PROJECT REPORT

ON

PAINT SHOP MANAGEMENT SYSTEM

(UNDER THE PARTIAL FULFILLMENT OF THE UNIVERSITY)

FOR THE COURSE OF T.Y.B.SC (COMPUTER SCIENCE)

SUBMITTED BY:

SINGH VIVEK ANIL

GUIDED BY:

PROF. SHYLASHREE DEV

DEPARTMENT OF COMPUTER SCIENCE

PARLE TILAK VIDYALAYA ASSOCIATIONS

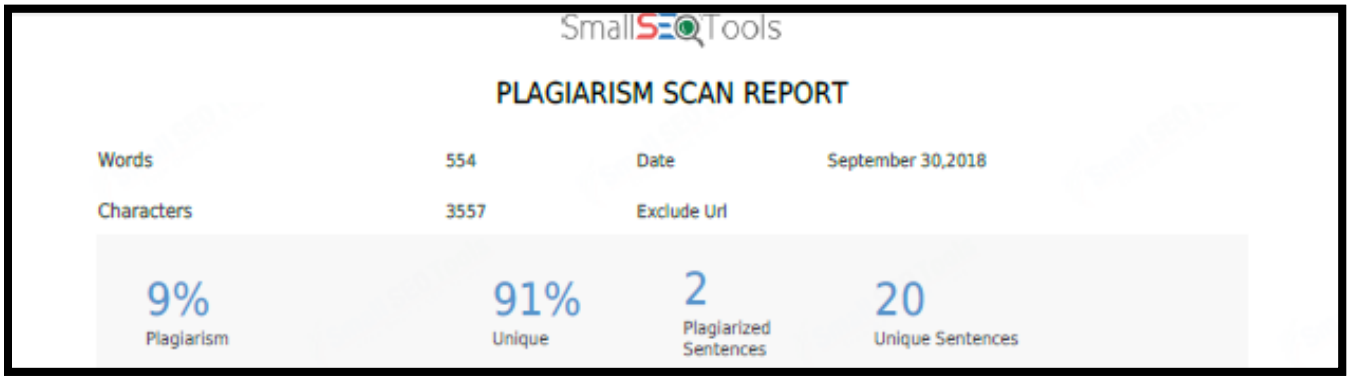
MULUND COLLEGE OF COMMERCE

S.N.ROAD,MULUND(WEST).MUMBAI-80

UNIVERSITY OF MUMBAI

2018-19

Plagiarism Report



SmallSEOTools

PLAGIARISM SCAN REPORT

Words	554	Date	September 30,2018
Characters	3557	Exclude Url	

9%	91%	2	20
Plagiarism	Unique	Plagiarized Sentences	Unique Sentences

ACKNOWLEDGEMENT

I like to extend my gratitude to Dr.Mrs.Sonali Pednekar, our Incharge Principal and all staff of Mulund College of Commerce for providing us moral support, conducive work environment and the much needed inspiration to complete this project on time.

I also take this opportunity to thank our Course Coordinator Prof. Mrs. Reena Shah and all the faculties of Department of Computer Science for giving us the most needed guidance and continuous encouragement throughout the duration of the Programme.

I wish to extend my deepest gratitude and special thanks to my project guide Prof. MRS.SHYLASHREE DEV, for giving their generous support, necessary inputs and companionship during my project work.

I would like to convey my special thanks to the Management and all the staff of the college for providing the required infrastructure and resource to enable the completion and enrichment of my project.

I am extremely grateful to the University of Mumbai for having prescribed this project work to me as a part of the academic requirement in the Final year of Bachelor of Science in Computer Science.

Finally I thank all my fellow friends who have directly or indirectly helped me in completing my project.

INDEX

1.TITLE

Title of the Project: Paintshop Management System

Type of Project: Java Desktop

Developed by: Vivek Singh

2.INTRODUCTION

Paintshop Management System is a simple and understandable desktop application which help to understand a paintshop system. This system maintains all the records of the customer and provide a proper detail and varities to the customer. Basically this system is only for the admin which can we done all the things sales, payement, etc.

In this system user or consumer can think about it and can take his own time to go for it or not even they can give suggestion to improve the Desktop application. help is always available for customer.

Paintshoop Management System is providing a ideas to the customer about the coloring and other facilities

2.1 Features:

- Esay to understand.
- Ensure data accuracies.
- Minimum time needed for the various processing.
- User friendliness and interactive.
- Validators and validation.
- Payment done cheque or cash

2.2 Advantages:

- In this system user can easily get refund and provide many more facilities.
- Damage and loss of material should be pay by shopkeeper.
- Easy way to communicate Customer
- Payment done after the delivery advance must be provided.
- Fast Delivery
- Cost effective. there will we minimium Amount comparing to other Paintshop management System company

3.REQUIREMENT SPECIFICATION

3.1 Software Requirements:

- Java Jframe Swings
- NetBeans 8.2
- Mysql Database
- Windows 7 and above

3.2 Hardware Requirements:

- Minimum RAM 2GB
- Printer – Resolution 360dpi
- Processor – Intel(R) Core (TM) i3-3220 and above
- Hard Disk – 50 GB

3.3 Data Requirements:

- String
- Email
- Integer
- Date

3.4 Fact Finding Questions:

1. What are the difficulties you are finding with the current system?
2. Will you be able to compensate with new technology?
3. How the data is maintained? Is it manual or computer used to store data?
4. Should I make new software for you?
5. How do you keep track of records?
6. What all things do you want to add in the new system?
7. What is the current system about?
8. How is the work currently being done?

4.SYSTEM DESIGN DETAILS

4.1 Event Table:

No.	Event	Trigger	Source	Activity	Response	Destination
1.	User wants to register	New Registration	User	Filling up Registration Form	New applicant registration	Database
2.	Admin Log in	Log in	User/ Admin	Enter Log in id and password	Confirmation and proceed	Database
3.	Product Register	New product's details	Admin	Filling up the Job details	Details submitted and proceed	Database
4.	Company register	New Company form	Admin	Entering all the essential fields	Details submitted and proceed	Database
5.	Category	Category form	User	Enter the detail	Details submitted and proceed	Database

Table 4.1 Event Table for Paintshop Management System

4.2 Class Diagram:

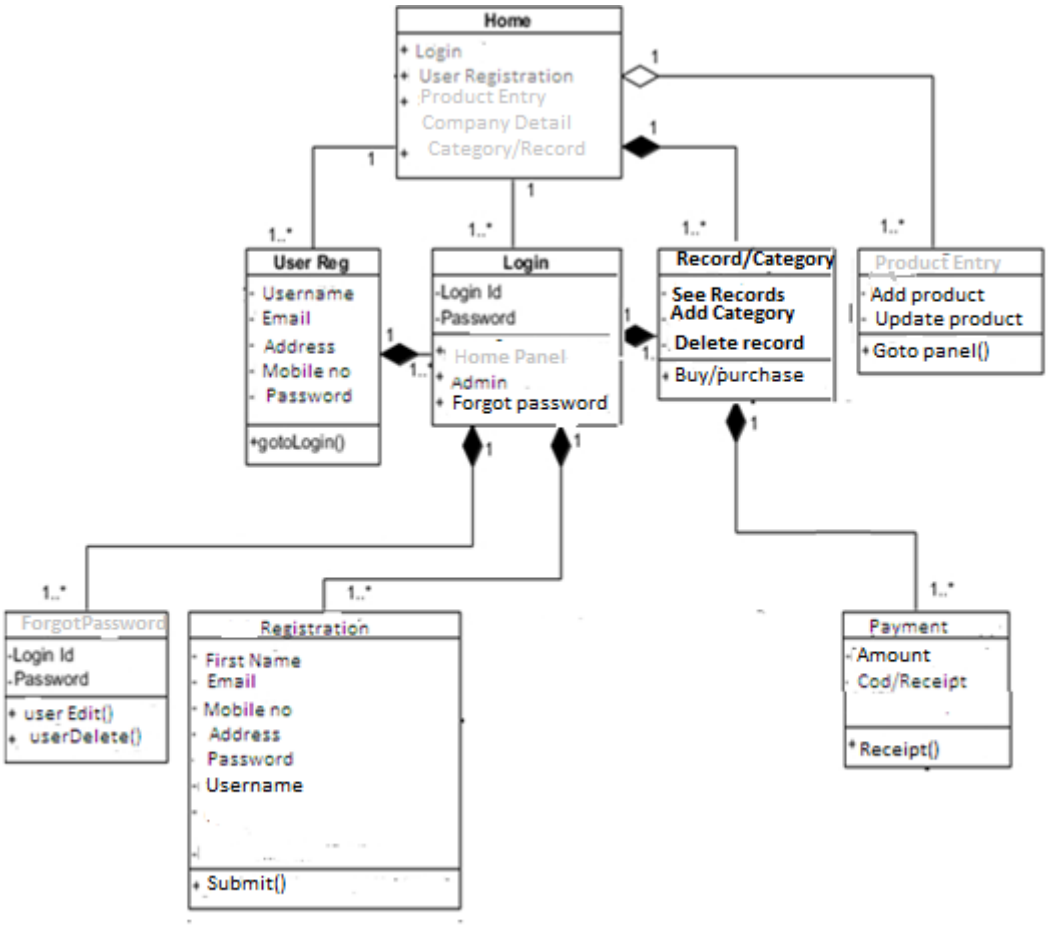


Figure 4.2 Class Diagram for Paintshop Management System

4.3 Use Case Diagrams:

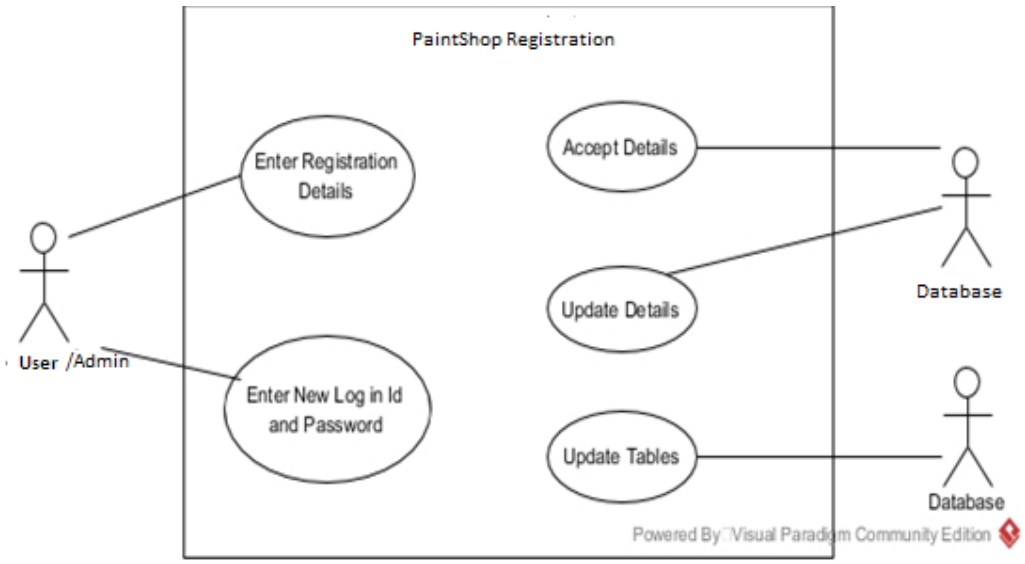


Figure 4.3.1 Use Case Diagram for Paintshop Registration

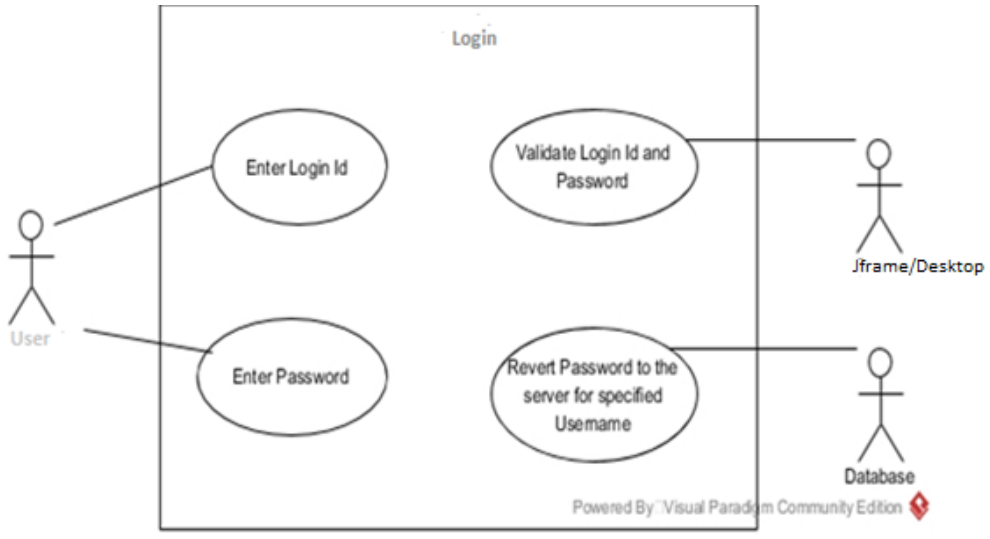


Figure 4.3.2 Use Case Diagram for Login

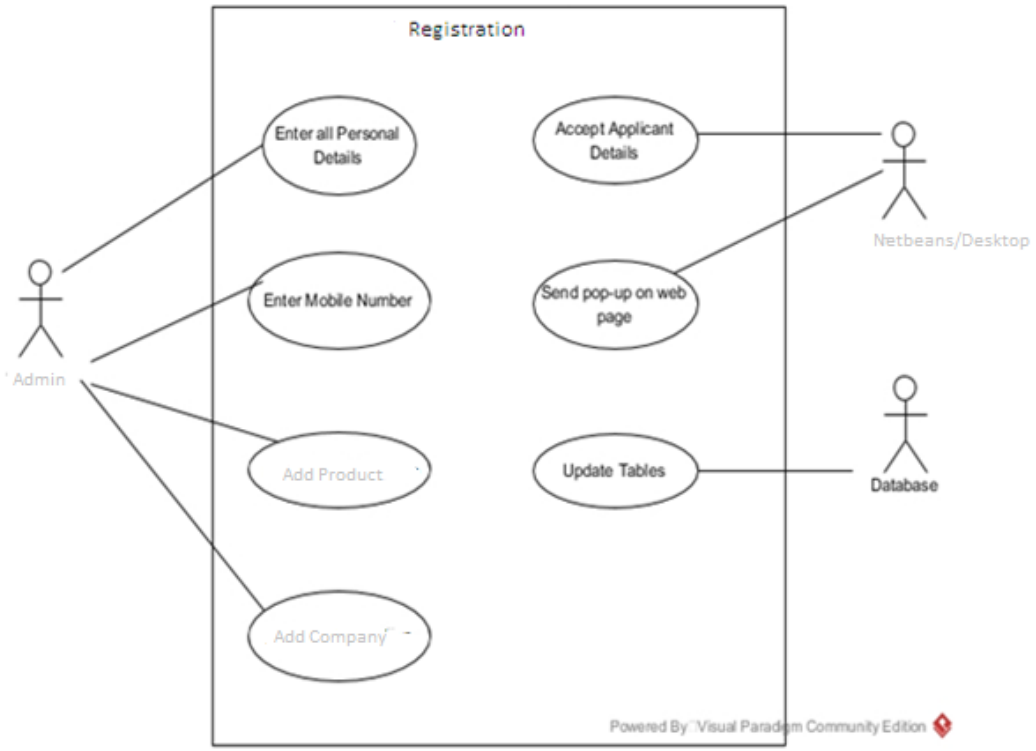


Figure 4.3.3 Use Case Diagram for Product Entry

4.4 Sequence Diagrams:

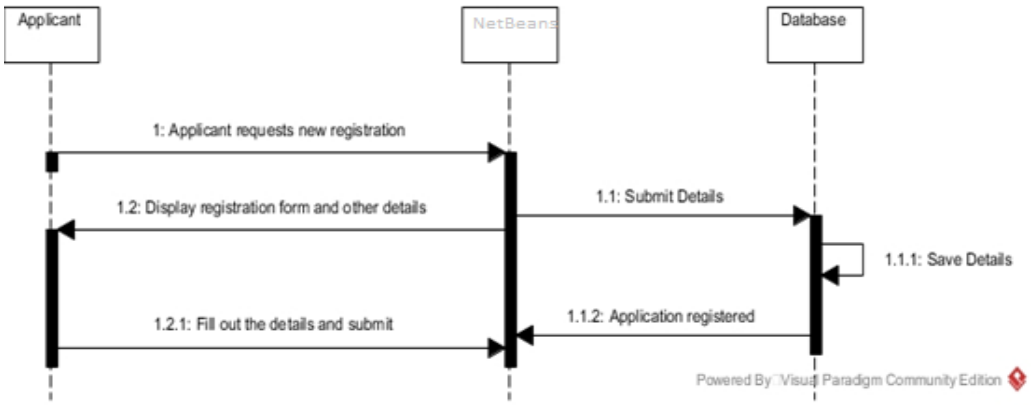


Figure 4.4.1 Sequence Diagram for Paintshop Management System

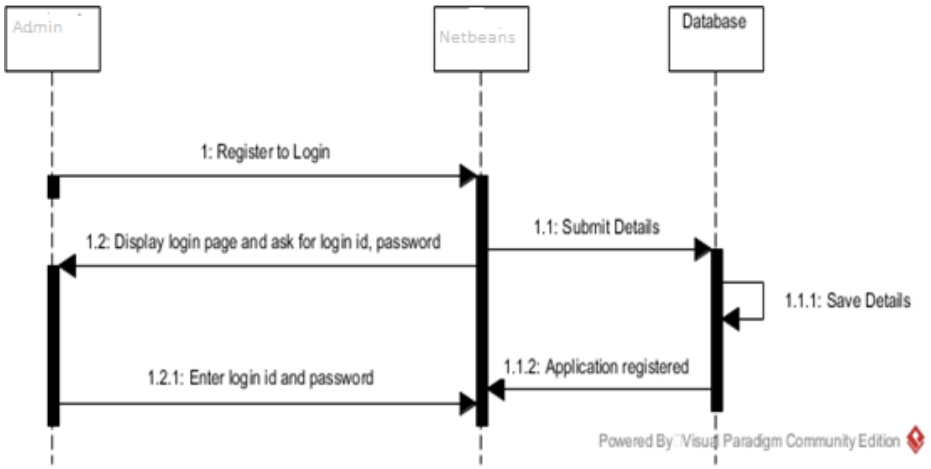


Figure 4.4.2 Sequence Diagram for Login

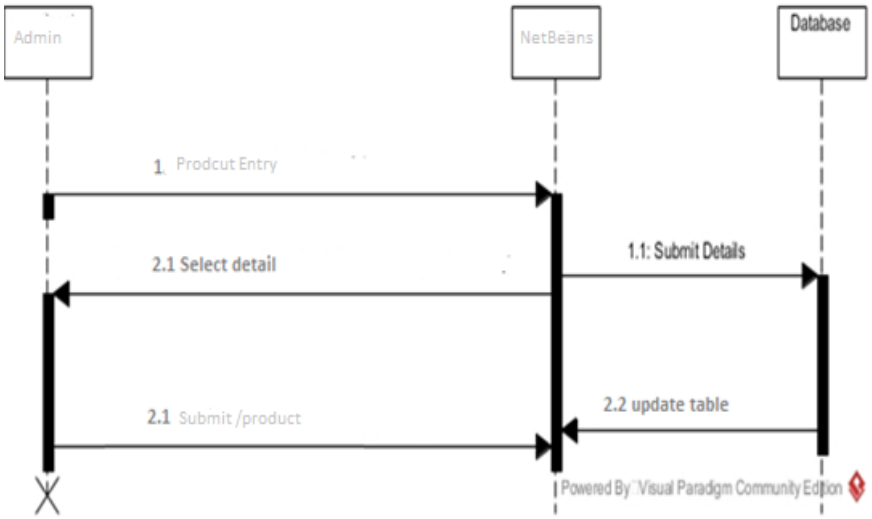


Figure 4.4.3 Sequence Diagram for Product

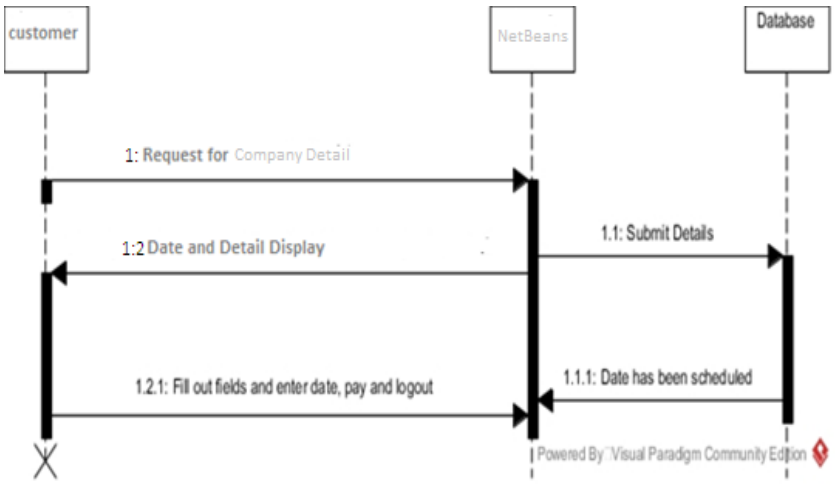


Figure 4.4.4 Sequence Diagram for Company

4.5 Activity Diagram:

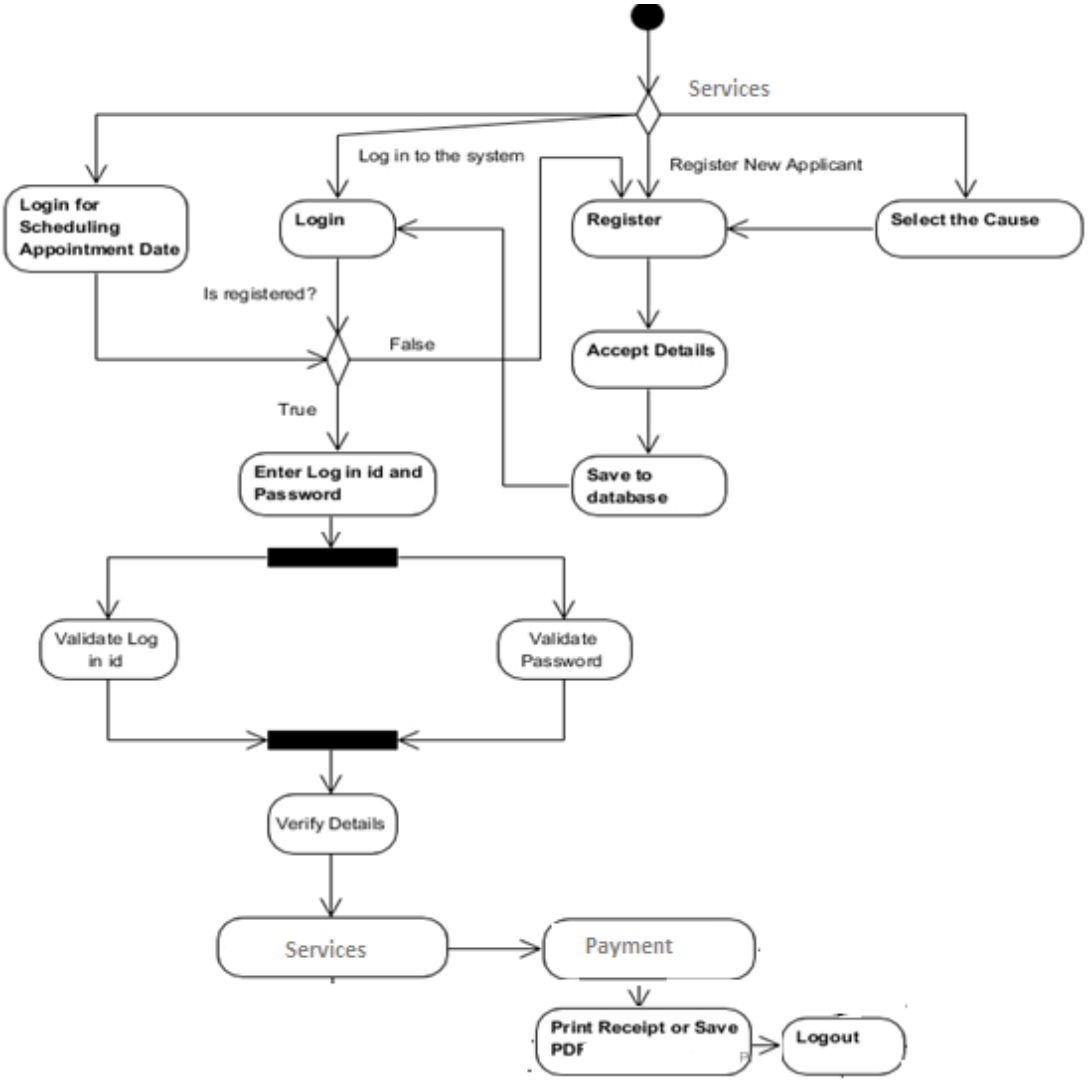


Figure 4.5 Activity Diagram for Paintshop Management System

4.6 State Diagrams:

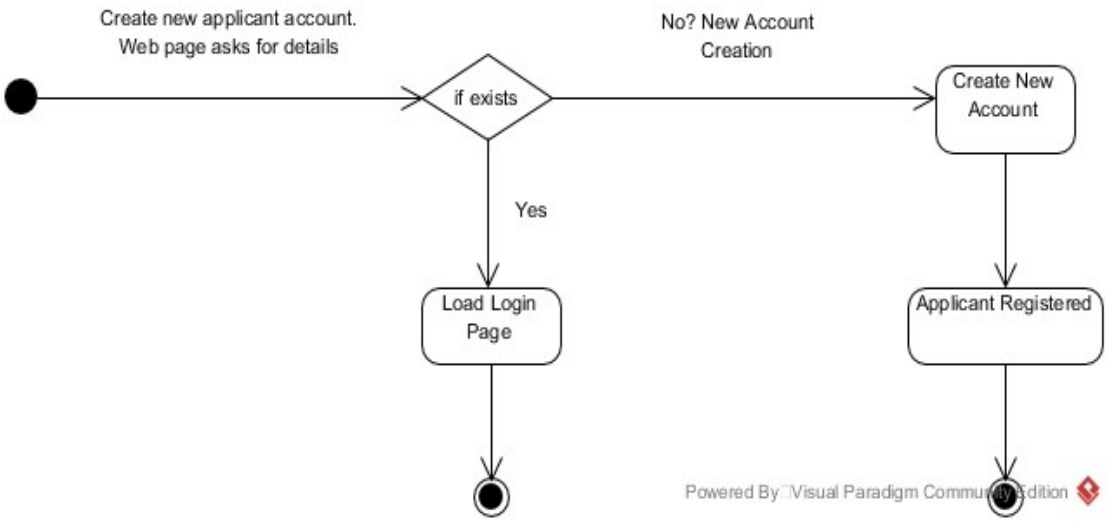


Figure 4.6.1 State Diagram for Paintshop Reg.

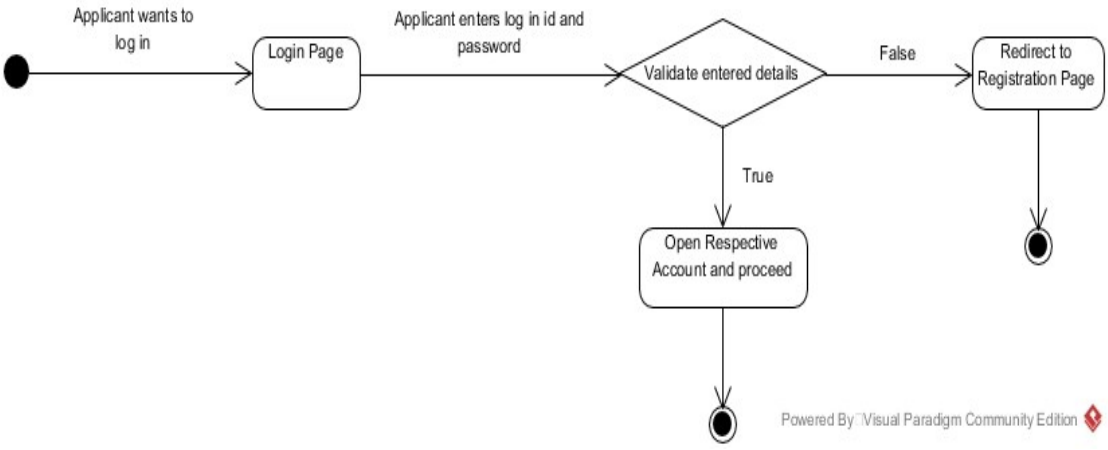


Figure 4.6.2 State Diagram for Login

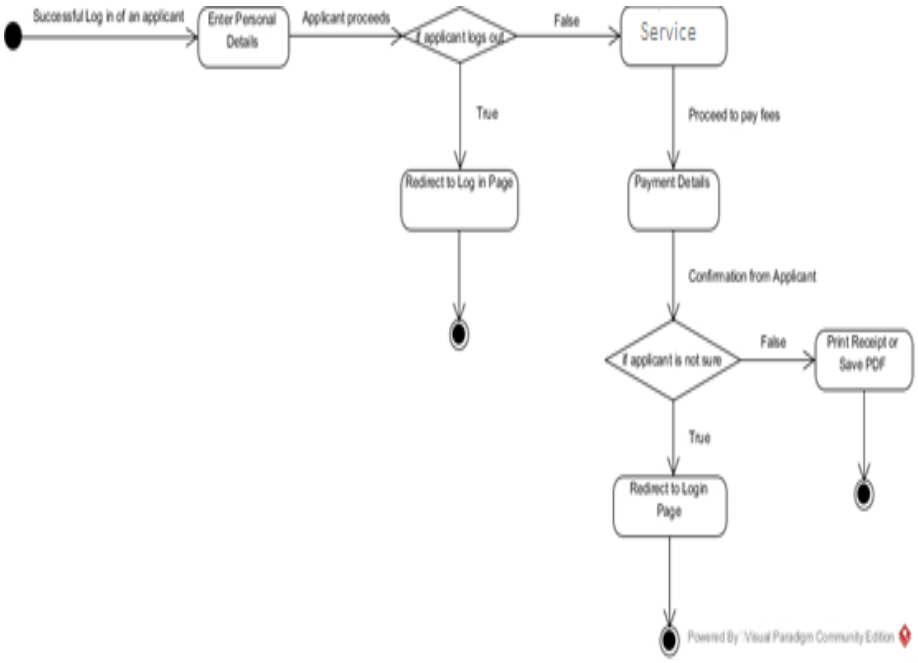


Figure 4.6.3 State Diagram for Paintshop Management System

4.7 Package Diagram:

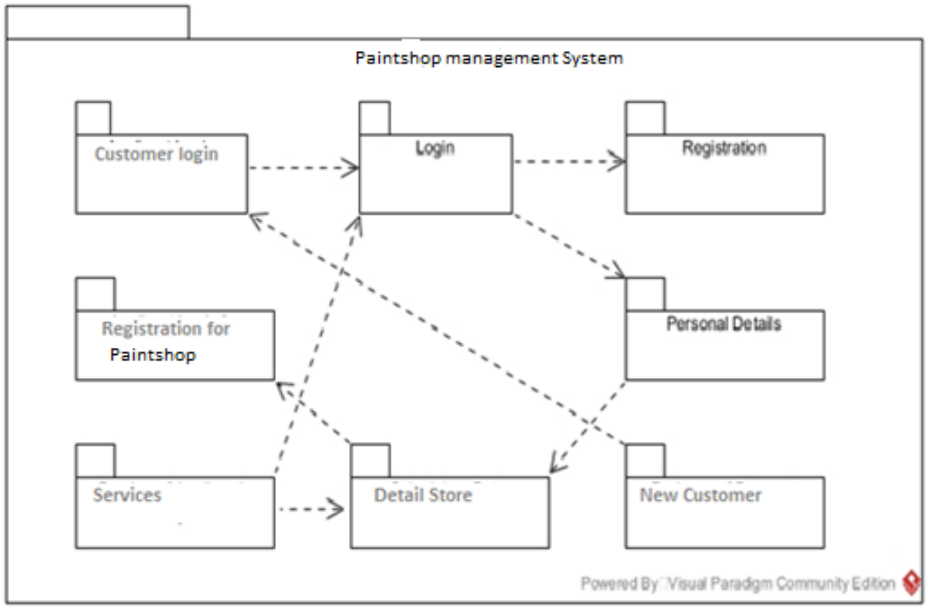


Figure 4.7 Package Diagram for Paintshop Management System

4.8 Component Diagram:

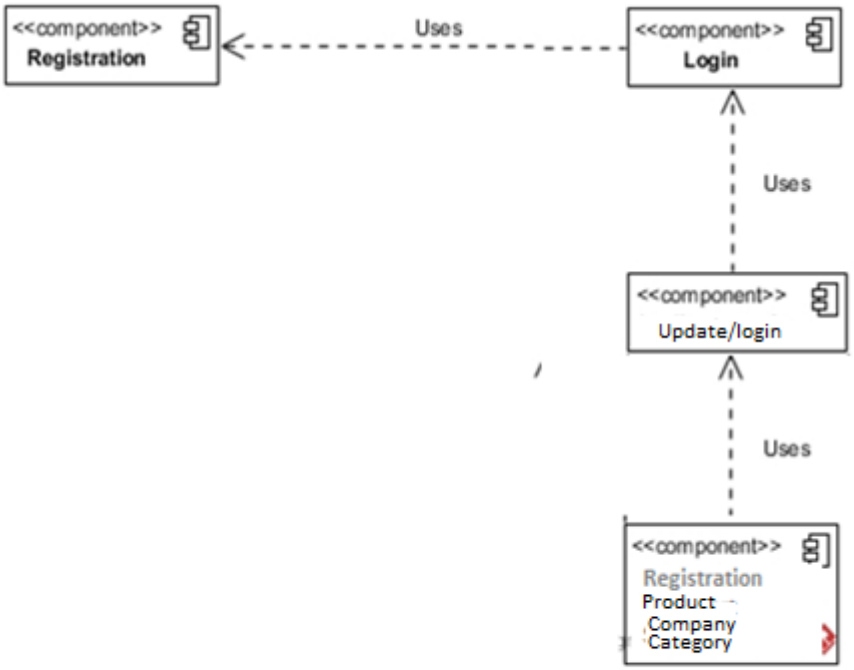


Figure 4.8 Component Diagram for Paintshop Management System

4.9 Deployment Diagram:

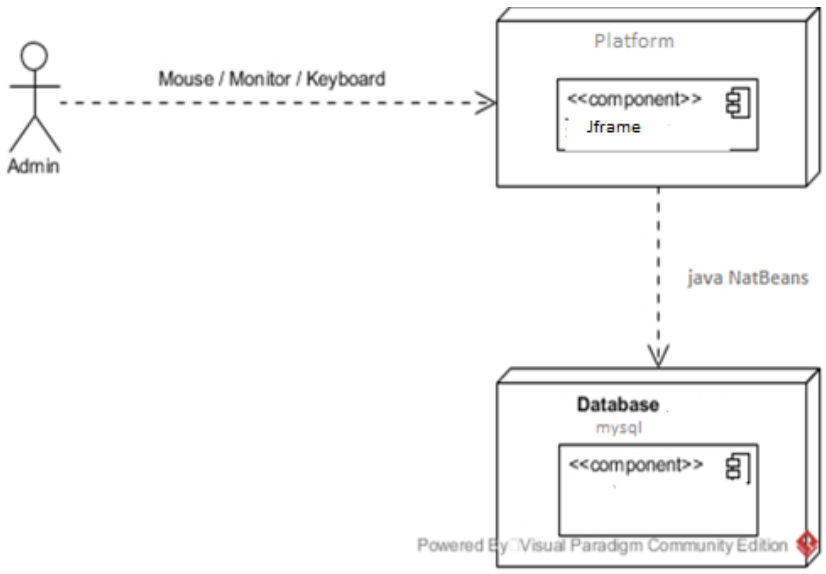


Figure 4.9 Deployment Diagram for Paintshop Management System

4.10 Data Flow Diagram:

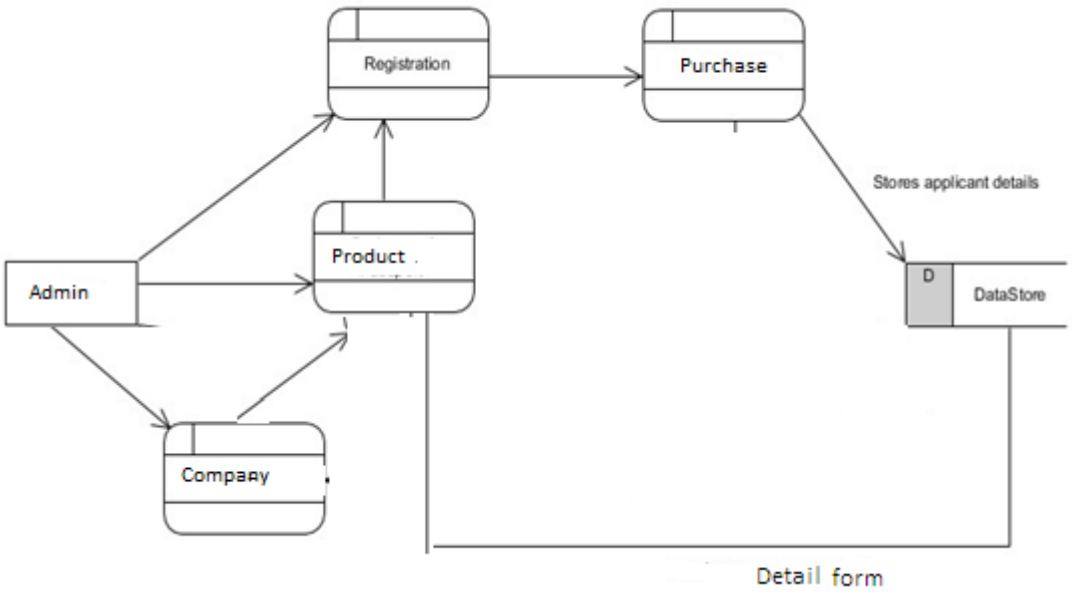


Figure 4.10 Data Flow Diagram for Paintshop Management System

4.11 Database Design:

4.11.1 RegLog:

Stores registration detail.

Field	Data Type	Description	Attributes
Name	Varchar	Store Name of user	Allow Null Field Size = 20
Email	Varchar	Stores Email Id.	Allow Null Field Size = 50
Address	Varchar	Stores Address	Allow Null Field Size = 100
Password	Varchar	Stores Password	Allow Null Field Size = 20
Mobile No.	Varchar	Stores Mobile number	Allow Null Field Size = 15

Table 4.11.1 Database for Storing Registration Details

4.11.2 Product Table:

Stores Product Form Detail:

Field	Data Type	Description	Attributes
Product Name	Varchar	Stores Date	Allow Null Field Size = 20
Quantity	Varchar	Stores Source	Allow Null Field Size = 50
Rate	Varchar	Stores Destinatio	Allow Null Field Size = 50

Table 4.11.2 Database for Storing Product Details

4.11.3 Company Details:

Stores Company Form Detail:

Field	Data Type	Description	Attributes
Name	Varchar	Stores Name	Allow Null Field Size = 20
Email	Varchar	Stores Email	Allow Null Field Size = 50
Address	Varchar	Stores Address	Allow Null Field Size = 100
Contact no	Integer	Store Contact no	Allow Null Field Size = 5
Country	Varchar	Store Country	Allow Null Field Size = 50

4.11.4: Category form

Stores Category:

Field	Data Type	Description	Attributes
Category Name	Varchar	Stores Name	Allow Null Field Size = 20

Table 4.11.2 Database for Storing Category Details

5.SYSTEM IMPLEMENTATION

Login:

```
package Forms;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
public class Log extends javax.swing.JFrame {
   ResultSet rs = null;
  Connection conn = null;
    PreparedStatement pst = null;
    public Log() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
}
  public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
```

```
try {
        conn = ConnectionToDatabase.connectToDb();
        String sql = "SELECT username,password FROM users
WHERE username=? AND password=?";
        pst = conn.prepareStatement(sql);
        pst.setString(1, jTextField1.getText());
        pst.setString(2, jPasswordField1.getText());
        rs = pst.executeQuery();
        if (rs.next()) {
           conn.close();
           JOptionPane.showMessageDialog(null, "Welcome " +
jTextField1.getText());

           MainPanel mp = new MainPanel();
           mp.loggedInAsLbl.setText(jTextField1.getText());
           this.dispose();
           mp.setVisible(true);
        } else {
           conn.close();
           JOptionPane.showMessageDialog(null, "Please check your
username and password and try again!");
        }
        } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e)
      }
```

Registration:

```
package Forms;
```

```
import javax.swing.UIManager;
import java.sql.*;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
public class UserRegistration extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
public UserRegistration() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
      this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
   }
private void registerBtnActionPerformed(java.awt.event.ActionEvent
evt) {
         try{
            String sql = "INSERT INTO users
(fullname,email,contactno,address,username,password,usertype)
VALUES (?,?,?,?,?,?,?)";
            pst = conn.prepareStatement(sql);
            pst.setString(1, fullnameTxt.getText());
            pst.setString(2, emailTxt.getText());
            pst.setString(3, contactNoTxt.getText());
            pst.setString(4, addressTxt.getText());
            pst.setString(5, userTxt.getText());
            pst.setString(6, passTxt.getText());
            pst.setString(7,
userTypeCombo.getSelectedItem().toString());
```

```
            pst.execute();
conn.close();
            JOptionPane.showMessageDialog(null, "Account has been
registered!");
            Log l=new Log();
            l.setVisible(true);
            this.dispose();
          }catch(Exception e){
            JOptionPane.showMessageDialog(null, e.getMessage());
          }
   }
```

Forgot password:

```
package ChangePassword;
import Forms.ConnectionToDatabase;
import Forms.MainPanel;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JOptionPane;
import java.sql.*;
import javax.swing.JFrame;
public class ChangePassword extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
public ChangePassword() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
```

```
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
        populateUser();
    }
    public void populateUser(){
        try{
            String sql = "SELECT username FROM users";
            pst = conn.prepareStatement(sql);
            rs = pst.executeQuery();
            while(rs.next()){
                usernameCombo.addItem(rs.getString("username"));
            }
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
        public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);
Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
   public void changePassword(){
       if (newPassTxt.getText().equals(repeatPassTxt.getText())){
          try{
            String sql = "UPDATE users SET
password='"+newPassTxt.getText()+"' WHERE
username='"+usernameCombo.getSelectedItem()+"'";
            pst = conn.prepareStatement(sql);
            pst.execute();
```

```
            JOptionPane.showMessageDialog(null, "Password has been
updated");
            }catch(Exception e){
               JOptionPane.showMessageDialog(null, e);
            }
        }else{
           JOptionPane.showMessageDialog(null, "New password and
Re-type Password don't match!");
        }
}
```

MainPanel:

```
package Forms;
import ChangePassword.ChangePassword;
import Companies.CompaniesPanel;
import Createuser.CreateUser;
import Forms.Product.ProductPanel;
import NewSalesRecord.NewSalesRecord;
import ProductCategory.AddCategory;
import SalesRecord.SalesRecord;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import javax.swing.table.TableColumn;
import net.proteanit.sql.DbUtils;
public class MainPanel extends javax.swing.JFrame  {
```

```
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
public MainPanel() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
        updateTable();
}
    public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);
Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
 private void updateTable(){
        try{
          String sql = "SELECT  product_name as 'Product
Name',company_name as 'Company Name',product_category as
'Product Category',quantity as 'Quantity',price_per_unit as 'Price Per
Unit' FROM products";
            pst = conn.prepareStatement(sql);
            rs = pst.executeQuery();

            medicineTable.setModel(DbUtils.resultSetToTableModel(rs));
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
```

```
    private void
medicineBtnActionPerformed(java.awt.event.ActionEvent evt) {
          ProductPanel mp = new ProductPanel();
        close();
        mp.setVisible(true);
    }
private void
companiesBtnActionPerformed(java.awt.event.ActionEvent evt) {
        CompaniesPanel cp = new CompaniesPanel();
        close();
        cp.setVisible(true);
    }
private void
viewSalesRecordsBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        SalesRecord sr = new SalesRecord();
        close();
        sr.setVisible(true);
    }
private void
newSalesBtnActionPerformed(java.awt.event.ActionEvent evt) {
        NewSalesRecord nsr =  new NewSalesRecord();
        close();
        nsr.setVisible(true);
    }
private void
changePasswordBtnActionPerformed(java.awt.event.ActionEvent evt)
{
        ChangePassword cp = new ChangePassword();
        close();
```

```
        cp.setVisible(true);
    }
private void logOutBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        Log l  = new Log();
        close();
        l.setVisible(true);
    }
Private void
createUserBtnActionPerformed(java.awt.event.ActionEvent evt) {
        CreateUser cu = new CreateUser();
        close();
        cu.setVisible(true);
    }
private void
addNewProductCategoryBtnActionPerformed(java.awt.event.ActionE
vent evt) {
          AddCategory ac = new AddCategory();
          ac.setVisible(true);
    }
private void searchTxtKeyReleased(java.awt.event.KeyEvent evt) {
   try{
         String sql = "SELECT  product_name as 'Product
Name',company_name as 'Company Name',product_category as
'Product Category',quantity as 'Quantity',price_per_unit as 'Price Per
Unit' FROM products WHERE product_name LIKE '%"+
searchTxt.getText() +"%'";
           pst = conn.prepareStatement(sql);
           rs = pst.executeQuery();
           medicineTable.setModel(DbUtils.resultSetToTableModel(rs));
```

```
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new MainPanel().setVisible(true);
            }
        });
    }
}
```

Company Panel:

```
package Companies;
import Forms.ConnectionToDatabase;
import Forms.MainPanel;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import javax.swing.JFrame;
public class CompaniesPanel extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
String companyName,country,email,contactNo,address;
```

```
String tableClicked = "";
  public CompaniesPanel() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
      this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
      updateTable();
  }
      public void close(){
      WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
  }
private void updateTable(){
try{
       String sql = "SELECT  company_name as 'Company
Name',country as 'Country',email as 'Email',contact_no as 'Contact
No.',address as 'Address' FROM company";
         pst = conn.prepareStatement(sql);
         rs = pst.executeQuery();
medicineTable.setModel(DbUtils.resultSetToTableModel(rs));
      }catch(Exception e){
         JOptionPane.showMessageDialog(null, e.getMessage());
      }
  }
private void
insertComBtnActionPerformed(java.awt.event.ActionEvent evt) {
     InsertCompany im = new InsertCompany();
```

```
        close();
        im.setVisible(true);
    }
    private void
updateComBtnActionPerformed(java.awt.event.ActionEvent evt) {
            if ("".equals(tableClicked)){
                JOptionPane.showMessageDialog(null, "Please select a
company to update!");
            }else{
              UpdateCompany up = new UpdateCompany();
                up.companyName = this.companyName;
                up.companyNameTxt.setText(companyName);
                up.countryTxt.setText(country);
                up.emailTxt.setText(email);
                up.contactNoTxt.setText(contactNo);
                up.addressTxt.setText(address);
                close();
                up.setVisible(true);
            }
      }
    private void delComBtnActionPerformed(java.awt.event.ActionEvent
evt) {
          if ("".equals(tableClicked)){
                JOptionPane.showMessageDialog(null, "Please select a
company to delete!");
          }else{
              int des = JOptionPane.showConfirmDialog(null, "Are you
sure you want to delete this
company?","Delete",JOptionPane.YES_NO_OPTION);
```

```
            if (des==0){
                try{
                    String sql = "DELETE FROM company WHERE
company_name='"+tableClicked+"'";
                    pst = conn.prepareStatement(sql);
                    pst.execute();
                    pst.close();
                    rs.close();
                    updateTable();
                    JOptionPane.showMessageDialog(null, "Company has
been deleted!");
                }catch(Exception e){
                 JOptionPane.showMessageDialog(null, e.getMessage());
                 }
            }


        }
    }
private void searchBtnActionPerformed(java.awt.event.ActionEvent
evt) {
}
private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
      MainPanel mp = new MainPanel();
       close();
       mp.setVisible(true);
}
```

Insert Company:

```
package Companies;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;
import java.sql.*;
import javax.swing.JOptionPane;
public class InsertCompany extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;

    public InsertCompany() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
    }
   public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }

    public void clearData(){
       companyNameTxt.setText("");
```

```
        countryTxt.setText("");
        emailTxt.setText("");
        contactNoTxt.setText("");
        addressTxt.setText("");
    }
    private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        CompaniesPanel mp = new CompaniesPanel();
        close();
        mp.setVisible(true);
    }

    private void insertBtnActionPerformed(java.awt.event.ActionEvent
evt) {
            try{
                String sql = "INSERT INTO company
(company_name,country,email,contact_no,address) VALUES
(?,?,?,?,?)";
                pst = conn.prepareStatement(sql);
                pst.setString(1,companyNameTxt.getText());
                pst.setString(2,countryTxt.getText());
                pst.setString(3,emailTxt.getText());
                pst.setString(4,contactNoTxt.getText());
                pst.setString(5,addressTxt.getText());
                pst.execute();
                conn.close();
                JOptionPane.showMessageDialog(null, "Company has been
added!");
                clearData();
```

```
        }catch(Exception e){
           JOptionPane.showMessageDialog(null, e);
        }
    }
```

Update Company:

```
package Companies;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
public class UpdateCompany extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
String companyName;
public UpdateCompany() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
   }
  public void close(){
      WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);
```

```
Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
private void updateBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        try{
            String sql  = "UPDATE company SET company_name='"+
companyNameTxt.getText() +"',country='"+ countryTxt.getText()
+"',email='"+ emailTxt.getText() +"',contact_no='"+
contactNoTxt.getText() +"',address='"+ addressTxt.getText() +"'
WHERE company_name='"+ companyName +"'";
            pst = conn.prepareStatement(sql);
            pst.execute();
            conn.close();
            JOptionPane.showMessageDialog(null, "Company has been
updated!");
            companyName = companyNameTxt.getText();
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }

    private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        CompaniesPanel mp = new CompaniesPanel();
        close();
        mp.setVisible(true);
    }
```

Insert Product:

```
package Forms.Product;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;
import java.sql.*;
import javax.swing.JOptionPane;
public class InsertProduct extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
public InsertProduct() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
        populateCompany();
        populateCategory();
    }
   public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
public void clearData(){
```

```
        productNameTxt.setText("");
        companyNameTxt.setSelectedIndex(0);
        productCategoryTxt.setSelectedIndex(0);
        productQuantityTxt.setText("");
        productPriceTxt.setText("");
    }
    public void populateCategory(){
        try{
            String sql = "SELECT category FROM product_category";
            pst = conn.prepareStatement(sql);
            rs = pst.executeQuery();
            while(rs.next()){
                productCategoryTxt.addItem(rs.getString("category"));
            }
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
public void populateCompany(){
        try{
            String sql = "SELECT company_name FROM company";
            pst = conn.prepareStatement(sql);
            rs = pst.executeQuery();
             while(rs.next()){
               companyNameTxt.addItem(rs.getString("company_name"));
             }
        }catch(Exception e){
```

```
            JOptionPane.showMessageDialog(null, e);
        }
    }
    private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        ProductPanel mp = new ProductPanel();
        close();
        mp.setVisible(true);
    }
private void insertBtnActionPerformed(java.awt.event.ActionEvent
evt) {
            try{
                String sql = "INSERT INTO products
(product_name,company_name,product_category,quantity,price_per_
unit) VALUES (?,?,?,?,?)";
                pst = conn.prepareStatement(sql);
                pst.setString(1,productNameTxt.getText());

pst.setString(2,companyNameTxt.getSelectedItem().toString());

pst.setString(3,productCategoryTxt.getSelectedItem().toString());
                pst.setString(4,productQuantityTxt.getText());
                pst.setString(5,productPriceTxt.getText());
                pst.execute();
                conn.close();
                JOptionPane.showMessageDialog(null, "Products has been
added!");
                clearData();
            }catch(Exception e){
                JOptionPane.showMessageDialog(null, e);
```

```
        }
    }
```

Product Panel:

```
package Forms.Product;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JFrame;
import java.sql.*;
import javax.swing.JOptionPane;
public class InsertProduct extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
public InsertProduct() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
        populateCompany();
        populateCategory();
    }
   public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
```

```
    public void clearData(){
      productNameTxt.setText("");
      companyNameTxt.setSelectedIndex(0);
      productCategoryTxt.setSelectedIndex(0);
      productQuantityTxt.setText("");
      productPriceTxt.setText("");
    }
    public void populateCategory(){
       try{
          String sql = "SELECT category FROM product_category";
          pst = conn.prepareStatement(sql);
          rs = pst.executeQuery();
          while(rs.next()){
             productCategoryTxt.addItem(rs.getString("category"));
          }
       }catch(Exception e){
          JOptionPane.showMessageDialog(null, e);
       }
    }
  public void populateCompany(){
       try{
          String sql = "SELECT company_name FROM company";
          pst = conn.prepareStatement(sql);
          rs = pst.executeQuery();
           while(rs.next()){
             companyNameTxt.addItem(rs.getString("company_name"));
```

```
            }
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_
ON_CLOSE);
        setResizable(false);
        backBtn.setText("Back");
        backBtn.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt)
{
                backBtnActionPerformed(evt);
            }
        });
        insertBtn.setText("Insert");
        insertBtn.addActionListener(new java.awt.event.ActionListener()
{
            public void actionPerformed(java.awt.event.ActionEvent evt)
{
insertBtnActionPerformed(evt);
            }
        });
private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        ProductPanel mp = new ProductPanel();
        close();
        mp.setVisible(true);
    }
```

```
   private void insertBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        try{
            String sql = "INSERT INTO products
(product_name,company_name,product_category,quantity,price_per_
unit) VALUES (?,?,?,?,?)";
            pst = conn.prepareStatement(sql);
            pst.setString(1,productNameTxt.getText());

pst.setString(2,companyNameTxt.getSelectedItem().toString());

pst.setString(3,productCategoryTxt.getSelectedItem().toString());
            pst.setString(4,productQuantityTxt.getText());
            pst.setString(5,productPriceTxt.getText());
            pst.execute();
            conn.close();
            JOptionPane.showMessageDialog(null, "Products has been
added!");
            clearData();
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
   }
Update Product: package Forms.Product;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
```

```
import javax.swing.JFrame;
import javax.swing.JOptionPane;
public class UpdateProduct extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
String productName;
public UpdateProduct() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
        this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
   }
  public void close(){
      WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);
Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
   }
      updateBtn.setText("Update");
      updateBtn.addActionListener(new
java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt)
{
              updateBtnActionPerformed(evt);
          }
      });
updateBtnActionPerformed (java.awt.event.ActionEvent evt) {
       try{
```

```
            String sql  = "UPDATE products SET product_name='"+
productNameTxt.getText()
                    +"',company_name='"+ companyNameTxt.getText()
+"',product_category='"+
                    productCategoryTxt.getText() +"',quantity='"+
productQuantityTxt.getText() +
                    "',price_per_unit='"+ productPriceTxt.getText() +"'
WHERE product_name='"+ productName +"'";
            pst = conn.prepareStatement(sql);
            pst.execute();
            conn.close();
            JOptionPane.showMessageDialog(null, "Product has been
updated!");
            productName = productNameTxt.getText();
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
        ProductPanel mp = new ProductPanel();
        close();
        mp.setVisible(true);
    }
```

Sale Record:

```
package SalesRecord;
import Companies.*;
import Forms.ConnectionToDatabase;
import Forms.MainPanel;
```

```
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import javax.swing.JOptionPane;
import net.proteanit.sql.DbUtils;
import java.sql.*;
import javax.swing.JFrame;
public class SalesRecord extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst = null;
ResultSet rs = null;
String tableClicked = "";
   public SalesRecord() {
      initComponents();
      conn = ConnectionToDatabase.connectToDb();
      this.setExtendedState(this.getExtendedState() |
JFrame.MAXIMIZED_BOTH);
      updateTable();
   }
      public void close(){
      WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
   }
private void updateTable(){
  try{
        String sql = "SELECT  sales_id as 'ID',product_name as
'Product Name',company_name as 'Company Name',date_of_sale as
```

```
'Date of Sale',quantity as 'Quantity',amount as 'Amount' FROM
sales_record";
            pst = conn.prepareStatement(sql);
            rs = pst.executeQuery();
            medicineTable.setModel(DbUtils.resultSetToTableModel(rs));
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e.getMessage());
        }
    }
        searchBtn.setText("Search");
        searchBtn.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt)
{
                searchBtnActionPerformed(evt);
            }
        });
delRecordBtnActionPerformed(java.awt.event.ActionEvent evt) {
            if ("".equals(tableClicked)){
                JOptionPane.showMessageDialog(null, "Please select a
record to delete!");
            }else{
                try{
                    int des = JOptionPane.showConfirmDialog(null, "Are
you sure you want to delete this
record?","Delete",JOptionPane.YES_NO_OPTION);
                    if (des==0){
                        try{
                            String sql = "DELETE FROM sales_record
WHERE sales_id="'+tableClicked+"'";
```

```
                    pst = conn.prepareStatement(sql);
                    pst.execute();
                    pst.close();
                    rs.close();
                    updateTable();
                    JOptionPane.showMessageDialog(null, "Record
has been deleted!");
                }catch(Exception e){
                    JOptionPane.showMessageDialog(null,
e.getMessage());
                }
            }
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
private void searchBtnActionPerformed(java.awt.event.ActionEvent
evt) {


}
private void backBtnActionPerformed(java.awt.event.ActionEvent
evt) {
    MainPanel mp = new MainPanel();
    close();
    mp.setVisible(true);
}
private void formMouseClicked(java.awt.event.MouseEvent evt) {
}
```

Add Category:

```
package ProductCategory;
import Forms.ConnectionToDatabase;
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.*;
import javax.swing.JOptionPane;
public class AddCategory extends javax.swing.JFrame {
Connection conn = null;
PreparedStatement pst =null;
ResultSet rs = null;
public AddCategory() {
        initComponents();
        conn = ConnectionToDatabase.connectToDb();
    }
     public void close(){
        WindowEvent we = new
WindowEvent(this,WindowEvent.WINDOW_CLOSING);

Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(we);
    }
private void
addCategoryBtnActionPerformed(java.awt.event.ActionEvent evt) {
        try{
            String sql = "INSERT INTO product_category(category)
VALUES (?)";
            pst = conn.prepareStatement(sql);
```

```
            pst.setString(1, categoryTxt.getText());
            pst.execute();
            JOptionPane.showMessageDialog(null, "Product Category
Added!");
            close();
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

6.RESULTS

6.1 Validations:

6.1.1 User Reg Page Validations:

Name	This field should not be empty.
Address	This field should not be empty.
Mobile number	This field should not be empty.
Email Id	This field should not be empty. Please enter a valid Email-Id.
Login Id	This field should not be empty.
Password	This field should not be empty.

Table 6.1.1 User Register Page Validations

6.1.2 Login Page Validations:

Login Id	This field should not be empty.
Password	This field should not be empty.

Table 6.1.2 Login Page Validations

6.1.3 Details Validations:

Name	This field should not be empty..
Email	This field should not be empty.
Address	This field should not be empty.
Category	.This field should not be empty.
Product	This field should not be empty.
Company	This field should not be empty. .

Table 6.1.3 Applicant Product Validations

6.1.4Login Admin Validations:

Login Id	This field should not be empty.
Password	This field should not be empty.

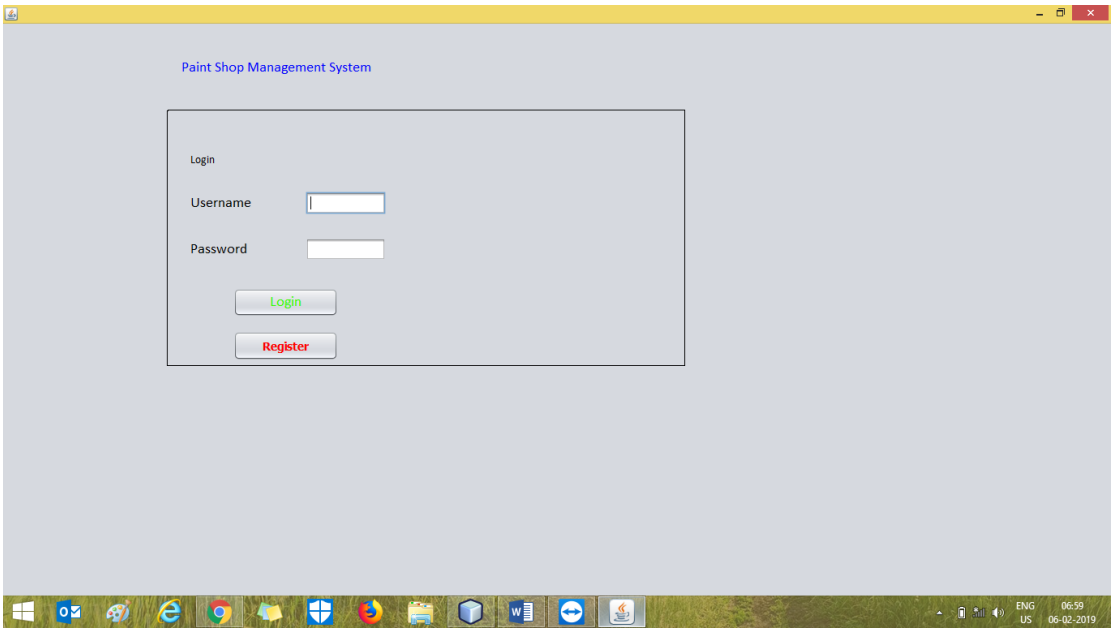
6.1.5 Naming Conventions:

No.	Page Name	Description
1.	Home	It is the main page for this website where applicant can avail the features
2.	User Reg	This page provides the registration form.
3.	Login	Applicants who have registered could get logged in through this page.
4.	Company	Eligible applicants passed further to enter their personal details into this page.
5.	Category	This page ask about location

Table 6.1.5 Naming Conventions for PaintShop Management System

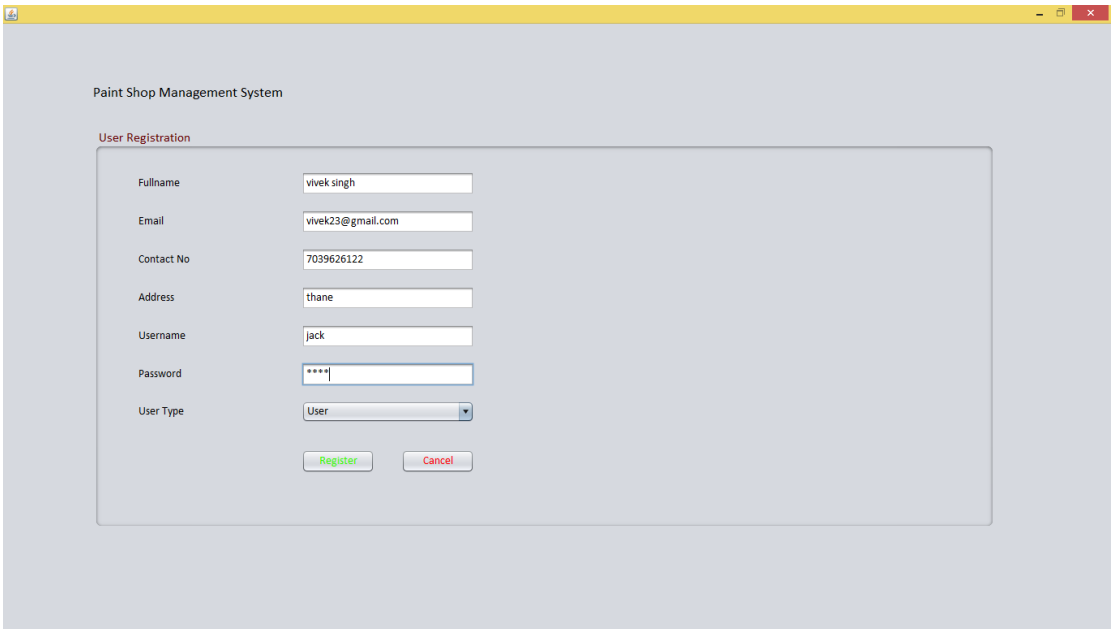
6.2 Screenshots:

Login:



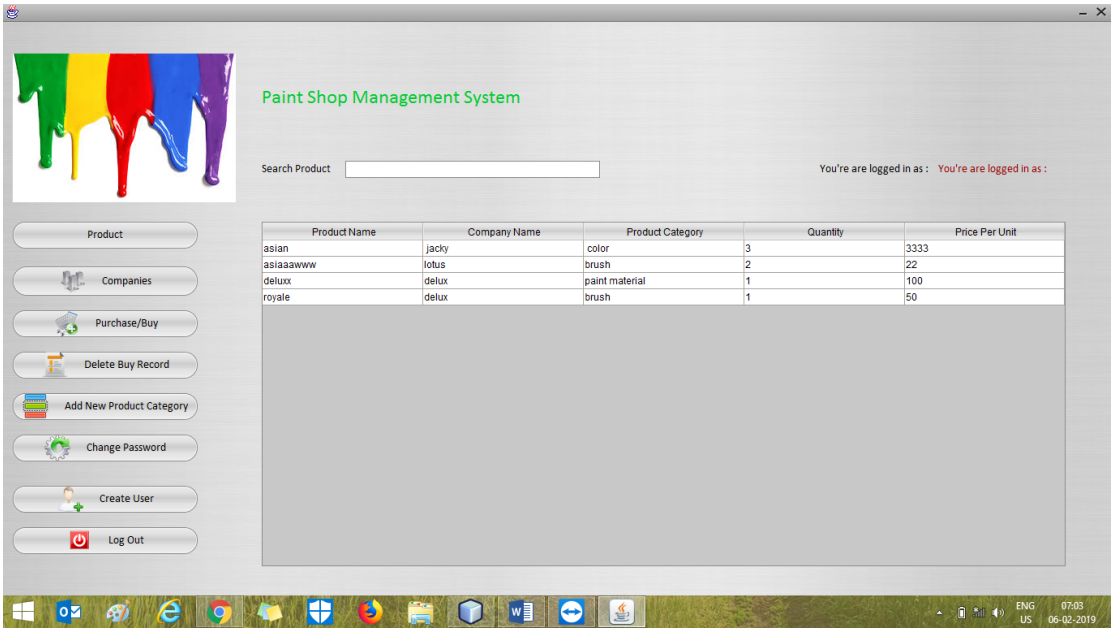
Screenshot 6.2.1: Login Page

User Registeration:



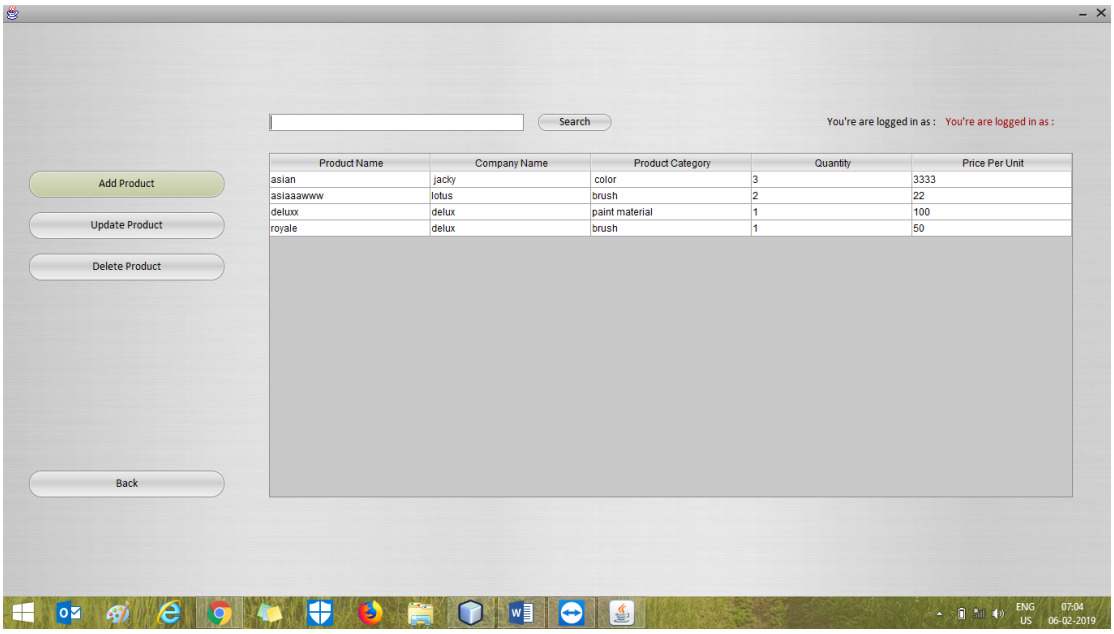
Screenshot 6.2.2: User Registration

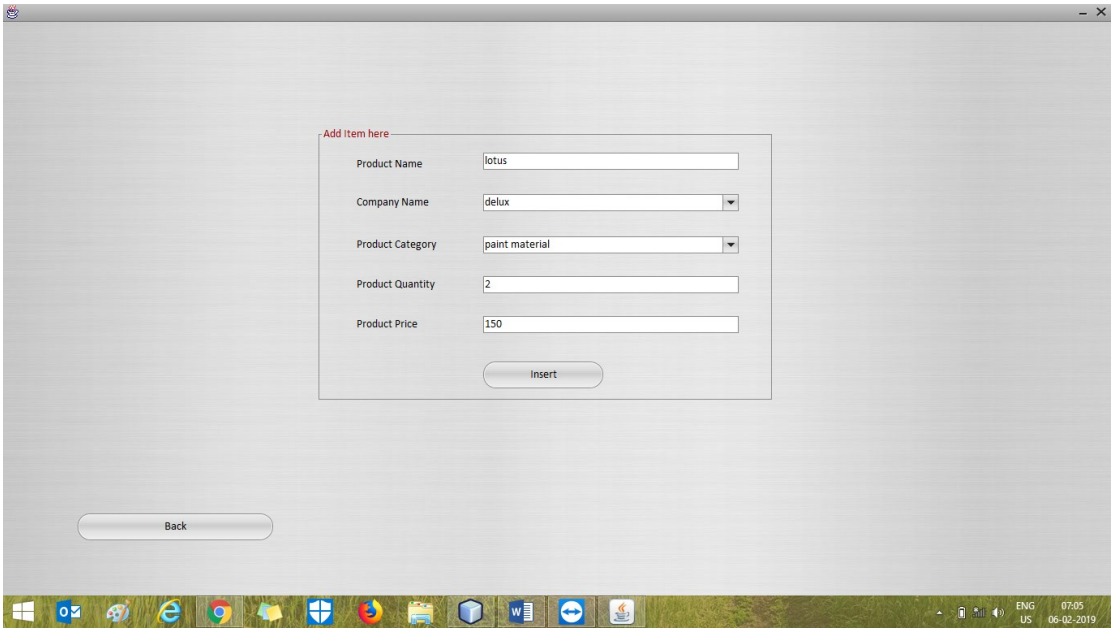
HomePage:



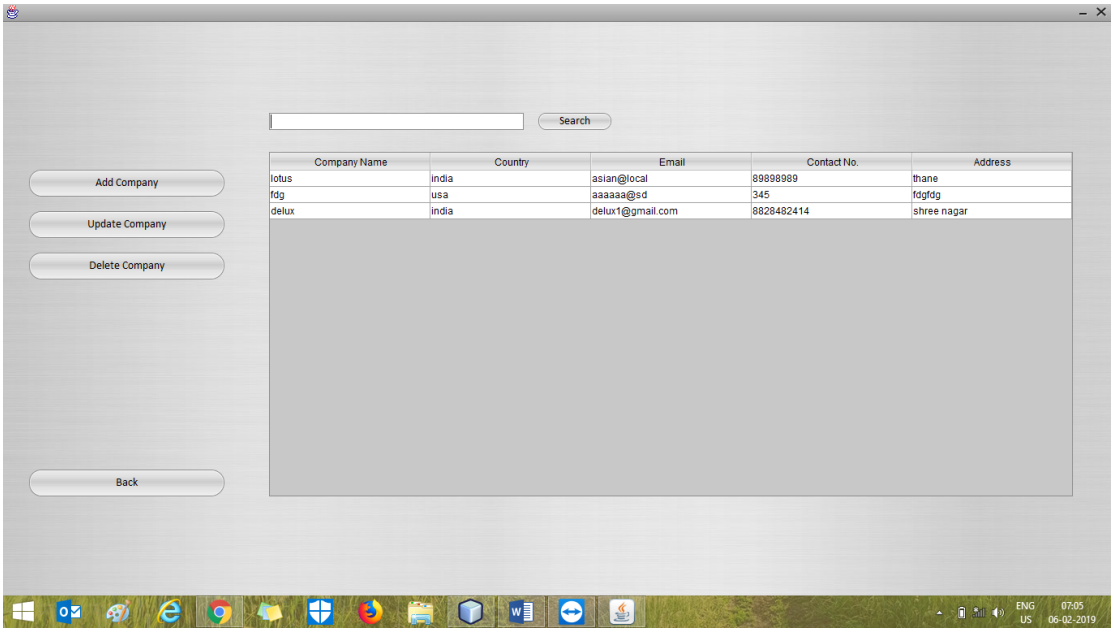
Screenshot 6.2.3 Home

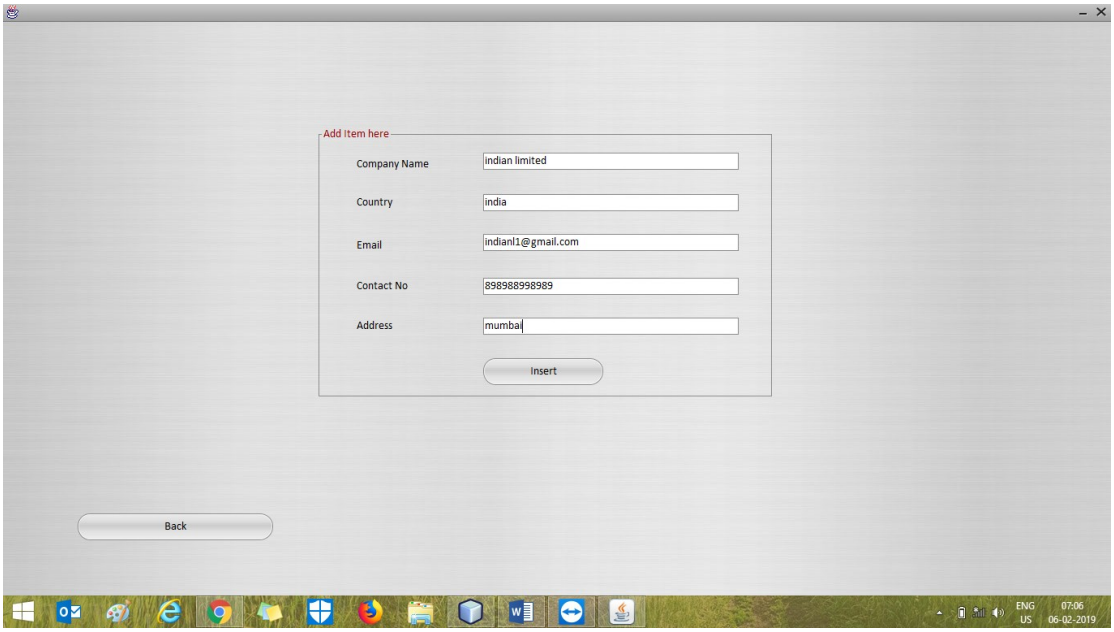
Add Product :





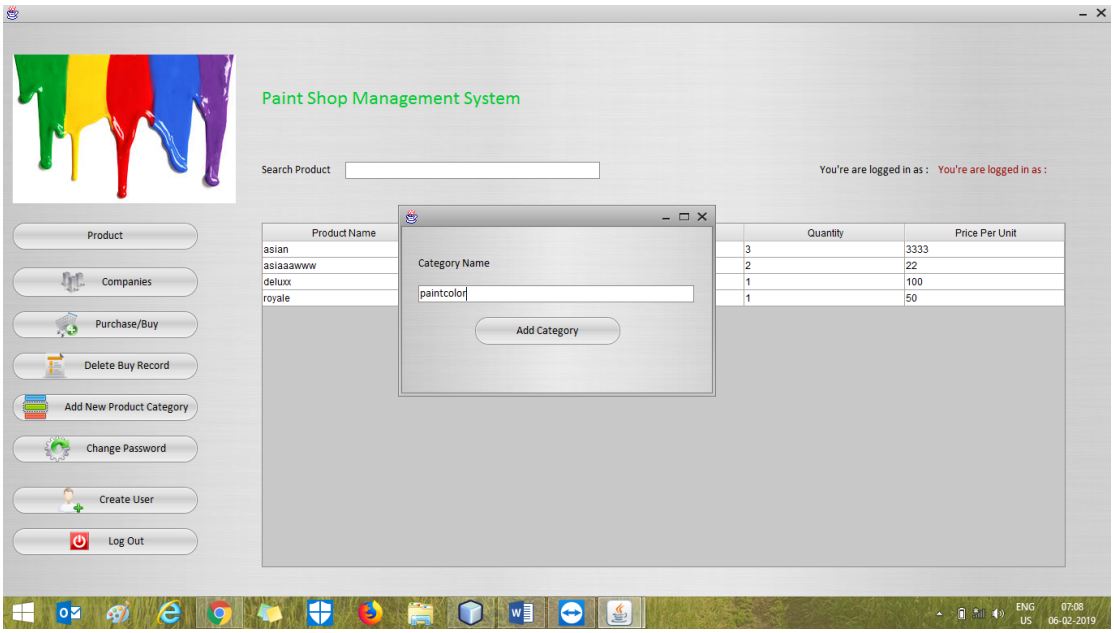
Screenshot 6.2.4 Product



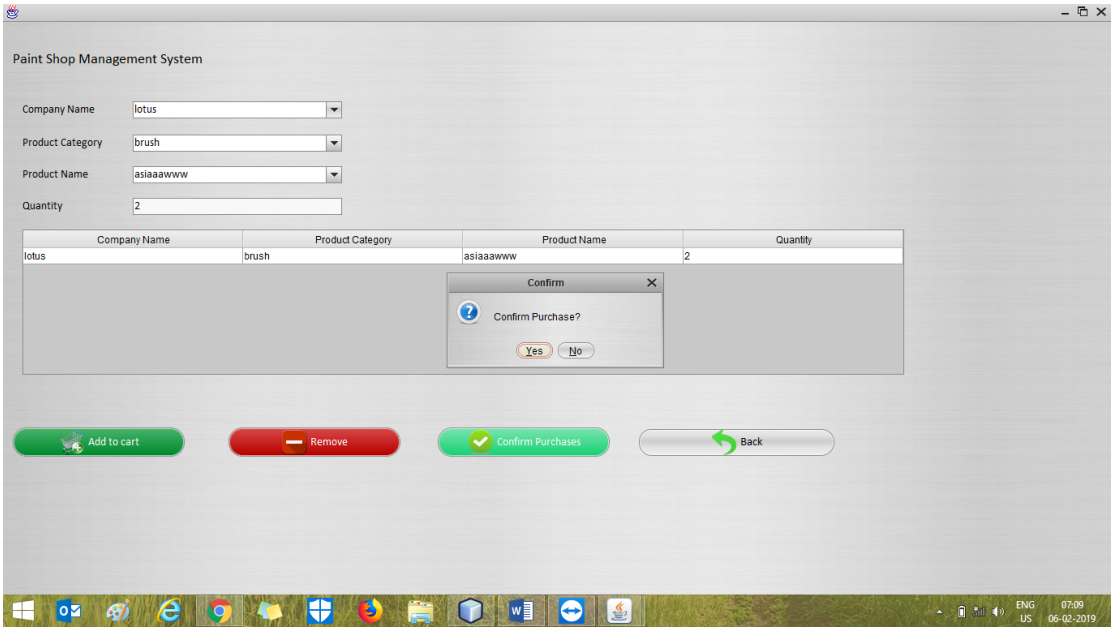


Screenshot 6.2.5 Company

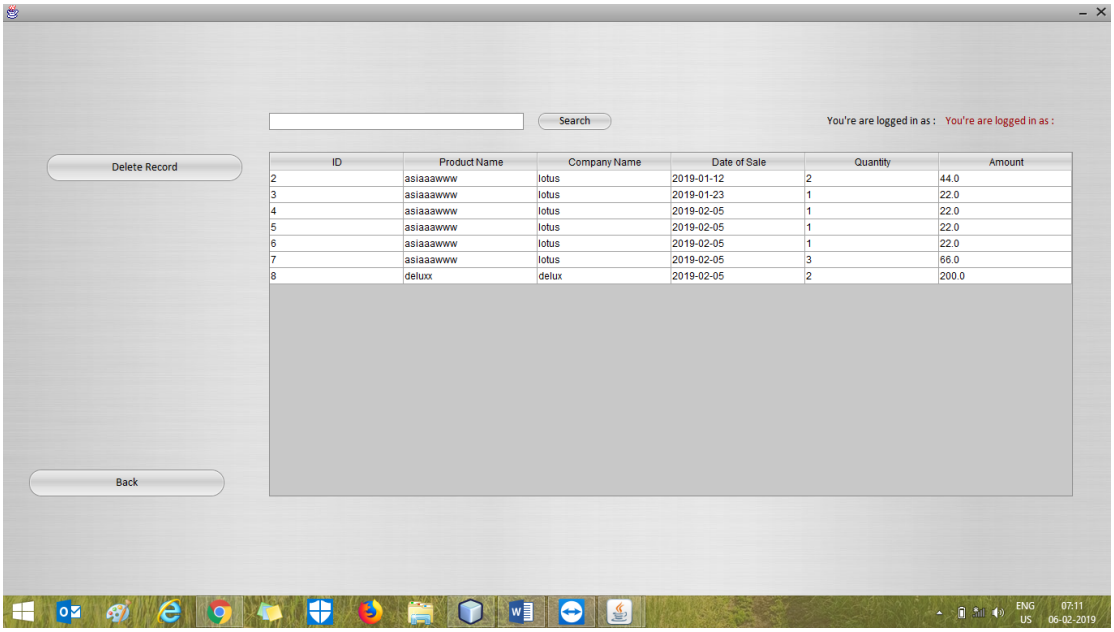
Add Category:



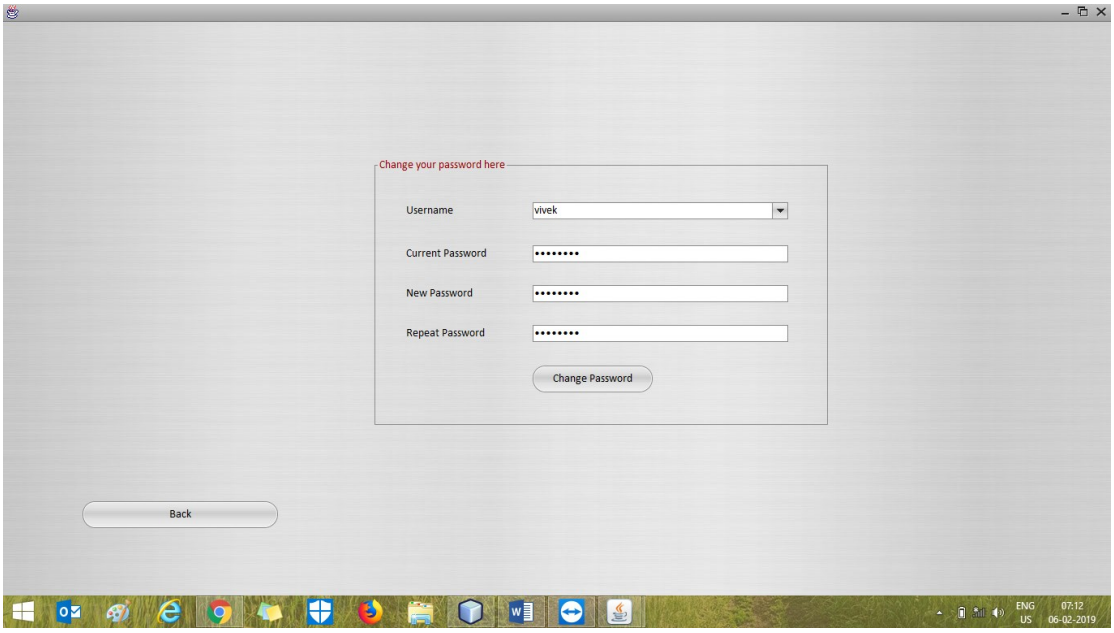
Screenshot 6.2.6 Category form



Screenshot 6.2.7 Purchase material

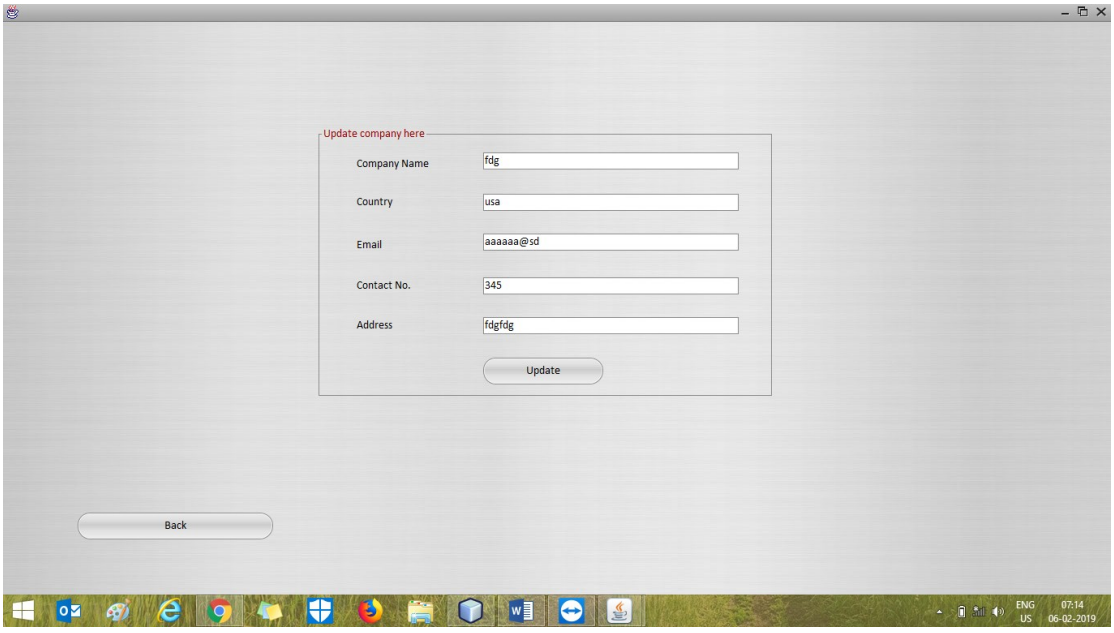


Screenshot 6.2.8 Delete Records



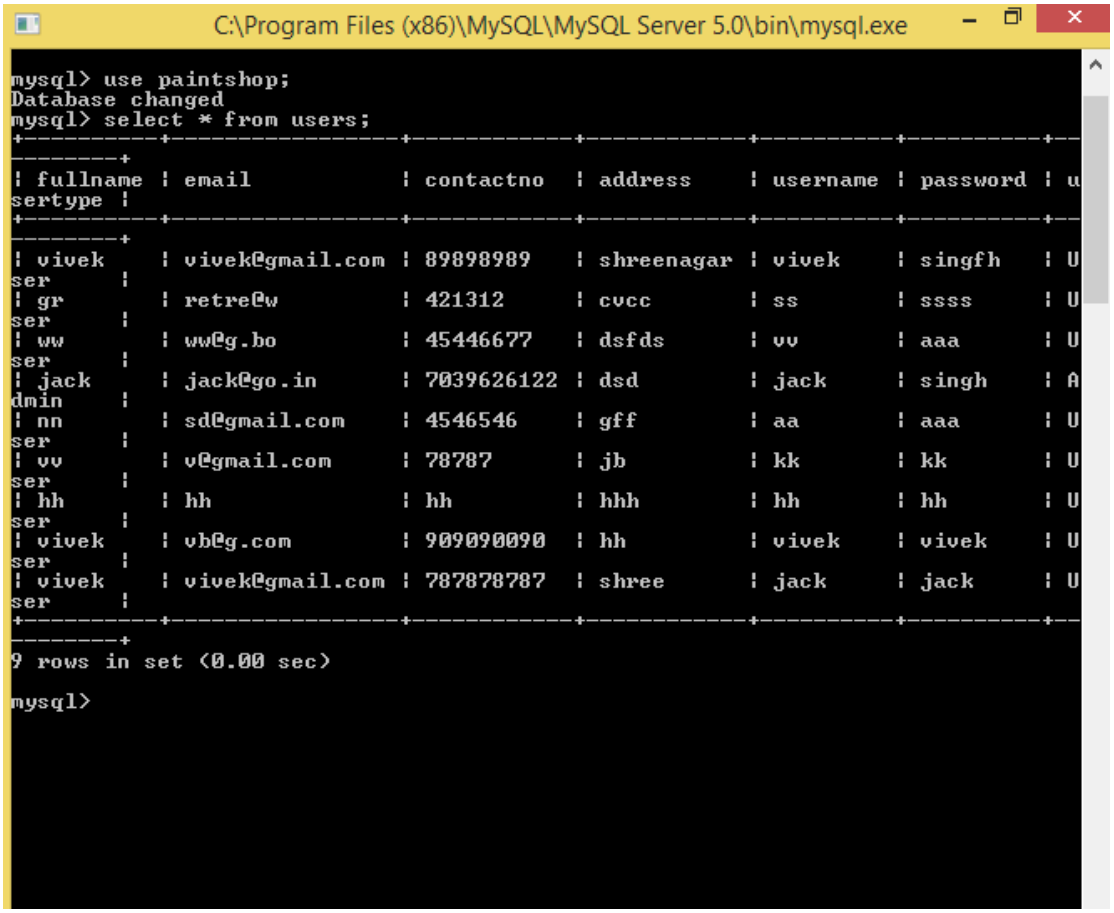
Screenshot 6.2.9 Change Password page

Update Company:



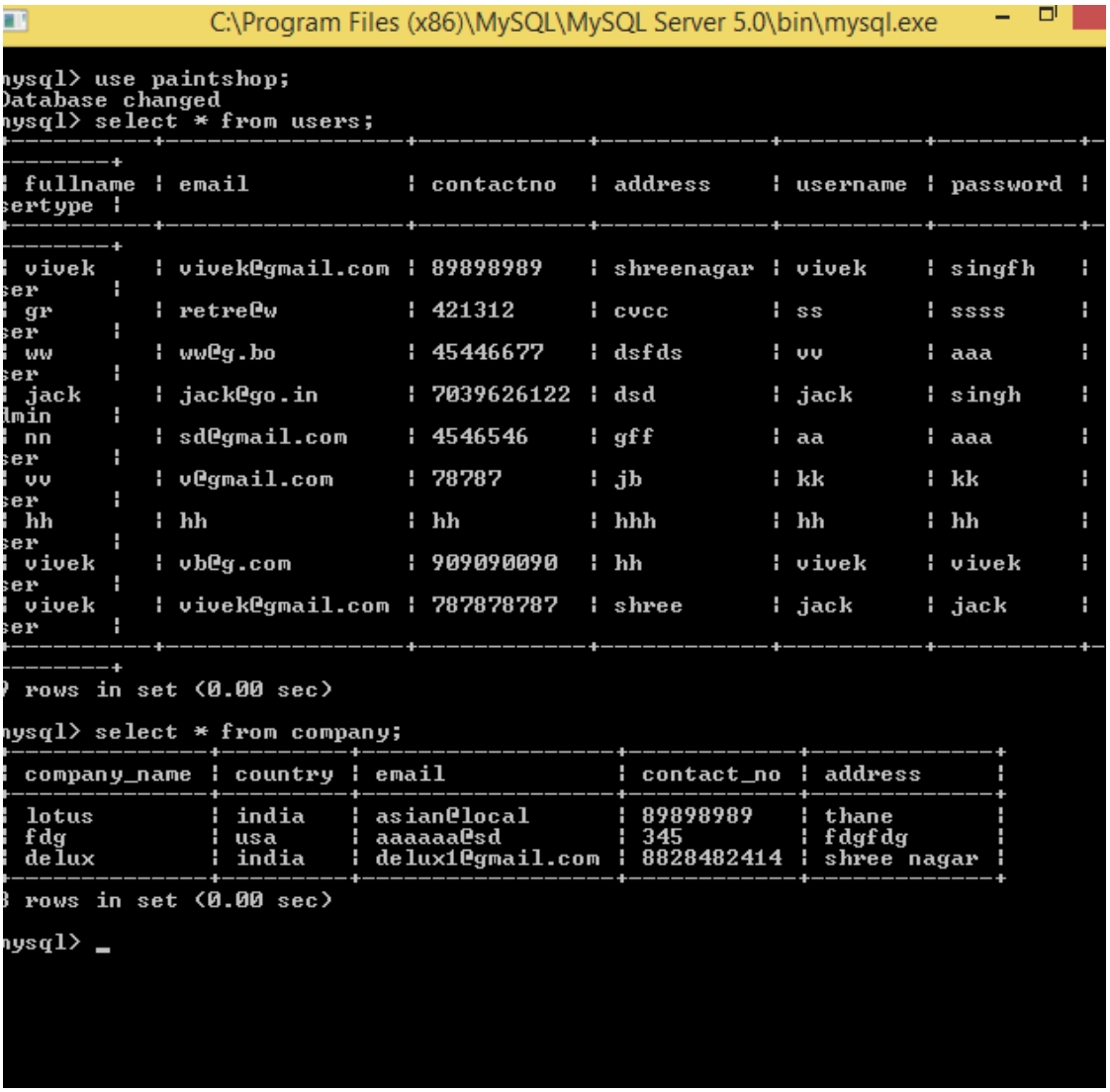
Screenshot 6.2.10 Update Company

Database:



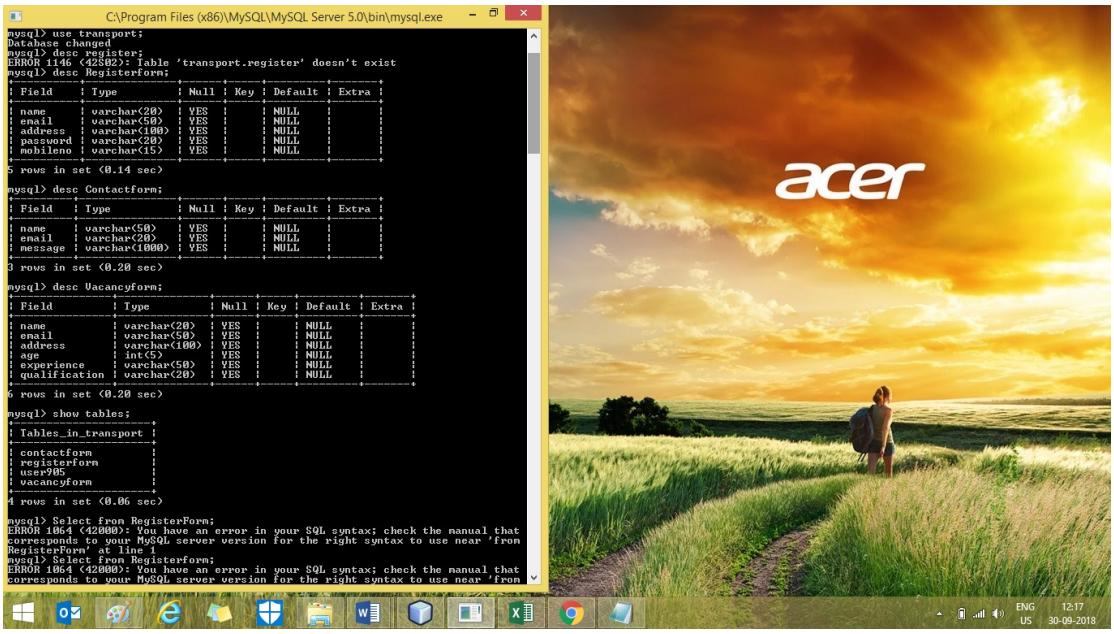
Screenshot 6.2.11 Database

Company Database:



Company database 6.2.14

RegLog Database:



Screenshot 6.2.15 Database to store Registration Details

7.CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion:

According to our system we are having some features which will make users to access the System smoothly through this Project customer/Admin get idea how to access the Paintshop management system.

This Paintshop management System fullfil the all customer needs.And this system maintain all the record and customer detail to overcome with his/her ideas about the paintshop management System.

This System must be send all the detail through email and various facilities this system must we verify all the details and then decide whether user are doing correctly or not.

7.2 Future Enhancements:

- Billing based System
- Receiving Payment Cash.
- All Record Maintain.

8.REFERENCES

8.1 Web References:

- https://www.youtube.com/
- https://docs.javapoint /tutorials

ANNEXURE

List of Figures:

List of Tables:

List of Screenshots: