDialog(null,"Error in table updation!");
          }
    }
```

CODE FOR INSURANCE SEARCH FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
       try
       {
          Class.forName("java.sql.Driver");
          Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
          Statement s=(Statement)c.createStatement();
          PreparedStatement pf=null;
          if(r1.isSelected())
          {
             ResultSet r=s.executeQuery("SELECT * FROM insurance WHERE
insurance_id="+t11.getText()+";");

             while(r.next())
```

```
   {
      String insurance_id=r.getString(1);
      String vehicle_number=r.getString(2);
      String name=r.getString(3);
      String insurance_period=r.getString(4);
      int amount=r.getInt(5);
      String address=r.getString(6);
      t.addRow(new
Object[]{insurance_id,vehicle_number,name,insurance_period,amount,address
});
               }
         }

         else if(r2.isSelected())
         {
            String cq=t11.getText();
            pf=c.prepareStatement("SELECT * FROM insurance WHERE
vehicle_number=?");
            pf.setString(1,cq);
            ResultSet r=pf.executeQuery();


           while(r.next())
   {
     String insurance_id=r.getString(1);
      String vehicle_number=r.getString(2);
      String name=r.getString(3);
      String insurance_period=r.getString(4);
      int amount=r.getInt(5);
      String address=r.getString(6);
      t.addRow(new
Object[]{insurance_id,vehicle_number,name,insurance_period,amount,address
});
               }
         }
          else if(r3.isSelected())
         {
            String cq=t11.getText();
```

```
    pf=c.prepareStatement("SELECT * FROM insurance WHERE name=?");
        pf.setString(1,cq);
         ResultSet r=pf.executeQuery();

              while(r.next())
   {
      String insurance_id=r.getString(1);
      String vehicle_number=r.getString(2);
      String name=r.getString(3);
      String insurance_period=r.getString(4);
      int amount=r.getInt(5);
      String address=r.getString(6);
      t.addRow(new
Object[]{insurance_id,vehicle_number,name,insurance_period,amount,address
});

                }
          }


      }
      catch(Exception e)
      {
         JOptionPane.showMessageDialog(null,"No such record found");
      }
   }
```

CODE FOR VIOLATIONS DETAILS FRAME

```
DefaultTableModel t=(DefaultTableModel)t1.getModel();
      try
      {
         Class.forName("java.sql.Driver");
         Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
         Statement s=(Statement)c.createStatement();
         PreparedStatement pf=null;
```

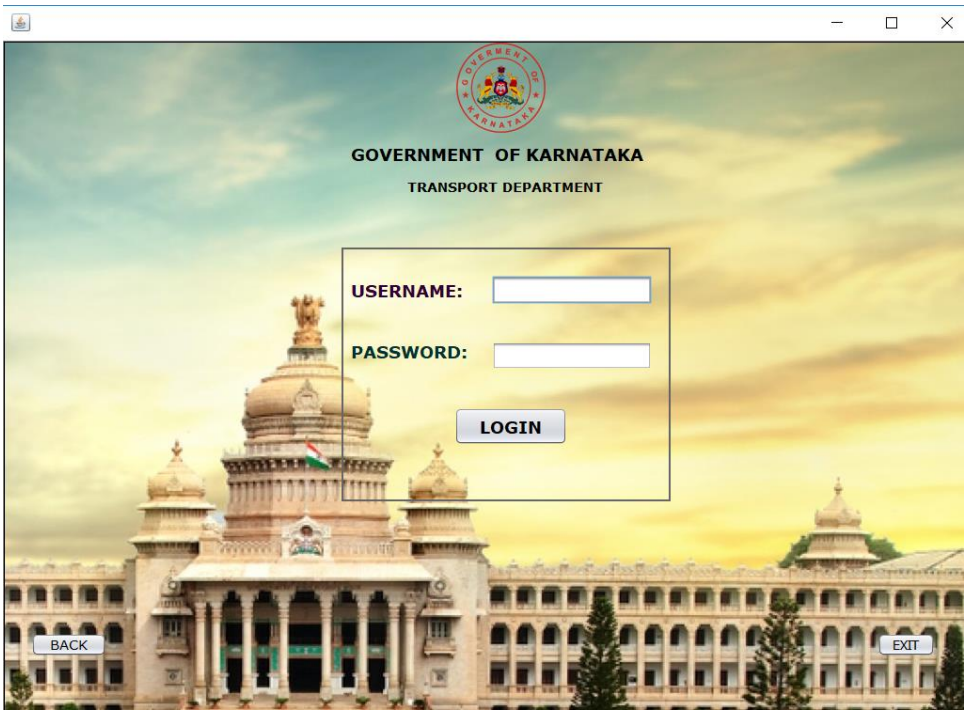
```
            String cq=t11.getText();
            pf=c.prepareStatement("SELECT * FROM violations WHERE
vehicle_number=?");
            pf.setString(1,cq);
            ResultSet r=pf.executeQuery();

            while(r.next())
   {
      int complaint_id=r.getInt(1);
      String vehicle_number=r.getString(2);
      String place=r.getString(3);
      String date=r.getString(4);
      String offense=r.getString(5);
      int amount=r.getInt(6);
      t.addRow(new
Object[]{complaint_id,vehicle_number,place,date,offense,amount});
               }

      }
         catch(Exception e)
      {
         JOptionPane.showMessageDialog(null,"No such record found");
      }
   }
```

CODE FOR VIOLATIONS UPDATE FRAME

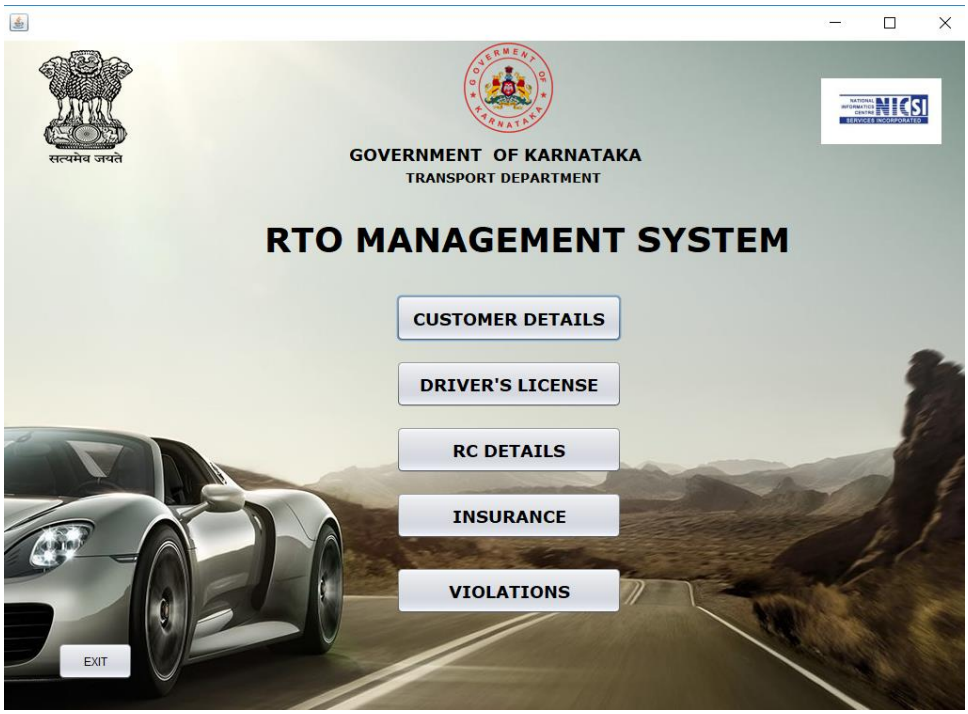
```
String i1=t1.getText();
            String i2=t2.getText();
            String i3=t3.getText();
            String i4=t4.getText();
            String i5=t5.getText();
            String i6=t6.getText();

      try
   {
```

```
        Connection
c=(Connection)DriverManager.getConnection("jdbc:mysql://localhost/rto","ro
ot","server");
        Statement s=(Statement)c.createStatement();
    s.executeUpdate("insert into violations
(complaint_id,vehicle_number,place,date,offense,amount)
values('"+i1+"','"+i2+"','"+i3+"','"+i4+"','"+i5+"','"+i6+"')");
     JOptionPane.showMessageDialog(null,"Inserted Successfully!");
     }
    catch(Exception e)
    {
    JOptionPane.showMessageDialog(null,"Error in insertion");
          }
    }
```

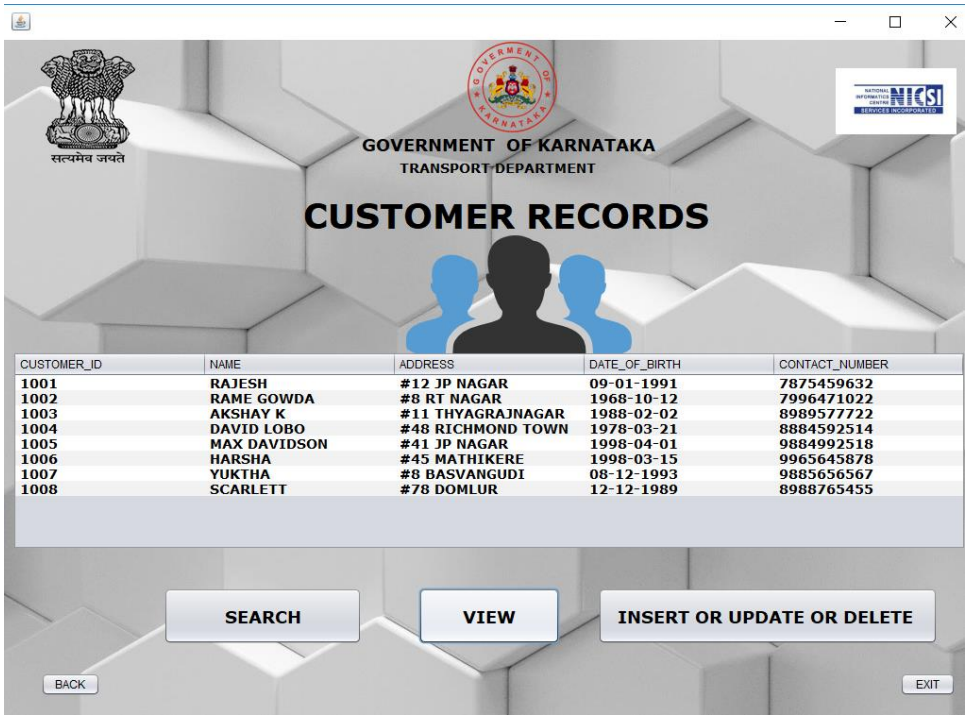
STEP 1: Welcome page



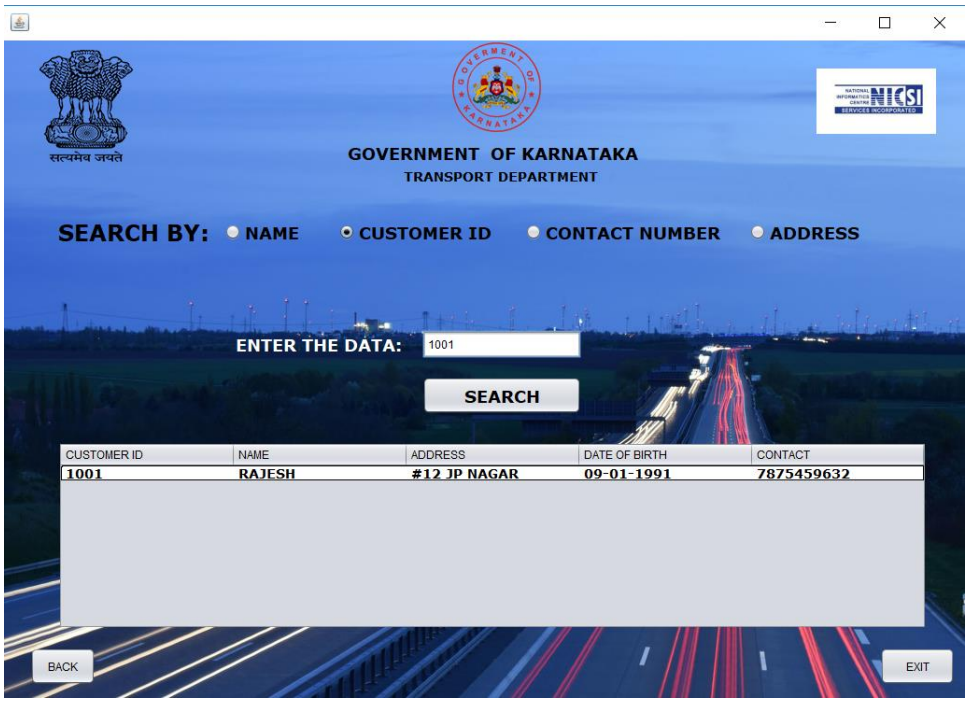
Step 2: User has to login by using user_name and password



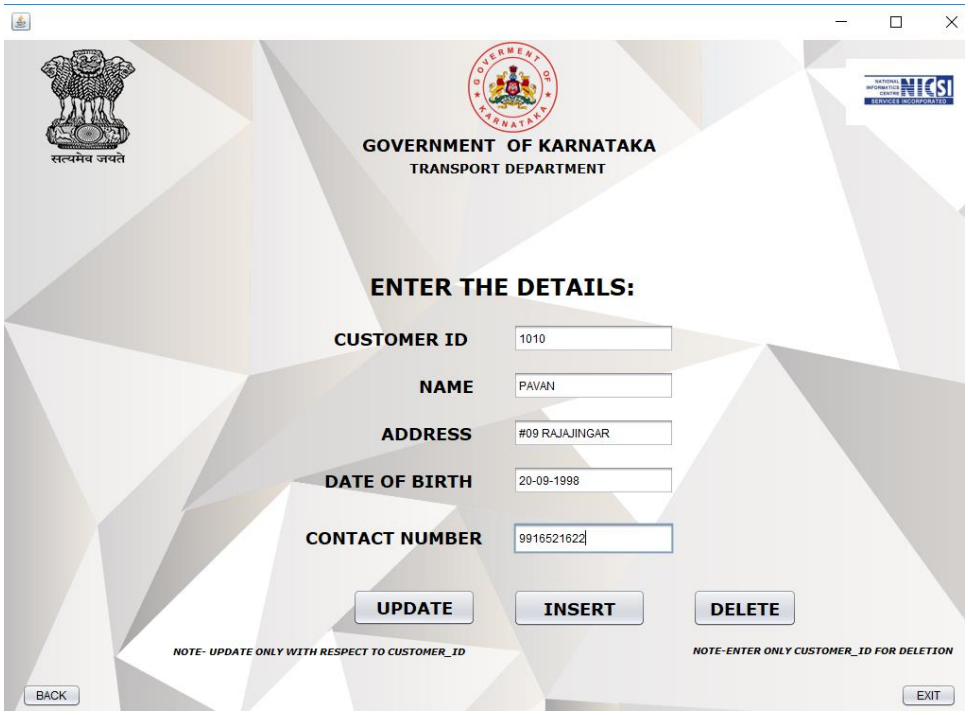
Step 3: User can select any one to view the data or edit the data



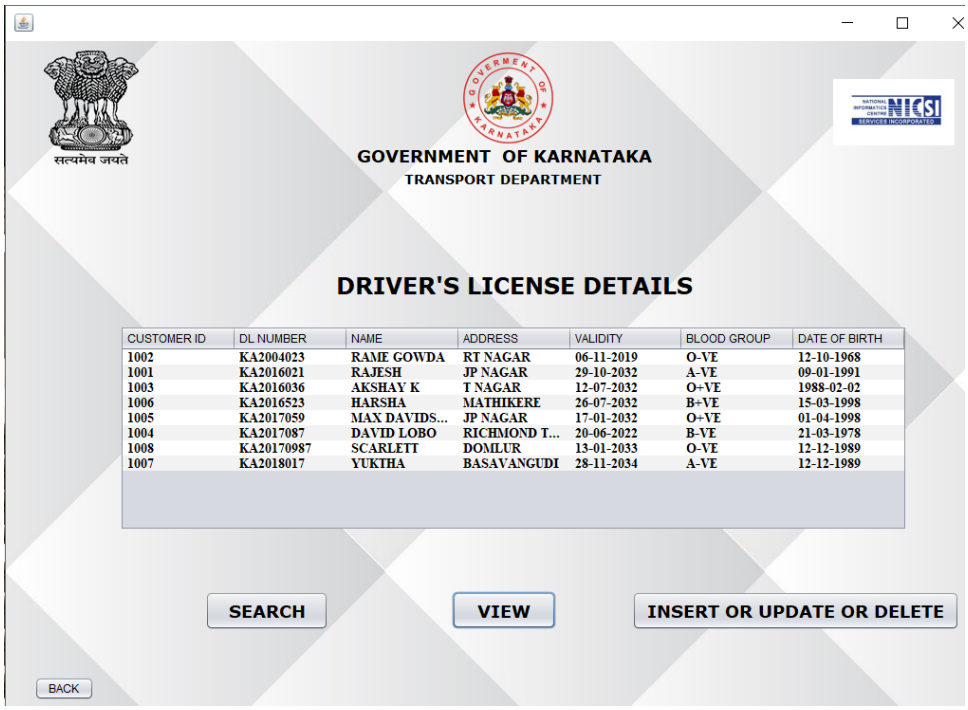
Step 4: When user selects customer record and he can use view buttons to view the data.



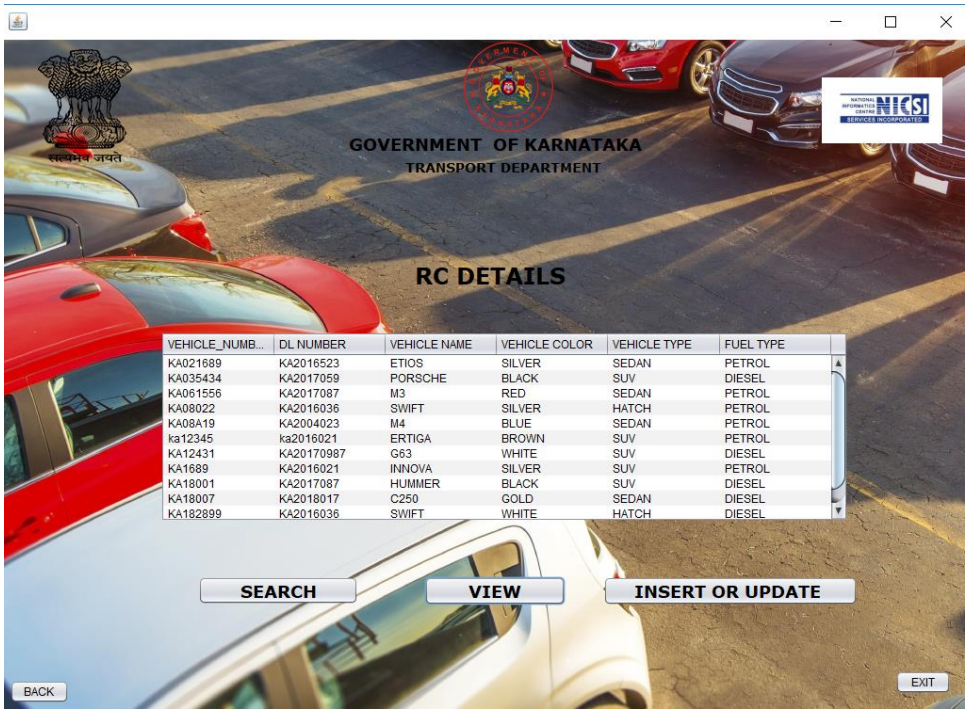
Step 5: User can search any data by few attributes as specified.



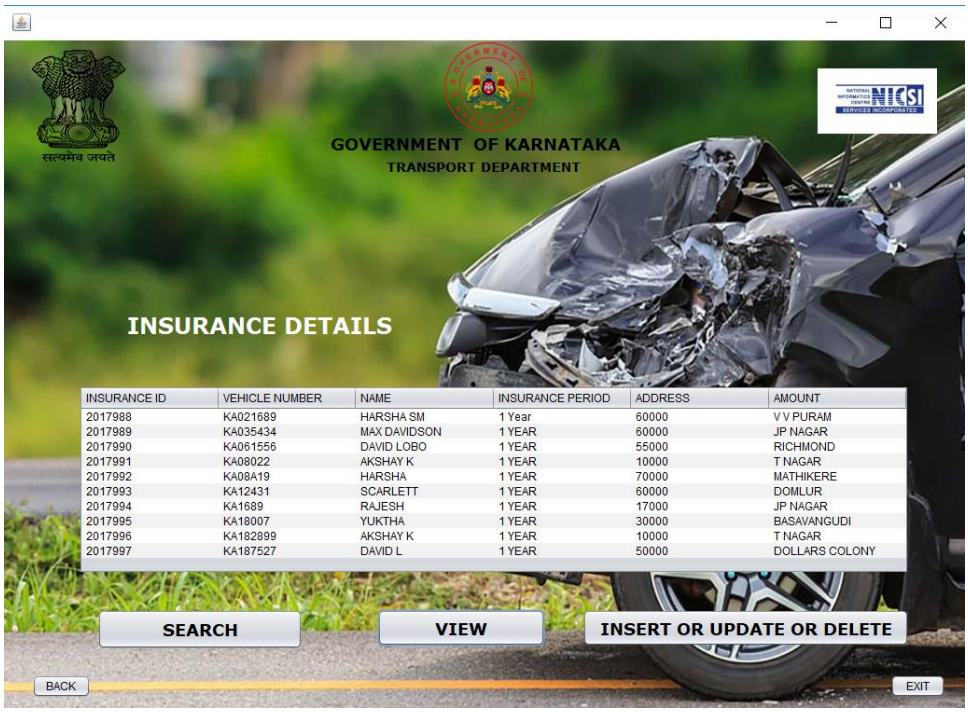
Step 6: User can update or insert or delete any customer record.



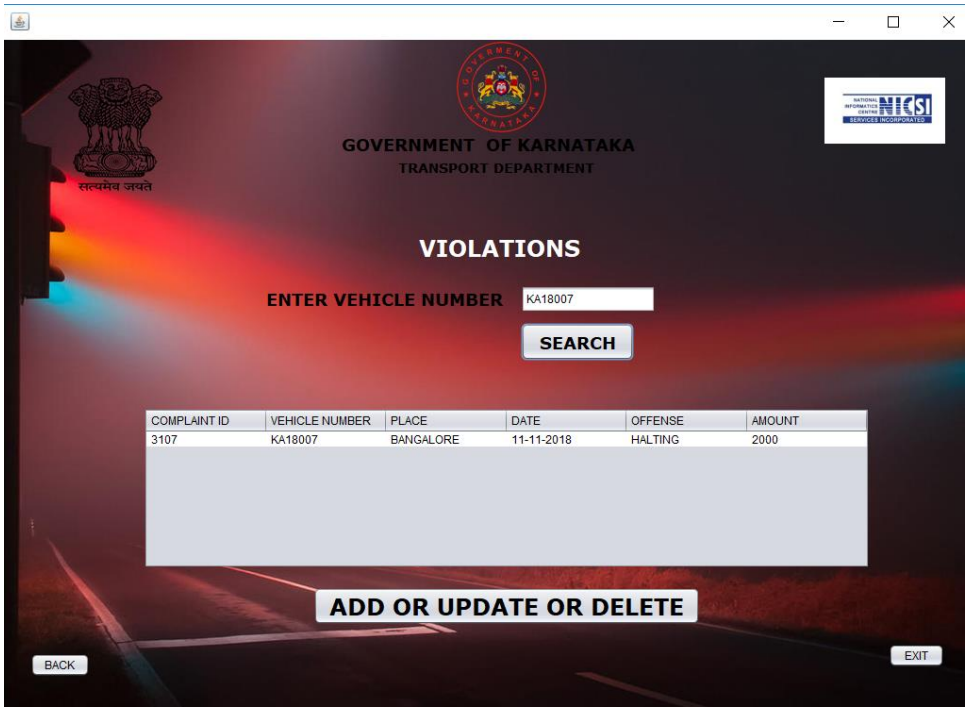
Step 7: When user selects DL record and he can use view buttons to view the data.



Step 8: When user selects RC record and he can use view buttons to view the data.



Step 9: When user selects insurance record and he can use view buttons to view the data.



Step 10: User can view the violations committed by a vehicle by entering vehicle number.

APPLICATIONS:

1. RTO Officers can use RTO Management System to access the data of any customer
2. Police Officers can use it to find details of a particular vehicle
3. Traffic Police Officers can use it to find the violations of a vehicle

CONCLUSION

RTO Management System manages the information in a quick manner, reduces the workload and expedite work procedure for the fast service of RTO. IT take care of all requirements on a RTO and is capable to provide easy and effective storage of information related to customers, dl and vehicle. The implementation of the system in the organisation will considerably reduce data entry, time.

It has facility where RTO officer can check the violations committed by a vehicle. It also has facility where RTO officers can search any data with tranquil.