MINI PROJECT

Source Code:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

import java.math.BigDecimal;
import java.sql.*;
import java.time.LocalDate;

import User.*;

public class HelloJavaFX extends Application {
    Connection con;
    user u;
    address ad;
    education ed;
    employmentInformation emp;
    emergencyContact ec;
    int id;
    String name;

    @Override
    public void start(Stage primaryStage) {
        initializeDatabaseConnection();

        TextField nameField = new TextField();
        nameField.setPromptText("Enter your name");

        Button displayButton = new Button("Display User Info");
        Button createButton = new Button("Create a New User");

        VBox root = new VBox(10, nameField, displayButton, createButton);
        Scene scene = new Scene(root, 400, 300);

        displayButton.setOnAction(event -> displayUserInfo(nameField, primaryStage));
        createButton.setOnAction(event -> createNewUser(primaryStage));

        primaryStage.setTitle("User Information System");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
```

```java
private void displayUserInfo(TextField nameField, Stage primaryStage) {
    name = nameField.getText().trim();
    name = nameField.getText().trim();
    if (name.isEmpty()) {
        showAlert("Invalid Input", "Please enter a valid name.");
        return;
    }

    String userInfo = u.showUser(name);
    if (userInfo == null || userInfo.contains("No user found")) {
        showAlert("Not Found", "No user found with the name " + name);
        return;
    }

    id = Integer.parseInt(userInfo.split(" ")[0]);
    TextArea resultArea = new TextArea();
    resultArea.setEditable(false);
    resultArea.setWrapText(true);

    resultArea.setText("User Info:\n" + userInfo + "\n\n");
    resultArea.appendText("Address Info:\n" + ad.showAddress(id) + "\n\n");
    resultArea.appendText("Education Info:\n" + ed.showEducation(id) + "\n\n");
    resultArea.appendText("Employment Info:\n" + emp.showEmploymentInfo(id) + "\n\n");
    resultArea.appendText("Emergency Contacts:\n" + ec.showEmergencyContacts(id) + "\n");

    Button updateButton = new Button("Update Information");
    updateButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

    Button deleteButton = new Button("Delete User");
    deleteButton.setOnAction(e -> {
        deleteUser(id);
        showAlert("Success", "User deleted successfully.");
        showMainMenu(primaryStage);
    });

    Button backButton = new Button("Back");
    backButton.setOnAction(e -> showMainMenu(primaryStage));

    VBox outputLayout = new VBox(10, resultArea, updateButton, deleteButton, backButton);
    primaryStage.setScene(new Scene(outputLayout, 400, 400));
}

private void createNewUser(Stage primaryStage) {
```

```
TextField firstNameField = new TextField();
firstNameField.setPromptText("Enter First Name");

TextField lastNameField = new TextField();
lastNameField.setPromptText("Enter Last Name");

TextField emailField = new TextField();
emailField.setPromptText("Enter Email");

TextField phoneField = new TextField();
phoneField.setPromptText("Enter Phone Number");

TextField addressField = new TextField();
addressField.setPromptText("Enter Address");

TextField streetField = new TextField();
streetField.setPromptText("Enter Street");

TextField cityField = new TextField();
cityField.setPromptText("Enter City");

TextField stateField = new TextField();
stateField.setPromptText("Enter State");

TextField postalCodeField = new TextField();
postalCodeField.setPromptText("Enter Postal Code");

TextField degreeField = new TextField();
degreeField.setPromptText("Enter Degree");

TextField institutionField = new TextField();
institutionField.setPromptText("Enter Institution Name");

TextField graduationYearField = new TextField();
graduationYearField.setPromptText("Enter Graduation Year");

TextField gradeField = new TextField();
gradeField.setPromptText("Enter Grade");


TextField jobTitleField = new TextField();
jobTitleField.setPromptText("Enter Job Title");

TextField companyField = new TextField();
```

```
companyField.setPromptText("Enter Company Name");

DatePicker startDatePicker = new DatePicker();
startDatePicker.setPromptText("Start Date");

DatePicker endDatePicker = new DatePicker();
endDatePicker.setPromptText("End Date");

TextField contactNameField = new TextField();
contactNameField.setPromptText("Enter Emergency Contact Name");

TextField contactPhoneField = new TextField();
contactPhoneField.setPromptText("Enter Emergency Contact Phone");

TextField contactEmailField = new TextField();
contactEmailField.setPromptText("Enter Emergency Contact Email");


Button saveButton = new Button("Save User");
saveButton.setOnAction(e -> {
    String firstName = firstNameField.getText().trim();
    String lastName = lastNameField.getText().trim();
    String email = emailField.getText().trim();
    String phone = phoneField.getText().trim();
    String address = addressField.getText().trim();

    String street = streetField.getText().trim();
    String city = cityField.getText().trim();
    String state = stateField.getText().trim();
    String postalCode = postalCodeField.getText().trim();

    String degree = degreeField.getText().trim();
    String institution = institutionField.getText().trim();
    String gradeStr = gradeField.getText().trim();
    int graduationYear = Integer.parseInt(graduationYearField.getText().trim());
    BigDecimal grade = new BigDecimal(gradeStr);

    String jobTitle = jobTitleField.getText().trim();
    String company = companyField.getText().trim();
    LocalDate startDate = startDatePicker.getValue();
    LocalDate endDate = endDatePicker.getValue();

    String contactName = contactNameField.getText().trim();
    String contactPhone = contactPhoneField.getText().trim();
```

```
            String contactEmail = contactEmailField.getText().trim();



            if (firstName.isEmpty() || lastName.isEmpty() || email.isEmpty() || phone.isEmpty() ||
address.isEmpty() ||
                street.isEmpty() || city.isEmpty() || state.isEmpty() || postalCode.isEmpty() ||
                jobTitle.isEmpty() || company.isEmpty() || startDate == null || endDate == null ||
                contactName.isEmpty() || contactPhone.isEmpty() || contactEmail.isEmpty()) {
                showAlert("Error", "All fields are required.");
            } else {

                 u.insertUser(firstName, lastName, email, phone, address);
                String userInfo = u.showUser(firstName);
                int userId = Integer.parseInt(userInfo.split(" ")[0]);
                ad.insertAddress(userId, street, city, state, postalCode);
                ed.insertEducation(userId, degree, institution, graduationYear, grade);
                emp.insertEmploymentInfo(userId, jobTitle, company, Date.valueOf(startDate),
Date.valueOf(endDate));
                ec.insertEmergencyContact(userId, contactName, "Emergency", contactPhone,
contactEmail);

                showAlert("Success", "User and related details created successfully.");
                showMainMenu(primaryStage);
            }
        });

        Button backButton = new Button("Back");
        backButton.setOnAction(e -> showMainMenu(primaryStage));

        VBox createUserLayout = new VBox(10, firstNameField, lastNameField, emailField,
phoneField, addressField,
                streetField, cityField, stateField, postalCodeField,  degreeField, institutionField,
graduationYearField, gradeField, jobTitleField, companyField,
                startDatePicker, endDatePicker, contactNameField, contactPhoneField,
contactEmailField, saveButton, backButton);

        ScrollPane scrollPane = new ScrollPane(createUserLayout);
        scrollPane.setFitToHeight(true);
        scrollPane.setFitToWidth(true);

        primaryStage.setScene(new Scene(scrollPane, 400, 500));
    }
```

```
private void showUpdateOptions(Stage primaryStage, int id) {
    Button updateUserButton = new Button("Update User Details");
    updateUserButton.setOnAction(e -> updateUser(primaryStage, id));

    Button updateAddressButton = new Button("Update Address");
    updateAddressButton.setOnAction(e -> updateAddress(primaryStage, id));

    Button updateEducationButton = new Button("Update Education");
    updateEducationButton.setOnAction(e -> updateEducation(primaryStage, id));

    Button updateEmploymentButton = new Button("Update Employment Info");
    updateEmploymentButton.setOnAction(e -> updateEmployment(primaryStage, id));

    Button updateEmergencyButton = new Button("Update Emergency Contact");
    updateEmergencyButton.setOnAction(e -> updateEmergencyContact(primaryStage, id));

    Button backButton = new Button("Back");
    backButton.setOnAction(e -> showMainMenu(primaryStage));

    VBox updateOptionsLayout = new VBox(10,
        updateUserButton,
        updateAddressButton,
        updateEducationButton,
        updateEmploymentButton,
        updateEmergencyButton,
        backButton
    );

    primaryStage.setScene(new Scene(updateOptionsLayout, 400, 300));
}
private void showMainMenu(Stage primaryStage) {
    TextField nameField = new TextField();
    nameField.setPromptText("Enter your name");

    Button displayButton = new Button("Display User Info");
    Button createButton = new Button("Create a New User");

    displayButton.setOnAction(event -> displayUserInfo(nameField, primaryStage));
    createButton.setOnAction(event -> createNewUser(primaryStage));

    VBox mainMenuLayout = new VBox(10, nameField, displayButton, createButton);
    primaryStage.setScene(new Scene(mainMenuLayout, 400, 300));
    }

    private void updateUser(Stage primaryStage, int id) {

        TextField firstNameField = new TextField();
        firstNameField.setPromptText("First Name");

        TextField lastNameField = new TextField();
        lastNameField.setPromptText("Last Name");

        TextField emailField = new TextField();
        emailField.setPromptText("Email");

        TextField phoneField = new TextField();
        phoneField.setPromptText("Phone Number");

        TextField addressField = new TextField();
        addressField.setPromptText("Address");

        Button saveButton = new Button("Save Changes");
        saveButton.setOnAction(e -> {
            String firstName = firstNameField.getText().trim();
            String lastName = lastNameField.getText().trim();
            String email = emailField.getText().trim();
            String phone = phoneField.getText().trim();
            String address = addressField.getText().trim();

            if (firstName.isEmpty() || lastName.isEmpty() || email.isEmpty() || phone.isEmpty() ||
address.isEmpty()) {
                showAlert("Error", "All fields are required.");
            } else {
                u.updateUser(id, firstName, lastName, email, phone, address);
                showAlert("Success", "User information updated successfully.");
                showMainMenu(primaryStage);
            }
        });

        Button backButton = new Button("Back");
        backButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

        VBox updateUserLayout = new VBox(10,
            new Label("Update User Information"),
            firstNameField,
            lastNameField,
                emailField,
                phoneField,
                addressField,
                saveButton,
                backButton
        );

        primaryStage.setScene(new Scene(updateUserLayout, 400, 400));
    }


    private void updateAddress(Stage primaryStage, int id) {
        TextField streetField = new TextField();
        streetField.setPromptText("Street");

        TextField cityField = new TextField();
        cityField.setPromptText("City");

        TextField stateField = new TextField();
        stateField.setPromptText("State");

        TextField postalCodeField = new TextField();
        postalCodeField.setPromptText("Postal Code");

        Button saveButton = new Button("Save Changes");
        saveButton.setOnAction(e -> {
            String street = streetField.getText().trim();
            String city = cityField.getText().trim();
            String state = stateField.getText().trim();
            String postalCode = postalCodeField.getText().trim();

            if (street.isEmpty() || city.isEmpty() || state.isEmpty() || postalCode.isEmpty()) {
                showAlert("Error", "All fields are required.");
            } else {
                ad.updateAddress(id, street, city, state, postalCode);
                showAlert("Success", "Address updated successfully.");
                showMainMenu(primaryStage);
            }
        });

        Button backButton = new Button("Back");
        backButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

        VBox updateAddressLayout = new VBox(10,
            new Label("Update Address"),
            streetField,
            cityField,
            stateField,
            postalCodeField,
            saveButton,
            backButton
        );

        primaryStage.setScene(new Scene(updateAddressLayout, 400, 400));
    }


    private void updateEducation(Stage primaryStage, int id) {
        TextField degreeField = new TextField();
        degreeField.setPromptText("Degree");

        TextField institutionField = new TextField();
        institutionField.setPromptText("Institution");

        TextField graduationYearField = new TextField();
        graduationYearField.setPromptText("Graduation Year");

        TextField gradeField = new TextField();
        gradeField.setPromptText("Grade");

        Button saveButton = new Button("Save Changes");
        saveButton.setOnAction(e -> {
            String degree = degreeField.getText().trim();
            String institution = institutionField.getText().trim();
            String graduationYear = graduationYearField.getText().trim();
            String grade = gradeField.getText().trim();

            if (degree.isEmpty() || institution.isEmpty() || graduationYear.isEmpty() ||
grade.isEmpty()) {
                showAlert("Error", "All fields are required.");
            } else {
                ed.updateEducation(id, degree, institution, Integer.parseInt(graduationYear), new
BigDecimal(grade));
                showAlert("Success", "Education information updated successfully.");
                showMainMenu(primaryStage);
            }
        });
        Button backButton = new Button("Back");
        backButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

        VBox updateEducationLayout = new VBox(10,
            new Label("Update Education"),
            degreeField,
            institutionField,
            graduationYearField,
            gradeField,
            saveButton,
            backButton
        );

        primaryStage.setScene(new Scene(updateEducationLayout, 400, 400));
    }


    private void updateEmployment(Stage primaryStage, int id) {
        TextField jobTitleField = new TextField();
        jobTitleField.setPromptText("Job Title");

        TextField companyNameField = new TextField();
        companyNameField.setPromptText("Company Name");

        TextField startDateField = new TextField();
        startDateField.setPromptText("Start Date (YYYY-MM-DD)");

        TextField endDateField = new TextField();
        endDateField.setPromptText("End Date");

        Button saveButton = new Button("Save Changes");
        saveButton.setOnAction(e -> {
            String jobTitle = jobTitleField.getText().trim();
            String companyName = companyNameField.getText().trim();
            String startDate = startDateField.getText().trim();
            String endDate = endDateField.getText().trim();

            if (jobTitle.isEmpty() || companyName.isEmpty() || startDate.isEmpty() ||
endDate.isEmpty()) {
                showAlert("Error", "All fields are required.");
            } else {
                emp.updateEmploymentInfo(id, jobTitle, companyName, Date.valueOf(startDate),
Date.valueOf(endDate));
                showAlert("Success", "Employment information updated successfully.");
                showMainMenu(primaryStage);
            }
        });

        Button backButton = new Button("Back");
        backButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

        VBox updateEmploymentLayout = new VBox(10,
            new Label("Update Employment Info"),
            jobTitleField,
            companyNameField,
            startDateField,
            endDateField,
            saveButton,
            backButton
        );

        primaryStage.setScene(new Scene(updateEmploymentLayout, 400, 400));
    }


    private void updateEmergencyContact(Stage primaryStage, int id) {
    TextField contactNameField = new TextField();
    contactNameField.setPromptText("Contact Name");

    TextField relationshipField = new TextField();
    relationshipField.setPromptText("Relationship");

    TextField contactPhoneField = new TextField();
    contactPhoneField.setPromptText("Contact Phone");

    TextField contactEmailField = new TextField();
    contactEmailField.setPromptText("Contact Email");

    Button saveButton = new Button("Save Changes");
    saveButton.setOnAction(e -> {
        String contactName = contactNameField.getText().trim();
        String relationship = relationshipField.getText().trim();
        String contactPhone = contactPhoneField.getText().trim();
        String contactEmail = contactEmailField.getText().trim();

        if (contactName.isEmpty() || relationship.isEmpty() || contactPhone.isEmpty() ||
contactEmail.isEmpty()) {
            showAlert("Error", "All fields are required.");
        } else {
            ec.updateEmergencyContact(id, contactName, relationship, contactPhone,
contactEmail);
            showAlert("Success", "Emergency contact updated successfully.");
            showMainMenu(primaryStage);
        }
    });

    Button backButton = new Button("Back");
    backButton.setOnAction(e -> showUpdateOptions(primaryStage, id));

    VBox updateEmergencyLayout = new VBox(10,
        new Label("Update Emergency Contact"),
        contactNameField,
        relationshipField,
        contactPhoneField,
        contactEmailField,
        saveButton,
        backButton
    );

    primaryStage.setScene(new Scene(updateEmergencyLayout, 400, 400));
}

    private void deleteUser(int id) {
        ad.deleteUser(id);
        ed.deleteUser(id);
        emp.deleteUser(id);
        ec.deleteUser(id);
        u.deleteUser(id);
    }

    private void initializeDatabaseConnection() {
        try {
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb", "root",
"Lingeswaran@21");
            u = new user(con);
            ad = new address(con);
            ed = new education(con);
            emp = new employmentInformation(con);
            ec = new emergencyContact(con);
        } catch (SQLException e) {
            showAlert("Database Error", "Unable to connect to the database.");
            e.printStackTrace();
        }
    }

    private void showAlert(String title, String message) {
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle(title);
        alert.setHeaderText(null);
        alert.setContentText(message);
        alert.showAndWait();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

Backend Code:
User pakage:

```
package User;
import java.sql.*;

public class user {
    private Connection con;
    public user (Connection con){
        this.con=con;
    }
    public String showUser(String name) {
        String query = "SELECT * FROM user WHERE first_name = ?";
        String result;

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setString(1, name);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                result = rs.getInt("user_id") + " " +
                         rs.getString("first_name") + " " +
                         rs.getString("last_name") + " " +
                         rs.getString("email") + " " +
                         rs.getString("phone_number") + " " +
                         rs.getString("address");
            } else {
                result = "No user found with the name " + name;
            }
        } catch (SQLException e) {
            result = "An error occurred: " + e.getMessage();
        }
        return result;
    }


    public void insertUser(String firstname, String lastname,String email,String phoneno,String
address){
        String query = "insert into user(first_name, last_name, email, phone_number, address)
values (?, ?, ?, ?, ?)";
        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setString(1, firstname);
            ps.setString(2, lastname);
            ps.setString(3, email);
            ps.setString(4, phoneno);
            ps.setString(5, address);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }
    public void updateUser(int userid,String firstname, String lastname,String email,String
phoneno,String address){
        String query = "update user set
first_name=?,last_name=?,email=?,phone_number=?,address=? where user_id=?";
        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setString(1, firstname);
            ps.setString(2, lastname);
            ps.setString(3, email);
            ps.setString(4, phoneno);
            ps.setString(5, address);
            ps.setInt(6, userid);
            ps.executeUpdate();
        }catch(SQLException e){
            System.err.println("a");
        }
    }
    public void deleteUser(int id){
        String query = "delete from user where user_id=?";
        try(PreparedStatement ps =  con.prepareStatement(query)){
            ps.setInt(1, id);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }
}
```

Address package:

```
package User;
import java.sql.*;

public class address {
    private Connection con;

    public address(Connection con) {
        this.con = con;
    }

    public String showAddress(int id) {
        String query = "SELECT * FROM address WHERE user_id=?";
        String result;

        try (PreparedStatement sc = con.prepareStatement(query)) {
            sc.setInt(1, id);
            try (ResultSet rs = sc.executeQuery()) {
                if (rs.next()) {
                    result = rs.getString("street") + " " +
                             rs.getString("city") + " " +
                             rs.getString("state") + " " +
                             rs.getString("postal_code");
                } else {
                    result = "No address records found.";
                }
            }
        } catch (SQLException e) {
            result = "An error occurred: " + e.getMessage();
        }

        return result;
    }

    public void insertAddress(int userId, String street, String city, String state, String postalCode){
        String query = "insert into address(user_id, street, city, state, postal_code) values (?, ?, ?,
?, ?)";
```

```
        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, street);
            ps.setString(3, city);
            ps.setString(4, state);
            ps.setString(5, postalCode);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }

    public void updateAddress(int userId, String street, String city, String state, String postalCode)
{
        String query = "update address set user_id=?, street=?, city=?, state=?, postal_code=?
where user_id=?";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, street);
            ps.setString(3, city);
            ps.setString(4, state);
            ps.setString(5, postalCode);
            ps.setInt(6, userId);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(" ");
        }
    }
    public void deleteUser(int id){
        String query = "delete from address where user_id=?";
        try(PreparedStatement ps =  con.prepareStatement(query)){
            ps.setInt(1, id);
            ps.executeUpdate();

        }catch(SQLException e){
            System.out.println("address");
        }
    }
}
```

Education package:

```
package User;
import java.math.BigDecimal;
import java.sql.*;

public class education {
    private Connection con;

    public education(Connection con) {
        this.con = con;
    }

    public String showEducation(int id) {
        String query = "SELECT * FROM education WHERE user_id=?";
        String result;

        try (PreparedStatement sc = con.prepareStatement(query)) {
            sc.setInt(1, id);
            ResultSet rs = sc.executeQuery();

            if (rs.next()) {
                result = rs.getInt("education_id") + " " +
                         rs.getInt("user_id") + " " +
                         rs.getString("degree") + " " +
                         rs.getString("institution_name") + " " +
                         rs.getInt("graduation_year") + " " +
                         rs.getBigDecimal("grade");
            } else {
                result = "No education records found for user with ID " + id;
            }
        } catch (SQLException e) {
            result = "An error occurred: " + e.getMessage();
        }

        return result;
    }



    public void insertEducation(int userId, String degree, String institutionName, int
graduationYear, BigDecimal grade)  {
        String query = "insert into education(user_id, degree, institution_name, graduation_year,
grade) values (?, ?, ?, ?, ?)";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, degree);
            ps.setString(3, institutionName);
            ps.setInt(4, graduationYear);
            ps.setBigDecimal(5, grade);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }

    public void updateEducation(int userId, String degree, String institutionName, int
graduationYear, BigDecimal grade){
        String query = "update education set user_id=?, degree=?, institution_name=?,
graduation_year=?, grade=? where user_id=?";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, degree);
            ps.setString(3, institutionName);
            ps.setInt(4, graduationYear);
            ps.setBigDecimal(5, grade);
            ps.setInt(6, userId);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(" ");
        }
    }
    public void deleteUser(int id){
        String query = "delete from education where user_id=?";
        try(PreparedStatement ps =  con.prepareStatement(query)){
            ps.setInt(1, id);
            ps.executeUpdate();

        }catch(SQLException e){
            System.out.println("education");
        }
    }
}
```

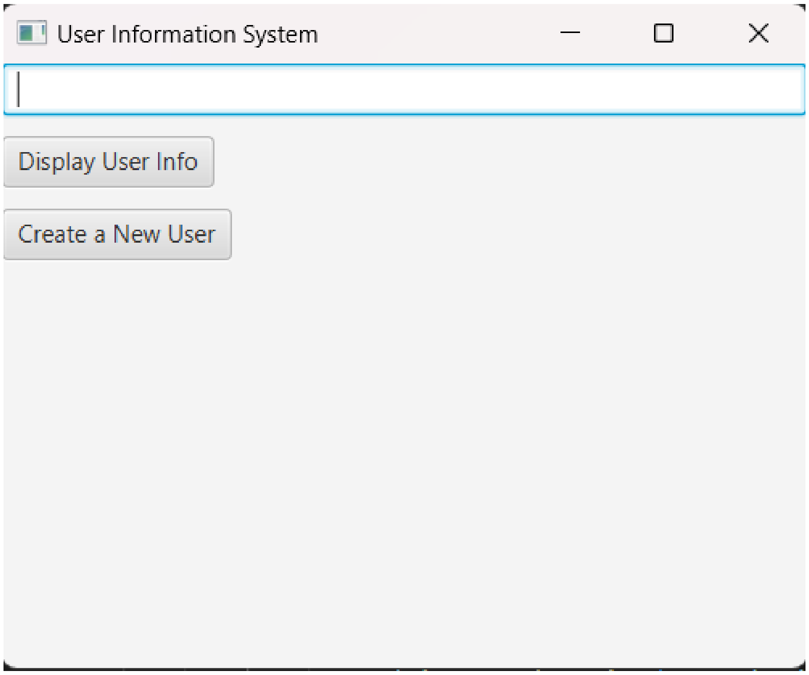
Emergencycontact package:

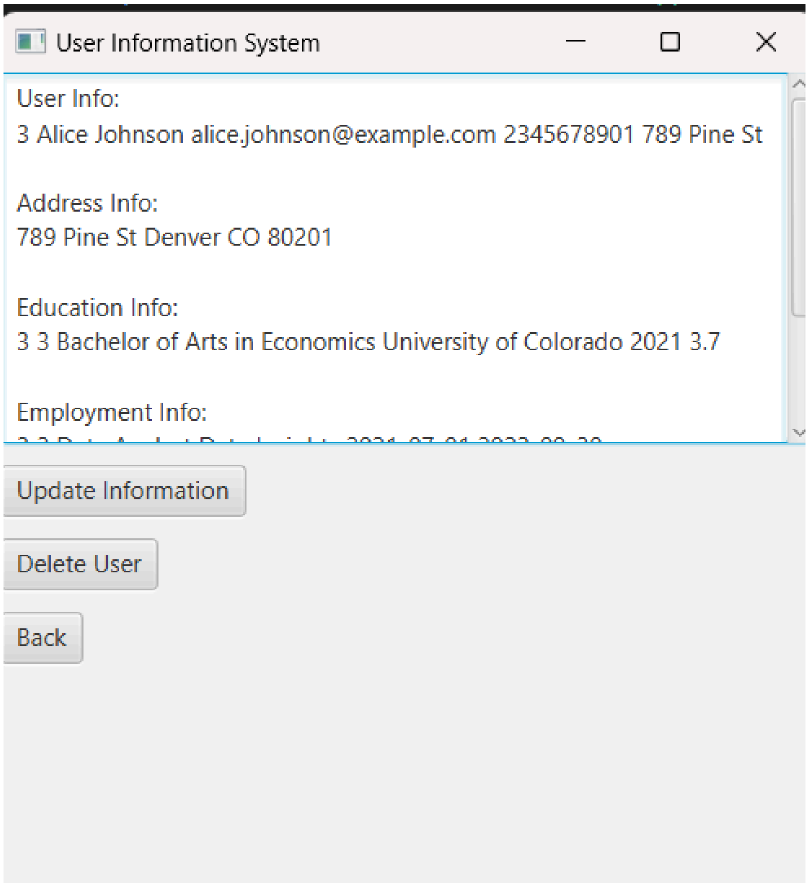
```
package User;
import java.sql.*;

public class emergencyContact {
    private Connection con;
    public emergencyContact(Connection con) {
        this.con = con;
    }

    public String showEmergencyContacts(int id) {
        String query = "SELECT * FROM emergency_contact WHERE user_id=?";
        String result;

        try (PreparedStatement sc = con.prepareStatement(query)) {
            sc.setInt(1, id);
            ResultSet rs = sc.executeQuery();

            if (rs.next()) {
                result = rs.getInt("contact_id") + " " +
                         rs.getInt("user_id") + " " +
                         rs.getString("contact_name") + " " +
                         rs.getString("relationship") + " " +
                         rs.getString("contact_phone") + " " +
                         rs.getString("contact_email");
            } else {
                result = "No emergency contact records found for user with ID " + id;
            }
        } catch (SQLException e) {
            result = "An error occurred: " + e.getMessage();
        }

        return result;
    }



    public void insertEmergencyContact(int userId, String contactName, String relationship, String
contactPhone, String contactEmail)  {
        String query = "insert into emergency_contact(user_id, contact_name, relationship,
contact_phone, contact_email) values (?, ?, ?, ?, ?)";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, contactName);
            ps.setString(3, relationship);
            ps.setString(4, contactPhone);
            ps.setString(5, contactEmail);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }

    public void updateEmergencyContact(int userId, String contactName, String relationship,
String contactPhone, String contactEmail) {
        String query = "update emergency_contact set user_id=?, contact_name=?, relationship=?,
contact_phone=?, contact_email=? where contact_id=?";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, contactName);
            ps.setString(3, relationship);
            ps.setString(4, contactPhone);
            ps.setString(5, contactEmail);
            ps.setInt(6, userId);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(" ");
        }
    }
    public void deleteUser(int id){
        String query = "delete from emergency_contact where user_id=?";
        try(PreparedStatement ps =  con.prepareStatement(query)){
            ps.setInt(1, id);
            ps.executeUpdate();

        }catch(SQLException e){
            System.out.println("emergency_contact");
        }
    }
}
```

EmployementInformation package:

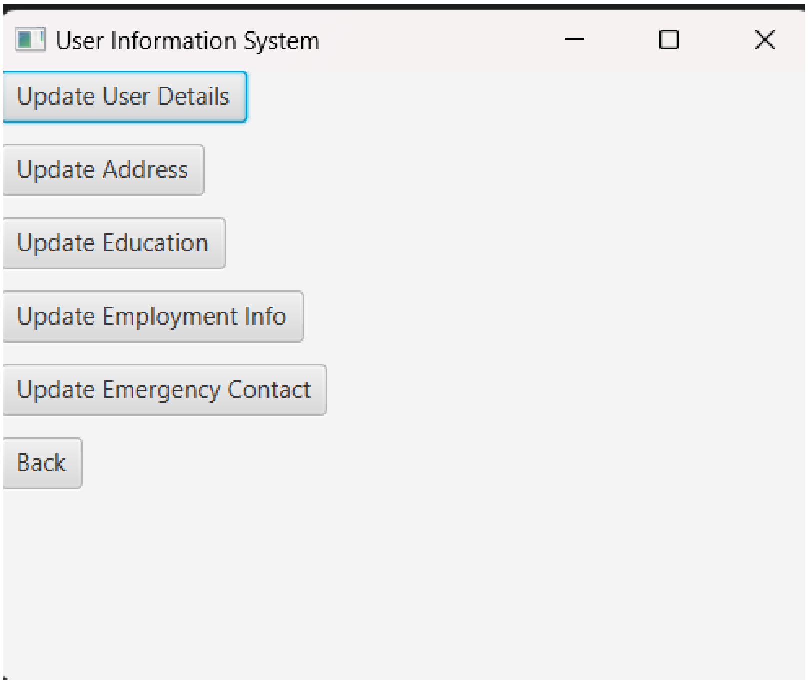
```
package User;
import java.sql.*;

public class employmentInformation {
    private Connection con;

    public employmentInformation(Connection con) {
        this.con = con;
    }
    public String showEmploymentInfo(int id) {
        String query = "SELECT * FROM employment_information WHERE user_id=?";
        String result;

        try (PreparedStatement sc = con.prepareStatement(query)) {
            sc.setInt(1, id);
            ResultSet rs = sc.executeQuery();

            if (rs.next()) {
                result = rs.getInt("employment_id") + " " +
                         rs.getInt("user_id") + " " +
                         rs.getString("job_title") + " " +
                         rs.getString("company_name") + " " +
                         rs.getDate("start_date") + " " +
                         rs.getDate("end_date");
            } else {
                result = "No employment records found for user with ID " + id;
            }
        } catch (SQLException e) {
            result = "An error occurred: " + e.getMessage();
        }

        return result;
    }



    public void insertEmploymentInfo(int userId, String jobTitle, String companyName, Date
startDate, Date endDate) {
        String query = "insert into employment_information(user_id, job_title, company_name,
start_date, end_date) values (?, ?, ?, ?, ?)";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, jobTitle);
            ps.setString(3, companyName);
            ps.setDate(4, startDate);
            ps.setDate(5, endDate);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(e);
        }
    }
    public void updateEmploymentInfo(int userId, String jobTitle, String companyName, Date
startDate, Date endDate){
        String query = "update employment_information set user_id=?, job_title=?,
company_name=?, start_date=?, end_date=? where user_id=?";

        try (PreparedStatement ps = con.prepareStatement(query)) {
            ps.setInt(1, userId);
            ps.setString(2, jobTitle);
            ps.setString(3, companyName);
            ps.setDate(4, startDate);
            ps.setDate(5, endDate);
            ps.setInt(6, userId);
            ps.executeUpdate();
        }catch(SQLException e){
            System.out.println(" ");
        }
    }
    public void deleteUser(int id){
        String query = "delete from employment_information where user_id=?";
        try(PreparedStatement ps =  con.prepareStatement(query)){
            ps.setInt(1, id);
            ps.executeUpdate();

        }catch(SQLException e){
            System.out.println("employment_information");
        }
    }
}
```

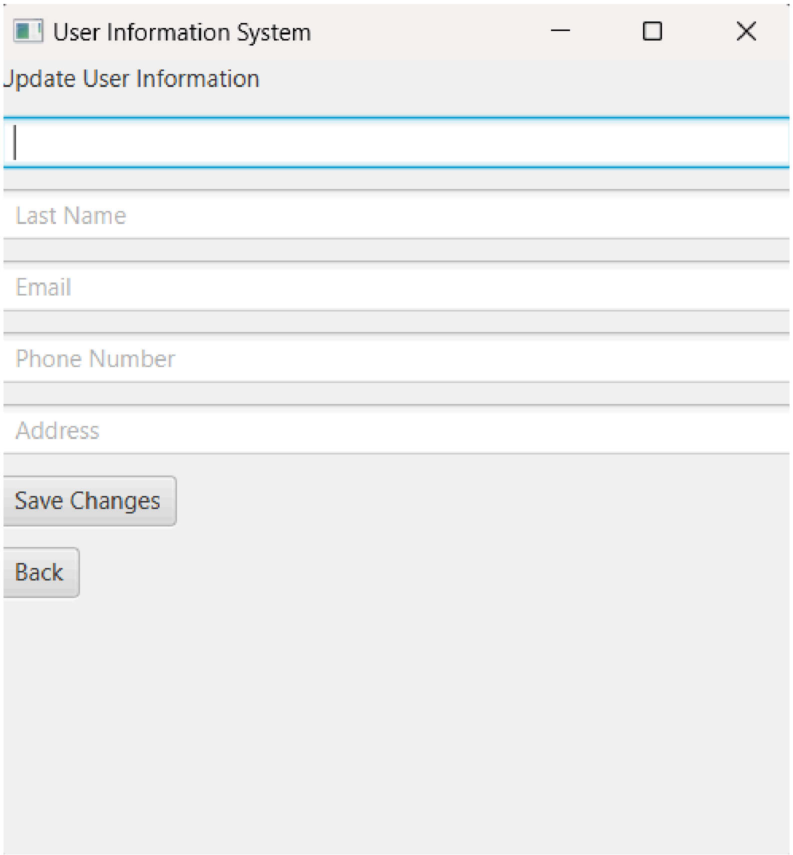
OUTPUT

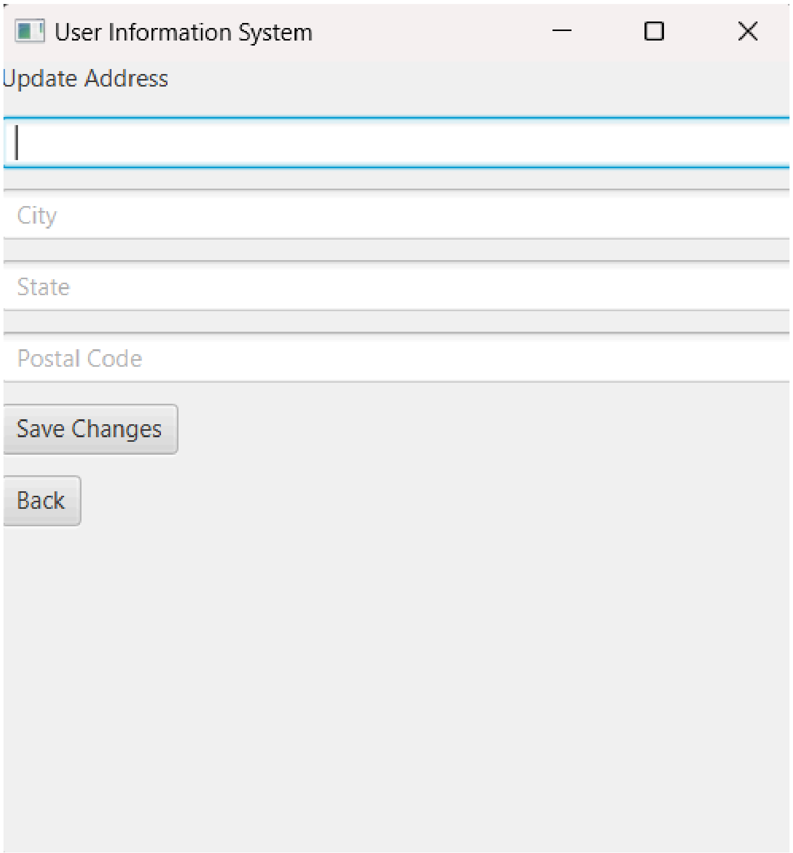


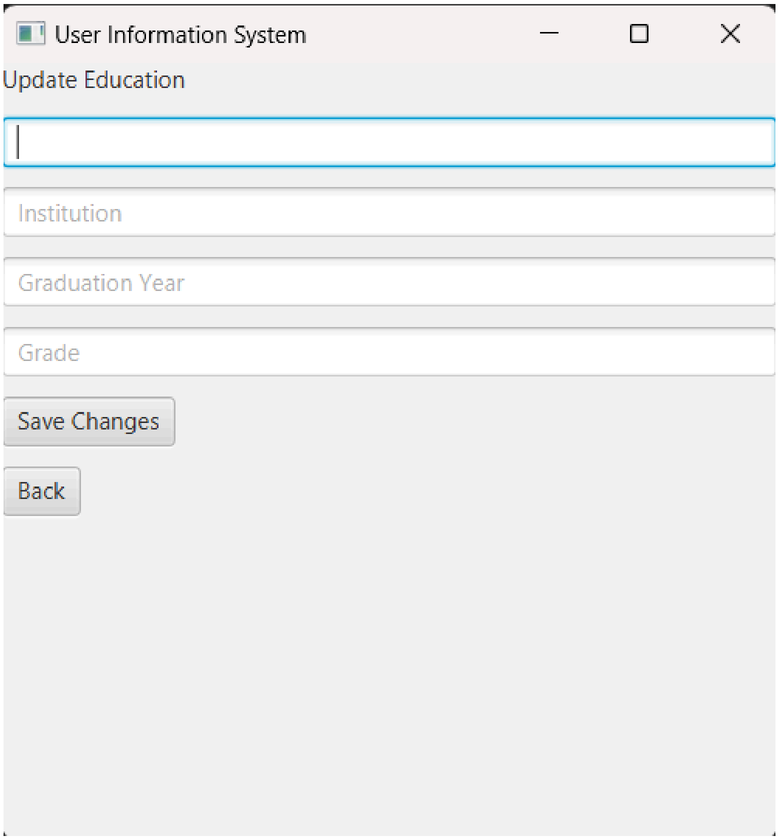


```
User Info:
3 Alice Johnson alice.johnson@example.com 2345678901 789 Pine St

Address Info:
789 Pine St Denver CO 80201

Education Info:
3 3 Bachelor of Arts in Economics University of Colorado 2021 3.7

Employment Info:
3 3 Data Analyst DataInsights 2021-07-01 2023-09-30
```


User Information System
Update User Details
Update Address
Update Education
Update Employment Info
Update Emergency Contact
Back


User Information System
Jpdate User Information
Last Name
Email
Phone Number
Address
Save Changes
Back


User Information System
Update Address
City
State
Postal Code
Save Changes
Back


User Information System
Update Education
Institution
Graduation Year
Grade
Save Changes
Back