Cos-106byjade

by Nandar Aung

Submission date: 28-Jan-2024 10:41PM (UTC+0800)
Submission ID: 2279697802
File name: COS106-_Programming_with_Java-3.docx (17.79M)
Word count: 892
Character count: 4719

COS106- Programming with Java

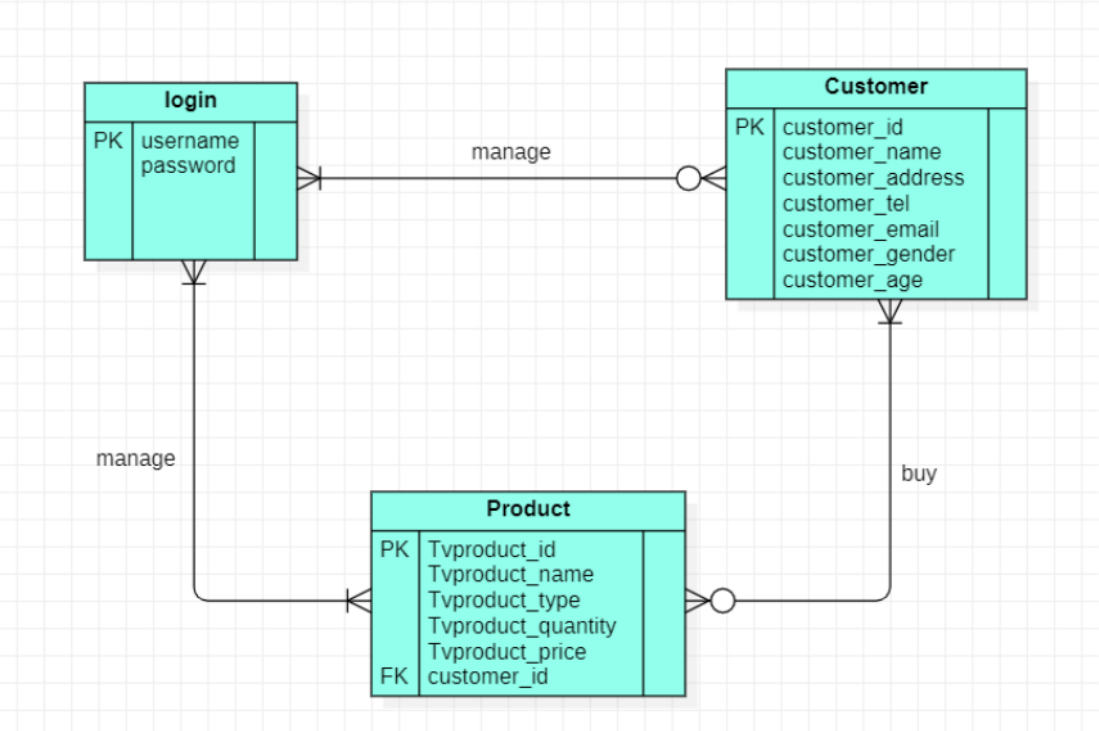
British University College
NandarAung (1334)

Table of Contents

ABOUT PROGRAM

The TV Repair System is a software application designed to assist TV repair shops in managing customer information, details of TV repairs, and work on TVs. Developed using the Java programming language for user-friendly and efficient functionality, the application consists of five forms: table creation, customer management, item management, column manipulation, and admin login. Admin users, in the client and item management forms, possess the authority to perform actions such as adding, updating, removing, and searching data, along with additional privileges. Administrators can search, modify, and remove data from the client and item forms. The create table form enables admins to modify table columns and create new tables in the database. The column form allows administrators to add, edit, and remove data from table columns. Admins can manipulate data in tables, including adding, updating, and deleting, using the data manipulation form. The application features screenshots of code snippets, and each form is linked to a database. The database has separate sections for managing logins, clients, and items, with diagrams showing how they are connected. There are rules to check that the data entered for logins, clients, and items is accurate. Moreover, the application has features to automatically create code, display selected information on the screen, and assist in adding new items to forms.

ENTITY-RELATIONSHIP DIAGRAM



Multiple users can access the program, as indicated in the diagram. The company may have zero or numerous customers. At a minimum, the business will market one or more products, and customers have the option to purchase one, multiple, or no products.

Designing Databases and Data

- **Login Table (userlogin)**

```
mysql> create database RepairDB;
Query OK, 1 row affected (0.00 sec)

mysql> shomysql>
mysql>
mysql> CREATE TABLE login (
    ->     username VARCHAR(10),
    ->     password VARCHAR(20)
    -> );
ERROR 1046 (3D000): No database selected
mysql> use RepairDB;
Database changed
mysql> CREATE TABLE login (
    ->     username VARCHAR(10),
    ->     password VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO login (username, password) VALUES
    ->     ('jade', '2024'),
    ->     ('justin', '2025'),
    ->     ('nick', '2026');
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0
```

```
mysql> select * from login;
+----------+----------+
| username | password |
+----------+----------+
| jade     | 2024     |
| justin   | 2025     |
| nick     | 2026     |
+----------+----------+
3 rows in set (0.00 sec)

mysql>
```

- **Customer Table (customer)**

```
mysql> desc customer;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| customer_id      | varchar(10)  | NO   | PRI | NULL    |       |
| customer_name    | varchar(255) | YES  |     | NULL    |       |
| customer_address | varchar(255) | YES  |     | NULL    |       |
| customer_tel     | int          | YES  |     | NULL    |       |
| customer_email   | varchar(255) | YES  |     | NULL    |       |
| customer_gender  | char(20)     | YES  |     | NULL    |       |
| customer_age     | int          | YES  |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
7 rows in set (0.01 sec)

mysql>
```

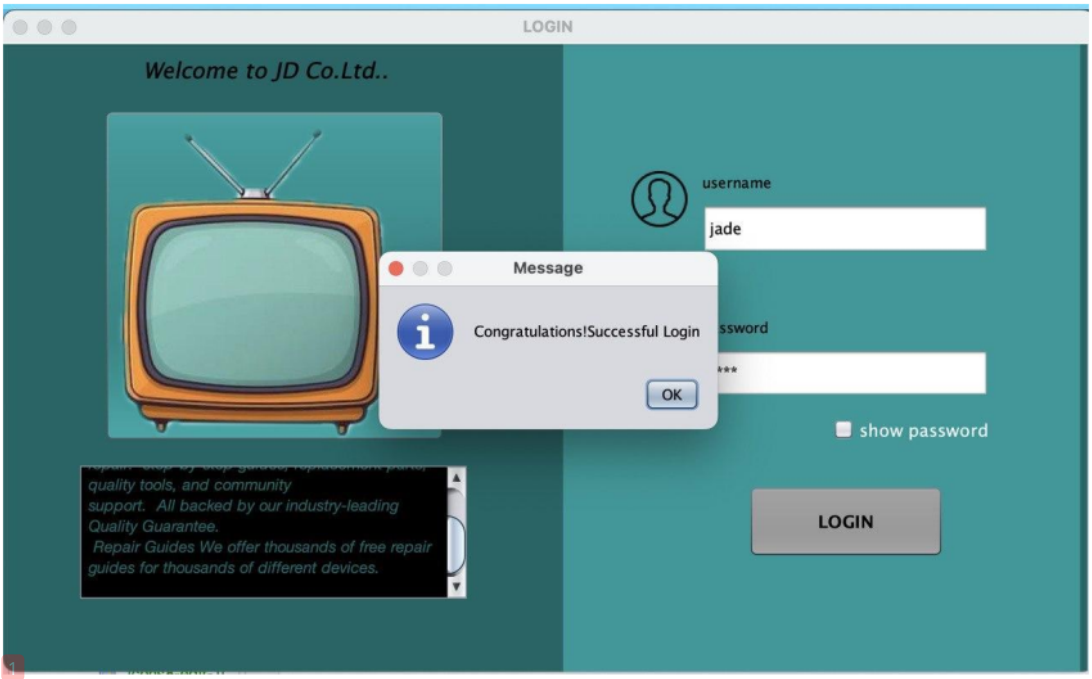
- **tv Product Table**

```
mysql> desc product;
+--------------------+--------------+------+-----+---------+-------+
| Field              | Type         | Null | Key | Default | Extra |
+--------------------+--------------+------+-----+---------+-------+
| product_id         | varchar(10)  | NO   | PRI | NULL    |       |
| product_name       | varchar(255) | YES  |     | NULL    |       |
| tvproduct_type     | varchar(50)  | YES  |     | NULL    |       |
| tvproduct_quantity | int          | YES  |     | NULL    |       |
| tvproduct_price    | int          | YES  |     | NULL    |       |
| customer_id        | varchar(10)  | YES  |     | NULL    |       |
+--------------------+--------------+------+-----+---------+-------+
6 rows in set (0.01 sec)

mysql>
```

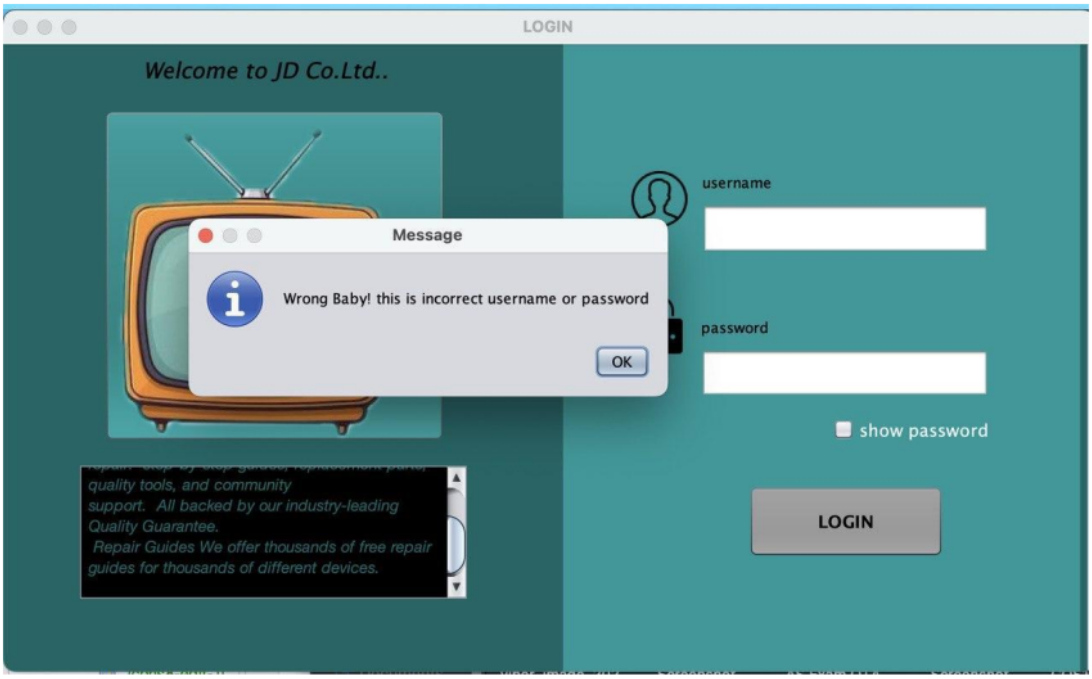
```
mysql> select * from product;
+------------+------------------+---------------+-------------------+---------------+-------------+
| product_id | product_name     | tvproduct_type | tvproduct_quantity | tvproduct_price | customer_id |
+------------+------------------+---------------+-------------------+---------------+-------------+
| 1          | Smart TV 55"     | LED           |                 2 |           800 | 1           |
| 2          | 4K OLED TV 65"   | OLED          |                 1 |          1500 | 2           |
| 3          | HD TV 32"        | LCD           |                 3 |           300 | 3           |
| 4          | Curved QLED TV 75" | QLED          |                 1 |          2500 | 4           |
+------------+------------------+---------------+-------------------+---------------+-------------+
4 rows in set (0.01 sec)

mysql>
```

Coding and Validation

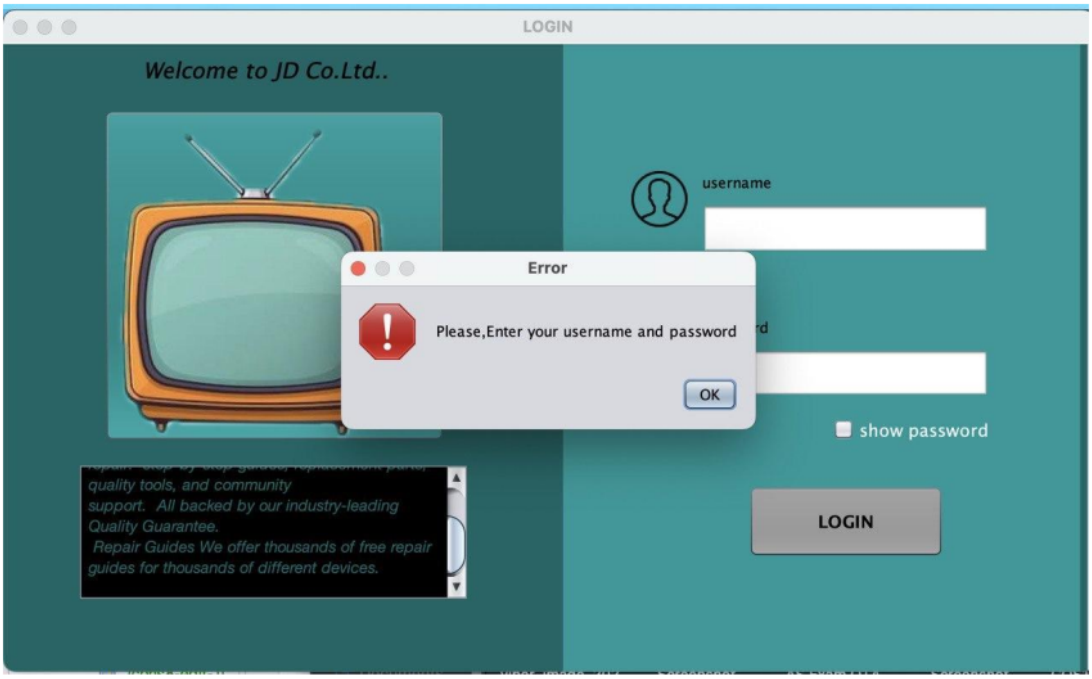
The initial program form ensures secure user interaction. Before taking any action, users must log in with their credentials. Incorrect inputs prompt an appropriate program response.



If the user enters an incorrect username and password, the login attempt will be unsuccessful.



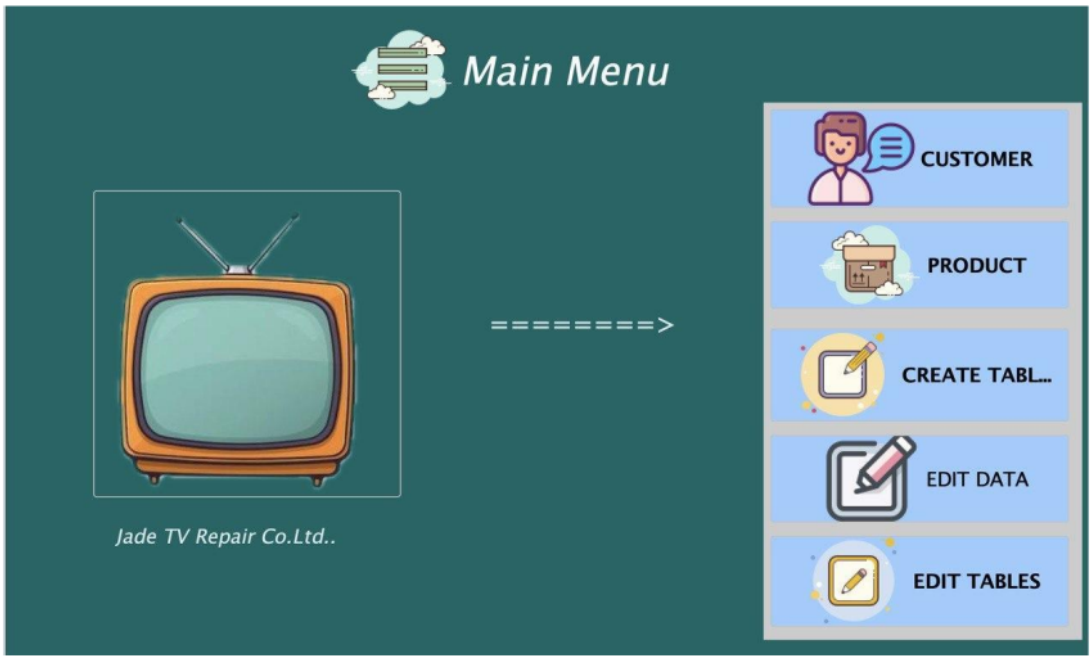
If the user provides an incorrect username or password, the program will display an error message.



- This represents the initial version of my program. When a user intends to perform an action, logging in is a prerequisite. In such instances, the user needs to input an accurate username and password. If the input is invalid, the program will display the following message.

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String s1 = txtId.getText();
    String s2 = txtPassword.getText();
    Connection con = null;
    ResultSet rs = null;
    Statement stmt = null;
    String url = "jdbc:mysql://localhost:3306/RepairDB";
    String driver = "com.mysql.cj.jdbc.Driver";
    String username = "root";
    String password = "root";
    try
    {
        con = DriverManager.getConnection(url,username,password);
        stmt = con.createStatement();
        rs = stmt.executeQuery("select * from login where username = '"+txtId.getText()+"'and password = '"+txtPassword.getText()+"'");
        int count = 0;
        while(rs.next())
        {
            count++;
        }
        if(s1.equals("")&&s2.equals(""))
        {
            JOptionPane.showMessageDialog(this,"Enter your ID and password","Error",JOptionPane.ERROR_MESSAGE);
        }
        else if (count == 1)
        {
            JOptionPane.showMessageDialog(this, "Congratulations!Successful Login");
            MainMenu MM = new MainMenu();
            MM.setVisible(true);
            this.setVisible(false);
        }
        else
        {
            txtId.setText("");
            txtPassword.setText("");
```

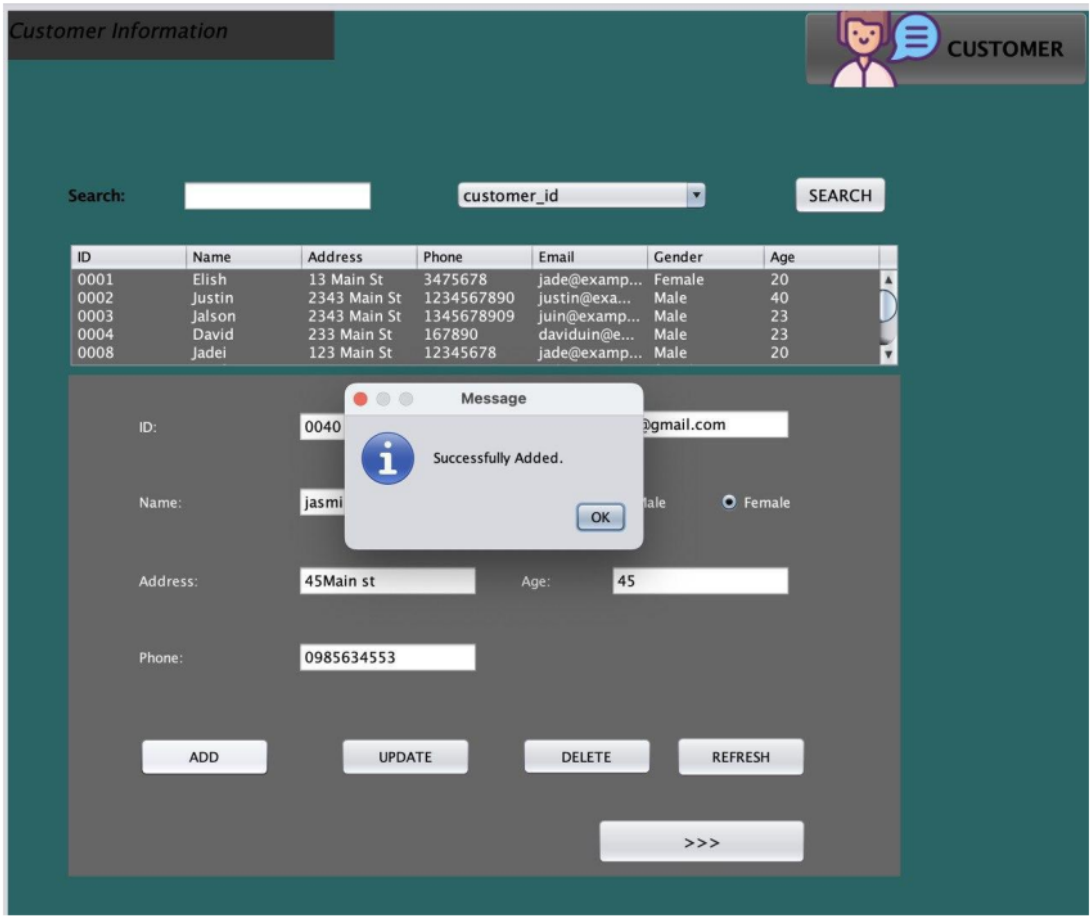
MAIN MENU FORM



In the Main Menu Form, each button can be clicked by the user to navigate to specific forms. For instance, clicking the 'customer' button will display the customer form, enabling the ability to examine and modify customer table data. In a similar vein, clicking the "product" button brings up the product form, where the user can check and edit data.

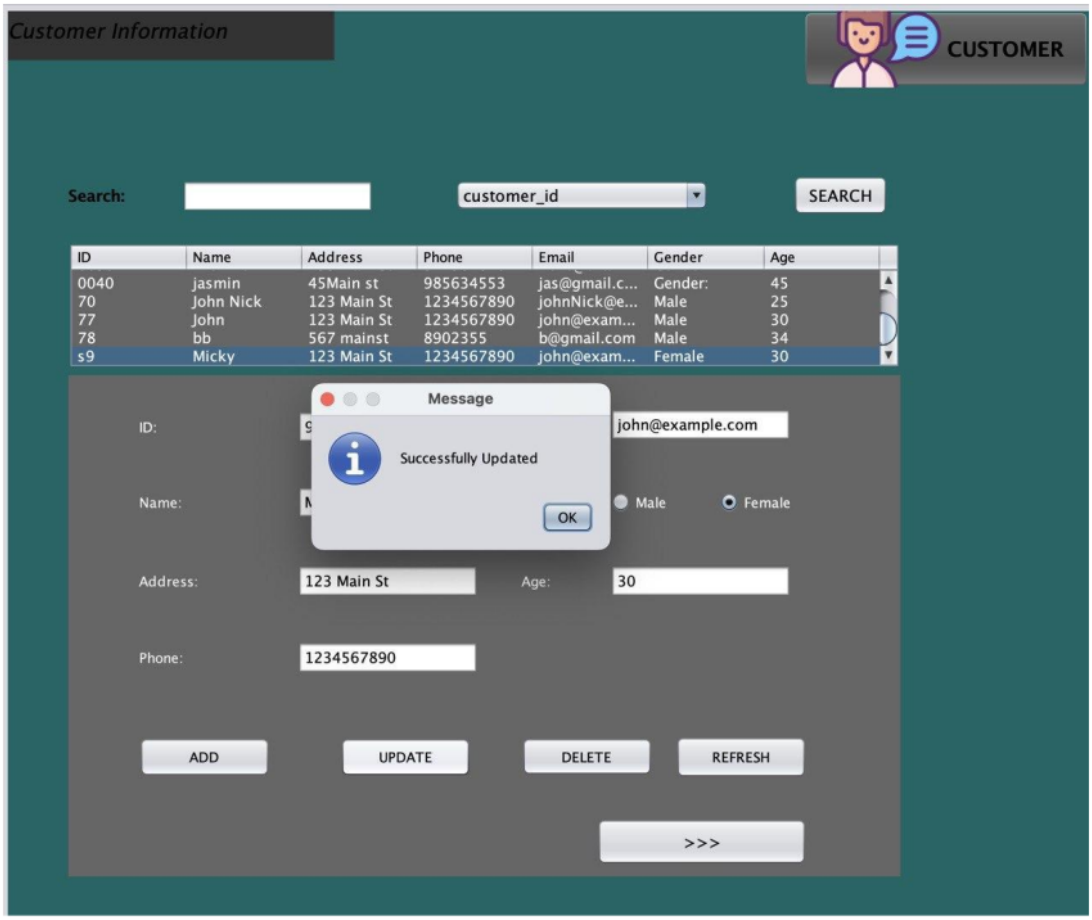
Click 'create tables' to open the table definition form and start creating tables. Use the 'edit data' button to access the Data Manipulation form in order to edit data. Click 'edit tables' to add, update, or remove columns from the tables in order to change the table definitions.

CLIENT INFORMATION



Selecting the 'customer' button in the main menu will display the customer form. To add a new entry to the database, input data into the text fields.

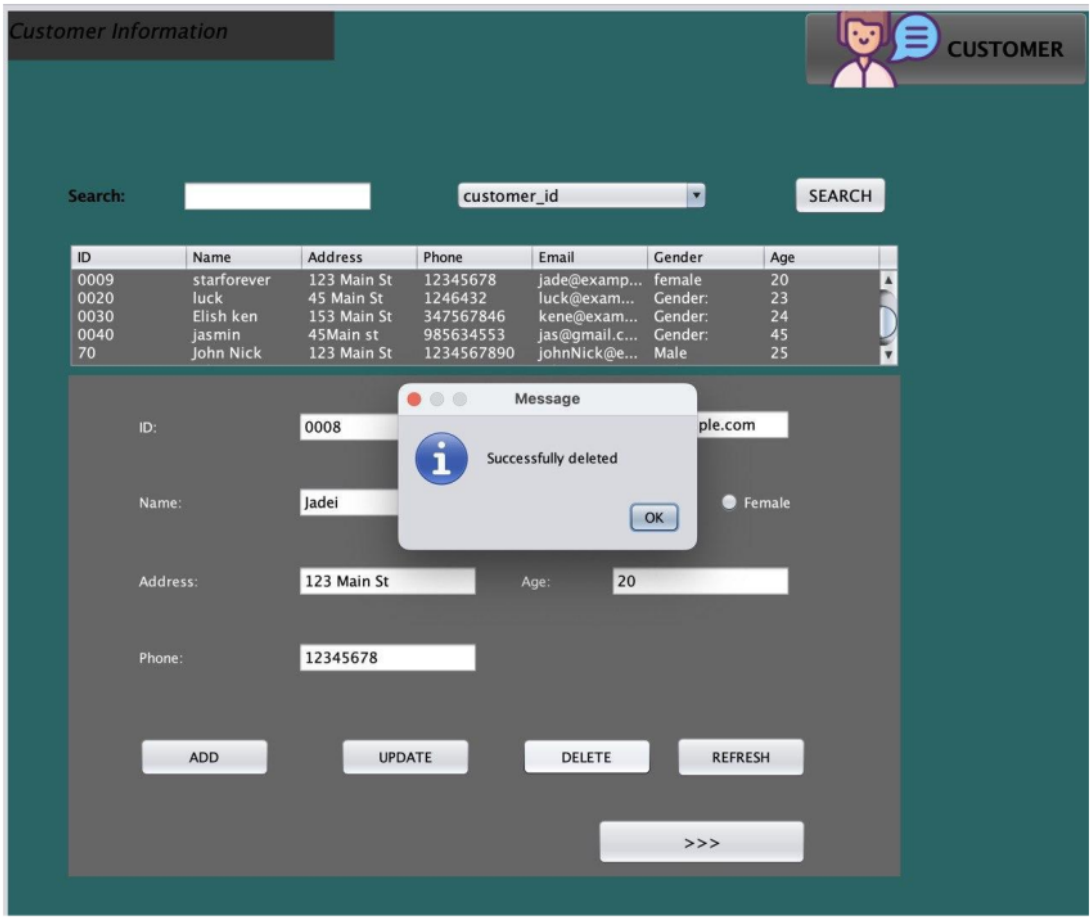
```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String gender = "";
    if(rdb1.isSelected())
    {
        gender = "Male";
    }
    else
    {
        gender = "Female";
    }
    ResultSet rs = null;
    Statement stmt = null;
    Connection con = null;
    String url = "jdbc:mysql://localhost:3306/RepairDB";
    String driver = "com.mysql.cj.jdbc.Driver";
    String username = "root";
    String password = "root";

    try
    {
        con = DriverManager.getConnection(url,username,password);
        stmt = con.createStatement();
        String query = "insert into customer values ('"+txtId.getText()+"', '"+txtName.getText()+"', '"+txtAddress.getText()+"','"+txtPhone.
        stmt.execute(query);
        JOptionPane.showMessageDialog(this,"Successfully Added.");
        Display();
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null,"Need to select row");
    }
```



Users can edit existing entries by modifying data directly in the text fields.

.

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String gender = "";
    if(rdb1.isSelected())
    {
        gender = "Male";
    }
    else
    {
        gender = "Female";
    }
    ResultSet rs = null;
    Connection con = null;
    Statement stmt = null;
    DefaultTableModel m = new DefaultTableModel();
    m = (DefaultTableModel)T1.getModel();
    int index = T1.getSelectedRow();
    if(index >= 0)
    {
        String url = "jdbc:mysql://localhost:3306/RepairDB";
        String driver = "com.mysql.cj.jdbc.Driver";
        String username = "root";
        String password = "root";
        try{
            con = DriverManager.getConnection(url,username,password);
            stmt = con.createStatement();
            String sql = "update customer set customer_id = '"+txtId.getText()+"',customer_name = '"+txtName.getText()+"',customer_address
            stmt.executeUpdate(sql);
            JOptionPane.showMessageDialog(this, "Successfully Updated");
            Display();
        }catch(Exception e){
            JOptionPane.showMessageDialog(null, e);
        }
    }
    else {
        JOptionPane.showMessageDialog(null,"Need to select row");
```



Users can remove rows by selecting them and using the delete button.

```
    private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        ResultSet rs = null;
        Connection con = null;
        Statement stmt = null;
        DefaultTableModel m = new DefaultTableModel();
        m = (DefaultTableModel)T1.getModel();
        int index = T1.getSelectedRow();
        if (index>=0)
        {
            m.removeRow(index);
            String url = "jdbc:mysql://localhost:3306/RepairDB";
            String driver = "com.mysql.cj.jdbc.Driver";
            String username = "root";
            String password = "root";
            try
            {
               con = DriverManager.getConnection(url,username,password);
               stmt = con.createStatement();
               String query = "delete from customer where customer_id = '"+txtId.getText()+"'";
               stmt.executeUpdate(query);
               JOptionPane.showMessageDialog(this,"Successfully deleted");
               Display();
               txtId.setText("");
               txtName.setText("");
               txtAddress.setText("");
               txtPhone.setText("");
               txtEmail.setText("");
               String gender = "";
               if (gender.equals("Male"))
               {
               rdb1.setSelected(true);
               rdb2.setSelected(false);
               }
               else if (gender.equals("Female"))
```

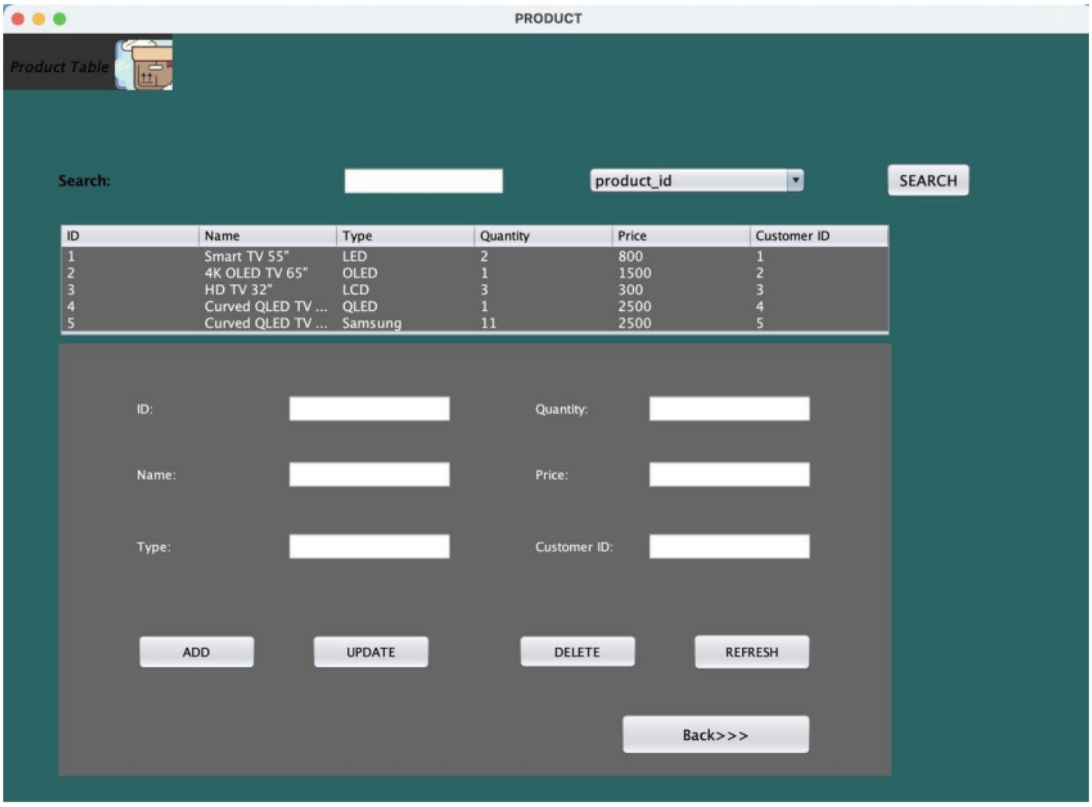
```
               Display();
               txtId.setText("");
               txtName.setText("");
               txtAddress.setText("");
               txtPhone.setText("");
               txtEmail.setText("");
               String gender = "";
               if (gender.equals("Male"))
               {
               rdb1.setSelected(true);
               rdb2.setSelected(false);
               }
               else if (gender.equals("Female"))
               {
               rdb1.setSelected(false);
               rdb2.setSelected(true);
               }
               else
               {
               rdb1.setSelected(false);
               rdb2.setSelected(false);
               }
                txtAge.setText("");
            }catch(Exception e)
            {
                JOptionPane.showMessageDialog(null,e);
            }
        }
        else
        {
            JOptionPane.showMessageDialog(null,"Need to select row");
        }
    }
```

Users have the ability to search for customer details by selecting the category and entering the corresponding information into the designated text field.

```
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ResultSet rs = null;
    Connection con = null;
    Statement stmt = null;
    String url = "jdbc:mysql://localhost:3306/RepairDB";
    String driver = "com.mysql.cj.jdbc.Driver";
    String username = "root";
    String password = "root";
    String search = txtSearch.getText();
    String column = cboSearch.getSelectedItem().toString();
    String sql = "select * from customer where "+column+ " Like '%"+search+"%'";
    try {
        con = DriverManager.getConnection(url,username,password);
        stmt=con.createStatement();
        rs = stmt.executeQuery(sql);
            if(rs!=null){
                T1.setModel(DbUtils.resultSetToTableModel(rs));
            }else{
                JOptionPane.showMessageDialog(this, "No data found");
            }
    }
    catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }
}
```

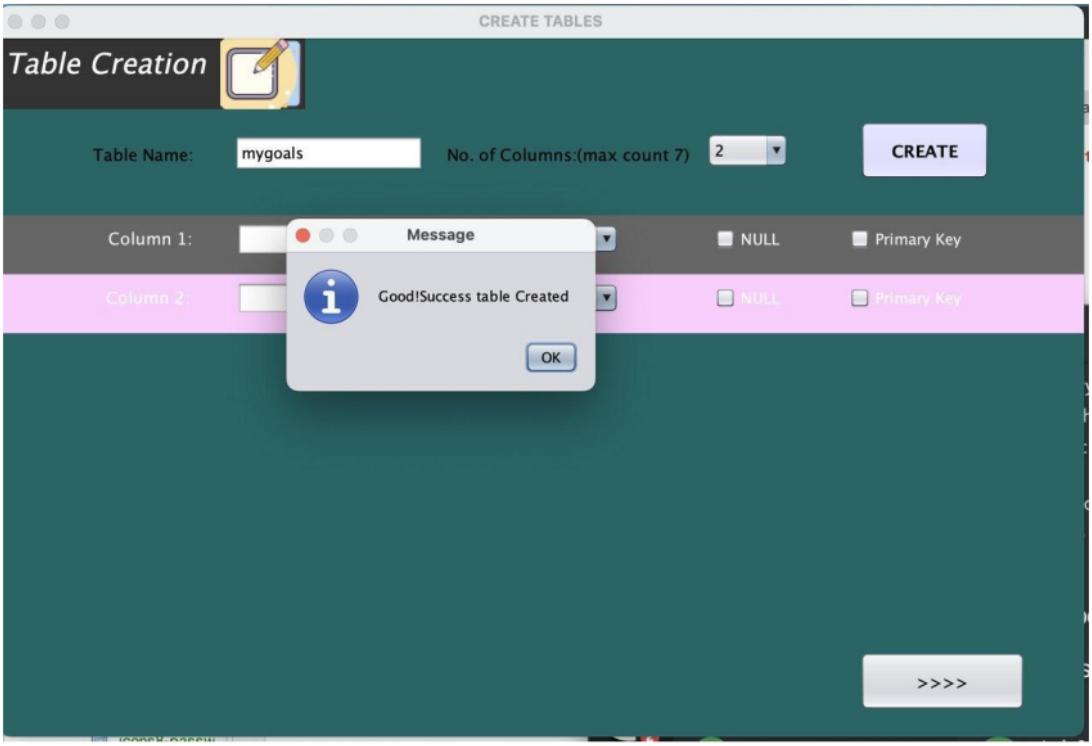
```
private void btnRefreshActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtId.setText("");
    txtName.setText("");
    txtAddress.setText("");
    txtPhone.setText("");
    txtEmail.setText("");
    String gender = "";
    if (gender.equals("Male"))
    {
    rdb1.setSelected(true);
    rdb2.setSelected(false);
    }
    else if (gender.equals("Female"))
    {
    rdb1.setSelected(false);
    rdb2.setSelected(true);
    }
    else
    {
    rdb1.setSelected(false);
    rdb2.setSelected(false);
    }
    txtAge.setText("");
    txtSearch.setText("");
    cboSearch.setSelectedIndex(0);
    Display();

}
```

PRODUCT FORM



The user can handle details about products in a manner similar to how you handle client information. In order to do this, you must add new data, update current entries, delete rows, refresh text fields, and search for specific items in the combo box based on their types.

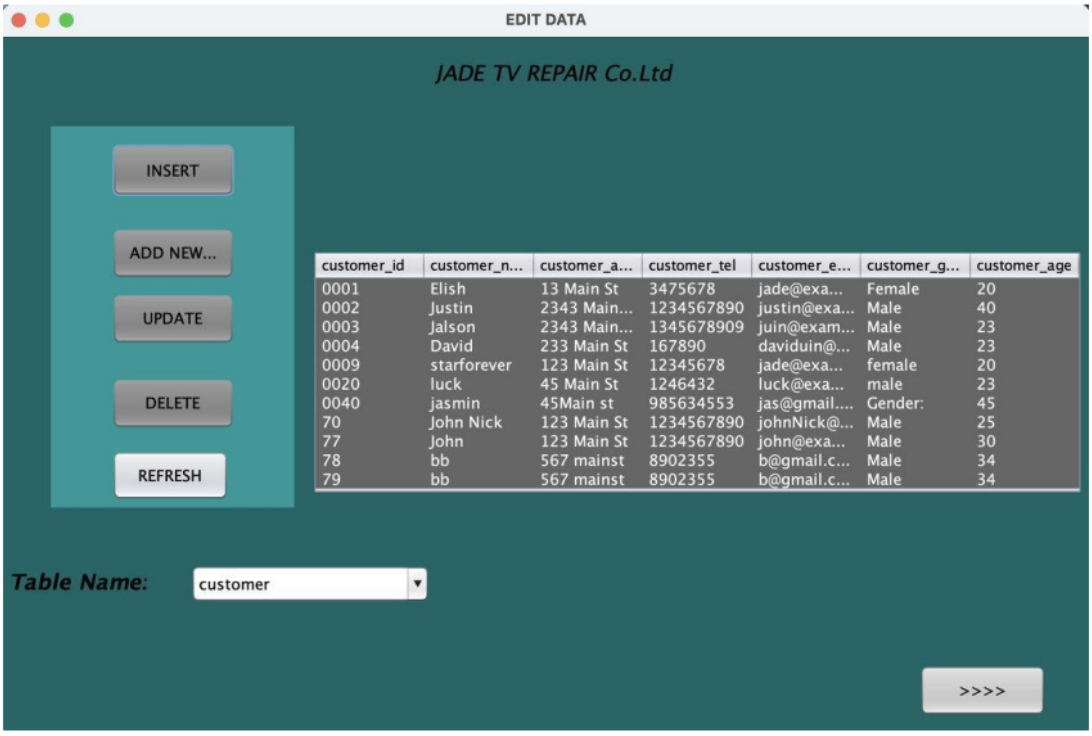
CREATE TABLES FORM



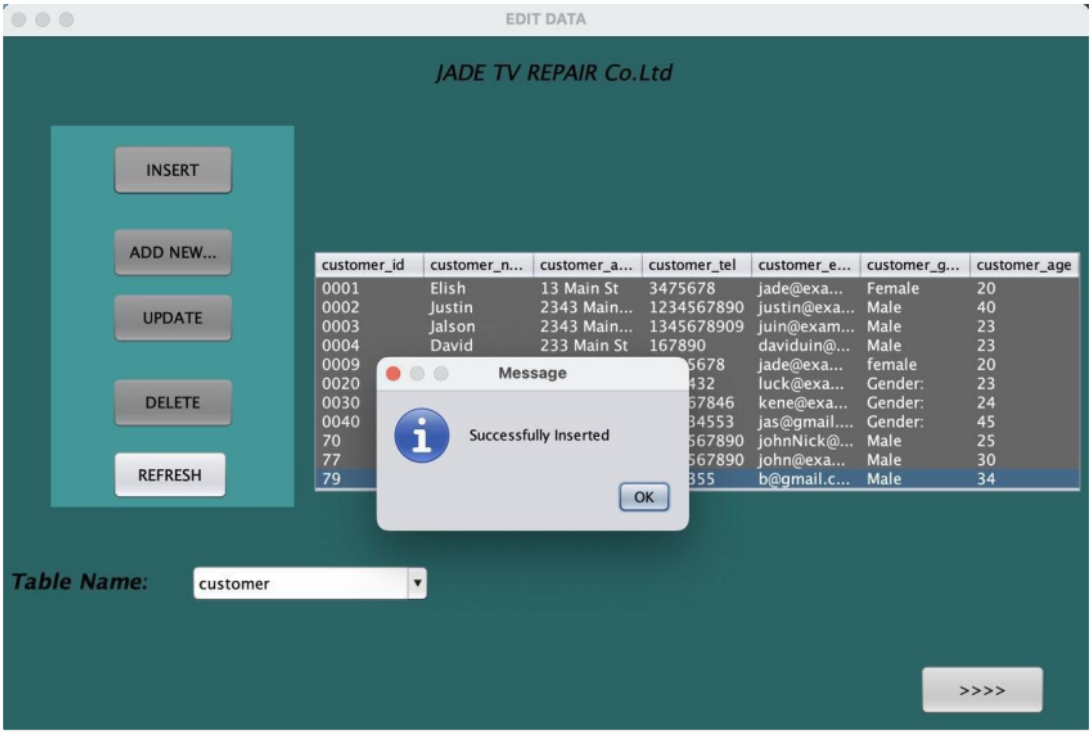
Clicking the 'create' button will add the newly defined table to the database.

```
    private void btnCreateActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (cboCol.getSelectedItem().equals("1"))
        {
        String [] allData = OneData();
        String tName = txtTable.getText();
        oneRow();
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB", "root","root");
            String sql = "create table RepairDB."+tName+"("+" "+allData[0]+" "+allData[1]+" "+allData[3]+""
                    +")";
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.execute();
            JOptionPane.showMessageDialog(this,"Good!Success table Created");
        }catch(Exception e)
        {
            JOptionPane.showMessageDialog(null,e);
        }
        }

        if (cboCol.getSelectedItem().equals("2"))
        {
        String [] allData = TwoData();
        String tName = txtTable.getText();
        oneRow();
        twoRow();
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB", "root","root");
            String sql = "create table RepairDB."+tName+"("+" "+allData[0]+" "+allData[1]+" "+allData[3]+","
                    + " "+" "+allData[4]+" "+allData[5]+" "+allData[6]+" "+allData[7]+""
                    +")";
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.execute();
            JOptionPane.showMessageDialog(this,"Good!Success table Created");
        }catch(Exception e)
```

DATA EDITION FORM



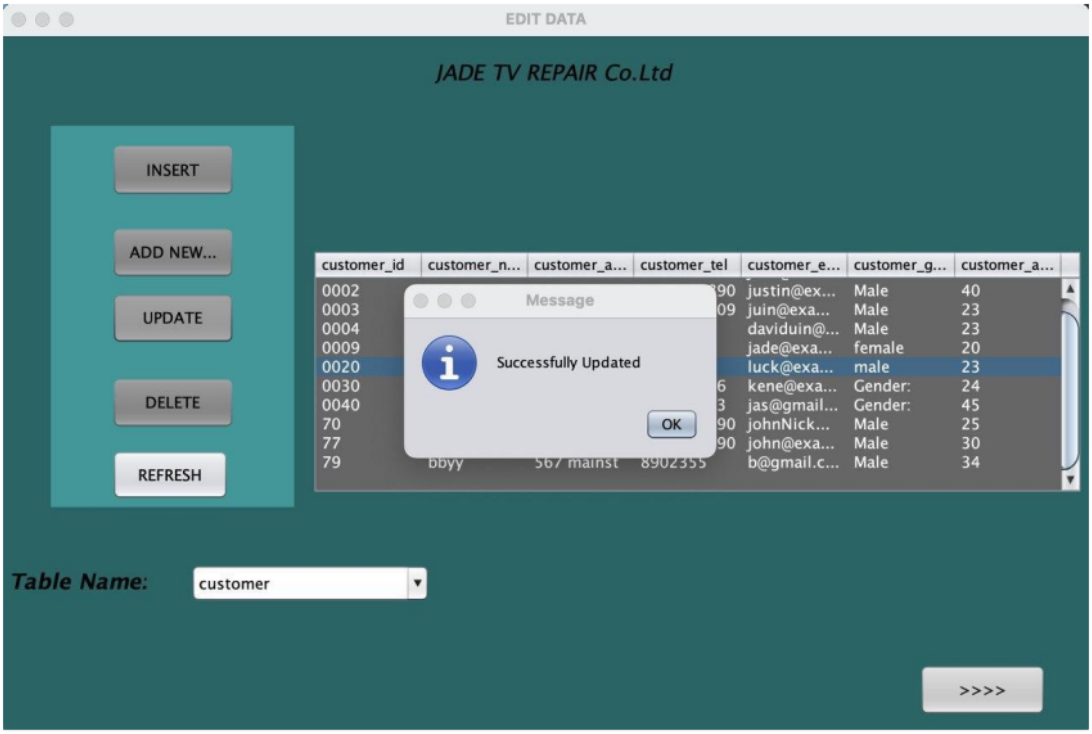
All database tables—aside from the "login" table—will be visible during Form Load in order to shield data from unauthorized alteration by database-accessible users.



To input new data into the table, users start by clicking the 'add new row' button, which generates a new row for data entry. After inputting the data, users can confirm by clicking the 'insert' button, and a message confirming successful insertion will be displayed.

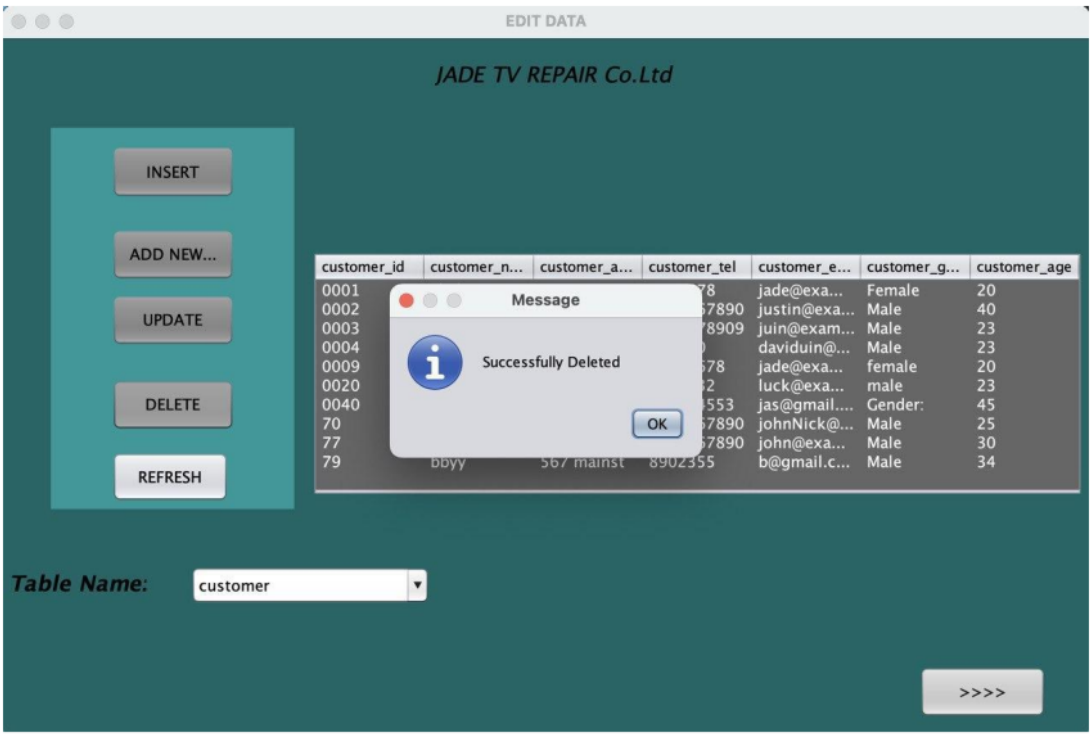
```
    private void btnInsertActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        DefaultTableModel m = (DefaultTableModel) T1.getModel();

        int index = T1.getSelectedRow();
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB", "root", "root");

            String tableName = cboTable.getSelectedItem().toString();
            StringBuilder sql = new StringBuilder("INSERT INTO RepairDB.");
            sql.append(tableName);
            sql.append(" VALUES (");

            for (int i = 0; i < m.getColumnCount(); i++)
            {
                if (i > 0)
                {
                    sql.append(",");
                }
                sql.append("?");
            }
            sql.append(")");

            PreparedStatement pstmt = con.prepareStatement(sql.toString());

            for (int i = 0; i < m.getColumnCount(); i++) {
                Object value = m.getValueAt(index, i);
                if (value == null)
                {
                    pstmt.setNull(i + 1, java.sql.Types.INTEGER);
                }
                else
                {
```

```
            String tableName = cboTable.getSelectedItem().toString();
            StringBuilder sql = new StringBuilder("INSERT INTO RepairDB.");
            sql.append(tableName);
            sql.append(" VALUES (");

            for (int i = 0; i < m.getColumnCount(); i++)
            {
                if (i > 0)
                {
                    sql.append(",");
                }
                sql.append("?");
            }
            sql.append(")");

            PreparedStatement pstmt = con.prepareStatement(sql.toString());

            for (int i = 0; i < m.getColumnCount(); i++) {
                Object value = m.getValueAt(index, i);
                if (value == null)
                {
                    pstmt.setNull(i + 1, java.sql.Types.INTEGER);
                }
                else
                {
                    pstmt.setString(i + 1, value.toString());
                }
            }
            pstmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "Successfully Inserted");
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "An error occurred while executing the SQL statement: " + e.getMessage(), "Error", JOptionPane.
        }
    }
```



Users can update by editing existing rows and entering new data. Upon completion, a message confirming the successful update will be displayed in the database.

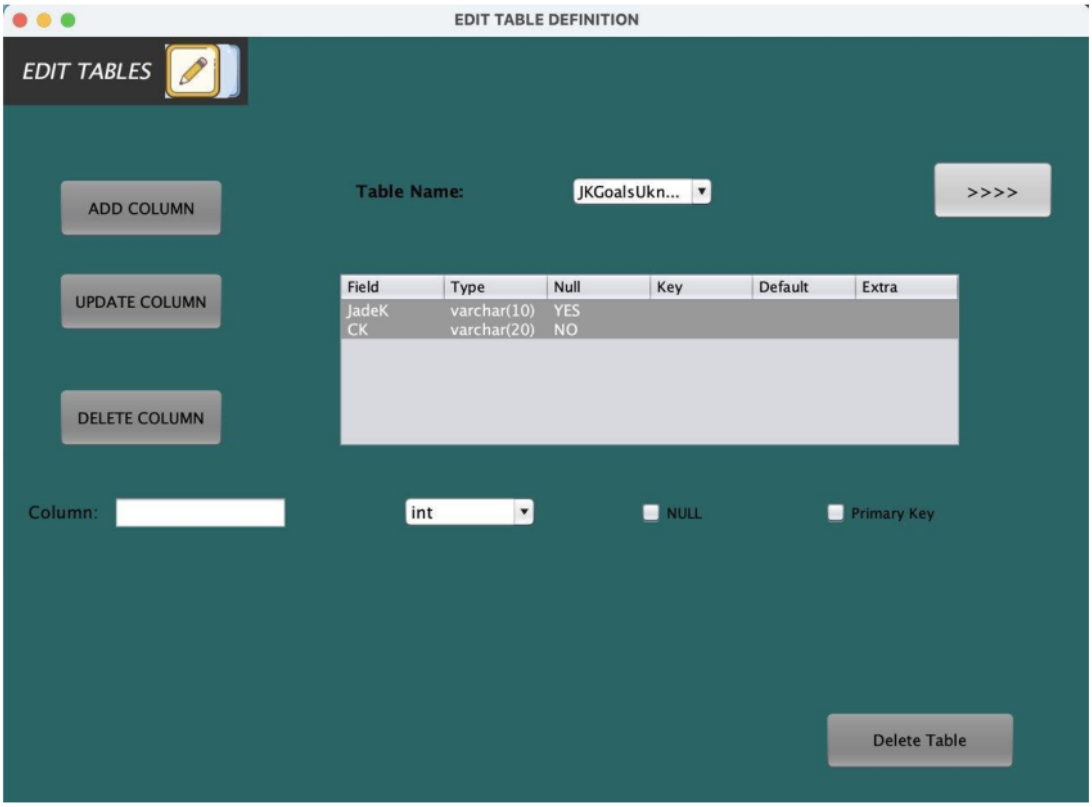
```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String tableName = cboTable.getSelectedItem().toString();

    String sql = "UPDATE " + tableName + " SET ";
    DefaultTableModel model = (DefaultTableModel) T1.getModel();
    int selectedRow = T1.getSelectedRow();

    for (int i = 0; i < model.getColumnCount(); i++)
    {
        String columnName = model.getColumnName(i);
        sql += columnName + " = ?";
        if (i < model.getColumnCount() - 1)
        {
            sql += ", ";
        }
    }
    sql += " WHERE " + model.getColumnName(0) + " = ?";

    try {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB", "root", "root");

        PreparedStatement pstmt = con.prepareStatement(sql);

        for (int i = 0; i < model.getColumnCount(); i++) {
            Object value = model.getValueAt(selectedRow, i);
            pstmt.setObject(i + 1, value);
        }
        Object primaryKeyValue = model.getValueAt(selectedRow, 0);
        pstmt.setObject(model.getColumnCount() + 1, primaryKeyValue);
        pstmt.executeUpdate();

        JOptionPane.showMessageDialog(this, "Successfully Updated");
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "An error occurred while executing the SQL statement: " + e.getMessage(),
```



Users can delete rows from tables by selecting the row and clicking the 'delete' button. After deletion, a message confirming 'Deleted Successfully' will be displayed.

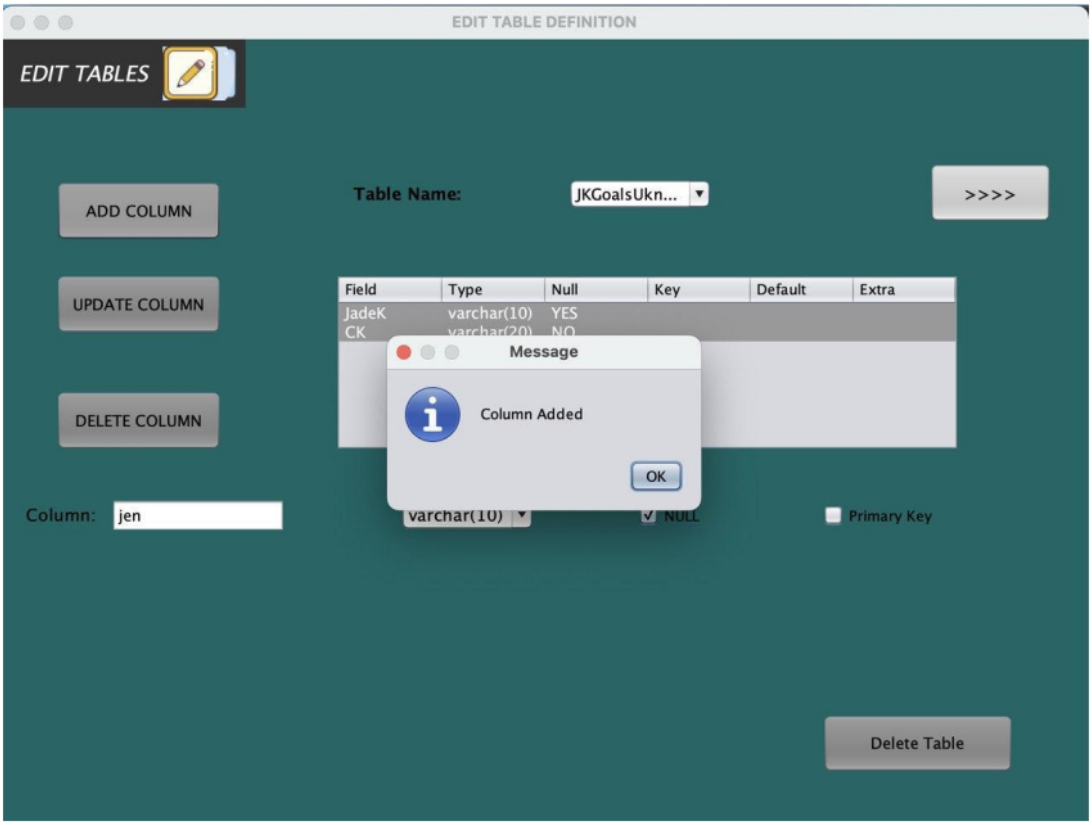
```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String tableName = cboTable.getSelectedItem().toString();
    DefaultTableModel model = (DefaultTableModel) T1.getModel();
    int selectedRow = T1.getSelectedRow();

    if (selectedRow != -1) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB", "root",
                "root");

            // Construct the SQL delete statement
            String sql = "DELETE FROM " + tableName + " WHERE " + model.getColumnName(0) + " = ?";

            PreparedStatement pstmt = con.prepareStatement(sql);

            // Set the value for the primary key column in the WHERE clause
            Object primaryKeyValue = model.getValueAt(selectedRow, 0);
            pstmt.setObject(1, primaryKeyValue);

            int rowsAffected = pstmt.executeUpdate();
            if (rowsAffected > 0) {
                model.removeRow(selectedRow);
                JOptionPane.showMessageDialog(this, "Successfully Deleted");
            } else {
                JOptionPane.showMessageDialog(this, "Failed to delete the row.", "Error",
                    JOptionPane.ERROR_MESSAGE);
            }
        } catch (Exception e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(null, "An error occurred: " + e.getMessage(), "Error",
                JOptionPane.ERROR_MESSAGE);
        }
    } else {
        JOptionPane.showMessageDialog(this, "No row selected.", "Error", JOptionPane.ERROR_MESSAGE);
    }
```

EDIT TABLE DEFINITION FORM



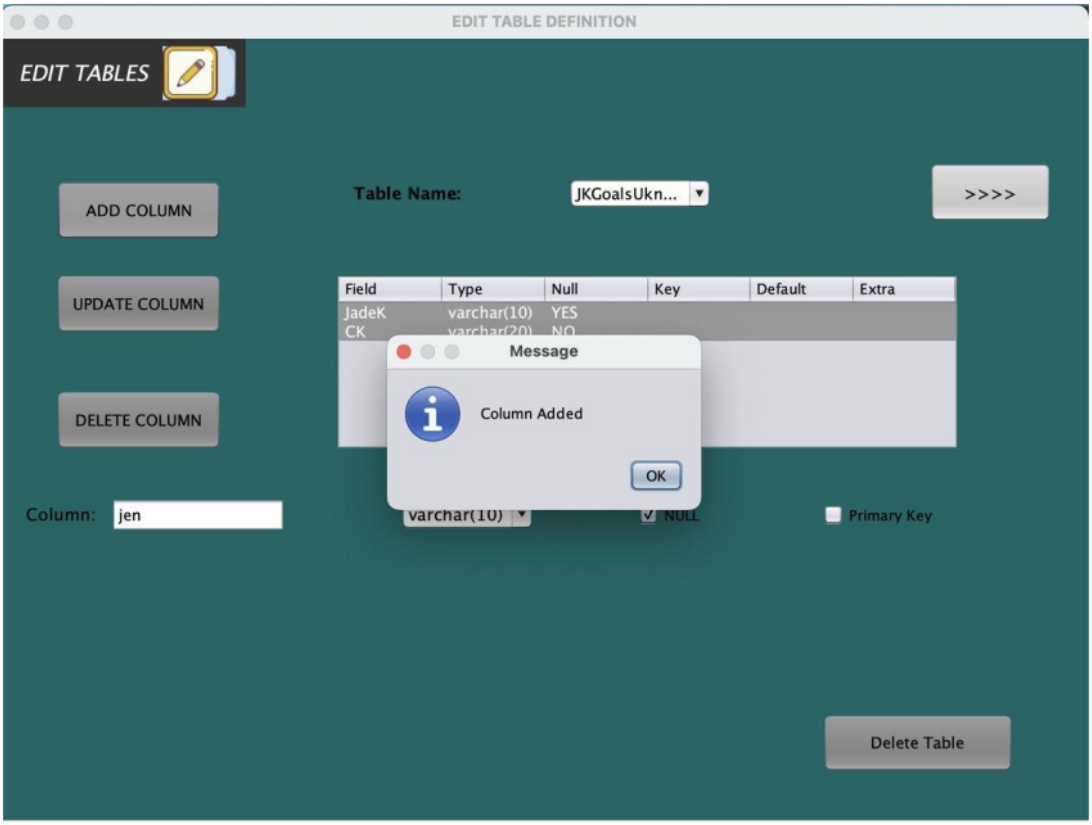
The customer and product tables will not be displayed when the form loads; instead, tables from the database will be displayed.

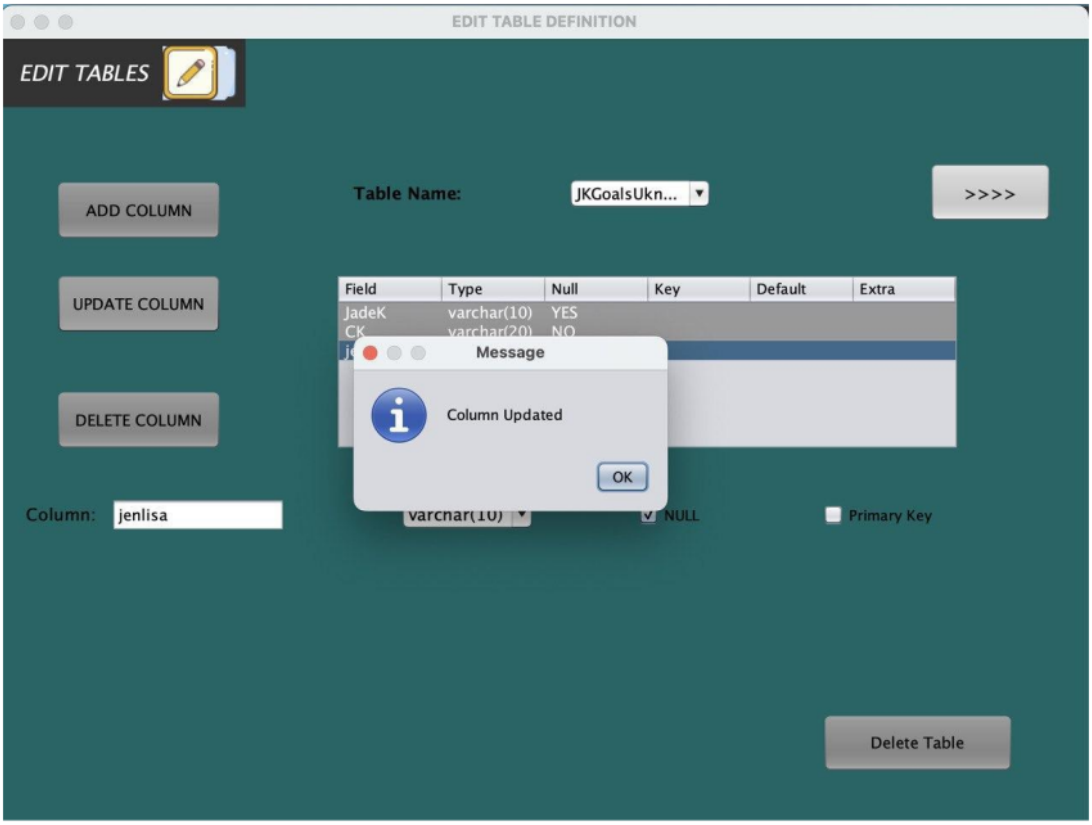
```
public DataDef()throws ClassNotFoundException {
    initComponents();
    ResultSet rs = null;
    try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB","root","root");
        DatabaseMetaData dbmd;

        dbmd=con.getMetaData();

        String[] type = {"TABLE"};
        rs = dbmd.getTables("RepairDB",null,"%",type);
        while(rs.next())
        {
            String tableName=rs.getString("TABLE_NAME");
            if(!tableName.equals("customer") && !tableName.equals("tv.project")){
            cboTable.addItem(tableName);

        }
        }

        Display();
    }catch(Exception ex)
    {
        Logger.getLogger(DataDef.class.getName()).log(Level.SEVERE,null,ex);
    }

}
```



The client has the option to introduce a new column to the table by providing the necessary information and selecting its data type. Upon successful addition, a message indicating it will be displayed.

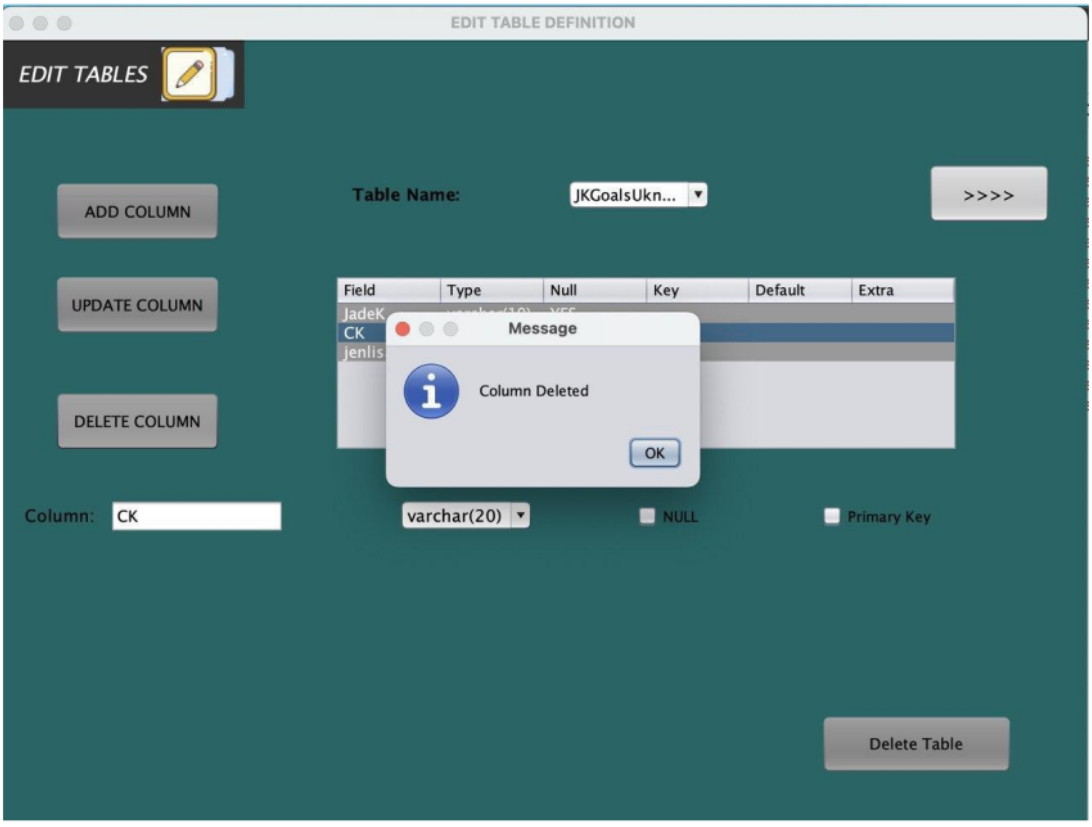
```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String Name = txtCol.getText();
    String type = cboCol.getSelectedItem().toString();
    String st;
    String pKey;
    if(chkNull.isSelected()==true)
    {
        st = "NULL";
    }
    else
    {
        st = "NOT NULL";
    }
    if (chkPk.isSelected()==true)
    {
        pKey = "Primary Key";
    }
    else
    {
        pKey = "";
    }
    try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB","root","root");
        String sql = "alter table RepairDB. "+cboTable.getSelectedItem()+" add "+Name+ " "+type+" "+st+" "+pKey+" ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(this,"Column Added");
        Display();
        Refresh();
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
```


EDIT TABLE DEFINITION
EDIT TABLES
ADD COLUMN
UPDATE COLUMN
DELETE COLUMN
Table Name:
JKGoalsUkn...
>>>>
Field
Type
Null
Key
Default
Extra
JadeK
varchar(10)
YES
CK
Message
Column Added
OK
Column:
jen
varchar(10)
NULL
Primary Key
Delete Table

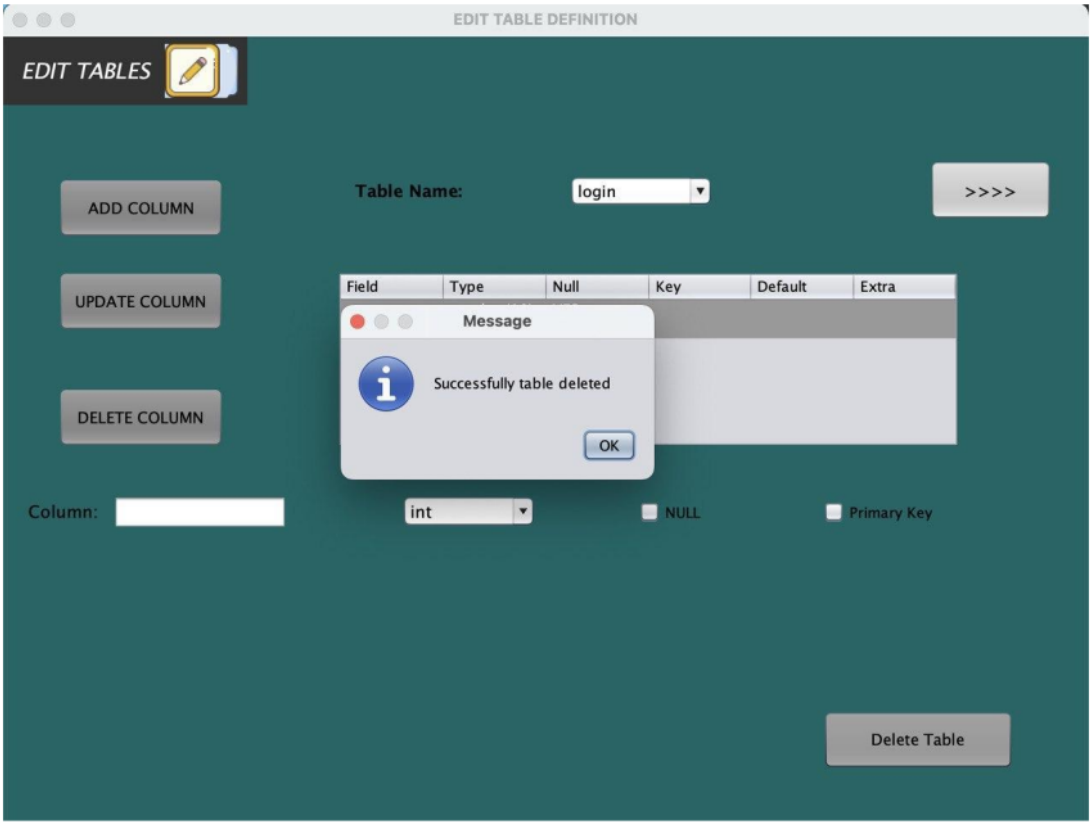


Selecting a row populates data in the corresponding controls. The user can input or select new data names and types before updating the row.

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DefaultTableModel m = new DefaultTableModel();
    m = (DefaultTableModel)T1.getModel();
    int index = T1.getSelectedRow();
    String oldName = m.getValueAt(index, 0).toString();
    String newName = txtCol.getText();
    String type = cboCol.getSelectedItem().toString();
    String st;
    String pKey;
    if(chkNull.isSelected()==true)
    {
        st = "NULL";
    }
    else
    {
        st = "NOT NULL";
    }
    if (chkPk.isSelected()==true)
    {
        pKey = "PRI";
    }
    else
    {
        pKey = "";
    }
     try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB","root","root");
        String sql = "alter table RepairDB. "+cboTable.getSelectedItem()+" change column "+oldName+" "+newName+" "+type+" "+st+" "+pKey+" ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(this,"Column Updated");
        Display();
        Refresh();
```



```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String cName = txtCol.getText();
    try
    {
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB","root","root");
        String sql = "alter table RepairDB. "+cboTable.getSelectedItem()+" drop column "+cName+" ";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.execute();
        JOptionPane.showMessageDialog(this,"Column Deleted");
        Display();
        Refresh();
    }catch(Exception e)
    {
        JOptionPane.showMessageDialog(null, e);
    }
}
```



```
    private void btnDropActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int i = cboTable.getSelectedIndex();
        //cboTable.removeItem(i);
        try
        {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/RepairDB","root","root");
            String sql = "drop table RepairDB."+ cboTable.getSelectedItem()+" ";
            cboTable.removeItem(cboTable.getSelectedItem());
            PreparedStatement pstmt = con.prepareStatement(sql);
            pstmt.execute();
            JOptionPane.showMessageDialog(this,"Successfully table deleted");
            Display();
            Refresh();
        }catch(Exception e)
        {
            JOptionPane.showMessageDialog(null, e);
        }
    }
```

Cos-106byjade

ORIGINALITY REPORT

2%	0%	0%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to TMC Education Group Student Paper	2%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		